

# Distance Functions for Defaults in Reactive Systems

Sofia Guerra

Department of Computer Science  
University College London  
London WC1E 6BT, UK  
S.Guerra@cs.ucl.ac.uk  
fax: +44 (0)20 7387 1397

**Abstract.** Default reasoning has become an important topic in software engineering. In particular, defaults can be used to revise specifications, to enhance reusability of existing systems, and to allow a more economic description of systems. In this paper we develop a framework for default specifications of reactive systems.

We present a formalisation of non-monotonicity in temporal logic based on the notion of default institution. Default institutions were defined as an extension of institutions in order to allow partial reuse of existing modules. The semantics of defaults is given by a (generalised) distance between interpretations. In this way, by defining a pre-order between temporal morphisms and using temporal logic as a specification language, we get a way of handling defaults in specifications of reactive systems. We illustrate the developed formalism with an example in which a specification is reused, but where the new behaviour contradicts the initial specification. In this example, the initial specification is seen as a default to which exceptions are added.

## 1 Introduction

Although default reasoning first appeared as a field within artificial intelligence, it has become an important topic in software engineering. It concerns the reasoning based on assumptions held to be true unless there is specific evidence to the contrary. Several issues show the benefits of defaults in specifications. In particular, defaults can be used to revise specifications, they enhance reusability of existing systems, and they allow a more economic description of systems. Defaults can also be used to handle inconsistencies resulting from the combination of different perspectives and views of people developing a large software system, i.e. in the viewpoints framework [3].

Many important computer programs such as operating systems, network communication protocols, and air traffic control systems exhibit ongoing behaviour which is ideally non-terminating, and thus infinite, reflecting their continuously operating nature. These systems maintain an ongoing interaction with their environment, and intermediate outputs of the program can influence subsequent intermediate inputs of the program. Such systems are called *reactive*

*systems*. Following the seminal work by Pnueli [9], temporal logics have proved to be suitable for modelling the reactive aspects of systems. Therefore, developing a formalism for default reasoning based on temporal logic will allow for the description of reactive software systems where we can directly talk about defaults.

The work of Goguen and Burstall on institutions [4] has shown that many aspects of specifications in the large, namely the ability to put together small specifications to form the specification of a complex system, depend only on some properties of the underlying logic. Institutions formalise the notion of ‘a logical system’. In addition, they provide a way of ‘gluing’ together theories and hence a way of structuring specification. Although institutions provide a way of structuring specifications, the existing specifications can be enriched but not modified. *Default institutions* [10] were proposed as an extension to the notion of institution in order to enable partial re-use of specifications. They are a generic framework for the treatment of exceptions to a norm, based on a logic and a generalised distance between its interpretations. By using this generalisation of distances, they give semantics to the combination of default modules with more specific information that can override the defaults. In this way, in the same way the theory of algebraic specifications is parameterised by institutions, theory of structuring with exceptions will be parameterised by default institutions.

In this paper we develop a formalism for non-monotonicity in temporal logic based on the concept of default institution. In this framework, the semantics of a default with an exception is given by selecting the models of the exception that are as close as possible to the models of the default, according to the given notion of distance function. The temporal instantiation of default institutions allows for the development of structured default specifications of reactive systems.

Although several mechanisms have been proposed for handling defaults in common-sense reasoning, not so many have been proposed for specifications of reactive systems. The motivations and intended models of frameworks for non-monotonic temporal reasoning about software systems or common-sense examples are different. Hence, the systems intended for these different purposes have to be dissimilar. Non-monotonic reasoning is invisible and virtually non-existent in industry. Morgenstern [8] tries to understand why non-monotonic reasoning and industry are so far apart. The reasons given are related to the fact that research has been focussed almost exclusively on problems of common-sense reasoning, while industry is primarily concerned with problems which appear to have very little to do with common-sense reasoning. Although industry offers fertile ground for non-monotonic researchers, it remains uncharted territory for the non-monotonic community [8]. In the strategy for integrating non-monotonic reasoning in industry, researchers should familiarise themselves with problems in industry, select a set to which non-monotonic reasoning appears to be relevant, and focus on those problems in their research. In this paper we focus on problems of non-monotonicity in specifications, and the decisions made are influenced accordingly.

Most of the work done in temporal defaults has been developed with common-sense examples in mind. However, [1, 6] propose an extension to the Object Specification Logic (OSL) [12] with defaults, to arrive at what they call Object Specification Logic with Defaults (OSD). OSD is based on temporal logic and the interpretations are organised in a pre-order, hence constituting a preferential model [7]. OSD is based on the temporal prioritisation originated from Shoham’s chronological ignorance [11], where earlier defaults are implicitly given higher priority. This approach has some problems which arise from chronological minimisation. The authors themselves highlighted these difficulties, and dealt with them by adding specific defaults with different levels of priority. However, when specifying a system, if using these ideas we would have to know exactly which defaults to add, and which levels of priority to assign to those defaults. These obstacles were considered here when developing the distances between temporal interpretations, and they were directly solved through the use of this framework; the specifier does not have to consider these problems when using the framework developed here. Default institutions are more powerful than preferential models [5] and hence, if the right instantiation is chosen, they allow us to deal with these obstacles directly within the framework.

This paper is structured as follows. Section 2 briefly presents default institutions and the idea of handling defaults by distances between interpretations. Section 3 is the main part of the paper. It begins by presenting the temporal logic that will be used. A simple example is then described, which will illustrate the framework being developed. Finally, we present the formalisation of non-monotonicity in temporal logic that is used to handle defaults in the specification of reactive systems. We conclude in section 4 by summarising the main points and sketching further work.

## 2 The Meaning of but

The algebraic specification school proposes a strong construct where an existing specification can be enriched, but not modified. In [10] default institutions are proposed in order to allow partial reuse of existing specification modules. They extend the general notion of a logic, of *institutions* [4], by including a notion of distance between interpretations. The modification of a specification  $D$  with an exception  $E$  is denoted by  $D$  **but**  $E$ , representing that a default  $D$  can be overridden by more specific properties  $E$ . The semantics of  $D$  **but**  $E$  is given by selecting the models of the exception  $E$  that are as close as possible to the models of the default  $D$  according to the given notion of distance between interpretations.

