Causal explanations for reactive real-time systems

Thomas Mari

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique
Spécialité : Informatique
Unité de recherche : Laboratoire d'Informatique de Grenoble

Explications causales pour les systèmes temps réel réactifs

Causal explanations for reactive real-time systems

Présentée par :

Thomas MARI

Direction de thèse :

Gregor GOESSLER	Directeur de thèse
Thao DANG Directeur de recherche, CNRS Délégation Alpes	Co-directrice de thèse

Rapporteurs :

Stefan LEUE
PROFESSEUR, University of Konstanz
Dejan NIKOVIC
SENIOR SCIENTIST, Austrian Institute of Technology

Thèse soutenue publiquement le **17 novembre 2023**, devant le jury composé de :

Gregor GOESSLER DIRECTEUR DE RECHERCHE, Centre Inria de l'Université Grenoble Alpes	Directeur de thèse
Thao DANG DIRECTRICE DE RECHERCHE, Verimag	Directrice de thèse
Patricia BOUYER-DECITRE DIRECTRICE DE RECHERCHE, Laboratoire Méthodes Formelles	Examinatrice
Eric GAUSSIER PROFESSEUR DES UNIVERSITES, University Grenoble Alps	Examinateur
Oleg SOKOLSKY ASSOCIATE PROFESSOR, University of Pennsylvania	Examinateur

Causal Explanations for Reactive Real-time Systems

Thomas MARI

last updated: February 15, 2024

Remerciements

Je tiens tout d'abord à remercier mes encadrants de thèse Gregor Goessler et Thao Dang pour leur aide et leur investissement durant mes quatre ans de doctorat. Grâce à mes encadrants, j'ai eu une grande liberté pour explorer mon sujet et choisir les directions de recherche. Néanmoins, il arrive d'arriver dans des impasses ou de trouver des contre-exemples aux théorèmes que l'on veut prouver. Dans ces moments-là, j'ai eu la chance d'avoir des réunions régulières avec mes encadrants qui me donnaient de l'inspiration, ainsi que de la rigueur dans mon travail. Je sortais de ces réunions plus motivé et grâce à cela, je n'ai pas vu le temps passé pendant ma thèse.

Je tiens à remercier mon jury de thèse et en particulier mes rapporteurs Stefan Leue et Dejan Nikovic qui ont accepté d'évaluer mon manuscrit, ainsi que mes examinateurs Patricia Bouyer-Decitre, Eric Gaussier, et Oleg Sokolsky. Mon jour de soutenance que j'ai redouté s'est avéré être un très bon jour, en partie grâce à mon jury qui a été très chaleureux et convivial. Une des raisons qui m'a motivée à soutenir était de présenter mon travail à des experts internationaux. Même dans un laboratoire de recherche, il est difficile de trouver des interlocuteurs, en dehors de ces encadrants, qui comprennent ce que l'on fait, et de ce fait, j'étais ravi de faire connaitre mon travail à mon jury de thèse.

J'ai toujours aimé le côté social du travail en recherche et j'ai eu la chance d'être accueilli dans l'équipe SPADES et le laboratoire VERIMAG. Pendant ces années, j'ai rencontré des gens passionnants et chaleureux. Je remercie en particulier les doctorants et post-doctorants, particulièrement Ana, Aina, Hadi, Alexandre, Léo, Baptiste, Akshay, Martin V, Maxime, Giovanni, Pietro, Ludmilla, Hugh (Jiajie) et Marco qui ont permis que mon doctorat soit festif, ludique, sportif et riche en expériences humaines et intellectuelles. Merci à mes cobureaux de longue date Aina, Hadi, Aurélie et Martin qui m'ont aidé à de nombreuses reprises pour des problèmes de compilation ou de rendu Latex.

Je remercie ma famille pour leur soutien, ainsi que de m'avoir permis d'avoir le goût des sciences et des valeurs qui m'ont permis d'arriver là.

Merci à tous les gens que j'ai oubliés, ce doctorat a été une expérience heureuse.

Contents

Chapter 1

Introduction

1.1 Context

In our daily lives, we rely more and more on embedded systems and the services they provide. By embedded system, we mean a system with computer hardware and software that is designed to provide a specific service. Examples of such systems are medical devices such as a pacemaker, household appliances such as automatic coffee machines, and transportation systems such as cruise control systems... Embedded systems can be complex: they are often multi-component systems in which components interact together and the resulting state space can be large. Moreover, the dynamics in real systems contribute to the overall complexity. Such systems can also be heterogeneous by mixing continuous and discrete dynamics. We can think of a computer system (with discrete dynamics) that interacts with a physical system (with continuous dynamics) such as an anti-lock braking system in a car. A failure happened in the system, such as a low heart rate for a pacemaker, it can be hard to understand what happened concretely and why it happened. Understanding the causes of the failure is crucial to correct or repair the system. Our goal is to provide explanations to engineers of why a behavior of the system violates a given specification.

Intuitively, we consider the system to be operating within an environment. In this context, we want to blame the decisions of the system that lead to the failure and not the actions of the environment. In this thesis, we adopt a model-based approach. There are two main advantages of having a model of the system. During the design phase, engineers can perform verification, testing, and debugging directly on the model. Such tasks are necessary in order to build a trusted system. Moreover, in terms of explanations, some analysis cannot be done on a model-free system. With the Ladder of Causation [32], Pearl classifies questions that can be answered based on how much causal information we have on the system. The first layer is Association. We only have data, and we can only infer association between facts. The second layer is Intervention. On top of observing the system, we can make an intervention, such as a controlled experiment, and infer the effects of actions. In this layer, we cannot know if the actions are the causes of the effects, and we cannot predict the effects. With the third and top layer called Counterfactuals, we can explain why an action has a specific effect. Moreover, we can do retrospection and infer

the effects of the alternatives of a past action. In [32], Pearl claims that a model of the system is needed to do counterfactual reasoning and that interventions and observations are not sufficient:

> *we cannot re-run an experiment on subjects who were treated with a drug and see how they behave had they not given the drug.* [32]

An embedded system has services to provide, and while doing that, it must satisfy expected properties. Some systems are critical, such as a pacemaker, in which the satisfaction of expected safety properties is crucial. A safety property is a property which, when violated in an execution, remains violated in every continuation of the execution. We illustrate critical systems with medical devices: In recent years, the treatment of diabetes with insulin injection has gone toward the use of more autonomous devices. In 2006, the authors of [35] presented the goal to develop an autonomous insulin delivery system, that senses glucose level and maintain normal blood glucose level via insulin delivery. Their motivation is to improve glycemic control and increase the quality of life of patients. In 2010, the authors of [8] compare two treatments: (manual) injection therapy and sensor-augmented pump therapy, with a (semi-autonomous) device that injects insulin on command. The latter treatment is semi-autonomous because the pump does not make decisions. In 2017, the authors of [37] use statistical analysis, on a medical device called a closed-loop insulin delivery system in which a

> *control algorithm autonomously increases and decreases subcutaneous insulin delivery on the basis of real-time sensor glucose concentrations.* [37]

That system makes autonomous decisions (insulin delivery) according to an environment (the patient's metabolism) actions and state. The closed-loop insulin delivery system must satisfy properties, it is evaluated and compared to another manual treatment according to the proportion of time spent in a given target glucose range. That critical system is expected to satisfy properties about hypoglycemic control [35]. In order to be used in practice, the patients should be able to trust and understand the behaviors of the autonomous devices. We argue that the use of autonomous devices to provide a service makes the users more dependent and therefore requires trust. In the study of the (semi-autonomous) insulin-pump, patients receive training [8], whereas in [37] (autonomous device) there is no mention of training.

From a cognitive perspective, explanations can be used to increase the understanding of an autonomous system. Given an observed abnormal behavior of a system that can be undesired or unexpected, an explanation would answer why it produces such behaviors. To back up this claim, we refer to Miller:

> *the primary function of explanation is to facilitate learning.* [30]

By *learning* we consider improving the understanding of the behavior of the system. In our context, we want to understand the behavior a posteriori, based on a log of the actions taken during a system execution. A good knowledge and understanding of a system is crucial in both design and maintenance. Furthermore, Miller argues in [30] that explaining decisions of autonomous systems to people increases their trust in those autonomous systems. Motivated by the goal of facilitating the making of trusted embedded systems we tackle the following problem.

Why explanations and not a counterexample? A counter-example contains parts that are not causally relevant, for example, the sequential portions of a faulty execution that are common to all executions or the potential cycles that only delay the violation. Even a minimal execution that violates some property can be too long to be readable by a human operator. Therefore, a counter-example is not always an explanation that facilitates the understanding of the violation of the safety property. The complexity of the system obfuscates the cause and effect mechanisms between the system's decisions and its overall behavior. Explaining an unsafe system in the context of an observed violation allows the user to focus on the specific mechanisms that took place in the observed behavior. Furthermore, explanations facilitate learning about the system at fault and a better understanding of the system is valuable for the developers for designing and maintaining reactive systems.

1.2 Problem Statement

Problem In this thesis, we tackle the problem of calculating an explanation for system failures. Given a model of the system, a safety property, and a log of an unsafe observed execution, compute automatically an explanation that answers why the observed execution violates the safety property.

We now present in more detail the inputs and output of the problem:

Input 1: a model More formally, the embedded systems we address are reactive systems:

> *Reactive systems [...] are repeatedly prompted by the outside world and their role is to continuously respond to external inputs.* [21]

The behavior of the system depends on both the actions of the outside world that we call environment, and the actions of the system that we call decisions. Reactive systems are dynamic, the states and properties depend on the time, and time can be continuous or discrete. The transitions between states are triggered by events, and the set of events is partitioned into controllable events denoting decisions of the system, and uncontrollable events denoting the action of the environment. We also only address input-deterministic systems, from one state of the system, one event leads to one state.

Input 2: a safety property Intuitively, a safety property states that something bad does not happen. We are interested in the analysis of finite behaviors that can be classified as safe or unsafe with respect to a safety property. If a safety property is not satisfied during a system execution, there is a bad finite prefix: the execution remains unsafe for every continuation. That finite prefix is sufficient to construct an explanation because every continuation of that prefix is still a bad prefix and the additional decisions of the system are not causally relevant.

Examples of safety properties on reactive systems are invariance properties:

- The patient's glucose level remains in a given range.
- There is always at most 1 second in between two heartbeats.

Input 3: an observation We want to explain a single observed unsafe behavior. As a matter of fact, different executions may violate the safety property for different reasons, and, therefore, would require different explanations. Moreover, in a real system, not all decisions or actions of the environment are observable. The third input to our problem is the projection of an execution onto the observable events. An explanation should reveal the causally relevant parts of the observed behaviors whether they are observable or unobservable.

Output: a causal explanation We choose to work on causal explanations. There is no consensus on what is an explanation. Halpern and Pearl wrote:

> *getting a good definition of explanation is a notoriously difficult problem, which has been studied for years.* [19]

The same authors propose a definition of a causal explanation for systems modeled by structural equations and with respect to actual causality. They introduce their contribution as follows:

> *The basic idea is that an explanation is a fact that is not known for certain but, if found to be true, would constitute an actual cause of the explanandum (the fact to be explained), regardless of the agent's initial uncertainty.* [19]

A causal explanation is not only a cause but also context in which the cause satisfies their definition of a cause [19]. In that sense, we aim to define a cause for safety property violation. A causal explanation is obtained by extracting and assembling the causes for some behaviors consistent with the explanation and provides a context in which those causes are relevant. A causal explanation is however more informative than the behavior it explains because not all parts of the behavior are causally relevant. We do not use the definition of cause in [19], and we define a quantitative notion of cause that we call contributory cause. Each contributory cause can occur at a different time in the observed violation and denotes the respective contribution to the violation of the safety property. Together, the contributory causes entail the observed violation of the safety property. Because reactive systems are dynamic, we want the explanations to be also dynamic by displaying how causes accumulate and contribute to the failure over time in order to provide the relevant information about the decisions that aggravated the state of the system and led finally to the violation.

1.3 Contributions

In this thesis, we formalize requirements for the explanations of observed safety property violations in reactive real-time systems and propose constructions of causal explanations that satisfy those requirements. While causal explanations such as [19] are often defined on static models such as structural equations, we compute explanations on dynamic models with discrete and continuous time. The supported models are discrete event systems and timed automata.

Contributions:

- Towards the goal of constructing explainable real-time systems, we propose a novel approach to computing explanations. We define and use a **robustness of choice function** that returns a robustness value for each state of the system. We define a contributory cause as a transition that decreases robustness and leads the system closer to the violation of the safety property. Our causal explanations are constructed with a robustness function. Classically, model-based diagnosis of discrete event systems aims at detecting failures and isolating faulty event occurrences based on normal/abnormal behavioral models of the system. With our approach, there is no need to model the failures or faulty events, the semantic model and the safety property are sufficient for explaining the failures.

- We provide a formal definition of causal explanations on discrete event systems, called **choice explanations**, constructed from a discrete robustness of choice function, called **level of choice**. We propose a symbolic approach to effectively construct explanations. Furthermore, we show the feasibility of our approach with a case study.

- We propose a framework where the explanations are equipped with a semantic function that returns, given an explanation, the set of executions that are explained by that explanation. The semantic of a causal explanation is the set of executions that contains the same causes encoded in the explanation, and it is what means the explanation. With a semantic function, an explanation is not only a syntactical object but a generalization of the observed violation. Intuitively, the semantics of explanations allow us to discriminate different explanations based on what behaviors they explain. We propose a set of requirements on the semantics of explanations with respect to the log and the safety property. Finally, we show that choice explanations can be cast in this new framework and satisfy the requirements.

- We extend the definition of choice explanations on dense-time models, based on the well-studied formalism of timed automata. We also propose a symbolic approach to effectively construct explanations for such models, and we illustrate our approach with a case study.

- We formalize a set of requirements for robustness of choice functions in view of explaining the failures of real-time systems. Those requirements are inspired by the properties of the discrete level of choice and are stronger in a dense-time model. The additional requirements for a dense-time model allow the explanations to convey the concept of urgency. Indeed, when letting time pass leads to bad states, and those bad states can be avoided via a discrete transition, delays reduce robustness continuously to signify the urgency of taking a discrete transition.

- Finally, we define an instance of robustness of choice function on real-time systems, and we prove it satisfies the requirements on robustness of choice of real-time systems.

1.4 Outline

In Chapter 2, we review the state of the art and compare it with our approaches to explanations and robustness.

In Chapter 3, we define the models and the concepts that are used in the thesis.

In Chapter 4 we tackle the problem of causal explanation of safety violation in discrete event systems. A preliminary version of the contributions in Chapter4 has been published in:

- Gregor Gössler, Thomas Mari, Yannick Pencolé, and Louise Travé-Massuyès. Towards Causal Explanations of Property Violations in Discrete Event Systems

Furthermore, we give a formal semantics of such explanations, and define a set of required properties between an explanation and its semantics, and prove that the choice explanations satisfy those requirements.

In Chapter 5 we tackle the problem of causal explanation of real-time system with the use of abstractions. We use the level of choice defined in 4 paired with strong-time abstract bisimulation. We adapt choice explanations in real-time system and construct explanations on a case study. The contributions in that Chapter 5 have been published in:

- Thomas Mari, Thao Dang, and Gregor Gössler. Explaining Safety Violations in Real-Time Systems. In Catalin Dima and Mahsa Shirmohammadi, editors, *Formal Modeling and Analysis of Timed Systems*, Lecture Notes in Computer Science, pages 100–116. Springer International Publishing, 2021

In Chapter 6 we formalize robustness of choice functions that are used to compute causal explanations on real-time systems.

Chapter 7 summarizes and concludes this thesis, we discuss the contributions and their limitations, and propose new perspectives.

1.5 Publications

Published

- Gregor Gössler, Thomas Mari, Yannick Pencolé, and Louise Travé-Massuyès. Towards Causal Explanations of Property Violations in Discrete Event Systems
- Thomas Mari, Thao Dang, and Gregor Gössler. Explaining Safety Violations in Real-Time Systems. In Catalin Dima and Mahsa Shirmohammadi, editors, *Formal Modeling and Analysis of Timed Systems*, Lecture Notes in Computer Science, pages 100–116. Springer International Publishing, 2021

In preparation We are working on a journal paper containing the contributions in Chapter 4, and on a conference paper containing the contributions in Chapter 6.

1.6 Summary in English

With the growing complexity of embedded systems, it is crucial to understand and localize sources of failures, for instance, to correct and improve the system. In this thesis, we are interested in counterfactual analysis, more precisely the problem of computing causal explanations. In order to do counterfactual analysis, one needs a model and to this end, we use discrete-time and dense-time models.

The first contribution of the thesis is a formal definition of causal explanations on discrete event systems, that we call choice explanations, constructed from a discrete function, that we call level of choice. We propose a symbolic approach to effectively construct such explanations. Furthermore, we illustrate our approach with several examples and a case study. We also define a semantic function paired with a choice explanation. This semantic function returns the set of behaviors explained by a choice explanation. We identify a set of properties between an explanation and the observed failure and we prove the choice explanations satisfy those properties. The second contribution is the extension of the definition of choice explanations to dense-time models, based on the well-studied formalism of timed automata and the strong time abstract bisimulation. The approach is also illustrated in several examples and a case study.

Regarding real-time systems, a major contribution of the thesis is a novel approach to construct causal explanations of failures from a quantitative function that we call robustness of choice or robustness for short. With our approach, there is no need to model the failures or faulty events, a semantic model and a safety specification are sufficient for explaining the failures. This robustness function assigns a value to every state of the system that reflects the ability of the system to avoid failure. We formalize a set of requirements for robustness functions in view of explaining the failures of a cyber-physical system. Some of these requirements are inspired by the properties of the discrete level of choice and are stronger in a dense-time model. The other additional requirements in a dense-time model allow to convey the concept of urgency in the explanation. Another important contribution is the definition of one instance of robustness function that can be computed for timed automata. With such an instance of robustness function, we can indeed compute causal explanations of the failures of real-time systems.

1.7 Résumé en francais

Avec la complexité croissante des systèmes embarqués, expliquer de manière concise les défaillances des systèmes est crucial pour comprendre ce qu'il s'est passé. Dans cette thèse, nous construisons des explications causales dont la formalisation est basée sur l'analyse contrefactuelle. Pour cela, nous avons besoin d'un modèle du système et nous considérons des modèles à temps discret et temps dense.

Parmi les contributions de cette thèse, nous proposons une nouvelle approche pour calculer des explications causales qui sont basées sur l'utilisation de fonctions de robustesse quantitative. Ces fonctions de robustesse renvoient pour chaque état du système une valeur de robustesse qui évalue la capacité du système à éviter une défaillance. Cette fonction nous permet d'identifier

et de mettre en exergue les éléments pertinents d'une exécution fautive, ainsi que de faire abstraction des éléments non pertinents. Les fonctions de robustesse sont définies par un ensemble de propriétés attendues dans le but d'expliquer la défaillance observée.

Pour les modèles à temps discret, nous proposons une approche symbolique pour construire des explications causales, appelées explications de choix. Ces explications sont basées sur une instance de fonction de robustesse appelé le niveau de choix.

Une autre contribution de cette thèse est d'appairer les explications causales pour des modèles à temps discretet avec une fonction sémantique qui retourne, pour une explication donnée, l'ensemble des comportements du système expliqué par cette explication. Grâce à cette fonction sémantique, nous formalisons un ensemble de propriétés attendues qui atteste de la qualité d'une explication par rapport à la défaillance observée. Nous proposons une approche symbolique pour calculer des explications causales et nous prouvons que les explications produites satisfont les propriétés attendues.

Chapter 2

State of the art

In this chapter, we situate our work in the state of the art of causal explanation and related problems.

Choice Explanations Our definition of effective choice transition for discrete event systems is based on [23], which leverages game theory to explain counterexample traces from model-checking by splitting the error trace into fated and free segments. In the free segments, the system could have avoided the violation of an expected property no matter how the environment behaves. However, our approach with choice explanations goes further because not all actions in the free segments are equally relevant. Our definition of choice explanations disregards events that are "compensated" by other events later in the trace, hence yielding a more concise explanation. A similar idea of highlighting choice states in which the execution could have taken a different outcome is followed in [4], the counter-example analysis focuses on the states in a counter-example from which there is a correct transition that avoids definitely the bug. Other faults that contribute to the system failure but are not in the neighborhood of the origin of a correct transition are discarded from the explanation. In this thesis, we consider reactive systems, for which correct transitions may not exist, for example when a bad state is always reachable. In contrast to our approach, [23, 4] assume full observability.

Counterfactual causation and dynamic system Counterfactual causation has been studied in many disciplines as a precise assessment of individual causes that contribute to an effect. The influential definition of *actual causality* on a structural equations model (SEM) [20] has subsequently been adapted to enable reasoning about system dynamics.

Causal explanations have been proposed to identify the causes of property violations in various discrete-time frameworks, such as LTL specifications [6], event order logic (EOL) [27], safety properties in programs [11], and more recently for reactive systems [10]. Like the latter, our approach distinguishes the system and its environment. Most of these approaches which are based on variants of actual causality [20].

In [6], a cause for the violation of the LTL property ϕ is a pair of state and variable such that switching the value of the variable in the traces changes the satisfaction of ϕ. In [10], a

cause to an effect property is a property over the input sequence. For a sequence of input that does not satisfy the cause, there is some counterfactual trace under contingency that does not satisfy the effect property (PC2), and the contingency is defined similarly to [6], by switching the outputs of the trace. In [10] a counterfactual trace under contingency is not always a trace of the model. In this thesis, we only take account of executions of the systems in our counterfactual analysis that is the computation of the robustness of choice. Once this function is computed, we can compute explanations of different violations of the safety property without additional counterfactual analysis. In [27], a cause for the violation of an LTL formula is an EOL formula that encodes the occurrences and the order of events. The paper [5] uses SAT-Solving techniques to compute actual causes that are EOL formula. Again, these approaches assume full observability.

In [28], the authors use actual causality to learn/repair a controller of a cyber-physical system that does satisfy an expected property. To deal with the infinite state space, the input and output space are discretized into cells to obtain a finite SEM. With the finite SEM, they search for actual causes, and if an actual cause is found it provides a counterfactual controller that satisfies the property.

Some form of counterfactual reasoning has been used by many authors to diagnose, localize, and repair faults. We only cite some representative examples here. These approaches exhibit individual causes rather than chains from cause to effect, they are less apt to explain how contributory causes accumulate over time and propagate to entail an effect. The seminal work of [33] proposes a framework of model-based fault diagnosis that defines a diagnosis as a minimal sets of components whose faults make the observations consistent with the system model. The use of a distance metric is explored in [17] to localize, based on an error trace, a possible fault as the difference between the error trace and a closest correct trace. Similarly, Delta debugging [42] starts from a failing and a passing input and finds a pair of a failing and a passing input with minimal distance.

The goal of program repair is, given a program that violates an expected property P, to construct a syntactically close program that satisfies P, see e.g. [40] for the repair of reactive programs. Closer to our setting, [24] uses MaxSMT to repair clock bounds in a network of timed automata so as to ensure an expected property.

In contrast to the related work cited above, that is based on some form of model to compare the actual execution with counterfactual traces, many recent techniques summarized under the umbrella term of *explainable AI* lack a model and hence, the possibility of counterfactual reasoning [32].

Notions of robustness Different notions of robustness have been used for validation purposes. In the context of timed automata, robustness was introduced to characterize the differences in terms of *property satisfaction* between a timed automaton model and its implementation where instantaneous transitions and exact guards are not realistic (see [9] for a survey). Robustness analysis, such as in [7], uses guard enlargement and then checks the satisfaction of a safety

property by the relaxed model, called robust satisfaction. Another approach implemented in the tool Shrinktech [34] uses safe refinement for robust implementation. Besides model-checking, robustness is also used for CPS light-weight verification, namely monitoring and testing (such as in the tools RT-AMT [31], Breach [12], and S-Taliro [2]). For sequential circuits, [13] proposes a qualitative notion of robustness that is based on the distance between signals modulo disturbances. For discrete probabilistic systems, a notion of quantitative robustness, called index of robustness [3], is defined for models of risk.

The notion of robustness we propose in this thesis, which is the **robustness of choice**, has some common elements with the above-mentioned notions. Indeed it also reflects the closeness to a property violation. However, our goal is causal explanation, that is we take as input an unsafe model and blame the system choices that lead to a faulty execution. Therefore, our notion of robustness additionally needs to reflect the bad choices made by the system that lead it closer to the violation. For discrete event systems, causal explanations can be obtained by identifying the free choices of a system that lead to a violation [23]. Our approach to computing an explanation from a quantitative robustness measure can be seen as a generalization of this work to dense time.

The idea of distinguishing the choices made by the systems from the actions from the external environment makes our notion of robustness similar to those in robust control theory [43]. However, the robustness of a controller is measured by the level of (parameter) uncertainty or external disturbance under which the controller can still keep the system functioning properly (so non-determinism lies only on the environment side). On the other hand, our notion of robustness can be seen as close in spirit to the Lyapunov function for input-to-state stability analysis of control systems with external input [36], in the sense that they reflect an entity (some form of energy in case of physical control systems) that should not globally increase along the system trajectories. However, the difference is that our notion is defined for systems with both controllable and uncontrollable inputs, and for different properties (reachability versus convergence toward an equilibrium).

Chapter 3

General definitions

3.1 Domains, Variables, and Functions

The notations $\mathbb{N}, \mathbb{R}_{\geq 0}, \mathbb{R}_{>0}$ denote respectively natural integers, positive reals or 0, and positive reals. Sets are denoted by upper case letters, given a set Y of sets we write $\bigcup Y$ for $\bigcup_{Z \in Y} Z$.

Variables and Clock Variables Variables are denoted by lower case letters such as x_i, the domain of x_i is written $\mathbb{X}_i$ or $dom(x_i)$. A clock variable is a variable that takes values in $\mathbb{R}_{\geq 0}$.

Functions and Valuations Given a function $f : X \to Y$, its domain X is denoted by $dom(f)$.

Let $X = \{x_1, ..., x_n\}$ be a finite set of variables, a valuation $v : X \mapsto \bigcup\{\mathbb{X}_1, ..., \mathbb{X}_n\}$ is a partial function such that $\forall i \in [1, n], v(x_i) \in \mathbb{X}_i$. Because v is a partial function we have $dom(v) \subset X$. The set of all valuations on variables in X is written $V(X)$. For a set X of clock variables, we define a X-valuation as a total valuation in $V(X)$, the set of X-valuations is written $dom(X)$.

Predicates and Clock Constraints For a finite non-empty set Y of variables, let $PRED(Y)$ denote the set of predicates over Y. For an empty set of variable $PRED(\emptyset) := \{\top, \bot\}$. Let $c \in PRED(Y)$ be a predicate and $v \in V(Y)$ a valuation such that $dom(v) \subset Y$, $c(v)$ denotes the predicate $c' \in PRED(Y \setminus dom(v))$ where all the occurrences of variable x in $dom(v)$ are replaced by $v(x)$. For a finite non-empty set X of clock variables, an atomic constraint on X is an inequality $x \sim k$ (resp. $x - x' \sim k$) where $x, x' \in X$, $\sim \in \{\leq, <, \geq, >\}$ and $k \in \mathbb{N}$. We say that an X-valuation $v \in dom(X)$ satisfies an atomic constraint $x \sim k$ (resp. $x - x' \sim k$) if $v(x) \sim k$ (resp. $v(x) - v(x') \sim k$). An X-constraint is a finite conjunction of atomic constraints on X. The set of X-constraints is written $\Phi(X)$. By abuse of notation, we use an X-constraint interchangeably with the sets of X-valuations that satisfy it.

3.2 Automata and Operations

3.2.1 Labeled Transition Systems

Definition 1 (Labeled Transition Systems (LTS)) *A* labeled transition system $\mathcal{A} = \langle \mathcal{V}, \Sigma, \rightarrow, \nu_0, \mathcal{V}^F \rangle$ *is a 5-tuple where* $\mathcal{V}$ *is a set of states,* Σ *is a set of events,* $\rightarrow \subseteq \mathcal{V} \times \Sigma \times \mathcal{V}$ *is a set of transitions,* $\nu_0 \in \mathcal{V}$ *is the initial state, and* $\mathcal{V}^F \subseteq \mathcal{V}$ *is the set of accepting states.*

As usual, we write $\nu \xrightarrow{e} \nu'$ for $\langle \nu, e, \nu' \rangle \in \rightarrow$.

$\mathcal{A}$ is *deterministic* if whenever $(\nu, e, \nu_1) \in \rightarrow$ and $(\nu, e, \nu_2) \in \rightarrow$, $\nu_1 = \nu_2$. For a state ν, we define its preset ${}^\bullet\nu := \{\nu' \in \mathcal{V} \mid \exists e \in \Sigma : \langle \nu', e, \nu \rangle \in \rightarrow\}$ and postset $\nu^\bullet := \{\nu' \in \mathcal{V} \mid \exists e \in \Sigma : (\nu, e, \nu') \in \rightarrow\}$ as the set of states preceding and following ν in terms of transitions.

In the following, we also distinguish the notion of trace (i.e. event sequence) from the notion of *run*.

Definition 2 (Runs) *Given a LTS* $\mathcal{A} = \langle \mathcal{V}, \Sigma, \rightarrow, \nu_0, \mathcal{V}^F \rangle$*, a run of* $\mathcal{A}$*, is a finite or infinite alternating sequence* $r = \nu_0 \xrightarrow{e_1} \nu_1 \xrightarrow{e_2} \ldots$ *of states in* $\mathcal{V}$ *and events in* Σ *with* $\langle s_0, v_0 \rangle = \nu^0$.

The length of a finite run $r = \nu_0 \xrightarrow{e_1} \nu_1 \xrightarrow{e_2} \ldots \xrightarrow{e_k} \nu_k$ *is* $|r| = k$.

A run r *is* accepting *if it is finite and* $\nu_{|r|} \in \mathcal{V}^F$*, or if it is infinite and* $\nu_k \in \mathcal{V}^F$ *for some* $k \geq 0$.

Let $run(\mathcal{A})$ *(resp.* $run^F(\mathcal{A})$*) denote the set of runs (resp. accepting runs) of* $\mathcal{A}$.

A *trace* $w \in \Sigma^*$ is a finite sequence of events. The concatenation of a trace w with some event a is written $w \cdot a$. The length of a trace w is written $|w|$ and it is the number of occurrences of events.

Definition 3 (Traces) *Given a LTS* $\mathcal{A} = \langle \mathcal{V}, \Sigma, \rightarrow, \nu_0, \mathcal{V}^F \rangle$*, and a run* $r = \nu_0 \xrightarrow{e_1} \nu_1 \xrightarrow{e_2} \ldots$ *of* $\mathcal{A}$*, the* trace *of the run* r *is the sequence of events* $e_1\ e_2 \ldots$ *labeling the transitions of* r.

The language $L(\mathcal{A})$ *(resp. accepting language* $L^F(\mathcal{A})$*) of* $\mathcal{A}$*, is the set of traces (resp. accepting traces) of runs of* $\mathcal{A}$.

3.2.2 Extended Automaton

Definition 4 (Extended automaton) *An extended automaton is a tuple* $\mathcal{A} = \langle S, \Sigma, X, inv, \delta, s^0, v^0, F \rangle$ *where*

- S *is the set of control states;*
- Σ *is an alphabet of events;*
- X *is the set of variables;*
- $inv : S \rightarrow PRED(X)$ *defines an invariant for each state*
- $\delta : S \times \Sigma \times S \rightarrow 2^{PRED(X) \times UPDATE(X)}$ *where* $UPDATE(X) = X \mapsto (\mathbb{X} \mapsto \bigcup\{\mathbb{X}_1, ..., \mathbb{X}_n\})$ *with* $x_i \mapsto (\mathbb{X} \mapsto \mathbb{X}_i)$ *is a partial function that defines an update;*

- $s^0 \in S$ *is the initial state;*
- $v^0 : X \to \bigcup\{\mathbb{X}_1, ..., \mathbb{X}_n\}$*, is the initial valuation;*
- $F \subseteq S$ *is the set of accepting control states.*

A transition $\langle s, e, s', g, f\rangle$ *is denoted by* $s \overset{e,g,f}{\to} s'$*, where* $s, s' \in S$*,* $e \in \Sigma$*,* $g \in PRED(X)$*, and* $f \in UPDATE(X)$*.*

Given a valuation $v : X \mapsto \bigcup\{\mathbb{X}_1, ..., \mathbb{X}_n\}$ and a partial update $f_i = (x_i \mapsto (\mathbb{X} \mapsto \mathbb{X}_i))$, we write $f_i(v)$ for the valuation v' obtained by applying f_i to the global valuation v, that is, $v'(x) = f_i(x_i)(v)$ and $\forall x \in X \setminus \{x_i\} : v'(x) = v(x)$.

Definition 5 (Semantics of an extended automaton) *The semantic LTS of an extended automaton* $\mathcal{A} = \langle S, \Sigma, X, inv, \delta, s^0, v^0, F\rangle$ *is the LTS* $[\mathcal{A}] = \langle \mathcal{V}, \Sigma, \to, \nu^0, \mathcal{V}^F\rangle$ *where*

- $\mathcal{V} = \{(s, v) \in S \times \mathbb{X} \mid inv(s)(v)\}$*;*
- $\to = \big\{\big((s, v), e, (s', v')\big) \in \mathcal{V} \times \Sigma \times \mathcal{V} \mid \delta(s, e, s') = \langle g, f\rangle \wedge g(v) \wedge v' = f(v)\big\}$*;*
- $\nu^0 = (s^0, v^0)$*;*
- $\mathcal{V}^F = F \times \mathbb{X}$*.*

where for a valuation $v : X \mapsto \bigcup\{\mathbb{X}_1, ..., \mathbb{X}_n\}$ *and an update* $f = \{x_{i_1} \mapsto f_1, ..., x_{i_k} \mapsto f_k\}$*, we write* $f(v)$ *as a shorthand notation for the valuation* $v' = (f_1 \circ ... \circ f_k)(v)$ *where we assume w.l.o.g. that* $\forall j \in [2, k]$*,* f_j *does not depend on* $x_{i_1}, ..., x_{i_{j-1}}$*.*

Intuitively, assignments are executed in an order that is consistent with the dependencies between variables in the update function. For instance, the update function $\{x \mapsto z + 1, y \mapsto x, z \mapsto z - 1\}$ is executed in the order $z := z - 1; x := z + 1; y := x$. This semantics allows us to instantaneously observe property violations.

Definition 6 (Determinism) *An extended automaton* $\mathcal{A}$ *is deterministic if* $[\mathcal{A}]$ *is deterministic.*

Proposition 1 *The semantic LTS of an extended automaton* $\langle S, \Sigma, X, inv, \delta, s^0, v^0, F\rangle$ *is deterministic if* $\forall s, s_1, s_2 \in S\ \forall e \in \Sigma$*:*

$$\begin{aligned}&(g_1, f_1) \in \delta(s, e, s_1) \wedge (g_2, f_2) \in \delta(s, e, s_2) \implies \\ &\quad (s_1 = s_2 \wedge f_1 = f_2) \vee \\ &\quad g_1 \wedge f_1^{-1}\big(inv(s_1)\big) \wedge g_2 \wedge f_2^{-1}\big(inv(s_2)\big) = false\end{aligned}$$

where $f^{-1}(P)$ *is the weakest predicate* P' *such that* $\forall v : P'(v) \implies P\big(f(v)\big)$*.*

In the sequel, we consider only deterministic extended automata. Determinism is preserved under composition.

Definition 7 (Composition of extended automata) *Let* $\mathcal{A}_i = \langle S_i, \Sigma_i, X_i, inv_i, \delta_i, s_i^0, v_i^0, F_i \rangle$ *for* $i \in \{1,2\}$*, be two extended automata with* $S_1 \cap S_2 = \emptyset$*,* v_1^0 *and* v_2^0 *are consistent, and for any* $s_i \overset{e,g_i,f_i}{\rightarrow} s_i'$*,* $i = 1,2$ *with* $e \in \Sigma_1 \cap \Sigma_2$*,* f_1 *and* f_2 *are consistent.*

The composition of $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *is the extended automaton*

$\mathcal{A}_1 \| \mathcal{A}_2 = \langle S_1 \times S_2, \Sigma_1 \cup \Sigma_2, X_1 \cup X_2, inv, \delta, \langle s_1^0, s_2^0 \rangle, v_1^0 \cup v_2^0, F_1 \times F_2 \rangle$ *such that* $\forall \langle s_1, s_2 \rangle \in S_1 \times S_2 : inv(s_1, s_2) = inv_1(s_1) \wedge inv_2(s_2)$*, and*

$$\begin{aligned}\delta = &\big\{\langle\langle s_1, s_2\rangle, e, \langle s_1', s_2\rangle\rangle, \delta_1(s_1, e, s_1')\rangle \mid e \in \Sigma_1 \setminus \Sigma_2\big\} \cup \\ &\big\{\langle\langle s_1, s_2\rangle, e, \langle s_1, s_2'\rangle\rangle, \delta_2(s_2, e, s_2')\rangle \mid e \in \Sigma_2 \setminus \Sigma_1\big\} \cup \\ &\big\{\langle\langle s_1, s_2\rangle, e, \langle s_1', s_2'\rangle\rangle, \delta_1(s_1, e, s_1') \otimes \delta_2(s_2, e, s_2')\rangle \mid \\ &\quad e \in \Sigma_1 \cap \Sigma_2\big\}\end{aligned}$$

where

$$\begin{aligned}\delta_1(s_1, e, s_1') \otimes \delta_2(s_2, e, s_2') = \big\{\langle g, f\rangle \mid \\ \exists \langle g_1, f_1\rangle \in \delta_1(s_1, e, s_1')\ \exists \langle g_2, f_2\rangle \in \delta_2(s_2, e, s_2') : \\ \big(g = g_1 \wedge g_2 \wedge \bigwedge_{x \in X_1 \cap X_2} f_1(x) = f_2(x)\big) \wedge g \neq \mathit{false} \\ \wedge f = f_2 \cup \{x \mapsto f_1(x) \mid x \in dom(f_1) \setminus dom(f_2)\}\big\}\end{aligned}$$

In the above definition, the transition relation of the composition consists of the interleaving transitions for events that belong to exactly one component, and the composition $\otimes$ of transitions labeled with events shared between both components. The composition makes sure that a transition labeled with a shared event is enabled if and only if there are transitions of the components such that the event is enabled in both transitions, and the update functions agree, in the state from which the transitions are issued, on their assignments to all shared variables. The expression $f_1(x) = f_2(x)$ is a predicate that characterizes the states in which both functions yield identical assignments to the shared variables.

Definition 8 (Consistent updates) *Two functions* $f_i \in UPDATE(X_i)$*,* $i = 1,2$*, are* consistent *if the functions do not induce any cyclic dependency between variables.*

Throughout this thesis, we assume the updates of any transition to be consistent.

Definition 9 (Projection of a semantic state) *Given a composed extended automaton* $\mathcal{A}_1 \| \mathcal{A}_2 = \langle S, \Sigma, X, inv, \delta, s^0, v^0, F \rangle$ *with* $[\mathcal{A}_1 \| \mathcal{A}_2] = \langle \mathcal{V}, \Sigma, \rightarrow, \nu^0, \mathcal{V}^F \rangle$ *and* $\nu = \langle \nu_1, \nu_2 \rangle \in \mathcal{V}$*, we define the projection of* ν *on* $\mathcal{A}_i$*,* $i = 1,2$*, as* $\nu_{\downarrow \mathcal{A}_i} := \nu_i$*.*

Since composition is associative, it is straightforward to extend projection to products of more than two extended automata.

Again, we distinguish the notion of *run* and *trace*.

Definition 10 (Run) *Given an extended automaton* $\mathcal{A} = \langle S, \Sigma, X, inv, \delta, s^0, v^0, F \rangle$ *with* $[\mathcal{A}] = \langle \mathcal{V}, \Sigma, \rightarrow, \nu^0, \mathcal{V}^F \rangle$*, a run of* $\mathcal{A}$ *is a run of* $[\mathcal{A}]$*.*

Let $run(\mathcal{A})$ *(resp.* $run^F(\mathcal{A})$*) denote the set of runs (resp. accepting runs) of* $\mathcal{A}$*.*

A run r *is* maximal *if it is infinite, or the last state* $\langle s_k, v_k\rangle$ *is a sink state.*

Given a run $r \in run(\mathcal{A})$*, let* $\hat{r} = s_0 \overset{e_0,g_0,f_0}{\to} s_1 \overset{e_1,g_1,f_1}{\to} \dots \overset{e_{k-1},g_{k-1},f_{k-1}}{\to} s_k$ *be the sequence of states and transitions of* $\mathcal{A}$ *such that* $\forall i \in \{0,...,k-1\} : g_i(v_i) \wedge v_{i+1} = f_i(v_i) \wedge inv(s_{i+1})(v_{i+1})$*.*

For two semantic states $\langle s, v\rangle, \langle s', v'\rangle$ *in* $\mathcal{V}$*, we define the parts of the runs in between those states as* $run(\mathcal{A}, s, v, s', v') = \{\langle s_i, v_i\rangle \overset{e_i}{\to} \dots \overset{e_{j-1}}{\to} \langle s_j, v_j\rangle \mid \forall k \in [i, j-1]\ \langle s_{k-1}, v_{k-1}\rangle \overset{e_k}{\to} \langle s_k, v_k\rangle \wedge \langle s_i, v_i\rangle = \langle s, v\rangle \wedge \langle s_j, v_j\rangle = \langle s', v'\rangle\}$*.*

Intuitively, $\hat{r}$ is the (unique) sequence of syntactic transitions that produces r.

Definition 11 (Traces) *The* language $L(\mathcal{A})$ *(resp. accepting language* $L^F(\mathcal{A})$*) of an extended automaton* $\mathcal{A}$ *is* $L([\mathcal{A}])$ *(resp.* $L^F([\mathcal{A}])$*).*

As the definition of accepting infinite traces will be used only on extended automata whose set of accepting states does not have any outgoing transition, the naive acceptance criterion is sufficient for our purposes.

In the sequel, we restrict ourselves to models for which the state space reachable from the initial state is finite.

For a state s, we define its preset ${}^\bullet s := \{s' \in S \mid \exists e \in \Sigma : \langle s', e, s', \cdot, \cdot\rangle \in \delta\}$ and postset $s^\bullet := \{s' \in S \mid \exists e \in \Sigma : \langle s, e, s', \cdot, \cdot\rangle \in \delta\}$ as the states preceding and following s by one transition.

Definition 12 (Projection) *Let* Σ, Σ' *be two sets of events such that* $\Sigma' \subseteq \Sigma$*. Let* $\cdot_{\downarrow\Sigma'} : \Sigma^* \to \Sigma'^*$ *be the projection defined by induction on traces:*

$$\forall e \in \Sigma, e_{\downarrow\Sigma'} = \begin{cases} e \cap \Sigma' & \text{if } e \cap \Sigma' \neq \emptyset \\ \epsilon & \text{otherwise} \end{cases}$$

and $\forall t \in \Sigma^*, \forall e \in \Sigma : (t \cdot e)_{\downarrow\Sigma'} = t_{\downarrow\Sigma'} \cdot e_{\downarrow\Sigma'}$ *where* ϵ *is the neutral element for the concatenation.*

3.2.3 Timed Automata

Definition 13 (Timed Automaton) *A timed automaton (TA)* $\mathcal{A}$ *is a tuple* $\langle \Sigma, L, X, \mathcal{I}, E, \ell_0, L^F\rangle$ *where:*

- Σ *is a finite set of discrete* events*;*
- L *is a finite set of* locations*;*
- $\ell_0 \subseteq L$ *is the* initial location*;*
- $L^F \subseteq L$ *is a set of* accepting locations*;*
- X *is a set of* clock variables*;*
- $\mathcal{I} : L \to \Phi(X)$ *specifies for each location an* invariant*;*
- $E \subseteq L \times \Phi(X) \times \Sigma \times 2^C \times L$ *is a set of edges of the form* $e = \langle \ell, g, \sigma, R, \ell'\rangle$ *where* ℓ *and* ℓ' *are respectively* source *and* target locations*;* σ *is an* event*;* g *is the* guard *of* e*; and* R *is a set of clocks to be* reset *when the edge is traversed.*

As usual, we write $\ell \xrightarrow{g,\sigma,R} \ell'$ for $\langle \ell, g, \sigma, R, \ell' \rangle \in E$. We formalize the semantics of timed automata using LTS.

Definition 14 (Semantic LTS) *The* semantic LTS *of a timed automaton* $\mathcal{A} = \langle \Sigma, L, X, \mathcal{I}, E, \ell_0, L^F \rangle$ *is the LTS* $[\mathcal{A}] = \langle \mathcal{V}, \Sigma', \rightarrow, \nu^0, \mathcal{V}^F \rangle$ *where:*

- $\Sigma' = \Sigma \cup \mathbb{R}_{\geq 0}$ *is the set of events;*
- $\mathcal{V} = \{\langle \ell, v \rangle) \mid \ell \in L \wedge v \in \mathcal{I}(\ell)\}$ *is the set of states of* $\mathcal{A}$*. Each state is a pair* $\langle \ell, v \rangle$ *where* $v \in \mathcal{I}(\ell)$ *is a clock valuation that satisfies the invariant of the location* ℓ*;*
- $\nu^0 = \langle \ell_0, \mathbf{0} \rangle$*;*
- $\mathcal{V}^F = \{\langle \ell, v \rangle \in \mathcal{V} \mid \ell \in L^F\}$*;*
- *the set of transitions are of two types, discrete and delay transitions:*

$$\begin{aligned} \rightarrow = & \left\{((\ell, v), \sigma, (\ell', v')) \mid \exists g, R : \langle \ell, g, \sigma, R, \ell' \rangle \in E \wedge v' = v[R := 0] \wedge v' \in \mathcal{I}(\ell')\right\} \\ & \cup \left\{\big((\ell, v), \delta, (\ell, v')\big) \mid t \in \mathbb{R}_{>0} \wedge v' = v + t \wedge v' \in \mathcal{I}(\ell)\right\} \end{aligned}$$

  The states (ℓ', v') *and* (ℓ, v') *are respectively called discrete-successor delay-successor of* (ℓ, v)*.*

For a state $\langle \ell, v \rangle$ and a time $t \in \mathbb{R}_{>0}$, we write $\langle \ell, v \rangle + t := \langle \ell, v + t \rangle$ where $v + t$ is the valuation such that $\forall x \in X, (v + t)(x) = v(x) + t$.

Note that since t is a real number, a delay transition $(\ell, v) \xrightarrow{t} (\ell, v + t)$ can be split into an arbitrary number k of delay transitions, that is, $(\ell, v) \xrightarrow{t_1} (\ell, v_1) \xrightarrow{t_2} (\ell, v_2) \ldots \xrightarrow{t_k} (\ell, v_k)$ such that $t = t_1 + t_2 + \ldots + t_k$.

3.2.4 Bisimulations

The sets of states and transitions of a timed automaton are infinite, and therefore, as in verification, finite abstractions can be used to construct explanations. In this work, we use time-abstracting bisimulations [38] to abstract away the quantitative information about time lapses in the run of a timed automaton. This leads to finite discrete abstractions of the original timed automata from which we can compute choice-based explanations [15]. The following definition is adapted from [38] so as to distinguish events, and accepting vs. non-accepting states.

