A Tailored Real Time Temporal Logic for Specifying Requirements of Building Automation Systems

M. Kronenburg, R. Gotzhein, C. Peper

SFB 501 Report 16/1996

A Tailored Real Time Temporal Logic for Specifying Requirements of Building Automation Systems

Martin Kronenburg†, Reinhard Gotzhein‡, Christian Peper‡
{kronburg, gotzhein, peper}@informatik.uni-kl.de

Report 16/1996

Sonderforschungsbereich 501

†Efficient Algorithms Group
‡Computer Networks Group

Computer Science Department
University of Kaiserslautern
Postfach 3049
67653 Kaiserslautern
Germany

A Tailored Real Time Temporal Logic for Specifying Requirements of Building Automation Systems*

Martin Kronenburg, Reinhard Gotzhein, Christian Peper

Computer Science Department, University of Kaiserslautern
Postfach 3049, 67653 Kaiserslautern, Germany
{kronburg, gotzhein, peper}@informatik.uni-kl.de

Abstract

A tailored real time temporal logic for specifying requirements of building automation systems is introduced and analyzed. The logic features several new real time operators, which are chosen with regard to the application area. The new operators improve the conciseness and readability of requirements as compared to a general–purpose real time temporal logic. In addition, some of the operators also enhance the expressiveness of the logic. A number of properties of the new operators are presented and proven.

1 Introduction

This paper presents the formal background developed and used during a case study performed in the context of the Sonderforschungsbereich (SFB) 501 at the University of Kaiserslautern. A central goal of the SFB 501 is to devise methods and techniques for the generic development of large software systems [SFB94]. As an initial application domain, the area of building automation has been chosen as an experimental environment. Therefore our case study also took place within this domain.

One objective of this case study was the formalization of a given set of requirements written in natural language. Thereby we wanted to make some experiences concerning this activity in order to be able to make suggestions how the formalization can be performed more efficiently when doing it again. The formalized requirements are described in [PeGoKr96]. [GoKrPe96] investigates one requirement in detail. In this report we concentrate on the formal background we developed and used during the formalization process. Suggestions how our experiences can be exploited to improve

*This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Sonderforschungsbereich (SFB) 501 *Development of Large Systems with Generic Methods*.

the formalization task will be sketched in the conclusion and discussed in more detail in a separate report.

The informal requirements are part of a document called *problem description*. This document was the only input in our case study. The overall task mentioned in the problem description is "the delivery of a new control system that satisfies user needs of one standard room." This standard room is represented by a room of our university building which is equipped with several sensors, actuators, and other devices. Among the needs that have to be fulfilled by the control system are requirements referring to the control of temperature, humidity, light, ventilation, security, and safety.

At the beginning of our case study we were free to select any formal language appropriate for formalizing the informal problem description. Therefore, the first step we did was an analysis of the informal requirements in order to find out the used concepts. We made the following two main observations:

- The informal requirements are stated in a *property oriented* way.
- *Real time*, i.e. metric aspects of time, plays a major role in the informal requirements.

Since we wanted the formalization to be as close as possible to the informal version, we decided to use a *real time temporal logic*. We employed a *logic* because such languages typically are property oriented (unlike, for instance, statecharts [Ha87] or SDL [BrHa93], which are operational). A *real time temporal* logic was chosen due to the second concept we detected.

During our case study, we have made the following observations:

- It is feasible to formalize typical requirements on building automation systems using real time temporal logic.
- Many of these requirements follow a small set of *patterns*.

By introducing tailored operators for the found typical patterns, we achieved two improvements. Firstly, the resulting specification became more concise leading to better readability. Secondly, the expressiveness of our logic was increased, allowing us to state more informal requirements formally.

The technique of adding temporal operators to enhance the expressiveness and intelligibility of traditional temporal logic has been employed before. In the case of linear non–metric temporal logic, additional operators can be found in [Bo82], [Go93], [Kr87], [La77], [La83], [Pn77], and [ScMe82]. In the case of branching non–metric temporal logic see for example [ClEmSi86], where the Computation Tree Logic (CTL) has been introduced. More complex extensions are the interval logic [ScMeVo83], [Me88], and in the case of linear metric temporal logic the Duration Calculus [ChHoRa91],[HaCh92] and MTL–$\int$ [LaHo95].

In this paper, we formally introduce the typical patterns we have detected, define the tailored temporal operators we used, and discuss several of their properties (Section

3). In Section 2, we describe the underlying general purpose temporal logic. In Section 4, we discuss some positive consequences of the fact that in order to formalize the requirements it is sufficient to use only a small set of typical patterns. Based on this, we point out future work. The appendix contains the proofs of the properties of the patterns presented in Sections 2 and 3.

2 Basic Real Time Temporal Logic

Before we introduce the (domain specific) requirement–patterns and the resulting new temporal operators, we will define the syntax and the semantics of the proposed *basic real time temporal logic* (bRTTL). In 2.1 we will list the main properties of bRTTL according to the criteria mentioned in [AlHe91]. Sections 2.2 and 2.3 deal with the syntax and the semantics of bRTTL, respectively.

2.1 Main Properties

bRTTL is based on the temporal logics described in [MaPn92],[MaPn93], and [AlHe90]. The following list contains the main properties of bRTTL as a temporal logic:

1. bRTTL is a *propositional temporal logic.*
   This means that the underlying logic used to express the time independent parts is the propositional calculus.

2. bRTTL is a *linear–time logic.*
   This means that bRTTL is interpreted over linear structures of states.

3. bRTTL employs the *temporal operators* $\Box, \Diamond, \mathcal{W}, \blacksquare, \blacklozenge$.

4. In order to express timing requirements in a formula, bRTTL employs *time–bounded* variants of some of the temporal operators.

5. In bRTTL, the *time domain* is the set of non–negative real numbers $\mathbf{R}_0^+$.

2.2 Syntax

Definition 1 Syntax of bRTTL
Let $\mathcal{P}$ be the set of all propositional formulae over a set $\mathcal{V}$ of propositional variables and the propositional operators $\neg, \wedge, \vee \rightarrow, \leftrightarrow$. Further let $\tau \in \mathbf{R}^+$.
The set $\mathcal{F}$ of the correct temporal formulae of bRTTL *is the minimal set fulfilling the following conditions:*

(i) $\mathcal{P} \subseteq \mathcal{F}$

(ii) If $\varphi, \psi \in \mathcal{F}$ *then also*

- $\neg\varphi, \varphi \vee \psi, \varphi \wedge \psi, \varphi \rightarrow \psi, \varphi \leftrightarrow \psi \in \mathcal{F}$
- $[\varphi], \Box\varphi, \Diamond\varphi, \blacksquare\varphi, \blacklozenge\varphi, \varphi\mathcal{W}\psi \in \mathcal{F}$
- $\Box_{\leq\tau}\varphi, \Diamond_{\leq\tau}\varphi, \blacksquare_{\leq\tau}\varphi, \blacklozenge_{\leq\tau}\varphi \in \mathcal{F}$

$p \in \mathcal{P}$ *is called a* state formula.

The operators are named:

$[\,]$	action–operator
$\Box$	henceforth–operator
$\Box_{\leq}$	indexed henceforth–operator
$\Diamond$	eventually–operator
$\Diamond_{\leq}$	indexed eventually–operator
$\blacksquare$	has–always–been–operator
$\blacksquare_{\leq}$	indexed has–always–been–operator
$\blacklozenge$	once–operator
$\blacklozenge_{\leq}$	indexed once–operator
$\mathcal{W}$	waiting–for–operator

Note, that only positive upper bounds "$\leq \tau$" for the timing restrictions of the operators are allowed. This restriction is made for the sake of simplicity, especially the definition of the semantics of the formulae will be shorter.

2.3 Semantics

To define the semantics of the formulae of bRTTL, we use a general model for real time systems based on the one proposed in [AlHe91].

Definition 2 State, timed state sequence, trace
Let $\mathcal{V}$ *and* $\mathcal{P}$ *as in Definition 1.*

a) A state *is a function* $\sigma : \mathcal{V} \rightarrow \{0, 1\}$.
It extends to $\sigma : \mathcal{P} \rightarrow \{0, 1\}$ *in the usual way.*
The set of all states is denoted as Σ.

b) A timed state sequence ρ *is a function*

$$\rho : \mathbf{R_0^+} \rightarrow \Sigma$$

c) A timed state sequence ρ *is a* trace *if there exists an interval sequence* $\mathcal{I} = I_0, I_1, \ldots$ *with:*

(i) $\forall i, i \in \mathbf{N} : I_i = [a_i, b_i)$ *with* $a_i \in \mathbf{R_0^+}, b_i \in \mathbf{R}^+ \cup \{\infty\}$[§]$, a_i < b_i$

[§] ∞ has the usual properties, e.g.: $\forall x, x \in \mathbf{R} : x < \infty$

(ii) $\forall i, i \in \mathbf{N} : b_i \neq \infty \rightsquigarrow b_i = a_{i+1}$

(iii) $\forall i, i \in \mathbf{N} : \forall t \forall t', t, t' \in I_i : \rho(t) = \rho(t')$

(iv) $\forall t, t \in \mathbf{R}_0^+ : \exists i, i \in \mathbf{N} : t \in I_i$

(v) *If ρ is not constant from any point T, i.e.: $\forall t, t \in \mathbf{R}_0^+ : \exists t', t' > t : \rho(t) \neq \rho(t')$, then: $\forall i, i \in \mathbf{N} : \forall \bar{t}, \bar{t} \in I_i : \forall \hat{t}, \hat{t} \in I_{i+1} : \rho(\bar{t}) \neq \rho(\hat{t})$.*

(vi) *If ρ is constant from a point T, i.e.: $\exists t, t \in \mathbf{R}_0^+ : \forall t', t' \geq t : \rho(t) = \rho(t')$, then: $\exists n : \mathcal{I} = I_0, I_1, \ldots, I_n$ and $\forall i, i \in \{0, \ldots, n-1\} : \forall \bar{t}, \bar{t} \in I_i : \forall \hat{t}, \hat{t} \in I_{i+1} : \rho(\bar{t}) \neq \rho(\hat{t})$.*

Such an interval sequence $\mathcal{I}$ is called compatible *with ρ.*
The set of all traces is denoted as Π.

