



HAL
open science

Algebraic modelling of Dynamic Fault Trees, contribution to qualitative and quantitative analysis

Guillaume Merle

► **To cite this version:**

Guillaume Merle. Algebraic modelling of Dynamic Fault Trees, contribution to qualitative and quantitative analysis. Automatic. École normale supérieure de Cachan - ENS Cachan, 2010. English. NNT: . tel-00502012v2

HAL Id: tel-00502012

<https://theses.hal.science/tel-00502012v2>

Submitted on 26 Jul 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N° ENSC-2010/233

**THÈSE DE DOCTORAT
DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN**

présentée par
Guillaume Merle

pour obtenir le grade de
DOCTEUR DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Spécialité :
ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE

**Algebraic modelling of Dynamic Fault Trees, contribution to
qualitative and quantitative analysis**

Analyse algébrique des arbres de défaillance dynamiques,
contribution aux analyses qualitative et quantitative

Thèse présentée et soutenue à Cachan le 7 juillet 2010 devant le jury composé de :

A. Bobbio	Professeur, Université du Piémont Oriental (Italie)	Président
C. Bérenguer	Professeur, Université de Technologie de Troyes	Rapporteur
F. Ortmeier	Professeur, Université de Magdeburg (Allemagne)	Rapporteur
A. Rauzy	CR CNRS HDR, Directeur R&D Dassault Systèmes	Examineur
J.-J. Lesage	Professeur, ENS de Cachan	Directeur de thèse
J.-M. Roussel	Maître de Conférences, ENS de Cachan	Co-encadrant



Laboratoire Universitaire de Recherche en Production Automatisée
(ENS CACHAN / EA 1385)
61 avenue du Président Wilson, Cachan Cedex, 94235, France

Acknowledgments

This PhD manuscript is the result of a three-year-long period, three years that I have spent at the Automated Production Research Laboratory (LURPA) of the École Normale Supérieure de Cachan, headed by Prof. Dr. Jean-Jacques Lesage when I arrived. I would hence first like to thank Prof. Dr. Jean-Jacques Lesage for accepting me in his laboratory. I am also thankful to Prof. Dr. Jean-Marc Faure, who welcomed me in the ISA team, the team in which I have been carrying out my research work over these years.

Many people have advised me and helped me during my PhD dissertation and the writing of this manuscript and I would like to thank them all.

First of all, I would like to express my deepest gratitude to Prof. Dr. Jean-Jacques Lesage for being my supervisor. His human qualities and charisma are exceptional; this manuscript would not be what it is without his help.

I would like to thank Dr. Jean-Marc Roussel, first for making me join the LURPA, and also for accepting to be my co-supervisor. I would like to thank him for his constant support and for his everlasting availability during these three years that we have been working together.

I am deeply grateful to Prof. Dr. Frank Ortmeier and to Prof. Dr. Christophe Bérenguer for accepting to review my work and I thank them for their precious advice which allowed me to improve this manuscript and make it clearer. I am also thankful to Prof. Dr. Andrea Bobbio and to Dr. Habil. Antoine Rauzy for spending time reading my dissertation and for agreeing to serve on my doctoral committee.

As a PhD represents a very long and very hard period of time which cannot be lived alone, I would like to acknowledge as well the many colleagues, teachers, students, and friends who assisted, advised, and supported my research and writing efforts over the years. A special thanks to Bruno, Said, Nathalie, and Marc, amongst others, and to Nicolas and Monique, without whom this manuscript would not be what it currently is. I would like to thank my family for supporting me for so many years. My final thanks go to Matthieu whose friendship helped me along these years.

Contents

1	Introduction	1
2	State of the art	5
2.1	Static Fault Trees	6
2.1.1	Definitions	6
2.1.2	Analysis methods	8
2.2	Dynamic Fault Trees	12
2.2.1	Definitions	12
2.2.2	Problem	15
2.2.3	Solutions	15
2.2.4	Contribution of this work	24
3	Algebraic framework	27
3.1	Hypotheses	29
3.1.1	Non-repairable events	29
3.1.2	Continuous failure time distributions	29
3.1.3	Statistical independence	30
3.1.4	Simultaneity	30
3.2	Temporal model of non-repairable events	31
3.3	Temporal model of Boolean operators	32
3.3.1	Operator OR	32
3.3.2	Operator AND	33
3.3.3	Algebraic structure of $(\mathcal{E}_{nr}, +, \cdot)$	33
3.4	Temporal operators	35
3.4.1	Operator non-inclusive BEFORE	35
3.4.2	Operator SIMULTANEOUS	36
3.4.3	Operator Inclusive BEFORE	37
3.5	Theorems	37
3.5.1	Theorems satisfied by operator non-inclusive BEFORE	38
3.5.2	Theorems satisfied by operator SIMULTANEOUS	39
3.5.3	Theorems satisfied by operator Inclusive BEFORE	40
3.5.4	Simplification theorems	41
3.6	Discussion on the similarities and differences between our approach and Pandora	41
4	behavioural model of dynamic gates	43
4.1	behavioural model of the PAND gate	44
4.2	behavioural model of the FDEP gate	44
4.3	behavioural model of Spare gates	45
4.3.1	behavioural model of Spare gates with 2 input events	46

4.3.2	behavioural model of Spare gates with 3 input events	48
4.3.3	behavioural model of Spare gates with n input events	48
4.3.4	Specific case of Cold and Hot spare events	49
5	Probabilistic model of dynamic gates	51
5.1	A few probabilistic expressions	52
5.2	Probabilistic model of the PAND gate	53
5.3	Probabilistic model of the FDEP gate	53
5.4	Probabilistic model of Spare gates	53
5.4.1	Failure distribution of spare events	54
5.4.2	Probabilistic model of Spare gates with 2 input events	55
5.4.3	Probabilistic model of Spare gates with 3 input events	60
5.4.4	Probabilistic model of Spare gates with n input events	61
6	Analysis of Dynamic Fault Trees	63
6.1	Dynamic FTA based on the structure function	64
6.1.1	Minimal canonical form of the structure function of Dynamic Fault Trees	64
6.1.2	Qualitative analysis of DFTs based on the structure function	67
6.1.3	Quantitative analysis of DFTs based on the structure function	68
6.2	Application examples	69
6.2.1	The Hypothetical Example Computer System (HECS)	70
6.2.2	The Hypothetical Cardiac Assist System (HCAS)	71
6.3	Dynamic FTA of the HECS	72
6.3.1	Minimal canonical form of the structure function of the HECS	72
6.3.2	Qualitative analysis of the HECS	76
6.3.3	Quantitative analysis of the HECS	78
6.4	Dynamic FTA of the HCAS	80
6.4.1	Minimal canonical form of the structure function of the HCAS	80
6.4.2	Qualitative analysis of the HCAS	85
6.4.3	Quantitative analysis of the HCAS	87
7	Conclusions	91
	Bibliography	93
A	Proofs of the theorems	101
A.1	Principle of the proof of the theorems	101
A.1.1	Theorems involving 2 events	101
A.1.2	Theorems involving 3 events	101
A.2	Proofs	102
A.2.1	Proofs of the theorems satisfied by Boolean operators	102
A.2.2	Proofs of the theorems satisfied by operator non-inclusive BE- FORE	110
A.2.3	Proofs of the theorems satisfied by operator SIMULTANEOUS	141

A.2.4 Proofs of the theorems satisfied by operator Inclusive BEFORE 162
 A.2.5 Proofs of simplification theorems 182

B Application of probabilistic models to the case of exponential distributions 189

B.1 Case of a single Spare gate with 2 input events 189
 B.1.1 Result obtained with Markov Chains 189
 B.1.2 Result obtained with our probabilistic model 190
 B.2 Case of 2 Spare gates sharing a spare event 191
 B.2.1 Result obtained with Markov Chains 191
 B.2.2 Result obtained with our probabilistic model 193
 B.3 Case of a single Spare gate with 3 input events 200
 B.3.1 Result obtained with Markov Chains 200
 B.3.2 Result obtained with our probabilistic model 201

List of Figures

2.1	An OR gate with 2 input events	6
2.2	An AND gate with 2 input events	7
2.3	A K-out-of-N gate	7
2.4	A simple Static Fault Tree (SFT) example	8
2.5	A BDD of the SFT in Fig. 2.4	10
2.6	A reduction rule of ZBDDs, from [Tang 2004]	11
2.7	A PAND gate and its equivalence with an AND gate, from [Vesely 1981]	13
2.8	An FDEP gate with 2 dependent basic events	13
2.9	A Spare gate with a primary event A and 2 spare events B , and C	14
2.10	2 Spare gates sharing a spare event C	14
2.11	A DFT example from [Coppit 2000]	16
2.12	A portion of the failure automaton of the DFT in Fig. 2.11	17
2.13	The CTMC corresponding to the portion of failure automaton in Fig. 2.12	18
2.14	A Dynamic PFT example from [Bobbio 2004]	19
2.15	A parameterized Warm Spare gate	19
2.16	The SWN of the Warm Spare gate in Fig. 2.15, from [Bobbio 2004]	20
2.17	The correspondence between the variable states, and the failure time intervals	21
2.18	The DTBN of a CSP gate for $n = 2$, from [Boudali 2005a]	22
2.19	The symbology of the 3 temporal gates of Pandora, from [Walker 2009]	23
2.20	A Temporal Truth Table from [Walker 2009]	24
3.1	An example of DFT with one repeated basic event	30
3.2	A non-repairable event	31
3.3	Timing diagrams of Boolean operator OR	32
3.4	Timing diagrams of Boolean operator AND	33
3.5	Timing diagrams of operator non-inclusive BEFORE (BF)	36
3.6	Timing diagrams of operator SIMULTANEOUS (SM)	36
3.7	Timing diagrams of operator Inclusive BEFORE (IBF)	37
4.1	Definition of gate PAND from [Vesely 1981]	44
4.2	An FDEP gate with 2 dependent basic events	45
4.3	A single Spare gate with one primary event A and one spare event B	46
4.4	Two Spare gates sharing a spare event	47
4.5	A single Spare gate with one primary event A and two spare events B and C	48
4.6	A single Spare gate with one primary event P and $(n - 1)$ spare events S_1 to S_{n-1}	49

5.1	A single Spare gate with one primary event A and one spare event B	54
6.1	The Hypothetical Example Computer System (HECS)	70
6.2	The DFT of the HECS	71
6.3	The Hypothetical Cardiac Assist System (HCAS)	71
6.4	The DFT of the HCAS	72
B.1	State transition diagram of a single Spare gate with 2 input events	189
B.2	State transition diagram of 2 Spare gates sharing a spare event	192
B.3	State transition diagram of a single Spare gate with 3 input events	200

List of Tables

2.1	Comparison between the algebraic approach and the approaches reviewed in Section 2.2.3	25
6.1	Failure rates of the basic events of the DFT of the HECS, from [Stamatelatos 2002]	80
6.2	Failure rates of the basic events of the DFT of the HCAS, from [Boudali 2005a]	88

Introduction

Within the field of requirements engineering and system safety analysis, many useful techniques have been developed. Examples of them are Fault Tree Analysis (FTA) [Vesely 1981], Failure Mode and Effects Analysis (FMEA) [Reifer 1979], Hazard and Operability analysis (HAZOP) [Fenelon 1985], and State Machine Hazard Analysis (SMHA) [Leveson 1987]. These methods have different coverage and validity: for instance, FTA is primarily a means for analyzing causes of hazards, whereas FMEA was developed by reliability engineers to allow them to predict equipment reliability, and it is a form of reliability analysis which focuses on successful functions rather than hazards and risks. In this PhD dissertation, we focus on FTA.

FTA was originally developed in 1961 by H.A. Watson at Bell Telephone Laboratories to evaluate the Minuteman Launch Control System for an unauthorized missile launch [Watson 1961]. One of the important handbooks of FTA, the "Fault Tree Handbook", was written by the U.S. Nuclear Regulatory Commission (NRC) in 1981 to serve as a reference text for the system safety and reliability course [Vesely 1981]. FTA is a top-down approach whose inputs consist of the knowledge of the system's functions as well as its failure modes and their effects. The result of the analysis is a set of combinations of component failures which can lead to a specific malfunction. The approach is graphical, constructing Fault Trees (FTs) using standardized Boolean logic symbols.

After that a FT has been constructed, both a *qualitative* and *quantitative* analysis can be performed on the FT. The purpose of qualitative analysis is to determine the minimal cut sets of the FT, which represent the combinations of basic events which will cause the system to fail and which cannot be reduced in number, whereas the quantitative analysis of FTs uses the minimal cut sets to calculate the probability of occurrence of the Top Event (TE), from the probability of occurrence of the basic events [Henley 2000, Shooman 1990]. The quantitative level hence requires the additional knowledge of the time-to-failure probability distributions of all the basic events.

One of the main restrictive assumptions in FTA is that basic events must be assumed to be statistically independent, and their interaction is described by means of Boolean OR/AND gates, so that only the combination of events is relevant, and not their sequence. Besides, events are considered as non-repairable [Vesely 1981]. We refer to this model as *SFT*. An algebraic relation between the TE and the basic events can be determined for any SFT. This algebraic relation is called the *structure function* of the SFT, and it allows to perform both analyses directly.

Several attempts have been reported in the literature to remove the constraints

of FTA, and they include various kinds of temporal and statistical dependencies in the model. A Priority-AND (PAND) gate has been introduced in [Fussell 1976] to model situations in which the failure of the gate occurs if the inputs fail in a pre-assigned order. However, the model that has received the greatest attention is the *Dynamic Fault Tree (DFT)*, proposed by Dugan et al. [Dugan 1992, Dugan 2000]. The DFT is based on the definition of new gates that induce temporal, as well as statistical dependencies (as the distribution of an event may depend on the failure date of another event): Priority-AND (PAND), Functional Dependency (FDEP), Warm Spare (WSP), and Sequence enforcing (SEQ). The structure function of such FTs cannot be determined since the classical Boolean algebra of Boolean variables is not sufficient and since no specific algebraic models of dynamic gates have been defined so far. The quantitative analysis of DFTs hence consists in exploding minimal modules [Dutuit 1996] of dynamic gates into their state-space representation, and computing numerically the related occurrence probability. In any case, the solution of a DFT forces a quantitative analysis, for which the failure distribution of components is most often limited by the models used. A common obstacle in any quantitative technique is the lack of accurate, reliable data on such failure distributions. To overcome this well-known deficiency, the qualitative analysis is often the only valuable information on the system dependability. Nevertheless, the qualitative analysis of DFTs has never been fully considered in the literature, and the concept of minimal cut set needs to be revisited to account for the possible order of the failure events.

