Abductive Reasoning over Temporal Specifications of Objects *

Paula Gouveia and Cristina Sernadas
Inst. Superior Técnico (Dep. de Matemática), Av. Rovisco Pais, 1000 Lisboa, Portugal
e-mail: {mpg,css}@math.ist.utl.pt

Abstract: In the context of reactive system specification, the certification task usually consists in ensuring that some envisaged property of the system behaviour follows from the specification. Considering, in particular, object-oriented system specification based on temporal logic, we have that, in general, the object specifications can only include certain kinds of formulae (e.g. constraints on action ocurrences and changes in attribute values) and the formulae expressing the intended properties of the system behaviour can be different. Therefore, whenever a property is not inferred from the specification, we can not just add it as a new specification axiom and traditional certification approaches do not use to provide help on how the specification could be enriched. Finding how to enrich a specification with suitable new axioms is an important issue that has not been explored. This task has an abuctive flavour and in this paper we address the problem of certification of temporal object specifications from this abductive perspective: the properties that are not satisfied are seen as abduction problems and the new axioms are seen explanations for these problems. The techniques we describe for the generation of explanations combine ideas from temporal logic proof techniques and from tableau based techniques for abductive reasoning. Given the particular relevance of certain properties, namely safety, response, persistence and guarantee properties, in the context of temporal specification of reactive systems, we concentrate our attention on the development of abductive techniques for these kinds of properties. We also describe the extension of these techniques to the case of object community specifications.

Keywords: temporal logic, object-oriented system specification and certification, abductive reasoning

1 Introduction

In many cases, when starting to specify a software system, the developers know that some properties should be inferred afterwards. Adopting a logic perspective, the problem in object-oriented contexts is that specifications should only include special kinds of formulae and, in general, formulae expressing the properties that have to be verified are much more. An object is a dynamic reactive entity capable of interacting with other objects. It has an internal state (represented by the values of its attributes) which can only be changed through the occurrence of its actions. Usually, when specifying the object behaviour, one can only include formulae that describe conditions over the initial values of attributes, the way the occurrence of actions modifies the attribute values as well as restrictions on its

*This work was partially supported by the PRAXIS XXI Program and FCT, as well as by PRAXIS XXI Projects 2/2.1/MAT/262/94 SitCalc, PCEX/P/MAT/46/96 ACL plus 2/2.1/TIT/1658/95 LogComp, and ESPRIT IV Working Groups 22704 ASPIRE and 23531 FIREworks.

occurrence/enabling and how the objects interact with each other. Therefore, there are many properties that cannot just be added as formulae to the specification.

A lot of work has been done with respect to certification of object-oriented system specifications. The goal is to be able to prove that some intended property of the system behaviour follows from its specification. However it seems that the traditional certification approach is not the only useful one. Herein, we address the problem of certification in the context of propositional temporal object specifications from a different point of view: whenever some envisaged property does not follow from the specification, we provide an enriched specification from which the property now follows.

Based on [12, 13], we consider an object specification as including a signature (a set of action symbols plus a set of attribute symbols), a set of temporal formulae (the specification axioms) and a set of valuation schemata (the specification axiom schemata). Some properties are recognized to be of particular relevance in the context of temporal specification of reactive systems [7, 8]. Following the classification presented therein we concentrate our attention on the generation of explanations for safety, response, guarantee and persistence problems. The enriched specification is obtained from the previous one adding new initial conditions over the attribute values and/or new restrictions on the enabling of actions and/or fairness requirements over action occurrences.

This kind of problem has an abductive flavour [11, 10]: the fact that the property does not follow from the specification is seen as an abduction problem and the set of new specification axioms is seen as an explanation for that problem. The techniques we use to generate the explanations are based on proof techniques in the context of temporal specification of reactive systems [7, 8] and on abductive reasoning techniques using tableaux [9]. We also present an extension of those techniques to the case of object community specifications, i.e., collections of interacting objects. In this case, new interactions between objects can also be generated. Previous work was already presented in [4] and further developed in [3].

The paper is structured as follows. In section 2 we present syntactic and semantic aspects of temporal object specification. Section 3 deals with abductive reasoning over object specifications. In subsection 3.1 we introduce the notion of abduction problem in the context of an object specification and the notion of explanation for an abduction problem. We provide the techniques for obtaining explanations for safety and response problems in the subsections 3.2 and 3.3, respectively. In subsection 3.4 we outline the extension of these techniques to the case of guarantee and persistence properties. In section 4 we introduce syntactic and semantic aspects of temporal object community specification. Section 5 presents an extension of the techniques described in section 3 with respect to safety properties to the case of object community specifications. In section 6 we present the concluding remarks and some comments on related and future work.

2 Specification of objects

In this section we introduce the syntactic and semantic aspects of temporal object specification. Based on [12, 13], an object specification includes an object signature, a set temporal formulae and a set of temporal formula schemata. The signature introduces the action and attribute symbols. As it is usual in object-oriented contexts, the formulae (the specification axioms) express constraints over the initial value of attributes and restrictions on action occurrence and action enabling. The formula schemata (the specification axiom schemata) are related to attribute updating.

Definition 2.1 An *object signature* is a pair

$$\Sigma = (Act, Att)$$

where $Act \neq \emptyset$ is a finite set of *action symbols* or *actions* and Att is a finite set of *attribute symbols* or *attributes*.

The set of formulae over Σ, F_Σ, is inductively defined as usual: (i) $\{\mathbf{t},\mathbf{f}\} \cup Att \subset F_\Sigma$; (ii) $\triangledown Act = \{\triangledown a : a \in Act\} \subset F_\Sigma$; (iii) $\diamond Act = \{\diamond a : a \in Act\} \subset F_\Sigma$; (iv) if $f, f' \in F_\Sigma$ then $\neg f, f \wedge f', \mathbf{X}\, f, (f' \,\mathbf{U}\, f) \in F_\Sigma$. ■

We consider the usual propositional abbreviations and the temporal abbreviations $\mathbf{F}$ and $\mathbf{W}$.

For each $a \in Act$, $\triangledown a$ expresses the occurrence of a and $\diamond a$ expresses that a is enabled. The enabling of action expresses that the occurrence of the action is allowed but not mandatory. An action only occurs if it is enabled. Objects are considered sequential entities in the sense that at each instant only one of its actions can occur.

Definition 2.2 An *interpretation structure* over $\Sigma = (Act, Att)$ is a map

$$\lambda : I\!N_0 \to \wp(Att \cup \triangledown Act \cup \diamond Act)$$

such that for each $k \in I\!N_0$: (i) $\lambda(k) \cap \triangledown Act$ is a singleton and (ii) if $\triangledown a \in \lambda(k)$ then $\diamond a \in \lambda(k)$. ■

Given $f \in F_\Sigma$, the *satisfaction* of f by λ at $k \in I\!N_0$, $\lambda, k \models f$, is defined as usual. We adopt the anchored version of temporal logic [6], i.e., an interpretation structure is a model of f iff $\lambda, 0 \models f$. We also assume that future states include the present state.

In the sequel, SC_Σ is the subset of F_Σ of all propositional formulae over $\{\mathbf{t},\mathbf{f}\} \cup Att$. A formula in SC_Σ is a *state constraint*. The set $FR_\Sigma = \bigcup_{a \in Act}\{(\mathbf{G}(\mathbf{F}\diamond a)) \Rightarrow (\mathbf{G}(\mathbf{F}\triangledown a))\}$ is also considered. Each formulae in FR_Σ expresses a *fairness requirement* over some action a stating that if a is recurrently enabled then a recurrently occurs. For each $a \in Act$, we introduce the sets $EC1^a_\Sigma = \{\mathbf{G}(\diamond a \Rightarrow q) : q \in SC_\Sigma\}$ and $EC2^a_\Sigma = \{\mathbf{G}(q \Rightarrow \diamond a) : q \in SC_\Sigma\}$. The formulae in $EC1^a_\Sigma$ and $EC2^a_\Sigma$ express *enabling conditions* over a. A formula in $EC1^a_\Sigma$ expresses a necessary condition on the enabling of a. A formula in $EC2^a_\Sigma$ expresses a sufficient condition on the enabling of a. In the sequel we consider $EC1_\Sigma = \bigcup_{a \in Act} EC1^a_\Sigma$ and $EC2_\Sigma = \bigcup_{a \in Act} EC2^a_\Sigma$ and $cnd(\mathbf{G}(\diamond a \Rightarrow q)) = q$. Formulae in $SC_\Sigma \cup EC1_\Sigma$ are important to specify safety aspects of the object behaviour and formulae in $EC2_\Sigma \cup FR_\Sigma$ are important to specify liveness aspects.

The way the occurrence of actions modifies the attribute values can be expressed through formulae $\mathbf{G}((\triangledown a \wedge q^{o_1 \ldots o_m}_{p_1 \ldots p_m}) \Rightarrow (\mathbf{X}\, q))$ where $a \in Act$, $o_1, \ldots, o_m$ are $m \geq 0$ distinct attribute symbols, $p_1, \ldots, p_m, q$ are state constraints and $q^{o_1 \ldots o_m}_{p_1 \ldots p_m}$ stands for the formula that is obtained from q by uniformly substituting each p_i for o_i, $0 \leq i \leq m$. This formula expresses that q holds after the occurrence of a whenever $q^{o_1 \ldots o_m}_{p_1 \ldots p_m}$ holds when a occurs. In particular, if q is o_i for some $0 \leq i \leq m$, the formula $\mathbf{G}((\triangledown a \wedge q^{o_1 \ldots o_m}_{p_1 \ldots p_m}) \Rightarrow (\mathbf{X}\, q))$ expresses that o_i is true after the occurrence of a whenever p_i holds before. When q is

$\neg o_i$ it expresses that o_i is false after the occurrence of a whenever $\neg p_i$ holds before. Given $o \in Att \backslash \{o_1, \ldots, o_m\}$ the formula with $q = o$ and the formula with $q = \neg o$ together state that the occurrence of a does not modify the value of o. To avoid the inclusion of a lot of such formulae in the specification, we consider formula schemata: $\mathbf{G}((\nabla a \wedge \alpha^{o_1 \cdots o_m}_{p_1 \ldots p_m}) \Rightarrow (\mathbf{X}\, \alpha))$, where α is a formula variable, is a *valuation schema*. The *q-instance* of this schema is the formula $\mathbf{G}((\nabla a \wedge q^{o_1 \cdots o_m}_{p_1 \ldots p_m}) \Rightarrow (\mathbf{X}\, q))$. In the sequel, S^a_Σ is the set of all valuation schemata for a, $S_\Sigma = \bigcup_{a \in Act} S^a_\Sigma$ and $\iota(S, C) = \{f \in F_\Sigma : f$ is the q-instance of some $s \in S$ and $q \in C\}$ for each $S \subseteq S_\Sigma$ and $C \subseteq SC_\Sigma$.