Condition (i) excludes *singular* intervals, i.e. intervals of type $[c, c]$, and other kinds of intervals, e.g. (a_i, b_i); (ii) guarantees that two neighboring intervals I_i and I_{i+1} are adjacent; (iii) guarantees that the state is invariant during each single interval I_i. Condition (iv) (together with (iii)) excludes Zeno–sequences of states; i.e. an infinite number of different states during a finite period of time is not allowed. In [LaHo95] this is called *finite variability*. Conditions (v) and (vi) guarantee that each interval I_i of the sequence $\mathcal{I}$ is a maximum interval in the sense that it ends if and only if the state changes. Due to these two conditions, there is for each timed state sequence at most one interval sequence $\mathcal{I}$ fulfilling (i) to (vi). Note, that *propositional* formulae have the same truth value during an interval I_i of $\mathcal{I}$. As we will see in Section 2.4, this can not be guaranteed for every bRTTL formula.

Now we can formally define the semantics of the temporal formulae of bRTTL:

Definition 3 Semantics of bRTTL
Let $\rho : \mathbf{R}_0^+ \to \Sigma$ be a trace and $\mathcal{I} = I_0, I_1, \ldots$ the interval sequence compatible with ρ. Further let $\varphi, \psi \in \mathcal{F}$ and $t, \tau \in \mathbf{R}^+$.
Then the satisfaction relation $\models$ is defined as follows:

(i) $(\rho, t) \models \varphi$ *iff* $\rho(t)(\varphi) = 1$ *if* $\varphi \in \mathcal{P}$

(ii) $\neg, \wedge, \vee, \rightarrow$, *and* $\leftrightarrow$ *are interpreted as usual.*

(iii) $(\rho, t) \models [\varphi]$ *iff* $(t = 0$ *and* $(\rho, 0) \models \varphi)$ *or* $(t > 0$ *and* $(\rho, t) \models \varphi$ *and* $\exists t', 0 \leq t' < t : \forall t'', t' \leq t'' < t : (\rho, t'') \models \neg\varphi)$

(iv) $(\rho, t) \models \Box\varphi$ *iff* $\forall t', t' \geq t$: $(\rho, t') \models \varphi$

(v) $(\rho, t) \models \Diamond\varphi$ *iff* $\exists t', t' \geq t$: $(\rho, t') \models \varphi$

(vi) $(\rho, t) \models \blacksquare\varphi$ *iff* $\forall t', 0 \leq t' \leq t : (\rho, t') \models \varphi$

(vii) $(\rho, t) \models \blacklozenge\varphi$ *iff* $\exists t', 0 \leq t' \leq t : (\rho, t') \models \varphi$

(viii) $(\rho, t) \models \varphi \mathcal{W} \psi$ *iff* $(\rho, t) \models \Box\varphi$ *or* $(\exists \tilde{t}, \tilde{t} \geq t : (\rho, \tilde{t}) \models \psi$ *and* $\forall t', t \leq t' < \tilde{t} : (\rho, t') \models \varphi)$

(ix) $(\rho, t) \models \Box_{\leq\tau}\varphi$ *iff* $\forall t', t \leq t' \leq t + \tau : (\rho, t') \models \varphi$

(x) $(\rho, t) \models \Diamond_{\leq\tau}\varphi$ *iff* $\exists t', t \leq t' \leq t + \tau : \rho_j \models \varphi$

(xi) $(\rho, t) \models \blacksquare_{\leq\tau}\varphi$ *iff* $\forall t', t_{low} \leq t' \leq t : (\rho, t') \models \varphi$ *and* $t_{low} = \max\{0, t - \tau\}$

(xii) $(\rho, t) \models \blacklozenge_{\leq\tau}\varphi$ *iff* $\exists t', t_{low} \leq t' \leq t : (\rho, t') \models \varphi$ *and* $t_{low} = \max\{0, t - \tau\}$

(xiii) $\rho \models \varphi$ *iff* $(\rho, 0) \models \varphi$

Based on the satisfaction relation $\models$ we define for every trace ρ a function

$$\rho_f : \mathbf{R_0^+} \times \mathcal{F} \to \{0, 1\}$$

$$\rho_f(t)(\varphi) = \begin{cases} 1 & : \quad (\rho, t) \models \varphi \\ 0 & : \quad (\rho, t) \not\models \varphi \end{cases}$$

2.4 Short Discussion of bRTTL

Note that there exist temporal formulae having different truth values during an interval I_i of an interval sequence $\mathcal{I}$. A simple one is the formula $[\varphi]$ which is valid exactly at one point. Another example is given by Figure 1. The corresponding interval sequence to the considered trace ρ is: $I_0 = [0, 1), I_1 = [1, 4), I_2 = [4, \infty)$. Then the formula $\Box_{\leq 2}\varphi$ is valid at $t = 1$ but not at $t = 3$.

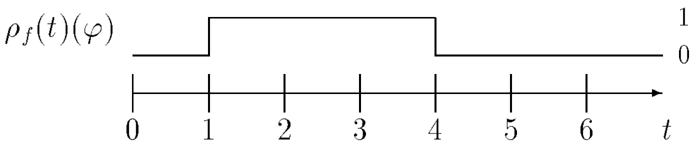

Figure 1: example 1

In the following we present and discuss briefly some relationships of the indexed operators and several properties of the waiting–for–operator. The following notations are used:

- If a formula φ is *equivalent* to a formula ψ, i.e. $\forall \rho, \rho \in \Pi : \rho \models \varphi$ iff $\rho \models \psi$, we write $\models \varphi \leftrightarrow \psi$.
- If ψ is a *logical consequence* of φ, i.e. $\forall \rho, \rho \in \Pi : \rho \models \varphi$ implies $\rho \models \psi$, we write $\models \varphi \to \psi$.
- If ψ is not a logical consequence of φ, we write $\not\models \varphi \to \psi$.

Let $\varphi, \varphi_1, \varphi_2, \varphi_3 \in \mathcal{F}$ and $\tau, \tau_1, \tau_2 \in \mathbf{R}^+$ in the following.

Similar to the unindexed versions of the operators, $\Box_{\leq}, \Diamond_{\leq}$ and $\blacksquare_{\leq}, \blacklozenge_{\leq}$ are *dual* operators, respectively:

$$\models \Box_{\leq\tau}\varphi \leftrightarrow \neg\Diamond_{\leq\tau}\neg\varphi \tag{IO1}$$

$$\models \blacksquare_{\leq\tau}\varphi \leftrightarrow \neg\blacklozenge_{\leq\tau}\neg\varphi \tag{IO2}$$

Furthermore, as for the unindexed versions, the box–operators, $\Box_{\leq}$ and $\blacksquare_{\leq}$, are stronger than the diamond–operators, $\Diamond_{\leq}$ and $\blacklozenge_{\leq}$, respectively. This relationship is independent of the time bounds of the operators:

$$\models \Box_{\leq\tau_1}\varphi \rightarrow \Diamond_{\leq\tau_2}\varphi \tag{IO3}$$

$$\models \blacksquare_{\leq\tau_1}\varphi \rightarrow \blacklozenge_{\leq\tau_2}\varphi \tag{IO4}$$

With respect to the waiting–for–operator, we present several sufficient conditions for the validity of $\varphi_1\mathcal{W}\varphi_2$, ($\mathcal{W}1$ to $\mathcal{W}5$), and show that the $\mathcal{W}$–operator is not transitive, ($\mathcal{W}6$). The sufficient conditions offer alternatives if the requirement $\varphi_1\mathcal{W}\varphi_2$ has to be realized. The proofs can be found in Appendix A. Tables 1 and 2 display the relations between different combinations of the $\mathcal{W}$–operator and the propositional operators $\wedge$ and $\vee$.

These tables as well as the ones presented in Section 3.3.1 have to be read in the following way. Each entry describes if a specific formula $\varphi \rightarrow \psi$ is valid. The formula standing in the first column always represents the formula φ, i.e. the premise. The formula in the uppermost row represents the formula ψ, i.e. the conclusion. A "+" points out that this instantiation of $\varphi \rightarrow \psi$ is valid, a "-" points out that it is not valid. These tables are needed to prove several properties of the new operators introduced in the following.