In this dissertation, we propose an algebraic framework which allows to determine the structure function of any DFT and hence extend the analytical methods commonly used to analyze SFTs to DFTs. One of the advantages of such analytical approaches is that the quantitative analysis of the FT does not depend on the failure distribution considered for basic events. The temporal dependencies between events that dynamic gates induce impose that events are not repairable, in accordance with [Vesely 1981]. To build an algebraic framework for DFTs, we define events as temporal binary variables; and we introduce, beside Boolean operators OR and AND, temporal operators BEFORE (BF), and SIMULTANEOUS (SM) [Merle 2007a]. We include the possibility that basic events are repeated without restriction. In this dissertation, we show that it is possible to determine an algebraic expression of the structure function of any DFT, and that this structure function can be reduced to a sum-of-product *canonical form* by means of a minimization algorithm which is provided. Each product term of the canonical form contains basic events connected by Boolean and temporal operators, and defines a *Cut Sequence Set (CSS)*, i.e. a set of sequences of (possibly ordered) basic events whose occurrence entails the TE. Finally, we show how to compute the probability of occurrence of the TE from the canonical form, by assigning to basic events any failure time distribution, thanks to a probabilistic model of dynamic gates.

Hence to sum up, the main hypotheses and the new achievements of this dissertation can be condensed into the following points.

- i* - We introduce a new algebraic framework with temporal operators defined on a set of temporal variables.
- ii* - Basic events can be repeated without restriction and are considered as non-repairable.
- iii* - Combining Boolean operators (OR, AND) with temporal operators BEFORE (BF) and SIMULTANEOUS (SM), the algebraic expression of the TE can always be minimized to a sum-of-product canonical form.
- iv* - The canonical form provides a systematic way to generate a list of non-redundant Cut Sequence Sets (CSSs) whose occurrence leads to the TE.
- v* - The probability of occurrence of the TE can be expressed in closed form with any failure distribution.

This dissertation is organised as follows. Chapter 2 recalls the definition of Static and Dynamic Fault Trees and recalls the existing methods which allow to analyze them. The algebraic framework which has been introduced to model dynamic gates is presented in Chapter 3, and the behavioural model of dynamic gates which can be deduced from it is presented in Chapter 4. These behavioural models allow to determine the structure function of DFTs, but a probabilistic model of dynamic gates is necessary to be able to perform the quantitative analysis of DFTs from this structure function. Such behavioural models are introduced in Chapter 5. Chapter 6 shows how the behavioural model of dynamic gates allows to determine the structure function of any DFT under a canonical form, and how both analyses can be performed directly from this structure function thanks to the probabilistic model of dynamic gates presented in Chapter 5. These aspects are illustrated on two DFT examples from the literature.

State of the art

Contents

2.1	Static Fault Trees	6
2.1.1	Definitions	6
2.1.2	Analysis methods	8
2.2	Dynamic Fault Trees	12
2.2.1	Definitions	12
2.2.2	Problem	15
2.2.3	Solutions	15
2.2.4	Contribution of this work	24

In this chapter, we review the different approaches used to perform Fault Tree Analysis. The case of Static Fault Trees and Dynamic Fault Trees is respectively addressed in Sections 2.1 and 2.2.

2.1 Static Fault Trees

In this section, we review most of the existing approaches which allow to perform the analysis of SFT. The definition of SFTs and static gates is recalled in Section 2.1.1, whereas some well-known approaches which allow to analyze SFTs are reviewed in Section 2.1.2.

2.1.1 Definitions

In this section, we recall the different elements which compose a SFT. First, we define what a SFT is in Section 2.1.1.1. Then, we recall the definition of static gates OR, AND, and K-out-of-N in Sections 2.1.1.2, 2.1.1.3, and 2.1.1.4, respectively. Finally, we define the structure function of a SFT in Section 2.1.1.5.

2.1.1.1 Static Fault Tree

According to [Dugan 2000], Static Fault Trees are composed of gates OR, AND, and K-out-of-N, which are combinatorial – or static – gates. The definition of these 3 gates is recalled in the following sections.

2.1.1.2 OR gate

An OR gate with 2 input events A and B is shown in Fig. 2.1. According to [Vesely 1981], the output fault or the OR gate occurs if at least one of the input faults occurs.

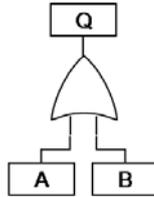


Figure 2.1: An OR gate with 2 input events

2.1.1.3 AND gate

An AND gate with 2 input events A and B is shown in Fig. 2.2. According to [Vesely 1981], the output fault or the AND gate occurs if all of the input faults occur.

2.1.1.4 K-out-of-N gate, or voting gate

A K-out-of-N gate – or voting gate – is shown in Fig. 2.3. The output fault of the voting gate occurs if at least K out of the N input faults occur.

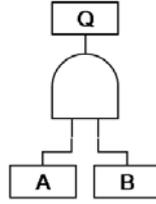


Figure 2.2: An AND gate with 2 input events

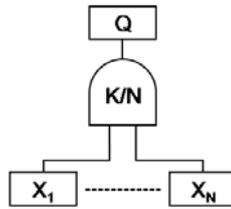


Figure 2.3: A K-out-of-N gate

2.1.1.5 Structure function

The 3 static gates defined in Sections 2.1.1.2 to 2.1.1.4 can easily be algebraically modeled by means of Boolean operators OR and AND. As a consequence, a Boolean model of events is sufficient to capture the failure modes of a SFT. An event A is worth 0 as long as it is functional and 1 as soon as it fails.

The Boolean model of static gates allows to determine an algebraic expression between the TE of the SFT and its basic events. This algebraic expression is called the *structure function* of the SFT, and it can be manipulated – developed and/or simplified – by means of the theorems of Boolean algebra.

Let us consider the FT in Fig. 2.4. This FT is composed of 3 OR gates and 2 AND gate and is consequently static. It can be noted that basic event A is a repeated event since it is the input event of 3 gates of the SFT.

The Boolean model of the top OR gate allows to write that $TE = M + N$, and the Boolean model of both middle gates allows to write that

$$\begin{cases} M = A + P \\ N = Q \cdot D \end{cases}$$

so

$$TE = A + P + Q \cdot D$$

Finally, the Boolean model of both bottom gates allows to write that

$$\begin{cases} P = A \cdot B \\ Q = A + C \end{cases}$$

The structure function of the whole SFT can hence be determined as

$$TE = A + A \cdot B + (A + C) \cdot D,$$

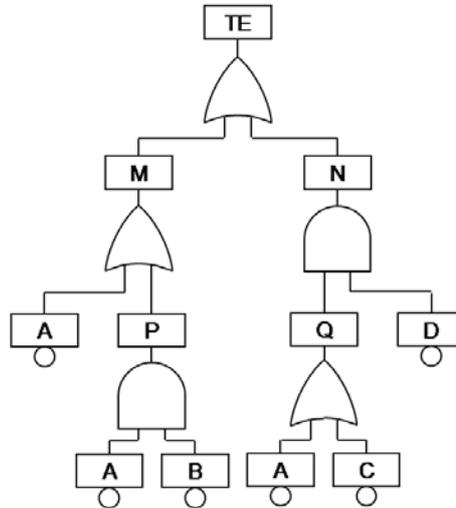


Figure 2.4: A simple SFT example

and this structure function can then be simplified, thanks to the absorption law of Boolean algebra $a + a \cdot b = a$, to

$$TE = A + C \cdot D. \quad (2.1)$$

This structure function can be exploited to perform the qualitative and quantitative analysis of SFTs, as explained in Section 2.1.2.1.

2.1.2 Analysis methods

In this section, we review most of the approaches that are currently used to determine the reliability of static systems. Three approaches dedicated to SFTs are reviewed in Sections 2.1.2.1 to 2.1.2.3, whereas a new form of formal safety analysis, called Deductive Cause-Consequence Analysis (DCCA) and presented in [Ortmeier 2005], is reviewed in Section 2.1.2.4. This latter approach is not based on SFTs.

2.1.2.1 Structure-function-based approach

The structure function of SFTs can be exploited to perform both the qualitative and quantitative analysis of SFTs.

On the one hand, the structure function can be converted into a sum-of-product canonical form, each product term representing a cut set for the SFT. On the other hand, the inclusion–exclusion formula [Trivedi 2001] allows to determine the failure probability of the TE whatever the failure distribution considered for basic events, as long as basic events are statistically independent.

Let us consider the SFT in Fig. 2.4, whose structure function has been determined in (2.1) and is

$$TE = A + C \cdot D$$

On the one hand, each product term of this structure function represents a cut set for the SFT [Rauzy 2001]. As a consequence, A and $C \cdot D$ are cut sets for the SFT in Fig. 2.4. C and D are not cut sets for the SFT, so $C \cdot D$ is minimal, and A is necessarily minimal. The structure function (2.1) thus allows to determine that A and $C \cdot D$ are minimal cut sets for the SFT in Fig. 2.4.

On the other hand, the inclusion–exclusion formula [Trivedi 2001] can allow to determine $Pr\{TE\}(t)$ as follows when basic events are statistically independent:

$$\begin{aligned} Pr\{TE\}(t) &= Pr\{A + C \cdot D\}(t) \\ &= Pr\{A\}(t) + Pr\{C \cdot D\}(t) - Pr\{A \cdot C \cdot D\}(t) \\ &= Pr\{A\}(t) + Pr\{C\}(t) \times Pr\{D\}(t) \\ &\quad - Pr\{A\}(t) \times Pr\{C\}(t) \times Pr\{D\}(t) \\ &= Pr\{A\}(t) + (1 - Pr\{A\}(t)) \times Pr\{C\}(t) \times Pr\{D\}(t) \end{aligned}$$

It can be noted that this expression does not depend on the failure distribution considered for basic events A , C , and D . However, the complexity of the calculation of the failure probability of the TE increases with the size of the FT. Monte-Carlo simulation can hence be quite useful to reduce this complexity. This approach is presented in Section 2.1.2.2.

The Sum of Disjoint Products (SDP) approach also allows to determine the failure probability of the TE of SFTs from their structure function. This approach consists in rewriting the structure function as a sum of disjoint terms by using the theorem of Boolean algebras $A + B = A + \bar{A} \cdot B$. The failure probability of the TE can then be computed by summing the probability of occurrence of all the terms of the structure function. For instance, in our case,

$$TE = A + C \cdot D = A + \bar{A} \cdot C \cdot D,$$

and $Pr\{TE\}(t)$ can hence be determined as

$$\begin{aligned} Pr\{TE\}(t) &= Pr\{A + \bar{A} \cdot C \cdot D\}(t) \\ &= Pr\{A\}(t) + (1 - Pr\{A\}(t)) \times Pr\{C\}(t) \times Pr\{D\}(t) \end{aligned}$$

2.1.2.2 Monte-Carlo simulation

Monte-Carlo simulation is another approach which allows to determine the failure probability of the TE of a SFT [Banks 1984]. Monte-Carlo simulation approaches can be considered as approximation methods from a statistical point of view. Such approaches rely on repeated random sampling to solve problems which are based on calculation. A Monte-Carlo approach thus provides a statistical result under the form of a *confidence interval*, whose size characterizes the accuracy of the approach.

The advantage of Monte-Carlo simulation approaches is that they converge much faster than numerical approaches when the size of the problem increases. The main disadvantage of Monte-Carlo approaches is that for each extra decimal place required, it is necessary to multiply the sample size by 100. Thus, to calculate π to five decimal places by throwing a needle would require about 10^{10} throws, or 1 throw per second for about 300 years.

2.1.2.3 BDD-based approach

An approach similar to the structure-function-based approach and which allows to perform the qualitative and quantitative analysis of SFTs consists in converting the SFT into a Binary Decision Diagram (BDD), the analysis being performed by means of some algorithms [Rauzy 1997].

Let us consider the SFT in Fig. 2.4. The BDD associated with the structure function in (2.1) can be computed by means of the function build that can be described through recursive equations to be applied on the structure function and its subparts. It can be noted that the size of the BDD encoding the structure function strongly depends on the chosen variable ordering. For instance, the BDD of the SFT in Fig. 2.4 with the variable ordering A, B, C, D is shown in Fig. 2.5.

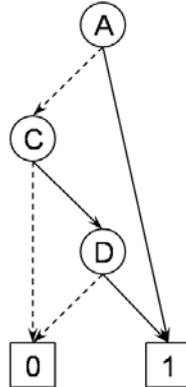


Figure 2.5: A BDD of the SFT in Fig. 2.4

On the one hand, the *exact* probability of the TE of the SFT in Fig. 2.4 can be computed given the probabilities of its basic events. This quantitative analysis is performed by means of a BDD traversal, the Shannon decomposition being applied on each node of the BDD. On the other hand, some algorithms provided in [Rauzy 1997] allow to determine the minimal cut sets of the SFT in Fig. 2.4 from the BDD in Fig. 2.5.

Another class of BDD, proposed in [Minato 1993, Minato 2001] and exploited in [Tang 2004], can also be used. Such a BDD, denoted as Zero-suppressed BDD (ZBDD), is obtained by deleting all the nodes whose 1-edge points to the 0-terminal node and by connecting the edge to the other subgraph directly, as shown in Fig.

2.6, and by sharing all isomorphic subgraphs. This data structure represents sets of combinations more efficiently than using originals ordered BDDs.

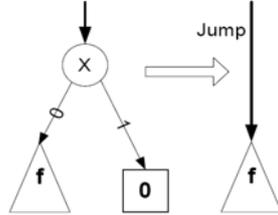


Figure 2.6: A reduction rule of ZBDDs, from [Tang 2004]

BDDs and ZBDDs hence represent efficient tools to perform the analysis of large SFTs.

