Modèle algébrique des arbres de défaillance intégrant des contraintes sur l'ordre d'occurrence des événements

Guillaume Merle, Jean-Marc Roussel

Modélisation algébrique des arbres de défaillance temporels

Guillaume Merle, Jean-Marc Roussel

Laboratoire Universitaire de Recherche en Production Automatisée
École Normale Supérieure de Cachan, 61 Avenue du Président Wilson, 94235 Cachan Cedex, France

guillaume.merle@lurpa.ens-cachan.fr, jean-marc.roussel@lurpa.ens-cachan.fr

Résumé— **Les arbres de défaillance permettent de représenter graphiquement les combinaisons de défaillances susceptibles de conduire à un événement redouté. Cette communication présente un cadre formel permettant d'étendre la simplification des arbres de défaillance statiques aux arbres comportant des portes ET PRIORITAIRE et OU PRIORITAIRE.**

Mots-clés— **Arbres de défaillance, coupes minimales, portes temporelles, approche algébrique.**

I. Introduction

La sûreté de fonctionnement des systèmes industriels est une préoccupation majeure de notre société. Les activités industrielles et humaines font parfois les gros titres des actualités avec leurs cortèges d'incidents, d'accidents ou d'événements catastrophiques. En effet, le risque zéro n'existe pas à cause de l'occurrence de défaillances humaines ou matérielles. Toutefois, pour tenter de réduire les risques à un niveau le plus faible possible, des méthodes, des techniques et des outils scientifiques ont été développés dès le milieu du XX^{e} siècle pour évaluer les risques potentiels, prévoir l'occurrence des défaillances et tenter de minimiser les conséquences des situations catastrophiques lorsqu'elles se produisent [13]. L'analyse d'un système à l'aide d'*arbres de défaillance* [6] est l'une des méthodes utilisées pour l'étude a priori d'un système.

L'analyse d'un système par arbre de défaillance permet d'exprimer, pour un événement redouté (dysfonctionnement ou accident), les scénarii conduisant à cet événement. Cette étude se fait en s'appuyant sur la connaissance des éléments constitutifs du système étudié. Un arbre de défaillance décrit de manière synthétique tout ce qui peut conduire à l'événement redouté. Une *analyse qualitative* permet d'évaluer l'effet d'une modification du système et de comparer les conséquences des mesures envisagées. Une *analyse quantitative* consiste à évaluer la probabilité d'occurrence de l'événement redouté à partir de celles des événements élémentaires.

Pour construire un arbre de défaillance, il est proposé dans [6] une procédure itérative qui débute par l'événement redouté, appelé *événement sommet*. Cet événement constitue la racine de l'arbre. La méthode de construction consiste à faire l'hypothèse de l'occurrence de l'événement redouté et à en chercher la ou les causes immédiates (appelées *événements intermédiaires*). Le processus est réitéré jusqu'à aboutir à des événements pour lesquels la recherche des causes ne s'impose plus. Ces événements nommés *événements élémentaires* (ou *événements de base*) constituent les feuilles de l'arbre [7]. L'événement sommet, ainsi que les événements intermédiaires, de l'arbre sont liés à leurs causes immédiates par l'intermédiaire d'une *porte* décrivant comment ces causes se combinent. Lorsque toutes les causes sont nécessaires pour engendrer la conséquence, la porte utilisée est de type ET. Lorsque seule une des causes est nécessaire, la porte utilisée est de type OU.

La pratique industrielle a démontré la pertinence d'une telle procédure pour la phase de construction d'un arbre de défaillance. Elle a également permis d'identifier le principal défaut de cette procédure : les arbres ainsi obtenus ne sont pas directement exploitables car ils peuvent comporter des parties redondantes incompatibles avec une analyse qualitative ou quantitative [6].

Pour un arbre de défaillance, l'élimination des parties redondantes (ou simplification) porte le nom de calcul de l'*ensemble des coupes minimales*. Cette opération est réalisée par application des propriétés communément utilisées dans l'algèbre de Boole [10]. Il est implicitement fait l'hypothèse qu'un événement présent dans un arbre est parfaitement défini par la seule connaissance de son état (défaillant ou non défaillant), information modélisable à l'aide d'une simple variable booléenne.

Dans le Fault Tree Handbook [10], considéré par la communauté comme l'ouvrage de référence, deux portes spécifiques ont été définies pour permettre de décrire des contraintes temporelles sur l'ordre d'occurrence des événements (portes temporelles ET PRIORITAIRE et OU PRIORITAIRE). Bien que cet ouvrage ait été édité en 1981, ces portes sont encore aujourd'hui peu utilisées, malgré leur potentialité [4], en raison de l'impossibilité d'exploiter directement les modèles qui les utilisent. En effet, le calcul de l'ensemble des coupes minimales ne peut pas être appliqué sur des arbres de défaillance comportant ce type de portes car elles n'ont pas d'opérations équivalentes au sein de l'algèbre de Boole.