Sufficient conditions:

$$\models \varphi_2 \rightarrow (\varphi_1\mathcal{W}\varphi_2) \tag{$\mathcal{W}1$}$$

$$\models \Box\varphi_1 \rightarrow (\varphi_1\mathcal{W}\varphi_2) \tag{$\mathcal{W}2$}$$

$$\models ((\varphi_1\mathcal{W}\varphi_3) \wedge \Box(\varphi_3 \rightarrow \varphi_2)) \rightarrow (\varphi_1\mathcal{W}\varphi_2) \tag{$\mathcal{W}3$}$$

$$\models ((\varphi_1\mathcal{W}\varphi_2) \wedge ((\varphi_1 \rightarrow \varphi_3)\mathcal{W}\varphi_2)) \rightarrow (\varphi_3\mathcal{W}\varphi_2) \tag{$\mathcal{W}4$}$$

$$\models ((\varphi_1\mathcal{W}\neg\varphi_2) \wedge (\varphi_2\mathcal{W}\neg\varphi_3)) \rightarrow (\varphi_1\mathcal{W}\neg\varphi_3) \tag{$\mathcal{W}5$}$$

$\mathcal{W}$ is not transitive:

$$\not\models ((\varphi_1\mathcal{W}\varphi_2) \wedge (\varphi_2\mathcal{W}\varphi_3)) \rightarrow (\varphi_1\mathcal{W}\varphi_3) \tag{$\mathcal{W}6$}$$

	$(\varphi_1 \wedge \varphi_2)\mathcal{W}\psi$	$(\varphi_1 \vee \varphi_2)\mathcal{W}\psi$	$(\varphi_1\mathcal{W}\psi)\wedge (\varphi_2\mathcal{W}\psi)$	$(\varphi_1\mathcal{W}\psi)\vee (\varphi_2\mathcal{W}\psi)$
$(\varphi_1 \wedge \varphi_2)\mathcal{W}\psi$	+	+	+	+
$(\varphi_1 \vee \varphi_2)\mathcal{W}\psi$	-	+	-	-
$(\varphi_1\mathcal{W}\psi)\wedge (\varphi_2\mathcal{W}\psi)$	+	+	+	+
$(\varphi_1\mathcal{W}\psi)\vee (\varphi_2\mathcal{W}\psi)$	-	+	-	+

Table 1: Distributivity of $\mathcal{W}$, first argument

	$\varphi\mathcal{W}(\psi_1 \wedge \psi_2)$	$\varphi\mathcal{W}(\psi_1 \vee \psi_2)$	$(\varphi\mathcal{W}\psi_1) \wedge (\varphi\mathcal{W}\psi_2)$	$(\varphi\mathcal{W}\psi_1) \vee (\varphi\mathcal{W}\psi_2)$
$\varphi\mathcal{W}(\psi_1 \wedge \psi_2)$	+	+	+	+
$\varphi\mathcal{W}(\psi_1 \vee \psi_2)$	-	+	-	+
$(\varphi\mathcal{W}\psi_1)\wedge (\varphi\mathcal{W}\psi_2)$	-	+	+	+
$(\varphi\mathcal{W}\psi_1)\vee (\varphi\mathcal{W}\psi_2)$	-	+	-	+

Table 2: Distributivity of $\mathcal{W}$, second argument

3 Tailored Real Time Temporal Logic

If a temporal logic like bRTTL is applied to specify the properties of a real time system, appropriate operators have to be found for the formalization of a customer's natural language requirements. This problem arises because the operators of bRTTL are *general purpose operators*. They can be used in many different contexts to express a lot of diverse properties.

The search space spread by these general purpose operators and the possible combinations of them is *extremely large* (in general infinite). Furthermore, these general purpose operators offer no help for finding an appropriate formula in a special situation. Consequently, the search space is also *unstructured*.

We think that this problem can not be solved in general. But w.r.t. a specific domain, the formalization of the requirements can be supported and improved by *tailoring* the general purpose logic to this domain. We made the experience that introducing new operators is a powerful mechanism for tailoring a logic. Each newly defined operator should represent a specific property pattern occurring very often in the requirements of the domain. As it will be pointed out in this section, new operators are defined in order to achieve two different goals:

The first group of new operators are abbreviations of more complex operator combinations. Using these new operators, the requirements can be formalized in a more concise and intelligible way. The second group of new operators extends the expressiveness of bRTTL allowing to formally state requirements that can not be expressed using pure bRTTL.

Such new operators are no improvement if they only increase the already large search space. Since these operators are no general purpose operators but *tailored* for a special domain, they extend the search space in a controlled way suitable for this domain. They can be interpreted as a kind of *guidance*, because they focus on domain specific parts of the search space. Therefore one should proceed in the following way, when formalizing requirements:

Since the tailored operators represent domain specific requirement patterns they should be considered at first. If this set of operators is not sufficient for formalizing all requirements, the general purpose operators are still needed. Possibly, there are still some requirements not formalizable by bRTTL. In this case new operators can be introduced in order to increase the expressiveness of the logic.

The new operators and corresponding patterns structure the search space in a domain specific way. In order to diminish the search space, the kind of allowed subformulae occurring in these patterns can be restricted. For example, it could be sufficient to allow only propositional formulae (with respect to the selected level of abstraction).

Such a tailored temporal logic can not only simplify the formalization of requirements. It can also be helpful when reasoning about the formal requirements and when using them in subsequent steps of the system development. Some hints are given in Section 4.

In the following, we will concentrate on the tailoring of bRTTL for our case study. We will describe several typical patterns of requirements and the introduced temporal operators resulting in a tailored real time temporal logic (tRTTL). Subsequently, we will discuss some properties of the patterns and the new temporal operators.

[MaPn93] presents a excellent, general overview over the classification of properties in the case of an untimed temporal logic for specifying reactive systems. The general idea of tailoring a formal description technique is also studied in [Go93].

3.1 Tailored Temporal Operators

In this section we present the typical patterns and tailored operators. Each of these patterns is first of all illustrated by an example taken from our case study. Next, we will give a general formulation of each pattern in natural language and a corresponding formula. Finally, the new temporal operator is defined.

If the new operator is an abbreviation of the combination of several basic operators, the symbol "$\models\!\!=\!\!\dashv_{def}$" is used: a trace ρ satisfies the formula on the left side (containing the new operator) if and only if ρ satisfies the formula on the right side.

3.1.1 Delayed Implication Pattern

Example: In the case of hazardous conditions, i.e. heavy rain or storm, the windows have to be closed as fast as possible.

General formulation of this pattern in natural language:

> *Every time property φ holds constantly for τ time units, property ψ holds at least after this time and then ψ will be valid at least as long as property φ is valid.*

The formal version of this pattern is: $\Box(\Diamond_{\leq\tau}(\psi\mathcal{W}\neg\varphi))$

Example: $\Box(\Diamond_{\leq\tau}(\texttt{WindowClosed}\,\mathcal{W}\neg\texttt{HazardousCondition}))$

Since this pattern was used very often, we wanted to have an operator expressing the relationship between φ and ψ in a more intuitive and concise way. Therefore we define the new *delayed implication operator* $\Mapsto_{\leq}$:

$$\boxed{\varphi \Mapsto_{\leq\tau} \psi \models\dashv_{def} \Diamond_{\leq\tau}(\psi\mathcal{W}\neg\varphi)}$$

Using the new operator $\Mapsto_{\leq}$ the example is written as

$$\Box(\texttt{HazardousCondition} \Mapsto_{\leq\tau} \texttt{WindowClosed})$$

We think that this kind of notation displays the relationship between the two predicates in a more intuitive and concise way, namely that `WindowClosed` depends on `HazardousCondition`. This dependence between the left and the right side of the operator $\Mapsto$ is briefly discussed in Section 3.3.2.

3.1.2 Delayed Equivalence Pattern

The delayed implication pattern displays only the dependence of a formula ψ on a formula φ. In some situations an extension of this pattern can be used that expresses also the same relationship between $\neg\psi$ and $\neg\varphi$.

Example: If in the case of hazardous conditions the windows that can be handled only manually are open, the user has to be warned as long as both pre–conditions are fulfilled.

General formulation of this pattern in natural language:

> *Every time property φ $(\neg\varphi)$ holds constantly for τ time units, property ψ $(\neg\psi)$ holds at least after this time and then ψ $(\neg\psi)$ will be valid at least as long as property φ $(\neg\varphi)$ is valid.*

The formal version of this pattern is: $\Box((\varphi \Mapsto_{\leq\tau} \psi) \wedge (\neg\varphi \Mapsto_{\leq\tau} \neg\psi))$

In order to abbreviate this expression, we define the *delayed equivalence operator*:

$$\boxed{\varphi \Leftrightarrow_{\leq\tau} \psi \models\!\!\!=\!\!\!|_{def} (\varphi \mapsto_{\leq\tau} \psi) \wedge (\neg\varphi \mapsto_{\leq\tau} \neg\psi)}$$

Example: $\Box((\texttt{HazardousCondition} \wedge \texttt{ManualWindowOpen}) \Leftrightarrow_{\leq\tau} \texttt{WarnedUser})$

Note, that the reaction time τ has to be the same for both dependencies, $\varphi \mapsto_{\leq\tau} \psi$ and $\neg\varphi \mapsto_{\leq\tau} \neg\psi$. Otherwise, the minimum of the two different reaction times can be taken. This was sufficient in our case study.

3.1.3 Limited Invariance Pattern

The following limited invariance pattern represents a property that prevents a system from oscillating. We think that this pattern can be used in many other domains, too.

Example: The window should not alternate constantly from open to close and from close to open.

General formulation of this pattern in natural language:

> *Each time property p becomes valid (invalid), it will be constantly valid (invalid) for at least τ time units.*

The formal version of this pattern is: $\Box([\varphi] \rightarrow \Box_{\leq\tau}\varphi)$

As an abbreviation we define the *limited invariance operator* $\bigtriangledown_{\leq}$:

$$\boxed{\bigtriangledown_{\leq\tau} \varphi \models\!\!\!=\!\!\!|_{def} [\varphi] \rightarrow \Box_{\leq\tau}\varphi}$$

Example: $\Box \bigtriangledown_{\leq\tau} \texttt{WindowOpen}$ and $\Box \bigtriangledown_{\leq\tau} \texttt{WindowClose}$

3.1.4 Accumulated Invariance Pattern

This pattern differs from the already introduced ones in the way that it cannot be defined syntactically by a bRTTL–formula. Instead, we have to define a new operator from scratch, i.e. in terms of traces. Thus, it increases the expressiveness of the logic.

Example: During one hour the windows have to be open for at least ten minutes to provide the room with sufficient fresh air.

General formulation of this pattern in natural language:

> *During each period of T time units, property φ is valid for at least τ time units.*

Let $p \in \mathcal{P}$. Given a trace ρ and a compatible interval sequence $\mathcal{I} = I_0, I_1, \ldots$ with $I_i = [a_i, b_i)$ the *accumulated invariance operator* $\bigoplus$ is defined as:

$$(\rho, t) \models \bigoplus_{\tau}^{T} p \quad \text{iff} \quad \sum_{j} (\min(b_j, t_i + T) - \max(a_j, t)) \geq \tau$$
$$\text{with } j \in \{k | (\rho, a_k) \models \varphi, \tilde{a} \leq a_k \leq t + T, t \in \tilde{I} = [\tilde{a}, \tilde{b})\}$$

We allow only propositional formulae $p \in \mathcal{P}$ as arguments of $\bigoplus$ because, as already mentioned, for this class of formulae it can be guaranteed that their truth value is constant during each interval. If $\tau > T$, then $\Box \bigoplus_{\tau}^{T} \varphi$ is not satisfiable, and if $\tau = T$, then $\Box \bigoplus_{\tau}^{T} \varphi$ is equivalent to $\Box\varphi$.