Definition 15 (Strong bisimulations) *A binary relation* $\sim$ *on an LTS* $G = \langle \mathcal{V}, \Sigma, E, \mathcal{V}^0, \mathcal{V}^F \rangle$ *is a* strong bisimulation *if for any pair of states* p *and* q *of* G *such that* $p \sim q$*, the following conditions hold:*

- $\forall \sigma \in \Sigma \; \forall p' \in \mathcal{V} : \big(p \xrightarrow{\sigma} p' \implies \exists q' : q \xrightarrow{\sigma} q' \wedge p' \sim q'\big)$*;*
- *the above condition also holds when* p *and* q *are swapped;*
- $p \in \mathcal{V}^F \iff q \in \mathcal{V}^F$*.*

Let $\mathcal{A}$ be a TA, the relation $\sim$ is a strong time-abstracting bisimulation (STAB) *if for any pair of states $p = \langle \ell_1, v_1 \rangle$ and $q = \langle \ell_2, v_2 \rangle$ of $[\mathcal{A}] = \langle \mathcal{V}, \Sigma \cup \mathbb{R}_{>0}, \rightarrow \rangle$ such that $p \sim q$, the following conditions hold:*

- $\ell_1 = \ell_2$;
- $\forall e \in \Sigma \setminus \mathbb{R}_{>0} \;\; \forall p' \in \mathcal{V} : \big(p \xrightarrow{e} p' \implies \exists q' : q \xrightarrow{e} q' \wedge p' \sim q'\big)$;
- $\forall t > 0 \; \forall p' \in \mathcal{V} : \big(p \xrightarrow{t} p' \implies \exists t' > 0 \; \exists q' : q \xrightarrow{t'} q' \wedge p' \sim q'\big)$;
- *the above conditions also hold when p and q are swapped.*

The STAB-quotient graph of $[\mathcal{A}]$ with respect to STAB is the LTS $[\mathcal{A}]_\sim = \langle Z, \Sigma \cup \{\delta\}, \rightarrow_\sim \rangle$. Z is the set of classes of $\mathcal{V}$ w.r.t. $\sim$ and is called the set of zones. *The location is the same for all states in a zone $z \in Z$. We say, by abuse of notation, for $v \in dom(X)$ and $z \in Z$, that $v \in z$ if v is the valuation of a state in z. The event δ abstracts the delays in $\mathbb{R}_{>0}$. In order for $[\mathcal{A}]_\sim$ to be deterministic, we remove from $\rightarrow_\sim$ the transitions $z \xrightarrow{\delta}_\sim z'$ such that $\exists z'' \in Z, z \xrightarrow{\delta}_\sim z'' \wedge z'' \xrightarrow{\delta}_\sim z'$. Whenever $z \xrightarrow{\delta}_\sim z'$, z' is called the δ-*successor *of z.*

Chapter 4

Discrete explanations on DES

Resumé

Les contributions de ce chapitre sont le fruit d'un travail commun avec Gregor Gössler, Yannick Pencolé, Louise Travé-Massuyès et moi-même. Dans ce travail, je me suis concentré sur les explications de choix, l'encodage de la sémantique des explications de choix, la formalisation des propriétés des explications et les preuves que les explications de choix satisfont ces propriétés. Yannick Pencolé s'occupe de l'implémentation et l'explication de l'étude de cas est calculée sur son logiciel.

Les contributions de ce chapitre sont les suivantes :

- Nous proposons un cadre formel pour le problème d'explication qui permet de traiter des systèmes avec variables en plus des événements. Le comportement d'un système est en effet modélisé par des automates étendus avec des variables, tout comme les observations du système.
- Une autre contribution consiste à proposer un modèle formel pour les explications et à définir les propriétés attendues des explications sur la base de la sémantique proposée.
- Dans ce modèle, nous définissons des explications de choix et prouvons qu'elles satisfont les propriétés attendues.
- En utilisant le cadre formel, nous concevons un système de régulateur de vitesse adaptatif, il est utilisé pour illustrer les concepts introduits et la pertinence des explications de choix est démontrée dans une étude de cas du trafic routier.

4.1 Introduction

The contributions in that chapter are a joint work with Gregor Gössler, Yannick Pencolé, Louise Travé-Massuyès and myself. In this work my focus was on the choice explanations, encoding the semantics of choice explanations, formalization of properties of explanations, and the proofs

that choice explanations satisfy those properties. Yannick Pencolé is doing the implementation and the explanation in the case study is computed on his software.

The contributions of the chapter are:

- We propose a formal framework for the explanation problem that allows dealing with variables in addition to events. The behavior of a system is indeed represented by automata extended with variables, just as the observations of the system.
- Another contribution lies in proposing a formal template for explanations and the definition of expected properties of explanations based on proposed semantics.
- In that template, we define choice explanations and prove that they satisfy the expected properties.
- Using the formal framework, we design an adaptive cruise control system, it is used to illustrate the introduced concepts and the relevance of choice explanations is shown in a case study of road traffic.

The chapter is organized as follows. Section 4.2 describes the problem statement. Section 4.6 presents the case study. Section 4.3 presents the formal framework of explanations and their semantics and defines the expected properties of explanations. Section 4.4 proposes a first attempt to define an explanation as a subsequence of events and describes why such a definition is not satisfactory. Section 4.5 then presents a new definition, called *choice explanation*, and proposes an algorithm to compute choice explanations for the violation of a property, these being consistent with the observations of the system. We prove that the choice explanations satisfy the expected properties of Section 4.3. Section 4.6 presents the implementation and explanations of the case study. Finally, Section 4.7 resumes the contributions and outlines some perspectives for future work.

4.2 Problem Statement

We address the problem of defining and computing different notions of explanation for the violation of a given behavioral property by a DES under partial observation. We consider that a behavioral model of the system is available and is represented as a deterministic extended automaton $\mathcal{A} = \langle S, \Sigma, X, inv, \delta, s^0, v^0, F \rangle$. This automaton can be monolithic or can result from a multi-component model, in which case the system is modeled by a set of interacting automata.

The system has a prefix-closed behavioral description. That means that every prefix of behavior respecting the behavioral description also respects the behavioral description so $S = F$. We assume the set Σ of events to be partitioned into *controllable events* Σ_c (e.g., events executed by the system) and *uncontrollable events* Σ_u (e.g., events executed by its environment). In a composition, we consider events shared by several components as controllable if they are controllable for all owners; all other shared events are uncontrollable. In addition, we assume Σ to be partitioned into *observable events* Σ^o that are recorded and *non-observable events* Σ^{uo}

that are not recorded. Both partitions are independent. In particular, there may be controllable events that are not observable (e.g., due to memory constraints in the logging system), and uncontrollable events that are observable (e.g., relevant events received from the system environment), and events that are neither controllable nor observable.

We assume that there are no unobservable cycles possible in $\mathcal{A}$, in other words, we assume that the number of unobservable events that can be generated by the system between any two observable events is bounded. Any run of the system can be associated with a *log* that identifies a finite sequence of observable events. We consider that the log is an extended automaton $\mathcal{L}$.

Definition 16 (Log) *Let* $\mathcal{A} = \langle S, \Sigma, X, inv, \delta, s^0, v^0, F\rangle$ *be a system model with* $\Sigma = \Sigma^o \uplus \Sigma^{uo}$*, and a run* $r = \langle s_0, v_0\rangle \xrightarrow{e_0} \langle s_1, v_1\rangle \xrightarrow{e_1} ... \xrightarrow{e_{k-1}} \langle s_k, v_k\rangle \in run(\mathcal{A})$ *with* $\hat{r} = s_0 \overset{e_0,g_0,f_0}{\rightarrow} s_1 \overset{e_1,g_1,f_1}{\rightarrow} ... \overset{e_{k-1},g_{k-1},f_{k-1}}{\rightarrow} s_k$*. The* log *of* r *with observable variables* $X' \subseteq X$ *is an extended automaton*

$\mathcal{L}_{X'}(r) = \langle S', \Sigma^o, X', inv', \delta', (s^0)', (v^0)', F'\rangle$ *where*

- $S' = \{s'_0, ..., s'_\ell\}$*, where* ℓ *is the number of observable events in* r*;*
- $inv' = \{s \mapsto \top \mid s \in S'\}$*;*
- $(v^0)'$ *is an arbitrary valuation;*
- *there exists a relation* $R \subseteq \{s_0, ..., s_k\} \times S'$ *such that* $(s_0, s'_0) \in R$ *and* $(s_k, s'_\ell) \in R$ *and* δ' *is minimal such that whenever* $(s_i, s'_j) \in R$ *with* $i \leq k-1$ *then*
  - *either* $e_i \in \Sigma^o$ *and* $s_i \overset{e_i,g_i,f_i}{\rightarrow} s_{i+1}$ *and* $s'_j \overset{e_i,true,f'_i}{\rightarrow} s'_{j+1}$ *with* $f'_i = \{x \mapsto v_{i+1}(x) \mid x \in X' \cap dom f_i\}$ *and* $(s_{i+1}, s'_{j+1}) \in R$*,*
  - *or* $e_i \in \Sigma^{uo}$ *and* $(s_{i+1}, s'_j) \in R$*;*
- $F' = \{s'_\ell\}$.

Note that the log is defined so as to enable, when composed with $\mathcal{A}$, exactly the set of prefixes of runs that produce the logged observations.

As the system is partially observable, a log $\mathcal{L}$ can be associated with several possible traces (these are the traces that are *consistent* with the log $\mathcal{L}$). Formally, the set of traces consistent with the log $\mathcal{L}$ is given by:

$$tr(\mathcal{L}) = \{t \in \Sigma^* \mid t_{\downarrow \Sigma_o} = \mathcal{L}\} \cap L(\mathcal{A})$$

On the other hand, we also assume the set X of variables includes a subset of *observable variables* $X' \subseteq X$, while the others remain unobserved. The valuation of an observable variable is observed each time an observable event makes an assignment to it.

A safety property P is respected by the system as long as the current run leads to a state where the property is satisfied. If the run of the system leads to a state that does not satisfy the property, then we say that the property is violated. The violation of a safety property is permanent, i.e. all continuations of a run violating the safety property also violate the property.

Throughout this chapter, we focus on explaining the violation of a (regular) safety property P in contrast to a desired property Q. Because an explanation is often more informative when it is given relative to some contrast case [30], we are indeed interested in *contrastive explanations* that can answer the question "Why did it happen that the safety property P was violated rather than the property Q satisfied?". Satisfaction or violation of P and Q is tracked by *observer* automata.

Definition 17 (Observer) *Given an extended automata* $\mathcal{A} = \langle S, \Sigma, X, inv, \delta, s^0, v^0, F\rangle$ *and* $\mathcal{O} = \langle S_\mathcal{O}, \Sigma_\mathcal{O}, X_\mathcal{O}, inv_\mathcal{O}, \delta_\mathcal{O}, s^0_\mathcal{O}, v^0_\mathcal{O}, F_\mathcal{O}\rangle$*, we say that* $\mathcal{O}$ *is an* observer *for the observed model* $\mathcal{A}$ *if*

- $\mathcal{O}$ *has a single accepting state that is a sink state;*
- $\mathcal{O}$ *is receptive:* $\forall s \in S_\mathcal{O}\ \forall v\ \forall e \in \Sigma_\mathcal{O}\ \exists s' \in S_\mathcal{O} : s \overset{e,g,f}{\to} s' \wedge g(v) \wedge inv_\mathcal{O}(s')\big(f(v)\big)$*; and*
- $\Sigma_\mathcal{O} \subseteq \Sigma$ *and* $\forall\big((s, e, s'), g, f\big) \in \delta_\mathcal{O}\ \forall x \in X : x \notin domf$*.*

Note that by hypothesis, all extended automata are deterministic. Moreover, determinism is preserved under composition. Hence, there is no need to require determinism of $\mathcal{A}\|\mathcal{O}$ here.

That is, an observer may observe a subset $\Sigma_\mathcal{O} \subseteq \Sigma$ of the events and a subset $X_\mathcal{O} \cap X$ of the variables of the observed model $\mathcal{A}$. It may also have "private" variables $X_\mathcal{O} \setminus X$, for instance, to count events. The observer does not update variables that are shared with $\mathcal{A}$.

Because the language accepted by an observer in isolation does not reflect its state constraints, we define below the safety property it models when composed with a model to be observed.

Given an observer $\mathcal{O}$ for a behavioral model $\mathcal{A}$ with $[\mathcal{A}\|\mathcal{O}] = \langle \mathcal{V}, \Sigma, \to, \nu^0, \mathcal{V}^F\rangle$, let $\mathsf{ok}_\mathcal{A}(\mathcal{O}) = \{\nu \in \mathcal{V} \mid \nu_{\downarrow\mathcal{O}} \notin F\}$ be the set of states in which $\mathcal{O}$ is not in its accepting state.

Considering $\mathcal{V}' = \mathsf{ok}_\mathcal{A}(\mathcal{O})$, the set of words reaching exactly the set of safe states with respect to $\mathcal{V}'$ in $[\mathcal{A}\|\mathcal{O}]$ can be defined as follows.

Definition 18 (Induced safe language) *Given a behavioral model* $\mathcal{A}$ *over* Σ *and an observer* $\mathcal{O}$ *with* $[\mathcal{A}\|\mathcal{O}] = \langle \mathcal{V}, \Sigma, \to, \nu^0, \mathcal{V}^F\rangle$*, the safe language induced by* $\mathcal{O}$ *on* $\mathcal{A}$ *with respect to a set of uncontrollable events* $U \subseteq \Sigma$*, written* $\mathsf{safeLang}_{\mathcal{A},U}(\mathcal{O})$*, is the greatest language* $X \subseteq L(\mathcal{A}) \setminus L^F(\mathcal{A}\|\mathcal{O})$ *such that* $\forall w \in X$*:*

- $\exists e \in \Sigma : w.e \in X$*, and*
- $\forall e \in U : (w.e \in L(\mathcal{A}) \implies w.e \in X)$*.*

We omit the parameter U *when we denote the safety language with respect to all the uncontrollable events of* $\mathcal{A}$*, i.e.* $U = \Sigma_u$*.*

Note that we rely on the determinism of the automata to ensure that the induced safe language is exactly the language that keeps the product safe with respect to observed violations.

Let us consider two observers $\mathcal{O}_P$ and $\mathcal{O}_Q$ that observe the safety property P and a contrastive property Q. We can now formally represent as an extended automaton:

1. the set of runs of the system that violate the safety property P: $\mathcal{A}\|\mathcal{O}_P$;
2. the set of runs of the system that violate the contrastive property Q: $\mathcal{A}\|\mathcal{O}_Q$
3. the set of runs of the system that violate the safety property P and the contrastive property Q: $\mathcal{A}\|\mathcal{O}_P\|\mathcal{O}_Q$;
4. the set of runs of the system that are consistent with a given log $\mathcal{L}$: $\mathcal{A}\|\mathcal{L}$;
5. the set of runs of the system that violate the safety property P or the contrastive property Q, *and* that are consistent with a given log $\mathcal{L}$: $\mathcal{A}\|\mathcal{O}_P\|\mathcal{L}$ and $\mathcal{A}\|\mathcal{O}_Q\|\mathcal{L}$ respectively;
6. the set of runs of the system that violate the safety property P and the contrastive property Q and that are consistent with a given log $\mathcal{L}$: $\mathcal{A}\|\mathcal{O}_P\|\mathcal{O}_Q\|\mathcal{L}$.

Note that composition being an associative operation, $\mathcal{A}\|\mathcal{O}_\diamondsuit\|\mathcal{L}$, $\diamondsuit \in \{P, Q\}$, is perfectly defined as $(\mathcal{A}\|\mathcal{O}_\diamondsuit)\|\mathcal{L} = \mathcal{A}\|(\mathcal{O}_\diamondsuit\|\mathcal{L})$. A state of $\mathcal{A}\|\mathcal{O}_\diamondsuit\|\mathcal{L}$ is a tuple $\langle \nu_a, \nu_p, \nu_l \rangle$ where ν_a, ν_p and ν_l are the respective states in $\mathcal{A}$, $\mathcal{O}_\diamondsuit$ and $\mathcal{L}$.

Given a system model $\mathcal{A}$, a safety property P, and a log $\mathcal{L}$ violating P, we aim at constructing a causal explanation for this violation in contrast to the satisfaction of a desired property Q. More precisely, our goal is to identify what controllable parts of the behavior are consistent with the log and can be blamed for the violation and eventually to which extent the uncontrollable events come into play. In this chapter, we propose two kinds of causal explanations, that characterize differently what action is causally relevant with respect to the observed violation. Our causal explanations are obtained by assembling those actions together in a consistent order.

4.3 Expected Properties of Explanations

In order to evaluate and compare different kinds of explanations, we formalize a set of expected properties. These properties are parametrized with the semantics of an explanation, in terms of a set of traces, which we define in the sequel for each of the approaches we study. An explanation function assigns an explanation to an observed behavior. Furthermore, an explanation function is paired with a semantic function that assigns to an explanation a set of behaviors of the model. With a semantic function, the explanation is a causal generalization of the observed violation of the safety property, in the sense that, for a given notion of causality, the explanation function encodes the causes of the violation in the explanation and the semantic function returns the set of behaviors that are consistent with those encoded causes.

Given a model $\mathcal{A}$, a trace tr , and an observer $\mathcal{O}$ for a given safety property, consider an *explanation function* $\mathcal{E}$ be such that $\mathcal{E}(\mathcal{A}, \mathcal{O}, tr)$ is a set of possible explanations for the violation of the property observed by $\mathcal{O}$ in tr. Furthermore, we assume that, given $\mathcal{A}$ and $\mathcal{O}$, an explanation $\varepsilon \in \mathcal{E}(\mathcal{A}, \mathcal{O}, tr)$ uniquely defines a set of runs of $\mathcal{A}$ that are said to be covered by ε. We call this set of runs $sem_{\mathcal{A}}(\varepsilon)$, the semantics of ε. Hence, $sem_{\mathcal{A}}(\varepsilon) \subseteq L(\mathcal{A})$. To assert the quality of explanation functions, we define formal requirements with respect to the safety property and to the trace in terms of properties.

Property 1 (Soundness) *An explanation ε is* sound *with respect to a safety property observed by $\mathcal{O}$ if*

$$sem_{\mathcal{A}}(\varepsilon) \cap \mathsf{safeLang}_{\mathcal{A},\emptyset}(\mathcal{O}) = \emptyset$$

An explanation function $\mathcal{E}$ is sound with respect to $\mathcal{O}$ if for any $\varepsilon \in \mathcal{E}(\mathcal{A}, \mathcal{O}, tr)$, ε is sound with respect to the property observed by $\mathcal{O}$.

Intuitively, ε is sound if the extensions of all traces in the semantics of ε eventually violate the observed property. In other words, a sound explanation represents a set of runs for which the property violation is unavoidable.

Property 2 (Completeness) *An explanation function $\mathcal{E}$ is* weakly complete *if $\forall \varepsilon \in \mathcal{E}(\mathcal{A}, \mathcal{O}, tr)$, $tr \in sem_{\mathcal{A}}(\varepsilon)$. An explanation ε is* complete *with respect to $\mathcal{O}$ and $\mathcal{L}$ if*

$$L^F(\mathcal{A} \| \mathcal{O} \| \mathcal{L}) \subseteq sem_{\mathcal{A}}(\varepsilon)$$

Completeness means that all runs whose log is $\mathcal{L}$ and that violate the safety property observed by $\mathcal{O}$, are covered by the explanation.

Remember that we are interested in *causal* explanations that exhibit a set of *contributory causes* that together form a *sufficient cause* that entails the violation of an expected safety property. Hence the formalization of explanations depends on what we consider as a contributory cause. Different definitions of contributory causes put the spotlight on different aspects of the behavior and may blame the property violation on different portions of a run. We discuss and formalize explanations based on two different types of causes in the next two sections.

Now let us discuss the *precision* of an explanation.

Proposition 2 (Precision) *Consider a model $\mathcal{A}$, two observers $\mathcal{O}_1$, $\mathcal{O}_2$ with disjoint sets of private variables, traces tr_1 and tr_2, and an explanation function $\mathcal{E}$ that is sound with respect to $\mathcal{O}_1$ and $\mathcal{O}_2$. We have for any $\varepsilon_i \in \mathcal{E}(\mathcal{A}, \mathcal{O}_i, tr_i)$, $i = 1, 2$,*

$$L^F(\mathcal{A} \| \mathcal{O}_1 \| \mathcal{O}_2) = \emptyset \implies sem_{\mathcal{A}}(\varepsilon_1) \cap sem_{\mathcal{A}}(\varepsilon_2) = \emptyset \tag{4.1}$$

If in addition, $\mathcal{E}$ is weakly complete,

$$L^F(\mathcal{A} \| \mathcal{O}_1 \| \mathcal{O}_2) = \emptyset \implies \mathcal{E}(\mathcal{A}, \mathcal{O}_1, tr_1) \cap \mathcal{E}(\mathcal{A}, \mathcal{O}_2, tr_2) = \emptyset$$

Intuitively, a sound and weakly complete explanation function is *precise* in the sense that two disjoint property violations have disjoint sets of explanations, and any two explanations for both property violations have disjoint semantics.

Proof. By soundness, $\forall \varepsilon_i \in \mathcal{E}(\mathcal{A}, \mathcal{O}_i, tr_i) : sem_{\mathcal{A}}(\varepsilon) \cap \mathsf{safeLang}_{\mathcal{A},\emptyset}(\mathcal{O}_i) = \emptyset$, $i = 1, 2$. Hence, from the state reached by any word $w \in sem_{\mathcal{A}}(\varepsilon_i)$, the violation of the property observed by $\mathcal{O}_i$ is inevitable. Equation (4.1) follows.

Now suppose that $\varepsilon \in \mathcal{E}(\mathcal{A}, \mathcal{O}_1, tr_1) \cap \mathcal{E}(\mathcal{A}, \mathcal{O}_2, tr_2)$. By Equation (4.1), it follows that $sem_{\mathcal{A}}(\varepsilon) = \emptyset$, which is in contradiction with weak completeness. Hence the second claim follows. □

Property 3 (Monotony of explanations) *Given a model $\mathcal{A}$, an observer $\mathcal{O}$, a couple $(\mathcal{E}, sem_{\mathcal{A}})$ is* monotonic *if for all $tr_i \in L^F(\mathcal{A}\|\mathcal{O}), \varepsilon_i \in \mathcal{E}(\mathcal{A}, \mathcal{O}, tr_i), i \in \{1, 2\}$, we have*

$$tr_2 \in sem_{\mathcal{A}}(\varepsilon_1) \implies sem_{\mathcal{A}}(\varepsilon_2) \subseteq sem_{\mathcal{A}}(\varepsilon_1)$$

Intuitively, $(\mathcal{E}, sem_{\mathcal{A}})$ is monotonic means that for a given notion of causality, $\mathcal{E}$ properly encodes the causes of a violation and $sem_{\mathcal{A}}$ properly decodes the explanations by returning only behaviors that share those encoded causes. In that context, $\varepsilon_1 \in \mathcal{E}(\mathcal{A}, \mathcal{O}, tr_1) \wedge tr_2 \in sem_{\mathcal{A}}(\varepsilon_1)$ means that the causes of tr_1 encoded in ε_1 are causes of tr_2. Moreover, $\mathsf{SEM}(\varepsilon_1) \subset \mathsf{SEM}(\varepsilon_2)$ means that the causes encoded in ε_2 are also encoded in ε_1. When the couple $(\mathcal{E}, sem_{\mathcal{A}})$ is monotonic, the explanation ε_1 is a causal generalization of the input trace tr_1, because $\forall tr \in sem_{\mathcal{A}}(\varepsilon_1)\ \forall \varepsilon \in \mathcal{E}(\mathcal{A}, \mathcal{O}, tr)$ ε encodes more causes (w.r.t. inclusion) than ε_1. Finally, the term monotonic comes from the monotony, with respect to the inclusion of semantics, of a sequence of explanations $(\varepsilon_i)_{i \in \mathbb{N}}$ such that $\forall i \in \mathbb{N}, \varepsilon^{i+1} \in \mathcal{E}(\mathcal{A}, \mathcal{O}, tr)$.

4.4 Sub-sequence Explanations

In this section, we base the explanation of a property violation on sub-sequences of traces. More specifically, we aim at retaining in the explanation only the events relevant to the violation of a safety property P.

A sub-sequence of a sequence of events $v \in \Sigma^*$ is a sequence of events $u \in \Sigma^*$ such that there exists a monotone function $\psi : [0, |u| - 1] \to [0, |v| - 1]$ such that $u = [v(\psi(i))]_{i \in [0, |u|-1]}$. We write $u \sqsubseteq v$ when u is a sub-sequence of v.

In the sequel we assume that the log $\mathcal{L}$ is consistent with some trace of a run that violates Q and then P.

Definition 19 (Sub-sequence explanation) *Given a behavioral model $\mathcal{A}$, an observer $\mathcal{O}_P$, and a trace $tr \in L^F(\mathcal{A}\|\mathcal{O}_P)$, the set of sub-sequence explanations for the violation of P is the language*

$$\boldsymbol{sse}(\mathcal{A}, \mathcal{O}_P, tr) = \min_{\sqsubseteq}\{w \in L^F(\mathcal{A}||\mathcal{P}) \mid w \sqsubseteq tr\}$$

We lift the definition of explanations to a log $\mathcal{L}$ as follows:

$$\boldsymbol{sse}(\mathcal{A}, \mathcal{O}_P, \mathcal{L}) = \bigcup_{tr \in L^F(\mathcal{A}\|\mathcal{L}\|\mathcal{O}_P)} \boldsymbol{sse}(\mathcal{A}, \mathcal{O}_P, tr)$$

Let us try and define semantics for a subsequence explanation ε that is sound and complete.

Clearly, soundness and completeness require that

$$L^F(\mathcal{A}\|\mathcal{O}\|\mathcal{L}) \subseteq sem_{\mathcal{A}}(\varepsilon) \subseteq L(\mathcal{A}) \setminus \mathsf{safeLang}_{\mathcal{A},\emptyset}(\mathcal{O})$$

Notice that both the lower and upper bound for $sem_{\mathcal{A}}(\varepsilon)$ are no feasible choices for the definition of $sem_{\mathcal{A}}(\varepsilon)$ since the definition of $sem_{\mathcal{A}}(\cdot)$ would be independent of its argument. Besides, the semantics should not depend on $\mathcal{O}$: intuitively, the explanation should itself contain the information of $\mathcal{O}$ that is causally relevant for the observed violation. Let us discuss possible options for the definition of $sem_{\mathcal{A}}$ on a concrete example.

Example 1 *We define the model $\mathcal{A}$ and observer $\mathcal{O}_P$ with their languages $trace(\mathcal{A}) = ab^* \cup ba^* \cup c^+$, and $L^F(\mathcal{A}||\mathcal{P}) = ab^* \cup ba^*$. By definition of subsequence explanation, the set of subsequence explanations is $\{"b","a"\}$ where $\forall w \in ab^* \cup ba^+, "a" \in \boldsymbol{sse}(\mathcal{A},\mathcal{O},w)$, and $\forall w \in ba^* \cup ab^+, "b" \in \boldsymbol{sse}(\mathcal{A},\mathcal{O},w)$.*

A subsequence explanation ε should at least occur in the traces of its semantics, that is $\forall \varepsilon\ sem_{\mathcal{A}}(\varepsilon) \subseteq \{w \in trace(\mathcal{A}) \mid \varepsilon \sqsubseteq w\}$. In the example, it implies $"a" \notin sem_{\mathcal{A}}("b") \wedge "b" \notin sem_{\mathcal{A}}("a")$.

Weak completeness and soundness respectively imply the equations 1 and 2.

$$ab^* \cup ba^+ \subseteq sem_{\mathcal{A}}("a") \wedge ba^* \cup ab^+ \subseteq sem_{\mathcal{A}}("b") \tag{1}$$

$$sem_{\mathcal{A}}("a") \subseteq ab^* \cup ba^* \wedge sem_{\mathcal{A}}("b") \subseteq ab^* \cup ba^* \tag{2}$$

Hence, there are four possible semantic functions (Figure 4.1) such that $\boldsymbol{sse}$ is weakly complete and sound with respect to the semantic functions. Among those semantics, only $sem^1_{\mathcal{A}}$ satisfies $\forall \varepsilon\ sem_{\mathcal{A}}(\varepsilon) \subseteq \{w \in trace(\mathcal{A}) \mid \varepsilon \sqsubseteq w\}$.

Explanation ε	$sem^1_{\mathcal{A}}$	$sem^2_{\mathcal{A}}$	$sem^3_{\mathcal{A}}$	$sem^4_{\mathcal{A}}$
$"a"$	$ab^* \cup ba^+$	$ab^* \cup ba^+$	$L^F(\mathcal{A}\|\|\mathcal{P})$	$L^F(\mathcal{A}\|\|\mathcal{P})$
$"b"$	$ba^* \cup ab^+$	$L^F(\mathcal{A}\|\|\mathcal{P})$	$ba^* \cup ab^+$	$L^F(\mathcal{A}\|\|\mathcal{P})$

Figure 4.1: Subsequence explanations and semantics

Furthermore, only the couple $(\boldsymbol{sse}, sem^4_{\mathcal{A}})$ is not monotonic because $\boldsymbol{sse}(\mathcal{A},\mathcal{O},"ab") = \{"a","b"\}$ and weak completeness implies that $"ab" \in sem_{\mathcal{A}}("a") \cap sem_{\mathcal{A}}("b")$. In that case, $(\boldsymbol{sse}, sem_{\mathcal{A}})$ is monotonic if $sem_{\mathcal{A}}("b") \subset sem_{\mathcal{A}}("a")$ (because $"ab" \in sem_{\mathcal{A}}("a")$) and $sem_{\mathcal{A}}("a") \subset sem_{\mathcal{A}}("b")$ (because $"ab" \in sem_{\mathcal{A}}("b")$).

Hence, for the model $\mathcal{A}$, none of the semantics of the subsequence explanations are pertinent.

This difficulty of defining semantics satisfying our requirements can be seen as a symptom of a more fundamental issue of sub-sequence explanations, namely, that they are merely "shortcuts" to the property violation whose run is incomparable, in general, with the actual run.

Besides the issues already pointed out, another shortcoming of sub-sequence explanations is the causal relevance of the events in the explanation. If we have a system that systematically performs an initialization then all the runs — and therefore also the sub-sequence explanation

— share a common prefix, even though it is debatable whether this initialization should be considered as a contributory cause. The notion of explanation we study in the next section avoids this downside by adopting a different characterization of contributory causes.

4.5 Choice Explanations

In this section, we present another kind of explanation where the contributory causes are *choices* of the system. For this purpose, we define the notion of *level of choice*, which can be seen as a generalization [23].

In this section, we require that the behavioral model $\mathcal{A} = \langle S, \Sigma, X, inv, \delta, s^0, v^0, S\rangle$ satisfies

$$\forall s, s' \in S \; s \neq s' \implies inv(s) \cap inv(s') = \emptyset \tag{H}$$

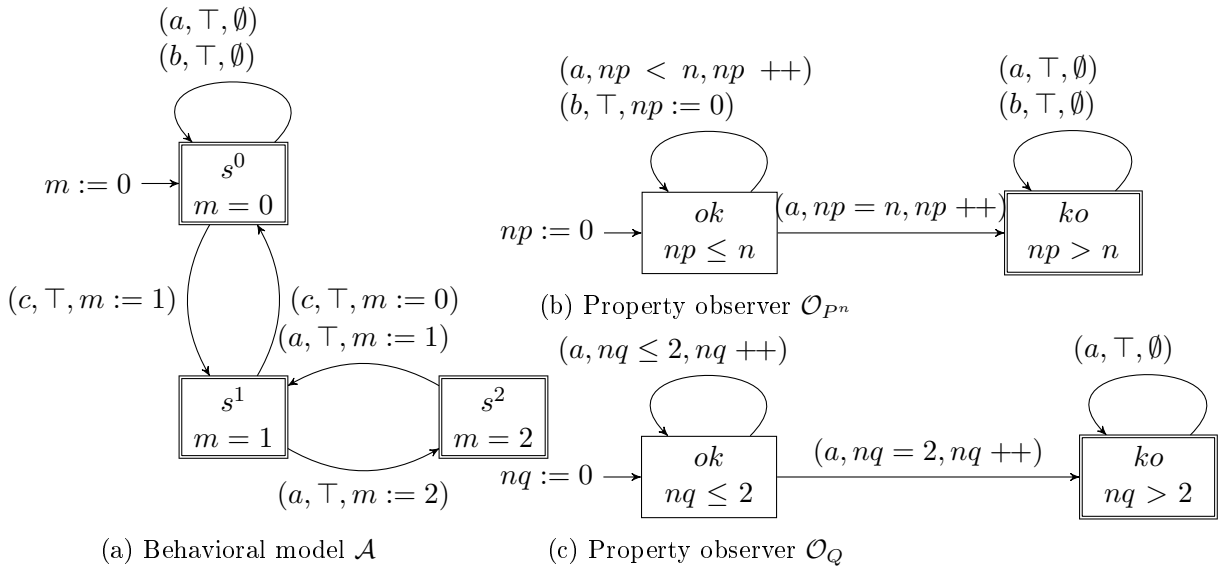
The motivation for such hypothesis is to construct extended automata $\mathcal{G}$ which when composed with $\mathcal{A}$ denotes specific behaviors of $\mathcal{A}||\mathcal{O}_P$, i.e. $[\mathcal{A}||\mathcal{G}] \subset [\mathcal{A}||\mathcal{O}_P]$ with a component wise inclusion. The guards and invariants are defined on the variables, and it is not possible to express constraints on the control state of $\mathcal{A}$ from the viewpoint of $\mathcal{G}$. When $\mathcal{A}$ satisfies the hypothesis H, there is no ambiguity on the control state of $\mathcal{A}$ when all the variables of $\mathcal{A}$ are variables of $\mathcal{G}$.

From a model $\mathcal{A}$ that does not satisfy the hypothesis, an equivalent model $\mathcal{A}'$ that satisfies the hypothesis H can be obtained. $\mathcal{A}'$ is the tuple $\langle S, \Sigma, X \cup \{b_s \mid s \in S\}, inv', \delta', s^0, v^0, S\rangle$ where the invariant inv' is defined with $\forall s \in S, inv'(s) = inv(s) \wedge b_s$, and the values of the new variables are updated if necessary with $\delta' = \{\langle s, e, s', g, f\rangle \mid \exists \langle s, e, s', g, f\rangle \in \delta : s = s'\} \cup \{\langle s, e, s', g, f \cup \{b_s := \mathit{false}, b_{s'} := \mathit{true}\}\rangle \mid \exists \langle s, e, s', g, f\rangle \in \delta : s \neq s'\}$. This transformation does not increase the state space of $\mathcal{A}$.

Example 2 (Running example) *We define the behavioral model and observers that are used to illustrate the notions in the section. The behavioral model $\mathcal{A}$ displayed in Figure 4.2a is a tuple $\langle S, \Sigma, \{m\}, inv, \delta, s^0, \{m \mapsto 0\}, S\rangle$ where the control states are $S = \{s^0, s^1, s^2\}$, there is a single variable m such that $\forall i \in \{0, 1, 2\}, inv(s_i) = (m = i)$, the set of events is $\Sigma = \{a, b, c\}$, the transitions δ are displayed in Figure 4.2a. All events are controllable, $\Sigma_c = \{a, b\}$.*

- *In the control state s^0, the system can execute the events a, b or transition to control state s^0 with the event c.*
- *In the control state s^1, the system cannot execute the event b, and has to either transition back to s^0 with c or execute the event a two times.*

We consider safety properties P^n, with respect to the events $\{a, b\}$ "not more than n consecutive occurrences of a". The property observer $\mathcal{O}_{P^n}$ of the safety property is displayed in Figure 4.2b and is the tuple $\mathcal{O}_{P^n} = \langle\{ok, ko\}, \{a, b\}, \{np\}, \{ok \mapsto np \leq n, ko \mapsto np > n\}, \delta_P, ok, \{np \mapsto 0\}, \{ko\}\rangle$ where the transitions δ_P displayed in Figure 4.2b. $\mathcal{O}_{P^n}$ shares the events a and b with $\mathcal{A}$ and



(a) Behavioral model $\mathcal{A}$

(b) Property observer $\mathcal{O}_{P^n}$

(c) Property observer $\mathcal{O}_Q$

Figure 4.2: Behavioral model $\mathcal{A}$ and property observers $\mathcal{O}_{P^n}$ and $\mathcal{O}_Q$

has a single sink state ko. The variable np is the number of consecutive occurrences of a and the execution of b resets np when the control state is ok.

We consider the contrastive property Q "not more than 2 occurrences of a", similarly to $\mathcal{O}_{P^n}$, the property observer $\mathcal{O}_Q$ (Figure 4.2c) tracks the number of occurrences of a with the variable nq.

The contrastive property Q is always violated before the safety property. In the rest of the example of the section the safety property is P^5 and the contrastive property is Q, and we have $\neg P^5 \implies \neg Q$.

4.5.1 Level of Choice

Definition 20 (Level of choice) *Given a LTS $G = \langle \mathcal{V}, \Sigma, \rightarrow, \nu_0, \mathcal{V}^F \rangle$ such that $\Sigma = \Sigma_c \uplus \Sigma_u$, we say that the function $\xi : \mathcal{V} \rightarrow (\mathbb{N} \cup \{\infty\})^2$ is the level of choice of $\mathcal{A}$ if:*

1. $\forall \nu \in \mathcal{V}^F : \xi(\nu) = \langle 0, 0 \rangle$
2. $\forall \nu \in \mathcal{V}$ *such that* ν *is not co-reachable,* $\xi(\nu) = \langle \infty, \infty \rangle$
3. *for all other states* $\nu \in \mathcal{V}$

$$\xi(\nu) = \langle \min\{\xi_c(\nu), \xi_u(\nu)\},\ \xi_{cu}(\nu) \rangle$$

with:

$$\xi_c(\nu) := \min^+\big(\{\xi_1(\nu') \mid \exists e \in \Sigma_c : \nu \xrightarrow{e} \nu'\}\big)$$
$$\xi_u(\nu) := \min\{\xi_1(\nu') \mid \exists e \in \Sigma_u : \nu \xrightarrow{e} \nu'\}$$
$$\xi_{cu}(\nu) := \min^+\big(\{\xi_2(\nu') \mid \exists e \in \Sigma : \nu \xrightarrow{e} \nu'\}\big)$$

where:

$$\min^+(G) = \begin{cases} \min(G) & \text{if } |G| \leq 1 \\ 1 + \min(G) & \text{otherwise} \end{cases}$$

and we set $\min \emptyset = \infty$.

4. *Maximality:* ξ *is in both components maximal among the functions fulfilling the preceding conditions, i.e.* $\forall \xi' : \mathcal{V} \to (\mathbb{N} \cup \{\infty\})^2$ *fulfilling the preceding conditions,* $\forall \nu \in \mathcal{V}, \xi'_1(\nu) \leq \xi_1(\nu) \wedge \xi'_2(\nu) \leq \xi_2(\nu)$.

We extend the definition of the level of choice onto extended automata, given a behavioral model $\mathcal{A} = \langle S, \Sigma, X, inv, \delta, s^0, v^0, S \rangle$ and a safety property P, let $\langle \mathcal{V}, \Sigma, \rightarrow, \nu_0, \mathcal{V}^F \rangle = [\mathcal{A}||\mathcal{O}_P]$. We define the level of choice of $\mathcal{A}||\mathcal{O}_P$ as the level of choice of the semantic LTS $[\mathcal{A}||\mathcal{O}_P]$.

The level of choice $\xi(\nu) = \langle \ell_1, \ell_2 \rangle$ has two components. ℓ_1 is qualified as the *controllable level of choice* as it indicates how many "bad choices" among controllable events still separate a state s from the states in $\mathcal{V}_F$ in the worst case, that is, for a hostile environment firing uncontrollable transitions. In contrast, ℓ_2 is qualified as the *closed-system level of choice* as it is the number of remaining "bad choices" without distinguishing controllable and uncontrollable events. ℓ_2 amounts to the levels of choice of [15]. Both are instrumental in constructing choice explanations. Intuitively, we will use ℓ_2 to identify uncontrollable events that challenge the system by moving it closer to the violation of P, and ℓ_1 to identify system events that do the same in spite of the existence of a "safer" alternative.

The runs of the behavioral model $\mathcal{A}$ violating the safety property end in an accepting state, i.e. in a state of $\mathcal{V}_F$. The first condition of Definition 20 requires that the level of choice of accepting states is 0, which means that no choices can be made in those states to violate the safety property because the violation already happened.

The second condition requires the level of choice of the states that are not co-reachable to be ∞; this means that in those states, no choice can be made to reach a violation.

The third condition is different for the controllable level of choice ℓ_1 and the closed-system level of choice ℓ_2. Given a state ν, let us define the "controllable postset" $\nu_c^\bullet$ and the "uncontrollable postset" $\nu_u^\bullet$ as the set of successors via controllable transitions and uncontrollable transitions respectively. Then:

- the closed-system level of choice $\xi_2(\nu)$ is determined considering indifferently successors in the controllable postset $\nu_c^\bullet$ and the uncontrollable postset $\nu_u^\bullet$. If the level of choice of the successors is uniform, then the level of choice of ν is the same as its successors because the decision about the out-going transition does not impact the level of choice. On the other hand, if the level of choice of the successors is not uniform, then the level of choice of ν is equal to the increment by one of the lowest level of choice in $\nu_c^\bullet \cup \nu_u^\bullet$. It indeed means that an impacting event has been issued by the system or the environment at ν.

- The controllable level of choice $\xi_1(\nu)$ only accounts for decisions of the system, independently of those of the environment. Indeed, we have $\xi_1 \leq \xi_u$ where $\xi_u(\nu)$ is the minimum

level of choice ξ_1 in $\nu_u^\bullet$. Therefore, the occurrence of an uncontrollable event does not decrease ξ_1. ξ_c is the counterpart to ξ_{cu}, it depends on $\nu_c^\bullet$ and ξ_1 instead of $\nu^\bullet$ and ξ_2. The controllable level of choice $\xi_1(\nu)$ is finally obtained as the minimum of both values.

The fourth condition requires the level of choice to identify all the bad choices of the system. For a non-maximal function f_2, that satisfies the other requirements of the level of choice, there is some state $\nu \in \mathcal{V}$ such that f_2 is uniform in $\nu^\bullet$ whereas ξ_2 is not. Such function f_2 is pessimistic compared to ξ_2 because it assumes that in ν, the system has no choices with respect to f_2. The following example illustrates that point.

Example 3 (The level of choice is maximal) *Let us consider the semantic LTS displayed in Figure 4.4a. All the states co-reachable with respect to $\mathcal{V}^F$, the function $f : \mathcal{V} \to (\mathbb{N} \cup \{\infty\})^2$ such that $\forall \nu \in \mathcal{V}\ f(\nu) = \langle 0, 0\rangle$ satisfies the other requirements of the level of choice. Furthermore, the cycle $\nu_2 \to \nu_4 \to \nu_3 \to \nu_2$ is disjoint with $\mathcal{V}^F$ and reachable by a controllable transition (solid edges in Figure 4.4a) from ν, the closed-system can choose between this cycle or $\mathcal{V}^F$ and therefore $\xi_2(\nu) > f(\nu) = 0$. Hence, the non-maximal function f does not identify the choice between the cycle, where the violation is avoided, and $\mathcal{V}^F$, where the violation already happened.*

Example 4 (Running example, level of choice) *To build an explanation, we compute the level of choice of $\mathcal{A}||\mathcal{O}_{P^5}$. The level of choice depends on $\mathcal{A}$ in order to include in the explanation some occurrence of c that decreases the capacity of the system to avoid the violation. Furthermore, not all occurrences of a are (bad) choices of the system, indeed when the control state of $\mathcal{A}$ is s^2, there are no other alternatives to the occurrence of a.*

The semantic LTS $[\mathcal{A}||\mathcal{O}_{P^5}]$ is given in Figure 4.3, in the labels, the tuple $\langle i, j\rangle$ denotes the state $\langle\langle s^i, s_p\rangle, \{m \mapsto i, np \mapsto j\}\rangle$ with $s_p = ok$ for non-accepting states and $s_p = ko$ for accepting states. For this running example, there are no uncontrollable events and we have $\xi_1 = \xi_2$. For a state ν, the red label l is the value of the level of choice $\xi(\nu) = \langle l, l\rangle$.

As expected, we notice that the occurrences of b increase the level of choice in non-accepting states. Furthermore, the occurrences of a when the control state is s^2 do not decrease the level of choice because there are no other alternatives. Switching from s^0 to s^1 when $np < 4$ decreases the level of choice because an occurrence of a is more forgiving in s^0 than in s^1. In the state $\langle\langle s^0, ok\rangle, \{m \mapsto 0, np \mapsto 1\}\rangle$ the worst outcome in terms of the level of choice is the occurrence of c (and not a).

The functions min and min^+ used in Definition 20 to define the level of choice have a monotonicity property that will be used to prove the soundness of the fixed-point algorithm that is proposed in Theorem 1 below to compute it.

Lemma 1 (Monotonicity of min and min^+) *For all finite set A and functions $f_1, f_2 \in A \to \mathbb{N} \cup \{\infty\}$, we have*

$$\begin{aligned}
&\forall \nu \in A, f_1(\nu) \leq f_2(\nu) \implies \\
&\quad min^+(\{f_1(\nu) \mid \nu \in A\}) \leq min^+(\{f_2(\nu) \mid \nu \in A\}) \wedge \\
&\quad \min(\{f_1(\nu) \mid \nu \in A\}) \leq \min(\{f_2(\nu) \mid \nu \in A\})
\end{aligned}$$

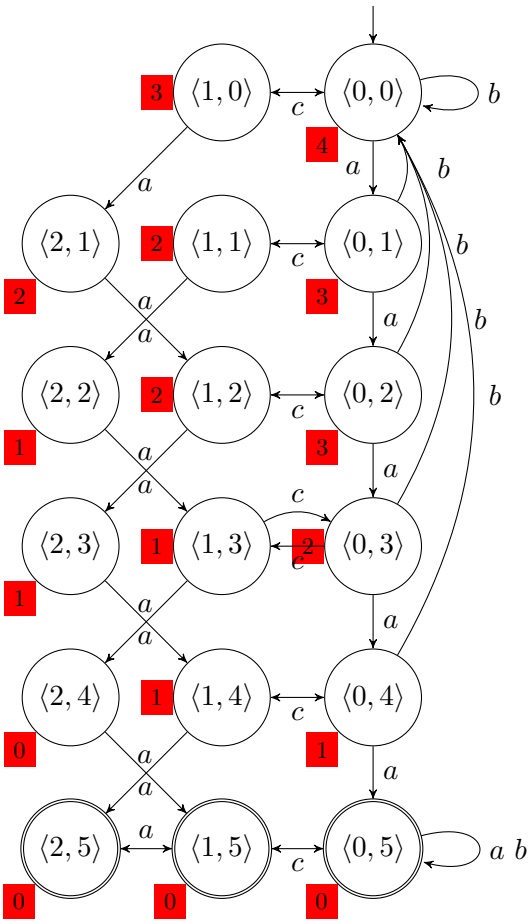


Figure 4.3: Semantic LTS $[\mathcal{A}||\mathcal{O}_{P^5}]$, the state label $\langle i, j\rangle$ denotes the state $\langle\langle s^i, s_p\rangle, \{m \mapsto i, np \mapsto j\}\rangle$ the level of choice ξ is displayed in the red labels, we display only one of the values of the couple ξ because $\Sigma = \Sigma_c$ and $\xi_1 = \xi_2$.

Proof. Let A be a finite set and $f_1, f_2 \in A \to \mathbb{N} \cup \{\infty\}$ such that $\forall \nu \in A, f_1(\nu) \leq f_2(\nu)$. Let $V_1, V_2 = \{f_1(\nu) \mid \nu \in A\}, \{f_2(\nu) \mid \nu \in A\}$.

- Case min

  $\exists \nu' \in A, \min V_2 = f_2(\nu')$ because A is finite. Because $\forall \nu \in A, f_1(\nu) \leq f_2(\nu)$, we have $\min V_2 \leq f_1(\nu') \leq f_2(\nu') = \min V_2$

- Case min^+

  If $|V_1| = 1$ or $|V_1| > 1$ and $|V_2| > 1$, we can use the previous equality on min and obtain :

  $min^+ V_1 = \min V_1 \leq \min V_2 \leq min^+ V_1$

  or

  $min^+ V_1 = 1 + \min V_1 \leq 1 + \min V_2 = min^+ V_2$.