Face à ce constat, nous avons souhaité définir un cadre formel homogène et univoque dédié à l'étude des arbres de défaillance. Nous attendons de ce cadre formel la possibilité de donner aux portes temporelles une sémantique qui permette d'établir les propriétés nécessaires à la simplification des arbres de défaillance. Pour que ce cadre formel puisse se substituer au cadre mathématique actuellement utilisé, il doit impérativement proposer une sémantique pour les portes OU et ET qui permette de retrouver toutes les pro-

priétés nécessaires au calcul des coupes minimales.

Cette communication est centrée principalement sur le cadre formel que nous proposons. La section II est consacrée à la présentation détaillée de la problématique et des hypothèses retenues. Le cadre formel est décrit dans la section III. Il repose sur la définition mathématique de tous les éléments présents dans un arbre de défaillance : événements, portes, ... Cette définition nous a permis d'établir les propriétés nécessaires à la simplification des arbres de défaillance. L'applicabilité de ces propriétés est illustrée dans la section IV.

II. Problématique

Dans la suite de cette communication, pour éviter toute confusion entre le concept d'événement utilisé dans un arbre de défaillance et le concept d'événement utilisé pour les systèmes à événements discrets, nous désignerons les événements présents dans les arbres de défaillance sous le terme de *faute*.

Considérons l'arbre de défaillance de la figure 1. Il a pour faute sommet s, pour fautes élémentaires a, b, c, d, et pour fautes intermédiaires m, n, p, q. Il comporte cinq portes (trois portes OU, une porte ET et une porte ET PRIORITAIRE) dont les définitions proposées par le Fault Tree Handbook [10] sont rappelées table I.

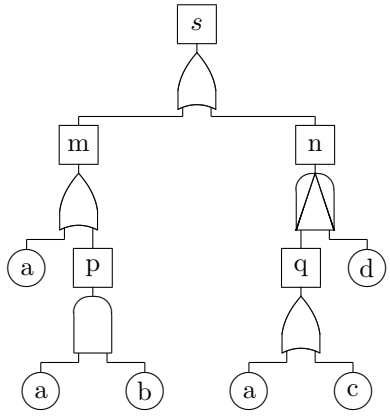


Fig. 1. *Arbre de défaillance comportant une porte temporelle*

Sous cette forme, cet arbre de défaillance n'est pas directement exploitable en raison des parties redondantes. Lorsque les arbres de défaillance sont *statiques* [3] (arbres ne comportant que des portes OU et ET), la seule connaissance de l'état des fautes élémentaires (défaillant ou non) est suffisante pour simplifier ces arbres. Pour l'arbre de la figure 1, la sous-arborescence m peut se réduire à la feuille a en s'appuyant sur la simplification suivante : $a + (a.b) = a$. Lorsque les arbres de défaillance sont *dynamiques* [3] (arbres comportant des portes temporelles, par exemple), cette seule connaissance de l'état des fautes ne suffit plus car les portes font également référence à l'ordre d'apparition des fautes (cf. définitions 3 et 4, table I). Le modèle booléen d'une faute ne comportant pas ce type d'information, il est évident que les propriétés classiques de l'algèbre de Boole ne peuvent pas apporter de solution.

De nombreux travaux ont été conduits pour donner une sémantique opérationnelle aux portes temporelles. Ces sémantiques sont essentiellement basées sur des modèles à états (réseaux de Petri [1], automates temporisés [2]) ou de logique temporelle [8] [12] afin de prendre en compte

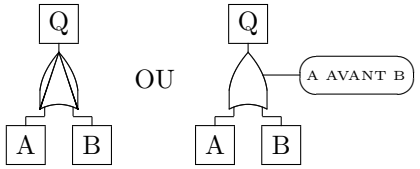
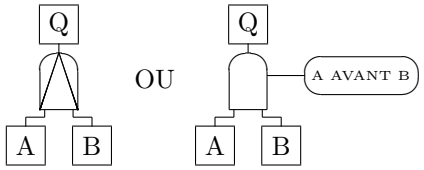
Symbole	Définition proposée par [10]
OU 	*La faute de sortie apparaît si au moins une des fautes d'entrée apparaît.* d'après figure IV-2 de [10]
ET 	*La faute de sortie apparaît si toutes les fautes d'entrée apparaissent.* d'après figure IV-5 de [10]
OU EXCLUSIF 	 d'après figure IV-11 de [10]
ET PRIORITAIRE 	 d'après figure IV-12 de [10]

TABLE I

Définition proposée par [10] des portes objet de notre étude

l'ordre d'apparition des fautes. Les modèles ainsi obtenus évitent de calculer explicitement les ensembles de coupes minimales. Dans [9] [11], les auteurs proposent de remplacer le calcul des ensembles de coupes minimales par le calcul des ensembles de séquences minimales.