The formal version of this pattern is: $\Box \bigoplus_{\tau}^{T} \varphi$

Example: $\Box \bigoplus_{\tau}^{T}$ `WindowOpen`

A similar operator, namely $\int^r$, is introduced in [LaHo95] in order to extend the metric temporal logic (MTL) (see [AlHe90]) to deal with duration aspects. The $\int^r$–operator differs from $\bigoplus_{\tau}^{T}$ in the way that $\int^r p$ is the time $t \in \mathbf{R}_{\mathbf{0}}^{+}$ the formula p is valid during the next r time units, while $\bigoplus_{\tau}^{T} p$ is a truth value. Again, the $\int^r$–operator is a general purpose operator allowing to express many duration properties while $\bigoplus_{\tau}^{T}$ is a tailored operator expressing exactly one kind of duration property.

In [LaHo95] the authors also consider a special property, the so called *limited–duration property*, given by a formula of the form $\Box(\int^T p \geq \tau)$. This property is in the following sense dual to our accumulated invariance pattern:

$$\Box(\int^T p \geq \tau) \quad \text{iff} \quad \Box \bigoplus_{T-\tau}^{T} \neg p$$

3.1.5 Invariance Pattern

This is the only pattern where it is not necessary to define a new operator.

Example: The room temperature has always to be greater than 0°C.

General formulation of this pattern in natural language:

Property φ is always valid.

The formal version of this pattern is: $\Box\varphi$

Example: $\Box$ `RoomTemperatureGtZero`

The formal version of this pattern is often also called a *safety* formula if φ is a past formula (e.g. [MaPn93]).

The set $\mathcal{F}$ of bRTTL–formulae is extended in the natural way including $\Rightarrow_{\leq}$, $\Leftrightarrow_{\leq}$, $\nabla_{\leq}$, and $\bigoplus$ as additional operators. The resulting logic is called *tailored real time temporal logic (tRTTL).*

3.2 tRTTL in Our Case Study

During our case study we made the experience that the patterns described in the last section are sufficient for formalizing the requirements, and that this formulation is concise and intelligible. Table 3 shows how many requirements (at the end of the case study there were 23 formal requirements) we were able to formalize using the patterns.

pattern	times used
invariance	10
delayed implication	9
delayed equivalence	1
limited invariance	2
accumulated invariance	1
other formulae	0

Table 3: Patterns used in the case study

According to this table, two patterns appear to be very typical, namely the classical *invariance pattern* and the newly introduced *delayed implication pattern*. Especially, if the delayed equivalence pattern is formulated as two delayed implication patterns, 21 of 24 requirements can be expressed by just these two patterns.

3.3 Properties of tRTTL

In this section we will discuss several properties of the new operators. Some proofs are presented in the appendix.

3.3.1 Combinations of $\wedge$ and $\vee$ with $\bigoplus, \nabla_{\leq}, \Mapsto_{\leq}, \Leftrightarrow_{\leq}$

For each of the new operators, $\bigoplus, \nabla_{\leq}, \Mapsto_{\leq}, \Leftrightarrow_{\leq}$, we consider the distributivity w.r.t. the two propositional operators "$\wedge$" and "$\vee$".

Table 4 deals with the operator $\nabla_{\leq}$. Some typical proofs and some counter examples for the assertions of Table 4 are given in Appendix B.1.

Table 5 displays the combinations of $\bigoplus$ with $\wedge$ and $\vee$. It shows that there is an total order on the four formulae according to implication, $\bigoplus_{\tau}^{T}(\varphi \wedge \psi)$ is the strongest and $\bigoplus_{\tau}^{T}(\varphi \vee \psi)$ the weakest one. Since all these relations are obvious no proofs are given.

	$\nabla_{\leq_\tau}(\varphi \wedge \psi)$	$\nabla_{\leq_\tau}(\varphi \vee \psi)$	$\nabla_{\leq_\tau}\varphi \wedge \nabla_{\leq_\tau}\psi$	$\nabla_{\leq_\tau}\varphi \vee \nabla_{\leq_\tau}\psi$
$\nabla_{\leq_\tau}(\varphi \wedge \psi)$	+	-	-	+
$\nabla_{\leq_\tau}(\varphi \vee \psi)$	-	+	-	-
$\nabla_{\leq_\tau}\varphi \wedge \nabla_{\leq_\tau}\psi$	-	+	+	+
$\nabla_{\leq_\tau}\varphi \vee \nabla_{\leq_\tau}\psi$	-	+	-	+

Table 4: Distributivity of $\nabla_{\leq}$

	$\bigoplus_\tau^T(\varphi \wedge \psi)$	$\bigoplus_\tau^T(\varphi \vee \psi)$	$\bigoplus_\tau^T\varphi \wedge \bigoplus_\tau^T\psi$	$\bigoplus_\tau^T\varphi \vee \bigoplus_\tau^T\psi$
$\bigoplus_\tau^T(\varphi \wedge \psi)$	+	+	+	+
$\bigoplus_\tau^T(\varphi \vee \psi)$	-	+	-	-
$\bigoplus_\tau^T\varphi \wedge \bigoplus_\tau^T\psi$	-	+	+	+
$\bigoplus_\tau^T\varphi \vee \bigoplus_\tau^T\psi$	-	+	-	+

Table 5: Distributivity of $\bigoplus$

In contrast to the two already considered operators the remaining two, $\mapsto_{\leq}$ and $\Leftrightarrow_{\leq}$, have two arguments. Therefore for each operator two tables are given.

While the Tables 6 and 7 show that for the operator $\mapsto_{\leq}$ some non–trivial relations are valid, Table 8 displays only trivial relations holding for $\Leftrightarrow_{\leq}$ w.r.t. its first argument. In the case of the second argument of the operator $\Leftrightarrow_{\leq}$ one can only prove that the combination $(\varphi \Leftrightarrow_{\leq_\tau} \psi_1) \wedge (\varphi \Leftrightarrow_{\leq_\tau} \psi_2)$ is stronger than any other one (see Table 9).

	$(\varphi_1 \wedge \varphi_2) \mapsto_{\leq_\tau} \psi$	$(\varphi_1 \vee \varphi_2) \mapsto_{\leq_\tau} \psi$	$(\varphi_1 \mapsto_{\leq_\tau} \psi) \wedge$ $(\varphi_2 \mapsto_{\leq_\tau} \psi)$	$(\varphi_1 \mapsto_{\leq_\tau} \psi) \vee$ $(\varphi_2 \mapsto_{\leq_\tau} \psi)$
$(\varphi_1 \wedge \varphi_2) \mapsto_{\leq_\tau} \psi$	+	-	-	-
$(\varphi_1 \vee \varphi_2) \mapsto_{\leq_\tau} \psi$	+	+	+	+
$(\varphi_1 \mapsto_{\leq_\tau} \psi) \wedge$ $(\varphi_2 \mapsto_{\leq_\tau} \psi)$	+	-	+	+
$(\varphi_1 \mapsto_{\leq_\tau} \psi) \vee$ $(\varphi_2 \mapsto_{\leq_\tau} \psi)$	+	-	-	+

Table 6: Distributivity of $\mapsto_{\leq}$, first argument

The Tables 4–9 show that a distributivity rule is only valid for the second argument of the delayed implication operator $\mapsto_{\leq}$; i.e.:

$$\varphi \mapsto_{\leq_\tau} (\psi_1 \wedge \ldots \wedge \psi_n) \models\!\!\!\!= \!\!\!\dashv (\varphi \mapsto_{\leq_\tau} \psi_1) \wedge \ldots \wedge (\varphi \mapsto_{\leq_\tau} \psi_n)$$

	$\varphi \Mapsto_{\leq\tau} (\psi_1 \wedge \psi_2)$	$\varphi \Mapsto_{\leq\tau} (\psi_1 \vee \psi_2)$	$(\varphi \Mapsto_{\leq\tau} \psi_1) \wedge (\varphi \Mapsto_{\leq\tau} \psi_2)$	$(\varphi \Mapsto_{\leq\tau} \psi_1) \vee (\varphi \Mapsto_{\leq\tau} \psi_2)$
$\varphi \Mapsto_{\leq\tau} (\psi_1 \wedge \psi_2)$	+	+	+	+
$\varphi \Mapsto_{\leq\tau} (\psi_1 \vee \psi_2)$	-	+	-	-
$(\varphi \Mapsto_{\leq\tau} \psi_1) \wedge (\varphi \Mapsto_{\leq\tau} \psi_2)$	+	+	+	+
$(\varphi \Mapsto_{\leq\tau} \psi_1) \vee (\varphi \Mapsto_{\leq\tau} \psi_2)$	-	+	-	+

Table 7: Distributivity of $\Mapsto_{\leq\tau}$, second argument

	$(\varphi_1 \wedge \varphi_2) \Leftrightarrow_{\leq\tau} \psi$	$(\varphi_1 \vee \varphi_2) \Leftrightarrow_{\leq\tau} \psi$	$(\varphi_1 \Leftrightarrow_{\leq\tau} \psi) \wedge (\varphi_2 \Leftrightarrow_{\leq\tau} \psi)$	$(\varphi_1 \Leftrightarrow_{\leq\tau} \psi) \vee (\varphi_2 \Leftrightarrow_{\leq\tau} \psi)$
$(\varphi_1 \wedge \varphi_2) \Leftrightarrow_{\leq\tau} \psi$	+	-	-	-
$(\varphi_1 \vee \varphi_2) \Leftrightarrow_{\leq\tau} \psi$	-	+	-	-
$(\varphi_1 \Leftrightarrow_{\leq\tau} \psi) \wedge (\varphi_2 \Leftrightarrow_{\leq\tau} \psi)$	-	-	+	+
$(\varphi_1 \Leftrightarrow_{\leq\tau} \psi) \vee (\varphi_2 \Leftrightarrow_{\leq\tau} \psi)$	-	-	-	+

Table 8: Distributivity of $\Leftrightarrow_{\leq}$, first argument

Excluding the well–known property of the henceforth operator,

$$\Box(\varphi_1 \wedge \ldots \wedge \varphi_n) \models\!\dashv \Box\varphi_1 \wedge \ldots \wedge \Box\varphi_n,$$

in no other case of the patterns introduced in Section 3.1 a formula can be split into "smaller" ones.