Definition 2.3 An *object specification* is a triple

$$ospc = (\Sigma, AxF, AxS)$$

where $\Sigma = (Act, Att)$ is an object signature, $AxF \subseteq SC_\Sigma \cup EC1_\Sigma \cup EC2_\Sigma \cup FR_\Sigma$ is finite and $AxS = \bigcup_{a \in Act} \{s_a\}$ where $s_a \in S^a_\Sigma$ for each $a \in Act$. ■

Definition 2.4 Let $ospc = (\Sigma, AxF, AxS)$ be an object specification where $\Sigma = (Act, Att)$. Let $a \in Act$, $p, q \in SC_\Sigma$, $f \in F_\Sigma$ and $A \subseteq F_\Sigma$:

- the interpretation structure λ over Σ is a *model* of *ospc* iff $\lambda \models AxF \cup \iota(AxS, SC_\Sigma)$; *ospc* is *consistent* iff it has a model.
- f (A) is *ospc-satisfiable* iff there is a model of *ospc* that is also a model of f (A); f is *ospc-valid*, $ospc \models f$, iff each model of *ospc* is also a model of f.
- *q is initially established in ospc* iff $ospc \models q$; *a establishes q from p in ospc* iff $ospc \models \mathbf{G}((\nabla a \wedge p) \Rightarrow (\mathbf{X}\, q))$. ■

Example 2.5 The triples $ospc_1 = (\Sigma_1, AxF_1, AxS_1)$ and $ospc_2 = (\Sigma_2, AxF_2, AxS_2)$ where

– $\Sigma_1 = (\{a_1, a_2, a_3, a_4\}, \{x, y, z\})$

– $AxF_1 = \{\neg y, \mathbf{G}(\diamond a_2 \Rightarrow z)\}$

$$\begin{aligned} AxS_1 = \{&\mathbf{G}((\nabla a_1 \wedge \alpha^{x\;y\;z}_{\neg z\;x\;\neg z}) \Rightarrow (\mathbf{X}\, \alpha)),\\ &\mathbf{G}((\nabla a_2 \wedge \alpha^{x\;y\;z}_{\neg y\;z\;\neg x}) \Rightarrow (\mathbf{X}\, \alpha)),\\ &\mathbf{G}((\nabla a_3 \wedge \alpha^{x\;y}_{\mathbf{t}\;\neg y}) \Rightarrow (\mathbf{X}\, \alpha)),\\ &\mathbf{G}((\nabla a_4 \wedge \alpha^{x\;y}_{\neg x\;\mathbf{t}}) \Rightarrow (\mathbf{X}\, \alpha))\} \end{aligned}$$

and

– $\Sigma_2 = (\{a_1, a_2, a_3\}, \{x, y, z, u\})$

– $AxF_2 = \{y\}$

$$\begin{aligned} AxS_2 = \{&\mathbf{G}((\nabla a_1 \wedge \alpha^{y\;z\;u}_{\mathbf{f}\;x\;\mathbf{f}}) \Rightarrow (\mathbf{X}\, \alpha)),\\ &\mathbf{G}((\nabla a_2 \wedge \alpha^{y\;z\;u}_{\mathbf{t}\;\mathbf{f}\;z}) \Rightarrow (\mathbf{X}\, \alpha)),\\ &\mathbf{G}((\nabla a_3 \wedge \alpha^{x}_{\neg x}) \Rightarrow (\mathbf{X}\, \alpha))\} \end{aligned}$$

are object specifications. ■

3 Abductive reasoning over object specifications

In this section we first introduce the notion of abduction problem in the context of an object specification and the notion of explanation for an abduction problem. Then we show how explanations for two kinds of abduction problems (safety and response) can be obtained. Finally, we outline the construction of explanations for guarantee and persistence problems. The generation of explanations relies on proof techniques in the context of temporal specification of reactive systems [7, 8] and on abductive reasoning techniques using tableaux [9].

In the sequel we assume a given object specification $ospc = (\Sigma, AxF, AxS)$, where $\Sigma = (Act, Att)$.

3.1 Abduction problems and explanations

In this subsection we introduce the notion of abduction problem in *ospc* and the notion of explanation for an abduction problem. An abduction problem represents some envisaged specification requirement that is not satisfied by *ospc*. As a consequence, one has to introduce further restrictions on the object behaviour which means that *ospc* has to be enriched with new axioms. Since the object behaviour can only be described through special kinds of formulae, in general, one can not just add a formula expressing that requirement as a new axiom of the specification. Therefore, new suitable axioms have to be found in order to enrich the specification in such a way that the specification requirement should be inferred from the enriched specification. Using an usual designation in abductive contexts, we say that these new axioms constitute an explanation for the abduction problem.

Definition 3.1 A formula $f \in F_\Sigma$ is an *abduction problem* in *ospc* iff

$$ospc \not\models f.$$

An *explanation* for an abduction problem $f \in F_\Sigma$ in *ospc* is a finite set

$$E \subseteq SC_\Sigma \cup EC1_\Sigma \cup EC2_\Sigma \cup FR_\Sigma$$

such that $ospc' = (\Sigma, AxF \cup E, AxS)$ is consistent and $ospc' \models f$. ∎

In the context of reactive system behaviour specification some properties have particular significance. Following the classification of properties of reactive systems specifiable by temporal formulae presented, for instance, in [7], we focus our attention in some special kinds of abduction problems: safety, response, guarantee and persistence problems.

Definition 3.2 We say that $f \in F_\Sigma$ is a

- *safety* problem iff $f \in \{\mathbf{G}(p \Rightarrow (\mathbf{X}\, q)), p \Rightarrow (q\, \mathbf{W}\, r), \mathbf{G}(p \Rightarrow (q\, \mathbf{W}\, r)) : p, q, r \in SC_\Sigma\}$;
- *response* problem iff $f \in \{\mathbf{G}(p \Rightarrow (q\, \mathbf{U}\, r)) : p, q, r \in SC_\Sigma\}$;
- *guarantee* problem iff $f \in \{p \Rightarrow (q\, \mathbf{U}\, r) : p, q, r \in SC_\Sigma\}$;
- *persistence* problem iff $f \in \{\mathbf{G}(p \Rightarrow (\mathbf{F}(\mathbf{G}\, q))) : p, q \in SC_\Sigma\}$. ∎

In particular, given $q \in SC_\Sigma$, the abduction problem $\mathbf{G}\, q$ (expressing the invariance of q) is a safety problem, the abduction problem $\mathbf{G}(\mathbf{F}\, q)$ is a response problem and $\mathbf{F}\, q$ is a guarantee problem.

Example 3.3 Consider $ospc_1$ in Example 2.5. We have that $\mathbf{G}(x \vee y)$ is a safety problem in $ospc_1$. Considering $ospc_2$ we have that $\mathbf{G}(y \Rightarrow (z\,\mathbf{U}\,u))$ and $\mathbf{G}(y \Rightarrow (z\,\mathbf{W}\,u))$ are a response problem and a safety problem in $ospc_2$, respectively. ∎

In general there are several possible explanations for a given abduction problem and, of course, some of them can be considered more interesting than others. So, as it is usual in abductive tasks, we establish some preference criteria between explanations that seem to be adequate in this context. The main idea is to give preference to those explanations that do not restrict the behaviour of the object more than it is necessary for the purpose at hand.

3.2 Explanations for safety problems

In this subsection we present the techniques for the generation of explanations for safety problems. They combine proof techniques for safety properties in the context of temporal specification of reactive systems [8] with tableau based abductive techniques [9].

In the sequel we assume the state constraints $p, q, r \in SC_\Sigma$.

Proposition 3.4

1. If $ospc \models \mathbf{G}((\nabla a \wedge p) \Rightarrow (\mathbf{X}\,q))$ for each $a \in Act$ then $ospc \models \mathbf{G}(p \Rightarrow (\mathbf{X}\,q))$.
2. If $ospc \models p \Rightarrow (q \vee r)$ and $ospc \models \mathbf{G}((q \wedge (\neg r)) \Rightarrow (\mathbf{X}(q \vee r)))$ then $ospc \models p \Rightarrow (q\,\mathbf{W}\,r)$.
3. If $ospc \models \mathbf{G}(r \Rightarrow (q \vee r))$ and $ospc \models \mathbf{G}((q \wedge (\neg r)) \Rightarrow (\mathbf{X}(q \vee r)))$ then $ospc \models \mathbf{G}(p \Rightarrow (q\,\mathbf{W}\,r))$.

Proof: Let λ be a model of *ospc*.

1. Let $k \in I\!N_0$ such that $\lambda, k \models p$. By Definition 2.2, $\lambda, k \models \nabla a$ for some $a \in Act$ and therefore $\lambda, k \models \mathbf{X}\,q$.

2. Assuming that $\lambda, 0 \models p$ then $\lambda, 0 \models r$ or $\lambda, 0 \models q$. In the first case we are done. In the second case, either $\lambda, k \models q \wedge (\neg r)$ for all $k \geq 0$ (and again the result easily follows) or there is $k' > 0$ such that $\lambda, k' \not\models q \wedge (\neg r)$ and $\lambda, k \models q \wedge (\neg r)$ for all $0 \leq k < k'$. If $\lambda, k' \not\models \neg r$ then once more the result follows. If $\lambda, k' \not\models q$, since $\lambda, k' - 1 \models q \wedge (\neg r)$, we have that $\lambda, k' \models r$ and again the result follows.

3. Given $k \in I\!N_0$ such that $\lambda, k \models p$ the result follows reasoning as in 2. ∎

¿From the above proposition we see that if we find an explanation for $\mathbf{G}((\nabla a \wedge p) \Rightarrow (\mathbf{X}\,q))$ for each action a that does not establish q from p, taking all those explanations we can obtain an explanation for a safety problem $\mathbf{G}(p \Rightarrow (\mathbf{X}\,q))$.

In what concerns safety problems of type $p \Rightarrow (q\,\mathbf{W}\,r)$ we have to find explanations for the initial establishment of $p \Rightarrow (q \vee r)$ and/or explanations for the safety problem $\mathbf{G}((q \wedge (\neg r)) \Rightarrow (\mathbf{X}(q \vee r)))$ which is a safety problem of a previous type. The particular case where $p = \mathbf{t}$ and $r = \mathbf{f}$ corresponds to a safety problem $\mathbf{G}\,q$. In this case we have to consider explanations for the initial establishment of q and/or explanations for $\mathbf{G}(q \Rightarrow (\mathbf{X}\,q))$.

