Finite Degradation Structures

Antoine Rauzy & Liu Yang
Norwegian University of Science and Technology, Trondheim, Norway
{antoine.rauzy, liu.yang}@ntnu.no

Abstract

Probabilistic risk and safety analyses are used in virtually all industries to assess whether the risk of operating complex technical systems is low enough to be socially acceptable. As of today, these analyses rely mainly on stochastic Boolean models such as fault trees or reliability block diagrams. These models are coarse approximations of the behavior of the systems under study.

In this article, we introduce the notion of finite degradation structure. Finite degradation structures encode the degradation order among the states of multistate systems, i.e. models in which variables can take a finite number of values rather than just two. This extension of Boolean formalisms makes it possible to increase significantly the capacity of expression without increasing significantly the complexity of the calculation of risk indicators.

Technically, finite degradation structures are finite semi-lattices associated with a random process. They form a monoidal category and provide a unified algebraic framework for Boolean reliability models and multistate systems. They shed a new light on central notions of system reliability theory such as those of coherent models and minimal cutsets.
Keywords: Multivalued logics, category theory, system reliability theory, combinatorial models, finite degradation structures

Notations and Acronyms

Throughout this article, we use the following notational conventions and acronyms.

$S \times T$**:** Cartesian product of the set S and T.

$X \cong Y$**:** X is isomorphic to Y.

$\mathcal{D} : \langle D, \leq_D, \bot_D \rangle$**:** Finite degradation structure $\mathcal{D}$, i.e. the semi-lattice, built over the finite set of constants D, the partial order $\leq_D$ over D and the least element $\bot_D$ of D for this partial order.

$\mathcal{A} \otimes \mathcal{B}$**:** Monoidal product of the finite degradation structures $\mathcal{A}$ and $\mathcal{B}$.

$\bigotimes_{\mathcal{X} \in \{\mathcal{X}_\infty, \ldots, \mathcal{X}_\setminus\}} \mathcal{X}$**:** $\mathcal{X}_\infty \otimes \cdots \otimes \mathcal{X}_\setminus$

FDS: Category of finite degradation structures.

$dom(V)$**:** Domain of the variable V. $dom(V)$ is a finite degradation structure.

$dom(\mathcal{V})$**:** $\bigotimes_{V \in \mathcal{V}} dom(V)$.

$var(f)$**:** Set of variables occurring in the finite degradation formula f.

$[\![f]\!]$**:** (Canonical) interpretation of the finite degradation formula f.

$[\![\mathcal{M}]\!]$**:** (Canonical) interpretation of the finite degradation model $\mathcal{M}$.

$\overline{\sigma}_{\mathcal{M}}$**:** Unique admissible extension of the assignment of the state variables of the finite degradation model $\mathcal{M}$ into an assignment of variables of $\mathcal{M}$.

PI (O)**:** Set of prime implicants of an observer O.

$\lceil\!\lceil O \rceil\!\rceil$**:** Coherent hull of the observer O.

$\lfloor \pi \rfloor$: Least minterm compatible with the product π.

MCS (O)**:** Set of minimal cutsets of an observer O.

CriticalStates (U)**:** Set of critical states of a subset U of the states of a finite degradation structure $\mathcal{D} : \langle D, <, \bot \rangle$.

1 Introduction

Probabilistic risk and safety analyses are used in virtually all industries to assess whether the risk of operating complex technical systems (aircrafts, nuclear power plants, offshore platforms…) is low enough to be socially acceptable. The WASH1400 report [1], which followed the Three Mile Island nuclear accident, is usually considered as the historical starting point of their worldwide, cross-industry adoption. As of today, these analyses rely mainly on stochastic Boolean models such as fault trees, reliability block diagrams, event trees or a combination of those. These modeling formalisms are well mastered by practitioners. Reference textbooks are available, e.g. [2, 3]. Safety standards such as IEC 61508 [4] (safety systems), ISO 26262 [5] (automotive industry), or ARP4761 [6] (avionic industry) recommend their use.

Models written in these formalisms encode however coarse approximations of the behavior of the systems under study. They do not make it possible to faithfully represent important features such as cold redundancies, resource sharing or reconfigurations. Of course, more powerful formalisms exists, e.g. Markov chains or stochastic Petri nets [7]. But the complexity of the calculation of risk indicators increases dramatically when leaving the realm of combinatorial models. This complexity frames actually the whole domain: a probabilistic risk/safety model always results of a tradeoff between the accuracy of the description and the ability one has to perform calculations on the model, within one's always limited computational resources [8]. The calculation of the main risk indicators is already #P-hard for combinatorial models [9]. For these models, it is however possible to overcome this theoretical intractability because polynomial approximation schemes exist that give very good practical results [10]. Such approximation schemes are much more delicate to design in the case of more powerful formalisms.

A good compromise would be to stay in the realm of combinatorial models, but to allow the representation of components that can be in more than two states (working or failed). In the reliability engineering literature, the term "multistate systems" designates extensions of Boolean models to the case where variables can take a finite (and in general small) number of values. This term is not very appropriate, but we shall use it here as it is widely accepted. Multistate systems have attracted over the years the attention of researchers and practitioners [11–14]. They are however seldom used in practice, probably due to the too small improvement they provide compared to Boolean formalisms. Most, if not all, published works on multistate systems assume actually that the states of a component are totally ordered, from the working state up to the failed state, going possibly through a number of degraded states.

In this article, we introduce the notion of finite degradation structure which releases this total order constraint. It does not release it fully however: the notion of degradation is kept and generalized. Namely, finite degradation structures are finite semi-lattices associated with a random process. The bottom element of the semi-lattice represents the working state. The partial order relation between elements is a degradation order. The random process describes the probability to be in a given state at a given time.

Each finite degradation structures forms a category, see e.g. [15] for a reference book. The category **FDS** of finite degradation structures is thus a category of categories. Furthermore, **FDS** is a monoidal category: it has a product that makes possible to describe systems as hierarchical assemblies of components. Epimorphisms (surjective mappings) of **FDS** encode abstractions and prove to be extremely useful in the context of reliability engineering.

Eventually, finite degradation structures provide a unified algebraic framework encompassing and extending all combinatorial models used in reliability engineering. Combined with the definition of suitable abstraction, it sheds a new light on the fundamental notions of system reliability theory such as those of coherent models, minimal cutsets and top event

probability from which all practical risk indicators are calculated. Finite degradation structures characterize eventually the algebraic properties a multi-valued logic should have to be used in the reliability engineering context. They can thus be seen as a new way of defining multi-valued logics by means of algebraic properties rather than by means of axioms, as it is in usually the case.

Finite degradation structures pave the way to a significant improvement of the process of probabilistic risk and safety analyses. The idea is to proceed in two steps: first, states of components or groups of dependent components are determined and their probabilities assessed by means, for instance, of Markov chains or discrete event simulations; second, the resulting finite degradation model is assessed by means of algorithms derived from those used to assess fault trees, see e.g. [16, 17]. Under the condition that systems under study can be split into small, independent groups of dependent components, which is often verified by industrial systems, it is thus possible to marry the expressive power of discrete event systems with the computational efficiency of combinatorial formalisms. This idea generalizes assessment methods for dynamic fault trees [18] without requiring that one merges dependent components into a macro-components, which is of interest for qualitative analyses. Note that static analysis techniques exist to automatically split discrete event models into independent parts, see e.g. [19].

Starting from a seemingly minor point, the relation order between states of multistate systems, the notion of finite degradation structures led us to revisit a sizable part of system reliability theory. The contribution of this article is to present and to organize this journey through the logical foundations of reliability engineering.

The remainder of this article is organized as follows. Section 2 explains the rational for finite degradation structures by means of an example stemmed from industrial practice. Section 3 introduces them formally both from an abstract and concrete point of views. Section 4 revisits the notion of prime implicant and minimal cutset in this framework. Section 5 presents some experimental results. Finally Section 6 concludes the article.

2 Illustrative Example

But before diving into formal developments of finite degradation structures, we shall provide the reader with some intuitive ideas by means of an example.

Fig. 1 shows a high integrity pressure protection system (HIPPS) as commonly found in oil and gas industry. This HIPPS is called TA4 in the ISO TR/12489 safety standard [20].