In general, we want to compare interpretations that may be very different in nature. Therefore, we need a way to relate elements of different nature that play a similar role. This is already done for the framework of algebraic specifications through the use of morphisms between interpretations. The notion of distance is then generalised, and we compare pairs of interpretations linked by a morphism; distances are the particular case when there is only one morphism between each

pair of interpretations. Interpretation morphisms (i.e. pairs of interpretations linked by an indication about which elements play similar roles) are compared by a pre-order  $\leq_{\Sigma}$  among morphisms (here, and in the following,  $\Sigma$  is a signature). This pre-order has to verify some constraints in order to capture the motivations for handling defaults. We explain some of the ideas underlying this pre-order and the notion of default institution. The formal definition of default institution and further explanation can be found in [10, 5].

If  $m, n, m'$ , and  $n'$  are interpretations for a given signature  $\Sigma$ , and  $h : m \rightarrow n$  and  $h' : m' \rightarrow n'$  are interpretation morphisms, intuitively  $h : m \rightarrow n \leq_{\Sigma} h' : m' \rightarrow n'$  means that  $m$  is ‘closer’ to  $n$  (according to  $h$ ) than  $m'$  to  $n'$  (according to  $h'$ ). The minimal morphisms for each of these orderings  $\leq_{\Sigma}$  are called *agreements*. Note that identity morphisms should always be agreements, since what can be closer to any interpretation than itself? We want interpretations linked by a minimal morphism (an agreement) to behave similarly, since the fact that a morphism  $h : m \rightarrow n$  is minimal represents that  $m$  is as close to  $n$  as possible. To guarantee that it is the case, we impose that if two interpretations  $m$  and  $n$  are linked by an interpretation morphism  $h : m \rightarrow n$  then the properties of  $m$  are also properties of  $n$ , i.e. if  $\mathcal{L}_{\Sigma}$  is the set of all formulae with signature  $\Sigma$  then  $\{\phi \in \mathcal{L}_{\Sigma} | m \Vdash \phi\} \subseteq \{\phi \in \mathcal{L}_{\Sigma} | n \Vdash \phi\}$ . This condition is called *weak abstractness* [10]. Intuitively, this means that our logic does not allow us to look at more details of the interpretations than the morphisms do. In addition, the usual condition of symmetry on distance functions is weakened towards *0-symmetry* [10]: if there is an agreement from  $m$  to  $n$ , then there is also one from  $n$  to  $m$ . These two conditions of 0-symmetry and weak abstractness together imply that two interpretations linked by an agreement satisfy exactly the same formulae. An extra restriction on the pre-order is the requirement that agreements should be transparent with respect to comparisons. This condition is called 0-equivalence and it means that composing a morphism with an agreement is equivalent (in the pre-order) to the initial morphism itself.

The semantics of the combination of a default  $D$  and an exception  $E$ , noted  $D \mathbf{but} E$ , is defined using the concept of (generalised) distance given by the pre-order on interpretation morphisms. An interpretation  $m$  is a model of  $D \mathbf{but} E$  if  $m$  is the domain of a minimal morphism among the morphisms whose domain satisfies  $E$  and whose codomain satisfies  $D$ . Note that models of  $D \mathbf{but} E$  always satisfy the exception  $E$ . To express this formally, some notation is explained:

- If  $E$  and  $D$  are sets of formulae,  $Mor(E, D)$  is the class of morphisms whose domain satisfy  $E$  and whose codomain satisfy  $D$ .
- $Min(E, D)$  is the class of minimal morphisms of  $Mor(E, D)$ .

Formally, the models of the **but** are the following:

**Definition 1 (Semantics of but).** *Let  $E$  and  $D$  be sets of formulae over a signature  $\Sigma$ , and  $m$  an interpretation. Then  $m$  is a model of  $D \mathbf{but} E$ , written  $m \Vdash D \mathbf{but} E$ , iff there is a morphism  $h \in Min(E, D)$  such that  $m = dom(h)$ .*

When using this machinery within a particular logic, the main choices we have to make are on the morphisms between interpretations and the pre-order

between these morphisms, i.e. we have to decide what is the pre-order  $\leq_{\mathcal{Y}}$ . This is exactly what we do in the next section for the temporal case.

### 3 The Temporal Setting

The structure of temporal argument and temporal discourse has had an interest in different fields over the years, probably as a result of the temporal facet of human reasoning. In Computer Science, the use of temporal logic as a formalism for specifying and verifying correctness of computer programs was stated by Pnueli in [9], and it has been widely explored since this landmark paper.

In the remainder of this section we briefly describe linear propositional temporal logic, and we define morphisms between temporal interpretations and a pre-order on them. The pre-order presented here is tailored by the applications at hand, namely specifications with defaults.

#### 3.1 Propositional Temporal Logic

The language of propositional temporal logic is based on the language of propositional logic, but various operators or ‘modalities’ are provided to reason about the change of truth values of assertions over time. Examples of common temporal operators include **G** (always in the future), **F** (eventually), **W** (unless), **X** (next) and **U** (until). The propositional temporal language is defined starting from a set of proposition letters  $\Sigma$  (a signature). The set  $\mathcal{L}_{\Sigma}$  of propositional temporal formulae is the least set such that:

- each  $p \in \Sigma$  is a formula ( $\Sigma \subset \mathcal{L}_{\Sigma}$ );
- if  $A$  and  $B$  are in  $\mathcal{L}_{\Sigma}$ , then  $\neg A$  and  $(A \wedge B)$  are in  $\mathcal{L}_{\Sigma}$ ;
- if  $A$  and  $B$  are in  $\mathcal{L}_{\Sigma}$ , then **X** $A$  and **A** $UB$  are in  $\mathcal{L}_{\Sigma}$ .