  Else, we have $|V_1| > 1$ and $|V_2| = 1$ and therefore $\forall \nu \in A, f_1(\nu) \leq f_2(\nu) \wedge f_2(\nu) = \min V_2$, therefore $max(V_1) \leq \min V_2$. Furthermore, $|V_1| > 1$ implies $1 + \min V_1 \leq \max V_1$. Hence, $min^+ V_1 = 1 + \min V_1 \leq \max V_1 \leq \min V_2 = min^+ V_2$.

□

The following fixed-point algorithm is proposed to compute the level of choice.

Theorem 1 (Computing the level of choice) *Given a behavioral model $\mathcal{A}$ and a safety property observer $\mathcal{O}_P$, let $\langle \mathcal{V}, \Sigma, \rightarrow, \nu_0, \mathcal{V}^F \rangle = [\mathcal{A}||\mathcal{O}_P]$. We define the sequence of functions $\xi^i : \mathcal{V} \rightarrow (\mathbb{N} \cup \{\infty\})^2$, $i \geq 0$, as follows. Let*

$$\xi^0(\nu) = \begin{cases} \langle 0, 0 \rangle & \text{if } \nu \in \mathcal{V}^F \\ \langle \infty, \infty \rangle & \text{otherwise} \end{cases}$$
$$\xi^{i+1}(\nu) = \langle \min\left\{\xi^i_c(\nu), \xi^i_u(\nu)\right\}, \xi^i_{cu}(\nu) \rangle$$

where $\xi^i_c, \xi^i_u, \xi^i_{cu}$ are defined as in Definition 20. The level of choice defined in Definition 20 is given by $\inf_{i \geq 0} \xi^i$ (computed component-wise). Furthermore, the fixed-point is computed in a finite number of iterations, i.e. $\exists j \in \mathbb{N}$ such that $\xi^j = \inf_{i \geq 0} \xi^i$.

Proof. To prove convergence in a finite number of iterations, we use Lemma 1 to imply that each update decreases the functions component-wise $\forall i \in \mathbb{N}, \xi^{i+1}_1 \leq \xi^i_1 \wedge \xi^{i+1}_2 \leq \xi^i_2$. Therefore, $(\xi^i_1)_{i \in \mathbb{N}}$ and $(\xi^i_2)_{i \in \mathbb{N}}$ are decreasing sequences of functions. Furthermore, $(\mathbb{N} \cup \{\infty\}, <)$ is a well funded order and therefore for all state $\nu \in \mathcal{V}$ there exists an iteration $j \in \mathbb{N}$ such that the fixed-point is reached for ν, i.e. $\xi^j_1(\nu) = (\inf_{i \geq 0} \xi^i_1)(\nu) \wedge \xi^j_1(\nu) = (\inf_{i \geq 0} \xi^i_2)(\nu)$. Because there is a finite number of states, there exists an iteration (the maximum of j) such that the fixed-point is reached for all states. Hence, $\exists j \in \mathbb{N}$ such that $\xi^j = \inf_{i \geq 0} \xi^i$.

1. The satisfaction of the condition 1 of Definition 20 is by definition.

2. The satisfaction of the condition 2 of Definition 20 is obtained by proving by induction on $i \in \mathbb{N}$ that $\forall \nu \in \mathcal{V}$ such that ν is not co-reachable, $\xi^i(\nu) = \langle \infty, \infty \rangle$.

   $\xi^0(\nu) = \langle \infty, \infty \rangle$ because $\nu \notin \mathcal{V}^F$. All successors ν' of ν is also not co-reachable, therefore by hypothesis we have $\ell_{cu}(\nu, \xi) = \ell_c(\nu, \xi) = \ell_u(\nu, \xi) = \langle \infty, \infty \rangle$. Therefore $\xi^{i+1}(\nu) = \langle \infty, \infty \rangle$.

3. The satisfaction of the condition 3 of Definition 20 is implied by the equality $\xi^{j+1} = \xi^j$.

4. Condition 4 of Definition 20 requires ξ to be maximal. Toward contradiction, let us suppose that $\exists \xi' : \mathcal{V} \rightarrow (\mathbb{N} \cup \{\infty\})^2$ that satisfies the conditions 1,2,3 and 4 of Definition 20 and $\exists \nu \in \mathcal{V}, \xi'_1(\nu) > \xi_1(\nu)$.

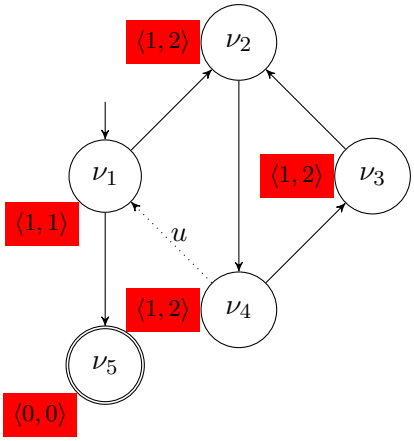
   Because ξ^i_1 is monotonous w.r.t. i and $\xi_1 \leq \xi^0_1$ on $\mathcal{V}$, let $i \in \mathbb{N}$ be the greatest integer such that $\forall \nu \in \mathcal{V}, \xi'_1(\nu) \leq \xi^i_1(\nu)$. i is maximal and therefore $\exists \nu_a \in \mathcal{V}, \xi'_1(\nu_a) > \xi^{i+1}_1(\nu_a)$.

   We apply Lemma 1 on $\{\nu' \in \mathcal{V} \mid \exists c \in \Sigma_c, \nu_a \xrightarrow{c} \nu'\}$ (resp. $\{\nu' \in \mathcal{V} \mid \exists u \in \Sigma_u, \nu_a \xrightarrow{u} \nu'\}$) , therefore $\xi'_c(\nu_a) \leq \xi^i_c(\nu_a)$ (resp. $\xi'_u(\nu_a) \leq \xi^i_u(\nu_a)$).

   Therefore,

   $\xi'(\nu_a) = min\{\xi'_c(\nu_a), \xi'_u(\nu_a)\} \leq \{\xi^i_c(\nu_a), \xi^i_u(\nu_a)\} = \xi^{i+1}_1(\nu_a)$

   It is a contradiction with $\exists \nu \in \mathcal{V}, \xi_1(\nu) > \xi^{i+1}_1(\nu)$.



(a) Semantic LTS $[\mathcal{G}]$

	ξ_1^i					ξ_2^i				
i	ν_1	ν_2	ν_3	ν_4	ν_5	ν_1	ν_2	ν_3	ν_4	ν_5
0	∞	∞	∞	∞	0	∞	∞	∞	∞	0
1	1	∞	∞	∞	0	1	∞	∞	∞	0
2	1	∞	∞	1	0	1	∞	∞	2	0
3	1	1	∞	1	0	1	2	∞	2	0
4	1	1	1	1	0	1	2	2	2	0

(b) Computation of ξ the level of choice of $\mathcal{G}$

Figure 4.4: Computation of the level of choice

Proving that ξ_2^* is maximal is a particular case of the previous proof where $\Sigma_u = \emptyset$ and $\xi_c = \xi_{cu}$.

□

Lemma 2 (Unicity of the level of choice) *The level of choice as defined in Definition 20 is unique.*

Proof. If two functions are maximal, then they are equal, which proves unicity.

$$\left[\forall \nu \in \mathcal{V}, \xi_1'(\nu) \leq \xi_1(\nu) \wedge \xi_2'(\nu) \leq \xi_2(\nu)\right] \wedge \left[\forall \nu \in \mathcal{V}, \xi_1(\nu) \leq \xi_1'(\nu) \wedge \xi_2(\nu) \leq \xi_2'(\nu)\right]$$
$$\implies \forall \nu \in \mathcal{V}, \xi_1'(\nu) = \xi_1(\nu) \wedge \xi_2'(\nu) = \xi_2(\nu)$$
$$\iff \xi = \xi'$$

□

Example 5 (Computation of the level of choice) *In this example, we illustrate the computation defined in Theorem 1 of the level of choice on the semantic LTS $[\mathcal{G}]$ in Figure 4.4a. Table 4.4b is the different iterations ξ_1^i, ξ_2^i for $i = 0, \ldots, 4$. The left side column is the iteration number i and the two other columns provide the values of ξ_1^i and ξ_2^i for the five states ν_1 to ν_5.*

4.5.2 Properties of the Level of Choice

Lemma 3 *Given a model $\mathcal{A}$ and an observer $\mathcal{O}_P$, let ξ be the level of choice function of $\mathcal{A}||\mathcal{O}_P$ and $\langle \mathcal{V}, \Sigma, \rightarrow, \nu^0, \mathcal{V}^F \rangle := [\mathcal{A}||\mathcal{O}_P]$, we define the set of states from where a violation of the property observed by $\mathcal{O}_P$ is inevitable as the smallest set $X \subset \mathcal{V}$ such that $\mathcal{V}^F \subset X$ and $\forall \nu \in \mathcal{V}, \nu^\bullet \neq \emptyset \wedge \nu^\bullet \subset X \implies \nu \in X$*

From any state $\nu \in \mathcal{V}$ with $\xi_2(\nu) = 0$, a violation of the property observed by $\mathcal{O}_P$ is inevitable.

Proof. We prove by induction on $i \in \mathbb{N}$ that $\forall \nu \in \mathcal{V}, \xi_2^i(\nu) = 0 \implies$ a violation of the property observed by $\mathcal{O}_P$ is inevitable from ν.

For $i = 0$, let $\nu \in \mathcal{V}$ such that $\xi_2^0(\nu) = 0$ it implies by definition of ξ^0 that $\nu \in \mathcal{V}^F$ and therefore by definition of $\mathcal{V}^F$, ν is accepting for the property observer.

For $i \in N$, let $\nu \in \mathcal{V}$ such that $\xi_2^{i+1}(\nu) = 0 \wedge \xi_2^i(\nu) \neq 0$. By definition of ξ^{i+1} we have $\xi_2^{i+1}(\nu) = \min^+\big(\{\xi_2^i(\nu') \mid \exists e \in \Sigma : \nu \xrightarrow{e} \nu'\}\big) = 0 \implies \forall \nu' \in \nu^\bullet, \xi_2^i(\nu') = 0$

Therefore $\forall \nu' \in \nu^\bullet, \xi^i(\nu') = 0 \wedge \nu^\bullet \neq \emptyset$. We use the induction hypothesis and all runs lead to states where the violations of P is inevitable. Hence a violation of the property observed by $\mathcal{O}_P$ is inevitable from ν. □

Intuitively, when the system is in a state ν in which the level of choice is $\xi(\nu) = \langle 0, 0\rangle$, it means that the violation is inevitable even if the environment is collaborative.

Proposition 3 *Given a model $\mathcal{A}$ and an observer $\mathcal{O}_P$, let ξ be the level of choice of $\mathcal{A}||\mathcal{O}_P$ and $\langle \mathcal{V}, \Sigma, \to, \nu^0, \mathcal{V}^F\rangle := [\mathcal{A}||\mathcal{O}_P]$, we have that:*

1. $\forall \nu \in \mathcal{V} : \xi_1(\nu) \leq \xi_2(\nu)$ *and, if* $\Sigma_u = \emptyset$, $\xi_1(\nu) = \xi_2(\nu)$;
2. $\forall \nu, \nu' \in \mathcal{V}\ \forall e \in \Sigma_u : \big(\nu \xrightarrow{e} \nu' \implies \xi_1(\nu') \geq \xi_1(\nu)\big)$.
3. $\forall \nu, \nu' \in \mathcal{V}\ \forall e \in \Sigma\ \forall i \in \{1, 2\} : \big(\nu \xrightarrow{e} \nu' \implies \xi_i(\nu) - \xi_i(\nu') \leq 1\big)$.
4. *If $\xi(\nu) = \langle \ell_1, \ell_2\rangle$ then any run from ν to $\mathcal{V}^F$ encompasses at least ℓ_2 transitions, at least ℓ_1 of which are controllable.*

Proof.

1. By induction on the iterations: the claim holds for ξ^0. Induction step: whenever $\xi_1^i(\nu) \leq \xi_2^i(\nu)$ then $\min\{\xi_c^i(\nu), \xi_u^i(\nu)\} \leq \xi_{cu}^i(\nu)$, hence $\xi_1^{i+1}(\nu) \leq \xi_2^{i+1}(\nu)$. For $\Sigma_u = \emptyset$, $\xi_1(\nu) = \xi_2(\nu)$ follows by equality of $\xi_c^i(\nu)$ and $\xi_{cu}^i(\nu)$.
2. We have by Definition 20 that $\xi_1(\nu) = \min\{\xi_c(\nu), \xi_u(\nu)\}$ and $\xi_u(\nu) = \min\big\{\xi_1(\nu'') \mid \exists e \in \Sigma_u : \nu \xrightarrow{e} \nu''\} \leq \xi_1(\nu')$, hence $\xi_1(\nu) \leq \xi_1(\nu')$.
3. We have by Definition 20 that $\xi_1(\nu) = \min\{\xi_c(\nu), \xi_u(\nu)\} \leq \min^+\big(\{\xi_1(\nu') \mid \exists e \in \Sigma : \nu \xrightarrow{e} \nu'\}\big) \leq 1 + \min\big(\{\xi_1(\nu') \mid \exists e \in \Sigma : \nu \xrightarrow{e} \nu'\}\big) \leq 1 + \xi_1(\nu')$, hence $\xi_1(\nu') - \xi_1(\nu) \geq -1$. $\xi_2(\nu) = \xi_{cu}^i(\nu) \leq 1 + \min\big(\{\xi_2(\nu') \mid \exists e \in \Sigma : \nu \xrightarrow{e} \nu'\}\big) \leq 1 + \xi_2(\nu')$, hence $\xi_2(\nu') - \xi_2(\nu) \geq -1$.
4. The first part of the claim follows directly from item 3. The second part then follows with item 2.

□

4.5.3 Effective Choice Transitions and Safe Alternatives

We say a state ν of $[\mathcal{A}\|\mathcal{O}_P]$ is a *choice state* if two successors of ν have different levels of choice. For an outgoing transition $\nu \xrightarrow{e} \nu'$ from ν, the evolution of the level of choice from $\xi(\nu)$ to $\xi(\nu')$ has a different interpretation depending on whether one considers the controllable level of choice or the closed-system level of choice:

- Considering the controllable level of choice, if $\xi_1(\nu') = \xi_1(\nu) - 1$ the transition can be seen as an *erroneous action* from the system that drives it closer to the violation. If $\xi_1(\nu') = \xi_1(\nu)$, it means that the transition does not involve any action from the system that drives it

closer to the violation although it may involve *bad actions* from the environment that do so. If $\xi_1(\nu') > \xi_1(\nu)$ then the transition repairs past erroneous actions from the system although bad actions from the environment may remain unrepaired. Those transitions are relevant to understand the actual erroneous actions leading to the violation. If $\xi_1(\nu') = \infty$, it means that if the rest of the run drives to the violation, responsibility must be endorsed by the environment.

- Considering the closed-system level of choice, if $\xi_2(\nu') = \xi_2(\nu) - 1$ the transition involves an action, either erroneous when from the system or bad when from the environment, that drives the system closer to the violation. If $\xi_2(\nu') = \xi_2(\nu)$ then the transition can be seen as a delay, it means that the violation is only delayed until reaching the next choice state at the same level. If $\xi_2(\nu') > \xi_2(\nu)$ then the transition repairs past erroneous or bad actions. The extreme case is when $\xi_2(\nu') = \infty$, it means that the rest of the run is safe.

In a run, the impact of an event depends on future events. If there is a loop in the run, then the erroneous or bad actions present in the loop are compensated by transitions that increase the level of choice and they might not be related to the violation at the end of the run.

The following notion characterizes the transitions of a run that are not compensated.

Definition 21 (Effective choice transitions) *Given a model $\mathcal{A}$, an observer $\mathcal{O}_P$ and a run $r = \nu_0 \xrightarrow{e_1} \nu_1 \dots \xrightarrow{e_n} \nu_n \in run(\mathcal{A}||\mathcal{P})$, let $\langle \mathcal{V}, \Sigma, \rightarrow, \nu^0, \mathcal{V}^F \rangle := [\mathcal{A}||\mathcal{O}_P]$ and ξ be the level of choice of $\mathcal{A}||\mathcal{O}_P$. We define two sets of transitions $\mathit{ECT}_{\xi_1}(r)$ and $\mathit{ECT}_{\xi_2}(r)$ in r such that $\forall k \in \{1,2\}$:*

$\forall i \in [1,n], \nu_{i-1} \xrightarrow{e_i} \nu_i \in \mathit{ECT}_{\xi_k}(r) \iff \forall j \in [i,n], \xi_k(\nu_{i-1}) > \xi_k(\nu_j)$

The set $\mathit{ECT}_{\xi}(r) = \mathit{ECT}_{\xi_1}(r) \cup \mathit{ECT}_{\xi_2}(r)$ is the set of effective choice transitions in r.

An effective choice transition of a run is a transition that decreases a level of choice, and that decrease is not compensated later in the suffix following the transition. A transition in $\mathsf{ECT}_{\xi_1}(r)$ is a bad choice of the system, as it decreases the ability of the system to avoid the violation of the safety property. A transition in $\mathsf{ECT}_{\xi_2}(r)$ is an error from the system or a bad action from the environment that drives effectively the system closer to the violation. The following notion defines an alternative choice of the system that does not decrease the level of choice. With the existence of a controllable safe alternative, an effective choice transition in $\mathsf{ECT}_{\xi_1}(r)$ is indeed a choice of the system because it can be avoided via another controllable transition.

Definition 22 (Safe alternatives) *Given a model $\mathcal{A}$, an observer $\mathcal{O}_P$ and a run $r \in run(\mathcal{A}||\mathcal{P})$, let ξ be the level of choice of $\mathcal{A}||\mathcal{O}_P$ and $\langle \mathcal{V}, \Sigma, \rightarrow, \nu^0, \mathcal{V}^F \rangle := [\mathcal{A}||\mathcal{O}_P]$, a couple $(\nu, e_1) \in \mathcal{V} \times \Sigma_c$ is a safe alternative if:*

- $\exists e_2 \in \Sigma, \nu_2 \in \mathcal{V}, \nu \xrightarrow{e_2} \nu_2 \in \mathit{ECT}_{\xi_1}(r)$
- *and* $\exists \nu_1 \in \mathcal{V}, \nu \xrightarrow{e_1} \nu_1 \wedge \xi_1(\nu) \leq \xi_1(\nu_1)$

Let $\mathit{ALT}_{\xi}(r)$ be the set of all safe alternatives of the run r.

A controllable transition that does not decrease ξ_1 is a safe alternative if from the same origin there is an effective choice transition that effectively decreases ξ_1 in the run r. The safe alternatives are defined with respect to the controllable label of choice ξ_1 only. An effective choice transition of the closed-system $\mathsf{ECT}_{\xi_2}(r)$ can be uncontrollable and cannot be always avoided via a controllable alternative. The following theorem states that a safe alternative always exists at the origin of an effective choice transition in $\mathsf{ECT}_{\xi_1}(r)$.

Theorem 2 *Existence of a safe alternative Given a model $\mathcal{A}$, an observer $\mathcal{O}_P$ and a run $r = \nu_0 \xrightarrow{e_1} \nu_1 \dots \xrightarrow{e_n} \nu_n \in run(\mathcal{A}||\mathcal{P})$, let ξ be the level of choice of $\mathcal{A}||\mathcal{O}_P$,*

$\forall \nu \xrightarrow{e} \nu' \in \mathit{ECT}_{\xi_1}(r), \exists e' \in \Sigma_c$ *such that* $(\nu, e') \in \mathit{ALT}_{\xi}(r)$

Proof. Let $\nu \xrightarrow{e} \nu' \in \mathsf{ECT}_{\xi_1}(r)$, we have $\xi_1(\nu) > \xi_1(\nu')$ by definition of $\mathsf{ECT}_{\xi_1}(r)$, and therefore with proposition 3, we have $\xi_1(\nu) = 1 + \xi_1(\nu') = 1 + \min\left\{\xi_1(\nu'') \mid \exists e' \in \Sigma_c : \nu \xrightarrow{e} \nu''\right\} \wedge |\left\{\xi_1(\nu'') \mid \exists e' \in \Sigma_c : \nu \xrightarrow{e} \nu''\right\}| > 1$ by definition of ξ_1. Hence $\exists e' \in \Sigma_c : \nu \xrightarrow{e'} \nu'', \xi_1(\nu) \leq \xi_1(\nu'')$ and $(\nu, e') \in \mathsf{ALT}_{\xi}(r)$.

□

Example 6 (Running example, effective choice transitions and safe alternatives) *Let us consider the trace $tr = a\ b\ a\ a\ c\ c\ a\ c\ c\ c\ a\ a$ in $L^F(\mathcal{A}||\mathcal{O}_P)$ and the run $r = \nu_0 \xrightarrow{e_1} \nu_1 \dots \nu_{11} \xrightarrow{e_{12}} \nu_{12} \in run^F(\mathcal{A}||\mathcal{O}_{P^5})$ such that $e_1 \dots e_{12} = tr$.*

The valuations of the variables m and np and the level of choice of the states ν_i for $i = 0, \dots, 12$ are displayed in Figure 4.6.

Figure 4.5a is the restriction of the semantic LTS $[\mathcal{A}||\mathcal{O}_P]$ to the transitions of r.

The set of effective choice transitions (Figure 4.5f) for the run r is

$\mathit{ECT}_{\xi}(r) = \mathit{ECT}_{\xi_1}(r) = \mathit{ECT}_{\xi_2}(r) = \{t_1, t_2, t_3, t_4\}$ where $t_1 = \nu_2 \xrightarrow{e_3} \nu_3, t_2 = \nu_6 \xrightarrow{e_7} \nu_7, t_3 = \nu_9 \xrightarrow{10} \nu_{10}$, and $t_4 = \nu_{10} \xrightarrow{e_{11}} \nu_{11}$.

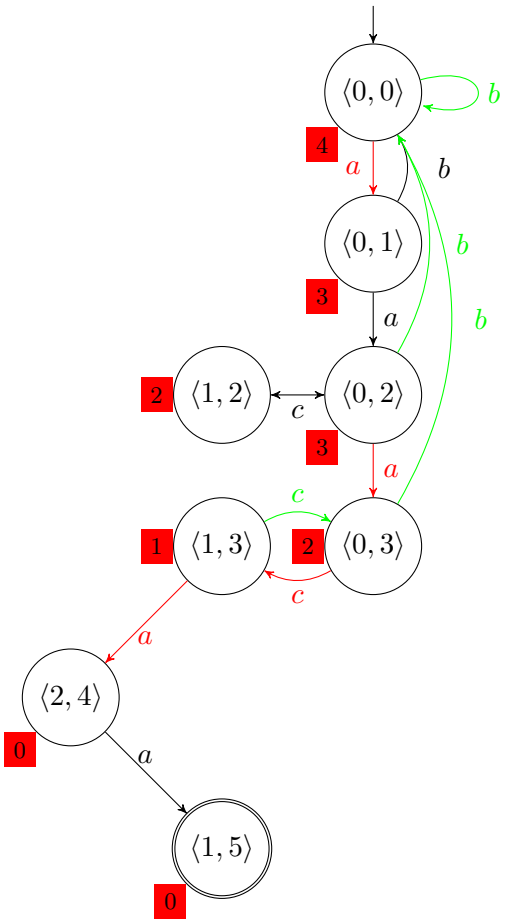
The initial (and maximal) value of the level of choice is 4, therefore there are 4 effective choice transitions that decrease effectively the level of choice from 4 to 0. The last effective choice transitions $t_4 = \nu_{10} \xrightarrow{e_{11}} \nu_{11}$ leads the system into $\nu_{11} = \langle 2, 4\rangle$, we can see in the semantic LTS Figure 4.3 that from $\langle 2, 4\rangle$ the violation of P^5 is inevitable (Lemma 3).

For each of the effective choice transitions of r (displayed in red in Figure 4.5a) there is a safe alternative (displayed in green in Figure 4.5a).

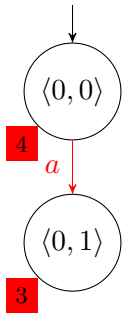
4.5.4 Ingredients of Choice Explanations

A choice explanation takes the level of choice into account by retaining effective choice transitions of a given run r that violates property P. It must also retain what happened in between two effective choice transitions. In addition, we want the explanations to be contrastive with respect to the property Q, which means that they contain only causes in the suffix of the explained traces for which the violation of Q is inevitable.

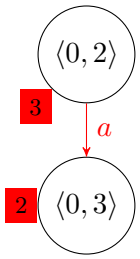
For a run r, $\mathsf{ECT}_{\xi}(r)$ is the set of transitions that lead effectively the system towards the violation of the safety property P by decreasing definitively ξ_1 or ξ_2 in its suffix. In order to build the explanation, we order the set of effective choice transitions. The observed order



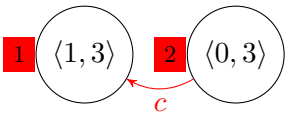
(a) Restriction of $[\mathcal{A}||\mathcal{O}_P]$ on the transitions and states of r. The effective choice transitions are displayed in red. The safe alternatives are displayed in green.



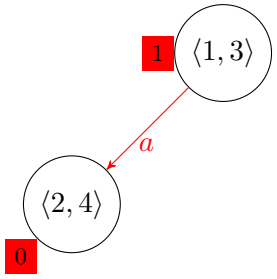
(b) t_1



(c) t_2



(d) t_3



(e) t_4

(f) Effective choice transitions $\mathsf{ECT}_\xi(r)$

Figure 4.5: Run r and $\mathsf{ECT}_\xi(r)$

of effective choice transition in r is total and too precise. We define a relaxed order " $<$ " that does not depend on the observation but only on the level of choice. This order allows potential concurrency between effective choice transitions and only orders two transitions when it is necessary with respect to the membership in $\mathsf{ECT}_{\xi_1}(r)$ and $\mathsf{ECT}_{\xi_2}(r)$.

Choice explanations are hence in the form of a partially ordered set of effective choice transitions, along with a function ϕ that maps ordered tuples in $<$ to what is "in between". More formally, an explanation takes the form

$$\langle \pi, \langle E, < \rangle, \phi \rangle$$

where $\langle E, < \rangle$ is a strict partial order (SPO) of transitions as defined in Definition 23 below, ϕ maps ordered tuples in $<$ on what is "in between", and π accounts for the prefix of the explained traces from which the contrastive property Q can always be satisfied.

i	0	1	2	3	4	5	6	7	8	9	10	11	12
e_i		a	b	a	a	c	c	a	c	c	c	a	a
ν_i	ν_0	ν_1	ν_2	ν_3	ν_4	ν_5	ν_6	ν_7	ν_8	ν_9	ν_{10}	ν_{11}	ν_{12}
m	0	0	0	0	0	1	0	0	1	0	1	2	1
np	0	1	0	1	2	2	2	3	3	3	3	4	5
$\xi(\nu_i)$	4	3	4	3	2	2	3	2	1	2	1	0	0

Figure 4.6: Run $r \in run^F \mathcal{A}||\mathcal{O}_{P^5}$

Definition 23 (SPO) *An relation is a couple $\langle E, < \rangle$ such that $< \subseteq A \times A$. For $e_1, e_2 \in E$, we write $e_1 < e_2 \iff \langle e_1, e_2 \rangle \in <$.*

A relation $\langle E, < \rangle$ is a strict partial order (SPO), if it satisfies the following conditions:

- *irreflexivity:* $\forall e \in E \ \neg(e < e)$
- *transitivity:* $\forall e_1, e_2, e_3 \in E \ e_1 < e_2 \wedge e_2 < e_3 \implies e_1 < e_3$
- *asymmetry:* $\forall e_1, e_2 \in E \ e_1 < e_2 \implies \neg(e_2 < e_1)$

An effective choice transition $t \in E$ represents a *contributory cause* within the explanation $\langle \pi, \langle E, < \rangle, \phi \rangle$.

Each cause is encoded in the explanation. Intuitively a cause is an event's occurrence and a context that leads the system closer to the violation of the safety property. One cause alone may not be sufficient. This is why we encode an ordered set of causes considering concurrency and precedence.

We chose strict partial order to encode the pattern of contributory causes. Given two contributory causes t_1 and t_2, $t_1 < t_2$ means that t_1 happened before t_2. Irreflexivity is motivated by the fact that if a cause happened before itself, then it may happen an unbounded number of times and therefore delay the violation of the safety property. It is therefore not consistent with the idea that a cause leads to the violation. Asymmetry prevents causes from looping. Transitivity allows a compact representation of the ordering.

Definition 24 (Operations on relations : max, Union) *Given an SPO $\langle E, < \rangle$, we define $max(\langle E, < \rangle) = \{t \in E \mid \forall t' \in E, \neg(t < t')\}$ is the set of maximal elements in E.*

The union of two relations $\langle E_1, \leq_1 \rangle$ and $\langle E_2, \leq_2 \rangle$ is the relation $\langle E_1, \leq_1 \rangle \cup \langle E_2, \leq_2 \rangle = \langle A \cup B, <_A \cup <_B \rangle$.

SPOC

Definition 25 (Strict partial order of choice (SPOC)) *Given a model $\mathcal{A}$, an observer $\mathcal{O}_P$ and a run $r \in run(\mathcal{A}||\mathcal{P})$, let ξ be the level of choice of $\mathcal{A}||\mathcal{O}_P$ and $\langle \mathcal{V}, \Sigma, \rightarrow, \nu^0, \mathcal{V}^F \rangle := [\mathcal{A}||\mathcal{O}_P]$,*

we define $\mathsf{SPOC}_\xi(r)$ as the tuple $\langle \mathsf{ECT}_\xi(r), < \rangle$ where:

- $<_1 = \{\langle \nu \xrightarrow{e} \nu', \nu_1 \xrightarrow{e_1} \nu_1' \rangle \in \mathsf{ECT}_\xi(r) \times \mathsf{ECT}_{\xi_1}(r) \mid max(\xi_1(\nu), \xi_1(\nu')) \geq \xi_1(\nu_1) \wedge \nu \xrightarrow{e} \nu' \neq \nu_1 \xrightarrow{e_1} \nu_1'\}$

- $<_2 = \{\langle \nu \xrightarrow{e} \nu', \nu_2 \xrightarrow{e_2} \nu_2' \rangle \in \mathit{ECT}_\xi(r) \times \mathit{ECT}_{\xi_2}(r) \mid max(\xi_2(\nu), \xi_2(\nu')) \geq \xi_2(\nu_2) \wedge \nu \xrightarrow{e} \nu' \neq \nu_2 \xrightarrow{e_2} \nu_2'\}$
- $<$ *is the transitive closure of* $(<_1 \cup <_2)$

For a run r of $\mathcal{A}||\mathcal{O}_P$, $\mathsf{SPOC}_\xi(r)$ is an ordered set of effective choice transitions of r. The order of $\mathsf{SPOC}_\xi(r)$ captures the constraints on the level of choice ξ. For $t, t' \in \mathsf{ECT}_\xi(r)$, $t < t'$ implies that if t happens after the last occurrence of t' in some run r', then either $\mathsf{ECT}_{\xi_1}(r) \neq \mathsf{ECT}_{\xi_1}(r')$ or $\mathsf{ECT}_{\xi_2}(r) \neq \mathsf{ECT}_{\xi_2}(r')$.

$\mathsf{SPOC}_\xi(r)$ is a partial ordered set of contributory causes of the violation r of the safety property. The next lemma states that the observed total order is always consistent with the SPOC_ξ order. In other words, SPOC_ξ is a generalization of the observed order and different traces may have the same SPOC_ξ.

Lemma 4 *Given a model $\mathcal{A}$ and an observer $\mathcal{O}_P$, let ξ be the level of choice of $\mathcal{A}||\mathcal{O}_P$ and $\langle \mathcal{V}, \Sigma, \rightarrow, \nu^0, \mathcal{V}^F \rangle := [\mathcal{A}||\mathcal{O}_P]$,*

we have $\forall r \in run(\mathcal{A}||\mathcal{O}_P)\ \langle t, t' \rangle \in \mathit{SPOC}_\xi(r) \implies Last(t, r) < Last(t', r)$

where $Last(\nu \xrightarrow{e} \nu', \nu_0 \xrightarrow{e_0} \nu_1 \ldots \xrightarrow{e_n} \nu_n) = \max\{i \in [1, n] \mid \nu_{i-1} \xrightarrow{e_n} \nu_i = \nu \xrightarrow{e} \nu'\}$

Proof. Let $r \in run(\mathcal{A}||\mathcal{P})$, we use the inductive definition of the transitive closure $<$ of $(<_1 \cup <_2)$, with $<_1, <_2$ defined in Definition 25 :

$$<^0 = <_1 \cup <_2$$
$$\forall i \in \mathbb{N}, <^{i+1} = <^i \cup \{\langle t, t'' \rangle \mid \exists t' \in E, \langle t, t' \rangle \in <^i \wedge \langle t', t'' \rangle \in <^i\}$$

We prove by induction on $i \in \mathbb{N}$ that $\langle t, t' \rangle \in <^i \implies Last(t, r) < Last(t', r)$

For $<^0 = <_1 \cup <_2$, $t, t' \in \mathsf{ECT}_\xi(r)$, if we have $t <_1 t'$ (resp. $t <_2 t'$) then any occurrence of t after the last occurrence of t' implies that $t' \notin \mathsf{ECT}_{\xi_1}(r)$ (resp $t' \notin \mathsf{ECT}_{\xi_2}(r)$), hence the property is satisfied.

For the induction step, if we have $t <^{i+1} t' \wedge \exists t'' \in \mathsf{ECT}_\xi(r), t <^i t'' \wedge t'' <^i t'$ we have :

$t'' <^i t'$ implies, with the induction hypothesis, that there is no occurrence of t'' after the last occurrence of t' and therefore the last occurrence of t'' happens before t'.

$t <^i t''$ implies, with the induction hypothesis, that there is no occurrence of t after the last occurrence of t'', hence there is no occurrence of t after the last occurrence of t'. □

Proposition 4 *Given a model $\mathcal{A}$ and an observer $\mathcal{O}_P$, let ξ be the level of choice of $\mathcal{A}||\mathcal{O}_P$. For all run $r \in run(\mathcal{A}||\mathcal{P})$, the relation $\mathit{SPOC}_\xi(r)$ is an SPO.*

Proof. $\forall t_1, t_2 \in E$, such that $t_1 < t_2$, we have $t_1 < t_2 \implies Last(t_1, r) < Last(t_2, r)$ (Lemma 4). Therefore, $t_1 < t_2 \implies \neg(t_2 < t_1)$.

Irreflexivity is a subcase of the proof above with $t_1 = t_2$.

Finally, in Definition 25, $<$ is the transitive closure of $(<_1 \cup <_2)$ and is therefore transitive.

□

The next lemma states that if r violates the safety property then its last observed effective choice transition is maximal in SPOC_ξ and leads to $\xi = \langle 0,0\rangle$.

Lemma 5 *Given a model $\mathcal{A}$ and an observer $\mathcal{O}_P$, let ξ be the level of choice of $\mathcal{A}||\mathcal{O}_P$, $\forall r \in run^F(\mathcal{A}||\mathcal{O}_P)$ such that $\mathsf{ECT}_\xi(r) \neq \emptyset$, $\mathsf{SPOC}_\xi(r)$ has a unique maximal element $\nu \xrightarrow{e} \nu'$ such that $\xi(\nu') = \langle 0,0\rangle$.*

Proof. Let $r = \nu_0 \xrightarrow{e_1} \nu_1 ... \nu_{n-1} \xrightarrow{e_n} \nu_n \in run^F(\mathcal{A}||\mathcal{O}_P)$ and $i = min\{j \in [0,n] \mid \xi(\nu_j) = \langle 0,0\rangle\}$.

i is well defined because $r \in run^F(\mathcal{A}||\mathcal{O}_P) \implies \nu_n \in \neg P \implies \xi(\nu_n) = \langle 0,0\rangle$

Furthermore $\mathsf{ECT}_\xi(r) \neq \emptyset$ and i minimal implies $\xi(\nu_{i-1}) > \langle 0,0\rangle$.

Proposition 3 implies that $\xi(\nu_{i-1})$ is either $\langle 0,1\rangle$ or $\langle 1,1\rangle$, and the set of state where $\xi_2 = 0$ is closed under the transition relation, hence $\nu_{i-1} \xrightarrow{e_i} \nu_i \in \mathsf{ECT}_{\xi_2}(r)$.

$\nu_{i-1} \xrightarrow{e_i} \nu_i$ is maximal w.r.t. $<_2$ because $\forall \nu \in \mathcal{V}, \xi_2(\nu) \geq 0 = \xi_2(\nu_i)$.

$\nu_{i-1} \xrightarrow{e_i} \nu_i$ is maximal w.r.t. $<_1 \cup <_2$ because either it is also maximal in $<_1$ whenever $\nu_{i-1} \xrightarrow{e_i} \nu_i \in \mathsf{ECT}_{\xi_1}(r)$ or it is not in the relation $<_2$.

To conclude the transitive closure does not change the maximal element of the relation and $\nu_{i-1} \xrightarrow{e_i} \nu_i \in \mathsf{ECT}_{\xi_1}(r)$ is the unique maximal element in $\mathsf{SPOC}_\xi(r)$.

□

This lemma is crucial because from a state in ν' such that $\xi(\nu') = \langle 0,0\rangle$, Lemma 3 states that a violation of the property observed by $\mathcal{O}_P$ is inevitable. The maximal effective choice transition in $\mathsf{SPOC}_\xi(r)$ is the last effective choice transition in the observed order of occurrence (Lemma 4).

Non choice extended automaton

This context of what happens in between two effective choice transitions, which is captured by a *non-choice extended automaton* as defined subsequently.

The following definition denotes, for any effective choice transition t, the upper bounds $\xi_1^{ec(r)}(t)$ (resp. $\xi_2^{ec(r)}$) on ξ_1 (resp. ξ_2). For all suffix of the run r after the occurrence of t which contains only states ν satisfying $\xi_1(\nu) \leq \xi_1^{ec(r)}(t) \wedge \xi_2(\nu) \leq \xi_2^{ec(r)}(t)$, the effective transitions preceding t remain effective.

Definition 26 (Upper bounds on ξ) *Given a model $\mathcal{A}$, a property observer $\mathcal{O}_P$ and a run $r \in run(\mathcal{A}||\mathcal{O}_P)$, let $\langle \mathcal{V}, \Sigma, \rightarrow, \nu^0, \mathcal{V}^F\rangle$ be the semantic LTS of $\mathcal{A}||\mathcal{O}_P$, let ξ be the level of choice of $\mathcal{A}||\mathcal{P}$ and $\langle \mathsf{ECT}_\xi(r), <\rangle = \mathsf{SPOC}_\xi(r)$, let us define $\xi_1^{ec(r)}, \xi_2^{ec(r)} : (\rightarrow) \rightarrow (\mathbb{N} \cup \{\infty\})$ such that:*

$\forall t \in \mathsf{ECT}_\xi(r)$*:*

$\xi_1^{ec(r)}(t) = \min\{\xi_1(\nu') \mid \nu \xrightarrow{e} \nu' \in pre(t) \cap \mathsf{ECT}_{\xi_1}(r)\}$

$\xi_2^{ec(r)}(t) = \min\{\xi_2(\nu') \mid \nu \xrightarrow{e} \nu' \in pre(t) \cap \mathsf{ECT}_{\xi_2}(r)\}$

where $pre(t) = \{t' \in \rightarrow \mid t' < t\} \cup \{t\}$

$\forall t \in (\rightarrow \setminus \mathsf{ECT}_\xi(r))$*:*

$\xi_1^{ec(r)}(t) = \infty$ *and* $\xi_2^{ec(r)}(t) = \infty$

For an effective choice transition t , the upper bounds $\xi_1^{ec(r)}(t)$ and $\xi_2^{ec(r)}(t)$ are the maximal values of ξ_1 and ξ_2 such that all the effective choice transitions smaller or equal to t in $\mathsf{SPOC}_\xi(r)$ remain effective in all runs containing only states ν such that $\xi_1(\nu) \leq \xi_1^{ec(r)}(t) \wedge \xi_2(\nu) \leq \xi_2^{ec(r)}(t)$ in the suffix after the last occurrence of t. For a transition $t \notin \mathsf{ECT}_\xi(r)$ then both upper bounds are infinite.

We can now define the *non-choice states* and the non-choice extended automaton that characterizes all the runs between consecutive effective choice transitions which maintain the preceding effective choice transitions, as in the observed violation r.

Definition 27 (Non-choice states) *Given a model $\mathcal{A}$ and an observer $\mathcal{O}_P$, let ξ be the level of choice of $\mathcal{A}||\mathcal{O}_P$ and $\langle \mathcal{V}, \Sigma, \rightarrow, \nu^0, \mathcal{V}^F \rangle = [\mathcal{A}||\mathcal{O}_P]$,*

$\forall r \in run^F(\mathcal{A}||\mathcal{O}_P), \langle \nu_1 \xrightarrow{e_1} \nu_1', \nu_2 \xrightarrow{e_2} \nu_2' \rangle \in (\rightarrow^2)$

the non-choice states $NCS^r(\nu_1 \xrightarrow{e_1} \nu_1', \nu_2 \xrightarrow{e_2} \nu_2')$ is the maximal subset $\mathcal{V}'$ of $\mathcal{V}$ such that:

- $\forall \nu \in \mathcal{V}'\ (\xi_1(\nu) \leq \xi_1^{ec(r)}(\nu_1 \xrightarrow{e_1}, \nu_1')) \wedge (\xi_2(\nu) \leq \xi_2^{ec(r)}(\nu_1 \xrightarrow{e_1}, \nu_1'))$
- *and there is a run from ν_1' to ν_2 in $\rightarrow \cap (\mathcal{V}' \times \Sigma \times \mathcal{V}')$*

The restriction $[\mathcal{A}||\mathcal{P}]|_{NCS^r(\nu_1 \xrightarrow{e_1} \nu_1', \nu_2 \xrightarrow{e_2} \nu_2')}$ is the LTS $\langle \mathcal{V}', \Sigma, \rightarrow \cap (\mathcal{V}' \times \Sigma \times \mathcal{V}'), \nu_1', \{\nu_2\} \rangle$

When $\nu_1 \xrightarrow{e_1} \nu_1', \nu_2 \xrightarrow{e_2} \nu_2'$ are two consecutive effective choice transitions in $\mathsf{SPOC}_\xi(r)$, the LTS $[\mathcal{A}||\mathcal{P}]|_{NCS^r(\nu_1 \xrightarrow{e_1} \nu_1', \nu_2 \xrightarrow{e_2} \nu_2')}$ denotes all the potential suffixes of runs after $\nu_1 \xrightarrow{e_1} \nu_1'$ such that:

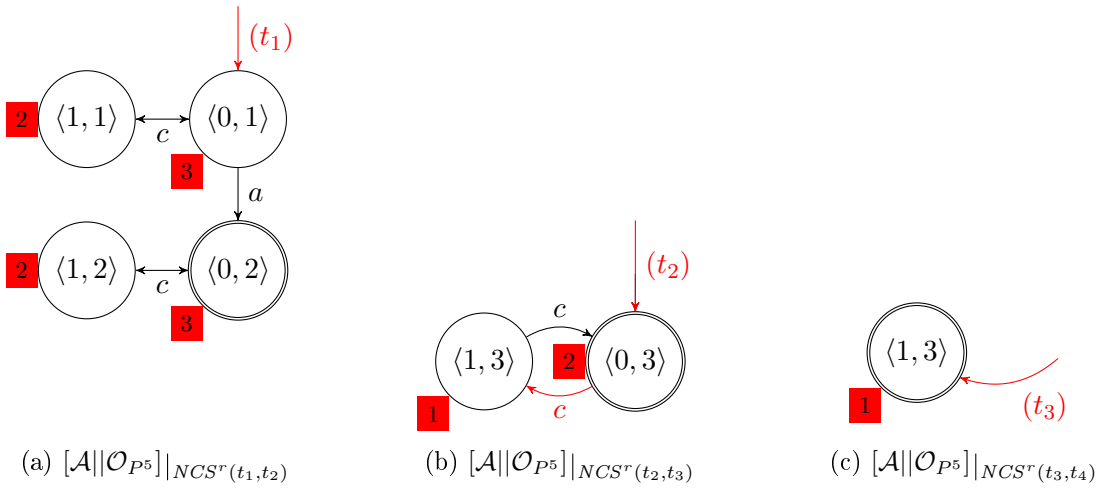
- all the effective choice transitions lower or equal to $\nu_1 \xrightarrow{e_1} \nu_1'$ in $\mathsf{SPOC}_\xi(r)$ remain effective choice transitions,
- the origin ν_2 of the next effective choice transition $\nu_2 \xrightarrow{e_2} \nu_2'$ is reached.

The next definition is used to encode the semantics of choice explanations. For a choice explanation of a trace tr, we want to define all the traces tr' that have the same effective choice transitions. The run between two consecutive effective choice transitions is not relevant as long as the effective choice transitions that already occurred, stays effective. The constraint is to remain in the non-choice state of Definition 27. The level of choice ξ and the property observer $\mathcal{O}_P$ is required to compute such states. Because, the semantics of an explanation is computed without the property observer $\mathcal{O}_P$ and without the level of choice ξ, the next definition extracts the part $\Pi^r_{\mathcal{O}_P}(t_1, t_2)$ of the property observer $\mathcal{O}_P$ required to encode the constraints on the level of choice, in the non-choice states between two consecutive effective choices.

Definition 28 (Non-choice extended automaton) *Given a model $\mathcal{A}$ with variables X, an observer $\mathcal{O}_P = \langle S_{\mathcal{O}_P}, \Sigma_{\mathcal{O}_P}, X_{\mathcal{O}_P}, inv_{\mathcal{O}_P}, \delta_{\mathcal{O}_P}, s^0_{\mathcal{O}_P}, v^0_{\mathcal{O}_P}, F_{\mathcal{O}_P} \rangle$, a run $r \in run^F[\mathcal{A}||\mathcal{P}]$, let $\langle \mathcal{V}, \Sigma, \rightarrow, \nu^0, \mathcal{V}^F \rangle$ $= [\mathcal{A}||\mathcal{P}]$, for all couples of transitions $t_1 = \langle \langle s_a, s_p \rangle, v \rangle \xrightarrow{e} \langle \langle s_a', s_p' \rangle, v' \rangle, t_2 \in \rightarrow$*

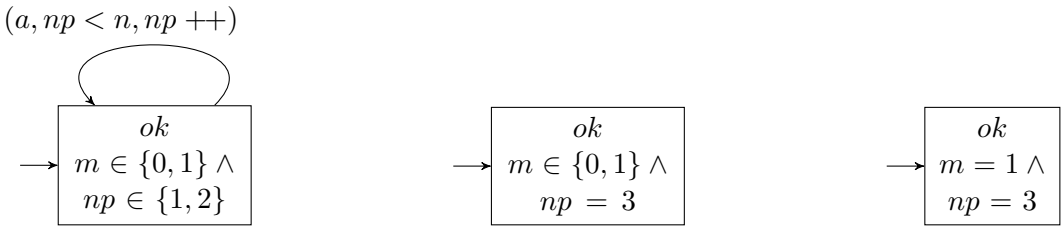
the non-choice extended automaton $\Pi^r_{\mathcal{O}_P}(t_1, t_2)$ is the extended automaton $\langle S, \Sigma_{\mathcal{O}_P}, X \cup X_{\mathcal{O}_P}, inv, \delta_p, s_0, v_0, \emptyset \rangle$ where

- *$s_0 = s_p'$ is the state of $\mathcal{O}_P$ that is reached with t_1*



(a) $[\mathcal{A}||\mathcal{O}_{P^5}]|_{NCS^r(t_1,t_2)}$ (b) $[\mathcal{A}||\mathcal{O}_{P^5}]|_{NCS^r(t_2,t_3)}$ (c) $[\mathcal{A}||\mathcal{O}_{P^5}]|_{NCS^r(t_3,t_4)}$

Figure 4.7: Non-choice states



(a) $\Pi^r_{\mathcal{O}_P}(t_1,t_2)$ (b) $\Pi^r_{\mathcal{O}_P}(t_2,t_3)$ (c) $\Pi^r_{\mathcal{O}_P}(t_3,t_4)$

Figure 4.8: Non-choice extended automata

- $v_0 = \{x \mapsto v'(x) \mid x \in X_{\mathcal{O}_P}\}$ *is the valuation that is reached with* t_1
- $S = \{s_p \mid \langle\langle s_a, s_p\rangle, v\rangle \in NCS^r(t_1,t_2)\}$
- $\forall s \in S, inv(s) = \bigvee_{\langle\langle s_a,s\rangle,v\rangle \in NCS^r(t_1,t_2)} v$
- $\delta_p = \{s_p \overset{e,g,f}{\rightarrow} s'_p \in \delta_{\mathcal{O}_P} \mid s_p, s'_p \in S\}$

In Definition 28, the initial and accepting states/valuation are dummy values which are not used to compute the semantics of an explanation, and we will use the effective choice transition's source and destination instead.