3.3.2 Further Discussion of $\Mapsto_{\leq}$ and $\Leftrightarrow_{\leq}$

Since the delayed implication operator $\Mapsto_{\leq}$ seems to be a very useful one in our domain we will discuss and analyze it and its extension, $\Leftrightarrow_{\leq}$, in more detail. On the one hand we will present a sufficient precondition for the pattern $\Box(\varphi \Mapsto_{\leq\tau} \psi)$. On the other hand we will check if properties like transitivity or symmetry holding for the undelayed implication ($\rightarrow$) and equivalence ($\leftrightarrow$) operator, respectively, are still valid for the delayed versions. The proofs and counter examples are given in Appendix B.

The sufficient precondition we present includes a pattern often mentioned in the literature as the *bounded response pattern*, $\varphi \rightarrow \Diamond_{\leq\tau}\psi$. According to the valid formula DI1

	$\varphi \Leftrightarrow_{\leq\tau} (\psi_1 \wedge \psi_2)$	$\varphi \Leftrightarrow_{\leq\tau} (\psi_1 \vee \psi_2)$	$(\varphi \Leftrightarrow_{\leq\tau} \psi_1) \wedge$ $(\varphi \Leftrightarrow_{\leq\tau} \psi_2)$	$(\varphi \Leftrightarrow_{\leq\tau} \psi_1) \vee$ $(\varphi \Leftrightarrow_{\leq\tau} \psi_2)$
$\varphi \Leftrightarrow_{\leq\tau} (\psi_1 \wedge \psi_2)$	+	-	-	-
$\varphi \Leftrightarrow_{\leq\tau} (\psi_1 \vee \psi_2)$	-	+	-	-
$(\varphi \Leftrightarrow_{\leq\tau} \psi_1) \wedge$ $(\varphi \Leftrightarrow_{\leq\tau} \psi_2)$	+	+	+	+
$(\varphi \Leftrightarrow_{\leq\tau} \psi_1) \vee$ $(\varphi \Leftrightarrow_{\leq\tau} \psi_2)$	-	-	-	+

Table 9: Distributivity of $\Leftrightarrow_{\leq\tau}$, second argument

the delayed implication pattern can be regarded as a variation of the bounded response pattern tailored to the special requirements of our domain.

$$\models ((\varphi \rightarrow \Diamond_{\leq\tau}\psi) \wedge \Box(\psi \rightarrow (\psi \mathcal{W} \neg\varphi))) \rightarrow (\varphi \mapsto_{\leq\tau} \psi) \tag{DI1}$$

Furthermore, (DI1) points out a way a requirement $\varphi \mapsto_{\leq\tau} \psi$ can be realized by splitting it into a time dependent response part $\varphi \rightarrow \Diamond_{\leq\tau}\psi$ and a time independent continuous part $\Box(\psi \rightarrow (\psi \mathcal{W} \varphi))$.

The valid formulae (DI2) and (DI3) show that each precondition taken alone is not strong enough to imply $\varphi \mapsto_{\leq\tau} \psi$.

$$\not\models (\varphi \rightarrow \Diamond_{\leq\tau}\psi) \rightarrow (\varphi \mapsto_{\leq\tau} \psi) \tag{DI2}$$

$$\not\models \Box(\psi \rightarrow (\psi \mathcal{W} \neg\varphi)) \rightarrow (\varphi \mapsto_{\leq\tau} \psi) \tag{DI3}$$

The delayed implication pattern is transitive in the following sense:

$$\models (\Box(\varphi_1 \mapsto_{\leq\tau} \varphi_2) \wedge \Box(\varphi_2 \mapsto_{\leq\pi} \varphi_3)) \rightarrow \Box(\varphi_1 \mapsto_{\leq\tau+\pi} \varphi_3) \tag{DI4}$$

But in contrast to the undelayed version, the contraposition rule does not hold:

$$\not\models \Box(\varphi \mapsto_{\leq\tau} \psi) \leftrightarrow \Box(\neg\psi \mapsto_{\leq\tau} \neg\varphi) \tag{DI5}$$

A typical property, symmetry, holding for $\leftrightarrow$ is not valid for the delayed equivalence operator:

$$\not\models \Box(\varphi \Leftrightarrow_{\leq\tau} \psi) \rightarrow \Box(\psi \Leftrightarrow_{\leq\tau} \varphi) \tag{DE1}$$

(DI5) and (DE1) show that different roles can be associated with the two subformulae of $\mapsto_{\leq}$ and $\Leftrightarrow_{\leq}$. The subformula on the left side of these operators plays an active role while the formula on the right side is passive and only reacts if necessary.

4 Conclusions and Future Work

In this report, we have presented a tailored real time temporal logic for specifying requirements of building automation systems. The logic features several new real time operators, which are chosen with regard to the application area. The new operators improve the conciseness and readability of formal requirements as compared to a general–purpose real time temporal logic. In addition, some of the operators enhance the expressiveness of the logic.

In the course of our case study, we have found that most requirements follow a relatively small number of patterns. This appears to be another occurrence of the 90–10–rule: 90% of the requirements are instantiations of these patterns, while 10% have a different form. In this report, we have presented such patterns and corresponding requirements. We have shown that tRTTL supports the concise specification of these patterns by tailored operators.

When a set of typical requirement patterns for a given application domain is already known, this can be exploited in several ways. All the following mentioned benefits are based on the main concept of *pattern reuse*.

Firstly, the effort of specifying requirements may be substantially reduced, if most requirements can be formalized by instantiating predefined patterns. Secondly, requirement patterns may be analyzed prior to their instantiations, thus simplifying the validation of requirements resulting from pattern instantiation. For instance, conflict detection as well as correctness proofs may be supported that way. Thirdly, we expect that the solutions for requirements that follow the same pattern will exhibit a high degree of similarity, which increases the potential for reuse in subsequent development phases.

Further research should include the following:

- Further case studies should be performed w.r.t. building automation systems, but also in other application domains. We expect that this will reveal additional requirement patterns. Also, it will show whether the set of patterns discussed in this report forms a stable basis, and how much adaption is necessary w.r.t. other application areas.

- The formal foundations for specifying requirements using property-oriented techniques should be strengthened. FDTs such as tRTTL support the formal specification of single requirements which have to be composed into a complete requirements document. However, composition may lead to inconsistencies due to conflicting requirements. For example, there can be a situation where it is not possible to satisfy the requirements $\Box(\texttt{HazardousCondition} \mapsto_{\leq_T} \texttt{WindowClosed})$ and $\Box \bigoplus_t^T \texttt{WindowOpen}$ together. In general, conflicts should be detectable based on the specification of requirements or requirement patterns, and be resolved.

  Note that in general, additional information about the relationship among predicates, expressing physical conditions of the environment into which the system will be embedded, is required to detect conflicts. In the example above, the fact that the predicates `WindowClosed` and `WindowOpen` can not be valid at the same time has to be stated and fed into the conflict detection approach. Using tRTTL, this relationship can be stated as $\Box(\texttt{WindowClosed} \rightarrow \neg\texttt{WindowOpen})$.

- It should be investigated to what degree the reuse of requirement patterns may support subsequent activities of system development. This should answer the important question whether a reuse of requirement patterns indeed leads to a reuse of solutions.

Acknowledgments
We would like to thank the members of project D1 and Team 2 of the SFB 501, who were involved in the requirement collection and improvement, and T. Deiß and Prof. J. Avenhaus for valuable discussions.

References

[AlHe90] **Rajeev Alur, Thomas A. Henzinger**. *Real-time Logics: Complexity and Expressiveness.* In Proc. of 5th Ann. IEEE Symp. on Logic in Computer Science, p. 390-401, 1990.

[AlHe91] **Rajeev Alur, Thomas A. Henzinger**. *Logics and Models of Real Time: A Survey.* In J. W. de Bakker, C. Huizing, W. P. de Roever, G. Rozenberg (Eds.), REX-Workshop, Real-Time: Theory in Practice, LNCS 600, 1991.

[Bo82] **G. v. Bochmann**. *Hardware Specification with Temporal Logic: An Example.* In IEEE Transactions on Computers, Vol C-31, No. 3, p. 223-231, 1982.

[BrHa93] **Rolv Bræk and Øystein Haugen**. *Engineering Real Time Systems. An object–oriented methodology using SDL.* Prentice Hall, 1993.

[ChHoRa91] **Z. Chaochen, C. A. R. Hoare, A. P. Ravn**. *A Calculus of Durations.* In Inform. Process. Lett. 40, p. 269-276, 1991.

[ClEmSi86] **E. M. Clarke, E. A. Emerson, A. P. Sistla**. *Automatic Verification of Finite–State Concurrent Systems using Temporal Logic Specifications.* ACM TOPLAS Vol. 8, No. 2, p 244-263, 1986.

[Go93] **Reinhard Gotzhein** *Open Distributed Systems – On Concepts, Methods, and Design from a Logical Point of View.* Vieweg, 1993.

[GoKrPe96] **Reinhard Gotzhein, Martin Kronenburg, Christian Peper**. *Specifying and Reasoning about Generic Real–Time Requirements — A Case Study.* SFB 501 Report 15/96, University of Kaiserslautern, 1996.