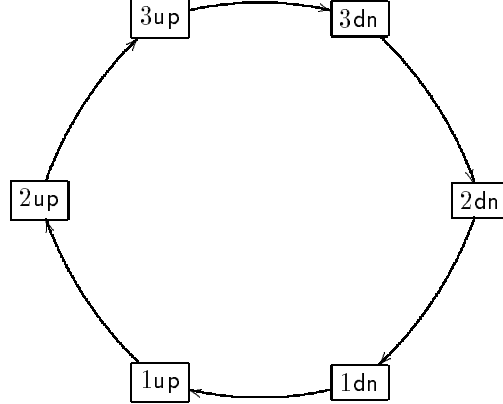
The semantics of the temporal language is given in terms of *frames*; a frame is an ordered pair  $\mathcal{F} = (W, R)$ , where  $W$  is the set of points in time ordered by a binary relation  $R$  of precedence between them:  $sRt$  is read ‘ $t$  is after  $s$ ’. In the sequel we will use the frame  $(\mathbb{N}, <)$ . An *interpretation* is a triple  $\mathcal{M} = (\mathbb{N}, <, V)$ , where  $V$  is a function assigning to every timepoint  $t$  (every natural number) a set of proposition letters  $V(t) \subseteq \Sigma$ , namely the set of proposition letters that are true at the timepoint  $t$ . The semantics of a formula  $A$  over an interpretation  $m$  at timepoint  $t$ , written  $m \Vdash_t A$ , is recursively defined as follows:

$m \Vdash_t p, p \in \Sigma$	iff	$p \in V(t)$ ;
$m \Vdash_t \neg A$	iff	it is not the case that $m \Vdash_t A$ ;
$m \Vdash_t A \wedge B$	iff	$m \Vdash_t A$ and $m \Vdash_t B$ ;
$m \Vdash_t \mathbf{X}A$	iff	$m \Vdash_{t+1} A$ ;
$m \Vdash_t \mathbf{A}UB$	iff	there exists $t' \in \mathbb{N}$ with $t \leq t'$ and $m \Vdash_{t'} B$ and for every $t'' \in \mathbb{N}$ , if $t \leq t'' < t'$ then $m \Vdash_{t''} A$ .

We say that a formula  $A$  is *valid on the interpretation*  $m$  iff  $m \models_0 A$ . The semantics of the other temporal operators can be derived from these.

Now that we have defined the propositional temporal logic, the problem is how to define morphisms between interpretations and a pre-order on the morphisms in such a way that the defaults in specifications of reactive systems behave in the expected way. To motivate these definitions let us look at an example.

### 3.2 An Example: a Ferris Wheel



Consider a Ferris Wheel with six chairs, where each chair can be in one of six positions. As the relative position of the chairs does not change, from the position of one of the chairs we know the position of all the other five. Hence, we choose one of the chairs and from now on we always talk about the same chair. The chair can be in one of three levels 1, 2 or 3, and it moves always in the same direction (clockwise). The name of each position indicates the level of the chair and whether it is going up or down: e.g. 3up indicates that it is at level 3 and it is going up. Following this convention, the names of the positions of the chair are: 1up, 2up, 3up, 3down, 2down and 1down. The chair goes from 1up to 2up, and from this to 3up, then it goes to 3down and starts going down to 2down and 1down and then to 1up again, and it never stops. A specification of this Ferris Wheel can be seen below. The first formula states the fact that the chair is always in one of the six positions and it is not in more than one at each moment. The other formulae describe the movement of the chair.

$$\begin{aligned}
 & \mathbf{G}((1\text{up} \vee 2\text{up} \vee 3\text{up} \vee 3\text{down} \vee 2\text{down} \vee 1\text{down}) \wedge \\
 & \quad \neg(1\text{up} \wedge 2\text{up}) \wedge \neg(1\text{up} \wedge 3\text{up}) \wedge \neg(1\text{up} \wedge 3\text{down}) \wedge \dots) \\
 & \mathbf{G}(1\text{up} \rightarrow \mathbf{X}2\text{up}) \qquad \qquad \qquad \mathbf{G}(2\text{up} \rightarrow \mathbf{X}3\text{up}) \\
 & \mathbf{G}(3\text{up} \rightarrow \mathbf{X}3\text{down}) \qquad \qquad \mathbf{G}(3\text{down} \rightarrow \mathbf{X}2\text{down}) \\
 & \mathbf{G}(2\text{down} \rightarrow \mathbf{X}1\text{down}) \qquad \qquad \mathbf{G}(1\text{down} \rightarrow \mathbf{X}1\text{up})
 \end{aligned}$$

Suppose now that we want to add an emergency lever that makes the chair reach level 1 as quickly as possible. If the chair is in one of the down positions,

then the fastest way of getting to level 1 is to keep going clockwise. However, if the chair is in one of the **up** positions then the best thing is to reverse the direction, and move counter-clockwise. We suppose that the **lever** stays pushed until the chair gets to level 1 when it stops, and then the **lever** is released. The action of pushing the lever is called **push**, and **lever** represents the fact that the lever is pushed. The formulae below represent what we want to add to the specification.

$$\begin{aligned} & \mathbf{G}(\text{push} \rightarrow \text{lever}) & \mathbf{G}(((3\text{down} \vee 2\text{down}) \wedge \text{lever}) \rightarrow \mathbf{X}\text{lever}) \\ & \mathbf{G}((\text{lever} \wedge 3\text{up}) \rightarrow \mathbf{X}(2\text{up} \wedge \text{lever})) & \mathbf{G}((\text{lever} \wedge 1\text{down}) \rightarrow \mathbf{X}(1\text{down} \wedge \neg\text{lever})) \\ & \mathbf{G}((\text{lever} \wedge 2\text{up}) \rightarrow \mathbf{X}(1\text{up} \wedge \text{lever})) & \mathbf{G}((\text{lever} \wedge 1\text{up}) \rightarrow \mathbf{X}(1\text{up} \wedge \neg\text{lever})) \end{aligned}$$

Instead of rewriting the specification from the beginning, and considering explicitly whether the **lever** has been pushed or not, we want to add these formulae as exceptions to the specification. If  $D$  is the conjunction of the formulae in the specification of the Ferris Wheel, and  $E$  the conjunction of the formulae that describe the **lever**, then the models of the Ferris Wheel with the emergency lever would be the models of the state constraint that satisfy  $D$  **but**  $E$ . However, in order to give semantics to **but** we have to know what are the morphisms between interpretations and how they are ordered, i.e. what is the pre-order  $\leq_{\Sigma}$ .