This safety instrumented system is in charge of preventing an overpressure in the pipe that could damage equipment, e.g. separators, located downstream. It works on demand, i.e. when an overpressure occurs in the pipe (the flow of oil, gas and water extracted from wells is actually irregular). It is made of three types of elements: sensors $S1-3$ in charge of

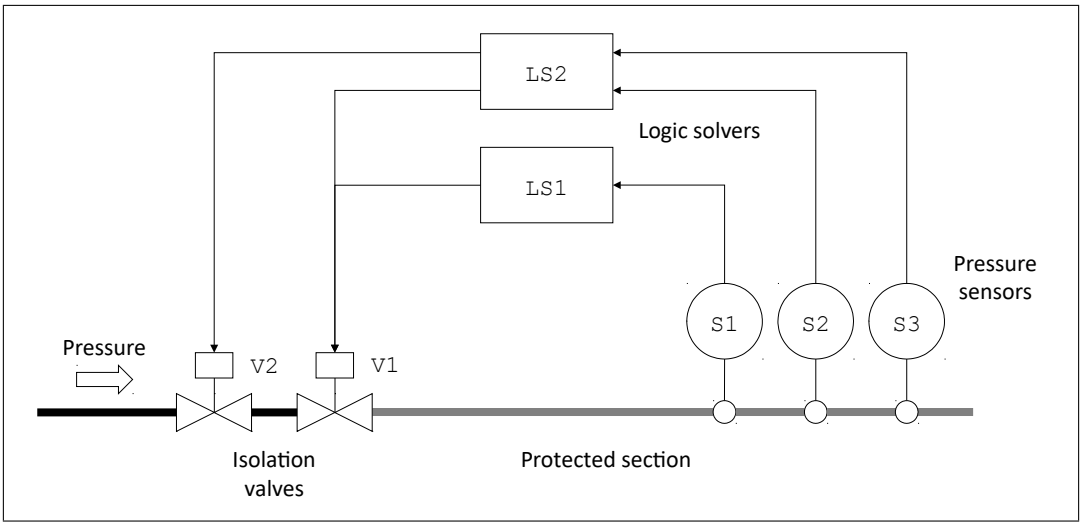

Figure 1: The high integrity pressure protection system TA4

detecting overpressure, logic solvers LS1 − 2 in charge of making the decision and the two isolation valves V1 and V2. The logic solver LS2 works according to a 1-out-of-2 logic, i.e. that it sends the order to close the valves if at least one out of two sensors S2 and S3 detects an overpressure.

According to the standard IEC61508 [4], failure modes of the components of a safety instrumented system can be classified along two directions: safe versus dangerous and detected versus undetected. In our example, safe failure modes are those which contribute to close the isolation valves, even though there is no overpressure (spurious triggers). Dangerous failure modes are those which contribute to keep the isolation valves open, even though there is an overpressure.

Logic solvers embed autotest facilities so that their failures are immediately detected. On the contrary, failure of valves remain undetected between two maintenance interventions. Failures of sensors may be detected or not.

ISO/TR 12489 makes the additional following assumptions.

- All components may fail (independently).
- Safe failures are always detected.
- Probabilities of safe and dangerous failure follow negative exponential distributions. The parameters of these distributions are given Table 1.
- Depending on the type of the component, a given ration of dangerous failures are detected.

Parameter	Sensor	Logic solver	Isolation valve
Safe failure rate	$3.00 \times 10^{-5}\ h^{-1}$	$3.00 \times 10^{-5}\ h^{-1}$	$2.90 \times 10^{-4}\ h^{-1}$
Dangerous failure rate	$5.90 \times 10^{-7}\ h^{-1}$	$5.70 \times 10^{-7}\ h^{-1}$	$2.76 \times 10^{-6}\ h^{-1}$
Detection ratio	0.9	1.0	0.0

Table 1: Reliability parameters for the HIPPS TA4

– The system is maintained once a month (once in 730 hours). The production is stopped during the maintenance. Components are as good as new after the maintenance.

Safe failures and dangerous failures are very different both in terms of frequency of occurrence and severity of consequences. Spurious triggers of the safety instrumented system have a strong economic consequences, but no impact on safety. In contrast, dangerous failures may lead to a catastrophic accident if they remain undetected. Probabilistic risk analyses aim at extracting the most probable scenarios of failure as well as at assessing the probability to be in a safe or dangerous failed state over the mission time of the system.

Our example is small enough (for pedagogical purposes) to make it possible to enumerate by hand all of the possible (global) states of the system and to calculate their probabilities. In real-life applications, such a brute-force approach is basically impossible because of the exponential blow-up of the number of states. Models have to be designed. As of today, Boolean models (fault trees and the like) are by far the most popular. They are however not well suited to represent systems like the above high integrity pressure protection system, because an accurate representation requires to consider more than two states (working or failed) for components and groups of components.

Finite degradation structures, which we shall define formally now, provide a formal algebraic setting to design and to perform risk assessment on such multivalued description.

3 Finite Degradation Stuctures

Finite degradation structures formalize an intuitive idea that is at the core of reliability engineering: components and systems can be in more or less degraded states or, to put it differently, there is a fundamental asymmetry in the possible states of a component or a system: the component or the system is “normally” working, but may degrade and eventually fail. The probability for a component or a system to be in a working state is in general much higher than the probability to be in a degraded or failed state. In other words, states of component or a system are “naturally” ordered with respect to the level of degradation.

This order is in general only a partial order, especially when considering systems made of multiple components.

3.1 Formal Definition

Recall that a *meet-semi-lattice* is a partially ordered set $\langle D, \leq \rangle$ such that any two elements $x, y \in D$ have a greatest lower bound $x \sqcap y$ in D. $x \sqcap y$ is called the meet of x and y. $x \sqcap y = x$ if and only if $x \leq y$.

If D is finite, then it has a unique least element, i.e. an element $\bot$ such that for any other element x, $\bot \leq x$. Assume for a contradiction that D has two such elements $\bot_1$ and $\bot_2$, then we have both $\bot_1 \leq \bot_2$ and $\bot_2 \leq \bot_1$, which by antisymmetry means that $\bot_1 = \bot_2$.

A *finite degradation structure* is thus a meet-semi-lattice $\langle D, \leq, \bot \rangle$ where:

- D is a finite set of constants representing the *states* of a component or a system.
- The partial order relation $\leq$ represents the *degradation order* among states.
- $\bot$, the least element of D, represents the state in which the component is as good as new.

The intuition behind this definition is that the state of a component cannot be less degraded than when it is new. Aside this state, the component may be in more or less degraded states. Some of these states may be comparable in terms of degradation level, i.e. that a state can be more degraded than another, while some other may be incomparable because they correspond to different types of degradation. States are thus organized according to a partial degradation order. As a component may have different failure modes, which are exclusive one another, there may be several distinct most degraded states. Given two states s and t, there is always at least one state that is less degraded than both s and t: the "as-good-as-new" state $\bot$.

Example 1. Fig. 2 shows the Hasse diagrams representing some finite degradation structures that play an important role in system reliability theory, either for their theoretical interest, or to describe the state space of components (they can be seen as on-the-shelf types for these components), or to characterize the state of systems. `w` stands for working, `d` for degraded and `f` for failed. The suffixes `s`, `d` and `u` stand respectively for safe, dangerous detected and dangerous undetected. On the figure, the degradation order is represented bottom-up.

The finite degradation structure **WF** is thus the "classical" Boolean domain working/-failed. In the finite degradation structure **WDF**, an intermediate degraded state is introduced. The finite degradation structure **SWF** is used to represent components in cold redundancy: the component is first in standby mode, then it is working, then it fails. We shall come back on the finite degradation structures **WFsd** and **W3F** represented on the figure.

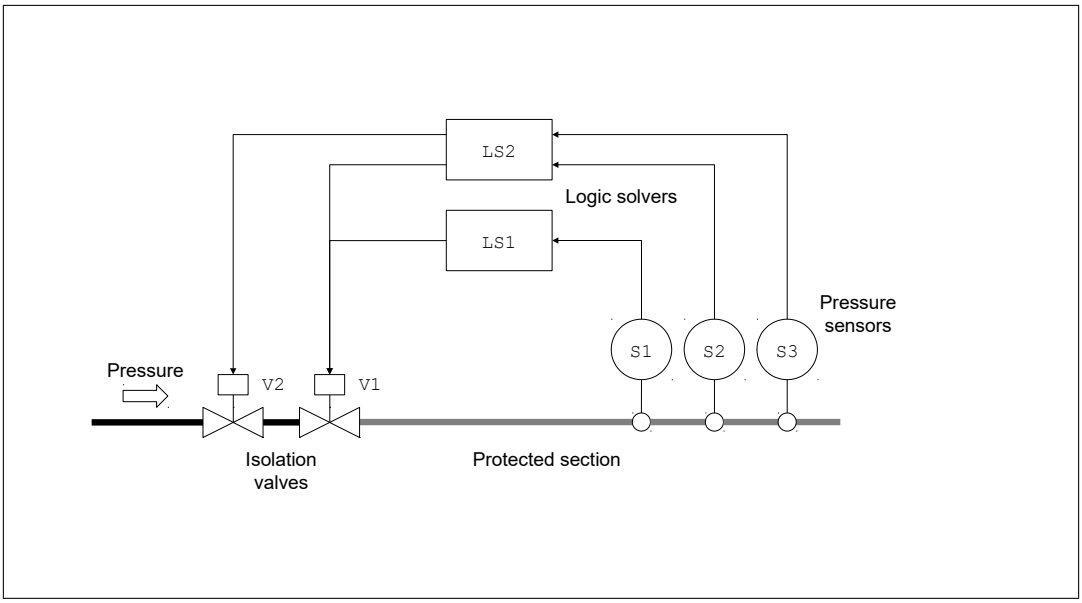

Figure 2: Some useful finite degradation structures

3.2 The Categorical Point of View

A finite degradation structure $\langle D, \leq, \bot \rangle$ is a *category*:

- The objects of this category are the states of D.
- For any two states $s, t \in D$, there is an arrow from s to t if and only if $s \leq t$. If it exists this arrow is unique (and called $\leq$).