2.1.2.4 Deductive Cause-Consequence Analysis (DCCA)

Deductive Cause-Consequence Analysis (DCCA) [Ortmeier 2005] is a recent formal safety analysis approach which is not only based on FTs. It is presented as a generalization of Failure Modes and Effects Analysis (FMEA) [McDermott 1996], Failure Modes, Effects, and Criticality Analysis (FMECA) [ECSS 2001], and Fault Tree Analysis [Stamatelatos 2002].

The formalization is done with Computational Tree Logic (CTL) [Emerson 1990], and the system models used are finite automata, thus allowing the use of model checkers like SMV [McMillan 1990]. A list of hazards on system level as well as a list of possible basic component failure modes is supposed to be given, and both are assumed to be described by predicate logic formula. The authors define a temporal logic property, called *criticality* of a set of failure modes, which indicates whether a given combination of failures may lead to the hazard or not. This combination of failures is called a *minimal critical set* if it is critical with regards to the criticality property, and if no proper subset of this combination is critical. The goal of DCCA is not to test all sets of failure modes to determine all the minimal critical sets, since it would require an effort which would be exponential in the number of failure modes, but to formally verify the results of informal safety analysis techniques. It can be noted that the results of DCCA have the same semantics as those of formal FTA [Schellhorn 2002].

The proof obligations of DCCA may be constructed automatically and the proofs can be done – for finite state systems – by model checking. Besides, the authors of [Ortmeier 2005] claim that DCCA formalization is strictly more precise than other formal safety analysis techniques like formal FTA. However, even though DCCA could be used to determine all the minimal critical sets of a system and hence perform its qualitative analysis, it does not address the problem of the determination of the failure probability of the system.

2.2 Dynamic Fault Trees

In this section, we review most of the existing approaches which allow to perform the analysis of Dynamic Fault Trees (DFT). The definition of DFTs and dynamic gates is recalled in Section 2.2.1. Contrary to SFTs, the structure function of DFTs cannot be determined, as explained in Section 2.2.2, so other approaches must be used to analyze DFTs. We review these approaches in Section 2.2.3. In Section 2.2.4, we present the contribution of this Ph.D. thesis. It consists in defining an algebraic framework to model dynamic gates and determine the structure function of DFTs to extend the analytical approaches commonly used to analyze SFTs to DFTs.

2.2.1 Definitions

In this section, we recall the different elements which compose a DFT. First, we define what a DFT is in Section 2.2.1.1. Then, we recall the definition of dynamic gates PAND, FDEP, and Spare respectively in Sections 2.2.1.2, 2.2.1.3, and 2.2.1.4.

2.2.1.1 Dynamic Fault Tree

According to [Dugan 2000], DFTs comprise basic events, static gates (OR, AND, and K-out-of-N), and dynamic gates (PAND, FDEP, WSP, and SEQ). However, it was shown in [Boudali 2007a] that the Sequence Enforcing (SEQ) gate is expressible in terms of the cold spare gate. As a consequence, dynamic gates can be limited to gates PAND, FDEP, and Spare, only.

The definition of these 3 gates is recalled in the following sections.

2.2.1.2 PAND gate

The Priority-AND (PAND) gate was defined in [Fussell 1976] as logically equivalent to an AND gate where the input events must occur in a specific order. Conventionally, the events attached to a PAND gate must occur in the order that they appear from left to right in the graphical representation. As a consequence, in the case of a PAND gate with 2 input events as shown in Fig. 2.7, A and B must occur and A must occur before B for Q to occur.

This case, which is the most common one in the literature, can thus be described by the equivalence shown in Fig. 2.7, from [Vesely 1981].

2.2.1.3 FDEP gate

The Functional Dependency (FDEP) gate was defined in [Dugan 1990]. This gate has:

- 1 trigger event (either a basic event or the output of another gate in the tree);
- a non-dependent output (reflecting the status of the trigger event);

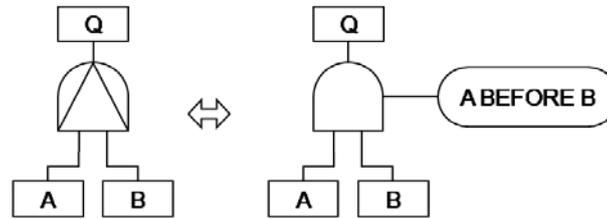


Figure 2.7: A PAND gate and its equivalence with an AND gate, from [Vesely 1981]

- one or more dependent basic events.

The dependent basic events are functionally dependent on the trigger event. When the trigger event occurs, the dependent basic events are forced to occur. The separate occurrence of any of the dependent basic events has no effect on the trigger event.

An FDEP gate with 2 dependent basic events A and B is shown in Fig. 2.8. Event T represents the trigger event, and the output of the gate is not represented.

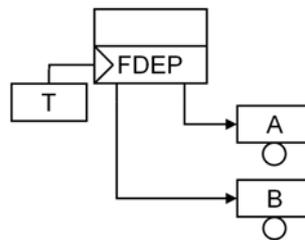


Figure 2.8: An FDEP gate with 2 dependent basic events

2.2.1.4 Spare gate

The Spare gate was defined in [Dugan 1990]. This gate has one primary input and one or more alternate inputs. All inputs are basic events. The primary input is the one that is initially powered on (or active), and the alternate input(s) specify the – initially unpowered or dormant – components that are used as replacements for the primary unit. It has one output which becomes true after all the input events occur.

A Spare gate with a primary event A and 2 spare events B , and C , is shown in Fig. 2.9.

According to [Stamatelatos 2002], which presents a Markovian definition of the Spare gates, associated with each input to the Spare gate is a dormancy factor – usually between zero and one – that multiplies the failure rate while the unit is spare. When the dormancy factor is equal to zero, the spare event is called a Cold

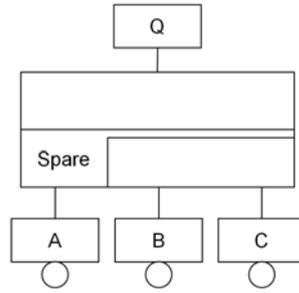


Figure 2.9: A Spare gate with a primary event A and 2 spare events B , and C

Spare (CSP), and it cannot fail while in its dormant mode. When the dormancy factor is equal to one, the spare event is called a Hot Spare (HSP), and its failure rate is the same in its dormant and active modes. When the dormancy factor is between zero and one, the spare event is called a Warm Spare (WSP). Cold spares and hot spares can hence be considered as particular cases of warm spare events.

Spare gates can also be used if spare units are shared. Then the basic event representing the spare unit has inputs to more than one Spare gate. The spare is available only to one of the Spare gates, depending on which of the primary units fails first. An example of 2 Spare gates sharing a spare event C is shown in Fig. 2.10.

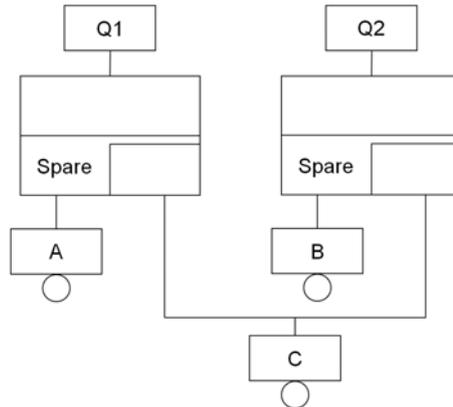


Figure 2.10: 2 Spare gates sharing a spare event C

If A fails first, the spare event C is made unavailable: $Q1$ occurs as soon as C occurs, and $Q2$ occurs as soon as B occurs. On the contrary, if B fails first, $Q1$ occurs as soon as A occurs, and $Q2$ occurs as soon as C occurs.

2.2.2 Problem

When systems exhibit a static behaviour (their failure is engendered by mere combinations of component failures), the SFT model allows to model the failure of the system and the algebraic model of static gates, based on Boolean operators OR and AND, allows to determine the structure function of the SFT. This structure function can then be exploited to perform both the qualitative and quantitative analysis of the SFT, as explained in Section 2.1.2.1.

However, when systems exhibit a dynamic behaviour (their failure is not engendered by combinations, but specific sequences of component failures), the SFT model is not sufficient to model the failure of the system and DFTs must be used. The structure function of such DFTs cannot be determined since no algebraic model has been defined for dynamic gates so far. As a consequence, other approaches must be used to perform the analysis of the DFT without using its structure function. The most significant approaches are reviewed in Section 2.2.3.

2.2.3 Solutions

In this section, we review most of the approaches that are currently used to determine the reliability of dynamic systems. We focus more particularly on DFTs. These approaches are reviewed in Section 2.2.3.1, and we focus on the most common ones in Sections 2.2.3.2 to 2.2.3.5.

2.2.3.1 Global view

Many approaches are dedicated to the determination of the failure causes of dynamic systems. Many of them define alternative Fault Trees, such as the approach presented in [Bouissou 2003], in which a new semantics is assigned to the traditional graphical representation of fault trees by means of a new kind of links called triggers which allow to combine conventional fault trees and Markov models. This new formalism is called "Boolean Logic Driven Markov Process" (BDMP) and allows to determine the reliability of dynamic systems. Another model, called time-to-failure tree, is presented in [Ejlali 2003]. SFTs and DFTs can be converted into time-to-failure trees and synthesized to a Field Programmable Gate Array (FPGA) to significantly accelerate Monte-Carlo simulation. Finally, the authors of [Clarhaut 2009] propose a multi-fault tree model which is based on the definition of 2 temporal operators, PAND and SEQ, which allow to model sequences of multiple failures. Each failure is assigned a relative reliability coefficient, and the knowledge of the number of scenarios leading to the considered failure allows to evaluate the dependability level of all possible equipment architectures of the system.

Most approaches are based on DFTs. Some of them are based on the conversion of the DFT into a state model. This conversion can be global, as for the Temporal Bayesian Network model presented in Section 2.2.3.4, or it can use the divide-and-conquer approach by converting the dynamic parts of the DFT into Continuous Time Markov Chains or Stochastic Petri Nets, as respectively presented in Sections 2.2.3.2

and 2.2.3.3, whereas the static parts of the DFT are solved by using BDD-based or other combinatorial methods. A novel analytic approach which proposes extensions to the logical foundations of FTs that enable use of temporal gates is presented in Section 2.2.3.5. Finally, some approaches allow the direct determination of the reliability of a DFT by means of analytic methods [Amari 2003] or Monte-Carlo simulation [Rao 2009].

2.2.3.2 Continuous Time Markov Chains (CTMC)

DFTs can be solved by automatic conversion to Continuous Time Markov Chains [Dugan 1992, Dugan 1993]. As illustrated in [Coppit 2000] on the DFT example shown in Fig. 2.11, the DFT studied is first converted into an intermediate representation which is called *failure automaton*. A failure automaton is a state machine which models the changing state of the system as failures occur [Coppit 2003]. A portion of the failure automaton of the DFT in Fig. 2.11 is shown in Fig. 2.12.

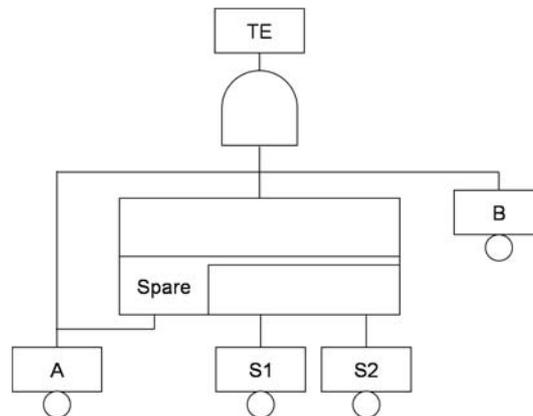


Figure 2.11: A DFT example from [Coppit 2000]

Each state of the failure automaton represents a Fault Tree state, an arc between two states indicating the basic event whose occurrence caused the state transition. The failure automaton obtained can then be converted into a Continuous Time Markov Chain (CTMC). Each state of the CTMC corresponds to a state of the failure automaton, and transitions also correspond one to one. The transition rates between states of the CTMC correspond to the rate of occurrence of the triggering basic events. The Markov chain obtained from the conversion of the portion of failure automaton in Fig. 2.12 is shown in Fig. 2.13.

The approach used in the Galileo tool [Dugan 2000] is modular [Gulati 1997] and based on CTMCs. During the traversal of the DFT, a subtree is marked as dynamic if a dynamic gate is present. If a subtree contains no dynamic gates, it is classified as static. After the traversal is completed, static subtrees are solved using BDD-based methods. The Markov method is used for dynamic subtrees. Each submodel is solved for the probabilities of failure, and is replaced by a basic event in

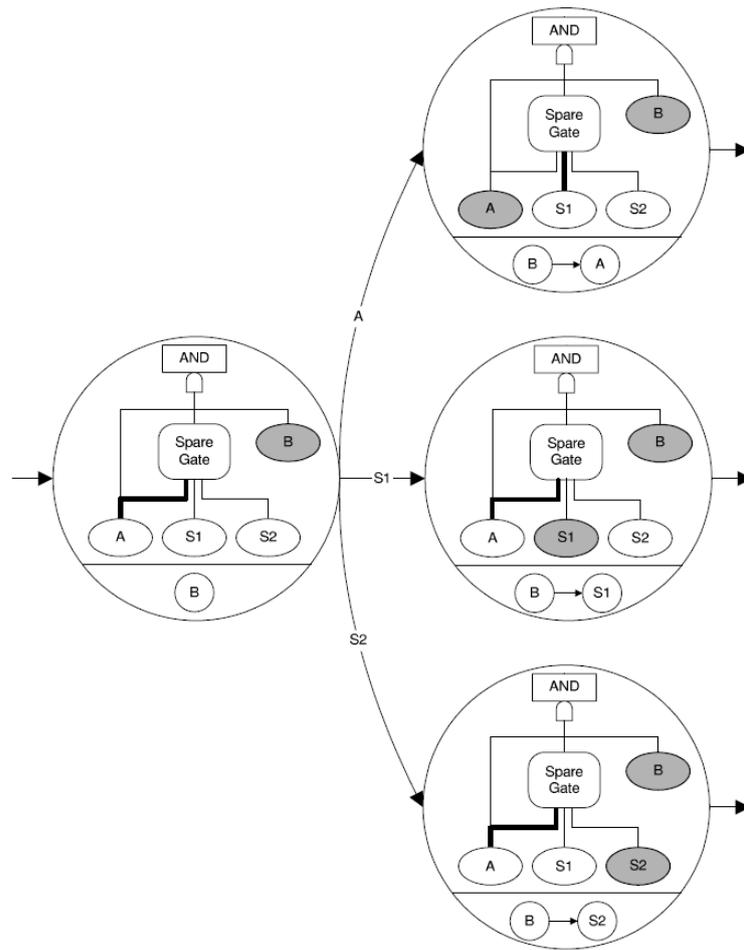


Figure 2.12: A portion of the failure automaton of the DFT in Fig. 2.11

the higher-level mode which is characterized by a failure probability. The reduced, top-level fault tree is then solved as a static tree with as many basic events as there are subtrees in the DFT.