### 3.3 The Pre-order

When defining the pre-order, the first thing to note is that not every formula should be overridable. In particular, state constraints have to be rigid, otherwise we could satisfy the exceptions by allowing the chair to be in more than one position at the same time. These formulae should not be invalidated and we see them as axioms that cannot be overridden. The details of how to do that are omitted, but the basic idea is to consider hierarchic specifications. A hierarchic specification consists of two parts: a set of axioms that correspond to the facts that must hold, plus a part of defaults. The defaults are organised by priority levels, and they express properties that are likely to be true but can be overridden by other information, either by the axioms or by defaults with a higher priority. The models of a hierarchic specification are the models of the axioms that satisfy as much of the defaults as possible, taking into consideration the priority among them. The pre-order of morphisms is generalised towards a pre-order of families of morphisms. More details can be found in [5], but here it would correspond to considering for minimisation only the models of these axioms. In our example, the state constraint is expressed by the first formula of the specification of the Ferris Wheel.

If we think about how to order the morphisms, a first possibility which follows from the ideas in [10] would be the following: if  $h : m \rightarrow n$  and  $h' : m' \rightarrow n'$  are morphisms between temporal interpretations for a signature  $\Sigma$ ,  $h : m \rightarrow n \leq_{\Sigma} h' : m' \rightarrow n'$  iff for all  $t \in \mathbb{N}$ ,

$$\text{if } m \text{ disagrees with } n \text{ at } t \quad \text{then} \quad m' \text{ disagrees with } n' \text{ at } t,$$

where by *disagree* we mean that they do not satisfy the same propositional symbols at that instant. This definition has two immediate problems. Firstly, if  $m$  disagrees with  $n$  at some point  $t$  then what happens afterwards might be meaningless. As an example consider the Ferris Wheel; let  $m$  be a model of the exceptions (the formulae that describe the lever) and  $n$  a model of the initial specification. Suppose that at some timepoint  $t$ ,  $m$  was at **2up**, the lever was not pushed and, as if by magic, in  $t + 1$  it was at **2down**. If  $n$  agreed with  $m$  at  $t$  (i.e.  $m$  was at **2up** as well) then obviously they disagree at  $t + 1$ . Moreover, even if  $m$  behaves as expected after  $t$  it will disagree with  $n$ , and these disagreements are meaningless because they are the result of what happened at  $t$ . This suggests that we want to see, at some point  $t$ , if  $m$  were the interpretation  $n$ , whether it would behave in the same way. Therefore, morphisms between interpretations are functions that choose, for each timepoint of the domain interpretation, a timepoint in the codomain that satisfies exactly the same symbols as the domain at that point. This represents the idea of supposing that  $m$  was  $n$ ; and then we compare the satisfiability of the propositional symbols at the next points. Following this idea, morphisms between interpretations are monotonic functions  $h : \mathbb{N} \rightarrow \mathbb{N}$  such that for every timepoint  $t$  and every symbol  $p$ ,  $m \Vdash_t p$  iff  $n \Vdash_{h(t)} p$ , and we check whether  $m \Vdash_{t+1} p$  iff  $n \Vdash_{h(t)+1} p$  in order to see how much  $m$  and  $n$  differ.

Another problem with that definition of the pre-order is that it blocks the occurrence of the actions that change the behaviour of the initial specification, like the action of pushing the lever. With that definition, the minimal models of the Ferris Wheel with the lever would be the ones where the lever is never pushed. This is highly undesirable, since we want to be able to push the lever whenever we feel like (whenever there is an emergency). This suggests that we only want to compare interpretations that have the same occurrences of actions at the same timepoints. However, this causes some problems when comparing interpretations with actions that have enabling conditions, as explained below. Nevertheless, we have to treat actions differently from the attributes in the minimisation process. Hence, we split each signature  $\Sigma$  in two disjoint sets: one for the actions  $\Sigma_{Act}$ , that cannot be minimised, and another set  $\Sigma_{Att}$  for the attributes. The language considered is built from the signature as explained before.

**Definition 2.** A signature is a pair of disjoint sets  $\Sigma = (\Sigma_{Act}, \Sigma_{Att})$ , where  $\Sigma_{Act}$  is the set of actions and  $\Sigma_{Att}$  is the set of attributes (or observations).

We impose that comparable interpretations agree at the first instant, i.e. we suppose that they satisfy the same symbols at the instant 0. The reason for this restriction is that when we have a system we know the initial state, and we are interested in the way it evolves; systems with different initial states will obviously be different and the meaning of the comparison is unclear.

We can now say what are the morphisms between interpretations. Firstly we introduce some notation. Let  $h : \mathbb{N} \dashrightarrow \mathbb{N}$  be a partial function and  $t \in \mathbb{N}$  be a natural number. Then

- $h(t) \uparrow$  means that  $h$  is undefined at  $t$ ;

- $h(t) \downarrow$  means that  $h$  is defined at  $t$ ;
- $\bar{h}(t) = \max\{h(t') \in \mathbb{N} : t' \leq t \text{ and } h(t') \downarrow\}$  is the image of the greatest timepoint less or equal than  $t$  at which the function is defined.

The idea of the morphisms is to relate two temporal interpretations  $m$  and  $n$  that have the same initial state (they satisfy exactly the same symbols at 0), in a way that, for each instant  $t_1$ , a timepoint  $t_2$  is chosen in a monotonic way, such that the interpretation domain  $m$  at  $t_1$  satisfies exactly the same symbols as the interpretation codomain  $n$  at  $t_2$ . This choice is partial to allow cases where such a  $t_2$  does not exist. These conditions are formally expressed by the following definition:

**Definition 3.** Let  $\Sigma$  be a signature and  $m$  and  $n$  be two temporal interpretations over  $\Sigma$ . A morphism  $h : m \rightarrow n$  is a partial function  $h : \mathbb{N} \dashrightarrow \mathbb{N}$  such that:

- $h(0) = 0$ ;
- for all  $t_1, t_2 \in \mathbb{N}$ , if  $t_1 < t_2$ ,  $h(t_1) \downarrow$  and  $h(t_2) \downarrow$  then  $h(t_1) < h(t_2)$ ;
- for all  $t \in \mathbb{N}$ , if  $h(t) \downarrow$  then for all  $p \in \Sigma$ ,  $m \Vdash_t p$  iff  $n \Vdash_{h(t)} p$ .