In order to build a concise explanation, the non-choice extended automaton extracts only the necessary part of the observer with respect to the effective choice transitions of the explained violation.

Example 7 (Running example $\mathsf{SPOC}_\xi(r)$, upper bounds on ξ and non-choice extended automata) *We illustrate the construction of a non-choice extended automaton and motivate why it is necessary to encode parts of the property observer in the explanation.*

We consider the same trace tr and run r defined previously in the running example, the effective choice transitions $\mathsf{ECT}_\xi(r)$ *and the respective level of choice of the states are displayed*

in Figure 4.5f. Firstly, we compute $\mathsf{SPOC}_\xi(r) = \langle\{t_1, t_2, t_3, t_4\}, <\rangle$ *with* $t_1 < t_2 < t_3 < t_4$*. The order* $<$ *is total, and therefore the effective choice transitions may only occur in an increasing order (Lemma 4).*

Secondly, we compute the upper bounds $\xi_1^{ec(r)} : (\rightarrow) \rightarrow (\mathbb{N} \cup \{\infty\})$ *on* ξ_1 *(Definition 26). We have:*

- $\xi_1^{ec(r)}(t_1) = 3$
- $\xi_1^{ec(r)}(t_2) = 2$
- $\xi_1^{ec(r)}(t_3) = 1$
- $\xi_1^{ec(r)}(t_4) = 0$
- $\forall t \notin \mathsf{ECT}_\xi(r), \xi_1^{ec(r)}(t) = \infty$

The computation of the non-choice states (Definition 27) between t_1 *and* t_2 *is done on* $[\mathcal{A}||\mathcal{O}_P]$ *(Figure 4.3). We start with the states reachable from* t_1 *(all the non-accepting states), we remove the states* ν *such that* $\xi_1(\nu) > \xi_1^{ec(r)}(t_1) = 3$*, and finally we keep only the states that are co-reachable with respect to* t_2*. Figure 4.7a is the restriction* $[\mathcal{A}||\mathcal{O}_{P^5}]|_{NCS^r(t_1,t_2)}$*, and we do the same computation for the other couples of effective choice transitions* (t_2, t_3) *and* (t_3, t_4)*. The respective LTS* $[\mathcal{A}||\mathcal{O}_{P^5}]|_{NCS^r(t_2,t_3)}$ *and* $[\mathcal{A}||\mathcal{O}_{P^5}]|_{NCS^r(t_3,t_4)}$ *are displayed in Figures 4.7b and 4.7b.*

Finally, we compute the non-choice extended automata (Definition 28)

$\Pi^r_{\mathcal{O}_P}(t_1, t_2), \Pi^r_{\mathcal{O}_P}(t_2, t_3), \Pi^r_{\mathcal{O}_P}(t_3, t_4)$ *displayed in Figure 4.8a,4.8b and 4.8c. The set of variables is the set* $X' = X \cup X_{\mathcal{O}_P}$ *that contains both the variables of* $\mathcal{A}$ *and* $\mathcal{O}_{P^5}$*.*

- $\Pi^r_{\mathcal{O}_P}(t_1, t_2) = \langle\{ok\}, \Sigma_{\mathcal{O}_P}, X', \{ok \mapsto m \in \{0,1\} \wedge np \in \{1,2\}\}, \{ok \overset{a,np<n,np++1}{\rightarrow} ok\}, ok, \{m \mapsto 0, np \mapsto 1\},$
- $\Pi^r_{\mathcal{O}_P}(t_2, t_3) = \langle\{ok\}, \Sigma_{\mathcal{O}_P}, X', \{ok \mapsto m \in \{0,1\} \wedge np = 3\}, \emptyset, ok, \{m \mapsto 0, np \mapsto 3\}, \emptyset\rangle$
- $\Pi^r_{\mathcal{O}_P}(t_3, t_4) = \langle\{ok\}, \Sigma_{\mathcal{O}_P}, X', \{ok \mapsto m = 1 \wedge np = 3\}, \emptyset, ok, \{m \mapsto 1, np \mapsto 3\}, \emptyset\rangle$

In the extended automata $\Pi^r_{\mathcal{O}_P}(t_1, t_2), \Pi^r_{\mathcal{O}_P}(t_2, t_3), \Pi^r_{\mathcal{O}_P}(t_3, t_4)$ *not all the syntactic transitions (Definition 28) are displayed and defined, therefore we have removed the transitions that do not occur in* $[\mathcal{A}||\mathcal{O}_{P^5}]|_{NCS^r(t_1,t_2)}$*,* $[\mathcal{A}||\mathcal{O}_{P^5}]|_{NCS^r(t_2,t_3)}$*, and* $[\mathcal{A}||\mathcal{O}_{P^5}]|_{NCS^r(t_3,t_4)}$ *respectively. Indeed, the invariants of the non-choice extended automata prevent the occurrence of those transitions. For example the invariant* $np = 3$ *prevents* $(a, np < n, np +\!+ 1)$ *from occurring. Removing or leaving those transitions changes the size and readability of the explanations, and we are still working on implementation choices that do not change the semantics of choice explanations.*

In order to encode the effective choice transitions in the explanation, we need the valuation of the variable np. However, the computation of the semantics of an explanation does not depend on $\mathcal{O}_{P^5}$ *but only on* $\mathcal{A}$*. For* $i = 1, 2, 3$*, with the extended automaton* $\Pi^r_{\mathcal{O}_P}(t_i, t_{i+1})$ *(Figure 4.8a, 4.8b, and 4.8c) and the transitions* t_i *and* t_{i+1}*, we are able to construct the LTS* $[\mathcal{A}||\mathcal{O}_{P^5}]|_{NCS^r(t_i,t_{i+1})}$ *(Figure 4.7a,4.7b, and 4.7c) by composing* $\mathcal{A}$ *with* $\Pi^r_{\mathcal{O}_P}(t_i, t_{i+1})$ *and removing any states that are not co-reachable with respect to* t_2*. The following lemma formalizes this construction.*

Lemma 6 makes the connection between the computation of explanations that depends on the safety property observer $\mathcal{O}_P$ and the computation of the semantics that does not require the property observer. The proof of requires the hypothesis (H).

Lemma 6 (Lemma of connection) *Given a behavioral model $\mathcal{A}$, an observer $\mathcal{O}_P$ and a run $r \in run^F(\mathcal{A}||\mathcal{O}_P)$, let ξ be the level of choice of $\mathcal{A}||\mathcal{P}$ and $\langle \mathcal{V}, \Sigma, \rightarrow, \nu^0, \mathcal{V}^F \rangle = [\mathcal{A}||\mathcal{O}_P]$.*

For all pair of effective choice transitions $\langle t_1, t_2 \rangle \in \mathit{SPOC}_\xi(r)$ where $t_1 = \nu_1 \xrightarrow{e_1} \langle s_1, v_1 \rangle$ and $t_2 = \langle s_2, v_2 \rangle \xrightarrow{e_2} \nu_2$, we have $run^F([\mathcal{A}||\mathcal{O}_P]|_{NCS^r(t_1,t_2)}) = run(\mathcal{A}||\Pi^r_{\mathcal{O}_P}(t_1, t_2), s_1, v_1, s_2, v_2)$.

Proof. Let us define the two semantic LTS $\mathcal{G} = [\mathcal{A}||\mathcal{O}_P]|_{NCS^r(t_1,t_2)} = \langle \mathcal{V}_\mathcal{G}, \Sigma_\mathcal{G}, X_\mathcal{G}, \rightarrow_\mathcal{G}, \nu^0_\mathcal{G}, \mathcal{V}^F_\mathcal{G} \rangle$ and $\mathcal{G}' = [\mathcal{A}||\Pi^r_{\mathcal{O}_P}(t_1, t_2)] = \langle \mathcal{V}_{\mathcal{G}'}, \Sigma_{\mathcal{G}'}, X_{\mathcal{G}'}, \rightarrow_{\mathcal{G}'}, \nu^0_{\mathcal{G}'}, \mathcal{V}^F_{\mathcal{G}'} \rangle$.

Firstly, we prove that $\mathcal{V}_{G'} = NCS^r(t_1, t_2)$. By definition of the composition $\mathcal{A}||\Pi^r_{\mathcal{O}_P}(t_1, t_2)$ and the semantic LTS $\mathcal{G}'$, a state $\langle \langle s_a, s_p \rangle, v \rangle \in \mathcal{V}'_\mathcal{G}$ satisfies:

- s_a is a control state of $\mathcal{A}$,
- s_p is a control state of $\mathcal{O}_P$ such that $\exists \langle \langle s'_a, s_p \rangle, v' \rangle \in NCS^r(t_1, t_2)$ (Defintion 28)
- v satisfies both the invariant of s_a and the invariant $\bigvee_{\langle \langle s_a, s \rangle, v' \rangle \in NCS^r(t_1,t_2)} v'$ of s_p in $\Pi^r_{\mathcal{O}_P}(t_1, t_2)$ (Defintion 28)

Hence $\exists \langle \langle s'_a, s_p \rangle, v \rangle \in NCS^r(t_1, t_2)$. However, the valuation v satisfies both the invariant (in the model $\mathcal{A}$) of s_a and s'_a. By hypothesis (H) on the behavioral model, different control states of $\mathcal{A}$ have disjoint invariants. Therefore, $s_a = s_a \wedge \langle \langle s_a, s_p \rangle, v \rangle \in NCS^r(t_1, t_2)$.

Inversely, a state in $NCS^r(t_1, t_2)$ is in $\mathcal{V}'_\mathcal{G}$ by Definition 28 of $\Pi^r_{\mathcal{O}_P}(t_1, t_2)$.

Hence, the states of $\mathcal{G}$ and $\mathcal{G}'$ are exactly $NCS^r(t_1, t_2)$.

Secondly, we have $\delta_p = \{s_p \xrightarrow{e,g,f} s'_p \in \delta_{\mathcal{O}_P} \mid s_p, s'_p \in S\}$ (Definition 28), therefore the syntactic transitions of $\mathcal{A}||\Pi^r_{\mathcal{O}_P}(t_1, t_2)$ are syntactic transitions of $\mathcal{A}||\mathcal{O}_P$. Hence, the reachable transitions in $\mathcal{G}$ and $\mathcal{G}'$ from the state $\langle s_1, v_1 \rangle$ are the same.

Finally, the accepting states of $\mathcal{G}$ are $\{\langle s_2, v_2 \rangle\}$ and the initial state of $\mathcal{G}$ is $\langle s_1, v_1 \rangle$ and all the runs in $run(\mathcal{A}||\Pi^r_{\mathcal{O}_P}(t_1, t_2), s_1, v_1, s_2, v_2)$ start in $\langle s_1, v_1 \rangle$ and end in the state $\langle s_2, v_2 \rangle$ (Definition 10. □

4.5.5 Choice Explanation and their Semantics

Intuitively, the semantics of a causal explanation for the violation of a safety property is the set of behaviors that share the causes that are reported in the explanation. Hence, the semantics of an explanation is a language of the behaviors having the same effective choice transitions, in a similar order than the observed violation.

One can compute the semantics of an explanation without knowing the violated property or the respective observer. The motivation for this constraint is to avoid the potential complexity of the property observer. It allows explanations to be comparable between each others via the semantics.

Definition 29 (Choice Explanation) *Given a behavioral model $\mathcal{A}$, an observer $\mathcal{O}_P$ of the safety property P, an observer $\mathcal{O}_Q$ of the safety property Q such that $\neg P \implies \neg Q$, let ξ be the level of choice function of $[\mathcal{A}||\mathcal{P}] = \langle \mathcal{V}, \Sigma, \rightarrow \rangle$, for all runs $r = \nu_0 \xrightarrow{e_1} \dots \xrightarrow{e_n} \nu_n \in run^F(\mathcal{A}||\mathcal{P})$ violating P, the choice explanation (**ce**) of the trace $e_1 \dots e_n$ is $\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, e_1 \dots e_n) = \langle \pi, \mathsf{SPOC}_\xi(r_c) \cup \{\langle t_\pi, t \rangle \mid t \in \mathsf{ECT}_\xi($ such that:*

Let i_Q be the minimal integer in $[1, n]$ such that $e_1 \dots e_{i_Q} \notin \mathsf{safeLang}_{\mathcal{A}}(\mathcal{O}_Q)$

- $t_\pi = \nu_{i_Q - 1} \xrightarrow{e_{i_Q}} \nu_{i_Q}$
- $\pi = e_1 \dots e_{i_Q - 1} e_{i_Q}$
- $r_c = \nu_{i_Q} \xrightarrow{e_{i_Q+1}} \dots \xrightarrow{e_n} \nu_n$ *is the "contrastive" suffix of r.*
- ϕ *such that*

  $\forall \langle t, t' \rangle$ *consecutives in* $\mathsf{SPOC}_\xi(r_c)$, *such that ν is the destination of t and ν' is the origin of t'*

  $\phi(\langle t, t' \rangle) = \Pi^{r_c}_{\mathcal{O}_P}(t, t')$

  and

  $\forall t \in \min(\mathsf{SPOC}_\xi(r_c)), \nu \xrightarrow{e} \nu' = t, \phi(t_\pi, t) = \Pi^{r_c}_{\mathcal{O}_P}(t_\pi, t)$.

The contrastive prefix π encodes the violation of the contrastive property Q by reaching a state in which the violation of Q is inevitable. At the end of this prefix, the transition t_π is the starting point from where we compute the contributory causes for the violation of the safety property P.

The core of the explanation is the partial order $\mathsf{SPOC}_\xi(r_c) \cup \{\langle t_\pi, t \rangle \mid t \in \mathsf{ECT}_\xi(r_c)\}$ of transitions. The minimal element of the order is the transition t_π and the maximal element is the contributory cause of r that leads to a state where the violation of the safety property P is inevitable. The rest of the elements is the set of contributory causes in r that lead to the violation of P. This element encodes the occurrences of the contributory causes in the explanation.

The last element of the explanation is a function, ϕ that encodes an invariant for each couple of consecutive contributory causes with an extended automaton. The invariants give the context in which the contributory causes remain effective.

With the definition of the choice explanation, we define the semantics of choice explanations. As opposed to the construction of the explanation, that definition only depends on an explanation and the behavioral model but not on the property observer.

Definition 30 (Semantics of choice explanation) *Given a model $\mathcal{A}$ such that $[\mathcal{A}] = \langle \mathcal{V}, \Sigma, \rightarrow, \nu^0, \mathcal{V}^F \rangle$, and an explanation $\varepsilon = \langle \pi, \langle E, < \rangle, \phi \rangle$, the semantics $sem_{\mathcal{A}}(\varepsilon)$ of ε is defined by :*

$\forall w = e_0 e_1 \dots e_n \in trace(\mathcal{A}), w \in sem_{\mathcal{A}}(\varepsilon)$

$\iff$ *π is a prefix of w and*

$\exists \psi : E \rightarrow [|\pi|, n]$ *injective and monotonic w.r.t. $<$ such that:*

- t_π *is the minimal element of E and $\psi(t_\pi) = |\pi|$, and*

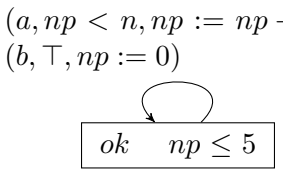


Figure 4.9: Non-choice extended-automaton $\Pi^{r_c}_{\mathcal{O}_P}(t_\pi, t_2)$

- $\forall t = \langle \nu, e, \nu' \rangle \in E, e_{\psi(t)} = e$, *and*
- $\forall (t_1, t_2) \in dom(\phi)$, *such that* $t_1 = \langle s_1, v_1 \rangle \xrightarrow{e_1} \langle s'_1, v'_1 \rangle$ *and* $t_2 = \langle s_2, v_2 \rangle \xrightarrow{e_2} \langle s'_2, v'_2 \rangle$, $e_{\psi(t_1)+1} \dots e_{\psi(t_2)-1} \in traces(run(\mathcal{A}||\phi(t_1, t_2), s'_1, v'_1, s_2, v_2))$

A trace w in the semantics of an explanation $\langle \pi, \langle E, < \rangle, \phi \rangle$, starts by the contrastive prefix π and is a trace of the model $\mathcal{A}$. There exists a function of occurrence ψ that returns the index of the occurrence of each of the events of the transitions in E. Furthermore, $\forall (t_1, t_2) \in dom(\phi)$, the sub-chain of the trace w between the index $\psi(t_1)$ and $\psi(t_2)$ of E must satisfy the invariant $\phi(t_1, t_2)$.

In the next subsection we prove the expected properties of explanations defined in Section 4.3 on choice explanations. Because the choice explanation explains a trace and not a log, we prove the weaker version of completeness with respect to a log. We prove another property (Theorem 3) that justifies the intuition we gave on the semantics with respect to the causes of a trace. Except for contrastive prefix for the contrastive property Q, the contributory causes of the explained trace are also contributory causes of all traces of the semantic of the choice explanation. That causal property implies the satisfaction of the expected properties on choice explanations.

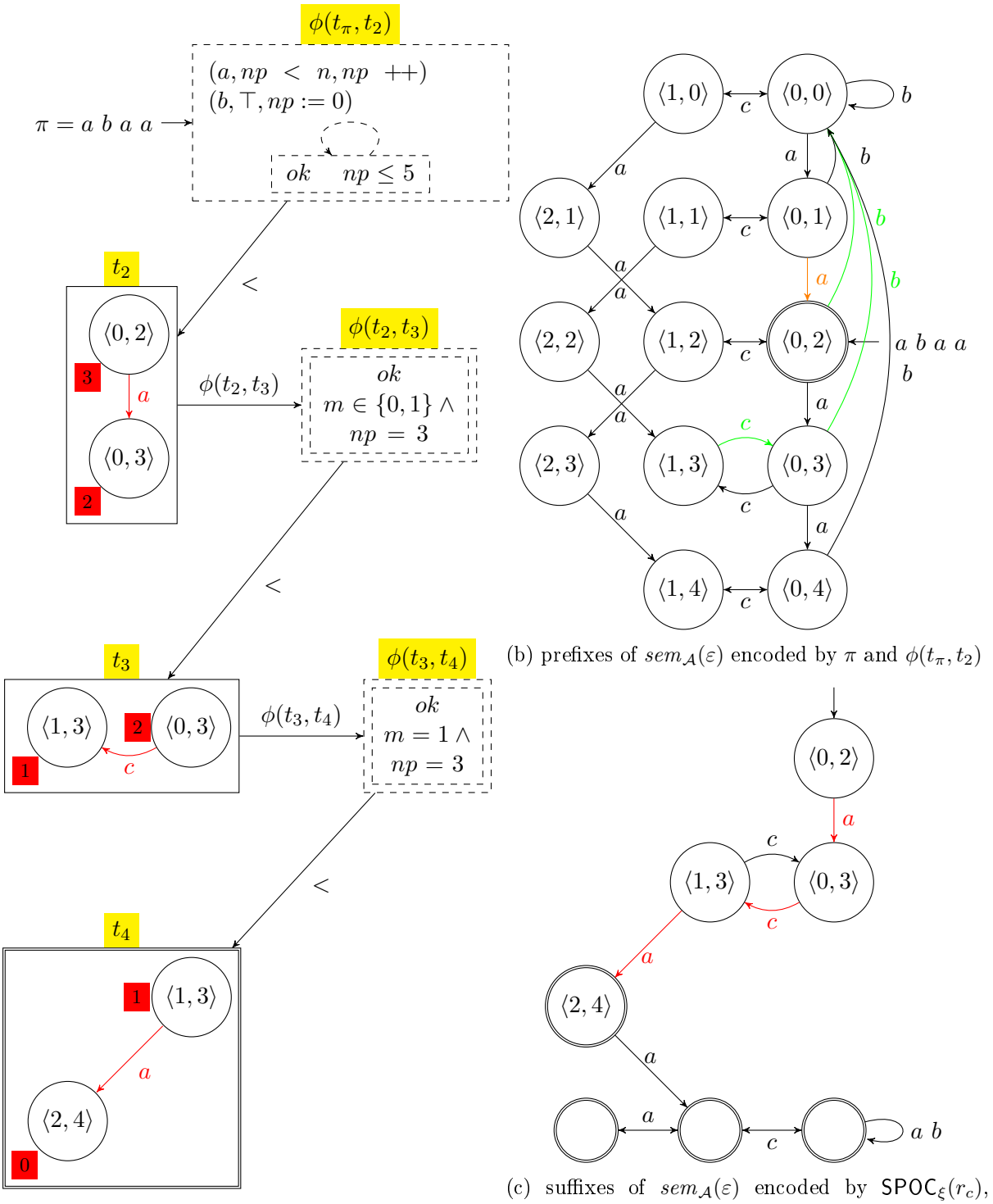
4.5.6 Example

Let us consider construct the choice explanation $\varepsilon = \mathbf{ce}(\mathcal{A}, \mathcal{O}_{P^5}, \mathcal{O}_Q, tr)$ for the running example where :

- The behavioral model $\mathcal{A}$ is defined in Figure 4.2a, all the events are controllable $\Sigma_c := \Sigma$.
- The safety property P^5 and property observer $\mathcal{O}_{P^5}$ are shown in Figure 4.2b.
- The contrastive property Q and property observer $\mathcal{O}_Q$ are shown in Figure 4.2c.
- $tr = a\ b\ a\ a\ c\ c\ a\ c\ c\ c\ a\ a$ in $L^F(\mathcal{A}||\mathcal{O}_P)$ and $r = \nu_0 \xrightarrow{e_1} \nu_1 \dots \nu_{11} \xrightarrow{e_{12}} \nu_{12} \in run^F(\mathcal{A}||\mathcal{O}_{P^5})$ such that $e_1 \dots e_{12} = tr$.

Firstly, we compute the contrastive prefix π: we have $a\ b\ a\ a \notin \mathsf{safeLang}_{\mathcal{A}}(\mathcal{O}_Q)$ and $a\ b\ a \in \mathsf{safeLang}_{\mathcal{A}}(\mathcal{O}_Q)$ and therefore $\pi = a\ b\ a\ a$.

Let t_π be the last transition $\nu_3 \xrightarrow{a} \nu_4 = \langle \langle s^0, ok \rangle, \{m \mapsto 0, np \mapsto 1\} \rangle \xrightarrow{a} \langle \langle s^0, ok \rangle, \{m \mapsto 0, np \mapsto 2\} \rangle$ of the trace π on the $[\mathcal{A}||\mathcal{O}_P]$.



(a) Explanation $\varepsilon = \mathbf{ce}(\mathcal{A}, \mathcal{O}_{P^5}, \mathcal{O}_Q, tr)$

(b) prefixes of $sem_{\mathcal{A}}(\varepsilon)$ encoded by π and $\phi(t_\pi, t_2)$

(c) suffixes of $sem_{\mathcal{A}}(\varepsilon)$ encoded by $\mathsf{SPOC}_\xi(r_c)$, $\phi(t_2, t_3)$, and $\phi(t_3, t_4)$

Figure 4.10: Explanation ε and $sem_{\mathcal{A}}(\varepsilon)$ is the concatenation of the prefixes and the suffixes by the

In the suffix of the run $r_c = \nu_4 \xrightarrow{e_5} \ldots \xrightarrow{e_{12}} \nu_{12}$ defined as Definition 29 there are 3 effective choice transitions (t_2, t_3, t_4) and displayed in Figure 4.5f. The transition t_1 is in $\mathsf{ECT}_\xi(r)$ but not in $\mathsf{ECT}_\xi(r_c)$ and it occurs in the contrastive prefix π. The safe alternatives (Definitions 22) to those effective choice transitions are displayed in green in Figure 4.3.

We have $\mathsf{SPOC}_\xi(r_c) = \langle \{t_2, t_3, t_4\}, <' \rangle$ where $<'$ is defined such that $t_2 <' t_3 <' t_4$. There is a unique maximal effective choice transition t_4, and after the occurrence of that transition the violation of P^5 is inevitable.

We define $\langle E, < \rangle = \mathsf{SPOC}_\xi(r_c) \cup \{\langle t_\pi, t \rangle \mid t \in \mathsf{ECT}_\xi(r_c)\}$ where $E = \{t_\pi, t_2, t_3, t_4\}$ and $t_\pi < t_2 < t_3 < t_4$. The first two elements $\langle \pi, \langle E, < \rangle \rangle$ of are computed, we compute the predicates $\phi(t_\pi, t_2), \phi(t_2, t_3), \phi(t_3, t_4)$.

For $t \in \{t_\pi, t_2, t_3, t_4\}$, we have $\xi_1^{ec(r_c)}(t) = \xi_1^{ec(r)}(t)$ (and $\xi_2^{ec(r_c)}(t) = \xi_2^{ec(r)}(t)$).

Therefore, we have $\phi(t_2, t_3) = \Pi^{r_c}_{\mathcal{O}_P}(t_2, t_3) = \Pi^{r}_{\mathcal{O}_P}(t_2, t_3)$ and $\phi(t_3, t_4) = \Pi^{r_c}_{\mathcal{O}_P}(t_3, t_4) = \Pi^{r}_{\mathcal{O}_P}(t_3, t_4)$, the predicates are displayed in Figure 4.8b and 4.8c.

The last predicate to compute in $\phi(t_\pi, t_2)$ and we have $\xi_1^{ec(r_c)}(t_\pi) = \infty$. There are no constraints on the level of choice and the non-choice states $NCS^r(t_\pi, t_2)$ are all the states of $\mathcal{A}||\mathcal{O}_{P^5}$ from which t_2 is reachable. The respective predicate is $\Pi^r_{\mathcal{O}_P}(t_\pi, t_2)$ displayed in Figure 4.9.

The contrastive choice explanation $\varepsilon = \mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr)$ is the tuple $\langle \pi, \langle E, < \rangle, \phi \rangle$ and it is displayed in Figure 4.10a. The explanation starts with the contrastive prefix π, then we have $np \leq 5$ until the last occurrence of the minimal effective choice transition t_2 in state where $m = 0$ and $np = 2$. The prefixes of the traces in the semantics of ε is displayed in Figure 4.3 and is obtained by concatenating the prefix π with the traces of the LTS $[\mathcal{A}||\phi(t_\pi, t_2)]$ where the destination of t_π is the initial state, and the origin of t_2 is the only accepting states.

The rest of the explanation encodes the occurrence of the effective choice transitions that reduced the capacity of the system to avoid a violation of P^5, between the consecutive effective choice transitions t_i and t_{i+1} is displayed the predicate $\phi(t_i, t_{i+1})$. The semantic of ε is the set of traces in $trace(\mathcal{A})$ that can be split into a prefix in Figure 4.10b and a suffix in Figure 4.10c. By definition of the semantics of choice explanation (Definition 30), a trace w of $trace(\mathcal{A})$ with a prefix w' in $sem_{\mathcal{A}}(\varepsilon)$ is in $sem_{\mathcal{A}}(\varepsilon)$ because the occurrence function ψ of w' is an occurrence function of w. After the maximal effective choice transition t_4, there are no constraints on the semantics on the continuation of the traces of $sem_{\mathcal{A}}(\varepsilon)$.

The explanation ε is sound with respect to $\mathcal{O}_{P^5}$ because a run in $run(\mathcal{A}||\mathcal{O}_{P^5})$ of a trace in $sem_{\mathcal{A}}(\varepsilon)$, reaches the state where $m = 2$ and $np = 4$ after the occurrence of t_4. From that state the violation of P^5 is inevitable and we have :

$$sem_{\mathcal{A}}(\varepsilon) \cap \mathsf{safeLang}_{\mathcal{A},\emptyset}(\mathcal{O}_{P^5}) = \emptyset$$

As expected, the trace tr is in $sem_{\mathcal{A}}(\varepsilon)$ with the prefix $a\ b\ a\ a\ c\ c$ and the suffix $a\ c\ c\ c\ a\ a$.

In the next subsection, the choice explanation algorithm **ce**/choice explanations satisfy the expected properties of explanations in section 4.3 except for the completeness with respect to $\mathcal{O}$ and $\mathcal{L}$. A choice explanation takes a trace as input and not a log. Indeed, different traces

in $L^F(\mathcal{A}||\mathcal{O}||\mathcal{L})$ may not share the same effective choices transitions and have different choice explanations.

4.5.7 Properties of Choice Explanations

The following theorem establishes the correspondence between the contributory causes of a trace and the semantics of a choice explanation. It states that a trace is in the semantics of a choice explanation of tr if and only if it has the same contrastive prefix π as tr for the contrastive property Q, and its contributory causes (with respect to the safety property P) contain the contributory causes of tr. This theorem is satisfactory, because all the other theorem on the semantics of explanation are corollary of that theorem.

Theorem 3 (Semantics of choice explanations is causal) *Given a behavioral model $\mathcal{A}$, an observer $\mathcal{O}_P$ of the safety property P, an observer $\mathcal{O}_Q$ of the safety property Q such that $\neg P \implies \neg Q$, a run $r \in run^F(\mathcal{A}||\mathcal{P})$ violating P, let ξ be the level of choice of $\mathcal{A}||\mathcal{O}_P$, tr be the trace of r, and $\langle \pi, \langle E, < \rangle, \phi \rangle = \mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr)$. For all runs $r' \in run(\mathcal{A}||\mathcal{P})$ such that tr' is the trace of r', we have the equivalence : $tr' \in sem_{\mathcal{A}}(\langle \pi, \langle E, < \rangle, \phi \rangle) \iff \pi$ is a prefix of tr' and $\mathsf{ECT}_{\xi_1}(r_c) \subseteq \mathsf{ECT}_{\xi}(r'_c) \wedge \mathsf{ECT}_{\xi_2}(r_c) \subseteq \mathsf{ECT}_{\xi_2}(r'_c)$ where r_c, r'_c are the contrastive suffixes of the run r, r' as defined in Definition 29.*

We prove each implication separately : For the first implication from left to right, we prove that the explanation encode well the effective choice transitions. For the second implication from right to left, we have to prove that, when a trace tr' has more effective choice transitions than tr, the stronger causal constraints in tr' implies the requirements on the semantics of the explanation of tr.

Proof. Let $r' = \nu_0 \xrightarrow{e_1} \nu_1 \ldots \xrightarrow{e_n} \nu_n \in run(\mathcal{A}||\mathcal{P})$ such that tr' is the trace of r'.

Implication from left to right. Let us suppose that $tr' \in sem_{\mathcal{A}}(\langle \pi, \langle E, < \rangle, \phi \rangle)$, we have directly that π is a prefix of tr' by Definition 30 of $sem_{\mathcal{A}}$.

Firstly, we prove that all effective choice transitions in $\mathsf{ECT}_{\xi_1}(r_c) \cup \mathsf{ECT}_{\xi_2}(r_c)$ are transitions in r'.

There exists an occurrence function ψ of tr' as defined in Definition 30 because $tr' \in sem_{\mathcal{A}}(\langle \pi, \langle E, < \rangle, \phi \rangle)$.

First, we prove by induction on $t \in (\mathsf{SPOC}_{\xi}(r_c) \cup \{\langle \bot, t \rangle \mid t \in \mathsf{ECT}_{\xi}(r_c)\})$ that

$$\nu_{\psi(t)-1} \xrightarrow{e_{\psi(t)}} \nu_{\psi(t)} = t \tag{4.2}$$

For $t = \bot$, by definition of $tr' \in sem_{\mathcal{A}}(\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr))$, the trace tr and tr' have the same prefix π, the respective prefix of the runs r ans r' ends therefore with the transition $\bot$, and we have $\nu_{\psi(t)-1} \xrightarrow{e_{\psi(t)}} \nu_{\psi(t)} = t$. For the induction step, let $t \in \mathsf{SPOC}_{\xi}(r_c)$, $\exists t_0 \in (\mathsf{SPOC}_{\xi}(r_c) \cup \{\langle \bot, t \rangle \mid t \in \mathsf{ECT}_{\xi}(r_c)\})$ $(t \neq \bot)$ such that $\langle t_0, t \rangle \in dom((\phi))$. By induction hypothesis, we have $\nu_{\psi(t_0)-1} \xrightarrow{e_{\psi(t_0)}} \nu_{\psi(t_0)} = t_0$. By definition of the semantics, we have

$$e_{\psi(t_0)+1} \dots e_{\psi(t)-1} \in traces(run(\mathcal{A}||\phi(t_0,t), s_{\psi(t_0)}, v_{\psi(t_0)}, s_{\psi(t)-1}, v_{\psi(t)-1}))$$

Lemma 6 gives us that the runs in $run(\mathcal{A}||\phi(t_0,t), s_{\psi(t_0)}, v_{\psi(t_0)}, s_{\psi(t)-1}, v_{\psi(t)-1})$ are runs in $\mathcal{A}||\mathcal{O}_P$ from $\nu_{\psi(t_0)}$ to $\nu_{\psi(t)-1}$ and that $\nu_{\psi(t)-1}$ is the origin of t. By definition of the semantic, $e_{\psi(t)}$ is the event of t, and because $\mathcal{A}||\mathcal{O}_P$ is deterministic we have $\nu_{\psi(t)-1} \overset{e_{\psi(t)}}{\to} \nu_{\psi(t)} = t$

Secondly, we prove that the effective choice transitions in $\mathsf{ECT}_{\xi_1}(r_c)$ are effective choice transitions of r'_c, let $t = \nu \overset{e}{\to} \nu' \in \mathsf{ECT}_{\xi_1}(r_c)$, t is the $\psi(t)$-th transition of r' (4.2). We have from the definition of effective choice transitions (Definition 21) $\nu_{\psi(t)-1} \overset{e_{\psi}(t)}{\to} \nu_{\psi(t)} \in \mathsf{ECT}_{\xi_1}(r'_c) \iff \forall j \in [\psi(t), n], \xi_1(\nu_{\psi(t)-1}) > \xi_1(\nu_j)$.

Let $j \in [\psi(t), n]$, we know there exists $t_1, t_2 \in \mathsf{SPOC}_\xi(r_c)$ such that :

- $t < t_1$ with respect to $\mathsf{SPOC}_\xi(r_c)$ or $t = t_1$, and
- $\langle t_1, t_2 \rangle \in dom((\phi))$, and
- $j \in [\psi(t_1), \psi(t_2)[$

because $tr' \in sem_{\mathcal{A}}(\langle \pi, \langle E, < \rangle, \phi \rangle)$.

We have $\nu_j \in NCS^r(t_1, t_2)$ (Lemma 6), and therefore $\xi_1(\nu_j) \leq \xi_1^{ec(r)}(t_1)$ (Definition 27).

Moreover, we have $\xi_1^{ec(r)}(t) \geq \xi_1^{ec(r)}(t_1)$ because $t \in pre(t_1)$ with respect to $\mathsf{SPOC}_\xi(r_c)$ and by Definition 26.

Because $t \in \mathsf{ECT}_{\xi_1}(r_c), \xi_1^{ec(r)}(t) = \xi_1(\nu_{\psi(t)-1}) - 1$

Hence, $\xi_1(\nu_{\psi(t)-1}) > \xi_1^{ec(r)}(t) \geq \xi_1^{ec(r)}(t_1) \geq \xi_1(\nu_j)$.

We have proved that $\forall j \in [\psi(t), n], \xi_1(\nu_{\psi(t)-1}) > \xi_1(\nu_j)$ and therefore $t \in \mathsf{ECT}_{\xi_1}(r'_c)$.

The proof $\mathsf{ECT}_{\xi_2}(r_c) \subseteq \mathsf{ECT}_{\xi_2}(r'_c)$ is the same proof using the closed system level of choice ξ_2 instead of ξ_1.

Implication from right to left. Let us suppose that π is a prefix of tr' and

$$\mathsf{ECT}_{\xi_1}(r_c) \subseteq \mathsf{ECT}_{\xi_1}(r'_c) \wedge \mathsf{ECT}_{\xi_2}(r_c) \subseteq \mathsf{ECT}_{\xi_2}(r'_c) \tag{4.3}$$

We define the occurrence function $\psi_{r'}$ such that $\forall \nu \overset{e}{\to} \nu' \in \mathsf{ECT}_\xi(r), \psi_r(\nu \overset{e}{\to} \nu')$ is the last occurrence of $\nu \overset{e}{\to} \nu'$ in r'. We prove that tr' and ψ_r satisfy the requirements in Definition 30 of $sem_{\mathcal{A}}(\langle \pi, \langle E, < \rangle, \phi \rangle)$.

The monotonicity of ψ_r w.r.t. $\mathsf{SPOC}_\xi(r)$ is obtained by direct application of Lemma 4. ψ_r is injective by definition of $\mathsf{SPOC}_\xi(r)$ and $\perp$.

First, we have $\psi_r(\perp) = |\pi|$ (requirement 1 of Definition 30) because π is a prefix of tr'. We can notice that by definition, $\perp$ occurs at most once in a run, because after its occurrence the violation of Q is inevitable.

By construction of ψ_r we have :

$$\forall t \in E, \nu_{\psi_r(t)-1} \overset{e_{\psi_r(t)}}{\to} \nu_{\psi_r(t)} = t \tag{4.4}$$

And therefore $\forall t \in E$, $e_{\phi_r(t)}$ is the event of t (requirement 2 of Definition 30) .

The last requirement of Definition 30 of $sem_{\mathcal{A}}$ is that ψ_r satisfies

$\forall (t_1,t_2) \in dom(\phi), e_{\psi_r(t_1)+1} \dots e_{\psi_r(t_2)-1} \in traces(run(\mathcal{A}||\phi(t_1,t_2), s_{\psi_r(t_1)}, v_{\psi_r(t_1)}, s_{\psi_r(t_2)-1}, v_{\psi_r(t_2)-1}))$

Let $(t_1,t_2) \in dom(\phi)$, such that $t_1 = \langle s_1, v_1\rangle \xrightarrow{e_1} \langle s_1', v_1'\rangle$ and $t_2 = \langle s_2, v_2\rangle \xrightarrow{e_2} \langle s_2', v_2'\rangle$, we have $\phi(t_1,t_2) = \Pi^r_{\mathcal{O}_P}(t_1,t_2)$ by Definition 29 of $\langle \pi, \langle E, <\rangle, \phi\rangle$

With respect to $\mathsf{SPOC}_\xi(r_c)$, we apply Lemma 6 and we have

$$run^F([\mathcal{A}||\mathcal{O}_P]_{|NCS^r(t_1,t_2)}) = run(\mathcal{A}||\Pi^r_{\mathcal{O}_P}(t_1,t_2), s_1', v_1', s_2, v_2) \tag{4.5}$$

The run r_c' has more effective choice transitions than r_c (4.3), we prove that there are less runs that preserve the effective transitions in $pre(t_1)$ of $\mathsf{SPOC}_\xi(r_c')$ than runs that preserve the effective transitions in $pre(t_1)$ of $\mathsf{SPOC}_\xi(r_c)$.

The upper bounds $\xi_1^{ec(r_c')}$ are more restrictive than $\xi_1^{ec(r_c)}$ for the same effective choice transition, formally (4.3) implies $\xi_1^{ec(r_c')}(t_1) \leq \xi_1^{ec(r_c)}(t_1) \wedge \xi_2^{ec(r_c')}(t_1) \leq \xi_2^{ec(r_c)}(t_1)$ by Definition 26 of $\xi_1^{ec(r_c')}, \xi_2^{ec(r_c')}, \xi_1^{ec(r_c)}$ and $\xi_2^{ec(r_c)}$.

Therefore, we have $NCS^{r_c'}(t_1,t_2) \subseteq NCS^{r_c}(t_1,t_2)$ (Definition 27).

The runs that remains in those states are also included, as a side effect:

$$run^F([\mathcal{A}||\mathcal{O}_P]_{|NCS^{r_c'}(t_1,t_2)}) \subseteq run^F([\mathcal{A}||\mathcal{O}_P]_{|NCS^{r_c}(t_1,t_2)}) \tag{4.6}$$

One of those runs is $\nu_{\psi_r(t_1)} \xrightarrow{e_{\psi_r(t_1)+1}} \nu_{\psi_r(t_1)+1} \dots \xrightarrow{e_{\psi_r(t_2)-1}} \nu_{\psi_r(t_2)-1} \in run^F([\mathcal{A}||\mathcal{O}_P]_{|NCS^{r_c'}(t_1,t_2)})$

Therefore $\nu_{\psi_r(t_1)} \xrightarrow{e_{\psi_r(t_1)+1}} \nu_{\psi_r(t_1)+1} \dots \xrightarrow{e_{\psi_r(t_2)-1}} \nu_{\psi_r(t_2)-1} \in run^F([\mathcal{A}||\mathcal{O}_P]_{|NCS^{r_c}(t_1,t_2)})$ (4.6)

We conclude with the application of Lemma 6 that makes the correspondence between $[\mathcal{A}||\mathcal{O}_P]_{|NCS^{r_c}(t_1,t_2)}$ and $\mathcal{A}||\phi(t_1,t_2)$ and we obtain that :

$e_{\psi_r(t_1)+1} \dots e_{\psi_r(t_2)-1} \in traces(run(\mathcal{A}||\phi(t_1,t_2), s_1', v_1', s_2, v_2))$.

The last item of requirement 30 of $sem_{\mathcal{A}}$ is therefore satisfied, and we have $tr' \in sem_{\mathcal{A}}(\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_1))$

□

This theorem proves the soundness with respect to $\mathcal{O}_P$ of choice explanation defined in Proposition 1.

Theorem 4 (Soundness of choice explanations) *Given a behavioral model $\mathcal{A}$, an observer $\mathcal{O}_P$ of the safety property P, an observer $\mathcal{O}_Q$ of the safety property Q such that $\neg P \implies \neg Q$, for all run $r = \nu_0 \xrightarrow{e_1} \dots \xrightarrow{e_n} \nu_n \in run^F(\mathcal{A}||\mathcal{P})$ violating P, let tr be the trace of r, the choice explanation $\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr)$ is sound with respect to $\mathcal{O}_P$.*

The soundness of the choice explanation is implied by the existence of a maximal effective choice transition in $\mathsf{SPOC}_\xi(r)$, which leads to $\xi = \langle 0, 0\rangle$, and occurs in any runs of the traces of the semantics of the explanation. In such state the violation is inevitable.

Proof. Let $\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr) = \langle \pi, \mathsf{SPOC}_\xi(r_c) \cup \{\langle \bot, t\rangle \mid t \in \mathsf{ECT}_\xi(r_c)\}, \phi\rangle$ be the explanation as defined in Definition 29. Let ξ be the level of choice of $\mathcal{A}||\mathcal{O}_P$ and $r' = \nu_0 \xrightarrow{e_1} \nu_1 \xrightarrow{e_2} \dots \xrightarrow{e_n} \nu_n$ be a run in $run(\mathcal{A}||\mathcal{P})$ such that its trace $tr' = e_1 \dots e_n$ is in $sem_{\mathcal{A}}(\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr))$,

Let t be the unique maximal element of $\mathsf{SPOC}_\xi(r_c)$ (Lemma 5), t is also an effective choice transition of r' (Theorem 3), we have $t = \nu_i \xrightarrow{e_i} \nu_i$ with $\xi(\nu_i) = \langle 0, 0\rangle$ for some $i \leq n$. With Lemma 3, from $\nu_{\psi(t)}$, $\neg P$ is inevitable because $\xi(\nu_{\psi(t)}) = \langle 0, 0\rangle$. Hence, $tr' \notin \mathsf{safeLang}_{\mathcal{A},\emptyset}(\mathcal{O}_P)$ □

This theorem proves the weak completeness Property 2. The explained trace is always contained in the semantic of its choice explanation. We compute explanation for a given trace and not for a log, except for some settings where the trace represents the whole log, a choice explanation alone is not complete because two traces consistent with the same log, may have different contributory causes or different contrastive prefixes, in that case each trace has its own choice explanation, and the explanations have disjoints semantics (Theorem 3).

Theorem 5 (Weak completeness of choice explanations) *Given a behavioral model $\mathcal{A}$, an observer $\mathcal{O}_P$ of the safety property P, an observer $\mathcal{O}_Q$ of the safety property Q such that $\neg P \implies \neg Q$, for all run $r = \nu_0 \xrightarrow{e_1} \ldots \xrightarrow{e_n} \nu_n \in run^F(\mathcal{A}||\mathcal{P})$ violating P, let tr be the trace of r, we have $tr \in sem_{\mathcal{A}}(\boldsymbol{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr))$*

The proof is the direct application of Theorem 3 because π is a prefix of tr (by definition of π in Definition 29) and $\mathsf{ECT}_{\xi_1}(r_c) \subseteq \mathsf{ECT}_{\xi}(r_c) \wedge \mathsf{ECT}_{\xi_2}(r_c) \subseteq \mathsf{ECT}_{\xi_2}(r_c)$ (r_c is a suffix of r).

This theorem proves that the monotony of choice explanations defined in Property 3.

Theorem 6 (Monotony of choice explanation semantics) *Given a behavioral model $\mathcal{A}$, an observer $\mathcal{O}_P$ of the safety property P, an observer $\mathcal{O}_Q$ of the safety property Q such that $\neg P \implies \neg Q$, for all $tr_1, tr_2 \in L^F(\mathcal{A}||\mathcal{O}_P)$*

$$tr_2 \in sem_{\mathcal{A}}(\boldsymbol{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_1)) \implies sem_{\mathcal{A}}(\boldsymbol{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_2)) \subseteq sem_{\mathcal{A}}(\boldsymbol{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_1))$$

The proof is obtained by 3 applications of Theorem 3.

Proof. Let $tr_1, tr_2 \in L^F(\mathcal{A}||\mathcal{O}_P)$, $tr \in trace(\mathcal{A}||\mathcal{O}_P)$ and r_1, r_2, r be the respective runs of tr_1, tr_2, tr in $\mathcal{A}||\mathcal{O}_P$, such that $tr_2 \in sem_{\mathcal{A}}(\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_1)) \wedge tr \in sem_{\mathcal{A}}(\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_2))$.

We apply the theorem 3 on both $tr_2 \in sem_{\mathcal{A}}(\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_1))$ and $tr \in sem_{\mathcal{A}}(\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_2))$, we have

All the traces tr_1, tr_2, tr have the same prefix π, that is the contrastive prefix for the contrastive property Q.

Let r_{c1}, r_{c2}, r_c be the contrastive suffix of the runs r_1, r_2, r as defined in Definition 29.

$$\begin{aligned} \mathsf{ECT}_{\xi_1}(r_{c1}) \subseteq \mathsf{ECT}_{\xi_1}(r_{c2}) \wedge \mathsf{ECT}_{\xi_2}(r_{c1}) \subseteq \mathsf{ECT}_{\xi_2}(r_{c2}) \\ \mathsf{ECT}_{\xi_1}(r_{c2}) \subseteq \mathsf{ECT}_{\xi_1}(r_c) \wedge \mathsf{ECT}_{\xi_2}(r_{c2}) \subseteq \mathsf{ECT}_{\xi_2}(r_c) \end{aligned} \tag{4.7}$$

Hence,

$$\mathsf{ECT}_{\xi_1}(r_{c1}) \subseteq \mathsf{ECT}_{\xi_1}(r_c) \wedge \mathsf{ECT}_{\xi_2}(r_{c1}) \subseteq \mathsf{ECT}_{\xi_2}(r_c) \tag{4.8}$$

We can apply Theorem 3 again on the last equation and we have $tr \in sem_{\mathcal{A}}(\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_1))$.