Regarding the qualitative analysis of DFTs, the CTMC which corresponds to a DFT provides the cut sequences of the DFT by providing the sequences of basic event failures which lead to the state of the CTMC in which the *TE* of the DFT occurs, but the determination of the minimal cut sequences requires further investigation and is not direct. In the Galileo tool, the qualitative analysis is not based on the CTMC, but it is performed by means of Zero-suppressed BDDs (ZBDD) [Tang 2004].

To conclude, the Continuous Time Markov Chain approach is a modular approach which is dedicated to the dynamic subtrees of DFTs only, the static subtrees being solved by means of faster BDD-based methods, and which allows to perform the quantitative analysis of DFTs. However, the CTMC does not provide the minimal cut sequences of the DFT which are required for the qualitative analysis, and

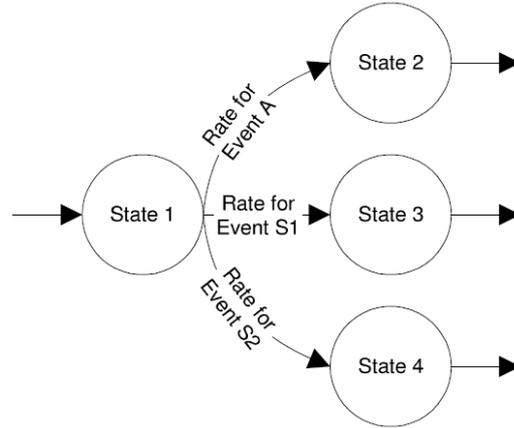


Figure 2.13: The CTMC corresponding to the portion of failure automaton in Fig. 2.12

the use of CTMCs forces the failure distribution function of basic events to be exponential [Dugan 2000]. Besides, the size of the CTMC increases with the number of basic events of the DFT and can lead to state space explosion.

2.2.3.3 Stochastic Petri Nets (SPN)

Another approach is presented in [Bobbio 2004] which proposes to include statistical dependencies in Fault Trees. The authors adopt a parameterization technique, referred to as Parametric FT (PFT), to fold equal subtrees or events to resort to a more compact FT representation. For instance, a Dynamic PFT example is shown in Fig. 2.14. The basic events $A(i)$, $B(j)$, $D(k)$, and $SP(h)$ in Fig. 2.14 represent replicated basic events whereas the intermediate event $SUB(i)$ represents a replicated module. In order to model cyclic behaviours, a new primitive called repair box is also introduced. A repair box, attached to an event, causes the starting of a repair activity of all the components that failed as the event occurs.

Parametric FTs can then be modularized starting from an algorithm presented in [Dutuit 1996]. A module may be classified as static or dynamic. Static modules contain common basic events and can be analyzed by means of suitable combinatorial techniques. Dynamic modules contain dynamic gates or repair boxes and require a state-space analysis which is obtained by translating the dynamic module into a high level colored Petri net in the form of a Stochastic Well-formed Net (SWN) [Chiola 1993]. This translation is obtained by translating each dynamic gate into a SWN, the translation rules being defined in [Bobbio 2004]. For instance, the SWN of the parameterized Warm Spare gate shown in Fig. 2.15 is shown in Fig. 2.16.

In Fig. 2.16, place SP_na contains the colored tokens of the spares which are not available because failed or already working; SP_curr contains the token relative to the spare which is currently replacing the main component. Transition SP_fail

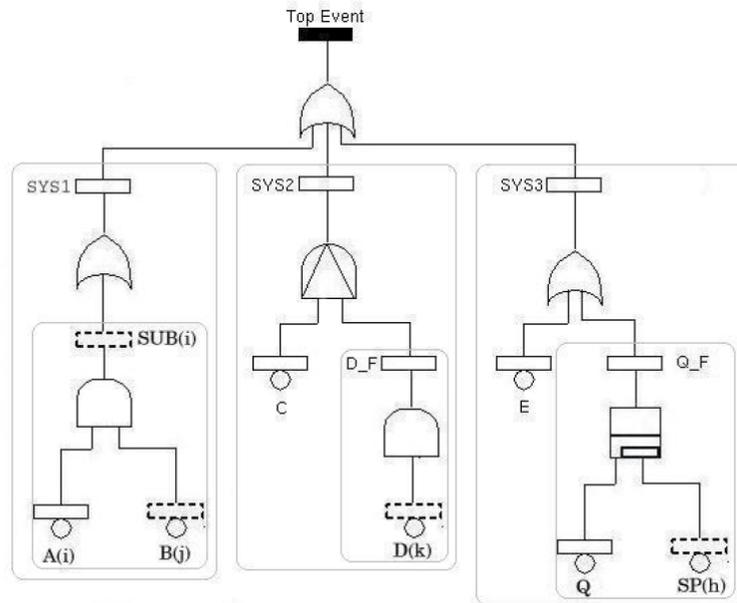


Figure 2.14: A Dynamic PFT example from [Bobbio 2004]

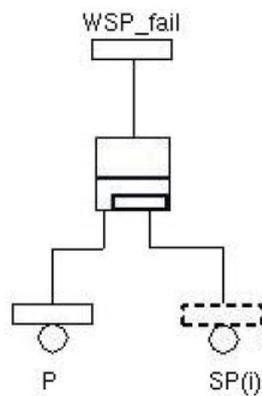


Figure 2.15: A parameterized Warm Spare gate

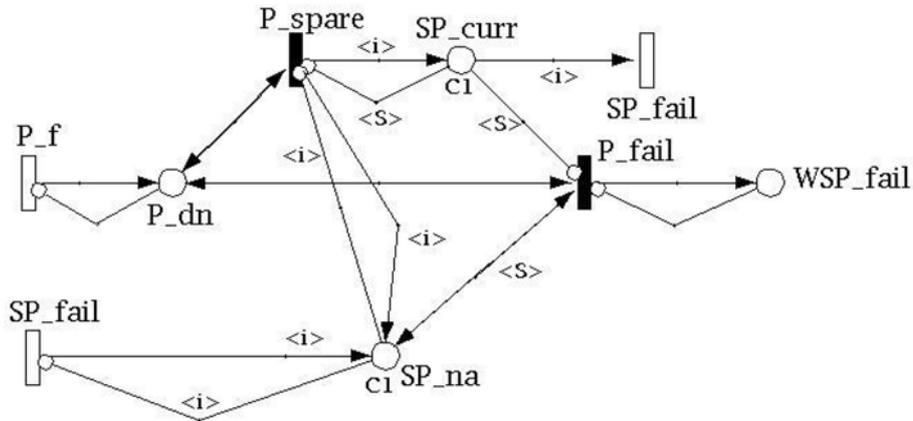


Figure 2.16: The SWN of the Warm Spare gate in Fig. 2.15, from [Bobbio 2004]

models the fault of a spare when in dormant condition, putting the relative token in SP_na. When the main component P fails (token in place P_dn), transition P_spare fires, putting the token relative to the spare to be used in SP_curr and SP_na. If later the spare fails (firing of transition SP_fail), if place SP_na contains a number of tokens equal to the number of spares, transition P_fail fires, modeling the general failure of the gate, else another spare starts working by means of the P_spare transition.

Once each module has been converted into a Petri net in the form of a SWN, it is analyzed in isolation by resorting to the underlying lumped CTMC [Delamare 2003]. The module failure probability, computed from the CTMC, is cast back into the original PFT by replacing the whole module with a single basic event. This approach is the one implemented in the Linux version of the Drawnet tool [Vittorini 2002].

A similar approach is presented in [Raiteri 2005], which consists in converting the whole DFT into a Generalized Stochastic Petri Net (GSPN). A CTMC can be generated from this GSPN [Chiola 1995], and the unreliability of the system is computed on the CTMC.

To conclude, the Stochastic Petri Net approach can be used both as a modular and a global approach which allows to perform the quantitative analysis of DFTs. However, the two approaches which exploit SPNs [Bobbio 2004, Raiteri 2005] do not exploit it to perform the qualitative analysis of DFTs, and the quantitative analysis is based on the conversion of SWNs or GSPNs into a CTMC, so the failure distribution function of basic events is necessarily exponential.

2.2.3.4 Temporal Bayesian Networks (TBN)

Temporal Bayesian Networks can also allow to perform the quantitative analysis of DFTs. A complete definition of Bayesian Networks (BN) can be found in [Jensen 1996].

A BN is a directed acyclic graph comprised of nodes and arcs. Nodes represent

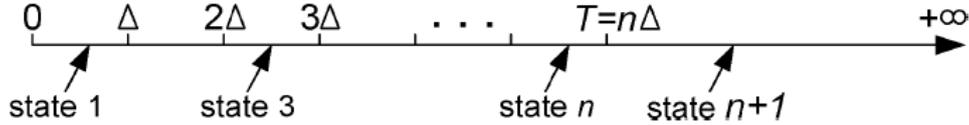


Figure 2.17: The correspondence between the variable states, and the failure time intervals

Random Variables (RV) and directed arcs between pairs of nodes represent dependencies between the RVs. A BN uniquely defines a joint probability distribution over all the RVs which are present in the graph. A marginal prior probability table is associated with each root node of the BN, and a Conditional Probability Table (CPT) is associated with all other nodes, the CPT of a RV specifying the probability of each of the variable's states conditioned on the value of each of its parent nodes. The joint probability distribution can then be determined by using the Chain Rule and assuming the conditional independence between the variables [Boudali 2005a].

A Discrete-Time BN (DTBN) framework is introduced in [Boudali 2005a]. A DFT can be translated into an equivalent DTBN: each basic event of the DFT is represented as a root node in the DTBN, and all gates in the DFT are intermediate nodes in the DTBN and possess a CPT associated with each one of them. The time line is divided into $n + 1$ intervals, and each node variable has a finite number $n + 1$ of states, as shown in Fig. 2.17. The n first states divide the time interval $(0, T]$, where T is the mission time, into n intervals whose length is $\Delta = \frac{T}{n}$, whereas the last state $n + 1$ represents the time interval $(T, +\infty)$. If a RV is in state $n + 1$, it hence means that the corresponding basic component or gate did not fail during the mission time.

The marginal prior probability tables and the CPTs can then be populated according to the failure distributions of basic events and to the gates of the DFT, and the probability of the leaf node of being in the interval $(T, +\infty)$ can then be obtained to determine the reliability of the system. The DTBN of a CSP gate is shown in Fig. 2.18 for $n = 2$.

If we look at the marginal prior probability table of A in Fig. 2.18, $P1$, $P2$, and $P3$ are the probabilities of A failing in the intervals $(0, \Delta]$, $(\Delta, T]$, and $(T, +\infty)$, respectively. The method to determine $P1$ and $P2$ from the failure distribution of A is detailed in [Boudali 2005a] and $P3$ satisfies $P1 + P2 + P3 = 1$. Regarding the CPT of B , it indicates that B can fail with a probability of $P12$ if A has failed in the time interval $(0, \Delta]$, and that B will not fail during the mission time if A fails during the time interval $(\Delta, T]$ or does not fail during the mission time. Finally, the CPT of the CSP gate simply indicates that the CSP gate fails as soon as B fails – in the same time interval.

This framework was extended to the case of Continuous-Time BN (CTBN) in [Boudali 2006]. Instead of being discrete, the RVs of the BN are continuous, and the probabilities are conditional and are expressed in terms of probability density

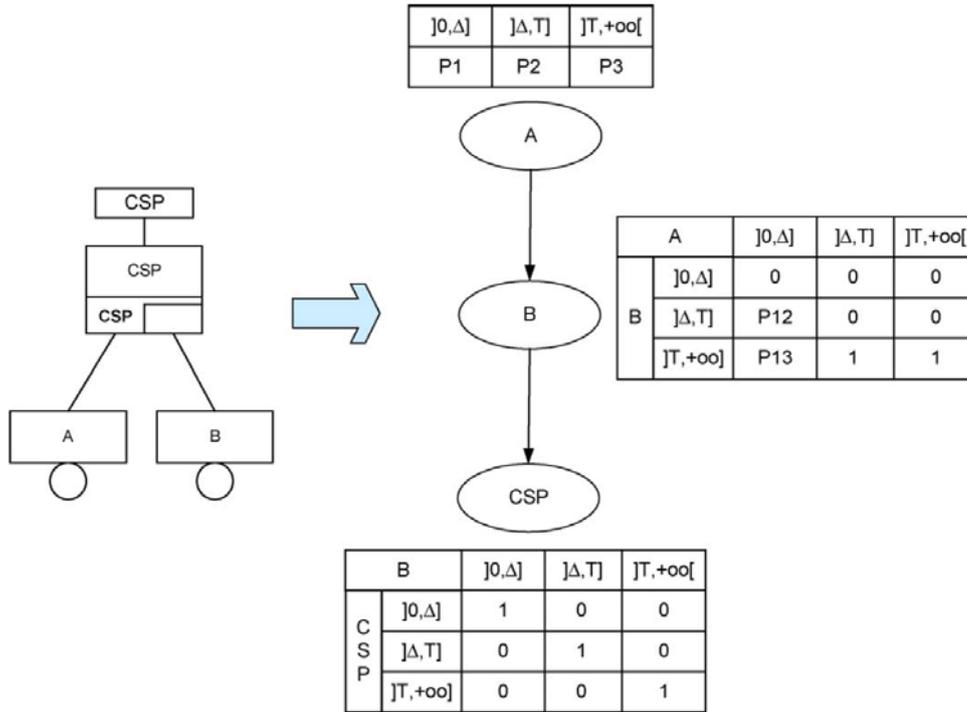


Figure 2.18: The DTBN of a CSP gate for $n = 2$, from [Boudali 2005a]

functions (PDF). Two specific functions are defined to capture the behaviour and dependencies found in complex systems, which are the unit-step function and the impulse function. The closed-form analytical solution for the failure distribution of the system can then be computed from the marginal PDF of basic events.