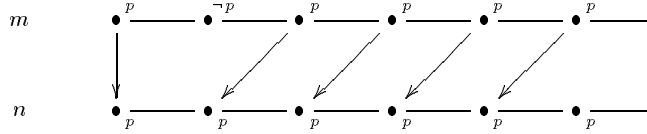
Let us look at an example of a morphism between two temporal interpretations.

*Example 1.* Let  $\Sigma = (\Sigma_{Act}, \Sigma_{Att})$  be a signature where  $\Sigma_{Act} = \emptyset$  and  $\Sigma_{Att} = \{p\}$ . Consider the expression **Gp but X¬p**.

Let  $m$  be a  $\Sigma$ -interpretation such that  $m \Vdash_t p$ , for all  $t \neq 1$ , and  $n$  the  $\Sigma$ -interpretation such that  $n \Vdash_t p$ , for all  $t \in \mathbb{N}$ . Then  $h : m \rightarrow n$  defined as follows is a temporal morphism:

- $h(0) = 0$ ;
- $h(1) \uparrow$ ; and
- $h(t) = t - 1$ , for every  $t > 1$ .

This morphism can be represented by the diagram below.



This morphism belongs to the class of morphisms whose domain satisfies **X¬p** and whose codomain satisfies **Gp**, i.e.  $h \in Mor(\mathbf{X}\neg p, \mathbf{G}p)$ . Note that at timepoint 1 the function  $h$  has to be undefined, as there is no point in  $n$  that satisfies  $\neg p$ . However, the choices for the other timepoints greater than 1 are arbitrary, as far as they are monotonic. We chose this morphism among several other possibilities as this will be minimal in  $Mor(\mathbf{X}\neg p, \mathbf{G}p)$  when we define the pre-order on interpretation morphisms. In fact, this morphism will also belong to the class  $Min(\mathbf{X}\neg p, \mathbf{G}p)$ , and hence  $m \Vdash \mathbf{G}p \text{ but } \mathbf{X}\neg p$ . We present the proof that this is a minimal morphism in the appendix.

Now that we have defined morphisms between temporal interpretations, we can return to considering the notion of ‘closeness’ in order to get the pre-order  $\leq_{\Sigma}$ .

**Concordance on Initial States** Although we cannot define the pre-order yet, taking into account the previous considerations, we know that we only want to compare interpretations with the same initial state. Hence,  $h : m \rightarrow n \leq_{\Sigma} h' : m' \rightarrow n'$  implies:

1.  $\forall p \in \Sigma. m \Vdash_0 p$  iff  $m' \Vdash_0 p$ .

**Minimisation of Discrepancies** Suppose  $h : m \rightarrow n$  is a morphism between two temporal interpretations. If  $m$  was ‘near’  $n$ , it would behave, at every timepoint, in the same way as  $n$ , if  $n$  was at a similar state. Discrepancies are the cases when this does not happen.

**Definition 4.** Let  $\Sigma = (\Sigma_{Act}, \Sigma_{Att})$  be a signature,  $h : m \rightarrow n$  be an interpretation morphism and  $t \in \mathbb{N}$  a timepoint. We say that there is a discrepancy at time  $t$  between  $m$  and  $n$  according to  $h$ , written  $m \overset{h}{\not\leftrightarrow}_t n$ , if  $h(t)$  is defined and there is an attribute  $r \in \Sigma_{Att}$  such that

$$(m \Vdash_{t+1} r \text{ and } n \not\Vdash_{h(t)+1} r) \quad \text{or} \quad (m \not\Vdash_{t+1} r \text{ and } n \Vdash_{h(t)+1} r).$$

This notion of discrepancy expresses the fact that, if the interpretation  $n$  was at the state that  $m$  is at some instant  $t$ , it would behave in a different way. This depends on the morphism  $h$ , that chooses from  $n$  an instant to compare with  $m$  at the instant  $t$ : by definition of morphism, if  $h : m \rightarrow n$  is a morphism, then  $m$  satisfies at  $t$  exactly the same symbols as  $n$  at  $h(t)$ . There is a discrepancy at  $t$  if they progress in different ways: at the following instants,  $t + 1$  for  $m$  and  $h(t) + 1$  for  $n$ , they satisfy different attributes. This notion of discrepancy is going to be used to define the pre-order: two interpretations are ‘nearer’ if they have less discrepancies. It is the justification for unexpected differences between morphisms.

If  $h : m \rightarrow n$  and  $h' : m' \rightarrow n'$  are morphisms, we want to know if  $m$  is closer to  $n$  (according to  $h$ ) than  $m'$  to  $n'$  (according to  $h'$ ). The main idea is to minimise the discrepancies. We could say that if  $h : m \rightarrow n \leq_{\Sigma} h' : m' \rightarrow n'$  and there is a discrepancy at  $t$  between  $m$  and  $n$  then there is also one between  $m'$  and  $n'$  at the same instant:  $\forall t \in \mathbb{N}$ , if  $m \overset{h}{\not\leftrightarrow}_t n$  then  $m' \overset{h'}{\not\leftrightarrow}_t n'$ .