Let $\mathcal{A} : \langle A, \leq_A, \bot_A \rangle$ and $\mathcal{B} : \langle B, \leq_B, \bot_B \rangle$ be two finite degradation structures and let ϕ be a mapping from $\mathcal{A}$ to $\mathcal{B}$. Then, we say that ϕ is structure preserving, if:

- For any two states s and t of A, $s \leq_A t$ implies that $\phi(s) \leq_B \phi(t)$.
- $\phi(\bot_A) = \bot_B$.

Structure preserving mappings are monotone functions sending the least element of their domain onto the least element of their codomain. This definition ensures that the image by a structure preserving mapping of a finite degradation structure is a finite degradation structure.

We can define the category **FDS** of finite degradation structures:

- Objects of **FDS** are finite degradation structures.

– Arrows/morphisms of **FDS** are structure preserving mappings between finite degradation structures.

It is easy to verify that **FDS** is actually a category as structure preserving mappings can be composed and it is possible to define an identity (which is indeed a structure preserving mapping) of any finite degradation structure.

Monomorphisms (injective mappings) between finite degradation structures encode extensions, i.e. operations by which states are added to a domain, in view of a finer grain analysis. For instance, we can extend **WF** into **WDF** by mapping `w` and `f` on themselves and adding the intermediate state `d`.

Epimorphisms (surjective mappings) between finite degradation structures encode abstractions: there exists an epimorphism between the finite degradation structure $\mathcal{A}$ and the finite degradation structure $\mathcal{B}$ if $\mathcal{B}$ is an abstraction of $\mathcal{A}$. We shall give in the sequel numerous examples of such abstractions.

Discussion: As we shall see, probabilistic risk assessment models involve not only morphisms between finite degradation structures but also mappings that do not preserve the structure, i.e. that are not monotone functions. Using general mappings to define **FDS** would have make this category very close to the "classical" category **FinSet** whose objects are finite sets and whose arrows are functions between finite sets. The advantage would have been to handle all operations we needed within the category. The drawback would have been to lose the centrality of the notion of degradation order, which is the important one from a reliability engineering point of view.

In any case, the most important constructions we shall use, such as the one of product defined in the next subsection and the notions related to minimal cutsets that we shall develop Section 4 work the same way if we consider structure preserving mappings or general functions.

3.3 Monoidal Product

One of the most interesting properties of **FDS** is that it has a product, i.e. the combination of two (or more) finite degradation structures is also a finite degradation structure. We shall now formalize this idea.

Let $\mathcal{A} : \langle A, \leq_A, \bot_A \rangle$ and $\mathcal{B} : \langle B, \leq_B, \bot_B \rangle$ be two finite degradation structures. We define $\mathcal{A} \otimes \mathcal{B} = \langle A \times B, \leq_{A\otimes B}, \bot_{A\otimes B} \rangle$ as follows.

– $A \times B$ is the Cartesian product of A and B.

– For all $\langle s_A, s_B \rangle, \langle t_A, t_B \rangle \in A \times B$, $\langle s_A, s_B \rangle \leq_{A\otimes B} \langle t_A, t_B \rangle$ if and only if $s_A \leq_A t_A$ and $s_B \leq_B t_B$.

– $\bot_{A\otimes B} = \langle \bot_A, \bot_B \rangle$.

It is easy to verify that $\mathcal{A} \otimes \mathcal{B}$ is a finite degradation structure.

The construction $\mathcal{A} \otimes \mathcal{B}$ comes with the two canonical projections $\pi_1 : \mathcal{A} \otimes \mathcal{B} \to \mathcal{A}$ such that $\pi_1(\langle s, t \rangle) = s$, and $\pi_2 : \mathcal{A} \otimes \mathcal{B} \to \mathcal{A}$ such that $\pi_2(\langle s, t \rangle) = t$.

The following property holds.

Proposition 1 (Product)**.** *$\otimes$ together with the two canonical projections π_1 and π_2 is a product for the category* **FDS***, i.e. that for any three finite degradation structures $\mathcal{A}$, $\mathcal{B}$ and C and pair of morphisms $\varphi_A : C \to \mathcal{A}$ and $\varphi_B : C \to \mathcal{B}$, there exists a unique morphism $\varphi : C \to \mathcal{A} \otimes \mathcal{B}$ such that:*

$$\begin{aligned} \varphi_A &= \pi_1 \circ \varphi \\ \varphi_B &= \pi_2 \circ \varphi \end{aligned}$$

φ is simply defined as $\varphi(s) = \langle \varphi_A(s), \varphi_B(s) \rangle$.

$\mathcal{A} \otimes \mathcal{B}$ is called the *monoidal product* of $\mathcal{A}$ and $\mathcal{B}$.

Note that $\otimes$ is still a product if we consider non-structure preserving mappings as the Cartesian product is a product in **FinSet**.

Recall that two mathematical objects X and Y are *isomorphic* if there is a morphism from X to Y and a morphism from Y to X. In this case, we note $X \cong Y$, the two objects can be considered as identical.

Proposition 2 (Properties of the monoidal product)**.** *Let $\mathcal{A}$, $\mathcal{B}$ and C be three finite degradation structures, then the following equalities hold.*

– $\mathcal{A} \otimes \mathcal{B} \cong \mathcal{B} \otimes \mathcal{A}$ *(Commutativity).*

– $\mathcal{A} \otimes (\mathcal{B} \otimes C) \cong (\mathcal{A} \otimes \mathcal{B}) \otimes C$ *(Associativity).*

– $\mathcal{A} \otimes \mathbf{1} \cong \mathbf{1} \otimes \mathcal{A} \cong \mathcal{A}$ *(Neutral Element).*

where $\mathbf{1} = \langle \{\bot\}, \bot \leq \bot, \bot \rangle$ denotes the finite degradation structure with a unique state.

FDS is thus a symmetric monoidal category. It enjoys other nice algebraic properties, but a full exposition would go beyond the scope of this article. The important point here is that it is possible to build the finite degradation structure of a system by composing the finite degradation structures of its components. It is also possible to group finite degradation structures of a subset of components of a system, so to consider them as a single component. As explained in the introduction, this mechanism is implicitly used to compile discrete event modeling formalisms such as AltaRica into fault trees [19,21]: first, the model is split into independent groups of components by means of static analysis techniques; then, these groups are compiled separately. Finite degradation structures provide a unified algebraic framework to generalize this idea.

Example 2. Consider our illustrative example described Section 2. According to our specifications, this system is made of seven components: the three sensors, the two logic solvers and finally the two valves. We assumed that each of these components can be either working (`w`), failed safe (`fs`), failed detected (`fd`), or failed undetected (`fu`), i.e. can be described with the finite degradation structure **W3F** pictured Fig. 2. The global state of the HIPPS can thus be described by the finite degradation structure $\mathbf{W3F}^7$. Thanks to the product $\otimes$, the partial order over states of individual components is lifted-up into a partial order over the states of the system.

We can now isolate, for instance, the subsystem made of the two sensors `S2` and `S3` and consider it as a macro-component that can be studied separately. In the fault tree framework, such groups of components are called modules [22].

We can now define formulas and models built on top of finite degradation structures, i.e. eventually give a syntax to the finite degradation calculus.

3.4 Formulas

We assume given a finite set $\mathcal{S}$ of finite degradation structures and a finite set $\mathcal{O}$ of symbols called *operators*.

Each operator o of $\mathcal{O}$ is associated with a mapping $[\![o]\!]$ from $\bigotimes_{1\leq i\leq n} s_i$, $n \geq 0$, into s, where both the s_i's and s are finite degradation structures. $\bigotimes_{1\leq i\leq n} s_i$ is called the domain of o and is denoted $dom(o)$. s is called the codomain of o and is denoted $codom(o)$.

Together, $\mathcal{S}$ and $\mathcal{O}$ form what is called an *operad*[1] [23].

Example 3. To deal with the case study presented Section 2, it is useful to introduce parallel || and series $\circledcirc$ compositions. These operators are mappings from $\mathbf{W3F} \otimes \mathbf{W3F}$ into **W3F**. They are defined as shown Table 2.