The Windows version of the Drawnet tool [Montani 2005] is based on Dynamic BNs which are quite similar to CTBNs [Bobbio 2001].

Both the DTBN and the CTBN framework are dedicated only to the quantitative analysis of DFTs. On the one hand, DTBNs can be solved by using a standard BN inference engine, but the solution provided is approximate, dependent on the time granularity [Boudali 2005b], and highly memory consuming [Boudali 2006]. On the other hand, the CTBN framework provides the closed-form solution for the reliability of the system and allows memory savings since there are no more CPTs to store; however, there is not a theory for exact BN inference in CTBNs with general distributions and a theory exists only in the case of Gaussian distributions [Lauritzen 2001] and mixtures of truncated exponentials [Moral 2001].

2.2.3.5 Qualitative temporal analysis (QTA)

In recent years, a new extension to FTA was proposed to enable FTs to model event sequences and relative temporal ordering. This extension is called Pandora [Walker 2006] and is built around a redefinition of the PAND gate. The definitions



Figure 2.19: The symbology of the 3 temporal gates of Pandora, from [Walker 2009]

used in Pandora are intended to remove the ambiguities relative to the definition of the PAND gate in [Stamatelatos 2002] and provide a framework for the qualitative analysis of PAND gates and of the other temporal gates defined. This framework is presented in [Walker 2009].

Events are considered as non-repairable, and 3 temporal relations, named *Before*, *Simultaneous*, and *After*, are defined to traduce the possible temporal relations between two events X and Y . Pandora defines the PAND gate to represent temporal relations Before and After, and introduces a new gate, the Simultaneous-AND (SAND) gate, to represent the temporal relation Simultaneous, so that gates PAND and SAND allow to represent all the temporal relations possible between two events in Pandora. Besides, given gates PAND and SAND represent conjunctions of events only, the authors of [Walker 2009] define a new gate, the Priority-OR gate (POR), in [Walker 2007b] to model disjunctions of events. The definition of the POR gate is based on the definition which can be found in [Vesely 1981] of the Exclusive-OR (XOR) gate, which was quite conflictual with the traditional definition of the XOR gate and was corrected in the later version of the Fault Tree Handbook [Stamatelatos 2002]. The symbology of the 3 temporal gates defined in Pandora is shown in Fig. 2.19. It can be noted that the authors retain for the POR gate the same symbol as the one of the XOR gate in [Stamatelatos 2002].

Pandora is concerned only with the order in which events occur, and this order is modeled by means of a property called the *sequence value*. The sequence value of an event X is noted $S(X)$. Given a set of events, $S(X) = 1$ indicates that the event occurs first in the set, $S(X) = 2$ indicates that it occurs second, and so forth. If $S(X) = 0$, then it means that the event has not occurred. These sequence values can then be used to construct *Temporal Truth Tables (TTT)* to demonstrate equivalences between expressions and hence simplification and development theorems for the FTs which contain the 3 temporal gates of Pandora. An example of TTT is shown in Fig. 2.20, where mathematical symbols $<$, $\&$, and $|$ respectively model gates PAND, SAND, and POR.

The 3 first columns in Fig. 2.20 enumerate all the possible failure sequences for events X , Y , and Z , and it can be noted that simultaneous failures are also considered and indicated by the same sequence value for events. The other columns of the TTT indicate the sequence values of different expressions in the different cases considered, and two expressions are considered as equivalent if the sequence values of both expressions are the same in all the cases considered, as it is the case for the

X	Y	Z	X<Y	Y<Z	X<Z	Y<X	X&Y	X Y	X<Y.Y<X	X<Y.Y<Z	X<Y.Y<Z.X<Z	X Y.Y	X<Y+X Y
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	1	2	0	2	0	0	0	0	0	0	0	0	0
0	2	1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0	1
1	0	1	0	0	0	0	0	1	0	0	0	0	1
1	0	2	0	0	2	0	0	1	0	0	0	0	1
2	0	1	0	0	0	0	0	2	0	0	0	0	2
1	1	0	0	0	0	0	1	0	0	0	0	0	0
1	2	0	2	0	0	0	0	1	0	0	0	2	1
2	1	0	0	0	0	2	0	0	0	0	0	0	0
1	1	1	0	0	0	0	1	0	0	0	0	0	0
1	1	2	0	2	2	0	1	0	0	0	0	0	0
1	2	1	2	0	0	0	0	1	0	0	0	2	1
2	1	1	0	0	0	2	0	0	0	0	0	0	0
1	2	2	2	0	2	0	0	1	0	0	0	2	1
2	1	2	0	2	0	2	0	0	0	0	0	0	0
2	2	1	0	0	0	0	2	0	0	0	0	0	0
1	2	3	2	3	3	0	0	1	0	3	3	2	1
1	3	2	3	0	2	0	0	1	0	0	0	3	1
2	1	3	0	3	3	2	0	0	0	0	0	0	0
2	3	1	3	0	0	0	0	2	0	0	0	3	2
3	1	2	0	2	0	3	0	0	0	0	0	0	0
3	2	1	0	0	0	3	0	0	0	0	0	0	0

Figure 2.20: A Temporal Truth Table from [Walker 2009]

two expressions $X < Y.Y < Z$ and $X < Y.Y < Z.X < Z$.

The qualitative temporal analysis of a FT can be performed after transforming the expression for the structure function of the FT into a simplified form called *base temporal form (BTF)* [Walker 2007a] and which consists of *doublets*. An expression in BTF contains only AND gates, OR gates, and doublets. A doublet is simply a pair of events connected by a single temporal gate, and is indicated by square brackets, e.g. $[X < Y]$. This expression in BTF can then be reduced and the minimal cut sequences of the FT can be determined from it.

Pandora allows to perform the qualitative analysis of FTs with temporal gates PAND, SAND, and POR. However, only one of these three gates is a dynamic gate as defined in [Dugan 1992]. Besides, Pandora does not allow to determine the failure probability of the *TE* of the FTs which it considers, and does hence not allow to perform the quantitative analysis of DFTs.

2.2.4 Contribution of this work

The approach that we propose aims at determining the structure function of DFTs to be able to extend the analytical techniques commonly used to analyze SFTs to DFTs. This algebraic framework is based on a temporal definition of events which allows to define three temporal operators named Non-inclusive Before (BF), Inclusive BEFORE (IBF), and Simultaneous (SM).

The BF operator was introduced in [Merle 2007a, Merle 2007b] to model dynamic gates PAND, FDEP [Merle 2010b], and Spare [Merle 2010c], and the SM operator was introduced in [Merle 2009] to allow to take into account the simultaneity of events. This behavioural model of dynamic gates allows to determine the structure function of any DFT, and a complete list of theorems which is provided allows to develop and simplify this structure function to a sum-of-product canonical

form.

A minimization criterion was introduced in [Merle 2010b] to reduce this canonical form of the structure function to a *minimal canonical form* by removing the redundant terms which may be contained in the canonical form. The determination of this minimal canonical form of the structure function for any DFT makes possible the direct determination of the cut sequences of the DFT. Besides, a probabilistic model of dynamic gates allows to perform the quantitative analysis of DFTs directly from this minimal canonical form of their structure function.

It can be noted that the probabilistic models which are provided allow to determine the failure probability of the *TE* of any DFT for any failure distribution considered for basic events, as it is the case for SFTs, since the reliability formulas that we use do not depend on the distribution considered either. If the considered failure distribution is not analytically integrable (as for the Weibull distribution), the probabilistic relation deduced from the minimal canonical form of the structure function can still be used by resorting to numerical integration.

A comparison between this algebraic approach and the approaches reviewed in Section 2.2.3 can be found in Table 2.1 (NA stands for Not Available).

Table 2.1: Comparison between the algebraic approach and the approaches reviewed in Section 2.2.3

Approach	Qualitative analysis	Quantitative analysis	Implementation	Modularity
CTMC	NA	Exponential distributions	Galileo [Dugan 2000]	Y
SPN	NA	Exponential distributions	Drawnet [Vittorini 2002]	Y/N
TBN	NA	Gaussian distributions, truncated exponentials	Drawnet [Montani 2005]	N
QTA	Minimal cut sequences	NA	Planned in HiP-HoPS [Papadopoulos 2001]	Y
Algebraic approach	Cut sequences	Any distribution	Not yet	Y

This algebraic approach allows to determine the cut sequences of any DFT, and the failure probability of the *TE* of any DFT can be computed for any failure distribution considered for basic events. Besides, it is a modular approach since the static and dynamic parts of the subtree are equivalent to static and dynamic parts in the structure function of the DFT. When a part of the structure function is static, the minimal cut sets can be determined directly, and its failure probability can be determined by means of the inclusion–exclusion formula; when a part of the

structure function is dynamic, the cut sequences can be determined directly, and its failure probability can be determined by means of the probability models provided.

Algebraic framework

Contents

3.1 Hypotheses	29
3.1.1 Non-repairable events	29
3.1.2 Continuous failure time distributions	29
3.1.3 Statistical independence	30
3.1.4 Simultaneity	30
3.2 Temporal model of non-repairable events	31
3.3 Temporal model of Boolean operators	32
3.3.1 Operator OR	32
3.3.2 Operator AND	33
3.3.3 Algebraic structure of $(\mathcal{E}_{nr}, +, \cdot)$	33
3.4 Temporal operators	35
3.4.1 Operator non-inclusive BEFORE	35
3.4.2 Operator SIMULTANEOUS	36
3.4.3 Operator Inclusive BEFORE	37
3.5 Theorems	37
3.5.1 Theorems satisfied by operator non-inclusive BEFORE	38
3.5.2 Theorems satisfied by operator SIMULTANEOUS	39
3.5.3 Theorems satisfied by operator Inclusive BEFORE	40
3.5.4 Simplification theorems	41
3.6 Discussion on the similarities and differences between our approach and Pandora	41

This chapter presents the algebraic framework which has been introduced to model dynamic gates. The hypotheses that have been made when designing this algebraic framework are presented in Section 3.1, and these hypotheses allow to define a temporal model of events in Section 3.2. DFTs may contain static gates, so the algebraic framework presented must be able to model static gates as well. A temporal model of Boolean operators OR and AND is hence provided in Section 3.3. The 3 temporal operators which have been defined to model dynamic gates are presented in Section 3.4. These temporal operators will allow to determine the structure function of DFTs, and a list of theorems is provided in Section 3.5 to manipulate expressions containing temporal operators, and hence the structure function of DFTs. Finally,

Section 3.6 discusses the semantics similarities and differences between our algebraic framework and Pandora, which was presented in Section 2.2.3.5 and is the closest approach to ours.

3.1 Hypotheses

3.1.1 Non-repairable events

The Fault Tree Handbook does not define events with time in mind. It only mentions that a basic event is "a basic initiating fault event that requires no further development" [Vesely 1981]. However, it has already been stated that events represent occurrences of faults rather than the existence of faults, and the Fault Tree Handbook further states that "under conditions of no repair, a fault that occurs will continue to exist". These statements are retained in this algebraic framework, and the events of FTs are considered as non-repairable. It is also assumed that events occur instantaneously. Therefore:

- an event in this algebraic framework represents the occurrence of a fault;
- the event is false as long as the fault has not yet occurred;
- when the fault occurs, the event becomes true and will remain true thereafter;
- an event is considered as non-repairable: it cannot go from true to false, only from false to true;
- an event is instantaneous: it goes instantly from false to true.

3.1.2 Continuous failure time distributions

According to [Stamatelatos 2002], the input data that must be supplied for a basic event is usually one of four basic types:

1. a *component failure probability* in some time interval;
2. an *event occurrence probability* in some time interval;
3. a *component unavailability* and
4. a *pure event probability*.

As an event in our algebraic framework represents the occurrence of a fault, and hence the failure of a component, points (1) and (2) are equivalent, as well as points (3) and (4). Besides, the input data must necessarily be time-sensitive since the basic events that we consider may be the input events of dynamic gates. As a consequence, we assume that each basic event is defined by an event occurrence probability in the set of positive times, and that this failure time distribution is continuous, so that two basic events cannot occur simultaneously.

3.1.3 Statistical independence

In the general case of 2 dependent basic events A and B , both events fail with a probability

$$Pr\{A \cdot B\} = Pr\{A\} \times Pr\{B|A\} = Pr\{B\} \times Pr\{A|B\},$$

according to [Grimaldi 2004], where $Pr\{B|A\}$ is the conditional probability of B given A , which means that the occurrence of A may affect the occurrence of B and vice versa.

We consider that basic events are statistically independent, so that any two events A and B will fail with a probability

$$Pr\{A \cdot B\} = Pr\{A\} \times Pr\{B\}.$$

3.1.4 Simultaneity

In a FT, simultaneity among events may arise in two ways. Independent basic events can occur simultaneously if they have a discrete probability distribution with a non-null probability mass exactly at the same time. Because the failure probability distributions are usually considered as continuous functions with infinite support, the simultaneous occurrence has null probability, and can be neglected. A second case of simultaneity may arise at any level of a FT when there are repeated basic events. FTs with repeated events represent the most powerful combinatorial model in dependability [Malhotra 1994], and require ad hoc analysis techniques.

Nevertheless, the presence of repeated events across modules of dynamic gates has not yet been explored in its full generality. In [Yuge 2008], repeated events are allowed, but the paper does not provide any algorithm to derive the list of the cut sequences.