We have proved that $sem_{\mathcal{A}}(\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_2)) \subseteq sem_{\mathcal{A}}(\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_1))$.

□

4.6 Choice Explanations: Case Study

An example inspired from the autonomous car domain to illustrate our proposal.

The system, named the *two-cars system*, considers a car equipped with a cruise controller and a sensor following another car named the front car. The sensor measures the distance of the car with respect to the front car as shown in Figure 4.11.

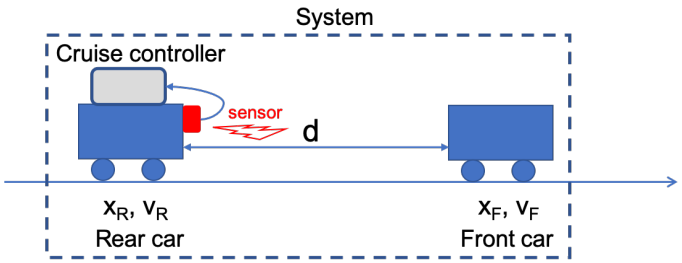


Figure 4.11: Two cars following each other

The position of the front car fcx and the position of the car cx evolve according to their respective speeds:

$$fcx \leftarrow fcx + \delta t \times fcs, \tag{4.9}$$

$$cx \leftarrow cx + \delta t \times cs, \tag{4.10}$$

where $cs \in [0,3]$ and $fcs \in [0,3]$ are the speeds of the car and the front car respectively, and *new* indicates the update after one time unit $\delta t = 1$.

We assume that the front car is free to choose its speed in $\{0,1,2,3\}$ and that the car cruise controller is designed to avoid collision in any situation, i.e. to maintain some distance between the two cars.

At any time, the distance between the two cars $sd \in [0,4]$ is given by:

$$sd = fcx - cx. \tag{4.11}$$

Using (4.9) in (4.11), we can derive the following update of the distance after one time unit:

$$\begin{cases} sd \leftarrow sd - cs + fcs & \text{if } sd + fcs \geq cs \\ sd \leftarrow 0 & \text{otherwise} \end{cases} \tag{4.12}$$

We also assume that the sensor may fail and if it fails it reports a measured distance smd that is twice the real distance sd. The sensor health status $sfs \in \{ok, ko\}$ hence affects the behavior of the cruise controller.

The two cars system is represented as three interacting components, the *front car*, the *cruise controller* of the car, and the *sensor* of the car. The events that are used in the models are the following:

- ar: (rear) car acceleration,
- brr: (rear) car break,
- crr: (rear) car cruise,
- af: front car acceleration,
- brf: front car break,
- crf: front car cruise,
- $fail$: sensor fault event.

Car and front car accelerations ar and af increase the car and front car speed by 1 unit. Car and front car break brr and brf decrease the car and front car speed by 1 unit. The cruise crr and crf actions maintain the speeds of the car and front car, respectively.

The three interacting components are modeled by three extended automata:

- An automaton that models the front car and its speed (fcs). The front car can accelerate (af), break (brf) or cruise (crf).
- An automaton that models the cruise controller. The cruise controller can accelerate (a), break (br) or cruise (cr), depending on the measured distance (smd) provided by the sensor. The action is enabled as long as the distance sd remains greater than 0.
- An automaton that models the sensor that provides a measured distance (smd) of the real distance (sd) between the controlled car and the front car. The sensor may fail ($fail$), if it fails the measured distance (smd) is twice the real distance (sd), which affects the behavior of the cruise controller (sfs).

Figure 4.12 presents the structural model of the system. Variables cs, smd and sfs are shared by the automata of the cruise controller and the sensor. Variable fcs is shared between the sensor and the front car automata. Variable sd is internal to the sensor automaton. Events a, br, cr are synchronized events in the three components. Events af, brf, crf are synchronized events in the cruise controller and the front car and event $fail$ is synchronized between the sensor and the cruise controller.

For any action of the front car (af, brf, crf), the cruise controller responds with an action for the car (i.e. ar, brr, crr). For any action of the car, the sensor updates the measured distance smd based on the real distance sd that is function of the speed difference between the front car (fcs) and the car (cs) as given by (4.12). The event $fail$ only occurs in the sensor and at most one in a run. If it occurs, the measured distance smd and sfs signal are immediately updated

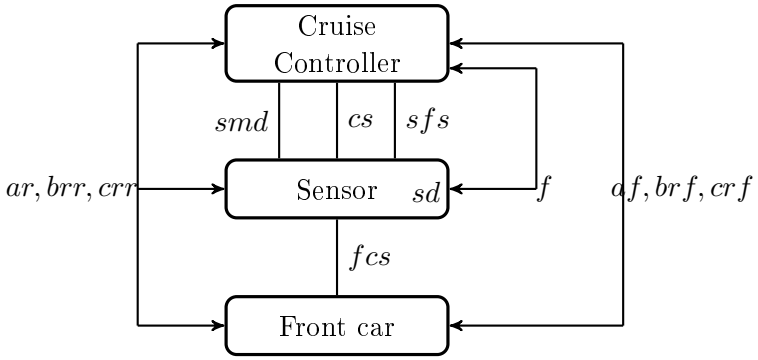


Figure 4.12: Structural model of the system.

in the sensor and in the cruise controller, which then believes that the distance is actually twice the real distance, i.e. $smd = 2 \times sd$. A variable cm is used to distinguish the states in which the front car makes an action, with $cm = active$, to the states in which the (rear) car makes an action, with $cm = ready$.

Figure 4.13 illustrates the extended automaton of the sensor, in which the two control states s_{ok} and s_{ko} reflect the health modes of the sensor and the initial situation is given by $s^0 = s_{ok}$, $v^0(smd) = 2$, $v^0(sfs) = ok$, $v^0(fcs) = 1$, $v^0(sd) = 2$, $v^0(cs) = 0$. The initial situation of the system is the car stopped at distance 2 of the front car whose speed is 1, with a correctly working sensor.

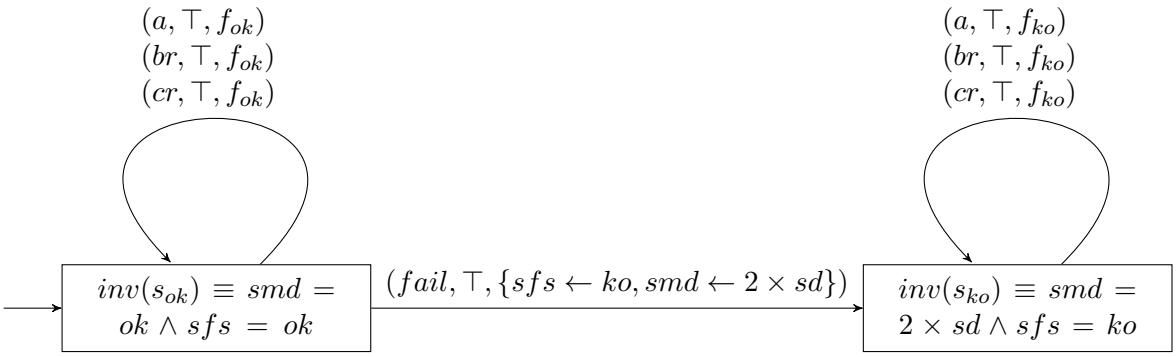


Figure 4.13: Extended automaton of the sensor. The partial update functions $f_{ok}, f_{ko} \in UPDATE(X)$ are defined by $f_{ok/ko}(sd) : \mathbb{X} \mapsto \mathbb{X}_{sd}$, $f_{ok/ko}(smd) : \mathbb{X} \mapsto \mathbb{X}_{smd}$, with $f_{ok/ko}(sd) = f_{ok}(smd) = \begin{cases} sd - cs + fcs & \text{if } sd + fcs \geq cs \\ 0 & \text{otherwise} \end{cases}$, and $f_{ko}(smd) = \begin{cases} 2 \times (sd - cs + fcs) & \text{if } sd + fcs \geq cs \\ 0 & \text{otherwise} \end{cases}$.

The behavior of the two cars system is obtained by composing the extended automata of the front car, the cruise controller, and the sensor. It can be explained intuitively. As long as the control state of the sensor is s_{ok}, (1) the front car freely chooses its speed in the interval $[0, 3]$ (2) the role of the cruise controller is to maintain the distance sd in the interval $[1, 4]$ in order

to enable platooning and avoid collision; (3) once the speed $cs \geq 1$ is reached, cs is maintained in the interval $[1, 3]$; and (4) in each state at least one of the events ar, brr, crr, af, brf, crf, $fail$ is enabled.

When a *fail* event has occurred, the sensor control state transitions to s_{ko} and the sensor reports twice the actual distance to the cruise controller that takes the decision of allowing cruise or acceleration for the car from an erroneous distance value. If sd=0 at some point, it indicates a collision.

We consider a safety property $P =$ "$sd > 0$", i.e. no collision, to be respected by the two cars system and a contrastive property $Q =$ "$d \geq 2$", i.e. a desirable goal, for which we design two respective observers according to Definition 17.

The observer $\mathcal{O}_P$ for the property violation $P =$ "$sd > 0$" is defined as an extended automaton compatible with the behavioral model $\mathcal{A}$. It synchronizes on the variables in P with $\mathcal{A}$, i.e. each time the variable distance sd is updated, and enters a sink state when $sd = 0$, i.e. when the safety property P is violated. The observer $\mathcal{O}_Q$ for the contrastive property is defined similarly and it enters a sink state when $sd < 2$, i.e. when the contrastive property Q is not respected.

Let $\diamondsuit \in \{P, Q\}$, then the observer $\mathcal{O}_\diamondsuit = \langle S_{\mathcal{O}_\diamondsuit}, \Sigma_{\mathcal{O}_\diamondsuit}, X_{\mathcal{O}_\diamondsuit}, inv_{\mathcal{O}_\diamondsuit}, \delta_{\mathcal{O}_\diamondsuit}, v^0_{\mathcal{O}_\diamondsuit}, F_{\mathcal{O}_\diamondsuit} \rangle$ can be taken as the observer of the violation of $\diamondsuit$ with:

- $S_{\mathcal{O}_\diamondsuit} = \{in_{\mathcal{O}_\diamondsuit}, out_{\mathcal{O}_\diamondsuit}\}$
- $\Sigma_{\mathcal{O}_\diamondsuit} = \{ar, brr, crr\}$
- $X_{\mathcal{O}_\diamondsuit} = \{sd\}$
- $inv_{\mathcal{O}_\diamondsuit}(in_{\mathcal{O}_\diamondsuit}) = \diamondsuit$
- $inv_{\mathcal{O}_\diamondsuit}(out_{\mathcal{O}_\diamondsuit}) = \neg\diamondsuit$
- $\delta_{\mathcal{O}_\diamondsuit}$:
  - $\langle in_{\mathcal{O}_\diamondsuit}, ar, in_{\mathcal{O}_\diamondsuit}, \top, \varnothing \rangle$
  - $\langle in_{\mathcal{O}_\diamondsuit}, brr, in_{\mathcal{O}_\diamondsuit}, \top, \varnothing \rangle$
  - $\langle in_{\mathcal{O}_\diamondsuit}, crr, in_{\mathcal{O}_\diamondsuit}, \top, \varnothing \rangle$
- $F_{\mathcal{O}_\diamondsuit} = \{out_{\mathcal{O}_\diamondsuit}\}$
- $v^0_{\mathcal{O}_\diamondsuit}(sd) = 2$

Note that invariants play an essential role in the observer. Indeed, the observer has two states, $in_{\mathcal{O}_\diamondsuit}$ whose invariant forces the property $\diamondsuit$, and $out_{\mathcal{O}_\diamondsuit}$ whose invariant forces the negation of the property $\neg\diamondsuit$. The state $out_{\mathcal{O}_\diamondsuit}$ is a sink state. The observer being initially in state $in_{\mathcal{O}_\diamondsuit}$, when any event occurs there are always two transitions, one that leaves the observer in the same state and the other that drives the observer towards the sink state. It is the invariants that decide which state is authorized according to the value of the variable sd that is known to the observer by synchronization with $\mathcal{A}$.

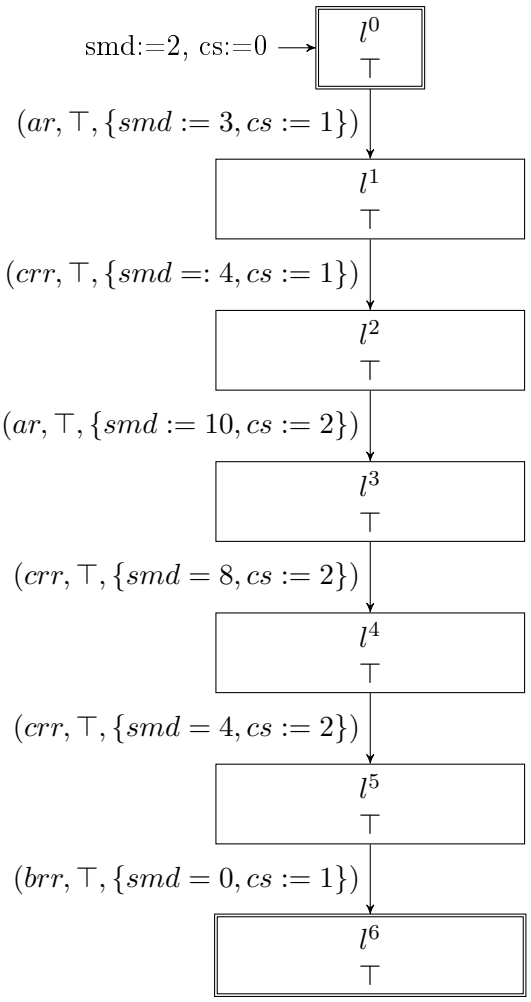
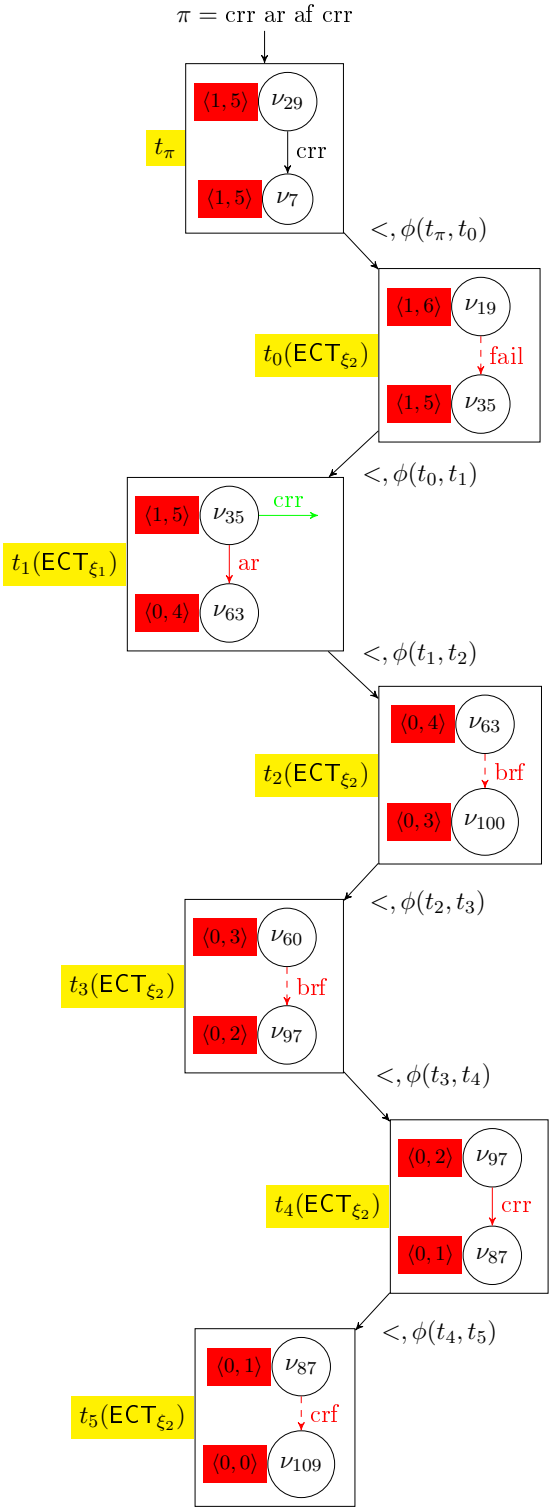


Figure 4.14: Log $\mathcal{L}_0$

For the two cars scenario, one of the logs that we may consider the two-cars system has produced is $\mathcal{L}_0$ (Figure 4.14), which is the sequence $ar{\cdot}crr{\cdot}ar{\cdot}crr{\cdot}crr{\cdot}br$ with the observed variables $X' = \{smd, cs\}$ (the measured distance and the speed of the rear car) and observed events $\Sigma^o = \{ar, brr, crr\}$ (rear car acceleration, braking and cruise). $\mathcal{L}_0$ is the extended automaton $\mathcal{L}_{X'}(r_0) = \langle\{l^0, \ldots, l^6\}, \Sigma^o, X', inv', \delta', (s^0)', (v^0)', F'\rangle$ for some run $r_0 \in run^F(\mathcal{A}||\mathcal{O}_P)$. Initially, the speed ($cs$) of the rear car is 0 and the measured (smd) distance is 0. At the end of the log, the measured distance is 0 and the safety property P is violated because $smd = 0 \implies sd = 0$.

The question to be answered is: what kind of *explanations* can we provide about the violation of the safety property $P =$ "$sd > 0$" in contrast to the desired property $Q =$ "$sd \geq 2$" based on the log $\mathcal{L}_0$, the model $\mathcal{A}$ and the observers $\mathcal{O}_P$ and $\mathcal{O}_Q$?

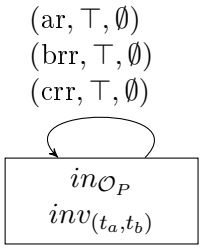
There are 4 traces tr_1, tr_2, tr_3, tr_4 in $L^F(\mathcal{A}||\mathcal{L}_0)$ that are consistent with the log. Those traces are also in $L^F(\mathcal{A}||\mathcal{O}_P)$ because the collision is observed with $smd = 0$ at the end of the log. The traces tr_1 and tr_2 (resp. tr_3 and t_4) share the same contrastive prefix π for the violation of Q and the same effective choice transitions (in the suffix after π), therefore we have $\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_1) = \mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_2)$ and $\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_3) = \mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_4)$.



(a) Contrastive prefix π and $\langle E, < \rangle$

ν_i	cm	cs	fcs	sd	sfs	smd
ν_{29}	ready	1	2	3	ok	3
ν_7	active	1	2	4	ok	4
ν_{19}	ready	1	2	4	ok	4
ν_{35}	ready	1	2	4	ko	8
ν_{63}	active	2	2	5	ko	10
ν_{100}	ready	2	1	5	ko	10
ν_{60}	active	2	1	4	ko	8
ν_{97}	ready	2	0	4	ko	8
ν_{87}	active	2	0	2	ko	4
ν_{109}	ready	2	0	2	ko	4

(b) Valuations of the states in E, when $cm = active$ the front car picks an action, in the states where $cm = ready$ the rear car picks an action and the variables in $\{cs, fcs, sd, smd\}$ are then updated.



(c) $\phi(t_a, t_b)$ for $a, b \in \{(\pi, 0), (0, 1), (1, 2), (2, 3), (4, 5)\}$: The variables are $\{cm, cs, fcs, sd, sfs, smd\}$, and the initial valuation is the valuation of the destination of t_a.

- $inv_{(t_\pi, t_0)}(in_{\mathcal{O}_P}) = sfs = ok \wedge sd = 4 \wedge cs = 1 \wedge fcs \in \{1, 2\}$
- $inv_{(t_0, t_1)}(in_{\mathcal{O}_P}) = sfs = ko \wedge ($
  $sd = 1 \wedge (cs = 1 \vee cs = 2 \wedge fcs \in \{1, 2\}) \vee$
  $sd \in \{2, 3\} \vee$
  $sd \in \{4, 5\} \wedge cs \in \{1, 2\})$
- $inv_{(t_1, t_2)}(in_{\mathcal{O}_P}) = sfs = ko \wedge sd = 5 \wedge fcs = 2 \wedge cs = 2$
- $inv_{(t_2, t_3)}(in_{\mathcal{O}_P}) = sfs = ko \wedge sd \in \{4, 5\} \wedge fcs = 1 \wedge cs = 2$
- $inv_{(t_3, t_4)}(in_{\mathcal{O}_P}) = sfs = ko \wedge sd = 4 \wedge fcs = 0 \wedge cs = 2$
- $inv_{(t_4, t_5)}(in_{\mathcal{O}_P}) = sfs = ko \wedge sd = 2 \wedge fcs = 0 \wedge cs = 2$

(d) Invariants of ϕ

Figure 4.15: Choice explanation $\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_1) = \langle \pi, \langle E, < \rangle, \phi \rangle$: the contrastive prefix π and the relation $\langle E, < \rangle$ is displayed in Figure 4.15a, the states of the transitions in E are given in Figure 4.15b, and the predicate ϕ is defined in Figure 4.15c and Figure 4.15d.

We present the explanation $\mathbf{ce}(\mathcal{A}, \mathcal{O}_P, \mathcal{O}_Q, tr_1) = \langle \pi, \langle E, < \rangle, \phi \rangle$ in Figure 4.15. In Figure 4.15a, we have the order in which the effective choice occurs, the controllability of the transitions (dashed means uncontrollable), and the evolution of the level of choice (in red). The valuations of the states in E are given in Figure 4.15b and the contents of the predicates ϕ are given in Figure 4.15c and Figure 4.15d.

In the following, we discuss what can we learn about the collision from the effective choice transitions in $\langle E, < \rangle$ and the predicates in ϕ.

- Contrastive prefix π and t_π: The contrastive prefix is $\pi =$ crr ar af crr and ends by reaching the state ν_7 with the transition t_π. From that state, the environment can force the system to reach a state where $sd \leq 2$.

- $t_0 \in \mathsf{ECT}_{\xi_2}(r)$: the sensor fails, and the measured distance smd is now twice the actual distance sd. Moreover, the failure of the sensor cannot be undone and the only way to reach $\xi_2 \geq 6$ again while $sfs = ko$ is to reach a state where $cs = 0$ (3 states out of 154 in $[\mathcal{A}||\mathcal{O}_P]$). Note that the predicate $\phi(t_0, t_1)$ encodes much more behaviors than the other predicates because it precedes the first (and only) effective choice transitions in ECT_{ξ_1} and there are no constraints on ξ_1 between t_0 and t_1. In the rest of the explanation after t_1, the predicates provide a more specific context about why the effective choice transitions remain effective.

- $t_1 \in \mathsf{ECT}_{\xi_1}(r)$: The car rear accelerates (controllable action) at the speed of $cs = 2$ to catch up with the front car (the measured distance is 10). That transition can be avoided via the safe alternative crr (a controllable transition that does not decrease ξ_1). The predicate $\phi(t_1, t_2)$ tells us that no actions happen between t_1 or t_2, otherwise t_1 is not effective or t_2 is not reachable. After this point we have $\xi_1 = 0$ and the front car can force a collision.

- $t_2, t_3 \in \mathsf{ECT}_{\xi_2}(r)$: The front car reduces its speed to 0 by braking two times (uncontrollable actions) and the distance sd starts to reduce (from 5 to 4). From the invariants of $\phi(t_2, t_3)$ and $\phi(t_3, t_4)$, we read that the rear car maintains its speed ($cs = 2$) (this is why t_2 and t_3 are effective).

- $t_4 \in \mathsf{ECT}_{\xi_2}(r)$: The rear car cruises and maintains its speed ($cs = 2$ in $\phi(t_4, t_5)$) while the front car is not moving ($fcs = 2$ in $\phi(t_4, t_5)$).

- $t_5 \in \mathsf{ECT}_{\xi_2}(r)$: Immediately, the front car is stopped and does not accelerate but cruises (crf), after that point the collision is inevitable.

Most of the actions that lead to the collision are uncontrollable (ECT_{ξ_2}), and the only controllable effective choice transitions are the acceleration (ECT_{ξ_1}) (instead of crr) and the cruise later (crr instead of brr) (ECT_{ξ_2}).

4.7 Conclusion

We have presented a framework in which explanations are paired with a semantic function that returns the set of behaviors that shares the same causes encoded in the explanation. We have formalized a set of expected properties on explanation functions with respect to its semantics function. Furthermore, we have presented a construction of explanations, called choice explanations, that is able to cope with partial observability of events. Effective choice explanations highlight the "fateful" choices in an execution, as well as alternative events that would have helped avoid the outcome. Effective choice explanations are therefore able to explain failures stemming from non-deterministic choices, such as concurrency bugs. The choice explanation function and its semantics satisfy the expected properties of soundness, weak-completeness and monotony. We have applied the constructions of choice explanations on a case study.

Chapter 5

Discrete explanations on TA

Resumé

Le travail présenté dans ce chapitre fait partie d'un effort continu pour construire des systèmes temps réel embarqués explicables. Nous étudions comment construire, à partir d'un modèle de système, d'une propriété de sûreté et d'un log d'exécution qui viole la propriété, une explication concise de la manière dont le log a entraîné la violation de la propriété.

Nous présentons les contributions suivantes. Nous fournissons une définition formelle des explications causales sur les modèles à temps dense, basée sur le formalisme bien étudié des automates temporisés. Nous proposons une approche symbolique pour construire efficacement des explications. Nous illustrons notre approche à l'aide de plusieurs exemples et d'une étude de cas.

5.1 Introduction

The work presented in this chapter is part of an ongoing effort to construct explainable embedded real-time systems. We investigate how to construct, from a system model, a safety property, and an execution log that violates the property, a concise explanation of how the log brought about the violation of the property.

We present the following contributions. We provide a formal definition of causal explanations on dense-time models, based on the well-studied formalism of timed automata. We propose a symbolic approach to effectively construct explanations. We illustrate our approach on several examples and a case study.

5.2 Explanations

Our goal is to explain, for a TA $\mathcal{A}$ modelling a system, an observer $\mathcal{P}$ of a safety property, and a timed log $\mathcal{L}$ of observable events of $\mathcal{A}$ that violates $\mathcal{P}$, how the violation came to happen. More precisely, in this work, we focus on *non-deterministic choices* in the execution that entails a failure, where there exist different choices that can avoid it. Prominent of such failures are

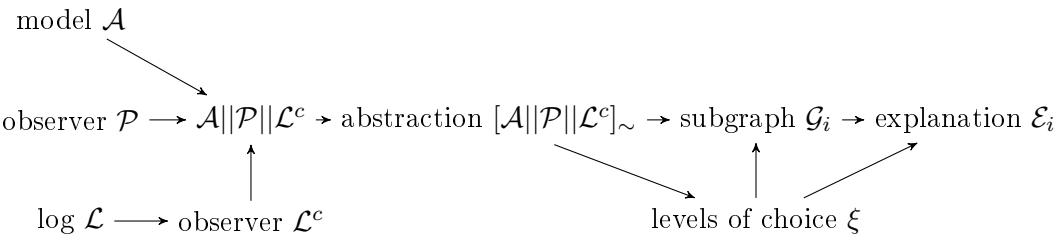


Figure 5.1: Overview of the approach.

"concurrency bugs" that occur in certain interleavings of threads, and deadline misses due to bad scheduling decisions in real-time systems.

To construct such explanations, we lift the *effective choice explanations* formalized in Chapter 4 to timed automata. The main steps of our construction, as shown in Figure 5.1, are the following.

1. Construct a timed *log observer* $\mathcal{L}^c$ from $\mathcal{L}$ that tracks, for any run ρ of $\mathcal{L}^c$, whether the observable behavior of ρ produces $\mathcal{L}$.

2. Compose $\mathcal{A}$, $\mathcal{P}$, and $\mathcal{L}^c$ to form a timed automaton $\mathcal{A}||\mathcal{P}||\mathcal{L}^c$, where $\|$ is the standard parallel composition of timed automata, see *e.g.* [41].

3. Construct a discrete abstraction $[\mathcal{A}||\mathcal{P}||\mathcal{L}^c]_\sim$ of the continuous-time semantics of $\mathcal{A}||\mathcal{P}||\mathcal{L}^c$, using a time-abstracting bisimulation.

4. Compute the *levels of choice* on $[\mathcal{A}||\mathcal{P}||\mathcal{L}^c]_\sim$ that, intuitively, represent, for each equivalence class $q \in [\mathcal{A}||\mathcal{P}||\mathcal{L}^c]_\sim$, the number of *bad choices* $\xi(q)$ left before violating $\mathcal{P}$.

5. Extract, from $[\mathcal{A}||\mathcal{P}||\mathcal{L}^c]_\sim$, a sequence of subgraphs $\mathcal{G}_i$ representing the traces that will be condensed to an explanation of length i.

6. Abstract away, from each $\mathcal{G}_i$, all transitions that do not decrease ξ, in order to obtain the explanation $\mathcal{E}_i$ that retains only those (discrete or timed) transitions that contribute to the failure.

We will now discuss each of these steps.

Log Observer.

To account for the fact that some events are not observable, the set $\Sigma = \Sigma^o \uplus \Sigma^{uo}$ of events is partitioned into two subsets of observable (Σ^o) and unobservable events (Σ^{uo}).

Definition 31 (Timed Log) *A timed log is a one-clock, deterministic and acyclic TA whose events are the observable events* Σ^o.

An example of timed log is a timed automaton roughly depicted as follows:

$$\ell_0 \overset{x=t_0,a,\emptyset}{\to} \ell_1 \overset{x=t_1,b,\emptyset}{\to} \ell_2 \overset{x=t_2,a,\emptyset}{\to} \ell_3$$

with ℓ_3 as an accepting location and ℓ_0 an initial location. The edge from the location ℓ_0 to the location ℓ_1 has the guard $x = t_0$; its event is a; and its set of clocks to be reset is empty.

As mentioned in the introduction, we are interested in explaining why a system, modeled as a timed automaton $\mathcal{A}$, violates a safety property. To this end, we define a *safety property observer* as a timed automaton with one sink state, to model the property violation of a timed log. In addition, this observer is required to be receptive with respect to all observable events of $\mathcal{A}$, as defined in the following. An LTS $\langle \mathcal{V}, \Sigma, \rightarrow, \nu_0, \mathcal{V}^F \rangle$ is *receptive* over $\Sigma_1 \subseteq \Sigma$ if at every reachable state ν, all events in Σ_1 are enabled, that is, $\forall \sigma \in \Sigma_1 \; \exists \nu' : \nu \xrightarrow{\sigma} \nu'$. Given a timed log $\mathcal{L}$ and a safety property observer $\mathcal{P}$, we say that $\mathcal{L}$ violates the safety property at hand if each run of $\mathcal{L}$ is a run of $\mathcal{P}$ that reaches the sink state.

Given an alphabet Σ and a timed log $\mathcal{L}$ over the observable alphabet $\Sigma^o \subseteq \Sigma$, we construct a *log observer* that accepts all runs over Σ and enters an accepting *sink* state whenever an observed behavior is inconsistent with the log.

Definition 32 (Log Observer) *The observer of a log* $\mathcal{L} = \langle \Sigma, L, X, \mathcal{I}, E, \ell_0, L^F \rangle$ *is the TA* $\mathcal{L}^c := \langle \Sigma, L', X, \mathcal{I}', E', \ell_0, L^F \rangle$ *where:*

- $L' = L \cup \{sink\}$ *where* $sink \notin L$ *is a fresh location;*
- $\mathcal{I}' = \mathcal{I} \cup \{sink \mapsto true\}$*;*
- $E' = E \cup E_1 \cup E_2$ *where*
  - $E_1 = \{\ell \overset{\mathcal{C}(\ell,\sigma),\sigma,\emptyset}{\rightarrow} sink \mid \ell \in L \wedge \sigma \in \Sigma\}$ *where* $\mathcal{C}(\ell, \sigma) = \neg \bigvee \{g \mid \exists \ell' \in L : \ell \overset{g,\sigma,\cdot}{\rightarrow} \ell'\}$.
  - $E_2 = \{sink \overset{true,\sigma,\emptyset}{\rightarrow} sink \mid \sigma \in \Sigma\}$.

Intuitively, E_1 is the set of edges from a location in L that are not consistent with $\mathcal{L}$, used to make the log observer receptive with respect to Σ^o.

Discrete Abstraction.

The sets of states and transitions of a timed automaton are infinite. We use the strong time-abstracting bisimulation (STAB) as a finite abstraction to construct explanations.

To obtain discrete abstractions for the timed automaton $\mathcal{A}||\mathcal{P}||\mathcal{L}^c$, we use the STAB of Definition 15. Given a TA over alphabet Σ with semantic LTS $[\mathcal{A}||\mathcal{P}||\mathcal{L}^c] = \langle \mathcal{V}, \Sigma \cup \mathbb{R}_{>0}, \rightarrow, \nu_0, \mathcal{V}^F \rangle$ and a partition $\tilde{\mathcal{V}}$ of $\mathcal{V}$, the quotient of $[\mathcal{A}||\mathcal{P}||\mathcal{L}^c]$ with respect to $\tilde{\mathcal{V}}$ is the LTS $[\mathcal{A}||\mathcal{P}||\mathcal{L}^c]_\sim = \langle \tilde{\mathcal{V}}, \Sigma', \rightarrow_\sim, \tilde{\nu}^0, \tilde{\mathcal{V}}^F \rangle$ where:

- $\Sigma' = \Sigma \cup \{\delta\}$ where δ is a fresh symbol;
- $\rightarrow_\sim = \{\langle \tilde{\nu}, \sigma, \tilde{\nu}' \rangle \mid \nu \xrightarrow{\sigma} \nu' \wedge \sigma \in \Sigma\} \cup \{\langle \tilde{\nu}, \delta, \tilde{\nu}' \rangle \mid \exists t \in \mathbb{R}_{>0} : \nu \xrightarrow{t} \nu'\}$;
- $\tilde{\nu}^0 = \tilde{\nu}^0$ and $\tilde{\mathcal{V}}^F = \{\tilde{\nu} \mid \nu \in \mathcal{V}^F\}$

and for $\nu \in \mathcal{V}$, $\tilde{\nu}$ denotes the element of $\tilde{\mathcal{V}}$ for which $\nu \in \tilde{\nu}$.

In particular, we are interested in the quotient with respect to the equivalence classes of states, called symbolic states, induced by STAB. This quotient graph can be computed by the existing timed automata model-checkers, such as UPPAAL and Kronos [41, 26]. In this work, we use the tool Minim [38] integrated in Kronos.

We then reduce the latter with respect to strong bisimulation so as to merge bisimilar states involving different locations.

Level of Choice.

In this chapter, we only consider the first component (ξ_1) of the level of choice function (Definition 20), the closed-system level of choice, and we call it the level of choice (ξ) for the sake of simplicity.

In the following, we define explanations based on the level of choice of the LTS $[\mathcal{A}||\mathcal{P}||\mathcal{L}^c]_\sim$.

Definition 33 (Effective Choice) *Given an LTS $G = \langle \mathcal{V}, \Sigma, \rightarrow, \mathcal{V}^F, \nu_0 \rangle$ and a level of choice function $\xi : \mathcal{V} \rightarrow \mathbb{N} \cup \{\infty\}$, a transition $s \xrightarrow{e} s' \in E$ is an effective choice transition iff $\xi(s) = 1 + \xi(s')$ and $\exists \rho = s' \xrightarrow{e_0} s_1 \xrightarrow{e_1} ... s_n \in run(G)$ such that $\xi(s_n) = 0$ and $(\max_{s \in \rho} \xi(s)) = \xi(s')$. When such a transition exists, the state s is called an effective choice state.*

Intuitively, an effective choice transition is a transition that decrements the level of choice and is prefix of a run ρ violating $\mathcal{P}$ along which the level of choice no longer exceeds $\xi(s')$.

Lemma 7 *Let $G = \langle \mathcal{V}, \Sigma, \rightarrow, \mathcal{V}^F, \nu_0 \rangle$ and ξ the level of choice of G. Let $\mathcal{V}^C = \{\nu \in \mathcal{V} \mid \xi(\nu) \in \mathbb{N} \setminus \{0\} \wedge \xi(\nu) = 1 + min\{\xi(\nu') \mid \nu' \in \nu^\bullet\}\}$ and $\mathcal{V}^{NC} = \{\nu \in \mathcal{V} \mid \xi(\nu) \in \mathbb{N} \setminus \{0\}\} \setminus \mathcal{V}^C$. Then $\rightarrow \cap (\mathcal{V}^{NC} \times \Sigma \times \mathcal{V}^{NC})$ is acyclic.*

The successors of a state in $\mathcal{V}^{NC}$ have the same level of choice. Lemma 7 states that from such state in $\mathcal{V}^{NC}$, any run eventually leads to a state with successors with different level of choice ($\mathcal{V}^C$) from which there is an outgoing effective choice transition. The proof is similar to the proof of Lemma 3 in Chapter 4. Indeed, the level of choice of the states in $\mathcal{V}^{NC}$ is the level of choice of the successors in $\mathcal{V}^C$, and for the same reason, the level of choice is 0 for some states because of their inevitable successors in $\mathcal{V}^F$.

Proof. Let $(\xi^i)_{i \in \mathbb{N}}$ be the iterations of the fixed-point of ξ as defined in Theorem 1. We have for some $i \in \mathbb{N}$ $\xi^i = \xi$ and for all $i \in \mathbb{N}$ and states $\nu \in \mathcal{V}, \xi^i(\nu) \geq \xi^{i+1}(\nu)$.

Let ψ a function such that $\forall \nu \in \mathcal{V}, \psi(\nu) \in \mathbb{N}$ such that $\forall j < \psi(\nu)$ $\xi^j(\nu) > \xi(\nu)$ and $\forall j \geq \psi(\nu)$ $\xi^j(\nu) = \xi(\nu)$. ψ is well defined because $(\xi^i)_{i \in \mathbb{N}}$ converges monotonically in a finite number of iterations (Theorem 1).

We prove by induction on $\psi(\nu)$ that :

$\forall \nu \in \mathcal{V}$ such that $0 < \xi(\nu) < \infty$, there exists un integer $n_\nu \in \mathbb{N}$ such that all runs from ν of length greater than n_ν contain a state ν', such that $\exists \nu'' \in \nu'^\bullet, \xi(\nu) = \xi(\nu'') + 1$.

For $\nu \in \mathcal{V}$ such that $\psi(\nu) = 0$, the property is satisfied because $\xi(\nu) = 0$.

For $\nu \in \mathcal{V}$ such that $\psi(\nu) > 0$, we consider the two cases : $\nu \in \mathcal{V}^C$ and $\nu \in \mathcal{V}^{NC}$.

For the case $\nu \in \mathcal{V}^C$, we have by definition of ξ, $\exists \nu'' \in \nu^\bullet, \xi(\nu) = \xi(\nu'') + 1$ and the integer $n_\nu = 0$ satisfies the property.

For the case $\nu \in \mathcal{V}^{NC}$, we have by definition of $\mathcal{V}^{NC}$ that

$$\forall \nu' \in \nu^\bullet, \xi(\nu') = \xi(\nu) \tag{5.1}$$

And therefore by definition of $\psi(\nu)$ and $\xi^{\psi(\nu)}$:

$$\forall \nu' \in \nu^\bullet, \xi^{\psi(\nu)-1}(\nu') = \xi^{\psi(\nu)}(\nu) \tag{5.2}$$

Hence, by combining the two equations, we obtain $\forall \nu' \in \nu^\bullet, \xi^{\psi(\nu)-1}(\nu') = \xi(\nu')$ and it implies, by definition of ψ that

$$\forall \nu' \in \nu^\bullet, \psi(\nu') < \psi(\nu) \tag{5.3}$$

We can use the induction hypothesis on all $\nu' \in \nu^\bullet$ and define $n_\nu = 1 + \max\{n_{\nu'} \mid \nu' \in \nu^\bullet\}$.

This property, proved by induction, implies the lemma because:

$\forall \nu \in \mathcal{V}^{NC}$ we have $0 < \xi(\nu) < \infty$ and $\exists n \in \mathbb{N}$ such that for some run $\nu \xrightarrow{e_1} \dots \xrightarrow{e_n} \nu_n$ from ν, let $i = \min\{j \in \mathbb{N} \mid \exists \nu'' \in \nu_j^\bullet, \xi(\nu) = \xi(\nu'') + 1\}$ and we have $\nu_i \in \mathcal{V}^C$ because $\xi(\nu) = \xi(\nu_i)$ because i is minimal and $\xi(\nu) = \xi(\nu'') + 1$.

In other words, $\rightarrow \cap (\mathcal{V}^{NC} \times \Sigma \times \mathcal{V}^{NC})$ is acyclic because for all states in $\mathcal{V}^{NC}$, there is a bound n_ν such that all the runs of length greater than n_ν contain a state $\mathcal{V}^C$.

□

Theorem 7 *Given an LTS $G = \langle \mathcal{V}, \Sigma, \rightarrow, \mathcal{V}^F, \nu_0 \rangle$, a level of choice function ξ on G, and a strong bisimulation $\sim$ on $\mathcal{V}$, we have that $s \sim s' \implies \xi(s) = \xi(s')$.*

Proof. Toward contradiction: Let us suppose that we have $G = \langle \mathcal{V}, \Sigma, \rightarrow, \mathcal{V}^F, \nu_0 \rangle$, a level of choice function ξ on G, and a strong bisimulation $\sim$ on $\mathcal{V}$ and

$$\exists \nu, \nu' \in \mathcal{V}, \nu \sim \nu' \wedge \xi(\nu) \neq \xi(\nu') \tag{H}$$

Let $n \in \mathbb{N}$ be the smallest level of choice such that there exists one state at level n which is bisimilar with a state at level $n' > n$. Formally, n is the smallest integer such that the following predicate $P(n)$ is true:

$$P(n) := \exists \nu, \nu' \in \mathcal{V}, (\nu \sim \nu') \wedge \xi(\nu) = n \wedge \xi(\nu) < \xi(\nu') \tag{P(n)}$$

The existence of n is implied by the hypothesis (H).

Case $n = 0$: Let $\nu \in \mathcal{V}$ such that $\xi(\nu) = 0$ and $\exists \nu' \in \mathcal{V}, \nu \sim \nu' \wedge \xi(\nu') > 0$. $\xi(\nu') > 0$ implies $\nu' \notin \mathcal{V}^F$ by Definition 20 of the level of choice and therefore $\nu \notin \mathcal{V}^F$ because $\nu \sim \nu'$ by Definition 15 of the STAB $\sim$.

$\xi(\nu') > 0$ implies that there exists a successors ν'' of ν' such that $\xi(\nu'') > 0$. Therefore from ν' we can build a run where $\xi^{-1}(0)$ can be avoided indefinitely (or reach a state non-coreachable w.r.t. $\mathcal{V}^F$) while from ν, all runs reach $\mathcal{V}^F$ within a bounded number of transitions. We can build two sequentially bisimilar runs, from ν and from ν' where the destinations are bisimilar but one is in $\mathcal{V}^F$ while the other destination is not.

Case $n > 0$: Because n is minimal we know that for a lower level of choice, bisimilarity implies the same level of choice:

$$\forall n' \in \mathbb{N}, n' < n \implies \forall \nu, \nu' \in \mathcal{V}, \nu \sim \nu' \wedge \xi(\nu) = n' \implies \xi(\nu) = \xi(\nu') \tag{5.4}$$

Let $\mathcal{V}_n^C = \{\nu \in \mathcal{V} \mid \xi(\nu) = n \wedge \xi(\nu) = 1 + min\{\xi(\nu') \mid \nu' \in \nu^\bullet\}\}$

Let $\nu \in \mathcal{V}$ be the closest (w.r.t. d) to $\mathcal{V}_n^C$ such that

$$\xi(\nu) = n \wedge \exists \nu' \in \mathcal{V}, (\nu \sim \nu') \wedge \xi(\nu') > n \tag{5.5}$$

ν exists because $P(n)$ is true.

Subcase $\nu \in \mathcal{V}_n^C$:

By Defintion 20 of the level of choice, $\xi(\nu)$, ν has one successor ν_{min} such that $\xi(\nu_{min}) = n-1$. Let $\nu'_{min} \in \nu'^\bullet$ such that $\nu_{min} \sim \nu'_{min}$ a successor of ν', it exists because $\nu \sim \nu'$. $\xi(\nu_{min}) < n$ therefore by induction hypothesis $\xi(\nu'_{min}) = \xi(\nu_{min}) = n - 1$.

Furthermore, by the definition of ξ, we have $\xi(\nu') \leq 1+min\{\xi(v'') \mid v'' \in \nu'^\bullet\} \leq 1+\xi(\nu'_{min}) = n$. That inequality contradicts with the inequality $\xi(\nu') > \xi(\nu)$.

Subcase $\nu \notin \mathcal{V}_n^C$:

All successors of ν are also at level n, and at least one, that we call ν_{min}, is strictly closer to $\mathcal{V}_n^C$ w.r.t. d, than ν. We have either $\nu_{min} \in \mathcal{V}_n^C$ or $(\forall \nu'_{min} \in \mathcal{V}, \nu_{min} \sim \nu'_{min} \implies \xi(\nu'_{min}) = n)$. Let ν'_{min} the successor of ν' such that $\nu_{min} \sim \nu'_{min}$, it exists because $\nu \sim \nu'$.

If $\nu_{min} \in \mathcal{V}_n^C$ then $\xi(\nu'_{min}) = \xi(\nu_{min})$ (previous case), therefore $\xi(\nu') \leq n + 1$, therefore $\xi(\nu') = n + 1$ because $\xi(\nu') > n$.

If $\nu_{min} \notin \mathcal{V}_n^C$ then $\xi(\nu'_{min}) = \xi(\nu_{min}) = n$, therefore $\xi(\nu') \leq n + 1$, therefore $\xi(\nu') = n + 1$.

$\xi(\nu') = n + 1 \wedge \xi(\nu'_{min}) = n \implies \exists \nu'_{max} \in \nu'^\bullet, \xi(\nu'_{max}) > n$ and because $\nu \sim \nu'$, $\exists \nu_{max} \in \nu'^\bullet_{max}, \nu_{max} \sim \nu'_{max}$ and because $\nu \notin \mathcal{V}_n^C$, $\xi(\nu_{max}) = \xi(\nu) = n$

For $x, y \in \mathcal{V}$, let $P(n, x, y) := x \sim y \wedge \xi(x) = n \wedge \xi(x) < \xi(y)$

We proved that $\exists \nu, \nu' \in \mathcal{V}^2, P(n, \nu, \nu') \implies \exists \nu_{max}, \nu'_{max} \in \nu^\bullet \times \nu'^\bullet, P(n, \nu_{max}, \nu'_{max})$

We can iterate until we have $P(n, x, y)$ with $x \in \mathcal{V}_n^C$. There is a bounded number of iterations because of Lemma 7 there are no cycles in $\mathcal{V}^{NC}$, especially in $\xi^{-1}(n) \cap (\mathcal{V}^{NC})$. We can then conclude with the same contradiction of the previous subcase. □

This theorem allows us to work on the LTS further reduced with respect to strong bisimulation instead of $[\mathcal{A}||\mathcal{P}||\mathcal{L}^c]$, as we know that bisimulation preserves the level of choice.

Once we have the abstraction $[\mathcal{A}||\mathcal{P}||\mathcal{L}^c]$ labeled with levels of choice, we are ready to extract the explanations. The basic idea is to extract, from $[\mathcal{A}||\mathcal{P}||\mathcal{L}^c]$, a LTS that retains only

- the states of $[\mathcal{A}||\mathcal{P}||\mathcal{L}^c]$ that are co-reachable from the final location of $\mathcal{L}^c$, *i.e.*, the states that are consistent with the observed log, and
- the edges along which the level of choice decreases, *i.e.*, the ones that bring the system closer to a violation of $\mathcal{P}$.