It is worth noticing that || is both associative and commutative and that it is an epimorphism from $\mathbf{W3F} \otimes \mathbf{W3F}$ to **W3F**. In contrast, $\circledcirc$ is only associative. It is not commutative and does not preserve the partial order. If the first component is failed dangerous undetected and the second one is working then the series of these two components is failed dangerous undetected. Now, if the first component is still failed dangerous undetected, but the second one is failed safe, then the series is failed safe.

We can now define formulas of the finite degradation calculus.

Let $\mathcal{S}$ be a finite set of finite degradation structures and let $\mathcal{O}$ be a finite set of operators on $\mathcal{S}$ defined as above. Let $\mathcal{V}$ be a finite set of symbols called *variables*. Each variable V of $\mathcal{V}$ is assumed to take its value in the support set of one of the finite degradation structures of $\mathcal{S}$. This finite degradation structure is called the domain of V and is denoted $dom(V)$.

[1] We would like to thank here the reviewer who pointed out this notion.

$\|\|$	w	fs	fd	fu
w	w	w	w	w
fs	w	fs	fs	fs
fd	w	fs	fd	fu
fu	w	fs	fu	fu

$\rhd$	w	fs	fd	fu
w	w	fs	fd	fu
fs	fs	fs	fd	fu
fd	fd	fs	fd	fu
fu	fu	fs	fd	fu

Table 2: The operators $\|\|$: $\mathbf{W3F} \otimes \mathbf{W3F} \to \mathbf{W3F}$ and $\rhd$: $\mathbf{W3F} \otimes \mathbf{W3F} \to \mathbf{W3F}$.

Then the set of *well formed (typed) formulas* over $\mathcal{S}$, $\mathcal{V}$ and $\mathcal{O}$ is the smallest set such that:

– Constants, i.e. members of finite degradation structures of $\mathcal{S}$, are well formed formulas. The type of a constant is the finite degradation structure it comes from.

– Variables of $\mathcal{V}$ are well formed formulas. The type of a variable V is simply its domain.

– If o is an operator of $\mathcal{O}$ such that $[\![o]\!] : \bigotimes_{1 \leq i \leq n} s_i \to s$, and $f_1, \ldots f_n$ are well formed formulas of types $s_1, \ldots s_n$, then $o(f_1, \ldots, f_n)$ is a well formed formula of type s.

In the sequel, we shall say simply formula instead of well formed typed formula.
The set of variables occurring in the formula f is denoted $var(f)$.

3.5 Finite Degradation Models

Finite degradation models are obtained by lifting up fault tree constructions to the finite degradation calculus. Namely, a *finite degradation model* $\mathcal{M}$ is a pair $\langle \mathcal{V} = \mathcal{S} \uplus \mathcal{F}, \mathcal{E} \rangle$ where:

– $\mathcal{S} = \{V_1, \ldots, V_m\}$, $m \geq 1$, is a finite set of *state variables*.

– $\mathcal{F} = \{W_1, \ldots, W_n\}$, $n \geq 1$, is a finite set of *flow variables*.

– $\mathcal{E} = \{e_1, \ldots, e_n\}$ is a finite set of *equations*.

Each equation e_j, $1 \leq j \leq n$ is a pair $\langle W_j, f_j \rangle$ where:

– W_j is the jth variable of $\mathcal{F}$.

– f_j is a formula built over the given sets of constants, variables and operators.

For the sake of clarity, the equation $\langle W_j, f_j \rangle$ is simply denoted as $W_j := f_j$. As there is a unique equation $W := f$ for each flow variable W, the formula f can be seen the definition of the variable W.

A finite degradation model $\mathcal{M} : \langle \mathcal{V}, \mathcal{E} \rangle$ is *well typed* if $codom(f_j) = dom(W_j)$ for each equation $W_j := f_j$ of $\mathcal{E}$.

We say that the flow variable W_j depends on the (state or flow) variable U if either $W \in var(f_j)$ or there exists a variable U' of $var(f_j)$ that depends on W.

A finite degradation model is *looped* if one of its flow variable depends on itself. It is *loop-free* or *data-flow* otherwise.

From now, we shall consider only well typed and data-flow models.

A *root variable* is a flow variable that occurs in none of the right members of equations. A finite degradation model is *uniquely rooted* if it has only one root variable. The unique root of such model represents in general the state of the system.

It is easy to see that finite degradation models generalize fault trees: state and flow variables play respectively the roles of basic and internal events, while equations play the role of gates. Moreover, the root variable plays the role of the top event. The terms "state" and "flow" comes from guarded transition systems [24].

Example 4. The high integrity pressure protection system presented Section 2 can be described by the following model.

$$
\begin{array}{rcl@{\qquad}rcl}
\texttt{HIPPS} & := & \texttt{SB1} \parallel \texttt{SB2} & & & \\
\texttt{SB1} & := & \texttt{CL1} \oslash \texttt{V1} & \texttt{SB2} & := & \texttt{CL2} \oslash \texttt{V2} \\
\texttt{CL1} & := & \texttt{LSL1} \parallel \texttt{LSL2} & \texttt{CL2} & := & \texttt{LSL2} \\
\texttt{LSL1} & := & \texttt{SL1} \oslash \texttt{LS1} & \texttt{LSL2} & := & \texttt{SL2} \oslash \texttt{LS2} \\
\texttt{SL1} & := & \texttt{S1} & \texttt{SL2} & := & \texttt{S2} \parallel \texttt{S3}
\end{array}
$$

The state variables of this model are:

- The S*i*'s that represent the states of the sensors.
- The LS*i*'s that represent the states of the logic solvers.
- The V*i*'s that represent the states of the valves.

The flow variables of this model are:

- HIPPS that describes the state of the system.
- SB1 and SB2 that describe respectively the states of the first and second safety barriers.

– CL1 and CL2 that describe respectively the states of the first and second command lines.

– LSL1 and LSL2 that describe respectively the states of the first and second logic solver lines.

– SL1 and SL2 that describe respectively the states of the first and second sensor lines.

It is easy to verify that the above model is data-flow and that HIPPS is its unique root variable.

Formulas and models are syntactic objects. To give them a meaning, we need to define how they are interpreted in terms of mappings from finite degradation structures.

3.6 Interpretation

Let f be a formula of the finite degradation calculus.

A *variable assignment* of f is a mapping from $var(f)$ to $\prod_{V \in var(f)} dom(V)$, i.e. a function that associates with each variable a value of its domain.

f is interpreted as a mapping $[\![f]\!] : \bigotimes_{V \in var(f)} dom(V) \to s$ where s is the codomain of the outmost operator of f, by lifting up as usual variable valuations. Let σ be a variable assignment of $var(f)$, then:

– If f is reduced to a constant c, then $[\![f]\!](\sigma) = c$.

– If f is reduced to a variable V, then $[\![f]\!](\sigma) = \sigma(V)$.

– If f is in the form $o(f_1, \ldots, f_n)$, then $[\![f]\!](\sigma) = [\![o]\!]([\![f_1]\!](\sigma), \ldots, [\![f_n]\!](\sigma))$.

A variable assignment σ is *admissible* in the model $\mathcal{M} : \langle \mathcal{V}, \mathcal{E} \rangle$ if $\sigma(v_j) = \sigma(f_j)$ for each equation $v_j := f_j$ of $\mathcal{E}$.

The following property holds, thanks to the fact that the models we consider are data-flow.

Proposition 3 (Unicity of admissible variable assignments)**.** *Let $\mathcal{M}$ be a finite degradation model and σ be a partial variable assignment that assigns values only to state variables of $\mathcal{M}$. Then there is a unique way to extend σ into an admissible total assignment σ' of variables of M.*

σ' is simply calculated bottom-up by propagating values in the set of equations.

In the sequel we shall denote by $\overline{\sigma}_{\mathcal{M}}$ the unique extension of the assignment the assignment σ of the state variables of the model $\mathcal{M}$ into an admissible assignment of the variables

of $\mathcal{M}$. We shall omit the subscript when the model $\mathcal{M}$ is clear from the context and call $\overline{\sigma}$ the *admissible extension* of σ.

It follows from the above property, that we can interpret a model $\mathcal{M} : \langle \mathcal{V} = \mathcal{S} \uplus \mathcal{F}, \mathcal{E} \rangle$ as a mapping:

$$[\![\mathcal{M}]\!] = \bigotimes_{V \in \mathcal{S}} dom(V) \to \bigotimes_{W \in \mathcal{F}} dom(W) \tag{1}$$

Note that flow variables include in particular the root variables of the model. It is possible, by substituting bottom-up flow variables by their definitions, to transform any finite degradation model into an equivalent formula defining each root variable. This formula may however be exponentially larger than the original model, which the reason why models (in the sense we defined them) are preferred in practice to mere formulas. It remains that, if we are not interested in flow variables but the root variable R, which is often the case, then the model can be interpreted as the mapping:

$$[\![\mathcal{M}]\!] = \bigotimes_{V \in \mathcal{S}} dom(V) \to dom(R) \tag{2}$$

Example 5. The seven state variables of the model described Example 4 take their values in the finite degradation structure `W3F`. The root variable `HIPPS` of the model takes its value in the finite degradation structure `W3F`. The model is thus be interpreted as a mapping from $\mathbf{W3F}^7$ into **W3F**.