Let us consider the DFT in Fig. 3.1, in which event A is a repeated basic event. If basic events A , B , and C occur according to sequences $[B, C, A]$, or $[C, B, A]$,

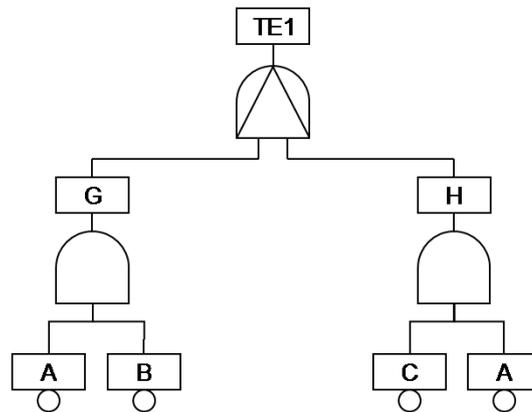


Figure 3.1: An example of DFT with one repeated basic event

intermediate events G and H occur simultaneously at the same time as A occurs. This example shows that intermediate nodes of a FT can occur simultaneously because of the presence of repeated basic events. The simultaneity problem has been briefly addressed in [Boudali 2007b], and has been solved by resorting to the concept of "non-determinism", a concept that is not easy to accept in engineering practice because many engineers believe that the behaviour of technical systems, and in particular control systems, must necessarily be deterministic. We assert that a choice must be made regarding the semantics of simultaneous events, and dynamic gates. For instance, in the case of simultaneous events in input to a PAND gate, two choices are possible (Fig. 3.1):

- if the order relation is considered strictly, when intermediate events G and H occur simultaneously, $TE1$ does not occur, and gate PAND would then be considered as being "non-inclusive" and
- if the order relation is not considered strictly, when intermediate events G and H occur simultaneously, $TE1$ occurs at the same time as G or H , and gate PAND would then be considered as being "inclusive".

Both interpretations of the order relation can be taken into account, and algebraically modeled.

3.2 Temporal model of non-repairable events

The structure function of SFTs is based on a Boolean model of events, and of basic events in particular. With this simple model, the only aspect which is taken into account is the presence or absence of failure. However, this Boolean model cannot render the order of occurrence of events which is necessary for the modeling of dynamic gates. To take into account the temporal aspect of events, we consider the top event, the intermediate events, and the basic events as *temporal functions*, which are piecewise right-continuous on $\mathbb{R}^+ \cup \{+\infty\}$, whose range is $\mathbb{B} = \{0,1\}$.

As we consider all events as non-repairable events, each of them is perfectly defined by its unique date of occurrence – noted $d(a)$ for an event a . A generic timing diagram of such an event a is given in Fig. 3.2. In this dissertation, we denote by \mathcal{E}_{nr} the set of these temporal functions, which corresponds to the set of non-repairable events.

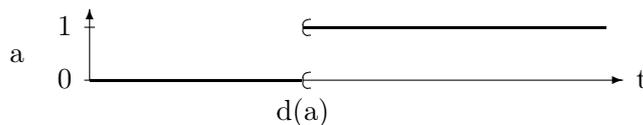


Figure 3.2: A non-repairable event

3.3 Temporal model of Boolean operators

DFTs may contain static gates, so the algebraic framework introduced must be able to capture the static behaviour of Boolean gates as well. Boolean gates are commonly modeled by Boolean operators OR and AND, so a temporal model of both operators is respectively provided in Sections 3.3.1 and 3.3.2. As it is proven in Section 3.3.3, the set of non-repairable events \mathcal{E}_{nr} equipped with the Boolean operators OR and AND as defined in Sections 3.3.1 and 3.3.2 has the same algebraic structure as the Boolean algebra of Boolean variables which is used to model SFTs, so the structure function of SFTs can still be determined and simplified thanks to this algebraic framework.

3.3.1 Operator OR

The temporal definition of operator OR (with symbol $+$), based on the dates of occurrence of a and b , is

$$\begin{aligned} + : \mathcal{E}_{nr} \times \mathcal{E}_{nr} &\longrightarrow \mathcal{E}_{nr} \\ (a, b) &\longmapsto a + b \end{aligned}$$

where

$$d(a + b) = \begin{cases} d(a) & \text{if } d(a) < d(b) \\ d(a) & \text{if } d(a) = d(b) \\ d(b) & \text{if } d(a) > d(b) \end{cases}$$

The result of the composition of two events a and b by operator OR is illustrated by the timing diagrams in Fig. 3.3 in three cases: Case 1: $d(a) < d(b)$, Case 2: $d(a) = d(b)$, Case 3: $d(a) > d(b)$.

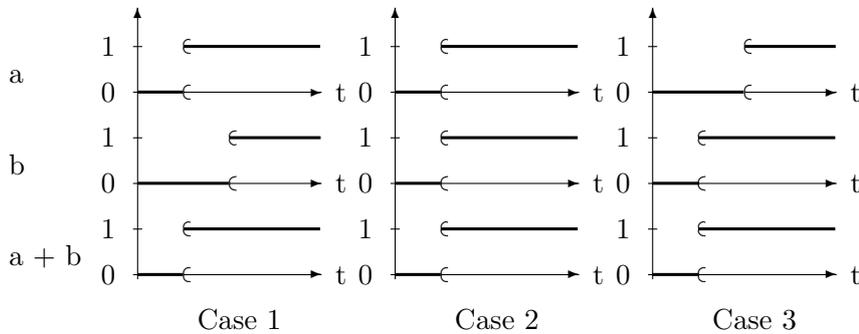


Figure 3.3: Timing diagrams of Boolean operator OR

3.3.2 Operator AND

The temporal definition of operator AND (with symbol \cdot), based on the dates of occurrence of a and b , is

$$\begin{aligned} \cdot : \mathcal{E}_{nr} \times \mathcal{E}_{nr} &\longrightarrow \mathcal{E}_{nr} \\ (a, b) &\longmapsto a \cdot b \end{aligned}$$

where

$$d(a \cdot b) = \begin{cases} d(b) & \text{if } d(a) < d(b) \\ d(a) & \text{if } d(a) = d(b) \\ d(a) & \text{if } d(a) > d(b) \end{cases}$$

The result of the composition of two events a and b by operator AND is illustrated by the timing diagrams in Fig. 3.4 in three cases: Case 1: $d(a) < d(b)$, Case 2: $d(a) = d(b)$, Case 3: $d(a) > d(b)$.

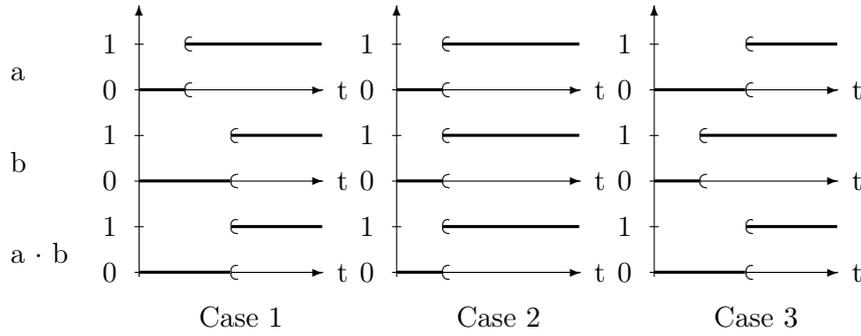


Figure 3.4: Timing diagrams of Boolean operator AND

3.3.3 Algebraic structure of $(\mathcal{E}_{nr}, +, \cdot)$

The identity elements of operators OR and AND in \mathcal{E}_{nr} are denoted by \perp , and \top , respectively, to which these dates can be assigned:

$$d(\perp) = +\infty \quad , \quad d(\top) = 0.$$

\perp is the never-occurring event whereas \top is the always-occurring event.

The operators OR and AND defined in Sections 3.3.1 and 3.3.2 satisfy the following theorems, for all $a, b, c \in \mathcal{E}_{nr}$:

- operators OR and AND are commutative

$$a + b = b + a \tag{3.1}$$

$$a \cdot b = b \cdot a \tag{3.2}$$

- operators OR and AND are associative

$$a + (b + c) = (a + b) + c \quad (3.3)$$

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c \quad (3.4)$$

- operators OR and AND are idempotent

$$a + a = a \quad (3.5)$$

$$a \cdot a = a \quad (3.6)$$

- operator AND is distributive over operator OR

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c) \quad (3.7)$$

- operator OR allows \perp as its identity element

$$a + \perp = a \quad (3.8)$$

- operator AND allows \top as its identity element

$$a \cdot \top = a \quad (3.9)$$

- operator AND allows \perp as an absorbing element

$$a \cdot \perp = \perp \quad (3.10)$$

The proofs of these 10 theorems can be found in Appendix A.

According to the definition of operator OR in Section 3.3.1, $+$ is an inner composition law on \mathcal{E}_{nr} which is associative (3.3) and allows \perp as its identity element (3.8), so $(\mathcal{E}_{nr}, +)$ is a monoid [Lang 2005]. Besides, $+$ is commutative (3.1) and idempotent (3.5), so $(\mathcal{E}_{nr}, +)$ is an idempotent monoid.

According to the definition of operator AND in Section 3.3.2, \cdot is an inner composition law on \mathcal{E}_{nr} which is associative (3.4) and allows \top as its identity element (3.9), so $(\mathcal{E}_{nr}, \cdot)$ is a monoid. Besides, \cdot is commutative (3.2) and idempotent (3.6), so $(\mathcal{E}_{nr}, \cdot)$ is an idempotent monoid.

On the one hand, $(\mathcal{E}_{nr}, +)$ is an idempotent monoid, so it is an Abelian monoid and its identity element is \perp . On the other hand, $(\mathcal{E}_{nr}, \cdot)$ is an idempotent monoid, so it is a monoid and its identity element is \top . Besides, \cdot is commutative (3.2) and left-distributive over $+$ (3.7) so \cdot is left-distributive and right-distributive over $+$, and \perp is an absorbing element over \cdot (3.10). As a consequence, $(\mathcal{E}_{nr}, +, \cdot)$ is a semiring. Given $(\mathcal{E}_{nr}, +)$ is idempotent, $(\mathcal{E}_{nr}, +, \cdot)$ is a dioid. \cdot is commutative (3.2) so $(\mathcal{E}_{nr}, +, \cdot)$ is an Abelian dioid, like $(\{0, 1\}, +, \cdot)$, and the properties of Boolean algebra that are commonly used for the simplification of SFTs can still be applied with our algebraic framework, and their structure functions can be determined as

usual. In particular, operators OR and AND satisfy the two additional following theorems, which are theorems which hold on Boolean algebras:

$$a + (b \cdot c) = (a + b) \cdot (a + c) \quad (3.11)$$

$$a + \top = \top \quad (3.12)$$

as well as the 2 following theorems, which are particular cases of theorems (3.7) and (3.11):

$$a + (a \cdot b) = a \quad (3.13)$$

$$a \cdot (a + b) = a \quad (3.14)$$

$(\mathcal{E}_{nr}, +, \cdot)$ thus has an algebraic structure which allows to express gates OR, AND, and K-out-of-N, and to determine and simplify the structure function of SFTs as it is commonly done by using the classical Boolean algebra of Boolean variables.

3.4 Temporal operators

This section presents the three temporal operators which have been defined to model the order of occurrence of events in dynamic gates. The non-inclusive BEFORE operator was introduced in [Merle 2007a, Merle 2007b] to model dynamic gates and be able to determine the structure function of any DFT. Its definition is given in Section 3.4.1. The SIMULTANEOUS operator was introduced in [Merle 2009] to allow to take into account the simultaneity of events, and an Inclusive BEFORE operator was introduced to be able to model an alternative inclusive version of dynamic gates. The definition of operators SIMULTANEOUS and Inclusive BEFORE is given in Sections 3.4.2 and 3.4.3, respectively.

3.4.1 Operator non-inclusive BEFORE

The formal definition of operator non-inclusive BEFORE (BF, with symbol \triangleleft), based on the dates of occurrence of a and b , is

$$\begin{aligned} \triangleleft : \mathcal{E}_{nr} \times \mathcal{E}_{nr} &\longrightarrow \mathcal{E}_{nr} \\ (a, b) &\longmapsto a \triangleleft b \end{aligned}$$

where

$$d(a \triangleleft b) = \begin{cases} d(a) & \text{if } d(a) < d(b) \\ +\infty & \text{if } d(a) = d(b) \\ +\infty & \text{if } d(a) > d(b) \end{cases}$$

The result of the composition of two events a and b by operator BF is illustrated by the timing diagrams in Fig. 3.5 in three cases: Case 1: $d(a) < d(b)$, Case 2: $d(a) = d(b)$, Case 3: $d(a) > d(b)$.

It can be noted that $a \triangleleft b$ occurs if a occurs before b , or if a occurs and b does not occur at all. Indeed, in this case, $b \equiv \perp$ and $a \triangleleft \perp = a$, according to theorem (3.21).

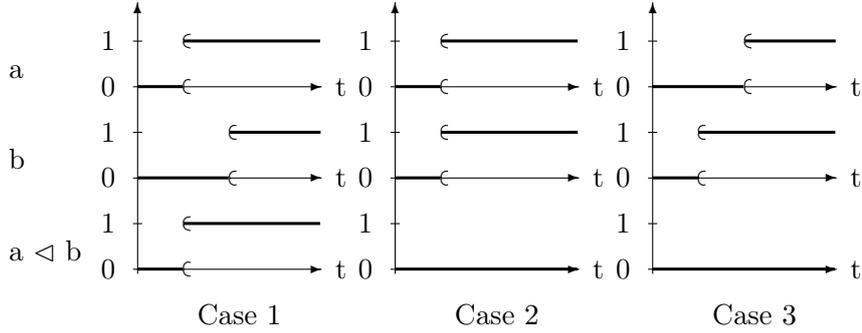


Figure 3.5: Timing diagrams of operator non-inclusive BEFORE (BF)

3.4.2 Operator SIMULTANEOUS

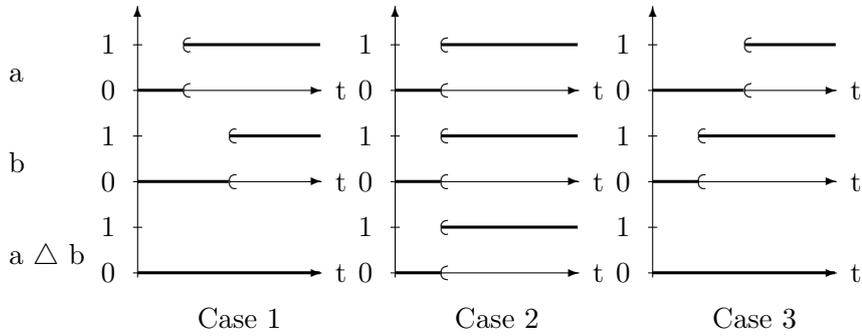


Figure 3.6: Timing diagrams of operator SIMULTANEOUS (SM)

The formal definition of operator SIMULTANEOUS (SM, with symbol Δ), based on the dates of occurrence of a and b , is

$$\begin{aligned} \Delta : \mathcal{E}_{nr} \times \mathcal{E}_{nr} &\longrightarrow \mathcal{E}_{nr} \\ (a, b) &\longmapsto a \Delta b \end{aligned}$$

where

$$d(a \Delta b) = \begin{cases} +\infty & \text{if } d(a) < d(b) \\ d(a) & \text{if } d(a) = d(b) \\ +\infty & \text{if } d(a) > d(b) \end{cases}$$

The result of the composition of two events a and b by operator SM is illustrated by the timing diagrams in Fig. 3.6 in three cases: Case 1: $d(a) < d(b)$, Case 2: $d(a) = d(b)$, Case 3: $d(a) > d(b)$.