L'approche que nous proposons est purement algébrique. Elle a pour objectif la définition de l'ensemble des propriétés nécessaires à la simplification d'arbres de défaillance temporels. Les travaux présentés concernent les arbres élaborés avec les quatre portes présentées dans la table I, dont la définition proposée est extraite de [10].

Deux points sont à noter vis-à-vis de cette table :

- la définition que donnent de nombreux auteurs de la porte OU EXCLUSIF n'est pas en accord avec celle donnée par le Fault Tree Handbook : la définition commune est que la faute de sortie est présente si une et une seule faute d'entrée est présente, ce qui est en désaccord avec la notion de priorité entre fautes. Pour éviter toute confusion, nous renommons la porte OU EXCLUSIF du Fault Tree Handbook porte OU PRIORITAIRE, comme le proposent les auteurs de [11].
- il y a ambiguïté au niveau du terme "AVANT" : il n'est pas précisé si ce terme est considéré au sens strict (A doit apparaître avant que B n'apparaisse) ou au sens large (A doit apparaître avant que B n'apparaisse ou en même temps que B).

Les travaux présentés dans la suite de cette communication ont été conduits avec les hypothèses suivantes :

- la classe de fautes étudiée se limite aux seules fautes non réparables (fautes persistantes).
- il n'y a aucune restriction sur l'ordre d'apparition des fautes. Deux fautes élémentaires peuvent apparaître

simultanément.
- le terme "AVANT" sera vu au sens strict. Cette hypothèse de travail correspond à celle retenue par [11].

En tenant compte de ces hypothèses, le comportement attendu pour les quatre portes étudiées peut être spécifié à l'aide de chronogrammes faisant apparaître le comportement de la sortie Q en fonction du comportement des entrées A et B. Les trois cas à envisager sont : A apparaît avant B, A apparaît en même temps que B, A apparaît après B.

Le comportement attendu des quatre portes étudiées est décrit table II (pour éviter tout risque de confusion, la valeur de A, B et Q aux points de discontinuité est représentée par un point noir).

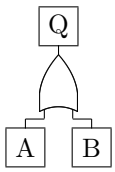
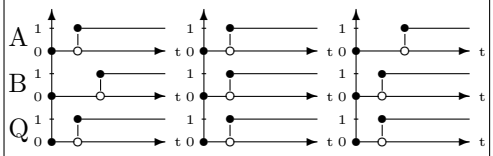
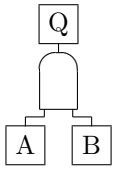
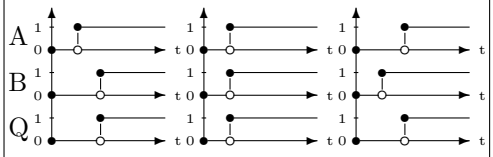
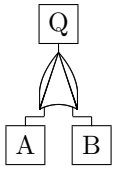
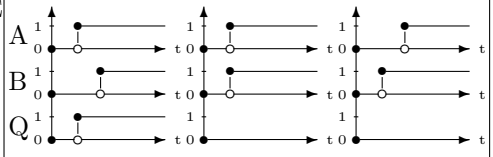
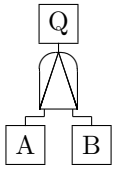
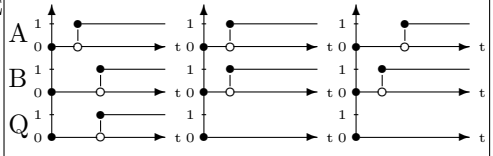
Symbole	Comportement attendu
OU 	
ET 	
OU PRIORITAIRE 	
ET PRIORITAIRE 	

TABLE II

Comportement attendu pour les quatre portes étudiées dans le cas de fautes non réparables

III. Cadre formel proposé

L'objectif de nos travaux étant l'obtention des propriétés nécessaires à la simplification des arbres de défaillance comportant des portes OU PRIORITAIRE et ET PRIORITAIRE, il nous est paru naturel de formaliser les portes d'un arbre de défaillance à l'aide d'une description mathématique pour disposer d'une forme algébrique adaptée à ce type de calcul. Pour éviter toute ambiguïté lors de cette phase de modélisation, nous avons tenu à définir mathématiquement chacun des concepts et éléments présents dans un arbre de défaillance. C'est d'ailleurs l'objet de la première partie de cette section. La deuxième partie est consacrée aux modèles des portes OU et ET. Il y est présenté leur modèle mathématique et la liste des propriétés qu'il a été possible de démontrer à partir de celui-ci. La troisième partie est consacrée à la loi de composition interne "AVANT" définie pour modéliser les portes temporelles. Le modèle mathématique des quatre portes étudiées est donné dans la quatrième partie.