In the sequel, we shall denote $\bigotimes_{V \in \mathcal{V}} dom(V)$ simply by $dom(\mathcal{V})$.

3.7 Probabilities

The states of a finite degradation structure $\mathcal{D}$ can be seen as the outcomes of a random experiment. More technically, we can see (the power set of) $\mathcal{D}$ as a probability space and define a *random process*, i.e. a time indexed family X_t, $t \in \mathbb{R}^+$, of random variables over this probability space, see e.g. [25] an introduction to random processes. This random process describes the probability $p_D(s,t) = X_t(s)$ to be in state $s \in D$ at time t.

The above definition makes no assumption about how the probabilities $p_F(s,t)$ are actually obtained in practice. This can be done via analytical formulas, numerical simulations or any other convenient means. Note that random processes can be also used to associate rewards with states. By integrating such a reward over a time period, it is possible, for instance, to assess the expected production of a plant over a time period.

The following properties hold that are at the core of the calculation of probabilistic risk indicators.

Proposition 4 (Composition of probabilities)**.** *Let $\mathcal{A} : \langle A, \leq_A, \bot_A \rangle$ and $\mathcal{B} : \langle B, \leq_B, \bot_B \rangle$ be two finite degradation structures, each associated with a random process. Let p_A and p_B be the probability functions associated respectively with $\mathcal{A}$ and $\mathcal{B}$ by their associated random processes. Then,*

$$p_{\mathcal{A}\otimes\mathcal{B}}(\langle s_A, s_B \rangle, t) \stackrel{def}{=} p_A(s_A, t) \times p_B(s_B, t) \tag{3}$$

defines a probability measure over the monoidal product $\mathcal{A} \otimes \mathcal{B}$. The construction of this probability measure assumes that events represented by states of $\mathcal{A}$ and $\mathcal{B}$ are statistically independent.

Now let $f : \mathcal{A} \to \mathcal{B}$ be a morphism. Then,

$$p_B(s_B, t) \stackrel{def}{=} \sum_{s_A \in f^{-1}(s_B)} p_A(s_A, t) \tag{4}$$

defines a probability measure over $\mathcal{B}$.

In other words, probabilized finite degradation structures compose naturally. The way we associate random processes with finite degradation structures defines actually a lax monoidal functor from **FDS** to the category of random processes, which is also a monoidal category[2].

We could have defined **FDS** by associating systematically a random process with each finite degradation structure and composing them as above. It is however convenient to be able to associate the monoidal product $\mathcal{A} \otimes \mathcal{B}$ with other probability structures than the natural one, so to take into account statistical dependencies. Note also that it is sometimes of interest to consider more complex measurable spaces, e.g. to work in the framework of Dempster-Shafer theory [26].

In the sequel, we shall omit the time when speaking about probability measures as keeping it just complexifies the notations, without bringing much to the point. Nevertheless, the above definition should be constantly borne in mind.

4 Prime Implicants and Minimal Cutsets

The notion of minimal cutset plays a central role in system reliability theory, as well as in practical probabilistic risk analyses. Intuitively, a minimal cutset is a minimal set of component failures that induces a failure of the system as a whole. In other words, minimal cutsets represent the most significant scenarios of failure. As the probability that a component is failed is in general much smaller than the probability that it is working correctly, minimal

[2] Thanks again to the reviewer would pointed out the notion of lax monoidal functor.

cutsets represent also the most probable scenarios of failure. The intuitive definition of minimal cutsets works fine for coherent (monotone) models for which the notion of minimal cutset coincide with the classical notion of prime implicant. However, it needs to be refined to handle non-coherent ones [10].

In this section, we shall generalize the notions of prime implicant and minimal cutset to multistate systems and give the latter a characterization in terms of states of finite degradation structures.

4.1 Observers

The main objective of probabilistic risk and safety analyses is to extract failure scenarios and to assess the cumulated probability of these scenarios. In fault trees, failure scenarios are represented by sets of basic events that induce the top event, i.e. combinations of values of state variables that induce a certain value of the root variable.

We can generalize this idea by considering (Boolean) observers.

Let $\mathcal{M} : \langle \mathcal{V} = \mathcal{S} \uplus \mathcal{F}, \mathcal{E} \rangle$ be a finite degradation model. An *observer* is a Boolean formula O over the values of the variables of $\mathcal{V}$.

Example 6. In the model defined Example 4, a number of observers are of interest, e.g.

- $\texttt{HIPPS} = \texttt{fs}$ that characterizes the states in which the system is in a safe failure mode.
- $\texttt{HIPPS} \in \{\texttt{fd}, \texttt{fu}\}$ that characterizes the states in which the system is in a dangerous failure mode.
- $\texttt{HIPPS} = \texttt{fu}$ that characterizes the states in which the system is in a dangerous undetected failure mode.

We could also consider more complex observers, e.g.

- $\texttt{HIPPS} \in \{\texttt{fd}, \texttt{fu}\} \wedge \texttt{S1} = \texttt{w}$ that characterizes the states in which the system is in a dangerous failure mode and the sensor S1 is working properly.

Note that observers do not need to be structure preserving mappings (assuming $0 < 1$). For instance, the observer above $\texttt{HIPPS} = \texttt{fs}$ is not.

Such an observer O is interpreted as a mapping from $dom(\mathcal{S})$ into $\{0, 1\}$, or equivalently, interpreting O as a characteristic function, as the subset of assignments σ of variables of $\mathcal{S}$ whose extension into admissible assignments of all variables satisfies O:

$$[\![O]\!] \quad = \quad \{\sigma \in dom(\mathcal{S}) : \overline{\sigma}(O) = 1\} \tag{5}$$

Observers are thus predicates in the logical sense.

4.2 Prime Implicants

Let us consider first the extension of the classical notion of prime implicant.

Let $\mathcal{M} : \langle \mathcal{V} = \mathcal{S} \uplus \mathcal{F}, \mathcal{E} \rangle$ be a finite degradation model.

A *product* (over $\mathcal{S}$) is a conjunct of *atoms* of the form $V = s$, where V is a variable of $\mathcal{S}$ and s is a state of $dom(V)$, such that each variable occurs at most once in the product. A *minterm* is a product in which all variables of $\mathcal{S}$ occur. Products and minterms one-to-one correspond respectively with partial and total assignments of variables of $\mathcal{S}$ (and by extension $\mathcal{V}$).

Let O be an observer of $\mathcal{M}$ and π be a product built over $\mathcal{S}$. Then, π entails O, which is denoted as usual by $\pi \models O$, if all minterms $\sigma = \pi \circ \rho$, where ρ is an assignment of the variables of $\mathcal{S} \setminus var(O)$, are such that $\overline{\sigma}(O) = 1$.

In order to lift up the definitions of prime implicant, we need first to generalize the notion of subsumption, i.e. to introduce an order relation $\sqsubseteq$ over products, so to be able to implement the idea of primality and minimality.

Let π and ρ be two products over $\mathcal{S}$. Then, $\pi \sqsubseteq \rho$ if the following conditions hold.

1. $\text{var}(\pi) \subseteq \text{var}(\rho)$.
2. For any atom $V = s$ of π, the atom $V = t$ of ρ verifies $s \leq t$.

Now,

- π is an *implicant* of O, if $\pi \models O$.
- π is a *prime implicant* of O, if it is an implicant of O and no product ρ such that $\rho \sqsubset \pi$ is.

The set of prime implicants of an observer O is denoted by $\text{PI}(O)$. It can be interpreted as the disjunction of its elements.

Example 7. As an illustration, consider again the model of Example 4 and the observer $O_{\texttt{fs}} : \texttt{HIPPS} = \texttt{fs}$.

The product $\pi = \texttt{LS2} = \texttt{fs} \wedge \texttt{V1} = \texttt{fs} \wedge \texttt{V2} = \texttt{w}$ is a prime implicant of $O_{\texttt{fs}}$:

- First, it is easy to verify that, whichever way we complete π into an assignment σ of state variables, we have $\overline{\sigma}(O_{\texttt{fs}}) = 1$.
- Second, if either we change the assignment of `LS2` to `w`, or the assignment of `V1` to `w`, or we remove any of the three atoms of π, then the resulting product is no longer an implicant of $O_{\texttt{fs}}$. For instance, $\texttt{LS2} = \texttt{fs} \wedge \texttt{V1} = \texttt{fs}$ is not an implicant of $O_{\texttt{fs}}$ because if the valve `V2` is failed dangerous undetected, then the whole system is failed dangerous undetected.