In Section 3.1.2, the hypothesis was made that the failure time distribution of basic events is continuous, and that two basic events cannot occur simultaneously. Hence, for any two statistically independent basic events a and b with the above characteristics, the following relation holds.

$$a \Delta b = \perp \tag{3.15}$$

Indeed, one of the hypotheses of DFTs is the stochastic independence of events. We express this hypothesis under the form of the deterministic equivalence given in (3.15).

3.4.3 Operator Inclusive BEFORE

Based on the previous two operators, we can introduce a non-strict or Inclusive BEFORE (IBF, with symbol \trianglelefteq) operator

$$a \trianglelefteq b = a \triangleleft b + a \triangle b \tag{3.16}$$

whose definition, based on the dates of occurrence of a and b , is

$$\begin{aligned} \trianglelefteq : \mathcal{E}_{nr} \times \mathcal{E}_{nr} &\longrightarrow \mathcal{E}_{nr} \\ (a, b) &\longmapsto a \trianglelefteq b \end{aligned}$$

where

$$d(a \trianglelefteq b) = \begin{cases} d(a) & \text{if } d(a) < d(b) \\ d(a) & \text{if } d(a) = d(b) \\ +\infty & \text{if } d(a) > d(b) \end{cases}$$

The result of the composition of two events a and b by operator IBF is illustrated by the timing diagrams in Fig. 3.7 in three cases: Case 1: $d(a) < d(b)$, Case 2: $d(a) = d(b)$, Case 3: $d(a) > d(b)$.

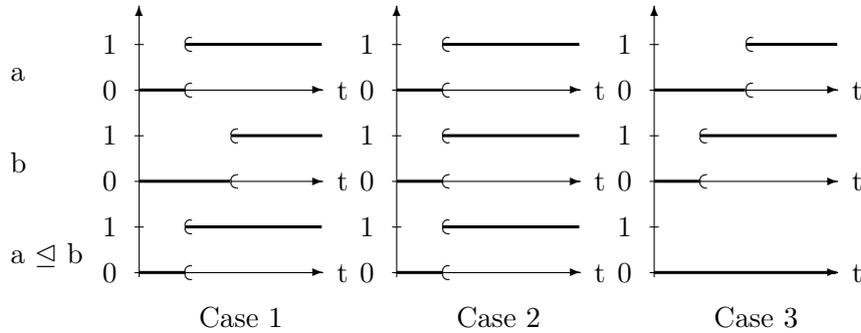


Figure 3.7: Timing diagrams of operator Inclusive BEFORE (IBF)

According to these timing diagrams, and to (3.16), $a \trianglelefteq b$ occurs in two cases: when a occurs strictly before b , Case 1 (which corresponds to $a \triangleleft b$); and when a occurs at the same time as b , Case 2 (which corresponds to $a \triangle b$). It can also be noted that $a \trianglelefteq b$ occurs if a occurs and b does not occur at all. Indeed, in this case, $b \equiv \perp$ and $a \trianglelefteq \perp = a$, according to theorem (3.50).

3.5 Theorems

The three temporal operators defined in Section 3.4 satisfy many theorems which allow to develop and simplify the structure function of DFTs. The proofs of these

theorems can be found in Appendix A. The theorems satisfied by operators non-inclusive BEFORE, SIMULTANEOUS, and Inclusive BEFORE are presented in Sections 3.5.1, 3.5.2, and 3.5.3, respectively. A few simplification theorems involving the 3 operators are also introduced in Section 3.5.4.

3.5.1 Theorems satisfied by operator non-inclusive BEFORE

Operator non-inclusive BEFORE has the following properties, for all $a, b, c \in \mathcal{E}_{nr}$:

- it is not commutative

$$(a \triangleleft b) \cdot (b \triangleleft a) = \perp \quad (3.17)$$

- it is not associative

$$\begin{aligned} a \triangleleft (b \triangleleft c) &= (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) \\ &= (a \triangleleft b) + (a \cdot b \cdot (c \triangle b)) \end{aligned} \quad (3.18)$$

$$(a \triangleleft b) \triangleleft c = (a \triangleleft b) \cdot (a \triangleleft c) \quad (3.19)$$

- it allows \perp as a left absorbing element and a right identity element

$$\perp \triangleleft a = \perp \quad (3.20)$$

$$a \triangleleft \perp = a \quad (3.21)$$

- it is not idempotent

$$a \triangleleft a = \perp \quad (3.22)$$

- it is right-distributive over operators $+$, \cdot , and \triangle only

$$a \triangleleft (b + c) = (a \triangleleft b) \cdot (a \triangleleft c) \quad (3.23)$$

$$a \triangleleft (b \cdot c) = (a \triangleleft b) + (a \triangleleft c) \quad (3.24)$$

$$a \triangleleft (b \triangle c) = (a \cdot (b \triangleleft c)) + (a \cdot (c \triangle b)) + (a \triangleleft b) + (a \triangleleft c) \quad (3.25)$$

$$a \triangleleft (b \triangle b) = (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b)) \quad (3.26)$$

$$(a + b) \triangleleft c = (a \triangleleft c) + (b \triangleleft c) \quad (3.27)$$

$$(a \cdot b) \triangleleft c = (a \triangleleft c) \cdot (b \triangleleft c) \quad (3.28)$$

$$(a \triangle b) \triangleleft c = (a \triangle b) \cdot (a \triangleleft c) = (a \triangle b) \cdot (b \triangleleft c) = (a \triangleleft c) \triangle (b \triangleleft c) \quad (3.29)$$

$$(a \triangle b) \triangleleft c = (a \triangle b) \cdot (a \triangleleft c) \quad (3.30)$$

- it satisfies the following simplification theorems

$$a + (a \triangleleft b) = a \quad (3.31)$$

$$(a \triangleleft b) + b = a + b \quad (3.32)$$

$$a \cdot (a \triangleleft b) = a \triangleleft b \quad (3.33)$$

- it satisfies the following redundancy theorem

$$(a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = (a \triangleleft b) \cdot (b \triangleleft c) \quad (3.34)$$

3.5.2 Theorems satisfied by operator SIMULTANEOUS

Operator SIMULTANEOUS has the following properties, for all $a, b, c \in \mathcal{E}_{nr}$:

- it is commutative

$$a \triangle b = b \triangle a \quad (3.35)$$

- it is associative

$$a \triangle (b \triangle c) = (a \triangle b) \triangle c = (a \triangle b) \cdot (b \triangle c) = (a \triangle c) \cdot (c \triangle b) \quad (3.36)$$

- it allows \perp as an absorbing element

$$a \triangle \perp = \perp \quad (3.37)$$

- it is idempotent

$$a \triangle a = a \quad (3.38)$$

- it is not distributive over any other operator but satisfies the following development theorems

$$\begin{aligned} a \triangle (b + c) &= (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) + (a \triangle c) \cdot (c \triangleleft b) \\ &= (a \triangle b) \cdot (b \trianglelefteq c) + (a \triangle c) \cdot (c \trianglelefteq b) \end{aligned} \quad (3.39)$$

$$\begin{aligned} a \triangle (b \cdot c) &= (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) + (a \triangle c) \cdot (b \triangleleft c) \\ &= (a \triangle b) \cdot (c \trianglelefteq b) + (a \triangle c) \cdot (b \trianglelefteq c) \end{aligned} \quad (3.40)$$

$$a \triangle (b \triangleleft c) = (a \triangle b) \cdot (b \triangleleft c) \quad (3.41)$$

$$a \triangle (b \trianglelefteq c) = (a \triangle b) \cdot (b \trianglelefteq c) \quad (3.42)$$

- it satisfies the following simplification theorems

$$a + (a \triangle b) = a \quad (3.43)$$

$$a \cdot (a \triangle b) = a \triangle b \quad (3.44)$$

- it satisfies the following redundancy theorem

$$(a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = (a \triangle b) \cdot (b \triangle c) \quad (3.45)$$

3.5.3 Theorems satisfied by operator Inclusive BEFORE

Operator Inclusive BEFORE has the following properties, for all $a, b, c \in \mathcal{E}_{nr}$:

- it is not commutative

$$(a \trianglelefteq b) \cdot (b \trianglelefteq a) = a \triangle b \quad (3.46)$$

- it is not associative

$$a \trianglelefteq (b \trianglelefteq c) = (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b)) + (a \triangle b) \cdot (b \trianglelefteq c) \quad (3.47)$$

$$(a \trianglelefteq b) \trianglelefteq c = (a \trianglelefteq b) \cdot (a \trianglelefteq c) \quad (3.48)$$

- it allows \perp as a left absorbing element and a right identity element

$$\perp \trianglelefteq a = \perp \quad (3.49)$$

$$a \trianglelefteq \perp = a \quad (3.50)$$

- it is idempotent

$$a \trianglelefteq a = a \quad (3.51)$$

- it is right-distributive over operators $+$, \cdot , and \triangle only

$$a \trianglelefteq (b + c) = (a \trianglelefteq b) \cdot (a \trianglelefteq c) \quad (3.52)$$

$$a \trianglelefteq (b \cdot c) = (a \trianglelefteq b) + (a \trianglelefteq c) \quad (3.53)$$

$$a \trianglelefteq (b \triangleleft c) = (a \triangleleft b) + (a \cdot b \cdot (c \trianglelefteq b)) + (a \triangle b) \cdot (b \triangleleft c) \quad (3.54)$$

$$\begin{aligned} a \trianglelefteq (b \triangle c) &= (a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) \\ &\quad + (a \triangle b) \cdot (b \triangle c) \end{aligned} \quad (3.55)$$

$$(a + b) \trianglelefteq c = (a \trianglelefteq c) + (b \trianglelefteq c) \quad (3.56)$$

$$(a \cdot b) \trianglelefteq c = (a \trianglelefteq c) \cdot (b \trianglelefteq c) \quad (3.57)$$

$$(a \triangle b) \trianglelefteq c = (a \triangle b) \cdot (a \trianglelefteq c) = (a \triangle b) \cdot (b \trianglelefteq c) = (a \trianglelefteq c) \triangle (b \trianglelefteq c) \quad (3.58)$$

$$(a \triangleleft b) \trianglelefteq c = (a \triangleleft b) \cdot (a \trianglelefteq c) \quad (3.59)$$

- it satisfies the following simplification theorems

$$a + (a \trianglelefteq b) = a \quad (3.60)$$

$$b + (a \trianglelefteq b) = a + b \quad (3.61)$$

$$a \cdot (a \trianglelefteq b) = a \trianglelefteq b \quad (3.62)$$

$$(a \trianglelefteq b) + (b \trianglelefteq a) = a + b \quad (3.63)$$

$$(a \cdot (b \trianglelefteq a)) + (b \cdot (a \trianglelefteq b)) = a \cdot b \quad (3.64)$$

$$(a \trianglelefteq b) + (a \cdot (b \trianglelefteq a)) = a \quad (3.65)$$

- it satisfies the following redundancy theorem

$$(a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = (a \trianglelefteq b) \cdot (b \trianglelefteq c) \quad (3.66)$$

3.5.4 Simplification theorems

Temporal operators satisfy the following simplification theorems, for all $a, b, c \in \mathcal{E}_{nr}$:

$$(a \sqsubseteq b) + (a \triangleleft b) = a \sqsubseteq b \quad (3.67)$$

$$(a \sqsubseteq b) + (a \triangle b) = a \sqsubseteq b \quad (3.68)$$

$$(a \triangleleft b) \cdot (a \triangle b) = \perp \quad (3.69)$$

$$(a \triangleleft b) \cdot (b \triangle c) = (a \triangleleft c) \cdot (b \triangle c) \quad (3.70)$$

$$(a \sqsubseteq b) \cdot (a \triangleleft b) = a \triangleleft b \quad (3.71)$$

$$(a \triangleleft b) \cdot (b \sqsubseteq a) = \perp \quad (3.72)$$

$$(a \sqsubseteq b) \cdot (a \triangle b) = a \triangle b \quad (3.73)$$

$$(a \triangleleft b) + (a \triangle b) + (b \triangleleft a) = a + b \quad (3.74)$$

$$(a \cdot (b \triangleleft a)) + (a \triangle b) + (b \cdot (a \triangleleft b)) = a \cdot b \quad (3.75)$$

$$(a \triangleleft b) + (a \triangle b) + (a \cdot (b \triangleleft a)) = a \quad (3.76)$$

$$(a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \sqsubseteq c) = (a \triangleleft b) \cdot (b \triangleleft c) \quad (3.77)$$

3.6 Discussion on the similarities and differences between our approach and Pandora

Pandora [Walker 2006] was designed to enable FTs to model event sequences and relative temporal ordering by adding new gates to the common dynamic gates, which are gates Priority-OR (POR) and Simultaneous-AND (SAND), while the PAND gate is the only dynamic gate considered in Pandora. We chose to consider all the dynamic gates considered in [Dugan 2000], and to include the aspects modeled by gates POR and SAND in Pandora in our framework under the form of specific temporal operators, without introducing any new gate. This is the first difference between our approach and Pandora: Pandora considers gates POR and SAND, and the dynamic gate PAND; our approach considers all the dynamic gates considered in [Dugan 2000]: gates PAND, FDEP, and Spare.