A. Définition mathématique des concepts présents dans un arbre de défaillance

A.1 Fautes non réparables

Dans le cas des fautes non réparables ou persistantes, la connaissance de la date d'apparition d'une faute permet de décrire cette faute sans ambiguïté à condition de considérer l'occurrence des fautes comme instantanée. Avant cette date, la faute est absente, à cette date et après cette date, la faute est présente. En tenant compte de ces hypothèses, une faute non réparable peut être décrite par le chronogramme de la figure 2.

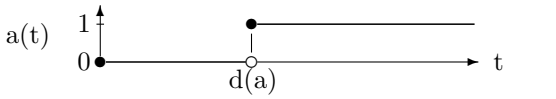


Fig. 2. *Une faute non réparable*

D'un point de vue mathématique, une faute non réparable est une fonction continue à droite par morceaux sur $\mathbb{R}^+$ et à valeurs dans $\mathbb{B} = \{0,1\}$ admettant au plus un seul changement de valeur (une seule discontinuité). Cette fonction respecte les caractéristiques suivantes :
- la fonction vaut 0 tant qu'il n'y a pas de faute,
- la fonction vaut 1 dès que la faute est apparue.

Soit $\mathcal{F}_{nr}$ l'ensemble des fautes non réparables. $\mathcal{F}_{nr}$ contient deux éléments particuliers car constants. Nous appellerons e la faute toujours présente ($\forall t \in \mathbb{R}^+$, $e(t) = 1$) et ϵ l'absence de faute ($\forall t \in \mathbb{R}^+$, $\epsilon(t) = 0$).

Définition 1 (**Date d'apparition d'une faute**) On appelle date d'apparition de la faute a (notée $d(a)$) l'instant t où la faute apparaît.

Cette date est mise en évidence sur la figure 2. Pour les fautes constantes, nous avons $d(e) = 0$ et $d(\epsilon) = +\infty$.

Définition 2 (**Équivalence entre fautes**) Soient a et b deux éléments de $\mathcal{F}_{nr}$. Les fautes a et b sont dites *équivalentes* ($a \approx b$) si et seulement si elles ont la même date d'apparition ($d(a) = d(b)$).

A.2 Fonctions de faute

Un arbre de défaillance décrit comment doivent se combiner les fautes élémentaires pour produire la faute sommet. D'un point de vue mathématique, la faute sommet peut être vue comme l'image du n-uplet des fautes élémentaires par la fonction de faute dont l'arbre de défaillance considéré est l'expression.

Définition 3 (**Fonction de faute**) On appelle fonction de faute d'ordre n toute application de $(\mathcal{F}_{nr})^n \longrightarrow \mathcal{F}_{nr}$ (avec $n \in \mathbb{N}^*$).

Soit $\Psi = \{f : (\mathcal{F}_{nr})^n \longrightarrow \mathcal{F}_{nr}\}$ l'ensemble des fonctions de faute. Ψ contient deux fonctions de faute particulières car constantes. Ces fonctions de faute sont notées $\top$ et $\bot$ et sont définies comme suit.

Définition 4 (Fonction $\top$) $\top$ est la fonction de faute qui, à tout n-uplet $(a_1, a_2, \ldots, a_n) \in (\mathcal{F}_{nr})^n$, associe la faute toujours présente e :

$$\begin{array}{rccc} \top : & (\mathcal{F}_{nr})^n & \longrightarrow & \mathcal{F}_{nr} \\ & (a_1, a_2, \ldots, a_n) & \longmapsto & e \end{array}$$

Définition 5 (Fonction $\bot$) $\bot$ est la fonction de faute qui, à tout n-uplet $(a_1, a_2, \ldots, a_n) \in (\mathcal{F}_{nr})^n$, associe l'absence de faute ϵ :

$$\begin{array}{rccc} \bot : & (\mathcal{F}_{nr})^n & \longrightarrow & \mathcal{F}_{nr} \\ & (a_1, a_2, \ldots, a_n) & \longmapsto & \epsilon \end{array}$$