[HaCh92] **M. R. Hansen, Z. Chaochen**. *Semantics and Completeness of Duration Calculus.* In J. W. de Bakker, C. Huizing, W. P. de Roever, G. Rozenberg (Eds.), REX-Workshop, Real-Time: Theory in Practice, LNCS 600, 1991.

[Ha87] **David Harel**. *Statecharts: A visual formalism for complex systems.* Science of Computer Programming, 8:231-274, 1987.

[Kr87] **F. Kröger**. *Temporal Logic of Programs.* EATCS Monographs on Theoretical Computer Science, VIII, Springer, 1987.

[La77] **Leslie Lamport**. *Proving the Correctness of Multiprocess Programs.* In IEEE Transaction on Software Engineering SE-3, 2, p. 125-143, 1977.

[La83] **Leslie Lamport**. *What good is temporal logic.* In R. E. A. Mason (Ed.), Information processing 83, IFIP, p. 657-668, 1983.

[LaHo95] **Yassine Lakhneche, Jozef Hooman**. *Metric temporal logic with durations.* Theoretical Computer Science, 138:169-199, 1995.

[MaPn92] **Zohar Manna, Amir Pnueli**. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer, 1992.

[MaPn93] **Zohar Manna, Amir Pnueli**. *Models for reactivity*. In Acta Informatica, 30:609-678, 1993.

[Me88] **P. M. Melliar–Smith**. *A Graphical Representation of Interval Logic*. In F. H. Vogt (Ed.), Concurrency 88, LNCS 335, p. 106-120, 1988.

[PeGoKr96] **Christian Peper, Reinhard Gotzhein, Martin Kronenburg**. *Formal Specification of Real–Time Requirements for a Building Automation System*. SFB 501 Report 1/97, University of Kaiserslautern, 1996.

[Pn77] **Amir Pnueli**. *The Temporal Logic of Programs*. In 19th Annual Symposium on Foundations of Computer Science, 1977.

[ScMe82] **R. L. Schwartz, P. M. Melliar–Smith**. *From State Machines to Temporal Logic: Specification Methods for Protocol Standards*. In IEEE Transactions on Communications, No. 12, p. 2486-2496, 1982.

[ScMeVo83] **R. L. Schwartz, P. M. Melliar–Smith, F. H. Vogt**. *Interval Logic: A Higher–Level Temporal Logic for Protocol Specification*. In H. Rudin, C. H. West (Eds.), Protocol Specification, Testing, and Verification III, p. 3-18, 1983.

[SFB94] *Development of Large Systems with Generic Methods, Project Proposal, SFB 501*. University of Kaiserslautern, 1994.

A Properties of the Waiting–For–Operator

This appendix provides the proofs of some properties of the waiting–for–operator presented in Section 2.3.

Let $\varphi, \varphi_1, \varphi_2, \varphi_3, \psi_1, \psi_2 \in \mathcal{F}$ for the rest of this appendix.

- $\models ((\varphi_1 \mathcal{W} \varphi_3) \wedge \Box(\varphi_3 \rightarrow \varphi_2)) \rightarrow (\varphi_1 \mathcal{W} \varphi_2)$ $\qquad (\mathcal{W}3)$

  To prove: For every trace ρ, one has to show:

  If $\rho \models \varphi_1 \mathcal{W} \varphi_3$ and $\rho \models \Box(\varphi_3 \rightarrow \varphi_2)$ then $\rho \models \varphi_1 \mathcal{W} \varphi_2$.

  Proof:

  Case 1: $\rho \models \Box\varphi_1$:

  According to the definition of $\mathcal{W}$ then also: $\rho \models \varphi_1 \mathcal{W} \varphi_2$.

  Case 2: $\rho \not\models \Box\varphi_1$:

  Then: $\exists \tilde{t}, \tilde{t} \geq 0 : (\rho, \tilde{t}) \models \neg\varphi_1$ and $\forall t', 0 \leq t' < \tilde{t} : (\rho, t') \models \varphi_1$.

  Now it is sufficient to prove: $\exists \bar{t}, 0 \leq \bar{t} \leq \tilde{t} : (\rho, \bar{t}) \models \varphi_2$.

  Due to $\rho \models \varphi_1 \mathcal{W} \varphi_3 : \exists \hat{t}, 0 \leq \hat{t} \leq \tilde{t} : (\rho, \bar{t}) \models \varphi_3$.

  Since $\rho \models \Box(\varphi_3 \rightarrow \varphi_2)$ is valid too, one gets especially $(\rho, \hat{t}) \models \varphi_3 \rightarrow \varphi_2$ and consequently $(\rho, \hat{t}) \models \varphi_2$. Let $\bar{t} = \hat{t}$.

  q.e.d.

- $\models ((\varphi_1 \mathcal{W} \varphi_2) \wedge ((\varphi_1 \rightarrow \varphi_3) \mathcal{W} \varphi_2)) \rightarrow (\varphi_3 \mathcal{W} \varphi_2)$ $\qquad (\mathcal{W}4)$

  To prove: For every trace ρ, one has to show:

  If $\rho \models \varphi_1 \mathcal{W} \varphi_2$ and $\rho \models (\varphi_1 \rightarrow \varphi_3) \mathcal{W} \varphi_2$ then $\rho \models \varphi_3 \mathcal{W} \varphi_2$.

  Proof:

  Case 1: $\rho \models \Box\varphi_3$:

  According to the definition of $\mathcal{W}$ then also: $\rho \models \varphi_3 \mathcal{W} \varphi_2$.

  Case 2: $\rho \not\models \Box\varphi_3$:

  Then: $\exists \tilde{t}, \tilde{t} \geq 0 : (\rho, \tilde{t}) \models \neg\varphi_3$ and $\forall t', 0 \leq t' < \tilde{t} : (\rho, t') \models \varphi_3$.

  Now it is sufficient to prove: $\exists \bar{t}, 0 \leq \bar{t} \leq \tilde{t} : (\rho, \bar{t}) \models \varphi_2$.

  Case 2.1: $\rho \models \Box\varphi_1$:

  Then: $(\rho, \tilde{t}) \models \neg(\varphi_1 \rightarrow \varphi_3)$ and

  $\forall t', 0 \leq t' < \tilde{t}, (\rho, t') \models \varphi_1 \rightarrow \varphi_3$

  Due to $\rho \models (\varphi_1 \rightarrow \varphi_3) \mathcal{W} \varphi_2 : \exists \hat{t}, 0 \leq \hat{t} \leq \tilde{t} : (\rho, \hat{t}) \models \varphi_2$.

  Let $\bar{t} = \hat{t}$.

  Case 2.2: $\rho \not\models \Box\varphi_1$:

  Then: $\exists \hat{t}, 0 \leq \hat{t} : (\rho, \hat{t}) \models \neg\varphi_1$ and

  $\forall t', 0 \leq t' < \hat{t} : (\rho, t') \models \varphi_1$.

  Case 2.2.1: $\hat{t} \leq \tilde{t}$:

  Due to $\rho \models \varphi_1 \mathcal{W} \varphi_2$:

  $\exists t'', 0 \leq t'' \leq \hat{t} \leq \tilde{t} : (\rho, t'') \models \varphi_2$. Let $\bar{t} = t''$.

Case 2.2.2: $\hat{t} > \tilde{t}$:

Then: $(\rho, \tilde{t}) \models \neg(\varphi_1 \rightarrow \varphi_3)$ and
$\forall t', 0 \leq t' \leq \tilde{t} : (\rho, t) \models \varphi_1 \rightarrow \varphi_3$.
Due to $\rho \models (\varphi_1 \rightarrow \varphi_3)\mathcal{W}\varphi_2$:
$\exists t'', 0 \leq t'' \leq \tilde{t} : (\rho, t'') \models \varphi_2$. Let $\bar{t} = t''$.

q.e.d.

- $\models ((\varphi_1\mathcal{W}\neg\varphi_2) \wedge (\varphi_2\mathcal{W}\neg\varphi_3)) \rightarrow (\varphi_1\mathcal{W}\neg\varphi_3)$ ($\mathcal{W}5$)

To prove: For every trace ρ, one has to show:
If $\rho \models \varphi_1\mathcal{W}\neg\varphi_2$ and $\rho \models \varphi_2\mathcal{W}\neg\varphi_3$ then $\rho \models \varphi_1\mathcal{W}\neg\varphi_3$.

Proof:

Case 1: $\rho \models \Box\varphi_1$:
According to the definition of $\mathcal{W}$ then also: $\rho \models \varphi_1\mathcal{W}\neg\varphi_3$.

Case 2: $\rho \not\models \Box\varphi_1$:
Then: $\exists\tilde{t}, \tilde{t} \geq 0 : (\rho, \tilde{t}) \models \neg\varphi_1$ and $\forall t', 0 \leq t' < \tilde{t} : (\rho, t') \models \varphi_1$.
Now it is sufficient to prove: $\exists\bar{t}, 0 \leq \bar{t} \leq \tilde{t} : (\rho, \bar{t}) \models \neg\varphi_3$.
Due to $\rho \models \varphi_1\mathcal{W}\neg\varphi_2$: $\exists\hat{t}, 0 \leq \hat{t} \leq \tilde{t} : (\rho, \hat{t}) \models \neg\varphi_2$ and
$\forall t', 0 \leq t' < \hat{t} : (\rho, t') \models \varphi_2$.
Due to $\rho \models \varphi_2\mathcal{W}\neg\varphi_3$: $\exists t'', 0 \leq t'' \leq \hat{t} : (\rho, t'') \models \neg\varphi_3$. Let $\bar{t} = t''$.

q.e.d.

- $\not\models ((\varphi_1\mathcal{W}\varphi_2) \wedge (\varphi_2\mathcal{W}\varphi_3)) \rightarrow (\varphi_1\mathcal{W}\varphi_3)$ ($\mathcal{W}6$)

Counter example:

Let $\varphi_2 \equiv$ `true` and $\varphi_1 \equiv \varphi_3 \equiv$ `false`, then for every trace ρ:
$\rho \models \varphi_1\mathcal{W}\varphi_2$ and $\rho \models \varphi_2\mathcal{W}\varphi_3$ and $\rho \not\models \varphi_1\mathcal{W}\varphi_3$.