Example 7 illustrates the reason why prime implicants are not used in reliability engineering: The prime implicants of observers are "polluted" by information on the states of components that do not participate to the described failure, e.g. the atom $\mathtt{V2} = \mathtt{w}$.

The situation can be even more awkward from a safety analysis point of view. As an illustration, consider the observer $O_{\mathtt{fu}} : \mathtt{HIPPS} = \mathtt{fu}$ and the product $\pi = \mathtt{S1} = \mathtt{fu} \wedge \mathtt{S2} = \mathtt{fu} \wedge \mathtt{S3} = \mathtt{fu}$ that describes the catastrophic situation in which all pressure sensors are lost undetected. Then, if all other components are working properly, the system is failed undetected. But if, by chance, both logic solvers are failed safe, then the system is failed safe. Although correct from a purely logical and probabilistic point of view, it is not very reasonable to count on the safe failures of logic solvers to avoid an accident.

The problem comes from the fact that the series operator does not preserve the partial order, and consequently the definition of HIPPS is non monotone, which is spelled non-coherent in the reliability engineering literature.

To get rid of this problem, we have to lift up to finite degradation calculus the idea introduced in reference [10] to deal with non-coherent Boolean formulas, a notion we shall now formalize.

4.3 Coherence

The notion of coherence plays an important role in system reliability theory. It captures the intuitive idea that the more components of system are degraded, the more likely the system as whole is failed. We generalize it here to the case of finite degradation structures.

Let $\mathcal{M} : \langle \mathcal{V} = \mathcal{S} \uplus \mathcal{F}, \mathcal{E} \rangle$ be a finite degradation model and let O be an observer of $\mathcal{M}$.

O is said *coherent* if, for any two assignments σ and τ of variables of $\mathcal{S}$, $\sigma < \tau$ and $\overline{\sigma}(O) = 1$ implies that $\overline{\tau}(O) = 1$. In other words, $[\![O]\!]$ is a monotone function.

Boolean models of technical systems are in general coherent. However, non-coherent models, or more exactly "almost" coherent models, are sometimes designed.

Non-coherence is used as a modeling trick to make descriptions shorter, but when the model is assessed it is interpreted as a coherent one, via the calculation of minimal cutsets (see next section and reference [27] for an in-depth discussion). With finite degradation models, the situation is slightly different, as exclusive cases can be considered (like failed safe/failed dangerous), as illustrated by Example 7. This makes the (generalization of the) notion of coherent hull, originally introduced for the Boolean case [10], even more interesting.

Let $\mathcal{M} : \langle \mathcal{V} = \mathcal{S} \uplus \mathcal{F}, \mathcal{E} \rangle$ be a finite degradation model and let O be an observer of $\mathcal{M}$. The *coherent hull* of O, denoted by $\lceil\!\lceil O \rceil\!\rceil$, is the smallest coherent set of elements of $dom(\mathcal{S})$ that contains $[\![O]\!]$. Formally,

$$\lceil\!\lceil O \rceil\!\rceil \stackrel{def}{=} \{\tau \in dom(\mathcal{S}); \exists \sigma \in dom(\mathcal{S}); \sigma \leq \tau \wedge \overline{\sigma}(O) = 1\}$$

The following property is a direct consequence of the definition.

Proposition 5 (Coherent hulls)**.** *Let $\mathcal{M} : \langle \mathcal{V} = \mathcal{S} \uplus \mathcal{F}, \mathcal{E} \rangle$ be a finite degradation model and let O be an observer of $\mathcal{M}$. Then, the following inclusion holds.*

$$[\![O]\!] \quad \subseteq \quad \lceil\!\lceil O \rceil\!\rceil$$

Moreover, $[\![O]\!] = \lceil\!\lceil O \rceil\!\rceil$ if and only if O is coherent.

Example 8. Consider again the observer $O_{\mathtt{fu}} : \mathtt{HIPPS} = \mathtt{fu}$ of our example. Consider the minterm π defined as follows.

$$\pi \quad = \quad \mathtt{S1} = \mathtt{fu} \wedge \mathtt{S2} = \mathtt{S3} = \mathtt{fd} \wedge \mathtt{LS1} = \mathtt{LS2} = \mathtt{fs} \wedge \mathtt{V1} = \mathtt{V2} = \mathtt{w}$$

We have $\overline{\pi}(\mathtt{HIPPS}) = \mathtt{fs}$, therefore $\pi \notin [\![O_{\mathtt{fu}}]\!]$.

Now, consider the minterm τ defined as follows.

$$\tau \quad = \quad \mathtt{S1} = \mathtt{fu} \wedge \mathtt{S2} = \mathtt{S3} = \mathtt{fd} \wedge \mathtt{LS1} = \mathtt{LS2} = \mathtt{V1} = \mathtt{V2} = \mathtt{w}$$

We have $\overline{\tau}(\mathtt{HIPPS}) = \mathtt{fu}$, therefore $\tau \in [\![O_{\mathtt{fu}}]\!]$. But as $\tau < \pi$, $\pi \in \lceil\!\lceil O_{\mathtt{fu}} \rceil\!\rceil$.

The following proposition brings us back in the realm of **FDS**.

Proposition 6 (Coherent hulls as epimorphisms)**.** *Let $\mathcal{M} : \langle \mathcal{V} = \mathcal{S} \uplus \mathcal{F}, \mathcal{E} \rangle$ be a finite degradation model and let O be an observer of $\mathcal{M}$. Then, $\lceil\!\lceil O \rceil\!\rceil$ is an epimorphism from $dom(\mathcal{S})$ into $\mathbf{2} = \langle \{0, 1\}, 0 < 1, 0 \rangle$.*

The (probability of the) coherent hull provides a conservative approximation of (the probability of) the observer. In many practical cases, this turns out to be a very good approximation:

$$p(\lceil\!\lceil O \rceil\!\rceil) \quad \approx \quad \sum_{\sigma \in [\![O]\!]} p(\sigma) \tag{6}$$

4.4 Minimal Cutsets

Let π be a product built over a set of variables $\mathcal{V}$, we denote by $\lfloor \pi \rfloor$, the smallest minterm compatible with π. Formally,

$$\lfloor \pi \rfloor \quad \stackrel{def}{=} \quad \sigma \in dom(\mathcal{V});\ \pi(V) = \begin{cases} \pi(V) & \text{if } V \in var(\pi) \\ \bot_{dom(V)} & \text{otherwise} \end{cases}$$

We are now ready to lift up the notion of minimal cutset.

Let $\mathcal{M} : \langle \mathcal{V} = \mathcal{S} \uplus \mathcal{F}, \mathcal{E} \rangle$ be a finite degradation model let O be an observer of $\mathcal{M}$ and finally let π be a product built over $\mathcal{S}$. Then,

– π is a *cutset* of O if $\lfloor \pi \rfloor \in [\![O]\!]$.

– π is a *minimal cutset* of O if it is a cutset of O and no product ρ such that $\rho \sqsubset \pi$ is.

The set of minimal cutsets of an observer O is denoted by MCS (O). It can be interpreted as the disjunction of its elements.

Example 9. Consider again the observer $O_{\mathtt{fu}}$. The product $\pi = \mathtt{S1} = \mathtt{fu} \wedge \mathtt{S2} = \mathtt{fd} \wedge \mathtt{S3} = \mathtt{fd}$ is a minimal cutset of $O_{\mathtt{fu}}$.

– It is a cutset, because the minterm τ defined as in example 8 verifies $\tau \in [\![O]\!]$.

– It is minimal because no product σ smaller than π is such that $\overline{\sigma}(O) = 1$.

The following theorem establishes the relationship between prime implicants and minimal cutsets.

Theorem 7 (Prime Implicants versus Minimal Cutsets)**.** *Let $\mathcal{M} : \langle \mathcal{V} = \mathcal{S} \uplus \mathcal{F}, \mathcal{E} \rangle$ be a finite degradation model and let O be an observer of $\mathcal{M}$. Then, the following equality holds.*

$$\text{MCS}\,(O) \quad = \quad \text{PI}\,(\lceil\!\lceil O \rceil\!\rceil)$$

To prove the above theorem it suffices to remark that a product $\pi \in \lceil\!\lceil O \rceil\!\rceil$ if and only if $\overline{\lfloor \pi \rfloor}(O) = 1$.

We shall now give a characterization in terms of states of the notion of minimal cutsets.

4.5 Critical States

Let $\mathcal{D} : \langle D, <, \bot \rangle$ be a finite degradation structure and let $U \subseteq D$. A state $s \in D$ is *critical* for U if $s \in U$ and there is no state $t \in U$ such that $t < s$. The set of critical states of U is denoted CriticalStates (U).

Example 10. The minterm τ defined as in Example 8 is critical for the subset $[\![O]\!]$ of $dom(\mathtt{S1}) \otimes \cdots \otimes dom(\mathtt{V2})$.

The above example is by no means a coincidence, as stated by the following theorem.