Définition 6 (**Équivalence entre fonctions de faute**) Soient f et g deux éléments de Ψ. Les fonctions de faute f et g sont dites *équivalentes* ($f \sim g$) si et seulement si $\forall (a_1, a_2, \ldots, a_n) \in (\mathcal{F}_{nr})^n$,

$$f(a_1, a_2, \ldots, a_n) \approx g(a_1, a_2, \ldots, a_n)$$

Remarque : Les concepts de *faute* et de *fonction de faute* sont tous deux mathématiquement décrits à l'aide de fonctions, mais ces fonctions sont de nature différente :

- les *fautes* sont des fonctions de $\mathbb{R}^+ \rightarrow \mathbb{B}$ dont l'ensemble est noté $\mathcal{F}_{nr}$,
- les *fonctions de faute* sont des fonctions de $(\mathcal{F}_{nr})^n \rightarrow \mathcal{F}_{nr}$ dont l'ensemble est noté Ψ.

A.3 Expression d'une fonction de faute

Un arbre de défaillance est l'expression, donnée sous la forme graphique, d'une fonction de faute définie sur le n-uplet des fautes élémentaires présentes dans cet arbre. Simplifier cet arbre de défaillance consiste à rechercher pour cette fonction de faute une expression équivalente qui ne comporte pas de parties redondantes. L'arbre de défaillance présenté sur la figure 1 est la représentation graphique de la fonction de faute qui, au quadruplet de fautes élémentaires (a, b, c, d), associe la faute sommet s.

Définition 7 (**Équivalence entre arbres**) Deux arbres de défaillance sont dits *équivalents* si et seulement si ils permettent de décrire la même fonction de faute ou deux fonctions de faute équivalentes.

Nous retrouvons entre un arbre de défaillance et une fonction de faute le même lien qu'entre une expression et une fonction booléennes. Ce lien est le suivant :

- une fonction booléenne peut être représentée par plusieurs expressions booléennes,
- deux expressions booléennes sont équivalentes si elles représentent la même fonction booléenne,
- le passage d'une expression booléenne à une autre s'appuie sur les propriétés établies pour des *lois de composition interne* définies sur l'ensemble des fonctions booléennes [5].

Par analogie aux expressions et fonctions booléennes, nous avons retenu de simplifier les arbres de défaillance en nous appuyant sur les propriétés qu'il est possible d'établir pour les lois de composition interne définies sur l'ensemble des fonctions de faute.

A.4 Modèle mathématique retenu pour les éléments d'un arbre de défaillance

Chaque type de porte d'un arbre de défaillance est caractérisé par une loi de composition interne sur un ensemble de fonctions de faute donné. Il s'agit de l'ensemble des fonctions de faute définies sur le n-uplet des fautes élémentaires présentes dans l'arbre de défaillance. La faute sommet, comme chaque faute intermédiaire, est l'image, par une fonction de faute, des n fautes élémentaires. Dans un souci d'homogénéité, il est également nécessaire de considérer chaque feuille de l'arbre comme une fonction de faute. Cette fonction de faute est particulière puisque l'image du n-uplet de fautes élémentaires par cette fonction est la faute élémentaire dont elle porte le nom.

Définition 8 (**Fonction de faute élémentaire**) On appelle fonction de faute élémentaire chacune des n fonctions de faute σ_i ($i \in \{1, 2, \ldots, n\}$) définies par :

$$\begin{array}{rccc} \sigma_i : & (\mathcal{F}_{nr})^n & \longrightarrow & \mathcal{F}_{nr} \\ & (a_1, a_2, \ldots, a_i, \ldots, a_n) & \longmapsto & a_i \end{array}$$

Chaque feuille de l'arbre de défaillance présenté sur la figure 1 est l'une des quatre fonctions de faute élémentaires.

B. Lois de composition interne OU et ET

Dans la suite de ce document, nous noterons $\mathcal{A}$ tout n-uplet $(a_1, a_2, \ldots, a_n)$ de $(\mathcal{F}_{nr})^n$. Soient $f, g, h \in \Psi$.

B.1 Définition des lois de composition interne OU et ET

Ces lois de composition interne modélisent le comportement des portes OU et ET décrit dans la table II.