- $\not\models ((\varphi\mathcal{W}\psi_1) \wedge (\varphi\mathcal{W}\psi_2)) \rightarrow (\varphi\mathcal{W}(\psi_1 \wedge \psi_2))$

This is the only property presented in Table 1 and Table 2 we will prove.

Counter example:

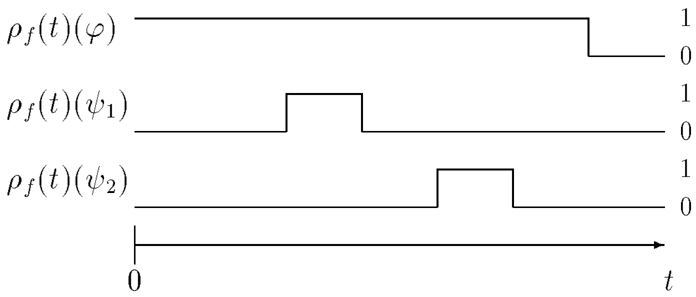

Figure 2: Counter example for $(\varphi\mathcal{W}\psi_1) \wedge (\varphi_1\mathcal{W}\psi_2)) \rightarrow (\varphi\mathcal{W}(\psi_1 \wedge \psi_2))$

B Properties of the new Operators

This appendix provides the proofs of some properties of the new operators presented in Section 3.

Let $\varphi, \varphi_1, \varphi_2, \varphi_3, \psi, \psi_1, \psi_2 \in \mathcal{F}$ and $\tau, \pi \in \mathbf{R}^+$ for the rest of this appendix.

B.1 Limited Invariance Operator

All non–trivial valid properties presented in Table 4 are proven. One counter example is given.

- $\models \nabla_{\leq\tau}(\varphi \wedge \psi) \rightarrow (\nabla_{\leq\tau}\varphi \vee \nabla_{\leq\tau}\psi)$

  Expansion of $\nabla_{\leq}$ leads to:

  $\models ([\varphi \wedge \psi] \rightarrow \Box_{\leq\tau}(\varphi \wedge \psi)) \rightarrow (([\varphi] \rightarrow \Box_{\leq\tau}\varphi) \vee ([\psi] \rightarrow \Box_{\leq\tau}\psi))$

  To prove: For every trace ρ, one has to show:

  If $\rho \models [\varphi \wedge \psi] \rightarrow \Box_{\leq\tau}(\varphi \wedge \psi)$ and $\rho \models [\varphi] \wedge [\psi]$
  then $\rho \models \Box_{\leq\tau}\varphi$ or $(\rho, t) \models \Box_{\leq\tau}\psi$.

  Proof:

  Due to $\rho \models [\varphi] \wedge [\psi]$: $\rho \models [\varphi \wedge \psi]$
  $\rightsquigarrow \rho \models \Box_{\leq\tau}(\varphi \wedge \psi)$
  $\rightsquigarrow \rho \models \Box_{\leq\tau}\varphi$ or $\rho \models \Box_{\leq\tau}\psi$

  q.e.d.

- $\models (\nabla_{\leq\tau}\varphi \wedge \nabla_{\leq\tau}\psi) \rightarrow (\nabla_{\leq\tau}(\varphi \vee \psi))$

  Expansion of $\nabla_{\leq}$ leads to:

  $\models (([\varphi] \rightarrow \Box_{\leq\tau}\varphi) \wedge ([\psi] \rightarrow \Box_{\leq\tau}\psi)) \rightarrow ([\varphi \vee \psi] \rightarrow \Box_{\leq\tau}(\varphi \vee \psi))$

  To prove: For every trace ρ, one has to show:

  If $\rho \models [\varphi] \rightarrow \Box_{\leq\tau}\varphi$ and $\rho \models [\psi] \rightarrow \Box_{\leq\tau}\psi$ and $\rho \models [\varphi \vee \psi]$
  then $\rho \models \Box_{\leq\tau}(\varphi \vee \psi)$

  Proof:

  Due to $\rho \models [\varphi \vee \psi]$ also: $\rho \models [\varphi]$ or $\rho \models [\psi]$
  $\rightsquigarrow \rho \models \Box_{\leq\tau}\varphi$ or $\rho \models \Box\psi$
  $\rightsquigarrow \rho \models \Box(\varphi \vee \psi)$

  q.e.d.

- $\models (\nabla_{\leq\tau}\varphi \vee \nabla_{\leq\tau}\psi) \rightarrow (\nabla_{\leq\tau}(\varphi \vee \psi))$

  Expansion of $\nabla_{\leq}$ leads to:

  $\models (([\varphi] \rightarrow \Box_{\leq\tau}\varphi) \vee ([\psi] \rightarrow \Box_{\leq\tau}\psi)) \rightarrow ([\varphi \vee \psi] \rightarrow \Box_{\leq\tau}(\varphi \vee \psi))$

  To prove: For every trace ρ, one has to show:

  If $(\rho \models [\varphi] \rightarrow \Box_{\leq\tau}\varphi$ or $\rho \models [\psi] \rightarrow \Box_{\leq\tau}\psi)$ and $\rho \models [\varphi \vee \psi]$
  then $\rho \models \Box_{\leq\tau}(\varphi \vee \psi)$.

Proof:

Due to $\rho \models [\varphi \vee \psi]$: $\rho \models [\varphi]$ or $\rho \models [\psi]$
$\leadsto \rho \models \Box_{\leq\tau}\varphi$ or $\rho \models \Box\psi$
$\leadsto \rho \models \Box(\varphi \vee \psi)$

q.e.d.

- $\not\models \nabla_{\leq\tau}(\varphi \vee \psi) \rightarrow (\nabla_{\leq\tau}\,\varphi \vee \nabla_{\leq\tau}\,\psi)$

  Counter example:

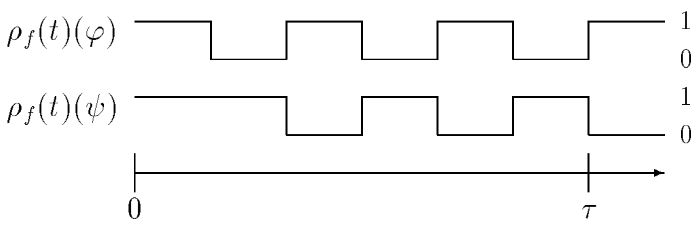

Figure 3: Counter example for $\nabla_{\leq\tau}(\varphi \vee \psi) \rightarrow (\nabla_{\leq\tau}\,\varphi \vee \nabla_{\leq\tau}\,\psi)$

B.2 Delayed Implication Operator

- $\models ((\varphi \Rrightarrow_{\leq\tau} \psi_1) \wedge (\varphi \Rrightarrow_{\leq\tau} \psi_2)) \leftrightarrow (\varphi \Rrightarrow_{\leq\tau} (\psi_1 \wedge \psi_2))$

  This is the only property of the operator $\Rrightarrow_{\leq}$ presented in the Tables 6 and 7 we will prove.

  Proof:

  "$\rightarrow$": $\models ((\varphi \Rrightarrow_{\leq\tau} \psi_1) \wedge (\varphi \Rrightarrow_{\leq\tau} \psi_2)) \rightarrow (\varphi \Rrightarrow_{\leq\tau} (\psi_1 \wedge \psi_2))$

  Expansion of $\Rrightarrow_{\leq}$ leads to:

  $\models (\Diamond_{\leq\tau}(\psi_1\mathcal{W}\neg\varphi) \wedge \Diamond_{\leq\tau}(\psi_2\mathcal{W}\neg\psi)) \rightarrow (\Diamond_{\leq\tau}(\psi_1 \wedge \psi_2)\mathcal{W}\neg\varphi)$

  To prove: For every trace ρ, one has to show:

  If $\rho \models \Diamond_{\leq\tau}(\psi_1\mathcal{W}\varphi)$ and $\rho \models \Diamond_{\leq\tau}(\psi_2\mathcal{W}\varphi)$
  then $\rho \models \Diamond_{\leq\tau}(\psi_1 \wedge \psi_2)\mathcal{W}\varphi)$.

  Case 1: $\rho \models \Diamond_{\leq\tau}\neg\varphi$:

  Then: $\rho \models \Diamond_{\leq\tau}((\psi_1 \wedge \psi_2)\mathcal{W}\neg\varphi)$

Case 2: $\rho \models \Box_{\leq\tau}\varphi$:

Due to $\rho \models \Diamond_{\leq\tau}(\psi_1 \mathcal{W} \neg\varphi)$:

$\exists t', 0 \leq t' \leq t + \tau : (\rho, t') \models \psi_1 \mathcal{W} \neg\varphi$

Analogously: $\exists t'', 0 \leq t'' \leq t + \tau : (\rho, t'') \models \psi_2 \mathcal{W} \neg\varphi$

Let $\tilde{t} = \max\{t', t''\}$.

$\rightsquigarrow (\rho, \tilde{t}) \models \psi_1 \mathcal{W} \neg\varphi$ and $(\rho, \tilde{t}) \models \psi_2 \mathcal{W} \neg\varphi$ (Note: $\rho \models \Box_{\leq\tau}\varphi$)

$\rightsquigarrow (\rho, \tilde{t}) \models (\psi_1 \wedge \psi_2) \mathcal{W} \neg\varphi$ (see Table 1)

$\rightsquigarrow \rho \models \Diamond_{\leq\tau}(\psi_1 \wedge \psi_2) \mathcal{W} \neg\varphi$

"$\leftarrow$": $\models (\varphi \mapsto_{\leq\tau} (\psi_1 \wedge \psi_2)) \rightarrow ((\varphi \mapsto_{\leq\tau} \psi_1) \wedge (\varphi \mapsto_{\leq\tau} \psi_2))$

Expansion of $\mapsto_{\leq}$ leads to:

$\models (\Diamond_{\leq\tau}(\psi_1 \wedge \psi_2) \mathcal{W} \neg\varphi) \rightarrow (\Diamond_{\leq\tau}(\psi_1 \mathcal{W} \neg\varphi) \wedge \Diamond_{\leq\tau}(\psi_2 \mathcal{W} \neg\varphi))$

To prove: For every trace ρ, one has to show:

If $\rho \models \Diamond_{\leq\tau}(\psi_1 \wedge \psi_2) \mathcal{W} \varphi)$

then $\rho \models \Diamond_{\leq\tau}(\psi_1 \mathcal{W} \varphi)$ and $\rho \models \Diamond_{\leq\tau}(\psi_2 \mathcal{W} \varphi)$.