Finally, explanations for $\mathbf{G}(p \Rightarrow (q\,\mathbf{W}\,r))$ can be obtained from explanations for the previous two kinds of safety problems.

Explanations for the initial establishment of q in *ospc* can be generated using a tableau for a set of formulae taken from AxF and including $\neg q$. We say that this set is the *generator* *of explanations for the initial establishment of* q *in ospc* and we denote it by $G^{\overset{i}{\to} q}_{ospc}$. Its definition is presented in the context of Proposition 3.8. In a similar way, explanations for the establishment of q from p by $a \in Act$ in *ospc* can be generated using a tableau for a set of formulae which are chosen taken into account formulae in AxF, schemata in AxS and including $\neg p$. We say that this set is the *generator of explanations for the establishment of* q *from* p *by* $a \in Act$ *in ospc* and we denote it by $G^{p \overset{a}{\to} q}_{ospc}$. Its definition is presented in the context of Proposition 3.8. Since these sets are finite subsets of SC_{Σ}, we just need propositional tableaux.

We consider propositional tableaux as in [1] and the usual notions of *closed*, *open* and *exhausted tableau branch* and *closed*, *open* and *exhausted tableau.* Recall that there is a closed tableau for some set A of propositional formulae iff A is an inconsistent set. For our purposes we also need the notion of closure of a tableau.

Definition 3.5 Let t be and exhausted open tableau for a finite set A of propositional formulae. A *closure* of t is a non empty and satisfiable set of literals Cl such that for each open branch ρ of t there is a literal l in ρ such that its conjugate, $\bar{l}$, is in Cl.

A closure Cl of t is said to be *minimal* if there is no other closure Cl' of t such that $Cl' \subset Cl$. ■

Example 3.6 Consider the object specification $ospc_1$ introduced in Example 2.5. Figure 1 depicts a tableau for $\{x \vee y, \neg((\neg z) \vee x)\} \subseteq SC_{\Sigma_1}$ which is open and exhausted.

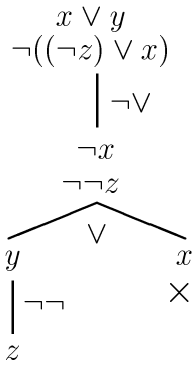

Figure 1: Tableau for $\{x \vee y, \neg((\neg z) \vee x)\}$

The sets $Cl_1 = \{x\}$, $Cl_2 = \{\neg y\}$ and $Cl_3 = \{\neg z\}$ are minimal closures of the tableau. ■

Let $Cl_1, \ldots, Cl_n$, $n \geq 1$, be closures of a tableau. In the sequel, for each $1 \leq i \leq n$, $\bigwedge Cl_i$ represents a conjunction of the literals in Cl_i and $\bigvee_{1 \leq i \leq n}(\bigwedge Cl_i)$ represents a disjunction of those conjunctions.

Proposition 3.7 Let t be an exhausted open tableau for a set A of propositional formulae. If $Cl_1, \ldots, Cl_n$, where $n \geq 1$, are closures of t then there is a closed tableau for the set $A \cup \{\bigvee_{1 \leq i \leq n}(\bigwedge Cl_i)\}$ and therefore this set is inconsistent.

Proof: Assuming the usual tableau construction rule $\vee$ ($\wedge$) generalized to disjunctions (conjunctions) of more than two formulae, a closed tableau t' for $A \cup \{\bigvee_{1 \leq i \leq n}(\bigwedge Cl_i)\}$ can be easily obtained from t as follows. For each open branch ρ of t: (i) apply rule $\vee$ to

$\bigvee_{1\leq i\leq n}(\bigwedge Cl_i)$ obtaining new branches ρ_i^1, $1\leq i\leq n$, each ρ_i^1 extending ρ with a node labeled with $\bigwedge Cl_i$; (ii) for each ρ_i^1, $1\leq i\leq n$, apply rule $\wedge$ to $\bigwedge Cl_i$ obtaining the branch ρ_i^2 which extends ρ_i^1 with a node labeled with the literals in Cl_i. By Definition 3.5, t' is closed since $Cl_1,\ldots,Cl_n$ are closures of t. ∎

We can now describe how we can generate the explanations for the initial establishment of q and for the establishment of q from p by $a\in Act$ in *ospc*.

Proposition 3.8 Let $a\in Act$ and assume that $AxS\cap S^a_\Sigma=\{\mathbf{G}((\triangledown a\wedge\alpha^{o_1\cdots o_n}_{p_1\ldots p_n})\Rightarrow(\mathbf{X}\,\alpha))\}$.

1. If there is a closed tableau for $G^{\overset{i}{\to}q}_{ospc}=(AxF\cap SC_\Sigma)\cup\{\neg q\}$ then q is initially established in *ospc*.
2. If there is a closed tableau for $G^{p\overset{a}{\to}q}_{ospc}=cnd(AxF\cap EC1^a_\Sigma)\cup\{p,\neg q^{o_1\cdots o_n}_{p_1\ldots p_n}\}$ then a establishes q from p in *ospc*.
3. If $Cl_1,\ldots,Cl_n$ are closures of an exhausted open tableau for $G^{\overset{i}{\to}q}_{ospc}$ and
$$\overline{p}=\bigvee_{1\leq i\leq n}(\bigwedge Cl_i)$$
then q is initially established in $ospc'=(\Sigma,AxF\cup\{\overline{p}\},AxS)$.
4. If $Cl_1,\ldots,Cl_n$ are closures of an exhausted open tableau for $G^{p\overset{a}{\to}q}_{ospc}$ and
$$\overline{f}=\mathbf{G}(\diamond a\Rightarrow\bigvee_{1\leq i\leq n}(\bigwedge Cl_i))$$
then a establishes q from p in $ospc'=(\Sigma,AxF\cup\{\overline{f}\},AxS)$.

Proof: 1. Let λ be a model of *ospc*. By Definition 2.4, $\lambda\models AxF\cap SC_\Sigma$. Since $G^{\overset{i}{\to}q}_{ospc}$ is inconsistent, $\lambda\models q$.

2. Let λ be a model of *ospc* and $k\in I\!N_0$ such that $\lambda,k\models\triangledown a\wedge p$. By Definitions 2.2 and 2.4, $\lambda,k\models\diamond a$ and $\lambda,k\models cnd(AxF\cap EC1^a_\Sigma)$. Since $G^{p\overset{a}{\to}q}_{ospc}$ is inconsistent $\lambda,k\models q^{o_1\cdots o_n}_{p_1\ldots p_n}$. By Definition 2.4, $\lambda,k\models\mathbf{G}((\triangledown a\wedge q^{o_1\cdots o_n}_{p_1\ldots p_n})\Rightarrow(\mathbf{X}\,q))$ and therefore $\lambda,k\models\mathbf{X}\,q$.

3. Taking into account *ospc*$'$ it follows that $G^{\overset{i}{\to}q}_{ospc'}=G^{\overset{i}{\to}q}_{ospc}\cup\{\overline{p}\}$. By Proposition 3.7, there is a closed tableau for $G^{\overset{i}{\to}q}_{ospc'}$. Thus, by 1., q is initially established in *ospc*$'$.

4. Taking into account *ospc*$'$ it follows that $G^{p\overset{a}{\to}q}_{ospc'}=G^{p\overset{a}{\to}q}_{ospc}\cup\{\bigvee_{1\leq i\leq n}(\bigwedge Cl_i))\}$. By Proposition 3.7, there is a closed tableau for $G^{p\overset{a}{\to}q}_{ospc'}$. Thus, by 2., a establishes q from p in *ospc*$'$. ∎

¿From the above proposition it follows that: (i) if $\overline{p}$ is *ospc*-satisfiable then $\{\overline{p}\}$ constitutes an explanation for the initial establishment of q in *ospc* and (ii) if $\overline{f}$ is *ospc*-satisfiable then $\{\overline{f}\}$ constitutes an explanation for the establishment of q from p by a in *ospc*. In the sequel, we often identify any explanation which is a singleton with the formula it contains.

In some cases it is not possible to find a closure for an exhausted tableau for $G^{p\overset{a}{\to}q}_{ospc}$ because all sets of literals obtained from its open branches are unsatisfiable. This corresponds to the cases where a establishes $\neg q$ (see [3]). Therefore, the only way to restrict the occurrence of a in order to establish q from p is to forbid the occurrence of a. In this situation $\mathbf{G}(\diamond a \Rightarrow \mathbf{f})$ can provide an explanation.

An important question is which tableau closures should one choose in order to get more interesting explanations . Given $\overline{p}$ and $\overline{p}'$ obtained from tableau closures as described in Proposition 3.8 (3), we say that $\overline{p}$ is more *liberal* than $\overline{p}'$ iff $\models \overline{p}' \Rightarrow \overline{p}$. Given $\overline{f}$ and $\overline{f}'$ obtained from tableau closures as described in Proposition 3.8 (4), we say that $\overline{f}$ is more *liberal* than $\overline{f}'$ iff $\models cnd(\overline{f}') \Rightarrow cnd(\overline{f})$. This notion is used as preference criterion between explanations: we prefer the more liberal ones since they restrict less the object behaviour. The more liberal explanations can be obtained from disjunctions of minimal closures (see [3]).

Example 3.9 In the context of the object specification $ospc_1$ introduced in Example 2.5 we have that

$$G_{ospc_1}^{x \vee y \overset{a_1}{\rightarrow} x \vee y} = \{x \vee y, \neg((\neg z) \vee x)\}$$

and, taking into account the tableau and the tableau closures presented in Example 3.6, we have that

$$f_1 = \mathbf{G}(\diamond a_1 \Rightarrow x) \qquad f_2 = \mathbf{G}(\diamond a_1 \Rightarrow (\neg y)) \qquad f_3 = \mathbf{G}(\diamond a_1 \Rightarrow (\neg z))$$
$$f_4 = \mathbf{G}(\diamond a_1 \Rightarrow (x \vee (\neg y) \vee (\neg z)))$$

are explanations for the establishment of $x \vee y$ from $x \vee y$ by a_1 in $ospc_1$. The explanation f_4 is more liberal than the explanations f_1, f_2 and f_3. ■

In what concerns explanations for the establishment of q from p by a, another preference criterion is also considered: explanations should take into account q and p as much as possible. Assume, for instance, that $\mathbf{G}(\diamond a \Rightarrow o) \in AxF$, for some $a \in Act$ and $o \in Att$. The enabling constraint $\mathbf{G}(\diamond a \Rightarrow (\neg o))$ could be considered as an explanation for the establishment of q from p by a. In fact, in the enriched specification, the occurrence of a is not possible since a it would only be enabled whenever we had both o and $\neg o$. Therefore a trivially establishes q from p in the new specification. This situation can be generalized as the following proposition indicates.