Définition 9 (**Loi de composition interne OU**)

$$\begin{array}{rccc} + : & \Psi \times \Psi & \longrightarrow & \Psi \\ & (f, g) & \longmapsto & f + g \end{array}$$

$f + g$ étant défini, pour tout $\mathcal{A} \in (\mathcal{F}_{nr})^n$, par :

$$(f+g)(\mathcal{A}) = \begin{cases} f(\mathcal{A}) & \text{si } d(f(\mathcal{A})) < d(g(\mathcal{A})) \\ g(\mathcal{A}) & \text{si } d(f(\mathcal{A})) > d(g(\mathcal{A})) \\ f(\mathcal{A}) & \text{si } d(f(\mathcal{A})) = d(g(\mathcal{A})) \end{cases}$$

Définition 10 (**Loi de composition interne ET**)

$$\begin{array}{rccc} . : & \Psi \times \Psi & \longrightarrow & \Psi \\ & (f, g) & \longmapsto & f.g \end{array}$$

$f.g$ étant défini, pour tout $\mathcal{A} \in (\mathcal{F}_{nr})^n$, par :

$$(f.g)(\mathcal{A}) = \begin{cases} g(\mathcal{A}) & \text{si } d(f(\mathcal{A})) < d(g(\mathcal{A})) \\ f(\mathcal{A}) & \text{si } d(f(\mathcal{A})) > d(g(\mathcal{A})) \\ f(\mathcal{A}) & \text{si } d(f(\mathcal{A})) = d(g(\mathcal{A})) \end{cases}$$

Remarque : Ces définitions semblent privilégier f par rapport à g lorsque $d(f(\mathcal{A})) = d(g(\mathcal{A}))$. Ce n'est pas le cas, puisque nous avons démontré que les lois de composition interne OU et ET sont commutatives.

B.2 Propriétés des lois de composition interne OU et ET

Ces deux définitions ont permis de démontrer les 14 propriétés suivantes :

$$f + g \sim g + f \quad (1)$$
$$f.g \sim g.f \quad (2)$$
$$f + (g + h) \sim (f + g) + h \quad (3)$$
$$f.(g.h) \sim (f.g).h \quad (4)$$
$$f + f \sim f \quad (5)$$
$$f.f \sim f \quad (6)$$
$$f + (g.h) \sim (f + g).(f + h) \quad (7)$$
$$f.(g + h) \sim (f.g) + (f.h) \quad (8)$$
$$f + (f.g) \sim f \quad (9)$$
$$f.(f + g) \sim f \quad (10)$$
$$f + \bot \sim f \quad (11)$$
$$f.\top \sim f \quad (12)$$
$$f + \top \sim \top \quad (13)$$
$$f.\bot \sim \bot \quad (14)$$

L'obtention des propriétés 1 à 10 était pour nous un point de passage obligé car ces propriétés sont celles utilisées pour le calcul de l'ensemble des coupes minimales des arbres de défaillance statiques.

Pour démontrer ces propriétés, nous avons utilisé deux sortes de démonstration en fonction du nombre de fonctions de faute participant à la relation :

- si une propriété utilisait deux fonctions de faute f et g, cette propriété a été démontrée en considérant 3 cas différents, selon la date d'apparition des deux fautes $f(\mathcal{A})$ et $g(\mathcal{A})$: 2 cas pour lesquels les deux dates étaient différentes et 1 cas pour lequel elles étaient égales,
- si une propriété utilisait trois fonctions de faute f, g et h, cette propriété a été démontrée en considérant 13 cas différents, selon la date d'apparition des trois fautes $f(\mathcal{A})$, $g(\mathcal{A})$ et $h(\mathcal{A})$: 6 cas pour lesquels les trois dates étaient différentes, 6 cas pour lesquels deux dates sur les trois étaient égales, et 1 cas pour lequel les trois dates étaient égales.

Lorsque cela a été possible, toute nouvelle propriété a été démontrée en utilisant les propriétés déjà établies.

C. Loi de composition interne AVANT

Cette loi a été introduite pour modéliser le concept de priorité entre fautes sur lequel repose la définition des portes OU PRIORITAIRE et ET PRIORITAIRE.

C.1 Définition

Le comportement attendu pour la composition de f et g par la loi de composition interne AVANT (notée $f \lhd g$) est illustré par les chronogrammes de la figure 3 (Cas 1 : $d(f(\mathcal{A})) < d(g(\mathcal{A}))$, Cas 2 : $d(f(\mathcal{A})) = d(g(\mathcal{A}))$, Cas 3 : $d(f(\mathcal{A})) > d(g(\mathcal{A}))$).