The problem with this definition is that if  $m'$  did not behave as expected before an instant  $t$  that allows it to agree with  $n'$  at  $t$ , but  $m$  did behave as expected until  $t$ , which makes it disagree with  $n$  at  $t$ , then we could not conclude the relation we wanted. Thinking again about the Ferris Wheel example, suppose  $h : m \rightarrow n$  and  $h' : m' \rightarrow n'$  are morphisms,  $m$  and  $m'$  satisfy **lever** at the same instants and they agree at 0. Suppose  $t$  is a timepoint where  $m$  is at 2up and the **lever** is pushed at  $t$ , and this is the first time the lever is pushed. We can also suppose that before  $t$ ,  $m$  satisfies all the formulae of the initial specification (without the **lever**). In these conditions there is a discrepancy between  $m$  and  $n$  at  $t$ . However, if  $m'$  is at a **down** point at  $t$  (because it did not follow the default formulae at some previous instant) then there is no discrepancy at  $t$  between  $m'$  and  $n'$ , and we would not prefer the morphism  $h$  to the morphism  $h'$ . This

is obviously undesirable, as the interpretation  $m$  had always behaved as desired but  $m'$  had not. To solve this we impose that if  $h : m \rightarrow n \leq_{\Sigma} h' : m' \rightarrow n'$  and if there is a discrepancy between  $m$  and  $n$  at  $t$  that does not exist between  $m'$  and  $n'$  then there must be a reason for this: either because  $h'$  is undefined at  $t$  or because at a previous instant there was a discrepancy in  $h'$  that did not exist in  $h$ . This gives rise to the following condition that we add to the previous we already had: if  $h : m \rightarrow n \leq_{\Sigma} h' : m' \rightarrow n'$  then

$$2. \forall t_1 \in \mathbb{N}. ((m \xrightarrow{h}_{t_1} n \wedge m' \not\xrightarrow{h'}_{t_1} n' \wedge h'(t_1) \downarrow) \Rightarrow (\exists t_2 < t_1. (m \not\xrightarrow{h}_{t_2} n \wedge h(t_2) \downarrow \wedge (m' \xrightarrow{h'}_{t_2} n' \vee h'(t_2) \uparrow))))).$$

**Synchronisation of Actions** We said that the occurrence of actions should not be blocked in the minimisation process. One way of doing it would be to restrict the comparisons between morphisms whose domain satisfies the same actions at the same instants. This does not cause any problem when the actions do not have enabling conditions, like in the Ferris Wheel example. However, this is not the case when we consider examples where there are enabling conditions on the defaults that are not overridden by exceptions. Since we would not block the occurrence of actions in the domains of morphisms, they would be able to occur even if they did not satisfy the enabling conditions of the defaults. The following example illustrates this problem.

*Example 2.* Consider a library with books. Each book of the library can be available to be taken by a user. The attribute `available` means that a book is available to be taken by a user. A book can be `taken` when it is `available`, which makes it not `available`. Similarly, a book can be `returned` when not `available`, which causes it to be `available` again.

Suppose now that we want to consider reserved books, having the added property that they may not be taken out of the library. In this case, when the book is `suspended` it stops being `available` until it is `resumed` again.

The signature for this specification is  $\Sigma = (\Sigma_{Act}, \Sigma_{Att})$ , where  $\Sigma_{Act} = \{\text{taken}, \text{suspended}, \text{returned}, \text{resumed}\}$  and  $\Sigma_{Att} = \{\text{available}\}$ . Hence, the set of defaults  $D$  of this specification will include:

$$\begin{aligned} & \mathbf{G}(\text{taken} \rightarrow \text{available}) \\ & \mathbf{G}(\text{taken} \rightarrow \mathbf{X}\neg\text{available}) \\ & \mathbf{G}(\text{available} \wedge \neg\text{taken} \rightarrow \mathbf{X}\text{available}) \end{aligned}$$

and the following will be included in the exceptions that describe the reserved book:

$$\begin{aligned} & \mathbf{G}(\text{suspended} \rightarrow \text{available}) \\ & \mathbf{G}(\text{suspended} \rightarrow \mathbf{X}\neg\text{available}). \end{aligned}$$

Hence, the occurrence of the action **suspended** should not be restricted. However, **taken** should only occur when **available** is true; **available** should be seen as an enabling condition for **taken**. If, as suggested earlier, two morphisms are only compared when their interpretation domain have the same occurrences of actions, we do not have a way of preferring interpretations in which **taken** occurs only when **available** is true.

As we showed in the previous example, actions that are always enabled, i.e. actions that do not have enabling conditions, should not be blocked, even if they are or cause inconsistencies with the defaults. However, if the defaults restrict their occurrence, these conditions should be taken into consideration. In order to do that, we impose the following: in the cases where there are no enabling conditions for the occurrence of actions, we will only compare morphisms if their domains have the same occurrences at the same timepoints. If the domain of two morphisms  $h$  and  $h'$  do not satisfy the same occurrences of actions at the same timepoints, the morphisms are comparable only if there was a reason for the differences between these occurrences, either by a discrepancy in a previous timepoint, or by the map being undefined at the point in which the action occurs. Hence, we can add to the previous conditions the following two:

3.  $\forall a \in \Sigma_{Act}. \forall t_1 \in \mathbb{N}. (m \Vdash_{t_1} a \wedge m' \not\Vdash_{t_1} a) \Rightarrow$   
 $((\exists t_2 < t_1. (m' \not\overset{h'}{\mapsto}_{t_2} n' \wedge h'(t_2) \downarrow \wedge (m \overset{h}{\mapsto}_{t_2} n \vee h(t_2) \uparrow))) \vee (h(t_1) \uparrow \wedge h'(t_1) \downarrow))$
4.  $\forall a \in \Sigma_{Act}. \forall t_1 \in \mathbb{N}. (m' \Vdash_{t_1} a \wedge m \not\Vdash_{t_1} a) \Rightarrow$   
 $((\exists t_2 < t_1. (m \overset{h}{\mapsto}_{t_2} n \wedge h(t_2) \downarrow \wedge (m' \overset{h'}{\mapsto}_{t_2} n' \vee h'(t_2) \uparrow))) \vee (h(t_1) \downarrow \wedge h'(t_1) \uparrow))$

**Surjectivity and Weak Abstractness** To define the pre-order there are two further conditions that we need to impose. We want minimal morphisms to be as defined as possible: a completely undefined function does not have any discrepancy. Hence, in the same way as we did with the occurrence of actions in conditions 3 and 4, if  $h \leq_{\Sigma} h'$  and there is a timepoint  $t_1$  such that  $h(t_1) \uparrow$  and  $h'(t_1) \downarrow$ , then there was a reason for it, namely there was a timepoint  $t_2$  before  $t_1$  at which there was a discrepancy that did not occur in  $h$ , or at that point  $h(t_2)$  was defined and  $h'(t_2)$  was undefined. This corresponds to condition 5 of definition 5 below.