To this end we proceed as follows.

LTS Splitting.

There are multiple runs of the LTS $[\mathcal{A}||\mathcal{P}||\mathcal{L}^c]_\sim$ to be explained, and two runs may contain the same transition which is an effective choice transition in one run but not in the other. Those two runs have inconsistent explanations that cannot be displayed on the same graph without ambiguities. The goal of LTS splitting, is to duplicate states to make different versions of the same transition which are either effective in all runs (that contain it) or not effective in all runs. Given an acyclic LTS $G = \langle \mathcal{V}, \Sigma, \rightarrow, \mathcal{V}^F, \mathcal{V}^0 \rangle$ equipped with a level of choice function $\xi : \mathcal{V} \to \mathbb{N} \cup \{\infty\}$, we compute a split LTS as follows. For any $\nu \in \mathcal{V}$ let

$$bounds(\nu) = \begin{cases} \big\{ \max\{\xi(\nu), bounds(\nu')\} \mid \nu' \in \nu^\bullet \big\} & \text{if } \nu^\bullet \neq \emptyset \\ \{\xi(\nu)\} & \text{otherwise} \end{cases}$$

We define the split LTS $\mathcal{G} = \langle \Sigma, \mathcal{V}', \rightarrow', (\mathcal{V}')^0, (\mathcal{V}')^F \rangle$ where

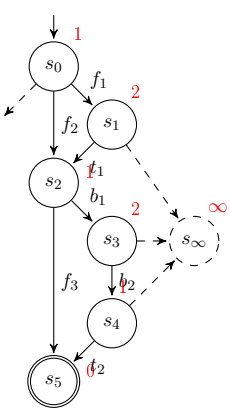
$$\begin{aligned} \mathcal{V}' &= \big\{ (\nu, b) \mid \nu \in \mathcal{V} \wedge b \in bounds(\nu) \big\} \\ \rightarrow' &= \big\{ \big((\nu, b), e, (\nu', b')\big) \mid (\nu, e, \nu') \in \rightarrow \wedge b = \max\{b', \xi(\nu)\big\} \\ (\mathcal{V}')^0 &= \big\{ (\nu, b) \in \mathcal{V}' \mid \nu \in \mathcal{V}^0 \big\} \\ (\mathcal{V}')^F &= \{ (\nu, b) \mid \nu \in \mathcal{V}^F \} \end{aligned}$$

That is, we duplicate the states according to the maximum levels of choice that may be encountered in the future, and update the edges so as to point to the matching copy. We extend ξ to the split LTS by putting $\xi\big((\nu, b)\big) := \xi(\nu)$. Intuitively, $\mathcal{G}$ accepts the same traces as G, while ensuring that each state is either effective choice or not, independent of the future behavior.

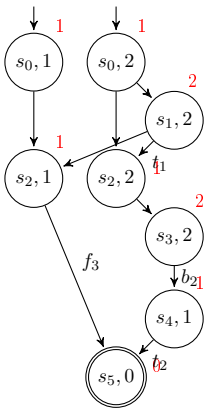
Example 8 *Consider the LTS shown in Figure 5.2a. From s_0 a fault f_1 or f_2 may occur, and f_1 may be coped with until a timeout t_1 occurs. From s_2, a second fault f_3 will entail a system failure. However, the initial fault can be handled by primary and secondary fallback mechanisms b_1 and b_2, until a timeout t_2 occurs. The split LTS is shown in Figure 5.2b.*

Subgraph Extraction.

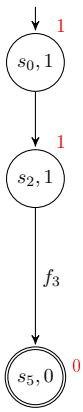
The explanations in the split LTS are bounded in length by $[\xi(\nu_0), \max \xi]$, where $\max \xi$ is the maximum finite level of choice in $\mathcal{G}$. Our experiments suggest that explanations are easier to



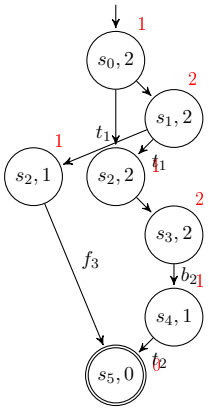
(a) LTS with levels of choice.



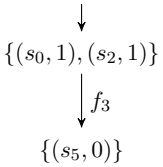
(b) Split LTS. Only labels of effective choice transitions are shown.



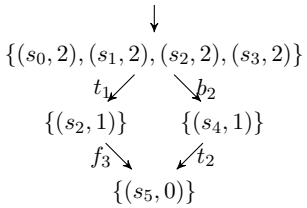
(c) Subgraph 1



(d) Subgraph 2



(e) Explanation $\mathcal{E}_1$. The shortest explanation of the system failure is the occurrence of event f_3.



(f) Explanation $\mathcal{E}_2$. The events that ultimately entailed the system failure are either $t_1; f_3$ or $b_2; t_2$.

Figure 5.2: Splitting, subgraph extraction, and explanations obtained after determinization. Dashed transitions are in the model but not consistent with the log.

grasp when only explanations of the same length are presented simultaneously. We therefore extract, from the split LTS $\mathcal{G} = \langle \Sigma, \mathcal{V}', \rightarrow', (\mathcal{V}')^0, (\mathcal{V}')^F \rangle$, for $l \in [\xi(\nu_0), \max \xi]$, the subgraph $\mathcal{G}_l$ of explanations of length l by restricting the LTS $\langle \Sigma, \mathcal{V}'', \rightarrow'', (\mathcal{V}'')^0, (\mathcal{V}'')^F \rangle$ where

$$
\begin{aligned}
\mathcal{V}'' &= \{(\nu, b) \in \mathcal{V}' \mid b \leq l\} \\
\rightarrow'' &= \rightarrow' \cap (\mathcal{V}'' \times \Sigma \times \mathcal{V}'') \\
(\mathcal{V}'')^0 &= \{(\nu, b) \in (\mathcal{V}')^0 \mid b = l\} \\
(\mathcal{V}'')^F &= (\mathcal{V}')^F \cap \mathcal{V}''
\end{aligned}
$$

to the states that are reachable from $(\mathcal{V}'')^0$, and from which some state in $(\mathcal{V}'')^F$ is reachable. Notice that $\mathcal{G}_l$ is the empty LTS when there is no explanation of length l. We then construct, for each (non-empty) subgraph, an explanation by applying the standard determinization (τ-elimination) algorithm based on subset construction [1] to "collapse" the non-pertinent parts of

the subgraph.

Example 9 *From the split LTS of Figure 5.2b, two subgraphs of constant height, in terms of effective choice transitions, are extracted (Figures 5.2c and 5.2d). By determinization we obtain the explanations $\mathcal{E}_1$ and $\mathcal{E}_2$ of Figures 5.2e and 5.2f that highlight the decisive events for disjoint scenarios, with increasing complexity of the explanation.*

Theorem 8 *For each trace $w \in trace(\mathcal{G})$ there exists $l \in [\xi(\nu_0), \max \xi]$ such that $w \in trace(\mathcal{G}_l)$.*

Proof. Let $n \in \mathbb{N}$ and $w = e_1e_2e_3...e_n \in trace(\mathcal{G})$. Let $\rho = \nu_0 \xrightarrow{e_1} \nu_1 \xrightarrow{e_2} ... \xrightarrow{e_n} \nu_n \in run(\mathcal{G})$. Let $\langle b_0, b_1, ..., b_n \rangle \in \mathbb{N}^{n+1}$ such that $\forall i \in [0,n], b_i = \max\{\xi(\nu_j) \mid j \in [i,n]\}$. Let $l = \max\{\xi(\nu_i) \mid i \in [0,n]\}$. We prove that $\rho' = (\nu_0, b_0) \xrightarrow{e_1} (\nu_1, b_1) \xrightarrow{e_1} ... \xrightarrow{e_n} (\nu_n, b_n) \in run(\mathcal{G}_l)$.

First of all, let us prove that the states in the run ρ' are in the split LTS $\mathcal{G}' = \langle \mathcal{V}', \Sigma, \rightarrow', (\mathcal{V}')^F, (\mathcal{V}')^0 \rangle$. We know that $\forall i \in [0,n], \nu_i \in \mathcal{V}$, and therefore $\big(\forall i \in [0,n], b_i \in bounds(\nu_i)\big) \implies \big(\forall i \in [0,n], (\nu_i, b_i) \in \mathcal{V}'\big)$.

By induction on $n - i$, we prove that $\forall i \in [0,n], b_i \in bounds(\nu_i)$.

For $i = n$, $\rho \in run(\mathcal{G}) \implies \nu_n \in \mathcal{V}^f \implies bounds(\nu_n) = \{0\}$ and $\nu_n \in \mathcal{V}^f \implies \xi(\nu_n) = 0$, hence $b_n = 0 \in bounds(\nu_n)$.

For $i < n$, $b_i = max\{\xi(\nu_i), \xi(\nu_{i+1}), ..., \xi(\nu_n)\} = max(\xi(\nu_i), max\{\xi(\nu_{i+1}), ..., \xi(\nu_n)\}) = max(\xi(\nu_i), b_{i+1})$. By induction hypothesis we know that $b_{i+1} \in bounds(\nu_{i+1})$. Because $\nu_{i+1} \in \nu_i^\bullet$ then by construction $b_i = max(\xi(\nu_i), b_{i+1})$. We have that $\forall i \in [0,n], b_i \in bounds(\nu_i)$ and therefore the states in ρ' are state of the split LTS.

Now let us prove that the transitions of ρ' are also in the split LTS. $\forall i \in [1,n], (\nu_{i-1}, b_{i-1}) \xrightarrow{e_i} (\nu_i, b_i) \in \rightarrow'$ because $\nu_{i-1} \xrightarrow{e_i} \nu_i \in \rightarrow$ and $b_i = max(b_{i+1}, \xi(\nu_i))$

Let $\mathcal{G}_l = \langle \Sigma, \mathcal{V}'', \rightarrow'', (\mathcal{V}'')^0, (\mathcal{V}'')^F \rangle$ be the given a subgraph for the explanation of length l.

$\forall i \in [1,n], (\nu_{i-1}, b_{i-1}) \xrightarrow{e_i} (\nu_i, b_i) \in \rightarrow'' = \rightarrow' \cap (\mathcal{V}'' \times \Sigma \times \mathcal{V}'')$ because $\nu_{i-1} \xrightarrow{e_i} \nu_i \in \rightarrow'$ and $b_i \leq l$ by definition of l and b_i.

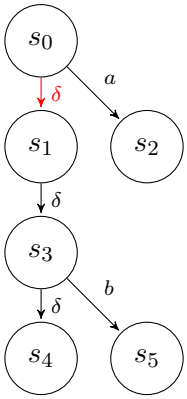
The run ρ' is in $run(\mathcal{G}_l)$ and therefore $w \in trace(\mathcal{G}_l)$. □

This result means that any log-consistent violation is contained in some subgraph after split and extraction.

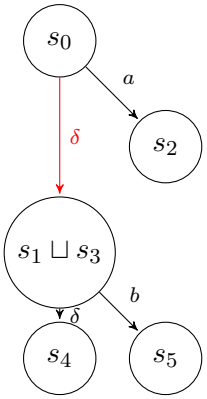
5.2.1 Further Improvements

Compressing δ-sequences and estimating time delays.

Our explanations may still encompass sequences of discretized time delays whose intermediate states are distinguished by the bisimulation. Our motivation for eliminating such sequences of delays is twofold. First, to construct a more concise explanation. In the case study, compressing δ-sequences allows us to reduce the number of transitions in the explanation from 21 to 7. Second, compressing sequences of time delays allows us to quantitatively estimate the possible time delays of the concrete runs that are summarized by the explanation.



(a) Effective choice transition in red, a and b are discrete events, the transitions in black are either effective choice or not.



(b) δ^+-compression.

Figure 5.3: Compression of sequences of discretized time delays.

Given a subgraph $\mathcal{G}_l = \langle \Sigma, \mathcal{V}, \rightarrow, \mathcal{V}^0, \mathcal{V}^F \rangle$, a δ^+-*sequence* is an atomic sequence of transitions $\sigma = \nu_1 \xrightarrow{\delta} \nu_2 \xrightarrow{\delta} ... \xrightarrow{\delta} \nu_{n+1}$, that is,

$$\forall i \in \{1, ..., n\}\ \forall e \in \Sigma\ \forall \nu' \in \mathcal{V} : \left(\nu_i \xrightarrow{e} \nu' \implies e = \delta \wedge \nu' = \nu_{i+1}\right)$$

such that $\nu_1 \xrightarrow{\delta} \nu_2$ is an effective choice transition.

As illustrated in Figure 5.3, we replace each maximal δ^+-sequence σ with a single transition $\nu_1 \xrightarrow{\delta} \nu_{n+1}$. The second condition requires the first transition of the sequence to be effective choice, which ensures the information about effective choice states, used in the sequel, to be preserved.

In order for the explanation to convey quantitative timing information, we estimate, for each abstract time transition $\nu \xrightarrow{\delta} \nu'$, the range of concrete delays represented by the transition, given the location invariants in ν and ν'. For each state s and clock c, there exists the constants $\inf_c^s, \sup_c^s \in \mathbb{N} \cup \{\infty\}$ such that in s, $\inf_c^s \leq c \leq \sup_c^s$. If s' is a time successor of s then we can estimate the delay to be between $\delta_{\text{inf}} = \max_{c \in C}(\inf_c^{s'} - \sup_c^s)$ and $\delta_{\text{sup}} = \min_{c \in C}(\sup_c^{s'} - \inf_c^s)$. In the case where δ_{inf} is negative it is set to 0.

Safe alternatives.

The rationale of our construction of explanations is to highlight the events that contributed to the violation of a safety property. Complementary to this information, and equally crucial for understanding how the property was violated, is the question "how could the outcome have been avoided?". Providing this information is the goal of *safe alternatives*.

Definition 34 (Safe alternative) *A transition $\nu \xrightarrow{e} \nu'$ of an LTS $\mathcal{G}_l$ is a safe alternative iff ν is an effective choice state and $\xi(\nu') \geq \xi(\nu)$.*

Intuitively, given a choice state, a safe alternative is a transition that would have contributed to avoiding the violation by not decreasing the level of choice.

State constraints.

So far we have focused our attention on the *events* of a failing run. The complementary information crucial for understanding the outcome, is in which *states* some relevant event took place. In this section, let us assume the TA to be equipped with a function $\pi : L \to 2^{\Pi}$ that labels each location with a set of atomic propositions. A straight-forward approach for displaying the states in the explanation would be to compute, for each aggregate state of the determinized subgraph consisting of a set Q of locations, the disjunction of the invariants (resp. of the atomic proposition) of the locations in L. This would, however, lead to unreadably complex expressions.

We therefore make the design choice to label each state $q = \{\nu_1, ..., \nu_k\} \in Q$ returned by determinization, with a *convex* predicate of the form $\nu_1 \sqcup \cdots \sqcup \nu_k := \hat{C}(q) \wedge SP(q)$, where $\hat{C}(q)$ is the weakest convex clock constraint that is implied by the invariants of the locations of all effective choice states in q. It is straight-forward to compute this clock constraint from the DBMs of the involved location invariants.

Similarly, concerning the function SP that aggregates, for a state q, atomic propositions of the states in q, multiple definitions are possible. We settle for the conjunction of the atomic propositions that hold in all effective choice states in q. This set is therefore obtained as:

$$SP(q) = \bigcap_{\nu_i \in q:\ \nu_i \text{ is an effective choice state}} \ \bigcap_{\ell \in \nu_i} \pi(\ell)$$

An example of the obtained state predicates is shown in Figure 5.7.

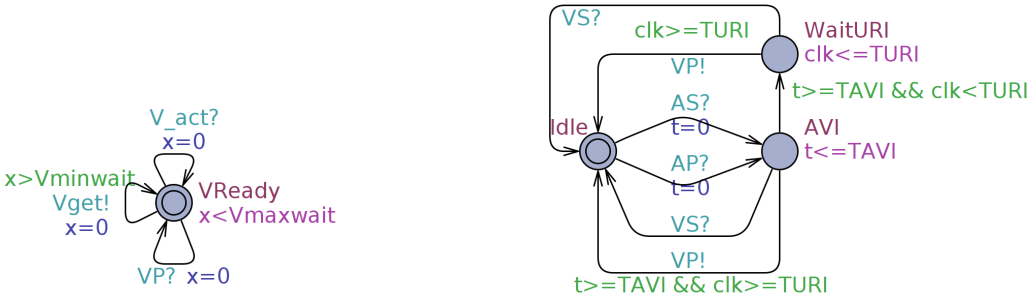
5.3 Implementation and Case Study

We have implemented our results in a tool written in Python. It relies on Kronos [41] for the composition of timed automata, Minim [38] for the generation of the quotient graph, and CADP [14] for reductions up to bisimulation.

We illustrate our approach on the dual-chamber implantable pacemaker model of [22]. It is a multi-component system where components are timed automata communicating over channels [26]. We use the model of [25] that we have translated into the Kronos format. The model consists of 5 timed automata for the components of the pacemaker and 2 timed automata that model its environment, that is, the atrial and ventricular behavior of a heart. Whenever delays between sensed atrial or ventricular events exceed a threshold, the pacemaker produces an AP (atricular pace) or VP (ventricular pace) event.

Figure 5.4a shows the model of the ventricular behavior, with a frequency between $Vminwait =$ 500 ms and $Vmaxwait = 1100$ ms, so as to allow a for fault. On the pacemaker model we increase the upper rate interval $TURI$ from 400 ms to 1600 ms. This parameter determines the minimum delay between two ventricular events VP in the AVI component shown in Figure 5.4b. In order to compare our results, we have taken the same parameters as in [25].

Among the safety properties discussed in [22] we focus on the requirement that the time between two ventricular events (sensed or paced) never exceeds 1000 ms. The safety property



(a) Heart ventricular component.

(b) Pacemaker ventricular component AVI.

Figure 5.4: Two components of the model.

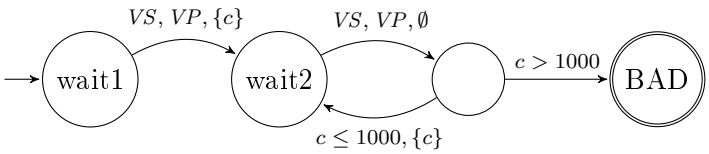


Figure 5.5: Safety property observer.

observer is shown in Figure 5.5. If the heart model is safe then the system is also safe. We have therefore modified the parameters of the pacemaker so as to allow for unsafe behaviors.

We fix the set of observable events as the set $\Sigma^{obs} = \{VP, VS, AS, AGET, AP, VGET\}$ of signals exchanged between the components, whereas we consider all internal events of the components as unobservable. We use Uppaal to obtain a witness trace for the violation of the property, from whose projection on the observable events we construct the timed log shown in Figure 5.6.

Let us now apply our approach to explain the causes of the violation. The sizes of the timed automata, discrete abstractions, and explanation are shown in Table 5.1.

1. The first step is to generate the log observer, which consists of 8 locations and 60 transitions.

2. We invoke Kronos to compose the components, the safety property observer, and the log observer.

3. We use Minim to compute the quotient graph with respect to strong time-abstracting bisimulation.

4. The levels of choice and safe alternatives are computed.

5. We remove the states that are not consistent with the log or the property violation. The difference in size is important because the observed behavior is only a small part of the behavior of the model.

6. In this case study the effective choice states are not ambiguous, therefore splitting does not change the LTS. Similarly, the extracted subgraph amounts to the full graph here.

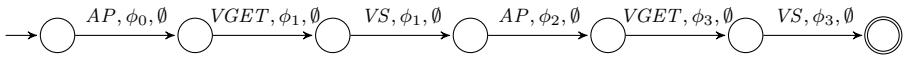


Figure 5.6: Timed log. All location invariant are *true*, $\phi_0 = (x = 850)$, $\phi_1 = (850 \leq x \leq 1000)$, $\phi_2 = (1700 \leq x \leq 1800)$, $\phi_3 = (1850 \leq x \leq 2200)$.

Automaton	#states	#transitions
$\mathcal{A}\|\|\mathcal{P}$	96	296
$\mathcal{L}$	7	6
$\mathcal{L}^c$	8	60
$\mathcal{A}\|\|\mathcal{P}\|\|\mathcal{L}^c$	117	332
$[\mathcal{A}\|\|\mathcal{P}\|\|\mathcal{L}^c]_\sim$	1817	5041
$[\mathcal{A}\|\|\mathcal{P}\|\|\mathcal{L}^c]_\sim / \sim$	1763	4931
log-consistent	50	52
split	50	49
δ^+-compression	35	34
explanation	8	7

Table 5.1: Sizes of the timed automata (number of locations), discrete abstractions, and resulting explanation.

7. Two maximal δ^+-sequences of length 11 (resp. 6) are found, encompassing 10 (resp. 6) effective choice transitions. After δ^+-compression, the quantitative time delays along the abstract delay transitions (called "time_succ" in our implementation) are computed.

8. After determinization we obtain the explanation shown in Figure 5.7.

In our case, the explanation is a sequence of discrete and δ-transitions. Each of them moves the system closer to the violation of the safety property, in spite of safe alternatives (shown as transitions without a target state) that would have avoided the violation. In our case the safe alternatives are mostly events sent by the heart model. This is logical because we know that *VGET* from the heart induces immediately a ventricular event in the *PVRP* component (not shown here) that would have avoided the violation. In order to understand why the pacemaker fails to adjust the ventricular events rate, we need to look at the state predicates. If we focus on the last three transitions, we see that the pacemaker waits 150 ms and takes an internal transition *I_AVI*. If we look at the model of the *AVI* component, we see that this event could only occur because we increased *TURI*. From this point the violation becomes unavoidable after a further delay of 400 ms.

Comparison with TarTar [25]. We use the same model as in [25], up to the fact that we have increased the parameters *Aminwait* and *Vminwait* (the minimal atrial and ventricular rate of the heart model, respectively) from 1 ms to 500 ms in order to reduce the size of the quotient graph to a size that can be managed by Minim. This modification does not impact the safety violation we are interested in.

TarTar focuses on fixing time delay parameters in order to repair safety violations, and proposes a repair of the bounds on *TURI*. Our approach is more general in the sense that

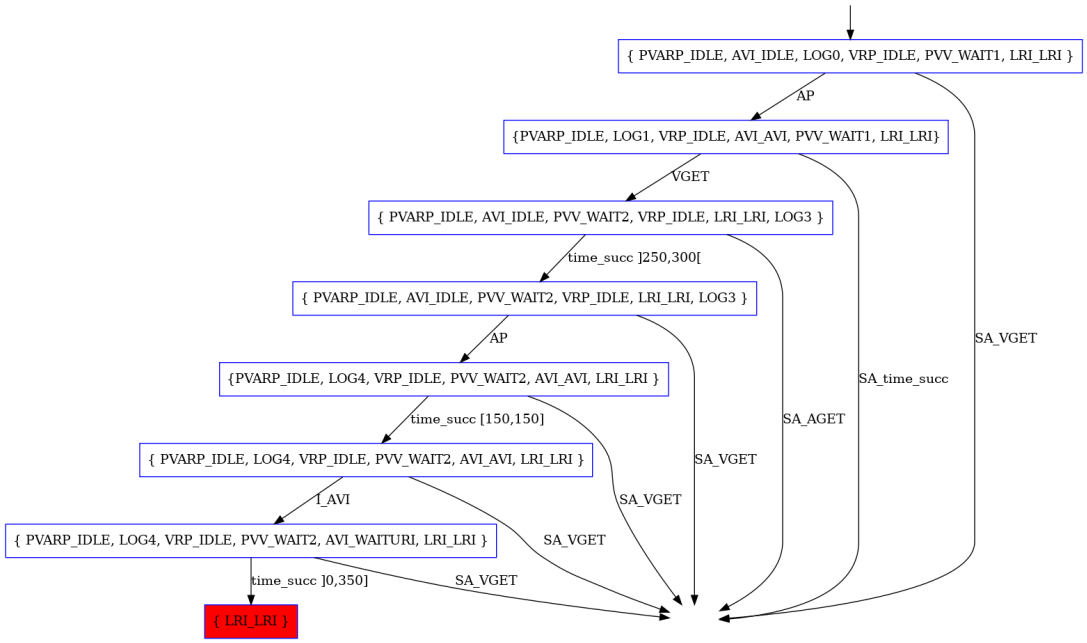


Figure 5.7: Pacemaker explanation

it does not restrict its attention to time delays. On the other hand, it does not propose a repair. In particular, our explanations are useful to explain failures caused by nondeterministic behavior, or when there are no admissible repairs but one still wants to understand the causes of a violation.

Chapter 6

Hybrid explanations on TA

Resumé

Dans ce chapitre, nous nous concentrons sur les explications causales pour les systèmes temps réel modélisés comme des automates temporisés.

Étant donné un modèle d'automate temporisé A, une propriété de sécurité P, et une trace tr qui viole P, notre objectif est de définir une fonction de robustesse qui mesure, pour chaque état de A visité par tr, à quel point le système est proche de violer P. Dans le chapitre précédent, nous avons proposé une telle fonction que nous avons appliquée à une abstraction discrète de A. Cependant, cette étape de discrétisation conduit à un problème que nous appelons "mauvaise surprise", illustré dans la Figure 6.1. Pour modéliser l'urgence de prendre une transition discrète afin d'éviter une violation de la propriété de sûreté, nous voulons que la robustesse du système diminue continuellement au fil du temps lorsque les délais se rapprochent de la violation. Pour modéliser l'urgence de prendre une transition discrète pour éviter une violation de la propriété de sécurité, nous voulons que la robustesse du système diminue de manière continue dans le temps lorsque les délais se rapprochent de la violation. À cette fin, l'utilisation d'une fonction de robustesse avec un co-domaine discret n'est pas possible, en raison des discontinuités temporelles.

Le terme "explications hybrides" désigne les explications qui prennent en compte à la fois les causes discrètes, avec des effets discrets sur la robustesse, et les causes de retard, avec des effets continus sur la robustesse. Afin d'exclure ce phénomène, ainsi que d'autres comportements indésirables des explications, nous commençons par formaliser un ensemble d'exigences formelles sur les fonctions de robustesse. Des exemples de ces exigences cruciales sont *controllability of causes* garantissant que la robustesse se comporte bien sur les systèmes ouverts avec un environnement incontrôlable, et *safe alternatives* garantissant que seuls les *mauvais choix* faits par le système diminuent la robustesse, alors que les comportements forcés ne le font pas.

Ces exigences caractérisent une famille de fonctions de robustesse adaptées à la construction d'explications causales. Intuitivement, une fonction de robustesse mesure la capacité du système à éviter la violation de P. Nous définissons ensuite une *cause contributive* comme une transition dans tr qui diminue la robustesse de façon monotone. Ainsi, chaque cause contributive rapproche le système de la violation de P. Notre explication de la violation de P par tr est la séquence

des causes contributives survenant dans r.

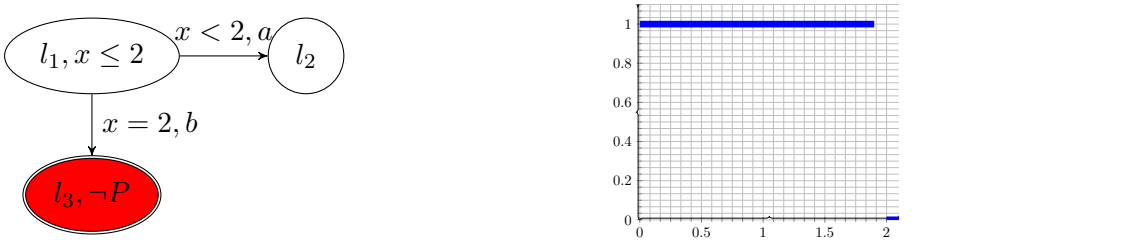
Dans ce chapitre, nous apportons les principales contributions suivantes.

- Nous formalisons et discutons un ensemble d'exigences formelles sur les fonctions de robustesse (Section 6.3).
- Nous proposons une famille concrète de fonctions ρ_κ et prouvons qu'elles sont effectivement des fonctions de robustesse (Section 6.4).
- Nous illustrons la construction d'explications causales à partir de ρ_κ sur un exemple simple (Section 6.5).

6.1 Introduction

In this chapter, we focus on causal explanations for real-time systems modeled as timed automata. Given a timed automaton model A, a safety property P, and a trace tr that violates P, our goal is to define a robustness function that measures, for each state of A visited by tr, how close the system is to violating P. In the previous chapter we proposed such a function that we applied to a discrete abstraction of A. However, this discretization step leads to an issue that we call "bad surprise", illustrated in Figure 6.1. To model the urgency of taking a discrete transition in order to avoid a violation of the safety property, we want the robustness of the system to decrease continuously over time when delays lead closer to the violation. To model the urgency of taking a discrete transition in order to avoid a violation of the safety property, we want the robustness of the system to decrease continuously over time when delays lead closer to the violation. For this purpose, using a robustness function with a discrete co-domain is not feasible due to time discontinuities.

The term hybrid explanations denotes explanations that into take account both discrete causes, with discrete effects on robustness, and delay causes, with continuous effects on robustness.



(a) Timed Automaton

(b) Discrete Robustness $\xi(\langle \ell_1, x \rangle)$ [29]

Figure 6.1: Bad surprise: a discrete-valued robustness function does not faithfully model the urgency for the system to take a discrete transition, in order to avoid the violation of P.

In order to rule out this phenomenon, and other undesirable behaviors of explanations, we start with formalizing a set of formal requirements on robustness functions. Examples of such crucial requirements are *controllability of causes* ensuring that robustness is well-behaved on

open systems with an uncontrollable environment, and *safe alternatives* ensuring that only *bad choices* made by the system decrease robustness, whereas forced behaviors do not.

These requirements characterize a family of robustness functions that are fit to construct causal explanations. Intuitively, a robustness function measures the capacity of the system to avoid the violation of P. We then define a *contributory cause* as a transition in tr that monotonically decreases robustness. Hence, each contributory cause brings the system closer to the violation of P. Our explanation of tr violating P is the sequence of contributory causes occurring in r.

In this chapter we make the following main contributions.

- We formalize and discuss a set of formal requirements on robustness functions (Section 6.3).
- We propose a concrete family of functions ρ_κ and prove that they are indeed robustness functions (Section 6.4).
- We illustrate the construction of causal explanations from ρ_κ on a simple example (Section 6.5).

6.2 Preliminaries

We require that the model $\mathcal{A}$ is non-zeno: on every sufficiently long run, time diverges. This implies that there are no self-loops in the STAB-quotient graph $[\mathcal{A}]_\sim$. The objective of this hypothesis is to rule out, in the computation of robustness functions, timed strategies that rely on zeno behaviors.

The *falling edge* $z^\downarrow$ of a zone z is the set of states:

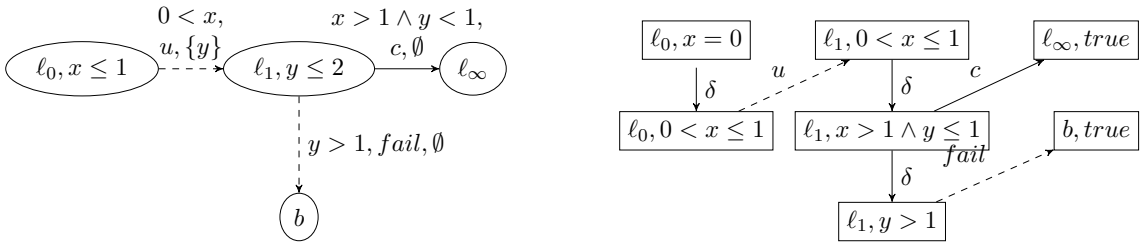
$$z^\downarrow := \{v \in z \mid \exists \epsilon > 0, \forall \epsilon' \in]0, \epsilon], v + \epsilon' \notin z\} \cup \{v \notin z \mid \exists \epsilon > 0, \forall \epsilon' \in]0, \epsilon], v - \epsilon' \in z\} \qquad (6.1)$$

We define for each zone z, the distance $d(z) : dom(X) \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ to $z^\downarrow$ such that $\forall \langle \ell, v\rangle \in z, \langle \ell, v\rangle + d(z)(v) \in z^\downarrow$. In case where $z^\downarrow = \emptyset$, $d(z)(v) = \infty$. The function $reset : Z \times \Sigma \to 2^X$ is defined such that $reset(z, e)$ is the set of clocks X' in the TA edge $E(\langle \ell, e\rangle) = \langle \ell', g, X'\rangle$, where ℓ is the location of z.

We define the controllability of the system by adapting the formalism of timed strategies in [39]. The events (resp. transitions) are split in a partition $\Sigma = \Sigma_c \uplus \Sigma_u$ (resp. $\to_c \uplus \to_u$) of *controllable* and *uncontrollable* events (resp. transitions).

Definition 35 (Timed strategies) *Let $\mathcal{A} = \langle \Sigma, L, X, \mathcal{I}, E\rangle$ be a TA, $\Sigma = \Sigma_c \uplus \Sigma_u$ a partition of events, $[\mathcal{A}] = \langle \mathcal{V}, \Sigma \cup \mathbb{R}_{>0}, \to\rangle$, a set $\hat{\mathcal{V}} \subset \mathcal{V}$, and a state $\nu_0 \in \mathcal{V}$. A timed strategy w.r.t. invariance of $\hat{\mathcal{V}}$ is a subgraph $\mathcal{G}_s = \langle \mathcal{V}_s, \Sigma \cup \mathbb{R}_{>0}, \to_s\rangle$ of $[\mathcal{A}]$ with continuity ($\nu \xrightarrow{t} \nu + t \implies \forall \tau \in [0, t], \nu \xrightarrow{\tau} \nu + \tau$) and additivity ($\xrightarrow{t_1}\xrightarrow{t_2} = \xrightarrow{t_1+t_2}$) such that $\nu_0 \in \mathcal{V}_s$, $\mathcal{V}_s \subset \hat{\mathcal{V}}$, and*

- $\forall \nu \in \mathcal{V}_s, \forall e \in \Sigma_u, \nu' \in \mathcal{V}, \nu \xrightarrow{e}_u \nu' \implies \nu \xrightarrow{e}_{us} \nu'$, *that is the strategy is closed under $\to_u$*
- $\forall \nu \in \mathcal{V}_s, \exists t \in \mathbb{R}_{\geq 0}, \left(\nu \xrightarrow{t} \nu + t \implies \nu \xrightarrow{t}_s \nu + t\right) \vee \left(\exists t' < t, e \in \Sigma_c, \nu'' \in \mathcal{V}_s, \nu \xrightarrow{t'}_s \nu' \xrightarrow{e}_{cs} \nu''\right)$



(a) TA $\mathcal{A}$ (b) STAB-quotient graph $[\mathcal{A}]_\sim$

Figure 6.2: Model and its quotient graph. Uncontrollable transitions are dashed.

z	ℓ_0, $x = 0$	ℓ_0, $0 < x \leq 1$	ℓ_1, $0 < x \leq 1$	ℓ_1, $x > 1 \wedge y \leq 1$	ℓ_1, $y > 1$	ℓ_∞, $true$	b, $true$
Robustness ρ	0	x	$x - y$	$1 - y$	0	∞	0

Table 6.1: Robustness of each system state.

We call a state $\nu \in \mathcal{V}_s$ a winning state.

The winning states characterize the states from which the system is able to avoid a set of locations B. We focus on the prefix of the run containing only winning states, from where a property violation is caused by a (bad) choice made by the system. The goal of the explanation is thus to identify these choices.

6.3 Robustness of Choice

Given a TA $\mathcal{A}$ of which the set of discrete events is partitioned into *controllable* and *uncontrollable* events, a successor-closed subset B of *bad locations* representing an expected safety property, and a run $r \in run(\mathcal{A})$ reaching B that is violating the safety property, our goal is to construct an explanation of *why* this run r reaches B. To this end, we first compute a *robustness of choice function*.

Before formalizing the requirements characterizing a robustness function, we illustrate our approach with the computation of an explanation, given a robustness function.

6.3.1 Illustrative Example

We illustrate the construction of an explanation from a robustness of choice function on the TA in Figure 6.2a. The set of bad locations is $\{b\}$. There are two clocks x and y. Once $y > 1$, the uncontrollable transition $fail$ takes the system to b. Let us consider the run r of the trace $0.5 \cdot u \cdot 1.2 \cdot fail$ where the real numbers denote delays. We consider the robustness function ρ shown in Table 6.1, for each zone of the quotient graph (Figure 6.2b).

Let us now motivate the robustness function of Table 6.1. From any state in the zone $\langle \ell_\infty, true \rangle$, the location b is not reachable, hence the robustness ρ in $\langle \ell_\infty, true \rangle$ is ∞. In the zone $\langle b, true \rangle$, the robustness ρ is 0 because b is a bad location. From a states $\langle \ell_1, v \rangle$ with $v(y) > 1$, the system cannot avoid b because $\langle \ell, v \rangle$ is a not winning state. Hence $\rho(\langle \ell, v \rangle) = 0$.

steps	0.5		$u \cdot 0.5$		0.5	$0.2 \cdot fail$	
zones	ℓ_0, $x = 0$	ℓ_0, $0 < x \leq 1$	ℓ_0, $0 < x \leq 1$	ℓ_1, $0 < y \leq 1$	ℓ_1, $x > 1 \wedge y \leq 1$	ℓ_1, $y > 1$	b, *true*
ρ	$0 \nearrow 0.5$		0.5		$0.5 \searrow 0$	0	

Table 6.2: Evolution of robustness ρ along the trace. Colors depend on the evolution of ρ.

For any state $\nu \in \langle \ell_1, x > 1 \wedge y \leq 1 \rangle$, there is a controllable transition $\nu \xrightarrow{c}_c \nu'$ that allows the system to avoid the bad locations. Concurrently, the delays from ν lead to non-winning states in $\langle l_1, y > 1 \rangle$. Hence, to avoid reaching b the system should not let time pass and should take the controllable transition. In that regard, the robustness function $1 - y$ decreases continuously with a rate of -1.

From the zone $\langle \ell_1, 0 < x \leq 1 \rangle$ there are no discrete transitions and the system is forced to wait $(1 - x)$ and reach $\langle \ell_1, 0 < x \leq 1 \rangle$ with the robustness $1 - (y + (1 - x)) = x - y$. Since there is no alternative, ρ remains constant. The actions of the environment, uncontrollable discrete transitions, or delays from states with no outgoing controllable transition are not causes, since they are not avoidable. Our explanations contain only controllable choices made by the system which reduce robustness.

A delay in the zone $\langle \ell_0, 0 < x \leq 1 \rangle$ increases the time spent in the zone $\langle \ell_1, x > 1 \wedge y \leq 1 \rangle$ in which the system can take a controllable transition to avoid b. From a state $\nu \in \langle \ell_0, 0 < x \leq 1 \rangle$, the transition $\nu \xrightarrow{u}_u \nu'$ stops such delays and $\rho(\nu') = x - y$ with $y = 0$. Because $\nu \xrightarrow{u}_u \nu'$ is uncontrollable and cannot be avoided, it does not decrease ρ and therefore $\rho(\nu) = \rho(\nu')$.

For all states in the zone $\langle \ell_0, x = 0 \rangle$, the robustness ρ is 0, since the environment can lead the system arbitrarily close to b by taking u as soon as possible, such that the zone $\langle \ell_1, x > 1 \wedge y \leq 1 \rangle$ is reached with y arbitrarily close to 1.

We now construct an explanation for the following run r reaching b :

$$r = \langle \ell_0, \{x \mapsto 0, y \mapsto 0\} \rangle \xrightarrow{0.5} \xrightarrow{u}_u \xrightarrow{1.2} \xrightarrow{fail}_u \langle b, \{x \mapsto 1.7, y \mapsto 1.2\} \rangle.$$

We split r into monotonic sequences of transitions with respect to ρ. The color *green* is used for monotonic increasing, *black* for constant, and *red* for monotonic decreasing sequences, as shown in Table 6.2.

The explanation why r reaches b is the set of transitions in r decreasing the robustness ρ : $\{\langle \ell_1, x \mapsto 1, y \mapsto 0.5 \rangle \xrightarrow{0.5} \{\langle \ell_1, x \mapsto 1.5, y \mapsto 1 \rangle\}$.

6.3.2 Expected Properties

In this section, we define the properties a robustness of choice function is expected to satisfy, in order to construct concise and meaningful explanations. Given a run r that enters a set B of bad locations and a robustness of choice function ρ with respect to B, we define a *contributory cause* with respect to ρ as a transition $\nu \xrightarrow{e} \nu'$ in r with $e \in \Sigma \cup \mathbb{R}_{>0}$ such that:

- $\rho(\nu) > \rho(\nu')$, that is this transition $\nu \xrightarrow{e} \nu'$ decreases the robustness ρ, and

- for any state ν'' in the suffix of the run r after this transition, $\rho(\nu') \geq \rho(\nu'')$, which means that the decrease of robustness from $\rho(\nu)$ to $\rho(\nu')$ is *effective* in r.

A cause is a decision of the system, in a given run, that effectively reduces its robustness. This means that, following the occurrence of the cause, its effects are not canceled out by an increase in robustness. Our goal is thus to construct an explanation in terms of a sequence of contributory causes, which are transitions in the run at hand that altogether entail the violation. The following properties are derived from properties of the level of choice and effective choice transitions on discrete event systems and adapted to real-time systems. We establish in the sequel a set of formal requirements for robustness of choice functions. Each requirement is justified in light of the ability to explain the failure.

Given a TA $\mathcal{A} = \langle \Sigma, L, X, \mathcal{I}, E \rangle$ with $[\mathcal{A}] = \langle \mathcal{V}, \Sigma \cup \mathbb{R}_{>0}, \rightarrow_c \cup \rightarrow_u \cup \rightarrow \rangle$, a set $B \subseteq L$ of bad locations, a partition $\Sigma = \Sigma_c \uplus \Sigma_u$ of controllable and uncontrollable events, and a constant $\kappa \in \mathbb{N}_{>0}$, a function $\rho : \mathcal{V} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ is a robustness of choice function if it satisfies the following properties.

Soundness. For all $\nu \in \mathcal{V}$:

$$\rho(\nu) > 0 \implies \nu \text{ is winning} \wedge \qquad (\textit{soundness } 1)$$

$$\rho(\nu) = \infty \iff B \text{ is not reachable from } \nu \qquad (\textit{soundness } 2)$$

Intuitively, *soundness* requires ρ to correctly characterize winning states w.r.t. invariance of $(L \setminus B) \times dom(X)$. For a state $\nu \in \mathcal{V}$ such that $\rho(\nu) > 0$, the system can avoid B. The states that are not winning have robustness 0. Notice that some winning states may have robustness 0, as justified by Example 6.2 for the states in the zone $\langle \ell, x = 0 \rangle$.

Controllability of causes.

$$\forall \nu, \nu' \in \mathcal{V}\ \forall a \in \Sigma_u \text{ s.t. } \nu \xrightarrow{a}_u \nu', \rho(\nu) \leq \rho(\nu') \qquad (\textit{controllability of causes})$$

Controllability of causes ensures that uncontrollable transitions do not decrease robustness, and therefore are not considered as contributory causes. This requirement ensures that explanations focus on the actions made by the system, and not those of the environment.

Safe and unsafe alternatives.

Definition 36 (CSA, UA) *Let CSA and UA be the predicates on $\mathcal{V}$ such that:*

$$\begin{aligned} &\exists \nu' \in \mathcal{V}\ \exists e \in \Sigma \cup \mathbb{R}_{>0}\ s.t.\ \nu \xrightarrow{e} \nu' \wedge \rho(\nu) > \rho(\nu') \\ &\quad \implies \exists a \in \Sigma_c\ \exists \nu'' \in \mathcal{V}\ s.t.\ \nu \xrightarrow{a}_c \nu'' \wedge \rho(\nu) \leq \rho(\nu'') \qquad (CSA(\nu)) \\ &\exists \nu' \in \mathcal{V}\ \exists e \in \Sigma \cup \mathbb{R}_{>0}\ s.t.\ \nu \xrightarrow{e} \nu' \wedge \rho(\nu) < \rho(\nu') \\ &\quad \implies \exists a \in \Sigma \cup \mathbb{R}_{>0}\ \exists \nu'' \in \mathcal{V}\ s.t.\ \nu \xrightarrow{a} \nu'' \wedge \rho(\nu) \geq \rho(\nu'') \qquad (UA(\nu)) \end{aligned}$$

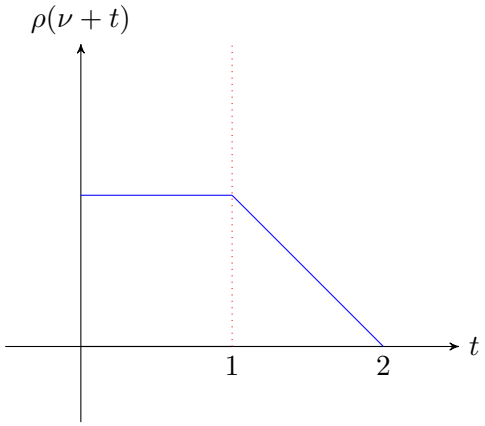


Figure 6.3: Robustness $\rho(\nu + t)$ of the states in a delay $\nu \xrightarrow{2} \nu + 2$ where for $t > 1$, there is a discrete controllable transition from $\nu + t$ that increases ρ and for $t \in [0, 1]$, there are no discrete transitions from $\nu + t$.

The predicate $CSA(\nu)$ requires that whenever a transition from ν decreases robustness, there is also a controllable transition from ν, called safe alternative, that does not decrease robustness. The predicate $UA(\nu)$ requires that whenever a transition from ν increases robustness, there is also a transition from ν, called unsafe alternative, that does not increase robustness.
Safe alternatives.

$$\forall \nu \in \mathcal{V}\ \epsilon \in \mathbb{R}_{>0}\ \exists t \in \mathbb{R}_{\geq 0} \text{ s.t. } \nu \xrightarrow{t} \nu + t \wedge \rho(\nu) - \epsilon \leq \rho(\nu + t) \wedge CSA(\nu + t) \quad (\textit{safe alternatives})$$

Unsafe alternatives.

$$\forall \nu \in \mathcal{V}\ \epsilon \in \mathbb{R}_{>0}\ \exists t \in \mathbb{R}_{\geq 0} \text{ s.t. } \nu \xrightarrow{t} \nu + t \wedge \rho(\nu) + \epsilon \geq \rho(\nu + t) \wedge UA(\nu + t) \quad (\textit{unsafe alternatives})$$

A safe alternative to a robustness decreasing transition is a transition that does not decrease robustness. The property requires the existence of a controllable safe alternative, or a delay leading to a state from which: all outgoing transitions do not decrease robustness, or there is an outgoing controllable safe alternative. That delay, preceding a controllable safe alternative, does not decrease robustness by more than an arbitrarily small $\epsilon > 0$. With that requirement, a contributory cause is avoidable in the sense that there is an alternative transition from the starting state of the corresponding transition of the cause and this alternative transition is not itself a contributory cause in any other runs. The following Example 10 illustrates why in some cases a ϵ-relaxation $\rho(\nu) - \epsilon \leq \rho(\nu + t)$ of the inequality $\rho(\nu) \leq \rho(\nu + t)$ is needed.