Définition 11 (**Loi de composition interne AVANT**)

$$\begin{array}{rccc} \lhd : & \Psi \times \Psi & \longrightarrow & \Psi \\ & (f, g) & \longmapsto & f \lhd g \end{array}$$

$f \lhd g$ étant défini, pour tout $\mathcal{A} \in (\mathcal{F}_{nr})^n$, par :

$$(f \lhd g)(\mathcal{A}) = \begin{cases} f(\mathcal{A}) & \text{si } d(f(\mathcal{A})) < d(g(\mathcal{A})) \\ \bot(\mathcal{A}) & \text{si } d(f(\mathcal{A})) \geq d(g(\mathcal{A})) \end{cases}$$

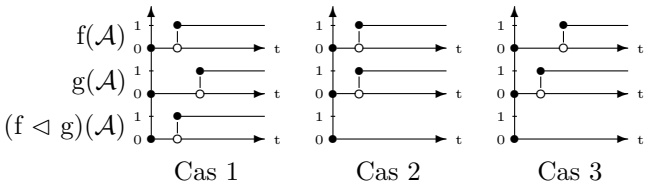


Fig. 3. Comportement attendu pour $(f \lhd g)(\mathcal{A})$

C.2 Propriétés de la loi de composition interne AVANT

Ces trois définitions ont permis de démontrer les 13 propriétés suivantes :

$$f + (f \lhd g) \sim f \quad (15)$$
$$g + (f \lhd g) \sim f + g \quad (16)$$
$$f.(f \lhd g) \sim f \lhd g \quad (17)$$
$$f + ((f \lhd g).h) \sim f \quad (18)$$
$$f \lhd (g + h) \sim (f \lhd g).(f \lhd h) \quad (19)$$
$$(f + g) \lhd h \sim (f \lhd h) + (g \lhd h) \quad (20)$$
$$f \lhd (g.h) \sim (f \lhd g) + (f \lhd h) \quad (21)$$
$$(f.g) \lhd h \sim (f \lhd h).(g \lhd h) \quad (22)$$
$$(f \lhd g).(g \lhd h).(f \lhd h) \sim (f \lhd g).(g \lhd h) \quad (23)$$
$$(f \lhd g).(g \lhd f) \sim \bot \quad (24)$$
$$f \lhd f \sim \bot \quad (25)$$
$$f \lhd \bot \sim f \quad (26)$$
$$\bot \lhd f \sim \bot \quad (27)$$

L'obtention des propriétés 15 à 27 était l'objectif opérationnel de ce travail. Associées aux propriétés 1 à 14, elles permettent la simplification des arbres de défaillance comportant les portes temporelles étudiées. Ces propriétés sont *suffisantes* pour traiter des arbres de défaillance ne présentant pas de combinaisons imbriquées de portes temporelles (portes temporelles dont les arborescences d'entrée comportent elles-même des portes temporelles).
Pour être à même de simplifier tout arbre temporel quelle que soit sa structure, nous sommes actuellement en train de déterminer les propriétés de développement pour les formes suivantes : $f \lhd (g \lhd h)$ et $(f \lhd g) \lhd h$.

D. Modèle mathématique des portes

La table III reprend les éléments de la table I et présente le modèle mathématique associé à chaque porte. Pour chacune des portes temporelles, les deux formulations ont été démontrées comme équivalentes.

IV. Simplification d'arbres de défaillance comportant des portes temporelles

Les propriétés présentées ci-dessus permettent de simplifier l'arbre de la figure 1. La méthode consiste à exprimer la sortie de chaque porte en fonction de ses entrées, à développer l'expression obtenue et à la simplifier (dans un souci de lisibilité, les fonctions de faute élémentaires seront

notées de la même manière que la faute élémentaire à laquelle elles font référence). L'arbre de défaillance simplifié équivalent à celui de la figure 1 est représenté figure 4.

$$
\begin{aligned}
s \quad &\sim \quad m + n \sim (a + p) + (d.(q \lhd d)) \\
&\sim \quad (a + (a.b)) + (d.((a + c) \lhd d)) \\
&\overset{(9,20)}{\sim} \quad a + (d.((a \lhd d) + (c \lhd d))) \\
&\overset{(3,8)}{\sim} \quad a + d.(a \lhd d) + d.(c \lhd d) \\
&\overset{(2,18)}{\sim} \quad a + d.(c \lhd d)
\end{aligned}
$$

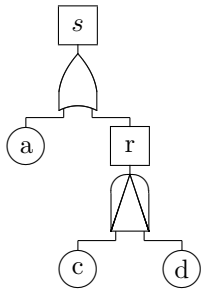


Fig. 4. *Arbre simplifié équivalent à celui de la figure 1*

V. Conclusion

Le premier apport de ce travail est la définition d'un cadre homogène et univoque pour la formalisation des portes temporelles. Ce cadre nous a permis de déterminer des propriétés nécessaires à la simplification d'arbres de défaillance comportant des portes OU PRIORITAIRE et ET PRIORITAIRE. C'est à notre avis le principal apport de ce travail. Il convient de rappeler que ces propriétés ont été établies avec les hypothèses suivantes : les fautes élémentaires sont non réparables, et la notion de priorité est vue au sens strict.