Proposition 3.10 Let $a \in Act$. Assume that $Cl_1, \ldots, Cl_n$, $n \geq 1$, are closures both of an open exhausted tableau for the set $G_{ospc}^{p \overset{a}{\rightarrow} q}$ and of an open exhausted tableau for the set $cnd(AxF \cap EC1_{\Sigma}^{a})$. If $\overline{f} = \mathbf{G}(\diamond a \Rightarrow \bigvee_{1 \leq i \leq n}(\bigwedge Cl_i))$ is $ospc$-satisfiable then $\overline{f}$ is an explanation for the establishment of q' from p' by a for any $p', q' \in SC_{\Sigma}$.

Proof: Let $p', q' \in SC_{\Sigma}$ and $ospc' = (\Sigma, AxF \cup \{\overline{f}\}, AxS)$. ¿From Proposition 3.7, there is a closed tableau for $cnd(AxF \cap EC1_{\Sigma}^{a}) \cup \{\bigvee_{1 \leq i \leq n}(\bigwedge Cl_i)\}$ and, as a consequence, this set is inconsistent. On the other hand, by Proposition 3.8, $cnd(AxF \cap EC1_{\Sigma}^{a}) \subset G_{ospc}^{p' \overset{a}{\rightarrow} q'}$ and $G_{ospc'}^{p' \overset{a}{\rightarrow} q'} = G_{ospc}^{p' \overset{a}{\rightarrow} q'} \cup \{\bigvee_{1 \leq i \leq n}(\bigwedge Cl_i)\}$ which means that $G_{ospc'}^{p' \overset{a}{\rightarrow} q'}$ is inconsistent. Therefore, there exists a closed tableau for it and so, by Proposition 3.8, a establishes q' from p' in $ospc'$. ■

¿From Proposition 3.10, we see that the explanations obtained from the closures referred therein are not very interesting since they do not take into account the problem at hand. Therefore, we give preference to explanations that come from closures that do not verify the above condition. The generation of these unwanted explanations can be easily prevented.

Another question is the following: do the closures we obtain from tableaux for $G_{ospc}^{\stackrel{i}{\to} q}$ and $G_{ospc}^{p \stackrel{a}{\to} q}$ depend on the particular open exhausted tableau for them we consider? The answer is negative: they are independent on the particular open exhausted tableau. In [3] we give a proof of the following result: if t and t' are open exhausted tableaux for the same set of propositional formulae then they have that same set of closures.

Finally, the following proposition summarizes the methods for the generation of explanations for the safety problems described in Definition 3.2.

Proposition 3.11 Let $A_{p_1p_2} = \{a \in Act\colon ospc \not\models \mathbf{G}((\nabla a \wedge p_1) \Rightarrow (\mathbf{X}\, p_2))\}$ for $p_1, p_2 \in SC_\Sigma$. Assume that, for each $a \in A_{p_1p_2}$, E_a is an explanation for the establishment of p_2 from p_1 by a in *ospc*.

1. If
$$E = \bigcup_{a \in A_{pq}} E_a$$
is *ospc*-satisfiable then E is an explanation for the safety problem $\mathbf{G}(p \Rightarrow (\mathbf{X}\, q))$ in *ospc*.

2. Let $q_1 = p \Rightarrow (q \vee r)$, $q_2 = q \wedge (\neg r)$ and $q_3 = q \vee r$ and
$$E = E^i_{q_1} \cup (\bigcup_{a \in A_{q_2q_3}} E_a)$$
where $E^i_{q_1}$ is an explanation for the initial establishment of q_1 in *ospc* if $ospc \not\models q_1$ and $E^i_{q_1} = \emptyset$ otherwise. If E is *ospc*-satisfiable then E is an explanation for the safety problem $p \Rightarrow (q\, \mathbf{W}\, r)$ in *ospc*.

3. Let
$$E = E_1 \cup E_2$$
where (i) E_1 is an explanation for the safety problem $\mathbf{G}(p \Rightarrow (q \vee r))$ whenever it is in fact a problem and $E_1 = \emptyset$ otherwise and (ii) E_2 is an explanation for the safety problem $\mathbf{G}((q \wedge (\neg r) \Rightarrow (\mathbf{X}(q \vee r)))$ whenever it is in fact a problem and $E_2 = \emptyset$ otherwise. If E is *ospc*-satisfiable then E is an explanation for the safety problem $\mathbf{G}(p \Rightarrow (q\, \mathbf{W}\, r))$ in *ospc*.

Proof: Let $ospc' = (\Sigma, AxF \cup E, AxS)$ and λ a model of $ospc'$.

1. Let $a \in Act$. If $a \notin A_{pq}$ then a establishes q from p in *ospc* and thus also in $ospc'$. If $a \in A_{pq}$, then a establishes q from p in $ospc'' = (\Sigma, AxF \cup E_a, AxS)$ and therefore a establishes q from p in $ospc'$. By Proposition 3.4.1., $ospc' \models \mathbf{G}(p \Rightarrow (\mathbf{X}\, q))$. Since E is *ospc*-satisfiable, $ospc'$ is consistent and the result follows.

2. If q_1 is initially established in *ospc* then it is also initially established in $ospc'$. Otherwise, q_1 is initially established in $ospc'' = (\Sigma, AxF \cup E^i_{q_1}, AxS)$ and so its initially established in $ospc'$. Given $a \in Act$ and reasoning as in 1. $ospc' \models \mathbf{G}(q_2 \Rightarrow (\mathbf{X}\, q_3))$. By Proposition 3.4.2., $ospc' \models p \Rightarrow (q\, \mathbf{W}\, r)$. Since E is *ospc*-satisfiable, $ospc'$ is consistent and the result follows.

3. If $\mathbf{G}(p \Rightarrow (q \vee r))$ is *ospc*-valid then it is also *ospc*$'$-valid. Otherwise it is *ospc*$''$-valid where *ospc*$'' = (\Sigma, AxF \cup E_1, AxS)$ and again it is also *ospc*$'$-valid. The case of $\mathbf{G}((q \wedge (\neg r) \Rightarrow (\mathbf{X}(q \vee r)))$ is similar. By Proposition 3.4.3, *ospc*$' \models \mathbf{G}(p \Rightarrow (q\, \mathbf{W}\, r))$. Since E is *ospc*-satisfiable, *ospc*$'$ is consistent and the result follows. ■

The particular case of 2. where $p = \mathbf{t}$ and $r = \mathbf{f}$ corresponds to an explanation for a safety problem $\mathbf{G}\, q$ in *ospc*. In this case, q_1, q_2 and q_3 are all equivalent to q.

The *ospc*-satisfaction of the sets E as considered in the above Proposition, i.e., the consistency of *ospc*$' = (\Sigma, AxF \cup E, AxS)$, can be checked using temporal tableaux as for instance the ones presented in [5].

Example 3.12 Considering the object specification *ospc*$_1$ introduced in Example 2.5 we have that $\mathbf{G}(x \vee y)$ is a safety problems in *ospc*$_1$.

1. The formula $x \vee y$ is not initially established in *ospc*$_1$. From an exhausted tableau for the set $G^{\overset{i}{\to} x \vee y}_{ospc_1} = \{\neg y, \neg(x \vee y)\}$ we easily obtain x as an explanation for the initial establishment of $x \vee y$ in *ospc*$_1$.

2. The action a_1 does not establish $x \vee y$ from $x \vee y$ in *ospc*$_1$. In Example 3.9, several explanations for the establishment of $x \vee y$ from $x \vee y$ from a_1 were indicated being $f_4 = \mathbf{G}(\diamond a_1 \Rightarrow (x \vee (\neg y) \vee (\neg z))$ the more liberal one. However, in some situations it can happen that a more liberal explanation should not be the more interesting one. Since our goal is to have $x \vee y$ as an invariant of the specification, $x \vee y$ should hold in every state. On the other hand, we want a_1 to preserve $x \vee y$. Ensuring that a_1 only occurs provided that $x \vee (\neg y) \vee (\neg z)$ holds or ensuring that a_1 only occurs provided that $x \vee (\neg z)$ holds is, in this particular situation, the same. Taking into account that $x \vee y$ also holds, whenever we have $\neg y$ we also must have x. So the information that a_1 is allowed to occur whenever we have $\neg y$ is, *in this case*, superfluous. Therefore, the explanation $\mathbf{G}(\diamond a_1 \Rightarrow (x \vee (\neg z))$ (obtained from the closures Cl_1 and Cl_3 of Example 3.6) is more interesting since it is simpler and it expresses the same information with respect to the conditions in which a can be enabled. This can be generalized to the case of any safety problem $\mathbf{G}\, q$. The situation can be detected and the generation of these more simple explanations ensured. The idea is to discard a minimal closure Cl of the tableau for $G^{q \overset{a}{\to} q}_{ospc}$ if there is another minimal closure Cl' of it such that $((\bigwedge Cl) \wedge q) \Rightarrow (\bigwedge Cl')$ is valid (this validity can be easily tested using a tableau).

3. The actions a_2, a_3 and a_4 establish $x \vee y$ from $x \vee y$ in *ospc*$_1$.

4. The set $E = \{x,\ \mathbf{G}(\diamond a_1 \Rightarrow (x \vee (\neg z)))\}$ is an explanation for the safety problem $\mathbf{G}(x \vee y)$ in *ospc*$_1$. ■

Example 3.13 Considering the object specification *ospc*$_2$ introduced in Example 2.5 we have that $\mathbf{G}(y \Rightarrow (z\, \mathbf{W}\, u))$ is a safety problem in *ospc*$_2$.

1. The formula $\mathbf{G}(y \Rightarrow (z \vee u))$ is a safety problem in *ospc*$_2$.

1.1. The actions a_1 and a_3 establish $y \Rightarrow (z \vee u)$ from $y \Rightarrow (z \vee u)$ in *ospc*$_2$.

1.2. The action a_2 does not establish $y \Rightarrow (z \vee u)$ from $y \Rightarrow (z \vee u)$ in *ospc*$_2$. ¿From the closures $Cl_1 = \{z\}$ and $Cl_2 = \{y, \neg u\}$ of the exhausted tableau for the set $G^{y \Rightarrow (z \vee u) \overset{a_2}{\to} y \Rightarrow (z \vee u)}_{ospc_2} = \{y \Rightarrow (z \vee u), \neg(\mathbf{t} \Rightarrow (\mathbf{f} \vee z))\}$ depicted in Figure 2, we can obtain $\mathbf{G}(\diamond a_2 \Rightarrow (z \vee (y \wedge (\neg u))))$

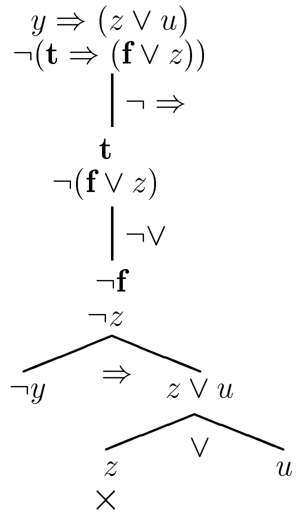

Figure 2: Tableau for $\{y \Rightarrow (z \vee u), \neg(\mathbf{t} \Rightarrow (\mathbf{f} \vee z)\}$

as an explanation for the establishment of $y \Rightarrow (z \vee u)$ from $y \Rightarrow (z \vee u)$ by a_2. However, since we want $y \Rightarrow (z \vee u)$ to be an invariant, reasoning as in **2.** of Example 3.12 we can conclude that $f_1 = \mathbf{G}(\diamond a_2 \Rightarrow z)$ is more simple and, in this particular context, allows the occurrence of a_2 as much as the enabling constraint above.