Example 10 (Why is a ϵ-relaxation needed?) *Figure 6.3 presents a situation where for the states $\nu + t$ with $t \in [0, 1]$, we have $\neg(CSA(\nu + t))$ because there is one transition $\nu + t \xrightarrow{2-t} \nu + 2$ that decreases ρ and there are no discrete controllable safe alternatives from $\nu + t$. Moreover, for $t \in]1, 2]$, we have $CSA(\nu + t)$ because there is a controllable safe alternative.*

A ϵ-relaxation is needed, because all delays from $\{\nu + t \mid t \in [0, 1]\}$ to $\{\nu + t \mid t \in]1, 2]\}$

decreases robustness. Furthermore, for any $\epsilon \in \mathbb{R}_{>0}$*, we have:*

$$\forall t \in [0,1]\ \exists \tau \in \mathbb{R}_{>0}\ s.t.\ \rho(\nu) + \epsilon \geq \rho(\nu + 1 + \tau) \wedge CSA(\nu + 1 + \tau) \tag{6.2}$$

because $\lim_{\tau>0,\tau\to 0} \rho(\nu + 1 + \tau) = \rho(\nu + 1)$.

Hence, in the example ρ *satisfies safe alternatives but not the stronger version without relaxation* ($\epsilon = 0$)*:*

$$\forall \nu \in \mathcal{V}\ \exists t \in \mathbb{R}_{\geq 0}\ s.t.\ \nu \xrightarrow{t} \nu + t \wedge \rho(\nu) \leq \rho(\nu + t) \wedge CSA(\nu + t)$$

Notice that, if for $t \geq 1$ *(instead of* $t > 1$*) there is a discrete controllable transition from* $\nu + t$ *that increases* ρ *and for* $t \in [0,1]$*, the stronger property with* $\epsilon = 0$ *is satisfied.*

Later in the chapter, in Example 12, we provide a model and a robustness function for which *safe alternatives* is satisfied but, as for Example 10, not without relaxation.

Unsafe alternatives requires the existence of a transition not increasing robustness for any robustness-increasing transition. A similar ϵ-relaxation as in *safe alternatives* is done here.

Both properties ensure that, on sequential portions of the run, robustness remains constant. Furthermore, all contributory causes can be avoided via an alternative run in which any decrease of robustness is not effective. Reducing robustness is therefore a choice of the system.

The following property *monotonicity* is similar to *unsafe alternatives* but takes account of runs not increasing robustness rather than transitions. For the same reasons that a ϵ-relaxation is needed for *unsafe alternatives*, we also need a ϵ-relaxation for *monotonicity.*

First, we define how much a transition increases robustness.

Definition 37 *We define the function bump as follows:*
For a discrete transition $\nu \xrightarrow{e} \nu'$*, with* $e \in \Sigma$*:*
$bump(\nu \xrightarrow{e} \nu') = max(0, \rho(\nu') - \rho(\nu))$.
For a delay transition $\nu \xrightarrow{t} \nu + t$*, with* $t \in \mathbb{R}_{>0}$*:*
$bump(\nu \xrightarrow{t} \nu + t) = \sup_{\langle t_0,\ldots,t_k\rangle \in samples(t)} \{\sum_{i=1}^{k} max(0, \rho(\nu + t_{i+1}) - \rho(\nu + t_i))\}$
and $samples(t) = \{\langle t_0, \ldots, t_k\rangle \in [0,t]^{k+1} \mid k \in \mathbb{N}_{>0} \wedge t_0 = 0 \wedge t_k = t \wedge t_0 < t_1 < \cdots < t_k\}$

The definition is straight forward for discrete transitions because robustness either increase or decrease. For a delay $\nu \xrightarrow{t} \nu + t$, robustness $\rho(\nu + \tau)$ might not be monotonic or continuous with respect to the time elapsed τ for $\tau \in [0,t]$ and we use $sample(t)$ to approximate robustness $\rho(\nu + \tau)$ by a piecewise linear and continuous function with respect to time elapsed τ.

For a discrete $\nu \xrightarrow{e} \nu'$ transition that decreases ρ, we have $bump(\nu \xrightarrow{e} \nu') = 0$. For a delay $\nu \xrightarrow{t} \nu + t$ such that $\forall a, b \subset [0,t], a \leq b \implies \rho(\nu + a) \geq \rho(\nu + b)$, we have $bump(\nu \xrightarrow{e} \nu') = 0$. In contrast, for a delay in which ρ does not decrease monotonically, *bump* measures the sum of

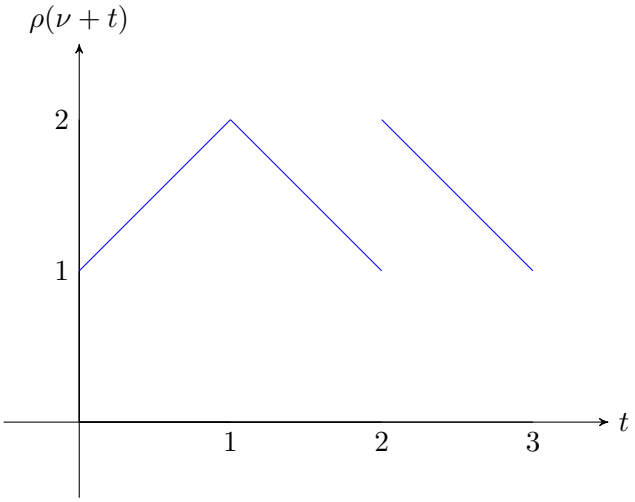


Figure 6.4: For the delay $\nu \xrightarrow{3} \nu{+}3$, we have $bump(\nu \xrightarrow{3} \nu{+}3) = 2$ because the monotonic increasing delay $\nu \xrightarrow{1} \nu + 1$ and the discontinuity in 2 both increase ρ by 1.

increases of robustness of discontinuities and delays for which ρ is monotonically increasing. An example of such delay is given in Figure 6.4.

Definition 38 *Given an approximation $\epsilon \in \mathbb{R}_{\geq 0}$, and a run $r = \nu_0 \xrightarrow{e_1} \nu_1 \ldots \xrightarrow{e_n} \nu_n \in run(\mathcal{A})$, we say that ρ is ϵ-monotonic decreasing in r if $\sum_{i=1}^{n} bump(\nu_{i-1} \xrightarrow{e_i} \nu_i) \leq \epsilon$.*

From any state with finite robustness, there is a run that reaches B and for which the cumulative increase of robustness is arbitrary small.
Monotonicity.

$$0 < \rho(\nu) < \infty \implies \forall \epsilon > 0, \text{ there exists a run } r \in run(\mathcal{A}) \text{ from } \nu \text{ to } B \times dom(X) \text{ such that } \rho \text{ is } \epsilon\text{-monotonic decreasing in } r \qquad (monotonicity)$$

Monotonicity requires the existence of a run in which robustness is monotonically decreasing, which can be thought of as a witness that justifies the robustness value. With that property, robustness is not too pessimistic because in the worst-case scenario robustness will decrease and B will be reached. An approximation that can be arbitrary small is needed for the same reasons that it is needed for *unsafe alternatives*.
No bad surprise.

$$\forall \nu \in \mathcal{V}\ \forall a \in \Sigma, \nu' \in \mathcal{V} \text{ s.t. } \nu \xrightarrow{a} \nu', \rho(\nu') - \rho(\nu) \geq -1 \wedge \forall t \in \mathbb{R}_{>0}, \text{ s.t. } \nu \xrightarrow{t} (\nu + t) : \rho(\nu + t) - \rho(\nu) \geq -\kappa t \qquad (no\ bad\ surprise)$$

No bad surprise states that a discrete transition decreases robustness by at most 1, and a delay of t by at most κt for a parameter $\kappa > 0$. This requirement entails that, in presence of robustness decreasing transitions, the risk of taking them is already "priced in", unlike discrete robustness in example 6.1 that do not satisfy *no bad surprise*. Choosing the parameter $\kappa > 0$ allows the

user to adjust the relative weight of discrete causes and delay-causes in the explanation. From a state ν, the system is at least k discrete causes and t time units of delay causes away from reaching B, for some k and t such that $\rho(x) = k + \kappa t$.

6.4 An Instantiation: ρ_κ

Hypothesis on the model. We make an hypothesis on the model $\mathcal{A} = \langle \Sigma, L, X, \mathcal{I}, E \rangle$ that from a location the guards of controllable transitions are disjoint from the guards of uncontrollable transitions, that is $\forall \ell \in L \, \forall e_1 \in \Sigma_c \, \forall e_2 \in \Sigma_u$ such that $E(\langle \ell, e_1 \rangle) = \langle \ell_1, g_1, r_2 \rangle \wedge E(\langle \ell, e_2 \rangle) = \langle \ell_2, g_2, r_2 \rangle$, we have $g_1 \cap g_2 = \emptyset$.

This hypothesis implies on the quotient graph $[\mathcal{A}]_\sim = \langle Z, \Sigma \cup \{\delta\}, \rightarrow_\sim \rangle$ the existence of a partition $Z = Z_c \uplus Z_u$ of the zones of $[A]_\sim$ such that Z_c contains all the zones z with at least one transition $z \xrightarrow{e}_\sim z'$ with $e \in \Sigma_c$ and no transition $z \xrightarrow{e}_\sim z'$ with $e \in \Sigma_u$.

6.4.1 The Robustness of Choice Function ρ_κ

We propose a robustness of choice function instance ρ_κ in the class of function $dom(F) := Z \to dom(X) \to \mathbb{R}_{\geq 0} \cup \{\infty\}$. In each zone z, for a state $\langle \ell, x \rangle \in z$, $\rho_\kappa(z)(x)$ is the robustness of $\langle \ell, x \rangle$ that depends on the robustness of both delay successors and discrete successors of $\langle \ell, x \rangle$. In order to formalize this dependence, we define a number of higher-order functions, namely $\boldsymbol{B}_{\Sigma'}, \boldsymbol{W}_{\Sigma'}, \delta, f_c, f_u : dom(F) \to dom(F)$ such that for all $\Sigma' \in \{\Sigma_c, \Sigma_u\}$ and a function $f \in dom(F)$ that over-approximates robustness ρ_κ, a zone $z \in Z$ and a state $\langle \ell, x \rangle \in z$:

- We define $\boldsymbol{B}_{\Sigma'}$ that returns the robustness of a most robust discrete successor of $\langle \ell, x \rangle$:

$$\boldsymbol{B}_{\Sigma'}(f)(z)(x) = \max\{f(z')(x[reset(z, e) := 0]) \mid e \in \Sigma', z \xrightarrow{e}_\sim z'\} \tag{6.3}$$

- We define $\boldsymbol{W}_{\Sigma'}$ that returns the robustness of a least robust discrete successor of $\langle \ell, x \rangle$: $\boldsymbol{W}_{\Sigma'}(f)(z)(x) = \min\{f(z')(x[reset(z, e) := 0]) \mid e \in \Sigma', z \xrightarrow{e}_\sim z'\}$

$$\boldsymbol{B}_{\Sigma'}(f)(z)(x) = \max\{f(z')(x[reset(z, e) := 0]) \mid e \in \Sigma', z \xrightarrow{e}_\sim z'\} \tag{6.4}$$

- the function δ is defined such that $\forall z \in Z$:

$$(\exists z' \in Z \text{ s.t. } z \xrightarrow{\delta}_\sim z') \implies \forall x \in dom(X), \delta(f)(z)(x) = f(z')(x + d(z)(x)) \tag{6.5}$$

$$(\nexists z' \in Z \text{ s.t. } z \xrightarrow{\delta}_\sim z') \implies \forall x \in dom(X), \delta(f)(z)(x) = \infty \tag{6.6}$$

The valuation $x + d(z)(x)$ is the valuation of a delay-successor of $\langle \ell, x \rangle$ on the falling edge $z^\downarrow$. Intuitively the value $\delta(f)(z)(x)$ is the robustness obtained when no discrete events occur in the zone z and the zone is left with a delay.

- The function min^+ is defined such that:

$$\forall V \subset \mathbb{R}_{\geq 0} \cup \{\infty\}, min^+(V) = \min \begin{cases} \max(V) \\ 1 + \min(V) \end{cases} \tag{6.7}$$

We lift min^+ pointwise to sets of $\mathbb{R}_{\geq 0} \cup \{\infty\}$-valued functions.

As we will see later, for a set of robustness values V that are the robustness of the successors of $\langle \ell, x \rangle$, $min^+(V)$ is an upper bound on the robustness of $\langle \ell, x \rangle$. In the definition of min^+, 1 is the greatest loss of robustness via a discrete transition allowed by *no bad surprise*, and the $min^+(V)$ is upper bounded by $\max(V)$ in order to satisfy *safe alternatives*.

- We define the predicate $P_z : dom(F) \to dom(X) \to dom(B)$ such that $\forall x \in z$

$$P_z(f)(x) \iff \boldsymbol{B}_{\Sigma_c}(f)(z)(x) \geq min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \delta\}(f)(z)(x). \tag{6.8}$$

For a controllable zone $z \in Z_c$, $P_z(f)(x)$ means that the state $\langle \ell, x \rangle \in z$ has a discrete controllable successor more robust than $min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \delta\}(f)(z)(x)$. We also introduce the delay $next_z(f)(x)$, for which $\neg P_z(f)(x+t)$ for $t < next_z(f)(x)$.

$$next_z(f)(x) = \min \begin{cases} d(z)(x) \\ \inf\{t \in \mathbb{R}_{>0} \mid P_z(f)(x+t)\} \end{cases} \tag{6.9}$$

- We define the *controllable future* $f_c : dom(F) \to dom(F)$ such that $\forall z \in Z_c$, s.t. $\exists z' \in Z, z \xrightarrow{\delta}_{\sim} z', \forall x \in dom(X)$

$$f_c(f)(z)(x) = \inf\{(min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \delta\})(f)(z)(x+t) \mid t \in \mathbb{R}_{\geq 0} \wedge t \leq next_z(f)(x)\} \tag{6.10}$$

such that $\forall z \in Z_c$, s.t. $\forall z' \in Z, \neg(z \xrightarrow{\delta}_{\sim} z'), \forall x \in dom(X)$

$$f_c(f)(z)(x) = (min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}\})(f)(z)(x) \tag{6.11}$$

In a controllable zone $z \in Z_c$, $f_c(f)(z)(x)$ is a quantification of the choices from $\langle \ell, x \rangle$ of the system. The choices are the least or most robust controllable successors with robustness $\boldsymbol{W}_{\Sigma_c}(f)(z)(x)$ and $\boldsymbol{B}_{\Sigma_c}(f)(z)(x)$, or the delays successors in the δ-successor of z with robustness $\delta(f)(z)(x)$. The controllable future does not only depend on robustness of discrete successors because the system can choose to wait, and if $\delta(f)(z)(x) > 0$, then $f_c(f)(z)(x) > 0$ in order to satisfy *soundness*. Moreover, we have $f_c(f)(z)(x) \leq min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \delta\}(f)(z)(x) \leq 1 + \min(\boldsymbol{W}_{\Sigma_c}, \delta)(f)(z)(x)$, that inequality is used later to prove that ρ_κ satisfies *no bad surprise* on discrete transitions. When $next_z(f)(x) > 0$, $f_c(f)(z)(x+t)$ is increasing for $t \in [0, next_z(f)(x)]$. $f_c(f)(z)(x)$ anticipates the potential decrease of $\boldsymbol{W}_{\Sigma_c}(f)(z)(x+t)$ while $\neg P_z(x)$ in order to satisfy *safe alternatives*.

- We define the *uncontrollable future* $f_u : dom(F) \to dom(F)$ such that $\forall z \in Z_c, \forall x \in$

$dom(X)$

$$f_u(f)(z)(x) = W_{\Sigma_u}(z)(x) \tag{6.12}$$

In an uncontrollable zone $z \in Z_u$, there are no choices of the system from $\langle \ell, x \rangle \in z$. The uncontrollable future $f_u(f)(z)(x)$ is the robustness of a least robust uncontrollable successor.

We are now ready to define a family of robustness functions called ρ_κ.

Definition 39 (Robustness ρ_κ) *Let $\mathcal{A} = \langle \Sigma, L, X, \mathcal{I}, E \rangle$ be a timed automaton with $\Sigma_c \uplus \Sigma_u = \Sigma$ and $[\mathcal{A}] = \langle \mathcal{V}_\mathcal{A}, \Sigma \cup \mathbb{R}_{>0}, \rightarrow \rangle$, $B \subset L$ be a set of bad locations closed under E, and $\kappa > 0$. Let $[\mathcal{A}]_\sim = \langle Z, \Sigma \cup \{\delta\}, \rightarrow_\sim \rangle$ be the STAB-quotient graph of $\mathcal{A}$ with the partition $Z = Z_c \uplus Z_u$ of controllable and uncontrollable zones.*

We define the function $\rho_\kappa \in dom(F)$ as the greatest fixed-point of the following equation: $\forall z \in Z, \nu = \langle \ell, x \rangle \in z$,

$$\rho_\kappa(z)(x) = 0 \text{ when } \ell \in B \tag{6.13}$$

$$\rho_\kappa(z)(x) = \min \begin{cases} \inf_{x+t \in z} f_u(\rho_\kappa)(z)(x+t) \\ \delta(\rho_\kappa)(z)(x) \end{cases} \quad \text{when } z \in Z_u \wedge \ell \notin B \tag{6.14}$$

$$\rho_\kappa(z)(x) = \min \begin{cases} \inf_{x+t \in z} \kappa t + f_c(\rho_\kappa)(z)(x+t) \\ (\kappa d + \delta)(\rho_\kappa)(z)(x) \end{cases} \quad \text{when } z \in Z_c \wedge \ell \notin B \tag{6.15}$$

Intuitively, robustness of a state $\nu = \langle \ell, x \rangle \in z$ depends on the transitions enabled in z. The terms $\inf(\cdot)$ stand for robustness when z is left by a discrete transition after a delay t, with f_u (resp. f_c) aggregating the robustness values of the successors of an uncontrollable (resp. controllable) zone.

When $\exists z' \in Z$ s.t. $z \xrightarrow{\delta}_\sim z'$, the linear functions $(\kappa d + \delta)(\rho_\kappa)(z)$ and $\delta(\rho_\kappa)(z)$ decrease with a rate of $-\kappa$ and 0 with respect to delays and have the value $\delta(\rho_\kappa)(z)(x) = \rho_\kappa(z')(x + d(z)(x))$ at the falling edge $z^\downarrow$ between z and z'.

The smoothing $\inf_{x+t \in z} \kappa t + f_c(\rho_\kappa)(z)(x+t)$ of the controllable future of delay successor $\langle \ell, x+t \rangle$ of $\langle \ell, x \rangle$ anticipates any decrease of robustness caused by an outgoing discrete transition.

Hence, the derivative of robustness $\rho_\kappa(z)(x+t)$ with respect to t within a controllable (resp. uncontrollable) zone is lower-bounded by $-\kappa$ (resp. 0).

Example 11 (Computation of ρ_κ in a zone) *Figure 6.5 is a fragment of a quotient graph used to illustrate the computation of the robustness function $\rho_\kappa(z_0)$ for which we only need the robustness of the successors in z_1, z_2, and z_3. For the sake of simplicity, let us suppose that $\rho_\kappa(z_1)$, $\rho_\kappa(z_2)$, and $\rho_\kappa(z_3)$ are constant with respective value c_1, c_2, c_3 and the distance $d(z)(x) = c - x$ with $c_1 \geq c_2$.*

For $x \in z_0$, we have

$$\inf_{x+t \in z} \kappa t + f_c(\rho_\kappa)(z_0)(x+t) = f_c(\rho_\kappa)(z_0)(x) = min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \delta\}(\rho_\kappa)(z_0)(x).$$

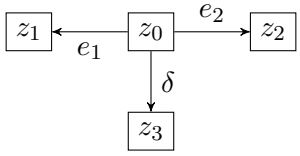


Figure 6.5: Quotient graph.

If the zone $z_0 \in Z_c$ is controllable, $\rho_\kappa(z_0)$ is a piecewise linear function with at most two pieces with the following values for each piece: $\min \begin{cases} \max(c_1, c_3) \\ 1 + \min(c_2, c_3) \end{cases}$ *and $\kappa(c-x)+c_3$. If $c_3 \leq c_2 \leq c_1$, then delays will continuously decrease robustness until z_3 is reached. If $c_1 = c_2 = c_3$, then robustness is constant and equals to c_1, and no transition in z_0 is a contributory cause. If $c_2 \leq c_1 \wedge 1 + c_2 \leq c_3$, robustness is $\rho_\kappa(z_0) = 1 + c_2$, and the transition labeled with the event e_2 decreases robustness by 1 while delays do not decrease robustness.*

When the zone $z_0 \in Z_u$ is uncontrollable, $\rho_\kappa(z_0)$ is constant $min(c_1, c_2, c_3)$.

6.4.2 Computation of ρ_κ

Let $\mathcal{A} = \langle \Sigma, L, X, \mathcal{I}, E \rangle$ be a timed automaton with $\Sigma_c \uplus \Sigma_u = \Sigma$, $[\mathcal{A}] = \langle \mathcal{V}, \Sigma \cup \mathbb{R}_{>0}, \rightarrow \rangle$, and $[\mathcal{A}]_\sim = \langle Z_\mathcal{A}, \Sigma \cup \{\delta\}, \rightarrow_\sim \rangle$ with $Z_\mathcal{A} = Z_c \uplus Z_u$. Let $B \subset L$ be a set of bad locations closed under E, and $\kappa \in \mathbb{N}_{>0}$.

ρ_κ is computed with a monotonically decreasing sequence of the fixed-point iterations, namely $(\rho^i_\kappa)_{i \in \mathbb{N}} \in dom(F)^\mathbb{N}$ from the function ρ^0_κ such that for all valuations $x \in z, \rho^0_\kappa(z)(x) = 0$ when the location ℓ of z is in B, and $\rho^0_\kappa(z)(x) = \infty$ otherwise.

Definition 40 *Linear and piecewise linear functions*

- *Let $X = \{x_1, \ldots, x_n\}$ be a set of clocks, $dom(X)$ the valuations*
- *A linear function over $dom(X)$ is a function l such that $\forall x \in dom(X), l(x) = \sum_{c \in X} \alpha_i x(x_i) + \beta$ for some $\alpha \in \mathbb{R}^n$ and $\beta \in \mathbb{R}_{\geq 0} \cup \{\infty\}$. A linear function $\lambda x. \sum_{c \in X} \alpha_i x(x_i) + \beta$ is denoted by $\langle \alpha, \beta \rangle$. Let $L(X)$ be the set of the linear functions over $dom(X)$.*
- $PRED(X) ::= l < 0 \mid l \leq 0 \mid \phi_1 \wedge \phi_2$ *where* $l \in \mathbb{R}$ *and* $\phi_1, \phi_2 \in PRED(X)$

  $\forall l \in L(X)\ \forall x \in dom(X), x \in (l < 0) \iff l(x) < 0$

  $\forall l \in L(X)\ \forall x \in dom(X), x \in (l \leq 0) \iff l(x) \leq 0$

  $\forall \phi_1 \wedge \phi_2 \in PRED(X)\ \forall x \in dom(X), x \in (\phi_1 \wedge \phi_2) \iff x \in \phi_1 \wedge x \in \phi_2$
- *$PL(X) = \{f \subset L(X) \times PRED(dom(X)) \mid f$ is finite $\}$ is the piecewise linear functions with integer coefficient defined on $\mathbb{X}$*

The next lemma states that each update of the fix-point equation transforms a piecewise linear function into a piecewise linear function.

Lemma 8 *Let $X = \{x_1, \ldots, x_n\}$ be a set of clocks, $\forall f \in Z \to PL(X), \forall z \in Z$, we have:*

$$\lambda x. \min \begin{cases} \inf_{x+t \in z} f_u(f)(z)(x+t) \\ \delta(f)(z)(x) \end{cases} \in PL(X) \wedge$$

$$\lambda x. \min \begin{cases} \inf_{x+t \in z} \kappa t + f_c(f)(z)(x+t) \\ (\kappa d + \delta)(f)(z)(x) \end{cases} \in PL(X)$$

To prove this Lemma 8, we prove that all the operations preserve piecewise linearity.

Proof. Firstly, we prove that:

$$\begin{aligned} \forall f \in Z \to PL(X)\ \forall z' \in Z\ \forall X' \subset X, \\ \lambda x. f(z')(x[X' := 0]) \in PL(X) \end{aligned} \tag{6.16}$$

With $g := \{\langle\langle \alpha[X' := 0], \beta\rangle, \phi[X' := 0]\rangle \mid \langle\langle \alpha, \beta\rangle, \phi\rangle \in f'(z')\}$ such that

- $\alpha[X' := 0] = \lambda c. \begin{cases} \alpha(c) \text{ if } c \notin X' \\ 0 \text{ else} \end{cases}$
- and $\phi[X' := 0]$ is the predicate ϕ in which any occurrence of a clock $c \in X'$ is replaced by 0.

We have $g \in PL(X)$ and $\forall x \in dom(X)$ such that $x[X' := 0] \in z' : g(x) = f(z')(X' := 0])$.
Secondly, for the minimum and maximum of piecewise linear functions we have:

$$\begin{aligned} \forall f_1, f_2 \in PL(X) \\ \lambda x. \min\{f_1(x), f_2(x)\} \in PL(X) \end{aligned} \tag{6.17}$$

Because $\lambda x. \min\{f_1(x), f_2(x)\} = \{\langle l_1, \phi_1 \wedge \phi_2 \wedge l_1 \leq l_2\rangle \mid \langle l_1, \phi_1\rangle \in f_1 \wedge \langle l_2, \phi_2\rangle \in f_2\} \cup \{\langle l_2, \phi_1 \wedge \phi_2 \wedge l_2 \leq l_1\rangle \mid \langle l_1, \phi_1\rangle \in f_1 \wedge \langle l_2, \phi_2\rangle \in f_2\}$
Similarly:

$$\begin{aligned} \forall f_1, f_2 \in PL(X) \\ \lambda x. \max\{f_1(x), f_2(x)\} \in PL(X) \end{aligned} \tag{6.18}$$

Because $\lambda x. \max\{f_1(x), f_2(x)\} = \{\langle l_2, \phi_1 \wedge \phi_2 \wedge l_1 \leq l_2\rangle \mid \langle l_1, \phi_1\rangle \in f_1 \wedge \langle l_2, \phi_2\rangle \in f_2\} \cup \{\langle l_1, \phi_1 \wedge \phi_2 \wedge l_2 \leq l_1\rangle \mid \langle l_1, \phi_1\rangle \in f_1 \wedge \langle l_2, \phi_2\rangle \in f_2\}$
Furthermore,

$$\forall c \in \mathbb{R}\ \forall f \in PL(X), \lambda x. f(x) + c = \{\langle\langle \alpha, \beta + c\rangle, \phi\rangle \mid \langle\langle \alpha, \beta\rangle, \phi\rangle \in f\} \tag{6.19}$$

We can now prove that

$$\forall f_1, \ldots, f_k \in PL(X) \tag{6.20}$$

$$\lambda x. min^+\{f_1(x), \ldots, f_k(x)\} \in PL(X) \tag{6.21}$$

Because by definition of 6.7: $\lambda x.min^{+}\{f_1(x),\ldots,f_k(x)\} = \lambda x.\min\begin{cases}\max\{f_1(x),\ldots,f_k(x)\}\\ 1+\min\{f_1(x),\ldots,f_k(x)\}\end{cases}$ and (6.19),(6.17) and (6.18).

Moreover, we have:

$$\forall l_1, l_2 \in L(X), \lambda x.l_1(x + l_2(x)) \in L(X) \tag{6.22}$$

Indeed, we have $\forall l_1 = \langle \alpha_1, \beta_1\rangle, l_2 = \langle \alpha_2, \beta_2\rangle \in L(X), \lambda x.f(x + l(x)) = \langle \alpha_1 + \hat{\alpha_1}\alpha_2, \beta_1 + \hat{\alpha_1}\beta_2\rangle$

Hence, we can use (6.22) to prove that:

$$\forall f \in PL(X)\ l \in L(X), \lambda x.f(x + l(x)) \in PL(X) \tag{6.23}$$

And more generally,

$$\forall f_1 \in PL(X)\ f_2 \in L(X), \lambda x.f_1(x + f_2(x)) \in PL(X) \tag{6.24}$$

We can now use the proven results to prove that the smoothing operators preserve piecewise linearity:

$$\forall f \in PL(X), \lambda x.inf\{f(x + t) \mid t \in \mathbb{R}_{\geq 0}\} \in PL(X) \tag{6.25}$$

$$\forall f \in PL(X), \lambda x.inf\{\kappa t + \ f(x + t) \mid t \in \mathbb{R}_{\geq 0}\} \in PL(X) \tag{6.26}$$

We can express $inf\{f(x + t) \mid t \in \mathbb{R}_{\geq 0}\}$ as the minimum of a finite number of local minima inf $inf\{l(x + t) \mid \langle l, \phi\rangle \in f, t \in \mathbb{R}_{\geq 0}, x + t \in \phi\}$. The function $\lambda x.inf\{l(x + t) \mid \langle l, \phi\rangle \in f, t \in \mathbb{R}_{\geq 0}, x + t \in \phi\}$ is itself a piecewise linear function. To prove that point, we express $\lambda x.inf\{l(x + t) \mid \langle l, \phi\rangle \in f, t \in \mathbb{R}_{\geq 0}, x + t \in \phi\}$ as the function $\lambda x.l(x + d(x))$ where $d \in PL(X)$ (where $d(x)$ is the delay from x to reach a given local minimum of l).

Such function d is piecewise linear because the function $\lambda t.l(x + t)$ is monotonic and ϕ is convex, therefore the local minimum is reached in t when one of the linear constraints of ϕ is active, i.e. $l' = 0$ such that $\phi = l' < 0 \wedge \phi' \vee \phi = l' \leq 0 \wedge \phi'$ for some $\phi' \in PRED(X)$ and $l'(x+t) = 0$. The value of t with respect to x is the linear function $-(\frac{1}{\hat{l'}})l'(x)$. Notice that $\hat{l'} \neq 0$ otherwise the linear constraint l' cannot be active.

Hence, $\lambda x.inf\{l(x+t) \mid t \in \mathbb{R}_{\geq 0} \wedge x+t \in \phi\}$ is piecewise linear because for $d \in PL(X), \lambda x.l(x+ d(x)) \in PL(X)$. Finally, $\lambda x.inf\{f(z)(x + t) \mid t \in \mathbb{R}_{\geq 0} \wedge x + t \in z\} \in PL(X)$ because it can be written as the minimum of a finite number of piecewise linear functions (6.17).

To prove that $\lambda x.inf\{\kappa t + f(z)(x + t) \mid t \in \mathbb{R}_{\geq 0} \wedge x + t \in z\} \in PL(X)$, we can adapt the proof for the case $\kappa = 0$ because for a linear function $l \in L(X)$ the function $\lambda t.l(x + t) + \kappa t$

is monotonic, hence the local minima are reached on active linear constraints and for $d \in PL(X), \lambda x.l(x + (1 + \kappa)d(x)) \in PL(X)$(6.24).

Hence, we can use the results above to prove that $f_u(f)(z), f_c(f)(z), \delta(f)(z), (\kappa d+\delta)(f)(z) \in PL(X)$, and finally with (6.17) and (6.25), we can conclude. □

The function ρ_κ computed on each of the iterations is piecewise linear since $\rho^0_\kappa(z)$ is constant and the higher-order functions $(\min, min^+, \delta, \boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \dots)$ in $dom(F) \to dom(F)$ used in the fixed-point equation transform, in each zone, a piecewise linear function into a piecewise linear function.

Theorem 9 (ρ_κ is a piecewise linear function) *If there exists $i \in \mathbb{N}$, such that $\rho^i_\kappa = \rho^{i+1}_\kappa$ then $\rho_\kappa \in Z \to PL(X)$.*

In the rest of the chapter we suppose that ρ_κ is computed with a finite number of iterations of the fixed point. We do not have a proof of convergence in a finite number of iterations so far. However, convergence can be proved easily on parts of the graph from which no cycle is reachable. We can exploit this fact to construct an explanation for the suffix of the run once it enters that sub-graph.

6.4.3 ρ_κ is a Robustness of Choice Function

Theorem 10 (*Controllability of causes*) *For all $\langle \ell, x\rangle \xrightarrow{a}_u \langle \ell', x'\rangle$ such that $\langle \ell, x\rangle \in z \in Z_u$ and $\langle \ell', x'\rangle \in z' \in Z$, we have $\rho_\kappa(z)(x) \leq \rho_\kappa(z')(x')$.*

Proof. When $\ell \in B$, by definition of B, we have also $\ell' \in B$, therefore by Definition 39 of ρ_κ, $\rho_\kappa(z)(x) = \rho_\kappa(z')(x') = 0$.

When $\ell \notin B$, by Equation (6.14), $\rho_\kappa(z)(x) \leq \inf_{x+t\in z} f_u(\rho_\kappa)(z)(x + t) \leq f_u(\rho_\kappa)(z)(x) = \boldsymbol{W}_{\Sigma_u}(\rho_\kappa)(z)(x) \leq \rho_\kappa(z')(x')$. □

Lemma 9 is useful to deduce inequalities between robustness of a state and robustness of its delay-successors in the same zone.

Lemma 9 *1. In uncontrollable zones, robustness is non-decreasing:*
$\forall z \in Z_u, \rho_\kappa(z)(x) = \inf_{x+t\in z} \rho_\kappa(z)(x + t)$.

2. In controllable zones $z \in Z_c$, $\rho_\kappa(z)(x) = \inf_{x+t\in z} \kappa t + \rho_\kappa(z)(x + t)$.

Proof. The definition of the distance $d(z)$ to the falling edge $z^\downarrow$ implies that $\forall t \geq 0 : d(z)(x) = t + d(z)(x + t)$. Hence, the equations (6.15) and (6.14) imply the result. □

Theorem 11 (*No bad surprise*, discrete transitions) $\forall z, z' \in Z\ \forall \langle \ell, x\rangle \in z\ \forall e \in \Sigma : \big(\exists \langle \ell', x'\rangle \in z' : \langle \ell, x\rangle \xrightarrow{e} \langle \ell', x'\rangle\big) \implies \rho_\kappa(z)(x) \leq \rho_\kappa(z')(x') + 1$.

Proof. For $z \in Z_u$, theorem 10 implies that uncontrollable discrete transitions increase ρ_κ, hence the inequality is satisfied.

For a controllable zone $z \in Z_c$ such that $\exists z' \in Z$ s.t. $z \xrightarrow{\delta}_{\sim} z'$, by definition of $\boldsymbol{W}_{\Sigma_c}$, $\boldsymbol{W}_{\Sigma_c}(\rho_\kappa)(z)(x) \leq \rho_\kappa(z')(x')$. From the fixed-point equation in definition 39, the definition of the controllable future f_c, and the definition of min^+, we have the following inequality that implies the desired equality:

$$\begin{aligned} \rho_\kappa(z)(x) \leq f_c(\rho_\kappa)(z)(x) \leq min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \delta\}(\rho_\kappa)(z)(x) \\ min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \delta\}(\rho_\kappa)(z)(x) \leq 1 + \boldsymbol{W}_{\Sigma_c}(\rho_\kappa)(z)(x) \end{aligned} \tag{6.27}$$

Additionally, when $\forall z' \in Z$ s.t. $\neg(z \xrightarrow{\delta}_{\sim} z')$ the respective inequality implies the desired equality

$$\begin{aligned} \rho_\kappa(z)(x) \leq f_c(\rho_\kappa)(z)(x) \leq min^+(\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c})(z)(x) \\ min^+(\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c})(z)(x) \leq 1 + \boldsymbol{W}_{\Sigma_c}(\rho_\kappa)(z)(x) \end{aligned} \tag{6.28}$$

□

Theorem 12 (*No bad surprise*, delays) $\forall z, z' \in Z\ \forall \langle \ell, x\rangle \in z\ \forall t > 0:$
$\big(\exists \langle \ell, x+t\rangle \in z' : \langle \ell, x\rangle \xrightarrow{t} \langle \ell, x+t\rangle\big) \implies \rho_\kappa(z)(x) \leq \kappa t + \rho_\kappa(z')(x+t).$

Proof. When $z = z' \in Z_c$ or $z = z' \in Z_u$, Lemma 9 directly implies the desired inequality because $\rho_\kappa(z)(x) \leq \inf_{x+t\in z} \rho_\kappa(z)(x+t) \leq \inf_{x+t\in z} \kappa t + \rho_\kappa(z)(x+t)$. When $\ell \in B$, $\rho_\kappa(z)(x) = 0$ implies the inequality.

For a delay from a zone z to another z', such a zone switching does not decrease ρ_κ because from the fixed-point equation $\rho_\kappa(z)(x + d(z)(x)) \leq \delta(\rho_\kappa)(z)(x)$ and $\delta(\rho_\kappa)(z)(x) = \rho_\kappa(z')(x + d(z)(x))$ by definition of δ. □

Theorem 13 *ρ_κ satisfies safe alternatives.*

Proof. We define $\forall z \in Z, depth(z)$ is the greatest integer $k \in \mathbb{N}$ such that $\exists z_1, \ldots, z_k \in Z$ such that $z \xrightarrow{\delta}_{\sim} z_1 \ldots \xrightarrow{\delta}_{\sim} z_k$.

We prove $\forall z \in Z, x \in z, \epsilon \in \mathbb{R}_{>0}, IH(z, x, \epsilon)$ by induction on $depth(z)$:

$$\begin{aligned} \exists t \in \mathbb{R}_{\geq 0} \text{ s.t. } \exists z' \in Z \wedge x + t \in z' \wedge \\ \rho_\kappa(z)(x) - \epsilon \leq \rho_\kappa(z')(x+t) \wedge \\ (\rho_\kappa(z')(x+t) \leq \boldsymbol{B}_{\Sigma_c}(\rho_\kappa)(z')(x+t) \vee depth(z') = 0) \end{aligned} \tag{$IH(z,x,\epsilon)$}$$

For $z \in Z, x \in z, \epsilon \in \mathbb{R}_{>0}$ such that $depth(z) = 0$, we have $IH(z, x, 0)$ by definition of IH with $t = 0$ and $z = z'$.

For $z \in Z$ such that $depth(z) > 0$, let $x \in z$ and $\epsilon > 0$:

$$\exists z_1 \in Z \text{ s.t. } z \xrightarrow{\delta}_{\sim} z_1 \wedge \forall x' \in z_1, \epsilon' > 0, IH(z_1, x', \epsilon') \tag{6.29}$$

For the case where $z \in Z_c \wedge \rho_\kappa(z)(x) \leq \boldsymbol{B}_{\Sigma_c}(\rho_\kappa)(z)(x)$ we have $IH(z, x, \epsilon)$ with $t' = 0$ (and $z' = z$).

For the case where $\rho_\kappa(z)(x) \leq \delta(\rho_\kappa)(z)(x)$, there exists $\tau \in \mathbb{R}_{\geq 0}$ such that for $x_1 = x + d(z) + \tau$:

$$x_1 \in z_1 \wedge \delta(\rho_\kappa)(z)(x) - \epsilon < \rho_\kappa(z_1)(x_1) \tag{6.30}$$

Let $\epsilon' \in \mathbb{R}_{>0}$ such that $\rho_\kappa(z_1)(x_1) - (\delta(\rho_\kappa)(z)(x) - \epsilon) \geq \epsilon'$, ϵ' is well-defined because (6.30).

By induction hypothesis on x_1 in z_1 we have:

$$\begin{aligned} &\exists t' \in \mathbb{R}_{\geq 0} \text{ s.t. } \exists z' \in Z \wedge x_1 + t' \in z' \wedge \\ &\quad \rho_\kappa(z_1)(x_1) - \epsilon' \leq \rho_\kappa(z')(x_1 + t') \wedge \\ &(\rho_\kappa(z')(x_1 + t') \leq \boldsymbol{B}_{\Sigma_c}(\rho_\kappa)(z')(x_1 + t') \vee \mathit{depth}(z') = 0) \qquad (IH(z_1, x_1, \epsilon')) \end{aligned}$$

Therefore

$$\rho_\kappa(z)(x) - \epsilon \leq \delta(\rho_\kappa)(z)(x) - \epsilon \leq \rho_\kappa(z_1)(x_1) - \epsilon' \leq \rho_\kappa(z')(x_1 + t')$$

Hence, we have $IH(z, x, \epsilon)$ with $t = d(z) + \tau + t'$.

$$\begin{aligned} &\rho_\kappa(z)(x) - \epsilon \leq \rho_\kappa(z')(x + d(z) + \tau + t') \wedge \\ &(\rho_\kappa(z')(x + d(z) + \tau + t') \leq \boldsymbol{B}_{\Sigma_c}(\rho_\kappa)(z')(x + d(z) + \tau + t') \vee \mathit{depth}(z') = 0) \end{aligned}$$

We have proved that We prove $\forall z \in Z, x \in z, \epsilon \in \mathbb{R}_{>0}, IH(z, x, \epsilon)$.

The last item to prove is: $\forall \langle \ell, x \rangle \in \mathcal{V}$ such that $\exists z \in Z \wedge x \in z$

$(\rho_\kappa(z)(x) \leq \boldsymbol{B}_{\Sigma_c}(\rho_\kappa)(z)(x) \vee \mathit{depth}(z) = 0) \implies CSA(\nu)$

$\rho_\kappa(z)(x) \leq \boldsymbol{B}_{\Sigma_c}(\rho_\kappa)(z)(x)$ implies that there is a controllable transition from $\langle \ell, x \rangle$ that increases robustness ρ_κ, by definition of $\boldsymbol{B}_{\Sigma_c}(\rho_\kappa)$, and therefore $CSA(\langle \ell, x \rangle)$.

Whenever $\mathit{depth}(z) = 0 \wedge z \in Z_c$, we have either $z \in Z_c$ and therefore such that $\rho_\kappa(z)(x) \leq \boldsymbol{B}_{\Sigma_c}(\rho_\kappa)(z)(x)$ ((6.11))

Whenever $\mathit{depth}(z) = 0 \wedge z \in Z_u$, there are no transition that decreases robustness ρ_κ because *controllability of causes* and Lemma 9 (All the delays remain in z because $\mathit{depth}(z) = 0$). □

Theorem 14 *ρ_κ satisfies unsafe alternatives.*

Proof. We prove $\forall z \in Z\ \forall \langle \ell, x \rangle \in z\ \forall \epsilon \in \mathbb{R}_{>0}, IH(z, x, \epsilon)$ by induction on $\langle Z, \xrightarrow{\delta}_\sim \rangle$:

$$\begin{aligned} &\exists t \in \mathbb{R}_{\geq 0} \text{ s.t. } \exists z' \in Z \wedge \langle \ell, x + t \rangle \in z' \wedge \\ &\quad \rho_\kappa(z)(x) + \epsilon \geq \rho_\kappa(z')(x + t) \wedge \\ &\quad (UA(\langle \ell, x + t \rangle)) \qquad (IH(z, x, \epsilon)) \end{aligned}$$

$$\exists \nu' \in \mathcal{V}\ \exists e \in \Sigma \cup \mathbb{R}_{>0} \text{ s.t. } \nu \xrightarrow{e} \nu' \wedge \rho(\nu) < \rho(\nu')$$
$$\implies \exists a \in \Sigma \cup \mathbb{R}_{>0}\ \exists \nu'' \in \mathcal{V} \text{ s.t. } \nu \xrightarrow{a} \nu'' \wedge \rho(\nu) \geq \rho(\nu'') \qquad (UA(\nu))$$

Base case ($\ell \in B \vee \forall z' \in Z, \neg(z \xrightarrow{\delta}_{\sim} z')$): For all the state $\langle \ell, x \rangle$, such that $\ell \in B$, we have $UA(\langle \ell, x \rangle)$ because by definition of B, all outgoing transitions from $\langle \ell, x \rangle$ remain in B and have robustness 0 by definition of ρ_κ (6.13). For the rest of the proof, we suppose that $\ell \notin B$.

Let $z \in Z$ such that $\forall z' \in Z, \neg(z \xrightarrow{\delta}_{\sim} z')$, we prove that $\forall \langle \ell, x \rangle \in z, UA(\langle \ell, x \rangle)$, and it implies $\forall \epsilon \in \mathbb{R}_{>0}, IH(z, x, \epsilon)$.

If $z \in Z_u$, we have $\rho_\kappa(z)(x) = \inf_{x+t \in z} \boldsymbol{W}_{\Sigma_u}(\rho_\kappa)(z)(x+t)$ (6.14) and either $\rho_\kappa(z)(x) = \boldsymbol{W}_{\Sigma_u}(\rho_\kappa)(z)(x)$ and there is a discrete successor with robustness $\boldsymbol{W}_{\Sigma_u}(\rho_\kappa)(z)(x)$, or we have $\rho_\kappa(z)(x) = \rho_\kappa(z)(x+t)$ for some $t \in \mathbb{R}_{>0}$. Therefore $\langle \ell, x \rangle$ has a successor with same robustness and $UA(\langle \ell, x \rangle)$.

If $z \in Z_c$, (6.11) and (6.15) imply:

$$\rho_\kappa(z)(x) = \inf_{x+t \in z} \kappa t + (min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}\})(\rho_\kappa)(z)(x+t)$$

When $\rho_\kappa(z)(x) = (min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}\})(\rho_\kappa)(z)(x)$, we have $UA(\langle \ell, x \rangle)$ because $(min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}\})(\rho_\kappa)(z)(x) \leq \boldsymbol{W}_{\Sigma_c}(\rho_\kappa)(z)(x)$ (some controllable transition decrease robustness from $\langle \ell, x \rangle$).

When $\rho_\kappa(z)(x) = (min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}\})(\rho_\kappa)(z)(x)$, we have or for some $t \in \mathbb{R}_{>0}$: $\rho_\kappa(z)(x) = \kappa t + (min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}\})(\rho_\kappa)(z)(x+t)$ and therefore $UA(\langle \ell, x \rangle)$ because $(min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}\})(\rho_\kappa)(z)(x+t) \geq \rho_\kappa(z)(x+t)$ and therefore $\rho_\kappa(z)(x) \geq \rho_\kappa(z)(x+t)$ (the delay t decreases robustness from $\langle \ell, x \rangle$).

Inductive case: Let $z \in Z$ such that $\exists z_1 \in Z, z \xrightarrow{\delta}_{\sim} z_1$, and by induction hypothesis on z_1 we have $\forall \langle \ell, x \rangle \in z_1\ \forall \epsilon \in \mathbb{R}_{>0}, UA(\langle \ell, x \rangle)$.

Let $\langle \ell, x \rangle \in z$, and we prove $\forall \epsilon \in \mathbb{R}_{>0}, IH(z, x, \epsilon)$.

From the definition of $\rho_\kappa(z)$ (6.14) and (6.15), we have:

$$z \in Z_u \implies \rho_\kappa(z)(x) = \min \begin{cases} \inf_{x+t \in z} f_u(\rho_\kappa)(z)(x+t) \\ \delta(\rho_\kappa)(z)(x) \end{cases}$$

$$z \in Z_c \implies \rho_\kappa(z)(x) = \min \begin{cases} \inf_{x+t \in z} \kappa t + f_c(\rho_\kappa)(z)(x+t) \\ (\kappa d + \delta)(\rho_\kappa)(z)(x) \end{cases}$$

Let us first define the function f such that $f(\tau)$ is the robustness of the delay successor

$\langle \ell, x+\tau \rangle$ of $\langle \ell, x \rangle$.

$$f := \lambda t. \begin{cases} \rho_\kappa(z)(x) & \text{if } x+\tau \in z \\ \rho_\kappa(z_1)(x+\tau) & \text{else if } x+\tau \in z_1 \end{cases} \tag{6.31}$$

Robustness $\rho_\kappa(z_1)$ in the zone z_1 is a piecewise linear function (Theorem 9), hence for some $T \in \mathbb{R}_{>0}$ small enough, f is linear on $[0,T]$ and equals $f(t) = \alpha t + \beta$ for some coefficients $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}_{\geq 0} \cup \{\infty\}$.

Subcase 1: Let us assume that $T > 0 \wedge \alpha \leq 0$, it implies $UA(\langle \ell, x \rangle)$ (and therefore $IH(z, x, \epsilon)$) because delays do not increase robustness.