Les travaux en cours portent sur l'obtention des formes développées des deux propriétés nécessaires pour simplifier des arbres présentant des compositions de portes temporelles. Nous travaillons également sur le développement d'un module de calcul symbolique, dédié à la simplification d'un arbre de défaillance, qui s'appuie sur l'ensemble de ces propriétés.

À court terme, nous souhaitons étudier l'impact de l'hypothèse retenue sur la priorité (priorité stricte) en développant une nouvelle loi pour laquelle la faute de sortie serait également émise si les deux fautes d'entrée sont simultanées (à l'opposé du cas 2 de la figure 3). Nous envisageons également d'élargir ces travaux aux autres portes temporelles.

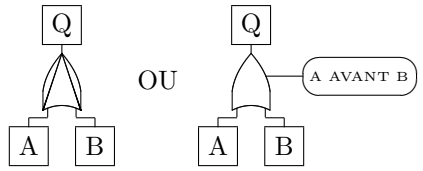
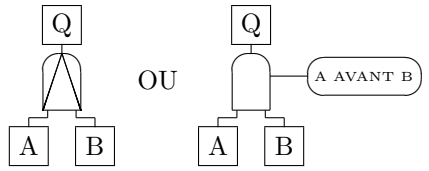
Symbole	Modélisation algébrique
OU 	Extrait de [10] : *La faute de sortie apparaît si au moins une des fautes d'entrée apparaît.* Modélisation algébrique proposée : $\boldsymbol{Q = A+B}$
ET 	Extrait de [10] : *La faute de sortie apparaît si toutes les fautes d'entrée apparaissent.* Modélisation algébrique proposée : $\boldsymbol{Q = A.B}$
OU PRIORITAIRE 	Extrait de [10] :  Modélisation algébrique proposée : $\boldsymbol{Q = (A+B).(A \lhd B)}$ $\boldsymbol{= A \lhd B}$
ET PRIORITAIRE 	Extrait de [10] :  Modélisation algébrique proposée : $\boldsymbol{Q = (A.B).(A \lhd B)}$ $\boldsymbol{= B.(A \lhd B)}$

TABLE III
Description formelle des différentes portes

Références

[1] A. Adamyan et D. He. *Sequential Failure Analysis Using Counters of Petri Net Models.* IEEE Transactions on Systems, Man, and Cybernetics - part A : Systems and Humans, vol. 33, n°1, Janvier 2003.

[2] I. Barragan Santiago, J.-M. Faure et Y. Papadopoulos. *Including systematic faults into fault tree analysis.* Proceedings of the 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS'2006), pp. 811-816, Chine, Août-Sept. 2006.

[3] D. Coppit, K. J. Sullivan et J. B. Dugan. *Formal Semantics of Models for Computational Engineering : a Case Study on Dynamic Fault Trees.* International Symposium on Software Reliability Engineering (ISSRE'2000), pp. 270-282, Oct. 2000.

[4] D. B. Dugan, K. J. Sullivan et D. Coppit. *Developing a Low-Cost High-Quality Software Tool for Dynamic Fault-Tree Analysis.* IEEE Transactions on Reliability, vol. 49, n°1, Mars 2000.

[5] R. P. Grimaldi. *Discrete and Combinatorial Mathematics.* Addison-Wesley, 5e édition, 2003.

[6] Y. Mortureux. *Arbres de défaillance, des causes et d'événement.* Techniques de l'Ingénieur, article SE4050, 24 pages, Oct. 2002.

[7] E. Niel et E. Craye. *Maîtrise des risques et sûreté de fonctionnement des systèmes de production.* Hermes Science, 2002.

[8] G. K. Palshikar. *Temporal fault trees.* Information and Software Technology **44**, pp. 137-150, 2002.

[9] Z. Tang et J. B. Dugan. *Minimal Cut Set/Sequence Generation for Dynamic Fault Trees.* Annual Reliability and Maintainability Symposium 2004 Proceedings, Los Angeles, Janv. 2004.

[10] W. E. Vesely, F. F. Goldberg, N. H. Roberts et D. F. Haasl. *Fault Tree Handbook.* Washington D.C., USA, *US Nuclear Regulatory Commission*, 1981.

[11] M. Walker et Y. Papadopoulos. *PANDORA : The time of Priority-AND gates.* INCOM'06, 12th IFAC Symposium on Information Control Problems in Manufacturing, France, Mai 2006.

[12] J. Xiang. *Fault Tree Analysis and Formal Methods for Requirements Engineering.* Thèse de doctorat, Japan Advanced Institute of Science and Technology, Sept. 2005.

[13] G. Zwingelstein. *Sûreté de fonctionnement des systèmes industriels complexes.* Techniques de l'Ingénieur, article S8250, 32 pages, Sept. 1999.