1.3. The formula $y \Rightarrow (z \vee u)$ is not initially established in $ospc_2$. From an exhausted tableau for the set $G_{ospc_2}^{\xrightarrow{i} y \Rightarrow (z \vee u)} = \{y, \neg(y \Rightarrow (z \vee u))\}$ we can obtain $f_2 = z \vee u$ as an explanation for the initial establishment of $y \Rightarrow (z \vee u)$.

2. The formulae $\mathbf{G}((z \wedge (\neg u)) \Rightarrow (\mathbf{X}(z \vee u)))$ is a safety problem in $ospc_2$.

2.1. The action a_2 and a_3 establish $z \vee u$ from $z \wedge (\neg u)$.

2.2. The action a_1 does not establish $z \vee u$ from $z \wedge (\neg u)$ in $ospc_2$. From an exhausted tableau for the set $G_{ospc_2}^{z \wedge (\neg u) \xrightarrow{a_1} z \vee u} = \{\neg(x \vee \mathbf{f}), z \wedge (\neg u)\}$ we can obtain the explanation $f_3 = \mathbf{G}(\diamond a_1 \Rightarrow (x \vee (\neg z) \vee u))$ for the establishment of $z \vee u$ from $z \wedge (\neg u)$ by a_1.

3. Finally, $E = \{f_1, f_2, f_3\}$ is an explanation for the safety problem $\mathbf{G}(y \Rightarrow (z\, \mathbf{W}\, u))$. ∎

3.3 Explanations for response problems

In this subsection we describe the techniques for the generation of explanations for response properties. They rely on proof techniques for response properties in the context of temporal specification of reactive systems [7] and on the techniques for the generation of explanations for safety properties presented above.

Proposition 3.14 If

(i) $ospc \models \mathbf{G}(p \Rightarrow (q\, \mathbf{W}\, r))$

(ii) there is $a \in Act$ such that

- $ospc \models \mathbf{G}((\nabla a \wedge q \wedge (\neg r)) \Rightarrow (\mathbf{X}\, r))$,
- $ospc \models \mathbf{G}((q \wedge (\neg r)) \Rightarrow \diamond a)$

– $ospc \models (\mathbf{G}(\mathbf{F} \diamond a)) \Rightarrow (\mathbf{G}(\mathbf{F} \triangledown a))$

then $ospc \models \mathbf{G}(p \Rightarrow (q \,\mathbf{U}\, r))$.

Proof: Let λ be a model of *ospc* and $k \in I\!N_0$ such that $\lambda, k \models r$. By (i), $\lambda, k \models q \,\mathbf{W}\, r$ which means that (a) $\lambda, k \models q \,\mathbf{U}\, r$ (and we are done) or (b) $\lambda, k \models \mathbf{G}(q \wedge (\neg r))$. Assuming (b), from (ii) we have that $\lambda, 0 \models \mathbf{G}(\mathbf{F} \diamond a))$ and $\lambda, 0 \models \mathbf{G}(\mathbf{F} \triangledown a))$. Thus there is $k' \geq k$ such that $\lambda, k' \models \triangledown a$. Since we also have $\lambda, k' \models q \wedge (\neg r)$ then $\lambda, k' + 1 \models r$ which contradicts (b). ∎

Based on the above proposition it is easy to see how we can generate explanations for response problems. Considering some $a \in Act$, we can obtain an explanation for a response problem $\mathbf{G}(p \Rightarrow (q \,\mathbf{U}\, r))$ in *ospc* as follows. Each explanation contains an explanation for the safety problem $\mathbf{G}(p \Rightarrow (q \,\mathbf{W}\, r))$ and an explanation for the establishment of $q \vee r$ from $q \wedge (\neg r)$ by a (whenever they are in fact abduction problems in *ospc*). It also contains the fairness requirement for a and the enabling constraint $\mathbf{G}((q \wedge (\neg r)) \Rightarrow \diamond a)$ (whenever they are not *ospc*-valid). We say that such an explanation is based on action a. As before, consistency of the final set has to be checked.

Example 3.15 Considering the object specification $ospc_2$ introduced in Example 2.5 we have that $\mathbf{G}(y \Rightarrow (z \,\mathbf{U}\, u))$ is a response problem in $ospc_2$.

1. The formula $\mathbf{G}(y \Rightarrow (z \,\mathbf{W}\, u))$ is a safety problem in $ospc_2$. In Example 3.13 we have already obtained an explanation for it.

2. Consider the action a_2.

2.1. We have that $ospc_2 \models \mathbf{G}((\triangledown a_2 \wedge z \wedge (\neg u)) \Rightarrow (\mathbf{X}\, u))$.

2.2. We have that $ospc_2 \not\models \mathbf{G}((z \wedge (\neg u)) \Rightarrow \diamond a_2)$ and $ospc \not\models (\mathbf{G}(\mathbf{F} \diamond a_2)) \Rightarrow (\mathbf{G}(\mathbf{F} \triangledown a_2))$.

3. The set

$$\begin{aligned} E' = \{ & z \vee u,\ \mathbf{G}(\diamond a_1 \Rightarrow (x \vee (\neg z) \vee u)),\ \mathbf{G}(\diamond a_2 \Rightarrow z), \\ & \mathbf{G}((z \wedge (\neg u)) \Rightarrow \diamond a_2), (\mathbf{G}(\mathbf{F} \diamond a_2)) \Rightarrow (\mathbf{G}(\mathbf{F} \triangledown a_2))\} \end{aligned}$$

is an explanation for the response problem $\mathbf{G}(y \Rightarrow (z \,\mathbf{U}\, u))$ based on action a_2. ∎

The explanations for response problems $\mathbf{G}(p \Rightarrow (q \,\mathbf{U}\, r))$ just presented constitute what we call *one step explanations* since they rely on one action whose occurrence will establish r. In [3] we also consider *several steps explanations* which are inspired in proof techniques for response properties in several steps presented, for instance, in [7]. In this case, the explanations rely on the occurrence of several actions that contribute to the final verification of r. For instance, in the case of response problems $\mathbf{G}(\mathbf{F}\, r))$, we can obtain explanations based on $a_1, \ldots, a_n \in Act$ and $q_1, \ldots, q_n \in SC_\Sigma$ where q_n is r. The state constraints $q_1, \ldots, q_n$ give a measure of how far we are from satisfying r. The occurrence of each action in *Act* should not increase that distance and the occurrence of each a_i, $1 \leq i \leq n$, (ensured through fairness requirements) should establish q_i and, as a consequence, decrease that distance.

3.4 Explanations for guarantee and persistence properties

In this subsection we briefly outline the techniques for the generation of explanations for guarantee and persistence problems.

With respect to guarantee problems, we have the following result.

Proposition 3.16 If

(i) $ospc \models p \Rightarrow (q\,\mathbf{W}\,r)$

(ii) there is $a \in Act$ such that

- $ospc \models \mathbf{G}((\triangledown a \wedge q \wedge (\neg r)) \Rightarrow (\mathbf{X}\,r))$,
- $ospc \models \mathbf{G}((q \wedge (\neg r)) \Rightarrow \diamond a)$
- $ospc \models (\mathbf{G}(\mathbf{F}\,\diamond a)) \Rightarrow (\mathbf{G}(\mathbf{F}\,\triangledown a))$

then $ospc \models p \Rightarrow (q\,\mathbf{U}\,r)$.

Proof: Similar to the one presented for Proposition 3.14 taking $k = 0$. ■

¿From Proposition 3.16 easily follows that the generation of explanations for guarantee problems is similar to the case of response problems. The only difference is related to the kind of safety problem we have in (i). As before, this corresponds to *one step explanations* and it is also possible to consider *several step explanations*.

Lets consider now the case of persistence properties.

Proposition 3.17 If

(i) $ospc \models \mathbf{G}(q \Rightarrow (\mathbf{X}\,q))$

(ii) $ospc \models \mathbf{G}(p \Rightarrow (\mathbf{F}\,q))$

then $ospc \models \mathbf{G}(p \Rightarrow (\mathbf{F}(\mathbf{G}\,q)))$.

Proof: Let λ be a model of $ospc$ and $k_1 \in I\!N_0$ such that $\lambda, k_1 \models p$. By (ii) we have that $\lambda, k_2 \models q$ for some $k_2 \geq k_1$. Using (i) and induction in $k \geq k_2$, we easily get that $\lambda, k \models q$ for all $k \geq k_2$. Therefore $\lambda, k_1 \models \mathbf{F}(\mathbf{G}\,q)$. ■

The Proposition 3.17 shows that the generation of explanations for persistence problems can rely on the previous techniques for the generation of explanations for safety and response problems.

4 Specification of object communities

In this section we introduce the syntactic and semantic aspects of temporal object community specifications. An object community consists of a collection of objects which interact with each other through the occurrence of actions: under certain conditions the occurrence of a certain action within an object implies the occurrence of an action within another object. The definitions presented herein are based on [12, 13].

Definition 4.1 An *object community signature* is a pair

$$\Sigma_{com} = (P, Sg)$$

where P is a finite set and Sg is a map assigning to each $ob \in Id_P = \wp(P)\backslash\{\emptyset\}$ a triple $Sg(ob) = (ob, Act_{ob}, Att_{ob})$ such that

- Act_{ob} (the set of actions) and Att_{ob} (the set of attributes) are finite sets;
- $\perp$ (the idle action) is in Act_{ob} whenever ob is a singleton;
- Act_{ob} and Att_{ob} are, respectively, the disjoint union of $Act_{\{i_k\}}$ and the disjoint union of $Att_{\{i_k\}}$, $1 \leq k \leq n$, whenever $ob = \{i_1, \ldots, i_n\}$, $n > 1$. ■

P is a set of identifiers. Each singleton in Id_P is a *prime* object identifier. The others elements of Id_P are *composite* object identifiers. An object community signature introduces signatures for prime and composite objects. We get the signature of each composite object from the signatures of the prime objects involved.