The last condition of the pre-order ensures that the morphisms are as surjective as possible; this has as a result that identities are minimal, and that any minimal morphism is surjective. In this way the condition of weak abstractness is verified. If we allow non-surjective morphisms to be minimal, then we could have two interpretations linked by an agreement that would not satisfy the same formulae. In the points of the codomain interpretation that were not mapped by any point of the domain, the interpretation could satisfy different propositional symbols that would result in the two interpretations not satisfying the same formulae. The formal definition of the pre-order is the following:

**Definition 5.** Let  $\Sigma = (\Sigma_{Act}, \Sigma_{Att})$  be a signature, and  $h : m \rightarrow n$  and  $h' : m' \rightarrow n'$  be interpretation morphisms. We say that  $h : m \rightarrow n \leq_{\Sigma} h' : m' \rightarrow n'$  iff:

1.  $\forall p \in \Sigma. m \Vdash_0 p$  iff  $m' \Vdash_0 p$ ;
2.  $\forall t_1 \in \mathbb{N}. ((m \xrightarrow{h}_{t_1} n \wedge m' \xrightarrow{h'}_{t_1} n' \wedge h'(t_1) \downarrow) \Rightarrow (\exists t_2 < t_1. (m \xrightarrow{h}_{t_2} n \wedge h(t_2) \downarrow \wedge (m' \xrightarrow{h'}_{t_2} n' \vee h'(t_2) \uparrow))))$
3.  $\forall a \in \Sigma_{Act}. \forall t_1 \in \mathbb{N}. (m \Vdash_{t_1} a \wedge m' \not\Vdash_{t_1} a) \Rightarrow ((\exists t_2 < t_1. (m' \xrightarrow{h'}_{t_2} n' \wedge h'(t_2) \downarrow \wedge (m \xrightarrow{h}_{t_2} n \vee h(t_2) \uparrow))) \vee (h(t_1) \uparrow \wedge h'(t_1) \downarrow))$
4.  $\forall a \in \Sigma_{Act}. \forall t_1 \in \mathbb{N}. (m' \Vdash_{t_1} a \wedge m \not\Vdash_{t_1} a) \Rightarrow ((\exists t_2 < t_1. (m \xrightarrow{h}_{t_2} n \wedge h(t_2) \downarrow \wedge (m' \xrightarrow{h'}_{t_2} n' \vee h'(t_2) \uparrow))) \vee (h(t_1) \downarrow \wedge h'(t_1) \uparrow))$
5.  $\forall t_1 \in \mathbb{N}. ((h(t_1) \uparrow \wedge h'(t_1) \downarrow) \Rightarrow (\exists t_2 < t_1. (m \xrightarrow{h}_{t_2} n \wedge h(t_2) \downarrow \wedge (m' \xrightarrow{h'}_{t_2} n' \vee h'(t_2) \uparrow))))$
6.  $(\forall t \in \mathbb{N}. (m \xrightarrow{h}_t n$  iff  $m' \xrightarrow{h'}_t n') \wedge (h(t) \downarrow$  iff  $h'(t) \downarrow)) \Rightarrow (\forall t \in \mathbb{N}. ((\exists t_1 \in \mathbb{N}. \overline{h}(t-1) < t_1 < h(t) \wedge \forall p \in \Sigma. (m \Vdash_{t_1} p$  iff  $n \Vdash_{t_1} p)) \Rightarrow (\exists t_2 \in \mathbb{N}. \overline{h'}(t-1) < t_2 < h'(t) \wedge \forall p \in \Sigma. (m' \Vdash_{t_2} p$  iff  $n' \Vdash_{t_2} p))))$ .

To sum up, we only compare morphisms if the domains have the same initial state (condition 1). The minimisation of the discrepancies is expressed by condition 2. This resembles chronological minimisation [11], where defaults in earlier instants have higher priority than the ones occurring later. However, some of the problems with chronological minimisation do not occur here, as for example it would postpone the occurrence of the actions for ever. Conditions 3 and 4 make possible comparisons of interpretations only if the same actions occur at the same timepoints (unless there was a reason). The symmetry of conditions 3 and 4 is a way of avoiding blockage of the occurrence of actions. Moreover, we prefer morphisms ‘more defined’ and ‘more surjective’ (conditions 5 and 6 respectively). This relation is in fact a pre-order and it verifies the conditions presented in the previous section. In particular, the conditions 0-symmetry and 0-equivalence result from the fact that a morphism  $h$  is minimal in  $\leq_{\Sigma}$  iff  $h$  is an identity [5].

Going back to our example, let  $\Sigma = \{\Sigma_{Act}, \Sigma_{Att}\}$  be the signature with  $\Sigma_{Act} = \{\text{push}\}$  and  $\Sigma_{Att} = \{3\text{down}, 2\text{down}, 1\text{down}, 1\text{up}, 2\text{up}, 3\text{up}, \text{lever}\}$ . We have that, if  $m$  is a model of the state constraint, and  $m \Vdash D$  but  $E$ , then  $m$  is one of the desired models: if the chair is at 3down, 2down, or at 3up or 2up but the emergency lever has not been pushed, then it behaves as before adding the lever; if the chair is at 1down or 1up and the lever has been pushed it stops; if it is at 2up or 3up and the lever has been pushed, then it changes direction, as specified in the formulae we added. This framework can be extended to the general case with several defaults with an arbitrary precedence between them [10, 5].

## 4 Concluding Remarks

In this paper we present a framework for specifications with exceptions for software systems that evolve over time. This framework is based on the notion of default institution [10], which allows partial re-use of specification modules. The semantics is parameterised by a notion of distance between interpretations.

We propose a temporal instantiation of this framework, which consists of the usual propositional linear temporal logic, but where the notion of morphism between interpretations is changed, and it is extended with a pre-order between these morphisms. This notion of ordering between morphisms of interpretations is used in order to deal with non-monotonicity in specifications. Hence, by describing specifications of reactive systems through the use of temporal logic, descriptions can include defaults and the temporal framework provides semantics for those specifications. We illustrate the developed formalism with an example where a specification is reused but where the new behaviour contradicts the initial specification. In this example the initial specification is seen as a default to which exceptions are added, avoiding the necessity of writing the whole specification from the beginning. This framework handles temporal default specifications in a way that it solves some of the problems of previous work in the area [1, 6].