Due to $\rho \models \Diamond_{\leq\tau}((\psi_1 \wedge \psi_2) \mathcal{W} \neg\varphi)$:

$\exists t', 0 \leq t' \leq t + \tau : (\rho, t') \models (\psi_1 \wedge \psi_2) \mathcal{W} \neg\varphi$

$\rightsquigarrow (\rho, t') \models \psi_1 \mathcal{W} \neg\varphi$ and $(\rho, t') \models \psi_2 \mathcal{W} \neg\varphi$ (see Table 1)

$\rightsquigarrow \rho \models \Diamond_{\leq\tau}(\psi_1 \mathcal{W} \neg\varphi)$ and $\rho \models \Diamond_{\leq\tau}(\psi_2 \mathcal{W} \neg\varphi)$

q.e.d.

- $\models ((\varphi \rightarrow \Diamond_{\leq\tau}\psi) \wedge \Box(\psi \rightarrow (\psi \mathcal{W} \neg\varphi))) \rightarrow (\varphi \mapsto_{\leq\tau} \psi)$ (DI1)

To prove: For every trace ρ, one has to show:

If $\rho \models \varphi \rightarrow \Diamond_{\leq\tau}\psi$ and $\rho \models \Box(\psi \rightarrow (\psi \mathcal{W} \neg\varphi))$ then $\rho \models \Diamond_{\leq\tau}(\psi \mathcal{W} \varphi)$.

Proof:

Case 1: $(\rho, t) \models \Diamond_{\leq\tau}\neg\varphi$:

$\rightsquigarrow \exists t', t \leq t' \leq t + \tau : (\rho, t') \models \neg\varphi \rightsquigarrow (\rho, t') \models \psi \mathcal{W} \neg\varphi$

$\rightsquigarrow (\rho, t) \models \Diamond_{\leq\tau}(\psi \mathcal{W} \neg\varphi)$

Case 2: $(\rho, t) \models \Box_{\leq\tau}\varphi$:

Due to $(\rho, t) \models (\varphi \rightarrow \Diamond_{\leq\tau}\psi) : \exists t'', t \leq t'' \leq t + \tau : (\rho, t'') \models \psi$.

Because of the second precondition, $\rho \models \Box(\psi \rightarrow (\psi \mathcal{W} \neg\varphi))$:

$(\rho, t'') \models \psi \mathcal{W} \neg\varphi$

$\rightsquigarrow (\rho, t) \models \Diamond_{\leq\tau}(\psi \mathcal{W} \neg\varphi)$

q.e.d.

- $\not\models (\varphi \rightarrow \Diamond_{\leq\tau}\psi) \rightarrow (\varphi \mapsto\!\!\!\Rightarrow_{\leq\tau} \psi)$ (DI2)

  Counter example:

  Let $\tau = 4$ time units. A trace ρ is given by Figure 4.

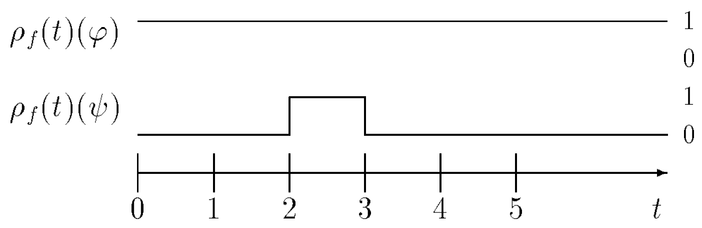

Figure 4: Counter example for (DI2)

- $\not\models \Box(\psi \rightarrow (\psi \mathcal{W} \neg\varphi)) \rightarrow (\varphi \mapsto\!\!\!\Rightarrow_{\leq\tau} \psi)$ (DI3)

  Counter example:

  Let $\varphi \equiv$ `true`, $\psi \equiv$ `false`, and $\tau \in \mathbf{R}^+$ then for every trace ρ:
  $\rho \models \Box(\psi \rightarrow (\psi \mathcal{W} \neg\varphi))$, but $\rho \not\models (\varphi \mapsto\!\!\!\Rightarrow_{\leq\tau} \psi)$.

- $\models (\Box(\varphi_1 \mapsto\!\!\!\Rightarrow_{\leq\tau} \varphi_2) \wedge \Box(\varphi_2 \mapsto\!\!\!\Rightarrow_{\leq\pi} \varphi_3)) \rightarrow \Box(\varphi_1 \mapsto\!\!\!\Rightarrow_{\leq\tau+\pi} \varphi_3)$ (DI4)

  To prove: For every trace ρ, one has to show:
  If $\rho \models \Box\Diamond_{\leq\tau}(\varphi_1 \mathcal{W} \varphi_2)$ and $\rho \models \Box\Diamond_{\leq\pi}(\varphi_2 \mathcal{W} \varphi_3)$ then
  $\rho_i \models \Box\Diamond_{\leq\tau+\pi}(\varphi_3 \mathcal{W} \neg\varphi_1)$.

  Proof:

  Case 1: $(\rho, t) \models \Diamond_{\leq\tau+\pi}\neg\varphi_1$:
  $\rightsquigarrow \exists t', t \leq t' \leq t + \tau + \pi : (\rho, t) \models \neg\varphi_1$
  $\rightsquigarrow (\rho, t) \models \varphi_3 \mathcal{W} \neg\varphi_1$
  $\rightsquigarrow (\rho, t) \models \Diamond_{\leq\tau+\pi}(\varphi_3 \mathcal{W} \neg\varphi_1)$

  Case 2: $\rho_i \models \Box_{\leq\tau+\pi}\varphi_1$:
  Case 2.1: $(\rho, t) \models \Box\varphi_1$:
  Due to $(\rho, t) \models \Diamond_{\leq\tau}(\varphi_2 \mathcal{W} \neg\varphi_1)$:
  $\exists t', t \leq t' \leq t + \tau : (\rho, t) \models \Box\varphi_2$
  Due to $(\rho, t') \models \Diamond_{\leq\pi}(\varphi_3 \mathcal{W} \neg\varphi_2)$:
  $\exists t'', t' \leq t'' \leq t' + \pi : (\rho, t') \models \Box\varphi_3$
  Since $t'' \leq t + \tau + \pi : (\rho, t) \models \Diamond_{\leq\tau+\pi}(\varphi_3 \mathcal{W} \neg\varphi_1)$

Case 2.2: $(\rho, t) \models \Diamond\neg\varphi_1$:

Let $t_{\min} = \min\{\hat{t} | \hat{t} > t + \tau + \pi \text{ and } (\rho, \hat{t}) \models \neg\varphi_1\}$.
Due to $(\rho, t) \models \Diamond_{\leq\tau}(\varphi_2 \mathcal{W} \neg\varphi_1)$:
$\exists t', t \leq t' \leq t + \tau : \forall \tilde{t}, t' \leq \tilde{t} \leq t_{\min} : (\rho, \tilde{t}) \models \Box\varphi_2$
Due to $(\rho, t') \models \Diamond_{\leq\pi}(\varphi_3 \mathcal{W} \neg\varphi_2)$:
$\exists t'', t' \leq t'' \leq t' + \pi : \forall \bar{t}, t'' \leq \bar{t} \leq t_{\min} : (\rho, \bar{t}) \models \Box\varphi_3$
Since $t'' \leq t + \tau + \pi : (\rho, t) \models \Diamond_{\leq\tau+\pi}(\varphi_3 \mathcal{W} \neg\varphi_1)$

q.e.d.

- $\models \Box(\varphi \Rrightarrow_{\leq\tau} \psi) \leftrightarrow \Box(\neg\psi \Rrightarrow_{\leq\tau} \neg\varphi)$ (DI5)

  Counter example:

  Let $\tau = 2$ time units. A trace ρ is given by Figure 5.
  Consider $t = 1$.

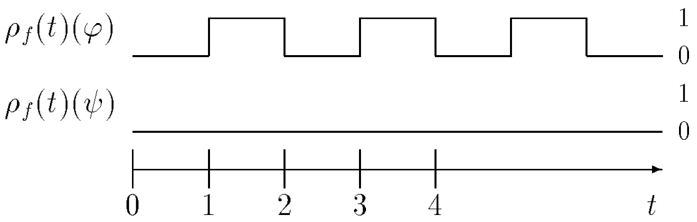

Figure 5: Counter example for (DI5)

B.3 Delayed Equivalence Pattern

- $\not\models \Box(\varphi \Leftrightarrow_{\leq\tau} \psi) \rightarrow \Box(\psi \Leftrightarrow_{\leq\tau} \varphi)$ (DE1)

  Counter example:

  Let $\tau = 1$ time unit. A trace ρ is given by Figure 6.
  Consider t=0.

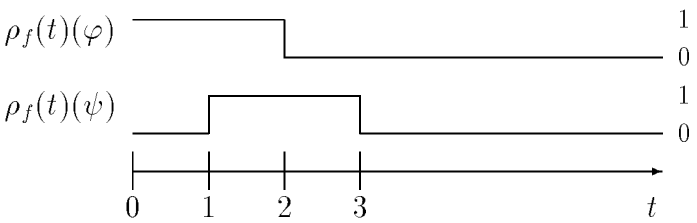

Figure 6: Counter example for (DE1)