Interaction between prime objects as well as properties related to the joint behaviour of different prime objects have to be expressed in the language of suitable composite objects. Given $ob \in Id_P$, the set of formulae over $Sg(ob)$, F_{ob}, is defined as in the case of a single object (section 2) and we also consider the set of state constraints (SC_{ob}), the set of enabling constraints of type 1 ($EC1_{ob}$), the set of enabling constraints of type 2 ($EC2^a_{ob}$) and the set of fairness requirements (FR_{ob}) as described therein.

We also consider as valuation schema $\mathbf{G}((\triangledown a_1 \wedge \ldots \wedge \triangledown a_n \wedge \alpha^{o_1 \ldots o_m}_{p_1 \ldots p_m}) \Rightarrow (\mathbf{X}\, \alpha))$, where $a_1, \ldots, a_n \in Act_{ob}$, and use $S^{a_1 \ldots a_n}_{ob}$ and S_{ob} in a similar way.

In the sequel, for each composite object identifier $ob = \{i_1, \ldots, i_n\}$, $n > 1$, we assume that $Act_{ob} = \bigcup_{1 \leq k \leq n} \{a_{i_k} : a \in Act_{\{i_k\}}\}$ and $Att_{ob} = \bigcup_{1 \leq k \leq n} \{o_{i_k} : o \in Att_{\{i_k\}}\}$. We consider, for each $1 \leq k \leq n$, the injection map $iAct^{ob}_{\{i_k\}} : Act_{\{i_k\}} \rightarrow Act_{ob}$ such that $iAct^{ob}_{\{i_k\}}(a) = a_{i_k}$ for each $a \in Act_{\{i_k\}}$ and the injection map $iAtt^{ob}_{\{i_k\}} : Att_{\{i_k\}} \rightarrow Att_{ob}$ such that $iAtt^{ob}_{\{i_k\}}(o) = o_{i_k}$ for each $o \in Att_{\{i_k\}}$. We also introduce the *translation map* $\tau^{ob}_{\{i_k\}} : F_{\{i_k\}} \rightarrow F_{ob}$ that extends the injections to the set of formulae.

Interactions between prime objects are described through *interaction formulae* in the language of suitable composite objects. The formula $f = \mathbf{G}(q \Rightarrow (\triangledown a_{i_1} \Rightarrow \triangledown b_{i_2})) \in F_{ob}$, where $ob = \{i_1, i_2\}$, expresses an interaction between the two distinct primes objects identified by $\{i_1\}$ and $\{i_2\}$. The condition of the interaction formula, denoted by $cnf(f)$, is the state constrain $q \in \tau^{ob}_{\{i_1\}}(SC_{\{i_1\}})$. The action $a_{i_1} \in iAct^{ob}_{\{i_1\}}(Act_{\{i_1\}})$, denoted by $tri(f)$, is the trigger action and the action $b_{i_2} \in iAct^{ob}_{\{i_2\}}(Act_{\{i_2\}})$, denoted by $tar(f)$, is the target action. This formula expresses that whenever q holds in the object $\{i_1\}$ and a occurs within $\{i_1\}$ then action b of object $\{i_2\}$ also occurs. The set of all interaction formulae in F_{ob} is denoted by $Intr_{ob}$.

Definition 4.2 An *object community specification* is a triple

$$ocspc = (\Sigma_{com}, \{AxF_{ob}\}_{ob \in Id_P}, \{AxS_{ob}\}_{ob \in Id_P})$$

where

- $\Sigma_{com} = (P, Sg)$ is an object community signature;

- $AxF_{\{i\}} \subseteq SC_{\{i\}} \cup EC1_{\{i\}} \cup EC2_{\{i\}} \cup FR_{\{i\}}$ for each $i \in P$,
- $AxF_{\{i_1,i_2\}} \subseteq Intr_{\{i_1,i_2\}}$ for each two distinct elements of $i_1, i_2 \in P$ and $AxF_{ob} = \emptyset$ for the other non-singular object identifiers;
- $AxS_{\{i\}} = \bigcup_{a \in Act_{\{i\}}} \{s_a\}$ where $s_a \in S^a_{\{i\}}$ and $s_\perp = \mathbf{G}((\triangledown\perp \wedge \alpha) \Rightarrow (\mathbf{X}\,\alpha))$ for each $i \in P$ and $AxS_{ob} = \emptyset$ otherwise. ■

The community behaviour is specified through (i) the specification of the behaviour of its prime objects, using, as before, state constraints, enabling constraints, fairness requirements and valuation schemata and (ii) the specification of interactions between pairs of prime objects using the interaction formulae.

Definition 4.3 Let $\Sigma_{com} = (P, Sg)$ be an object community signature. An *interpretation structure* over Σ_{com} is a map Λ assigning to each $\{i\} \in Id_P$ an interpretation structure $\lambda_{\{i\}}$ over the object signature $(Act_{\{i\}}, Att_{\{i\}})$ in the sense of Definition 2.4 and, to each $ob = \{i_1, \ldots, i_n\} \in Id_P$, where $n > 1$, a map $\lambda_{ob} : I\!N_0 \to \wp(\triangledown Act_{ob} \cup \diamond Act_{ob} \cup Att_{ob})$ such that $\lambda_{ob}(m) = \bigcup_{1 \leq k \leq n} \tau^{ob}_{\{i_k\}}(\lambda_{\{i_k\}}(m))$ for each $m \in I\!N_0$. ■

Given $f \in F_{ob}$, the satisfaction of f by Λ at $k \in I\!N_0$, denoted by $\Lambda, k \models_{ob} f$, is defined as in section 2 using now λ_{ob}. Other related notions, as for instance the notions of model of an object community specification, *ocspc*-valid and *ocspc*-satisfiable, are defined in a similar way.

¿From the above, it is easy to see that the behaviour of composite objects is determined by the behaviour of its prime components taking into account the interactions specified. Within a prime object one and only one action occurs at each instant. Hence in a composite object at each instant as many actions as its components occur.

Example 4.4 The triple $ocspc = ((P, Sg), \{AxF_{ob}\}_{ob \in Id_P}, \{AxS_{ob}\}_{ob \in Id_P})$ where

- $P = \{1, 2\}$
- $Sg(\{1\}) = (\{1\}, \{a, d, \perp\}, \{z, u\})$
- $Sg(\{2\}) = (\{2\}, \{c, e, \perp\}, \{x, w\})$
- $AxF_{\{1\}} = \{z\}$ and $AxS_{\{1\}} = \{\mathbf{G}((\triangledown a \wedge \alpha\,{}^{z}_{\neg z}) \Rightarrow (\mathbf{X}\,\alpha)), \mathbf{G}((\triangledown d \wedge \alpha\,{}^{u}_{\neg u}) \Rightarrow (\mathbf{X}\,\alpha))$ $\mathbf{G}((\triangledown\perp \wedge \alpha) \Rightarrow (\mathbf{X}\,\alpha))\}$
- $AxF_{\{2\}} = \{\neg x\}$ and $AxS_{\{2\}} = \{\mathbf{G}((\triangledown c \wedge \alpha\,{}^{x}_{\mathbf{t}}\,{}^{w}_{\neg w}) \Rightarrow (\mathbf{X}\,\alpha)), \mathbf{G}((\triangledown\perp \wedge \alpha) \Rightarrow (\mathbf{X}\,\alpha)),$ $\mathbf{G}((\triangledown e \wedge \alpha\,{}^{w}_{\mathbf{f}}) \Rightarrow (\mathbf{X}\,\alpha))\}$
- $AxF_{\{1,2\}} = AxS_{\{1,2\}} = \emptyset$

is an object community specification. We have that $ocspc \not\models_{\{1,2\}} \mathbf{G}(z_1 \vee x_2)$. ■

5 Safety problems in object community specifications

The notions of abduction problem, safety, response, guarantee and persistence problems we have introduced in section 3 are easily extended to the case of object community specifications. They can now represent properties related to the behaviour of several interacting objects. The techniques to generate the explanations can also be extended to this case [3].

In what concerns explanations for a given abduction problem f in *ocspc* we are going to consider now a new possibility for constraining more the community behaviour. As before, we can introduce new state constraints and/or new enabling constraints and/or new fairness requirements on the specification of the different prime objects. However, we can also specify new interactions between the objects. Instead of sets, the explanations are now families of sets of formulae. Each set of the family contains the new axioms we want to introduce in each prime or composite object.

In the sequel we assume the object community signature $\Sigma_{com} = (P, Sg)$, where $Sg(ob) = (Act_{ob}, Att_{ob})$ for each $ob \in Id_P = \wp(P) \backslash \{\emptyset\}$, and the object community specification $ocspc = (\Sigma_{com}, \{AxF_{ob}\}_{ob \in Id_P}, \{AxS_{ob}\}_{ob \in Id_P})$.

Definition 5.1 Let $ob \in Id_P$ and $f \in F_{ob}$ an abduction problem in *ocspc*. An *explanation for* f is a family of sets of formulae

$$\mathcal{E} = \{E_{ob}\}_{ob \in Id_P}$$

such that

- $E_{\{i\}} \subseteq SC_{\{i\}} \cup EC1_{\{i\}} \cup EC2_{\{i\}} \cup FR_{\{i\}}$ for each $i \in P$ and $E_{ob} \subseteq Intr_{ob}$ otherwise,
- $ocspc' = (\Sigma_{com}, \{AxF_{ob} \cup E_{ob}\}_{ob \in Id_P}, \{AxS_{ob}\}_{ob \in Id_P})$ is a consistent object community specification,
- $ocspc' \models_{ob} f$. ∎

An explanation introduces, for each object, a (possibly empty) set of new formulae in the language of the object. In the case of prime objects, the set only includes state constraints, enabling conditions or fairness requirements. In the case of composite objects involving two prime objects, the set only includes interaction formulae. The set is empty on the other cases.

In this section we are going to focus our attention on the extension of the techniques presented in section 3 with respect to safety problems to the case of object community specifications. In particular, we describe how explanations for safety problems $f \in F_{ob'}$ in *ocspc* where $ob' = \{i_1, i_2\}$, $i_1 \neq i_2$, can be generated. The generalization for the other cases is straightforward [3]. In the sequel we assume $ob' = \{i_1, i_2\} \in Id_P$ where $i_1 \neq i_2$ and $p, q, r \in SC_{ob'}$.