Further work is needed in order to study the applicability of this framework. Firstly, this could be generalised towards a multi-sorted first-order temporal logic or to other more powerful languages. Moreover, it is necessary to study the scalability of this formalism, and the development of algorithms to compute the models of but in a way similar to the work done for first-order logic [10].

## Acknowledgements

Thanks to Pierre-Yves Schobbens and Mark Ryan for useful discussions, and to Anthony Finkelstein and Anthony Hunter for reading previous drafts of this paper. The author acknowledges financial support from the European Union (RENOIR), and from PRAXIS XXI and Fundação Calouste Gulbenkian in Portugal.

## References

1. Stefan Brass, Udo Lipeck, and Pedro Resende. Specification of object behaviour with defaults. In Udo Lipeck and Gerhard Koschorreck, editors, *Proceedings of the International Workshop on Information Systems: Correctness and Reusability, ISCORE-93*, pages 155–177, 1993.
2. Dov Gabbay, C. J. Hogger, and J. A. Robinson, editors. *Nonmonotonic Reasoning and Uncertain Reasoning*, volume 3 of *Handbook of Logic in Artificial Intelligence and Logic Programming*. Oxford: Clarendon Press, 1994.
3. Anthony Finkelstein and Ian Sommerville. The viewpoints FAQ. *Software Engineering Journal*, 11(1):2–4, 1996.
4. Joseph A. Goguen and Rod M. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the ACM*, 39(1):95–146, January 1992.

5. Sofia Guerra. *Defaults in the Specification of Reactive Systems*. PhD thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, 1999.
6. Udo W. Lipeck and Stefan Brass. Object-oriented system specification using defaults. In K. V. Luck and H. Marburger, editors, *Management and Processing Complex Data Structures, Proceedings 3rd Workshop on Information Systems and Artificial Intelligence*, volume LNCS 777, pages 22–43, Berlin, 1994. Springer Verlag.
7. David Makinson. General patterns in non-monotonic reasoning. In Gabbay et al. [2], chapter 2, pages 35–110.
8. Leora Morgenstern. Inheritance comes of age: Applying nonmonotonic techniques to problems in industry. *Artificial Intelligence*, 103:237–271, 1998.
9. Amir Pnueli. The temporal logic of programs. In *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science*, pages 46–57, 1977.
10. Pierre-Yves Schobbens. *Exceptions in Algebraic Specifications*. PhD thesis, Université Catholique de Louvain, Faculté des Sciences Appliquées, 1992.
11. Yoav Shoham. Chronological ignorance: Experiments in nonmonotonic temporal reasoning. *Artificial Intelligence*, 36:279–331, 1988.
12. A. Sernadas, Cristina Sernadas, and J. F. Costa. Object specification logic. *Journal of Logic and Computation*, 5(5):603–630, 1995.

## A Example 1 (proof)

Firstly suppose that  $h' : m' \rightarrow n'$  is a morphism in  $Mor(\mathbf{X}\neg p, \mathbf{G}p)$  and that  $h' \leq_{\Sigma} h$ . We show that  $h \leq_{\Sigma} h'$  by checking the conditions of the pre-order:

1. Condition 1, 2 and 3 are true since by hypothesis  $h' \leq_{\Sigma} h$ .
2. Condition 4 is true because there is no timepoint  $t$  such that  $m \xleftrightarrow{h}_t n$  and  $m' \not\xleftrightarrow{h'}_t n'$ :  $m \xleftrightarrow{h}_t n$  iff  $t = 0$ , which implies  $m' \xleftrightarrow{h'}_t n'$ .
3.  $\{t : h(t) \uparrow\} = \{1\} \subseteq \{t : h'(t) \uparrow\}$ , and then condition 5 is also verified.
4. Condition 6 is true because for all  $t \in \mathbb{N}$ , there is no  $t^*$  such that  $\bar{h}(t-1) < t^* < h(t)$  and  $\forall p \in \Sigma. m \Vdash_t p$  iff  $n \Vdash_{t^*} p$ .

In the other direction, suppose that  $m$  is not in the conditions described and let  $h : m \rightarrow n$  be a morphism in  $Min(\mathbf{X}\neg p, \mathbf{G}p)$  whose domain is  $m$ .

- $m \not\Vdash_0 p$  is impossible since  $h$  is a morphism and since  $n \Vdash \mathbf{G}p$ , we have that  $n \Vdash_0 p$  and  $m \Vdash_0 p$ .
- $m \Vdash_1 p$  is impossible since  $m$  is a model of  $\mathbf{X}\neg p$ .
- $\exists t^* > 1$  with  $m \Vdash_{t^*} p$  (which implies  $h(t^*) \uparrow$ ). In this condition we prove that  $h$  is not minimal, contradicting the hypothesis. Let  $h' : m' \rightarrow n'$  be the morphism such that

- $n' \Vdash_t p$  for all  $t \in \mathbb{N}$ ;
- $m' \Vdash_t p$  iff  $p \in \mathbb{N} - \{1\}$ ; and
- $h'(0) = 0$ ,  $h'(1) \uparrow$ , and  $h'(t) = t - 1$  for  $t > 1$ .

As we have seen,  $h' \leq_{\Sigma} h$ . To see that this relation is strict, i.e. that  $h \not\leq_{\Sigma} h'$  we note that

- if  $h'(t^* - 1) \downarrow$ , then  $m \xleftrightarrow{h}_{t^*-1} n$  and  $m' \not\xleftrightarrow{h'}_{t^*-1} n'$ . Moreover,  $\forall t < t^* - 1$  if  $m \xleftrightarrow{h}_t n$  and  $h(t) \downarrow$  then  $m' \not\xleftrightarrow{h'}_t n'$  and  $h'(t) \downarrow$ . Hence, it fails condition 4.
- In the case that  $h'(t^* - 1) \uparrow$ , it fails condition 5.