Theorem 8 (Minimal Cutsets versus Critical States)**.** *Let $\mathcal{M} : \langle \mathcal{V} = \mathcal{S} \uplus \mathcal{F}, \mathcal{E} \rangle$ be a finite degradation model and let O be an observer of $\mathcal{M}$. Then, the following equality holds.*

$$\text{MCS}\,(O) \quad \cong \quad \text{CriticalStates}\,([\![O]\!])$$

Any minimal cutset π one-to-one corresponds with $\lfloor \pi \rfloor$. It is easy to verify that $\lfloor \pi \rfloor$ is a critical state for $[\![O]\!]$. Reciprocally, any critical state s one-to-one corresponds to a product π (once removed the variables assigned to the least state of their domain). It is easy to verify that π is a minimal cutset. The minimal cutset π that corresponds with a certain critical state σ is thus obtained by removing from σ the information about components that are working properly.

Extracting minimal cutsets of an observer O, or equivalently critical states for this observer, consists actually in defining an epimorphism $\kappa : dom(\mathcal{S}) \to \mathbf{WCF}$—where **WCF** is the finite degradation structure with three states `w` (working), `c` (failed and critical) and `f` (failed but non critical), such that `w` < `c` < `f`—as stated by the following proposition.

Proposition 9 (Minimal cutsets as epimorphisms)**.** *Let $\mathcal{M} : \langle \mathcal{V} = \mathcal{S} \uplus \mathcal{F}, \mathcal{E} \rangle$ be a finite degradation model and let O be an observer of $\mathcal{M}$. Then,* $\mathrm{MCS}(O)$ *is isomorphic to the epimorphism $\kappa : dom(\mathcal{S}) \to \mathbf{WCF}$ defined as follows.*

$$\kappa(\sigma) \quad = \quad \begin{cases} \mathtt{w} & \text{if } \sigma \notin \lceil\!\lceil O \rceil\!\rceil \\ \mathtt{c} & \text{if } \sigma \in \mathrm{CriticalStates}([\![O]\!]) \\ \mathtt{f} & \text{if } \sigma \in \lceil\!\lceil O \rceil\!\rceil \setminus \mathrm{CriticalStates}([\![O]\!]) \end{cases} \tag{7}$$

The proof follows from the definitions and theorem 8.

Proposition 9 closes the loop: finite degradation models, when assessed via minimal cutsets, can be seen as epimorphisms of **FDS**.

4.6 Probabilities

As in the binary case, minimal cutsets can be used to approximate probabilities of formulas via the so-called *rare event approximation*, denoted by p_{REA}, and *mincut upper bound*, denoted by p_{MCUB}, which are defined as follows.

Let $\mathcal{M} : \langle \mathcal{V} = \mathcal{S} \uplus \mathcal{F}, \mathcal{E} \rangle$ be a finite degradation model let O be an observer of $\mathcal{M}$ and let finally p be a random process associated with $dom(\mathcal{S})$. Then,

$$p_{REA}(O) \quad \stackrel{def}{=} \quad \sum_{\pi \in \mathrm{MCS}(O)} p(\pi)$$

$$p_{MCUB}(O) \quad \stackrel{def}{=} \quad 1 - \prod_{\pi \in \mathrm{MCS}(O)} 1 - p(\pi)$$

In practice, when the probabilities of atoms involved in minimal cutsets are sufficiently low and when the formula O is (nearly) coherent, both $p_{REA}(O)$ and $p_{MCUB}(O)$ provide

good approximation of $p(O)$.

$$p_{REA}(O) \approx p_{MCUB}(O) \approx p(O) \tag{8}$$

p_{MCUB} has the advantage over p_{REA} to be always comprised between 0 and 1, but the drawback to be less easy to calculate (especially when data structures such as zero-suppressed binary decision diagrams [28] are used to encode the minimal cutsets).

5 Experimental Results

This section presents some experimental results we obtained on the safety instrumented system presented Section 2. The whole model for this system can be interpreted as a function from $\mathbf{W3F}^7$ (as there are 7 basic components) into $\mathbf{W3F}$. However, logic solvers and valves can be only in three states: dangerous failures of logic solvers are immediately detected and dangerous failures of valves remain undetected between two tests. The system can thus be in $4^3 \times 3^4 = 5184$ states. This is indeed not very much, but adding a few components would make treatments requiring an explicit representation of the state space (like Markov chains) unfeasible (e.g. $3^{15} \approx 14 \times 10^6$ and $3^{20} \approx 3.4 \times 10^{12}$).

5.1 Assessment Techonology

For the purpose of the present article, we developed a package for Minato's zero-suppressed binary decision diagrams [28] that we adapted for the finite degradation calculus. This technique makes it possible to extract minimal cutsets as well as to calculate performance indicators.

The decision diagrams encoding the state of the HIPPS as well as the observers $O_{\texttt{fs}}$: `HIPPS = fs`, $O_{\texttt{fd}}$: `HIPPS = fd` and $O_{\texttt{fu}}$: `HIPPS = fu`, which encode the states in which the HIPPS is respectively failed safe, failed dangerous detected and failed dangerous undetected. These decision diagrams are made respectively of 89, 72, 52 and 79 nodes. The decision diagrams encoding the minimal cutsets of observers $O_{\texttt{fs}}$, $O_{\texttt{fd}}$ and $O_{\texttt{fu}}$ are made respectively 31, 7 and 17 nodes. All these diagrams as well as the following performance indicators presented below are calculated within in few seconds on an ordinary laptop (most of the computation time is taken by printing out results of calculations).

5.2 Minimal Cutsets

The observer $O_{\mathtt{fs}}$ has 37 minimal cutsets. Among these minimal cutsets, one finds the following ones.

$$\mathtt{S1} = \mathtt{fs} \wedge \mathtt{S2} = \mathtt{fs} \wedge \mathtt{S3} = \mathtt{fs}$$

$$\mathtt{S1} = \mathtt{fs} \wedge \mathtt{S2} = \mathtt{fs} \wedge \mathtt{S3} = \mathtt{fd}$$

$$\mathtt{S1} = \mathtt{fs} \wedge \mathtt{S2} = \mathtt{fd} \wedge \mathtt{S3} = \mathtt{fs}$$

$$\mathtt{S1} = \mathtt{fs} \wedge \mathtt{S2} = \mathtt{fd} \wedge \mathtt{S3} = \mathtt{fd}$$

The observer $O_{\mathtt{fd}}$ has the following 4 minimal cutsets.

$$\mathtt{LS1} = \mathtt{fd} \wedge \mathtt{LS2} = \mathtt{fd}$$

$$\mathtt{LS1} = \mathtt{fd} \wedge \mathtt{S2} = \mathtt{fd} \wedge \mathtt{S3} = \mathtt{fd}$$

$$\mathtt{S1} = \mathtt{fd} \wedge \mathtt{LS2} = \mathtt{fd}$$

$$\mathtt{S1} = \mathtt{fd} \wedge \mathtt{S2} = \mathtt{fd} \wedge \mathtt{S3} = \mathtt{fd}$$

Note that as the valves cannot be failed dangerous detected, no atom built over the variables V1 and V2 shows up in the minimal cutsets of observer $O_{\mathtt{fd}}$.

Finally, observer $O_{\mathtt{fu}}$ has the following 13 minimal cutsets.

$$\mathtt{V1} = \mathtt{fu} \wedge \mathtt{LS2} = \mathtt{fd}$$

$$\mathtt{V1} = \mathtt{fu} \wedge \mathtt{S2} = \mathtt{fd} \wedge \mathtt{S3} = \mathtt{fd}$$

$$\mathtt{V1} = \mathtt{fu} \wedge \mathtt{V2} = \mathtt{fu}$$

$$\mathtt{LS1} = \mathtt{fd} \wedge \mathtt{LS2} = \mathtt{fd} \wedge \mathtt{V2} = \mathtt{fu}$$

$$\mathtt{LS1} = \mathtt{fd} \wedge \mathtt{S2} = \mathtt{fu} \wedge \mathtt{S3} = \mathtt{fd}$$

$$\mathtt{LS1} = \mathtt{fd} \wedge \mathtt{S2} = \mathtt{fd} \wedge \mathtt{S3} = \mathtt{fu}$$

$$\mathtt{LS1} = \mathtt{fd} \wedge \mathtt{S2} = \mathtt{fd} \wedge \mathtt{S3} = \mathtt{fd} \wedge \mathtt{V2} = \mathtt{fu}$$

$$\mathtt{S1} = \mathtt{fu} \wedge \mathtt{LS2} = \mathtt{fd}$$

$$\mathtt{S1} = \mathtt{fu} \wedge \mathtt{S2} = \mathtt{fd} \wedge \mathtt{S3} = \mathtt{fd}$$

$$\mathtt{S1} = \mathtt{fd} \wedge \mathtt{LS2} = \mathtt{fd} \wedge \mathtt{V2} = \mathtt{fu}$$

$$\mathtt{S1} = \mathtt{fd} \wedge \mathtt{S2} = \mathtt{fu} \wedge \mathtt{S3} = \mathtt{fd}$$

$$\mathtt{S1} = \mathtt{fd} \wedge \mathtt{S2} = \mathtt{fd} \wedge \mathtt{S3} = \mathtt{fu}$$

$$\mathtt{S1} = \mathtt{fd} \wedge \mathtt{S2} = \mathtt{fd} \wedge \mathtt{S3} = \mathtt{fd} \wedge \mathtt{V2} = \mathtt{fu}$$