Proposition 5.2

1. If $ocspc \models_{ob'} \mathbf{G}((\triangledown a_{i_1} \wedge \triangledown b_{i_2} \wedge p) \Rightarrow \mathbf{X}\, q)$ for all $a \in Act_{\{i_1\}}$ and $b \in Act_{\{i_2\}}$ then $ocspc \models_{ob'} \mathbf{G}(p \Rightarrow (\mathbf{X}\, q))$.
2. If $ocspc \models_{ob'} p \Rightarrow (q \vee r)$ and $ocspc \models_{ob'} \mathbf{G}((q \wedge (\neg r)) \Rightarrow (\mathbf{X}(q \vee r)))$ then $ocspc \models_{ob'} p \Rightarrow (q\, \mathbf{W}\, r)$.
3. If $ocspc \models_{ob'} \mathbf{G}(r \Rightarrow (q \vee r))$ and $ocspc \models_{ob'} \mathbf{G}((q \wedge (\neg r)) \Rightarrow (\mathbf{X}(q \vee r)))$ then $ocspc \models_{ob'} \mathbf{G}(p \Rightarrow (q\, \mathbf{W}\, r))$.

Proof: 1. Let Λ be a model of *ocspc*. If $\Lambda, k \models_{ob'} p$, by Definition 4.3, $\Lambda, k \models_{ob'} \nabla a_{i_1}$ and $\Lambda, k \models_{ob'} \nabla b_{i_2}$ for some $a \in Act_{\{i_1\}}$ and $b \in Act_{\{i_2\}}$. Therefore, $\Lambda, k \models_{ob'} \mathbf{X}\, q$.

The proofs of 2. and 3. use arguments similar to the ones in the proof of Proposition 3.4.2. and 3.4.3. ■

The generation of explanations has a lot in common with the previous case of object specifications but one has to take into account that in the composite object $ob' = \{i_1, i_2\}$ two actions occur in each instant (one from each prime object). The generation of explanations for the establishment of initial conditions is rather similar. However, in the case of safety problems $\mathbf{G}(p \Rightarrow (\mathbf{X}\, q))$, we have to consider the establishment of q from p by any set $A \subseteq Act_{ob'}$ containing exactly two actions: one action from the prime object $\{i_1\}$ and another from the prime object $\{i_2\}$.

As before, for constructing explanations for the initial establishment of q, we build an exhausted tableau for a suitable set of propositional formulae in $SC_{ob'}$. This set takes now into account the initial conditions on attribute values specified for the prime objects involved. An explanation is a family of sets of state constraints in the language of the prime objects involved expressing new conditions on the initial values of the attributes of these prime objects. These state constraints are again obtained from closures of tableaux. Since each closure is a subset of $SC_{ob'}$ we have to extract from it, on one hand, the literals related to $\{i_1\}$ and, on the other hand, the literals related to $\{i_2\}$. Those literals are then translated back to the language of $\{i_1\}$ and $\{i_2\}$, respectively. A new state constraint for the prime object $\{i_1\}$ and a new state constraint for the prime object $\{i_2\}$ can be then constructed.

In what concerns explanations for the establishment of q from p by $a \in Act_{\{i_1\}}$ and $b \in Act_{\{i_2\}}$ we again consider an exhausted tableau for a suitable set of propositional formulae. This set takes into account not only the enabling constraints specified for a and b but also the specified interactions between the prime objects involved. Besides new enabling constraints over the actions, we consider also as possible explanations new interactions between the objects. The new enabling constraints will enrich the specification of the prime objects and therefore they are formulae in the language of these prime objects. The conditions of the enabling constraints are obtained, as before, from tableau closures proceeding as described in the previous paragraph. The interaction formulae will enrich the specification of the composite object ob' and so they are formulae in the language of ob'. The conditions of the interaction formulae are again obtained from tableau closures.

Proposition 5.3 Let $a \in Act_{\{i_1\}}$ and $b \in Act_{\{i_2\}}$ and assume that in the object community specification *ocspc* we have that $AxS_{\{i_1\}} \cap S^a_{\{i_1\}} = \{\mathbf{G}((\nabla a \wedge \alpha^{o^a_1 \ldots o^a_{n_a}}_{p^a_1 \ldots p^a_{n_a}}) \Rightarrow (\mathbf{X}\, \alpha))\}$ and that $AxS_{\{i_2\}} \cap S^b_{\{i_2\}} = \{\mathbf{G}((\nabla b \wedge \alpha^{o^b_1 \ldots o^b_{n_b}}_{p^b_1 \ldots p^b_{n_b}}) \Rightarrow (\mathbf{X}\, \alpha))\}$. Let $A = \{a_{i_1}, b_{i_2}\}$, $A_{\downarrow 1} = a$ and $A_{\downarrow 2} = b$.

1. If there is a closed tableau for $G^{\stackrel{i}{\to} q}_{ocspc} = (\bigcup_{1 \leq k \leq 2} \tau^{ob'}_{\{i_k\}}(AxF_{\{i_k\}} \cap SC_{\{i_k\}})) \cup \{\neg q\}$ then q is initially established in *ocspc*.

2. If there is a closed tableau for

$$G^{p \stackrel{A}{\to} q}_{ocspc} = \tau^{ob'}_{\{i_1\}}(AxF_{\{i_1\}} \cap EC^a_{\{i_1\}}) \cup \tau^{ob'}_{\{i_2\}}(AxF_{\{i_2\}} \cap EC^b_{\{i_2\}}) \cup$$
$$\{\neg cnd(f) : f \in Intr_{ob'} \cap AxF_{ob'}, tri(f) = a_{i_1}, tar(f) \neq b_{i_2}\} \cup$$

$$\{\neg cnd(f): f \in Intr_{ob'} \cap AxF_{ob'}, tri(f)=b_{i_2}, tar(f) \neq a_{i_1}\} \cup$$
$$\{p, \neg q^{o^a_1 \ldots o^a_{n_a}\, o^b_1 \ldots o^b_{n_b}}_{p^a_1 \ldots p^a_{n_a}\, p^b_1 \ldots p^b_{n_b}}\}$$

then A establishes q from p in $ospc$.

3. Let Cl be a closure of an exhausted tableau for $G^{\overset{i}{\to} q}_{ocspc}$ and $Cl_{i_k} \subseteq F_{\{i_k\}}$, $k \in \{1,2\}$, such that $Cl = \bigcup_{1 \leq k \leq 2} \tau^{ob'}_{\{i_k\}}(Cl_{i_k})$. Consider the family of sets $\mathcal{E} = \{E_{ob}\}_{ob \in Id_P}$ where
$$E_{\{i_k\}} = \{\bigwedge Cl_{i_k}\}$$
for each $k \in \{1,2\}$ such that $Cl_{i_k} \neq \emptyset$ and $E_{ob} = \emptyset$ for $ob \in Id_P \backslash \{\{i_1\},\{i_2\}\}$ or $ob = \{i_k\}$ with $k \in \{1,2\}$ such that $Cl_{i_k} = \emptyset$. We have that q is initially established in $ocspc' = (\Sigma_{com}, \{AxF_{ob} \cup E_{ob}\}_{ob \in Id_P}, \{AxS_{ob}\}_{ob \in Id_P})$.

4. Let Cl be a closure of an exhausted tableau for $G^{p \overset{A}{\to} q}_{ocspc}$ and $Cl_{i_k} \subseteq F_{\{i_k\}}$, $k \in \{1,2\}$, such that $Cl = \bigcup_{1 \leq k \leq 2} \tau^{ob'}_{\{i_k\}}(Cl_{i_k})$.

   (i) Consider the family of sets $\mathcal{E} = \{E_{ob}\}_{ob \in Id_P}$ where
   $$E_{\{i_k\}} = \{\mathbf{G}(\diamond A_{\downarrow k} \Rightarrow \bigwedge Cl_{i_k})\}$$
   for each $k \in \{1,2\}$ such that $Cl_{i_k} \neq \emptyset$ and $E_{ob} = \emptyset$ for $ob \in Id_P \backslash \{\{i_1\},\{i_2\}\}$ or $ob = \{i_k\}$ with $k \in \{1,2\}$ such that $Cl_{i_k} = \emptyset$. We have that A establishes q from p in $ocspc' = (\Sigma_{com}, \{AxF_{ob} \cup E_{ob}\}_{ob \in Id_P}, \{AxS_{ob}\}_{ob \in Id_P})$.

   (ii) Consider the family of sets $\mathcal{E} = \{E_{ob}\}_{ob \in Id_P}$ where $E_{ob} = \emptyset$ for each $ob \neq ob'$ and
   $$E_{ob'} = I_1 \cup I_2$$
   such that, for each $k \in \{1,2\}$, if $Cl_{i_k} = \emptyset$ then $I_k = \emptyset$ and, otherwise,
   $$I_1 = \{\mathbf{G}((\neg(\bigwedge \tau^{ob'}_{\{i_1\}}(Cl_{i_1}))) \Rightarrow (\triangledown a_{i_1} \Rightarrow \triangledown b'_{i_2}))\}$$
   where $b'_{i_2} \in iAct^{ob'}_{\{i_2\}}(Act_{\{i_2\}} \backslash \{b, \perp\})$ and
   $$I_2 = \{\mathbf{G}((\neg(\bigwedge \tau^{ob'}_{\{i_2\}}(Cl_{i_2}))) \Rightarrow (\triangledown b_{i_2} \Rightarrow \triangledown a'_{i_1}))\}$$
   where $a'_{i_1} \in iAct^{ob'}_{\{i_1\}}(Act_{\{i_1\}} \backslash \{a, \perp\})$. We have that A establishes q from p in $ocspc' = (\Sigma_{com}, \{AxF_{ob} \cup E_{ob}\}_{ob \in Id_P}, \{AxS_{ob}\}_{ob \in Id_P})$.

Proof: 1. The proof is similar to the proof of Proposition 3.8.1.

2. Let Λ be a model of $ocspc$ and $k \in \mathbb{N}_0$ such that $\Lambda, k \models_{ob'} \triangledown a_{i_1} \wedge \triangledown b_{i_2} \wedge p$. The proof is similar to the proof of Proposition 3.8.2. The only difference is that we have to assure that the sets related to the interactions are also satisfied at k. Consider the set $\{\neg cnd(f) : f \in Intr_{ob'} \cap AxF_{ob'}, tri(f) = a_{i_1}, tar(f) \neq b_{i_2}\}$ (the other is similar) and a formula $\neg cnd(f)$ in it. Since $\Lambda, k \models_{ob'} \triangledown a_{i_1}$, if $\Lambda, k \models_{ob'} cnd(f)$ then $\Lambda, k \models_{ob'} \triangledown tar(f)$ because $\Lambda \models_{ob'} AxF_{ob'}$. Thus $\Lambda, k \models_{ob'} \triangledown tar(f) \wedge \triangledown b_{i_2}$. This contradicts Definition 4.3 since $tar(f)$ and b_{i_2} are different actions from prime object $\{i_2\}$. Therefore, $\Lambda, k \models_{ob'} \neg cnd(f)$.