Subcase 2: Let us assume that $T > 0 \wedge \alpha > 0$, we can simplify the expression of $\rho_\kappa(z)(x)$ knowing that $\alpha > 0$ by removing the terms $(\delta(\rho_\kappa)(z)(x+\tau), (\kappa d + \delta)(\rho_\kappa)(z)(x+\tau))$ that are constant or decreasing with τ:

$$z \in Z_u \implies \forall \tau \in [0,T]\ \rho_\kappa(z)(x+\tau) = \inf_{(x+\tau)+t \in z} f_u(\rho_\kappa)(z)(x+\tau+t)$$
$$z \in Z_c \implies \forall \tau \in [0,T]\ \rho_\kappa(z)(x+\tau) = \inf_{(x+\tau)+t \in z} \kappa t + f_c(\rho_\kappa)(z)(x+\tau+t)$$

We can simplify further because $\alpha > 0 \implies \forall \tau \in [0,T], \rho_\kappa(z)(x+\tau) \neq \kappa t + \rho_\kappa(z)(x+\tau) \wedge \rho_\kappa(z)(x+\tau) \neq \rho_\kappa(z)(x+\tau)$, and we have the formula

$$z \in Z_u \implies \forall \tau \in [0,T]\ \rho_\kappa(z)(x+\tau) = f_u(\rho_\kappa)(z)(x+\tau)$$
$$z \in Z_c \implies \forall \tau \in [0,T]\ \rho_\kappa(z)(x+\tau) = f_c(\rho_\kappa)(z)(x+\tau)$$

When $z \in Z_u$, we can directly conclude because $f_u(\rho_\kappa)(z)(x)$ is the robustness of the least robust discrete successor of $\langle \ell, x \rangle$ (6.12), and therefore $UA(\langle \ell, x \rangle)$ (an outgoing discrete transition does not increase robustness).

When $z \in Z_c$, by definition of f_c (when it is increasing) (6.11), we have:

$$\forall \tau \in [0,T]\ \alpha\tau + \beta = f_c(\rho_\kappa)(z)(x+\tau) = min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \delta\}(\rho_\kappa)(z)(x+\tau)$$

By definition of $min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \delta\}(\rho_\kappa)(z)(x+\tau)$, we have $min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \delta\}(\rho_\kappa)(z)(x+\tau) = 1 + \boldsymbol{W}_{\Sigma_c}(\rho_\kappa)(z)(x+\tau)$ or $min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \delta\}(\rho_\kappa)(z)(x+\tau) = \boldsymbol{B}_{\Sigma_c}(\rho_\kappa)(z)(x+\tau)$ because $\alpha > 0$ and $\delta(\rho_\kappa)(z)(x+\tau)$ is constant with τ. In either cases, the inequality $\rho_\kappa(z)(x) \geq \boldsymbol{W}_{\Sigma_c}(\rho_\kappa)(z)(x)$ holds, it means that some outgoing controllable transition from $\langle \ell, x \rangle$ decrease robustness (by definition of $\boldsymbol{W}_{\Sigma_c}$ (6.4)).

Hence, we have therefore $UA(\langle \ell, x \rangle)$ and therefore $IH(z, x, \epsilon)$.

Subcase 3: Let us assume that $T = 0 \wedge x \notin z^{\downarrow}$. The state $\langle \ell, x \rangle$ is not on the falling edge $z^{\downarrow}$ (6.1) and there exists $T' \in \mathbb{R}_{>0}$ such that $f(t) = \alpha' t + \beta'$ for $t \in]0, T']$. *No bad surprise* implies that $\beta' \geq \beta$ (discontinuities only increase robustness) and therefore there are two cases on β: When $\beta = \beta'$ we can re-use the proofs for the two previous subcases because f is linear on $[0, T']$. When $\beta < \beta'$, we simplify the equation of $\rho_\kappa(z)(x)$ as in Subcase 2, and we obtain the same equality/inequality, $\rho_\kappa(z)(x) = \boldsymbol{W}_{\Sigma_u}(\rho_\kappa)(z)(x)$ when $z \in Z_u$ or $\rho_\kappa(z)(x) \geq \boldsymbol{W}_{\Sigma_c}(\rho_\kappa)(z)(x)$ when $z \in Z_c$. That equality/inequality implies $UA(\langle \ell, x \rangle)$ and therefore $IH(z, x, \epsilon)$.

Subcase 4: Let us assume that $T = 0 \wedge x \in z^{\downarrow}$. From the definition of $\rho_\kappa(z)$ (6.14) and (6.15), we have:

$$z \in Z_u \implies \rho_\kappa(z)(x) = \min \begin{cases} f_u(\rho_\kappa)(z)(x) \\ \delta(\rho_\kappa)(z)(x) \end{cases}$$

$$z \in Z_c \implies \rho_\kappa(z)(x) = \min \begin{cases} min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \delta\}(\rho_\kappa)(z)(x+t) \\ \delta(\rho_\kappa)(z)(x) \end{cases}$$

When $\rho_\kappa(z)(x) \neq \delta(\rho_\kappa)(z)(x)$, we can deduce the inequalities $\rho_\kappa(z)(x) \geq \boldsymbol{W}_{\Sigma_c}(\rho_\kappa)(z)(x)$ or $\rho_\kappa(z)(x) \geq \boldsymbol{W}_{\Sigma_u}(\rho_\kappa)(z)(x)$ as in Subcase 2 and we can conclude.

The case left missing is when $\rho_\kappa(z)(x) = \delta(\rho_\kappa)(z)(x)$ and we use the induction hypothesis on the zone z_1.

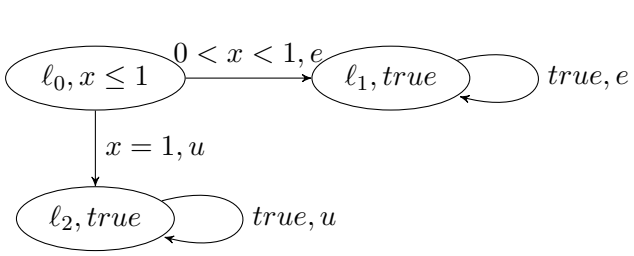
Robustness $\rho_\kappa(z_1)$ in the zone z_1 is a piecewise linear function (Theorem 9), hence for some $T \in \mathbb{R}_{>0}$ small enough, f is linear on $]0, T[$ and equals $f(t) = \alpha t + \beta$ for some coefficients $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}_{\geq 0} \cup \{\infty\}$. We have $\beta = \delta(\rho_\kappa)(z)(x)$ by definition of $\delta(\rho_\kappa)(z)(x)$ (6.5) when $z \xrightarrow{\delta}_{\sim} z_1$ and when the falling edge of z is contained in z (and not z_1). If $\alpha \leq 0$ any delay decreases robustness, and we have $UA(\langle \ell, x \rangle)$. Otherwise, let $\epsilon \in \mathbb{R}_{>0}$ and $t \in]0, \frac{\epsilon}{2\alpha}[$. We have the following inequality:

$$\rho_\kappa(z)(x) + \frac{\epsilon}{2} \geq \rho_\kappa(z_1)(x+t) \tag{6.32}$$

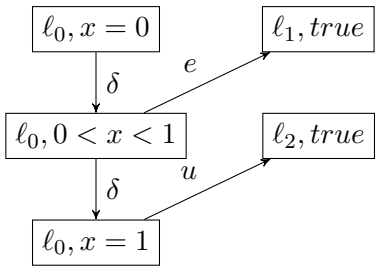
We use the induction hypothesis $IH(z_1, x+t, \frac{\epsilon}{2}$ on $\langle \ell, x+t \rangle$ in z_1 with approximation $\frac{\epsilon}{2}$:

There exists $\tau \in \mathbb{R}_{>0}$ such that $\exists z_2 \in Z$ such that $x+t+\tau \in z_2 \wedge \rho_\kappa(z_1)(x) + \epsilon > \rho_\kappa(z_1)(x+\tau)$. Let $\epsilon' \in \mathbb{R}_{>0}$ such that $\epsilon' \leq \rho_\kappa(z_1)(x) + \epsilon - \rho_\kappa(z_1)(x+\tau)$.

$$\begin{aligned} \exists \tau \in \mathbb{R}_{\geq 0} \text{ s.t. } \exists z' \in Z \wedge \langle \ell, x+t+\tau \rangle \in z' \wedge \\ \rho_\kappa(z_1)(x+t) + \frac{\epsilon}{2} \geq \rho_\kappa(z')(x+t+\tau) \wedge \\ (UA(\langle \ell, x+t+\tau \rangle)) \end{aligned} \qquad (IH(z_1, x+t, \tfrac{\epsilon}{2})$$



(a) TA



(b) Quotient graph

zone z	$\langle \ell_0, x = 0 \rangle$	$\langle \ell_0, 0 < x < 1 \rangle$	$\langle \ell_0, x = 1 \rangle$	$\langle \ell_1, true \rangle$	$\langle \ell_2, true \rangle$
$\rho_\kappa(z)(x)$	1	$1 - x$	0	∞	0

(c) Robustness ρ_κ with $B = \{l_2\}$, $\Sigma_c = \{e\}$, $\Sigma_u = \{u\}$ and $\kappa = 1$

Figure 6.6: Safe alternatives in falling edge.

By combining the inequalities, we have:

$$\rho_\kappa(z)(x) + \epsilon \geq \rho_\kappa(z')(x + t + \tau) \wedge (UA(\langle \ell, x + t + \tau \rangle)) \qquad (IH(z, x, \epsilon)$$

For the controllable case where $z \in Z_c$, by definition of ρ_κ in (6.15), we have:

$$\rho_\kappa(z)(x) = \min \begin{cases} \inf_{x+t \in z} \kappa t + f_c(\rho_\kappa)(z)(x + t) \\ (\kappa d + \delta)(\rho_\kappa)(z)(x) \end{cases}$$

□

Example 12 *Figure 6.6 shows why ρ_κ does not satisfy a stronger version of safe alternatives with $\epsilon = 0$, in general. From state $\langle \ell_0, x = 0 \rangle$, all possible transitions are delays that decrease robustness because ρ_κ satisfies no bad surprise in zones $\langle \ell_0, 0 < x < 1 \rangle$ and $\langle \ell_0, x = 1 \rangle$. However, for any $\epsilon > 0$ and $0 < \tau < \epsilon$, we have $\rho_\kappa(\langle \ell_0, 0 < x < 1 \rangle)(0 + \tau) = 1 - \tau$ and therefore $\rho_\kappa(\langle \ell_0, x = 0 \rangle)(0) = 1 \leq 1 - \tau + \epsilon = \rho_\kappa(\langle \ell_0, 0 < x < 1 \rangle)(0 + \tau) + \epsilon$. From this point, the transition with event e takes the system to the winning state $\langle \ell_1, true \rangle$.*

Theorem 15 (*Soundness* 1) $\forall z \in Z, \forall \langle \ell, x \rangle \in z, \rho_\kappa(z)(x) > 0 \implies \langle \ell, x \rangle$ *is winning.*

Proof. Let $z_0 \in Z$ and a state $\nu_0 = \langle \ell_0, x_0 \rangle \in z$ such that $\rho_\kappa(z_0)(x_0) > 0$, we take $\epsilon > 0$ such $\rho_\kappa(z_0)(x_0) - \epsilon > 0$, we build a strategy $\mathcal{G}_s^\epsilon = \langle \mathcal{V}_s, \Sigma, \rightarrow_s \rangle$ of $\mathcal{G}$ recursively from $\langle \{\nu_0\}, \Sigma, \emptyset \rangle$.

For any state $\nu = \langle \ell, x \rangle \in \mathcal{V}_s$ with no outgoing transitions already in $\rightarrow_s$:

If $\nu \in z$ with $z \in Z_u$, we add into $\rightarrow_s$ all the delays $\nu \xrightarrow{t} \nu + t$ s.t. $\nu + t \in z$, and also all outgoing uncontrollable transitions from the states $\nu + t$. If $\exists z' \in Z$ s.t. $z \xrightarrow{\delta}_\sim z'$, we add a transition $\nu \overset{d(z)+\tau}{\rightarrow} (\nu + d(z) + \tau)$ such that $\tau \geq 0, \nu + d(z) + \tau \in z' \wedge \rho_\kappa(z')(\nu + d(z) + \tau) > (\rho_\kappa(z_0)(x_0) - \epsilon)$.

If ν is in a controllable zone $z \in Z_c$, we add an outgoing controllable transition from ν increasing robustness. When such transition does not exist, we add into $\rightarrow_s$ the delay

$\nu \overset{d(z)(x)+\tau}{\to} (\nu + d(z)(x) + \tau)$ such that $\tau \geq 0, \nu + d(z) + \tau \in z' \wedge \rho_\kappa(z')(\nu + d(z)(x) + \tau) > (\rho_\kappa(z_0)(x_0) - \epsilon)$.

In the construction of $\mathcal{G}_s^\epsilon$, the existence of τ is implied by *safe alternatives*.

$\mathcal{G}_s^\epsilon$ satisfies all the conditions in Definition 35. Indeed, it (a) contains ν, (b) is closed w.r.t. $\to_u$ (by the property of controllability of causes), (c) contains all the outgoing delays in for a state in Z_u. (d) We have by direct induction that each state added with no outgoing transitions has a robustness greater than $(\rho_\kappa(z_0)(x_0) - \epsilon)$ and therefore greater than 0. Furthermore, we prove by induction that all the states added in $\mathcal{V}_s$ have robustness ρ_κ greater than 0. Initially $\rho_\kappa(z_0)(x_0) > 0$. *Controllability of causes* implies that uncontrollable transitions added in the strategy increase ρ_κ. From the fixed-point equation, we have that delays in an uncontrollable zone increase ρ_κ. Each controllable transition increases ρ_κ. Finally, when a delay $\langle \ell, x \rangle \overset{d(z)(x)+\tau}{\to} \langle \ell, x + d(z)(x) + \tau \rangle$ is added to $\to_s$, the intermediate states $\langle \ell, x + t \rangle$ for $t \in [0, d(z)(x) + \tau]$, $\rho_\kappa(z)(x + t) > 0$ because $\delta(z)(x) \geq \rho_\kappa(z)(x) \geq (\rho_\kappa(z_0)(x_0) - \epsilon) > 0$ and therefore the controllable future $f_c(\rho_\kappa)(z)(x + t)$ is lower bounded by a positive constant with respect to t.

Because any state of $B \times dom(X)$ has robustness 0 by definition of ρ_κ, and each state in $\mathcal{V}_s$ has robustness greater than 0, we have that $\mathcal{V}_s \cap B \times dom(X) = \emptyset$. Hence ν_0 is winning.

□

Theorem 16 (*Soundness* 2) *$\forall z \in Z, \langle \ell, x \rangle \in z$ we have*

$$\big(\rho_\kappa(z)(x) = \infty \iff B \text{ is not reachable from } \langle \ell, x \rangle\big).$$

Proof. Let $z \in Z, \langle \ell, x \rangle \in z$, we prove by direct induction on $i \in \mathbb{N}$ that B is not reachable from $\langle \ell, x \rangle \implies \rho_\kappa^i(z)(x) = \infty$. The induction step works because from all successors of $\langle \ell, x \rangle$, B is also not reachable.

We prove the other implication by contrapositive. B is reachable from $\langle \ell, x \rangle \implies \rho_\kappa(z)(x) < \infty$. We prove that implication by induction on the minimum number of transitions of the quotient graph, from z to a zone with a location in B. Since the zone z has a successor z' closer to B than z, we use the induction hypothesis on z' and obtain that $\rho_\kappa(z') < \infty$, if z' is a discrete successor then $f_c(\rho_\kappa)(z)(x') < \infty$ or $f_u(\rho_\kappa)(z)(x') < \infty$. If z' is a delay successor, then $\delta(\rho_\kappa)(x) < \infty$. In all the cases, $\rho_\kappa(z)(x) < \infty$. □

The following lemma is used to prove *monotonicity*. We can use the lemma because ρ_κ is a piecewise linear function, and it has, by definition of $PL(X)$, a finite number of pieces.

Lemma 10 (Continuity and ϵ-monotonicity) *For all piecewise linear functions f, such that $\exists \langle \ell, x \rangle \in \mathcal{V} \wedge T \in \mathbb{R}_{>0}$ s.t. $\langle \ell, x \rangle \overset{T}{\to} \langle \ell, x + T \rangle \wedge \lambda t. f(\langle \ell, x + t \rangle)$ is continuous in $[0, T]$, for all $\epsilon \in \mathbb{R}_{>0}$, there exists $\tau \in]0, T]$ such that f is is ϵ-monotonic decreasing in the run $\langle \ell, x \rangle \overset{\tau}{\to} \langle \ell, x + \tau \rangle$.*

Proof. The function f has a finite number of pieces, hence there exists a linear function $f' : [0, T'] \to \mathbb{R}_{\geq 0}$ such that for $T' \in]0, T]$ small enough, we have: $\forall t \in [0, T'], f(t) = \alpha t + \beta = f'(\langle \ell, x + t \rangle)$. The function f is monotonic. When $\alpha \leq 0$, for all $\tau \in]0, T']$, f is ϵ-monotonic

decreasing in $[\ell, x] \xrightarrow{\tau} \langle \ell, x+\tau \rangle$. When $\alpha > 0$, for all $\tau \leq \frac{\epsilon}{\alpha}$, f is ϵ-monotonic decreasing in $[\ell, x] \xrightarrow{\tau} \langle \ell, x+\tau \rangle$. □

With Lemma 11, we know that for all states with finite robustness ρ_κ, ρ_κ is a ϵ-monotonic decreasing in some run leaving the zone from that state, with ϵ arbitrary small. We can now prove *monotonicity* by constructing an arbitrary long run.

Lemma 11 (Monotonic step from one zone to another) *For a model $\mathcal{A}$ with a quotient graph $[\mathcal{A}]_\sim = \langle Z, \Sigma \cup \{\delta\}, \rightarrow_\sim \rangle$ such that has all cycles in $\rightarrow_\sim$ contain either a zone in B or a zone non co-reachable w.r.t. B, we have for all zone $z \in Z$, for all state $\langle \ell, x \rangle = \nu \in z$ such that $\ell \notin B$, and for all approximation $\epsilon \in \mathbb{R}_{>0}$:*

ρ_κ is ϵ-monotonic decreasing run in some run from ν to a zone $z' \in Z$ such that $z' \neq z$.

Proof. We prove the property by construction:

For a zone $z \in Z_u$ and a state $\langle \ell, x \rangle \in z$ such that $\ell \notin B \wedge \rho_\kappa(z)(x) < \infty$ we have from (6.14): $\forall t' \geq 0$ such that $x + t' \in z$

$$\rho_\kappa(z)(x+t') = \min \begin{cases} \inf_{(x+t')+t \in z} f_u(\rho_\kappa)(z)((x+t')+t) \\ \delta(\rho_\kappa)(z)(x+t') \end{cases} \tag{6.33}$$

Case 1: When $\rho_\kappa(z)(x) = \delta(\rho_\kappa)(z)(x)$, it implies that robustness is constant with delays, i.e. $\rho_\kappa(z)(x+t') = \rho_\kappa(z)(x)$, because $\inf_{(x+t')+t \in z} f_u(\rho_\kappa)(z)((x+t')+t)$ is increasing (not strictly) w.r.t. t' and $\delta(\rho_\kappa)(z)(x+t')$ is constant w.r.t. t'. Furthermore, by definition of δ and because $\rho_\kappa(z)(x) < \infty$, there exists a zone $z' \in Z$ such that $z \xrightarrow{\delta}_\sim z'$. Both functions $\rho_\kappa(z')(x+t)$ and $\rho_\kappa(z)(x+t)$ are continuous w.r.t. t and the limits for $x + t \in z^\downarrow$ is $\delta(\rho_\kappa)(z)(x)$. Therefore, we can conclude with Lemma 10 and $\exists t \in \mathbb{R}_{>0}$ such that $x + t \in z'$ and the run $\langle \ell, x \rangle \xrightarrow{t} \langle \ell, x+\tau \rangle$ is ϵ-monotonic.

Case 2: When $\rho_\kappa(z)(x) < \delta(\rho_\kappa)(z)(x)$ and $\rho_\kappa(z)(x) = \inf_{(x)+t \in z} f_u(\rho_\kappa)(z)((x)+t)$, it implies that robustness $\rho_\kappa(z)(x+t')$ is constant until $\inf_{(x+t')+t \in z} f_u(\rho_\kappa)(z)((x+t')+t) = f_u(\rho_\kappa)(z)((x+t'))$.

There are two cases (Case 1.a) and (Case 1.b):

$$\begin{aligned} &\exists T \in R_{\geq 0} \text{ s.t. } x + T \in z \wedge \\ &\forall t \in [0, T] \ \rho_\kappa(z)(x) = \rho_\kappa(z)(x+t) \wedge \\ &\rho_\kappa(z)(x+T) = f_u(\rho_\kappa)(z)((x+T)) \end{aligned} \qquad \text{(Case 1.a)}$$

For that first case, by definition of f_u in (6.12), the least uncontrollable successor ν' via an event $e \in \Sigma_u$ from $\langle \ell, x+T \rangle$ is less robust than $\langle \ell, x+T \rangle$ and therefore ρ_κ is ϵ-monotonic decreasing in $\langle \ell, x \rangle \xrightarrow{T} \langle \ell, x+T \rangle \xrightarrow{e}_u \nu'$. ν' is not in the same zone as $\langle \ell, x+T \rangle$ because $\langle \ell, x+T \rangle \xrightarrow{e}_u \nu'$ and by hypothesis $\rho_\kappa(z)(x) < \infty \implies \neg(z \xrightarrow{e}_\sim z)$ (Theorem 16). Finally, ν' is not in the same zone as $\langle \ell, x \rangle$ both $\langle \ell, x \rangle$ and $\langle \ell, x+T \rangle$ are in z.

$$\exists T \in R_{\geq 0} \text{ s.t. } x+T \notin z \wedge$$
$$\forall t \in [0,T[\ \rho_\kappa(z)(x) = \rho_\kappa(z)(x+t) \wedge x+t \in z$$
$$\rho_\kappa(z)(x) = lim_{t \to T} f_u(\rho_\kappa)(z)((x+t)) \qquad \text{(Case 1.b)}$$

For that second case, let $\epsilon \in \mathbb{R}_{>0}$, we have $\rho_\kappa(z)(x) < \infty$ and therefore $lim_{t \to T} f_u(\rho_\kappa)(z)((x+t)) < \infty$, by definition of the limit, there is a delay $\tau \in [0,T[$, such that the limit can be approximated $|\rho_\kappa(z)(x) - f_u(\rho_\kappa)(z)((x+\tau)| < \epsilon$. Let $e \in \Sigma_u$, and $\nu' \in \mathcal{V}$ such that ν' is the least robust discrete successor of $\langle \ell, x+\tau \rangle$ (with robustness $f_u(\rho_\kappa)(z)((x+\tau))$, ρ_κ is ϵ-monotonic decreasing in the run $\langle \ell, x \rangle \xrightarrow{\tau} \langle \ell, x+\tau \rangle \xrightarrow{e}_u \nu'$, and $\langle \ell, x \rangle \xrightarrow{\tau} \langle \ell, x+\tau \rangle \xrightarrow{e}_u \nu'$ leaves the zone z.

For a zone $z \in Z_c$ and a state $\langle \ell, x \rangle \in z$ such that $\ell \notin B \wedge \rho_\kappa(z)(x) < \infty$ we have from (6.15): $\forall t' \geq 0$ such that $x+t' \in z$

$$\rho_\kappa(z)(x+t') = \min \begin{cases} \inf_{(x+t')+t \in z} \kappa t + f_c(\rho_\kappa)(z)((x+t')+t) \\ (\kappa d + \delta)(\rho_\kappa)(z)(x+t') \end{cases} \tag{6.34}$$

Similarly, to the uncontrollable case $z \in Z_u$, robustness is decreasing or constant with respect to delay t' until either a state $\langle \ell, x+T \rangle$ from where some discrete transition decrease robustness or $\langle \ell, x+T \rangle \in z^{\downarrow}$.

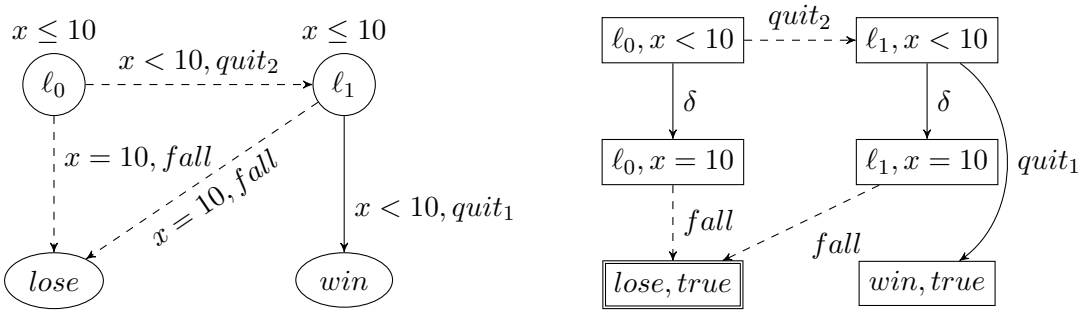
Indeed:

- the function $\lambda t'.(\kappa d + \delta)(\rho_\kappa)(z)(x+t')$ is decreasing because $\lambda t'.(\delta)(\rho_\kappa)(z)(x+t')$ (6.5) is constant w.r.t. t',
- the function $\lambda t'. \inf_{(x+t')+t \in z} \kappa t + f_c(\rho_\kappa)(z)((x+t')+t)$ is either decreasing or equals $\lambda t'.f_c(\rho_\kappa)(z)((x+t'))$,
- the function $\lambda t'.f_c(\rho_\kappa)(z)((x+t')+t)$ (6.10) is either constant (case when $next_z(\rho_\kappa)(x) > 0$ (6.9)) or equals $\lambda t'.min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \delta\}(\rho_\kappa)(z)(x+t')$
- the function $\lambda t'.min^+\{\boldsymbol{B}_{\Sigma_c}, \boldsymbol{W}_{\Sigma_c}, \delta\}(\rho_\kappa)(z)(x+t')$ (6.7) is either constant (when it is equal to $\lambda t'.\delta(\rho_\kappa)(z)(x+t')$ or $\lambda t'.1 + \delta(\rho_\kappa)(z)(x+t')$) or equals $\lambda t'.\boldsymbol{B}_{\Sigma_c}(\rho_\kappa)(z)(x+t')$ or $\lambda t'.1 + \boldsymbol{W}_{\Sigma_c}(\rho_\kappa)(z)(x+t')$.

Moreover, we have by definition of $\boldsymbol{B}_{\Sigma_c}$ (6.3) and $\boldsymbol{W}_{\Sigma_c}$ (6.4)

$$\rho_\kappa(z)(x+t') = \boldsymbol{B}_{\Sigma_c}(\rho_\kappa)(z)(x+t') \vee \rho_\kappa(z)(x+t') = 1 + \boldsymbol{W}_{\Sigma_c}(\rho_\kappa)(z)(x+t') \tag{6.35}$$
$$\implies \rho_\kappa(z)(x+t') \geq \boldsymbol{W}_{\Sigma_c}(\rho_\kappa)(z)(x+t') \tag{6.36}$$
$$\implies \exists \langle \ell', x' \rangle \in \mathcal{V}\ \exists e \in \Sigma_c\ \exists z' \in Z \text{ s.t. } \langle \ell', x' \rangle \in z' \wedge \langle \ell, x+t' \rangle \xrightarrow{e}_c \langle \ell', x' \rangle \tag{6.37}$$
$$\wedge\ \rho_\kappa(z)(x) \geq \rho_\kappa(z')(x') \tag{6.38}$$

Therefore, we can use the same method used for the case $z \in Z_u$ to construct a run r from $\langle \ell, x \rangle$ to another zone such that ρ_κ is ϵ-monotonic decreasing in r.



(a) TA $\mathcal{A}$, all reset are $\emptyset$

(b) Quotient graph $[\mathcal{A}]_\sim$

Figure 6.7: $\mathcal{A}$ and $[\mathcal{A}]_\sim$

□

Theorem 17 (*Monotonicity* of ρ_κ) *For a model $\mathcal{A}$ with a quotient graph $[\mathcal{A}]_\sim = \langle Z, \Sigma \cup \{\delta\}, \rightarrow_\sim \rangle$ such that has all cycles in $\rightarrow_\sim$ contain either a zone in B or a zone non co-reachable w.r.t. B, ρ_κ satisfies monotonicity.*

Proof. Let $z \in Z$, and $\langle \ell, x \rangle \in z$ such that $\rho_\kappa(z)(x) < \infty$, and $\epsilon > 0$. We build a run from $\langle \ell, x \rangle$ recursively using Lemma 11 with the approximation $\epsilon_i = \epsilon(\frac{1}{2})^i$ for $i \in \mathbb{N}_{>0}$. ρ_κ is ϵ-monotonic decreasing in that run because $\epsilon = \sum_{i=1}^{\infty} \epsilon_i = (\frac{1}{1-(1/2)} - 1)\epsilon$. Because there is a finite number of zones and each time we use Lemma 11 a new zone is traversed, a zone z is traversed at least twice by the run. By hypothesis, all cycles in $\rightarrow_\sim$ contain either a zone in B or a zone non co-reachable w.r.t. B. Each states of the run have a finite robustness, and therefore they are co-reachable w.r.t. B (Theorem 16), hence the run reaches B.

□

By Theorems 10, 15, 16, 11, 12, 13, 14, 17 it follows that ρ_κ is indeed a robustness function.

6.5 Example: The Chicken Run

The TA in Figure 6.7a models a scene of the movie *Rebel Without a Cause*. Time is measured by clock x. While $x < 10$, two drivers are racing towards a cliff in their cars. At $x = 10$, both cars fall off the cliff unless the drivers have jumped out before. However, the first driver to do so is the chicken and loses. In our model, Buzz the protagonist waits for Jim, his adversary, to quit first (modeled by the uncontrollable transition $quit_2$). Let $B = \{lose\}$ the set of bad locations where the protagonist falls off the cliff. We first compute the quotient graph in Figure 6.7b and the robustness function ρ_κ. For the initialization ρ_κ^0, robustness is 0 in the zones with a location in B, and robustness is ∞ in all other zones.

At the first iteration, we update robustness in zones $z_0 \in \{\langle \ell_0, x = 10 \rangle, \langle \ell_1, x = 10 \rangle\}$ from which there is only one outgoing transition $\xrightarrow{fall}_u$ that lead to zone $\langle lose, true \rangle$ where robustness is $\boldsymbol{W}_{\Sigma_u}(\rho_\kappa^0)(z_0)(x) = 0$. Hence, $\rho_\kappa^1(z_0)(x) = 0$.

At the second iteration, we update robustness in $z_1 = \langle \ell_1, x < 10 \rangle$. From a state $(x + t) \in z_1$, the protagonist has the choice between taking transition $quit_1$ and reaching robustness $\rho^1_\kappa(z_1)(x+t) = \boldsymbol{B}_{\Sigma_c}(\rho^1_\kappa)(z_1)(x+t) = \rho^1_\kappa(\langle win, true \rangle)(x+t) = \infty$, or waiting and reaching robustness $\delta(\rho^1_\kappa)(z_1)(x+t) = \rho^1_\kappa(\langle lose, true \rangle)(x + t + (10 - x - t)) = 0$. Hence, we have:

$$\rho^2_\kappa(z_1)(x) = \min \begin{cases} \inf_{x+t \in z_1} \kappa t + min^+(\{\infty, 0\}) \\ 0 + \kappa(10 - x) \end{cases} = \min \begin{cases} 1 \\ \kappa(10 - x) \end{cases}$$

Finally, we compute robustness in zone $z_2 = \langle \ell_0, x < 10 \rangle$. We have $\rho^3_\kappa(z_2)(x) = 0$ because the delay $z_2 \overset{10-x}{\to} \langle \ell_0, 10 \rangle$ leads to zone $\langle \ell_0, x = 10 \rangle$ where $\rho_\kappa(\langle \ell_0, x = 10 \rangle) = 0$: intuitively, the antagonist can wait for $10 - x$ seconds and make the protagonist reach B. In zone $\langle \ell_1, x < 10 \rangle$ delays monotonically decrease the robustness $\min\{1, \kappa(10-x)\}$ because a delay $\overset{10-x}{\to}$ reaches the zone $\langle \ell_1, x = 10 \rangle$ that is a losing state. The following table summarizes the robustness of each zone of the quotient graph.

zone z	$\langle \ell_0, x < 10 \rangle$	$\langle \ell_0, x = 10 \rangle$	$\langle \ell_1, x < 10 \rangle$	$\langle \ell_1, x = 10 \rangle$	$\langle lose_1, true \rangle$	$\langle lose_2, true \rangle$	$\langle win, true \rangle$
$\rho_\kappa(z)$	0	0	$\min\{1, \kappa(10-x)\}$	0	0	0	∞

Let us explain the run $r = \langle \ell_0, 0 \rangle \overset{6}{\to}_u \overset{quit_2}{\to}_u \overset{4}{\to}_u \overset{fall}{\to}_u \langle lose, 10 \rangle$ with $\kappa = 1$ where Jim quits and Buzz falls off the cliff. Until Jim quits, robustness is 0. Jim's quitting increases robustness to 1. Then, robustness is constant while Buzz waits for 3 seconds because he has the choice between quitting and waiting, with respective robustness ∞ and 0, hence the controllable future is $f_c(\rho_\kappa)(\langle \ell_{,1}, x < 10 \rangle)(x) = min^+(\boldsymbol{B}_{\Sigma_c}, \delta)(\rho_\kappa)(z)(x) = min^+(\{\infty, 0\} = 1$: a single wrong choice is fatal. Finally, while Buzz waits for another second, robustness decreases from 1 to 0. The following table summarizes how ρ_κ evolves over r.

steps	6	$quit_2$		3	1		$fall$	
zone z	ℓ_0, $x < 10$	ℓ_0, $x < 10$	ℓ_1, $x < 10$	ℓ_1 $x < 10$	ℓ_1, $x < 10$	ℓ_1, $x = 10$	ℓ_1, $x = 10$	$lose$, $true$
$\rho_\kappa(z)$	0	$0 \nearrow 1$		1	$1 \searrow 0$		0	

Hence, the explanation is the single contributory cause $\{\langle \ell_1, 9 \rangle \overset{1}{\to} \langle \ell_1, 10 \rangle\}$.

6.6 Conclusion

We have proposed a principled approach to construct explanations for the violation of an expected safety property by a system modeled by a timed automaton. Explanations are extracted from a new notion of robustness function. We have formalized a set of requirements on robustness functions that ensure the explanations to exhibit the set of contributory causes that together entail the observed violation. These requirements also provide a theoretical underpinning for the explanations based on discrete abstractions of Chapter 5. Each contributory cause is a choice made by the system that brings it closer to the violation, in spite of available safe alternatives. Our explanations are therefore causal, as they rely on counterfactual reasoning over the safe alternatives. We have instantiated our approach in the robustness function ρ_κ, and illustrated the latter and some of our design choices on several simple examples. Our results suggest that

the generated explanations effectively pinpoint the causes that contributed to the violation, thus significantly helping the user in understanding why a run violates the safety property.

Chapter 7

Conclusion and Perspectives

7.1 Summary of the contributions

We have proposed different approaches to explain the failures of reactive real-time systems. Our priority has been to propose explanations that satisfy properties. To this end, we have first identified the expected properties of an explanation and then formalized explanations to satisfy these properties. These properties provide the link between the explanation and the violation of the safety property to be explained. We have formalized several symbolic constructions of causal explanations for the systems modeled with discrete-event systems and timed automata.

In Chapter 4, we have proposed a symbolic approach to effectively construct explanations on discrete event systems, called choice explanations, constructed from a discrete robustness of choice function, called level of choice. Choice explanations can be decoded with a semantic function, into the set of runs that are explained by that explanation. We have identified a set of properties that the semantics of causal explanations should satisfy, and we have proved that the semantics of choice explanations satisfy those properties. A core theorem that we have proved is Theorem 3 (Semantics of choice explanations is causal), which is used to prove those expected properties. It states that choice explanations properly encode the causes, called effective choice transitions, and their order in the observed violation of the safety property. Furthermore, we illustrate choice explanations in a case study of a two-cars system.

We have extended the definition of choice explanations on timed automata. Furthermore, we propose a symbolic approach to effectively construct those explanations, and we illustrate our approach through several examples and a case study of a pacemaker. The concise explanation in the case study allows a user to focus on a small set of causes (7 transitions in the explanation), and reveal an unobservable transition that contributed to the violation. Thanks to that explanation it is possible to repair the systems by preventing the causes from happening in future runs.

We have proposed a novel approach to making causal explanations from what we call a robustness of choice function that, for each state of the system, returns a robustness value. The main motivation is to convey the concept of urgency in the explanation for causes that are delayed. We have defined a contributory cause as a transition that decreases robustness and leads the system closer to the violation of the safety property, and our causal explanations are

constructed from those contributory causes. With our approach, we can build causal explanations without a fault model because the computation of robustness of choice only requires the model of the system and the safety property. With our approach, there is no need to model the failures or faulty events, the semantic model and the safety property are sufficient for explaining the failures. We have formalized a set of requirements for robustness of choice functions to explain the failures of real-time systems. In addition, we have defined an instance of robustness of choice function on real-time systems modeled by timed automata, and we prove it satisfies the requirements on robustness of choice of real-time systems.

7.2 Perspectives

This thesis and the contributions we made enable different future works. A first perspective and a direct extension of my thesis work is the implementation of the symbolic computation of ρ_κ to evaluate the approach on more complex models and traces. It is not a trivial challenge, because we need an efficient implementation of the higher-order operators on piecewise linear functions in order for the computation to scale. Furthermore, we can also study the termination of the computation of ρ_κ in the general case.

We have proposed an encoding of the semantics of choice explanations for discrete event systems. That encoding does not require the safety property to be computed and that allows to convey in the explanation only the relevant part of the property observer for the failure at hand. In this thesis, we have focused on the properties the explanations satisfy. A perspective is to optimize the encoding of choice explanations in terms of size and readability. The goal is to find an efficient encoding of a behavior as an extended automaton such that when composed with the behavioral model, it decodes the encoded behavior. There are different options to tackle this goal because we can constrain both event occurrences and valuations.

A more long-term research direction is blaming quantitatively components in a multi-component system. With our current contributions, it is possible to compute the set of bad decisions of a component if we consider the rest of the components as the environment. However, if we change the point of view and compute a causal explanation for another component, by changing the controllable events, we will obtain a different explanation. A first approach to blaming components could be to study how changing the controllability of the system affects the causes. A second approach could be to adopt a framework similar to fault ascription in concurrent systems [16] where components are paired with a specification they may violate. In this framework, we can define quantitative causes that depend on both the evolution of the robustness of choice of the system and which components violate their specification.

Bibliography

[1] A.V. Aho, R. Sethi, and J.D. Ullman. *Compilers – Principles, Techniques, and Tools*. Addison Wesley, 1986.

[2] Yashwanth Annpureddy, Che Liu, Georgios Fainekos, and Sriram Sankaranarayanan. S-taliro: A tool for temporal logic falsification for hybrid systems. In Parosh Aziz Abdulla and K. Rustan M. Leino, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 254–257, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[3] Jack W. Baker, Matthias Schubert, and Michael H. Faber. On the assessment of robustness. *Structural Safety*, 30(3):253–267, May 2008.

[4] G. Barbon, V. Leroy, and G. Salaün. Debugging of concurrent systems using counterexample analysis. In *Fundamentals of Software Engineering - 7th International Conference, FSEN 2017, Tehran, Iran, April 26-28, 2017, Revised Selected Papers*, pages 20–34, 2017.

[5] Adrian Beer, Stephan Heidinger, Uwe Kühne, Florian Leitner-Fischer, and Stefan Leue. Symbolic causality checking using bounded model checking. In Bernd Fischer and Jaco Geldenhuys, editors, *Model Checking Software*, pages 203–221, Cham, 2015. Springer International Publishing.

[6] I. Beer, S. Ben-David, H. Chockler, A. Orni, and R.J. Trefler. Explaining counterexamples using causality. *Formal Methods in System Design*, 40(1):20–40, 2012.

[7] Jaroslav Bendík, Ahmet Sencan, Ebru Aydin Gol, and Ivana Černá. Timed Automata Robustness Analysis via Model Checking. *Logical Methods in Computer Science*, Volume 18, Issue 3:8375, July 2022.

[8] Richard M. Bergenstal, William V. Tamborlane, Andrew Ahmann, John B. Buse, George Dailey, Stephen N. Davis, Carol Joyce, Tim Peoples, Bruce A. Perkins, John B. Welsh, Steven M. Willi, and Michael A. Wood. Effectiveness of Sensor-Augmented Insulin-Pump Therapy in Type 1 Diabetes. 363(4):311–320.

[9] Patricia Bouyer, Nicolas Markey, and Ocan Sankur. Robustness in Timed Automata. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum,

Parosh Aziz Abdulla, and Igor Potapov, editors, *Reachability Problems*, volume 8169, pages 1–18. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[10] Norine Coenen, Bernd Finkbeiner, Hadar Frenkel, Christopher Hahn, Niklas Metzger, and Julian Siber. Temporal Causality in Reactive Systems. In Ahmed Bouajjani, Lukáš Holík, and Zhilin Wu, editors, *Automated Technology for Verification and Analysis*, volume 13505, pages 208–224. Springer International Publishing, Cham, 2022.

[11] A. Datta, D. Garg, D.K. Kaynar, D. Sharma, and A. Sinha. Program actions as actual causes: A building block for accountability. In C. Fournet, M.W. Hicks, and L. Viganò, editors, *IEEE 28th Computer Security Foundations Symposium, CSF 2015, Verona, Italy, 13-17 July, 2015*, pages 261–275. IEEE Computer Society, 2015.

[12] Alexandre Donzé. Breach, a toolbox for verification and parameter synthesis of hybrid systems. In *International Conference on Computer Aided Verification*, 2010.

[13] Laurent Doyen, Thomas A. Henzinger, Axel Legay, and Dejan Nickovic. Robustness of Sequential Circuits. In *2010 10th International Conference on Application of Concurrency to System Design*, pages 77–84. IEEE, 2010.

[14] H. Garavel, F. Lang, R. Mateescu, and W. Serwe. CADP 2011: a toolbox for the construction and analysis of distributed processes. *Int. J. Softw. Tools Technol. Transf.*, 15(2):89–107, 2013.

[15] G. Gössler, T. Mari, Y. Pencolé, and L. Travé-Massuyès. Towards Causal Explanations of Property Violations in Discrete Event Systems. In *DX'19 - 30th International Workshop on Principles of Diagnosis*, pages 1–8, Klagenfurt, Austria, November 2019.

[16] Gregor Gössler and Jean-Bernard Stefani. Fault ascription in concurrent systems. In Pierre Ganty and Michele Loreti, editors, *Trustworthy Global Computing*, pages 79–94, Cham, 2016. Springer International Publishing.

[17] A. Groce, S. Chaki, D. Kroening, and O. Strichman. Error explanation with distance metrics. *STTT*, 8(3):229–247, 2006.

[18] Gregor Gössler, Thomas Mari, Yannick Pencolé, and Louise Travé-Massuyès. Towards Causal Explanations of Property Violations in Discrete Event Systems.

[19] Joseph Y. Halpern and Judea Pearl. Causes and Explanations: A Structural-Model Approach. Part II: Explanations.

[20] Joseph Y. Halpern and Judea Pearl. Causes and explanations: A structural-model approach. part i: Causes. *British Journal for the Philosophy of Science*, 56(4):843–887, 2005.

[21] D. Harel and A. Pnueli. On the development of reactive systems. In Krzysztof R. Apt, editor, *Logics and Models of Concurrent Systems*, pages 477–498. Springer Berlin Heidelberg.

[22] Z. Jiang, M. Pajic, S. Moarref, R. Alur, and R. Mangharam. Modeling and verification of a dual chamber implantable pacemaker. In C. Flanagan and B. König, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 188–203, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[23] HoonSang Jin, Kavita Ravi, and Fabio Somenzi. Fate and free will in error traces. *STTT*, 6(2):102–116, 2004.

[24] M. Kölbl, S. Leue, and T. Wies. Clock bound repair for timed systems. In I. Dillig and S. Tasiran, editors, *Computer Aided Verification - 31st International Conference, CAV 2019*, volume 11561 of *LNCS*, pages 79–96. Springer, 2019.

[25] M. Kölbl, S. Leue, and T. Wies. Tartar: A timed automata repair tool. In S.K. Lahiri and C. Wang, editors, *Computer Aided Verification - 32nd International Conference, CAV 2020*, volume 12224 of *LNCS*, pages 529–540. Springer, 2020.

[26] K.G. Larsen, P. Pettersson, and W. Yi. UPPAAL in a Nutshell. *Int. Journal on Software Tools for Technology Transfer*, 1(1–2):134–152, October 1997.

[27] Florian Leitner-Fischer and Stefan Leue. Causality checking for complex system models. In Roberto Giacobazzi, Josh Berdine, and Isabella Mastroeni, editors, *VMCAI*, volume 7737 of *LNCS*, pages 248–267. Springer, 2013.

[28] Pengyuan Lu, Ivan Ruchkin, Matthew Cleaveland, Oleg Sokolsky, and Insup Lee. Causal Repair of Learning-enabled Cyber-physical Systems, April 2023.

[29] Thomas Mari, Thao Dang, and Gregor Gössler. Explaining Safety Violations in Real-Time Systems. In Catalin Dima and Mahsa Shirmohammadi, editors, *Formal Modeling and Analysis of Timed Systems*, Lecture Notes in Computer Science, pages 100–116. Springer International Publishing, 2021.

[30] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1 – 38, 2019.

[31] Dejan Ničković and Tomoya Yamaguchi. RTAMT: Online Robustness Monitors from STL. In Dang Van Hung and Oleg Sokolsky, editors, *Automated Technology for Verification and Analysis*, volume 12302, pages 564–571. Springer International Publishing, Cham, 2020.

[32] J. Pearl. Theoretical impediments to machine learning with seven sparks from the causal revolution. In *Proc. Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*, pages 3–3. ACM, 2018.

[33] R. Reiter. A theory of diagnosis from first principles. *Artif. Intell.*, 32(1):57–95, 1987.

[34] Ocan Sankur. Shrinktech: A Tool for the Robustness Analysis of Timed Automata. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C.

Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Natasha Sharygina, and Helmut Veith, editors, *Computer Aided Verification*, volume 8044, pages 1006–1012. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[35] Shlomit Shalitin and Moshe Phillip. Closing the loop: Combining insulin pumps and glucose sensors in children with type 1 diabetes mellitus. 7(s4):45–49.

[36] Eduardo D. Sontag. Input to state stability: basic concepts and results. *Nonlinear and optimal control theory*, Lecture Notes in Math. 126:163–220, 2008.

[37] Hood Thabit, Sara Hartnell, Janet M Allen, Andrea Lake, Malgorzata E Wilinska, Yue Ruan, Mark L Evans, Anthony P Coll, and Roman Hovorka. Closed-loop insulin delivery in inpatients with type 2 diabetes: A randomised, parallel-group trial. 5(2):117–124.

[38] S. Tripakis and S. Yovine. Analysis of timed systems using time-abstracting bisimulations. *Formal Methods Syst. Des.*, 18(1):25–68, 2001.

[39] Stavros Tripakis and Karine Altisen. On-the-Fly Controller Synthesis for Discrete and Dense-Time Systems. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Jeannette M. Wing, Jim Woodcock, and Jim Davies, editors, *FM'99 — Formal Methods*, volume 1708, pages 233–252. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[40] C. von Essen and B. Jobstmann. Program repair without regret. *Formal Methods in System Design*, 47(1):26–50, 2015.

[41] S. Yovine. KRONOS: A verification tool for real-time systems. *Software Tools for Technology Transfer*, 1(1+2):123–133, 1997.

[42] A. Zeller. *Why Programs Fail.* Elsevier, 2009.

[43] K. Zhou and J. Doyle. *Essentials of Robust Control.* Prentice-Hall, 1998.