$Time$	$p(O_{\mathtt{fs}})$	$p(O_{\mathtt{fd}})$	$p(O_{\mathtt{fu}})$
$73h$	5.04×10^{-4}	3.34×10^{-9}	4.90×10^{-8}
$146h$	1.98×10^{-3}	1.34×10^{-8}	1.96×10^{-7}
$219h$	4.37×10^{-3}	3.01×10^{-8}	4.41×10^{-7}
$292h$	7.62×10^{-3}	5.35×10^{-8}	7.83×10^{-7}
$365h$	1.17×10^{-2}	8.36×10^{-8}	1.22×10^{-6}
$438h$	1.65×10^{-2}	1.20×10^{-7}	1.76×10^{-6}
$511h$	2.20×10^{-2}	1.64×10^{-7}	2.40×10^{-6}
$584h$	2.82×10^{-2}	2.14×10^{-7}	3.13×10^{-6}
$657h$	3.51×10^{-2}	2.71×10^{-7}	3.96×10^{-6}
$730h$	4.25×10^{-2}	3.34×10^{-7}	4.89×10^{-6}

Table 3: Probabilities of observers $O_{\mathtt{fs}}$, $O_{\mathtt{fd}}$ and $O_{\mathtt{fs}}$ at different times.

5.3 Probabilities of Failures

Table 3 shows the evolution of probabilities of observers with the mission time.

Several remarks can be made here.

First, probabilities of all observers increase with the time. This is not surprising because components are non-repairable (between two maintenance operations).

Second, the probability of safe failure is much higher than the probability of a dangerous one. Again, no surprise here, given the reliability parameters of the components. A practical consequence of that, confirmed by the industrial experience, is that most of the production down-time is due to maintenance operations and spurious triggers of safety systems.

Third, the probability of an undetected dangerous failure is one order of magnitude higher than the probability of a detected one. This is due to valves that have both a quite high (safe) failure rate and whose failures cannot be detected between inspections.

6 Conclusion

In this article, we introduced the notion of finite degradation structures. This notion provides a powerful and unified algebraic framework for Boolean and multistate models. It relies on three fundamental ideas.

First, states of Boolean and multistate models shows a finite semilattice structure. The partial order amongst the states is a degradation order. The bottom element of the semilattice is the nominal operating state.

Second, finite degradation structures can be composed, thanks to a monoidal product, which allows not only to assemble components into models, but also to reason in a uniform way on components, subsystems and systems. Technically, finite degradation structures form a symmetric monoidal category. Some very common finite degradation structures can be seen as on-the-shelf types for modeling components.

Third, epimorphisms between finite degradation structures describe abstractions between models. Many concepts of fault tree analysis can be reinterpreted and better understood by means of such epimorphisms.

We revisited here the familiar notions of coherence, minimal cutsets and importance measures from the new perspective of finite degradation structures. The nicest result we obtained is probably the isomorphism between minimal cutsets and critical states.

The objective of the present article was to present the theoretical foundations of finite degradation structures. In forthcoming articles, we shall discuss implementation issues, i.e. how to lift-up the existing algorithmic corpus on Boolean models to the finite degradation calculus, and modeling methodologies that can be deployed to take a full benefit of the new theoretical framework we introduced here.

References

[1] N. C. Rasmussen, "Reactor Safety Study. An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants," U.S. Nuclear Regulatory Commission, Rockville, MD, USA, Tech. Rep. WASH 1400, NUREG-75/014, October 1975.

[2] J. D. Andrews and R. T. Moss, *Reliability and Risk Assessment (second edition)*. Materials Park, Ohio 44073-0002, USA: ASM International, 2002.

[3] H. Kumamoto and E. J. Henley, *Probabilistic Risk Assessment and Management for Engineers and Scientists*. Piscataway, N.J., USA: IEEE Press, 1996.

[4] "International iec standard iec61508 - functional safety of electrical/electronic/programmable safety-related systems (e/e/pe, or e/e/pes)," International Electrotechnical Commission, Geneva, Switzerland, Standard, April 2010.

[5] "Iso26262 functional safety - road vehicle," International Standardization Organization, Geneva, Switzerland, Standard, 2012. [Online]. Available: http://www.iso.org/iso/home.html

[6] "Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment," Society of Automotive Engineers, Warrendale, Pennsylvania, USA, Standard, July 2004.

[7] M. Ajmone-Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modelling with Generalized Stochastic Petri Nets*, ser. Wiley Series in Parallel Computing. New York, NY, USA: John Wiley and Sons, 1994.

[8] A. Rauzy, "Notes on computational uncertainties in probabilistic risk/safety assessment," *Entropy*, vol. 20, no. 3, 2018.

[9] L. G. Valiant, "The complexity of enumeration and reliability problems," *SIAM Journal of Computing*, vol. 8, no. 3, pp. 410–421, 1979.

[10] A. Rauzy, "Mathematical Foundation of Minimal Cutsets," *IEEE Transactions on Reliability*, vol. 50, no. 4, pp. 389–396, december 2001.

[11] A. Lisnianski and G. Levitin, *Multi-State System Reliability*, ser. Quality, Reliability and Engineering Statistics. London, England: World Scientific, 2003.

[12] B. Natvig, *Multistate Systems Reliability Theory with Applications*. Hoboken, NJ, USA: Wiley, 2010.

[13] G. Levitin and L. Xing, Eds., *Reliability and Performance of Multi-State Systems*, vol. 166, October 2017.

[14] E. Zaitseva and V. Levashenko, "Reliability analysis of multi-state system with application of multiple-valued logic," *International Journal of Quality and reliability Management*, vol. 34, pp. 862–878, 2017.

[15] S. Awodey, *Category Theory*, ser. Oxford Logic Guides. Oxford, United Kingdom: Oxford University Press, 2010, vol. 52.

[16] A. Rauzy, "BDD for Reliability Studies," in *Handbook of Performability Engineering*, K. B. Misra, Ed. Amsterdam, the Netherlands: Elsevier, 2008, pp. 381–396.

[17] ——, "Anatomy of an efficient fault tree assessment engine," in *Proceedings of International Joint Conference PSAM'11/ESREL'12*, R. Virolainen, Ed., Helsinki, Finland, June 2012.

[18] J. B. Dugan, S. J. Bavuso, and M. A. Boyd, "Dynamic fault-tree models for fault-tolerant computer systems," *IEEE Transactions on Reliability*, vol. 41, no. 3, pp. 363–377, September 1992.

[19] A. Rauzy, "Modes Automata and their Compilation into Fault Trees," *Reliability Engineering and System Safety*, vol. 78, no. 1, pp. 1–12, October 2002.

[20] "Iso/tr 12489:2013 petroleum, petrochemical and natural gas industries – reliability modelling and calculation of safety systems," International Organization for Standardization, Geneva, Switzerland, Standard, November 2013.

[21] T. Prosvirnova and A. Rauzy, "Automated generation of minimal cutsets from altarica 3.0 models," *International Journal of Critical Computer-Based Systems*, vol. 6, no. 1, pp. 50–79, 2015.

[22] Y. Dutuit and A. Rauzy, "A Linear Time Algorithm to Find Modules of Fault Trees," *IEEE Transactions on Reliability*, vol. 45, no. 3, pp. 422–425, 1996.

[23] M. Markl, S. Shnider, and J. Stasheff, *Operads in Algebra, Topology and Physics*, ser. Mathematical Surveys and Monographs. Providence, RI, USA: American Mathematical Society, 2002.

[24] A. Rauzy, "Guarded transition systems: a new states/events formalism for reliability studies," *Journal of Risk and Reliability*, vol. 222, no. 4, pp. 495–505, 2008.

[25] S. M. Ross, *Introduction to Probability Models*. Cambridge, MA, USA: Academic Press, 2009.

[26] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ, USA: Princeton University Press, 1976.

[27] O. Nusbaumer and A. Rauzy, "Fault tree linking versus event tree linking approaches: a reasoned comparison," *Journal of Risk and Reliability*, vol. 227, no. 3, pp. 315–326, June 2013.

[28] S.-I. Minato, "Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems," in *Proceedings of the 30th ACM/IEEE Design Automation Conference, DAC'93*. Dallas, Texas, USA: IEEE, 1993, pp. 272–277.

Received 14 March 2019