3. Taking into account $ocspc'$ it follows that the set $G^{p \xrightarrow{A} q}_{ocspc'}$ is equal to the set $G^{p \xrightarrow{A} q}_{ocspc} \cup \{\bigwedge \tau^{ob'}_{\{i_1\}}(Cl_{i_1}), \bigwedge \tau^{ob'}_{\{i_2\}}(Cl_{i_2})\}$ The proof follows the steps of the proof of Proposition 3.8.3.

4.(i) Taking into account $ocspc'$ it follows that the set $G^{p \xrightarrow{A} q}_{ocspc'}$ is equal to the set $G^{p \xrightarrow{A} q}_{ocspc} \cup \{\bigwedge \tau^{ob'}_{\{i_1\}}(Cl_{i_1}), \bigwedge \tau^{ob'}_{\{i_2\}}(Cl_{i_2})\}$. The proof follows arguments similar to the ones in the proof of Proposition 3.8.4.

4.(ii) Taking into account $ocspc'$ it follows that the set $G^{p \xrightarrow{A} q}_{ocspc'}$ is equal to the set $G^{p \xrightarrow{A} q}_{ocspc} \cup \{\neg(\neg(\bigwedge \tau^{ob'}_{\{i_1\}}(Cl_{i_1}))), \neg(\neg(\bigwedge \tau^{ob'}_{\{i_2\}}(Cl_{i_2})))\}$ which is for our purpose the same of having $G^{p \xrightarrow{A} q}_{ocspc} \cup \{\bigwedge(\tau^{ob'}_{\{i_1\}}(Cl_{i_1})), \bigwedge(\tau^{ob'}_{\{i_2\}}(Cl_{i_2}))\}$. The proof follows arguments similar to the ones in the proof of Proposition 3.8.4. ■

Example 5.4 Considering the object community specification *ocspc* introduced in Example 4.4 we have that the formula $\mathbf{G}(z_1 \vee x_2) \in F_{\{1,2\}}$ is a safety problem in *ocspc*. In *ocspc* we have that: (i) $z_1 \vee x_2$ is initially established; (ii) $A_1 = \{a_1, \perp_2\}$ and $A_2 = \{a_1, e_2\}$ do not establish $z_1 \vee x_2$ from $z_1 \vee x_2$.

1.1. The set $Cl = \{\neg z_1\}$ is a closure for any exhausted tableau for $G^{z_1 \vee x_2 \xrightarrow{A_1} z_1 \vee x_2}_{ocspc}$. This means that the family $\mathcal{E} = \{E_{ob}\}_{ob \in Id_P}$ where $E_{\{1\}} = \{\mathbf{G}(\diamond a \Rightarrow (\neg z))\}$ and $E_{ob} = \emptyset$ for each $ob \in \{\{2\}, \{1,2\}\}$ is an explanation for the establishment of $z_1 \vee x_2$ from $z_1 \vee x_2$ by A_1.

1.2. The set $Cl = \{\neg z_1\}$ is also closure for any exhausted tableau for $G^{z_1 \vee x_2 \xrightarrow{A_2} z_1 \vee x_2}_{ocspc}$ which means that above the family $\mathcal{E}$ is also an explanation for the establishment of $z_1 \vee x_2$ from $z_1 \vee x_2$ by A_2.

1.3. From **1.1.** and **1.2.** we can conclude that $\mathcal{E}$ is an explanation for the safety problem $\mathbf{G}(z_1 \vee x_2)$. In this explanation we constrain the occurrence of a in the prime object $\{1\}$ preventing its occurrence whenever we have z. In fact, if a only occurs whenever we have $\neg z$, since we also have $z_1 \vee x_2$, we would necessarily have x in prime object $\{2\}$. But x is not changed when either $\perp$ or e occur in the prime object $\{2\}$. Thus in the next state $z_1 \vee x_2$ again holds.

2. Considering the set Cl in **1.1.** we have that the family $\mathcal{E}' = \{E'_{ob}\}_{ob \in Id_P}$ where $E'_{\{1,2\}} = \{\mathbf{G}(z_1 \Rightarrow (\triangledown a_1 \Rightarrow \triangledown c_2))\}$ and $E'_{ob} = \emptyset$ for each $ob \in \{\{1\}, \{2\}\}$ is an explanation both for the establishment of $z_1 \vee x_2$ from $z_1 \vee x_2$ by A_1 and for the establishment of $z_1 \vee x_2$ from $z_1 \vee x_2$ by A_2. Therefore $\mathcal{E}'$ is another explanation for the safety problem $\mathbf{G}(z_1 \vee x_2)$. This explanation means that we can let a occur whenever we have z. This could bring some problems since after its occurrence we may not have again $z_1 \vee x_2$ since z would become false. However, if we assure that c occurs at the same time in prime object $\{2\}$, we have no problems since c makes x true. Since only one action can occur within a prime object, whenever a occurs and z holds neither $\perp$ nor e can occur within $\{2\}$.

■

Note that herein it is not possible to obtain an explanation from several closures at the same time taking, as before, the disjunction of the conjunction of the literals in each closure.

Preference criteria between explanations are similar to ones described in subsection 3.2. We give preference to those explanations that allow as much as possible the occurrence of the actions and to those that as much related as possible to the problem at hand.

The assertions of Proposition 3.11 can be extended in straightforward way to the present case having in mind that now an explanation is a family. Consistency of resulting object community specifications can also be checked as before with temporal tableaux.

6 Conclusions and future work

Traditionally, in the context of object specification, the certification task consists in proving that some envisaged property follows from the specification. However, in general, whenever the property can not be inferred from the specification and one can not just add it as a new axiom, no help on how the specification could be enriched is provided. Finding, in these situations, new suitable axioms is an important certification approach that has not been explored in the context of temporal object specification. This task has an abductive flavour and in this paper we addressed the problem of certification from that abductive point of view: the properties that are not satisfied are seen as *abduction problems* and the new axioms we want to obtain are seen as *explanations* for these problems. We introduced some techniques that allow the generation of new suitable axioms in such a way that the intended property is inferred from the enriched specification. These techniques combine ideas from temporal logic proof techniques [7, 8] and from tableau based techniques for abduction [9].

We considered object specifications as including a signature, a set of temporal formulae and a set of valuation schemata [12, 13]. The signature introduces the action symbols and the attribute symbols. The formulae restrict the object behaviour by imposing conditions on the initial value of the attributes (through state constraints) and by imposing conditions on the occurrence and enabling of actions (through enabling conditions and fairness requirements). The valuation schemata express attribute updates. Herein, we have focused our attention on the propositional fragment.

Following [7, 8], where a classification of relevant properties in the context of temporal specification of reactive systems is presented, we have concentrated our efforts on the generation of explanations for safety, response, guarantee and persistence problems. In the case of safety problems, those explanations consist of new state constraints and sufficient conditions on the enabling of actions. They are obtained from closures of tableaux for suitable sets of propositional formulae taken from the specification. The explanations for the other kinds of problems take into account not only explanations for certain safety problems but also necessary conditions on the enabling of actions as well as fairness requirements.

We have also discuss some preference criteria among the several explanations that can be generated. Our main concern was to give preference to those explanations that, on one hand, did not restrict the object behaviour more than necessary and, on the other hand, were as related as possible to the problem at hand.

Finally, we described how to extend these techniques to case of object community specifications (i.e. collections of interacting objects). An object community specification includes signatures for prime objects. ¿From these signatures we can obtain signatures for the composite objects. The axioms and axiom schemata for prime objects are as before. The only possible axioms of composite objects are interaction formulae expressing interactions between different prime objects. Abduction problems can now express properties of the joint behaviour of different prime objects. In this paper we focused on the case of safety problems. The explanations are now families of set of formulae. These formulae can be, as before, new state constraints or enabling conditions for prime objects but they can also be new interaction formulae.

There is in the literature some work about abductive reasoning in the context of event calculus as, for instance, [2]. Within this context one also deals with occurrences of events which can lead to state modifications. However, this work is related to planning tasks in which the goal is to abduce a certain order of event occurrences through which one can reach an intended final state from a certain initial state.

As future work we intend to address several topics. We intend to study the possibility of establishing certain conditions on the given specifications and on the explanations that assure the consistency of the enriched specifications. This would avoid, in some cases, the final step of consistency checking. Another issue is related to the study of other preference criteria and to the improvement of the explanation generation methods in order to prevent the generation of (more) uninteresting explanations according to our criteria. We also intend to extend this work to the case of communities of objects with classes and inheritance. Another very important issue is the extension of the described techniques to first order temporal logic.

References

[1] J. Bell and M. Machover. *A Course in Mathematical Logic.* North-Holland, 1977.

[2] K. Eshghi. Abductive planning with event calculus. In *Proceedings of the 5th ICLP*, 1988.

[3] P. Gouveia. *Abductive reasoning over temporal specifications of objects.* PhD thesis, Instituto Superior Técnico, 1998. (in Portuguese).

[4] P. Gouveia and C. Sernadas. Introducing explanations in temporal object specification. In R. Wieringa and R. Feenstra, editors, *IS-CORE'94 - Selected Papers*, pages 53–68. World Scientific Publishers, 1995.

[5] Y. Kesten and Z. Manna and H. McGuire and A. Pnueli. A decision algorithm for full propositional temporal logic. In *LNCS 697*, pages 97–109. Springer Verlag, 1993.

[6] Z. Manna and A. Pnueli. The anchored version of temporal framework. In J. de Baker and W-P. de Roever and G. Rozenberg, editors, *Linear Time, Branching and Partial Order in Logics and Models for Concurrency, LNCS 354*, pages 201–284. Springer Verlag, 1989.

[7] Z. Manna and A. Pnueli. Completing the temporal picture. *Theoretical Computer Science*, 83:97–130, 1991.

[8] Z. Manna and A. Pnueli. Models for reactivity. *Acta Informatica*, 30(7):609–673, 1993.

[9] M. Mayer and F. Pirri. First order abduction via tableau and sequent calculi. In *Bulletin of the IGPL 1 (1)*, pages 99–117. 1993.

[10] G. Paul. Approaches to abductive reasoning: an overview. *Artificial Intelligence Review*, 7:109–152, 1993.

[11] C. Peirce. Abduction and induction. In Buchler, editor, *Philosophical Writings of Peirce*, pages 150–156.Dover, 1955.

[12] A. Sernadas, C. Sernadas, and J. Costa. Object specification logic. *Journal of Logic and Computation*, 5(5):603–630, 1995.

[13] A. Sernadas, C. Sernadas, and J. Ramos. A temporal logic approach to object certification. *Data & Knowledge Engineering*, 19:267–294, 1996.