However, there is a semantics similarity between the temporal gates of Pandora and the temporal operators that we define to model dynamic gates:

- in Pandora, a POR gate, whose symbol is $|$, with two input events A and B means that either A occurs and B does not, or both occur and A occur first, which corresponds to $A \triangleleft B$ in our algebraic framework;
- in Pandora, a SAND gate, whose symbol is $\&$, with two input events A and B means that B occurs at the same time as A occurs, which corresponds to $A \triangle B$ in our algebraic framework; and

- in Pandora, a PAND gate, whose symbol is \triangleleft , with two input events A and B means that A occurs before B occurs, which corresponds to $B \cdot (A \triangleleft B)$ in our algebraic framework.

It can be noted that gates POR and PAND can be modeled by the single temporal operator Non-inclusive BEFORE (\triangleleft) in our framework. Besides, both gates could be considered as inclusive by considering the temporal operator Inclusive BEFORE (\trianglelefteq), whereas Pandora does not allow to take into account this second aspect.

On the one hand, Pandora does not provide any algebraic framework to define temporal gates, and the laws provided in [Walker 2009] are demonstrated by using Temporal Truth Tables in which all the possible sequences of event occurrences are considered, so Pandora does not really allow the determination of a structure function for DFTs. On the other hand, the algebraic framework that we provide allows to determine the same theorems as the ones listed in [Walker 2009], as the 3 temporal gates defined in [Walker 2009] can be modeled by means of temporal operators SM and BF, and to determine a structure function for DFTs. However, the minimal cut sequences obtained by both approaches are the same. Finally, no 'inclusive' model of temporal gates POR and PAND exists in Pandora, so all the theorems that we provide and which contain temporal operator IBF do not exist in Pandora.

behavioural model of dynamic gates

Contents

4.1	behavioural model of the PAND gate	44
4.2	behavioural model of the FDEP gate	44
4.3	behavioural model of Spare gates	45
4.3.1	behavioural model of Spare gates with 2 input events	46
4.3.2	behavioural model of Spare gates with 3 input events	48
4.3.3	behavioural model of Spare gates with n input events	48
4.3.4	Specific case of Cold and Hot spare events	49

This chapter presents the behavioural model of dynamic gates which has been built thanks to the 3 temporal operators defined in Chapter 3 in order to determine the structure function of DFTs. The behavioural model of gates PAND, FDEP, and Spare is presented in Sections 4.1, 4.2, and 4.3, respectively.

4.1 behavioural model of the PAND gate

The definition of gate PAND recalled in Section 2.2.1.2 is shown in Fig. 4.1. As explained in Section 3.1.4, two interpretations of the order relation "Before" can be taken into account, and algebraically modeled, thus leading to two behavioural models for the PAND gate. However, the non-strict inclusive interpretation of the PAND gate seems more coherent with the designers' expectations. For this reason, in the remainder of this dissertation, the inclusive model will be retained, even though both interpretations can be adopted without any change in the algebraic framework that we propose.

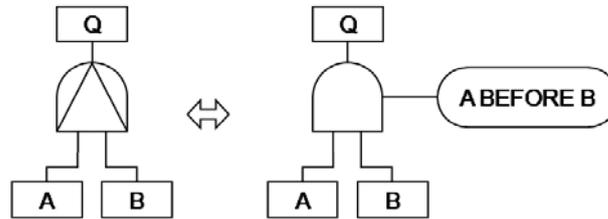


Figure 4.1: Definition of gate PAND from [Vesely 1981]

If we retain the inclusive interpretation of the PAND gate, then it is assumed that the output event Q of the PAND gate in Fig. 4.1 occurs if its input events A and B occur simultaneously. A behavioural inclusive model of gate PAND can hence be determined as

$$Q = (A \cdot B) \cdot (A \preceq B)$$

This model can be simplified, thanks to the theorems of Section 3.5, as follows:

$$\begin{aligned} Q &= (A \cdot B) \cdot (A \preceq B) \\ &\stackrel{(3.62)}{=} B \cdot (A \preceq B) \end{aligned}$$

In the previous equation, the notation $\stackrel{(3.62)}{=}$ indicates that the second expression – $B \cdot (A \preceq B)$ – is obtained from the first expression – $(A \cdot B) \cdot (A \preceq B)$ – by applying theorem (3.62) from Section 3.5. This notation will be used in the remainder of this dissertation.

For the reasons explained above, this inclusive model of the PAND gate will be the one retained in the remainder of this dissertation.

4.2 behavioural model of the FDEP gate

The definition of gate FDEP was recalled in Section 2.2.1.3, and an FDEP gate with 2 dependent basic events A and B is shown in Fig. 4.2.

In Fig. 4.2, basic events A and B can fail by themselves, or can be forced to fail by trigger event T . In accordance with [Boudali 2006], we choose to denote the

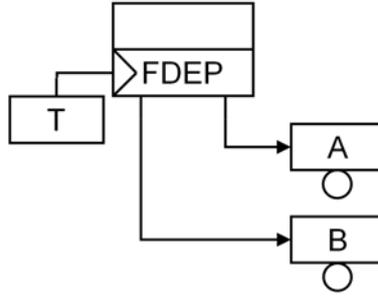


Figure 4.2: An FDEP gate with 2 dependent basic events

global behaviour of basic events A , and B by the substituted variables A_T , and B_T to explicitly indicate the effect of trigger T : basic event A fails (A_T) if it is forced to fail by the trigger event (T) or if it fails by itself before the trigger event fails ($A \leq T$). The behavioural model of gate FDEP thus is

$$\begin{cases} A_T = T + (A \leq T) \\ B_T = T + (B \leq T) \end{cases}$$

This model can be simplified, thanks to the theorems of Section 3.5, as follows:

$$\begin{cases} A_T = T + (A \leq T) \stackrel{(3.61)}{=} A + T \\ B_T = T + (B \leq T) \stackrel{(3.61)}{=} B + T \end{cases}$$

This simplification allows to prove that the behaviour of dynamic gate FDEP is equivalent to the behaviour of gate OR, as we showed in [Merle 2010a] and as it is proposed by some authors [Stamatelatos 2002].

4.3 behavioural model of Spare gates

According to [Stamatelatos 2002], it can be assumed that spares are cold, warm, or hot: cold spares do not fail (their dormancy is equal to zero), hot spares fail at the same rate as active (their dormancy is equal to one), and warm spares fail somewhere between cold and hot spares. In this section, we consider that there is only one type of spare event, which is the warm spare event, and that cold and hot spare events are only particular cases of the warm spare event. For this reason, the models of Spare gates which are presented correspond to the models of Warm Spare gates – Spare gates with warm spare events.

The behavioural model of Spare gates will be presented in an increasing order of complexity. The model of Spare gates with 2 input events is presented in Section 4.3.1, and the model of Spare gates with 3 input events is presented in Section 4.3.2. Section 4.3.3 describes how to generalize the models presented in Sections 4.3.1 and 4.3.2 to the case of Spare gates with n input events. Finally, the particular case of cold and hot spare events is considered in Section 4.3.4.

4.3.1 behavioural model of Spare gates with 2 input events

In this section, we completely detail the behavioural model of Spare gates with 2 input events in the main configurations which may be encountered in DFTs. The different cases are treated in an increasing order of complexity, from a single Spare gate in Section 4.3.1.1 to 2 Spare gates sharing a spare event in Section 4.3.1.2, and even to the generalization to n Spare gates sharing a spare event in Section 4.3.1.3.

4.3.1.1 behavioural model of a single Spare gate

Let us consider a Spare gate with 2 input events – the primary event A and one spare event B – as shown in Fig. 4.3.

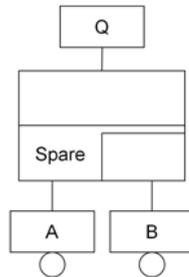


Figure 4.3: A single Spare gate with one primary event A and one spare event B

As stated in [Stamatelatos 2002], the output Q of the gate occurs when the primary and all spares have failed, so when A and B have failed, in this case. A and B are basic events and cannot fail simultaneously – $A \triangle B = \perp$ – so Q occurs if A and B fail according to sequences $[A, B]$ or $[B, A]$. It is important to note that in sequence $[A, B]$, B fails while in its active mode (denoted as B_a), whereas in sequence $[B, A]$, B fails while in its dormant mode (denoted as B_d). It is essential to distinguish both failure modes by using two different variables, for quantitative analysis purposes. Indeed, B does not have the same failure distribution when it fails during its dormant mode ($B \equiv B_d$) or during its active mode ($B \equiv B_a$). As we aim at making possible the quantitative analysis of DFTs from their structure function, this structure function must hence provide sufficient information to know whether spare events are in their dormant or active mode. The behavioural model of gate Spare can hence be expressed as

$$Q = B_a \cdot (A \triangleleft B_a) + A \cdot (B_d \triangleleft A).$$

Furthermore, as B cannot be both in an active state and in a dormant state, we have

$$B_d \cdot B_a = \perp.$$

4.3.1.2 behavioural model of 2 Spare gates sharing a spare event

Let us consider 2 Spare gates with 2 input events – with primary events A and B – sharing a spare event C , as shown in Fig. 4.4.

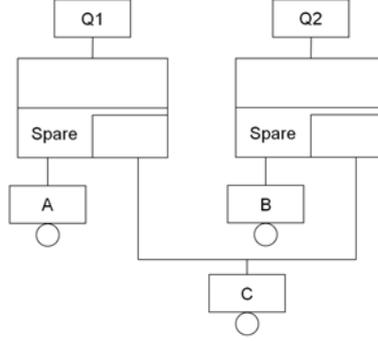


Figure 4.4: Two Spare gates sharing a spare event

If we focus on the Spare gate on the left side, $Q1$ occurs as soon as A and C have failed – as stated in Section 4.3.1.1 – or if A fails and C is made unavailable because B has failed before A . As a consequence, the behavioural model of the Spare gate on the left side is

$$\begin{cases} Q1 = C_a \cdot (A \triangleleft C_a) + A \cdot (C_d \triangleleft A) + A \cdot (B \triangleleft A) \\ C_d \cdot C_a = \perp \end{cases}$$

The algebraic expression for the Spare gate on the right side can be determined in the same way by symmetry. Consequently, the final behavioural model of any of two Spare gates sharing a spare event is

$$\begin{cases} Q1 = C_a \cdot (A \triangleleft C_a) + A \cdot (C_d \triangleleft A) + A \cdot (B \triangleleft A) \\ Q2 = C_a \cdot (B \triangleleft C_a) + B \cdot (C_d \triangleleft B) + B \cdot (A \triangleleft B) \\ C_d \cdot C_a = \perp \end{cases}$$

4.3.1.3 behavioural model of n Spare gates sharing a spare event

Let us consider n Spare gates with 1 output event Q_i and 2 input events: a primary event $P_i - i \in \{1, \dots, n\}$ – and a common spare event S .

If we focus on the first Spare gate, Q_1 occurs as soon as P_1 and S have failed – as stated in Section 4.3.1.1 – or if P_1 fails and S is made unavailable because the primary event of any of the other Spare gates has failed before P_1 . As a consequence, the behavioural model of the first Spare gate is

$$\begin{cases} Q_1 = S_a \cdot (P_1 \triangleleft S_a) + P_1 \cdot (S_d \triangleleft P_1) + \sum_{i \neq 1} P_i \cdot (P_i \triangleleft P_1) \\ S_d \cdot S_a = \perp \end{cases}$$

The algebraic expression for $Q_i, i \in \{1, \dots, n\}$, can be determined in the same way by symmetry. Consequently, the final behavioural model of any of n Spare gates sharing a spare event is

$$\begin{cases} Q_i = S_a \cdot (P_i \triangleleft S_a) + P_i \cdot (S_d \triangleleft P_i) + \sum_{j \neq i} P_j \cdot (P_j \triangleleft P_i) \\ S_d \cdot S_a = \perp \end{cases}$$

4.3.2 behavioural model of Spare gates with 3 input events

Let us consider a Spare gate with 3 input events – the primary event A and two spare events B and C – as shown in Fig. 4.5.

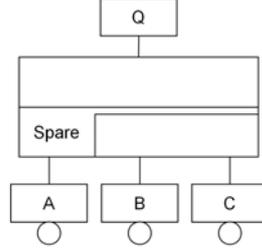


Figure 4.5: A single Spare gate with one primary event A and two spare events B and C

As stated in [Stamatelatos 2002], the output Q of the gate occurs when the primary and all spares have failed, so when A , B , and C have failed, in this case. A , B , and C are basic events and cannot fail simultaneously so Q occurs if A , B , and C fail according to sequences $[A, B, C]$, $[A, C, B]$, $[B, A, C]$, $[B, C, A]$, $[C, A, B]$, or $[C, B, A]$. It is important to note that, when the quantitative analysis is performed from the structure function, B and C will not have the same distribution function in the 6 sequences. For instance, in sequence $[A, B, C]$, both B and C fail during their active mode (denoted by B_a and C_a), whereas in sequence $[B, C, A]$, both B and C fail during their dormant mode (denoted by B_d and C_d). The behavioural model of the Spare gate can hence be expressed as

$$\begin{aligned} Q = & C_a \cdot (A \triangleleft B_a) \cdot (B_a \triangleleft C_a) + B_a \cdot (A \triangleleft C_d) \cdot (C_d \triangleleft B_a) \\ & + C_a \cdot (B_d \triangleleft A) \cdot (A \triangleleft C_a) + A \cdot (B_d \triangleleft C_d) \cdot (C_d \triangleleft A) \\ & + B_a \cdot (C_d \triangleleft A) \cdot (A \triangleleft B_a) + A \cdot (C_d \triangleleft B_d) \cdot (B_d \triangleleft A) \end{aligned}$$

As B and C cannot be both in an active state and in a dormant state, we have

$$\begin{cases} B_d \cdot B_a = \perp \\ C_d \cdot C_a = \perp \end{cases}$$

The behavioural model of many Spare gates with 3 input events sharing 2 spare events can be deduced from this model by considering the same approach as in Section 4.3.1.2.

4.3.3 behavioural model of Spare gates with n input events

The behavioural models of Spare gates with 2 and 3 input events presented in Sections 4.3.1 and 4.3.2 can be extended to the case of Spare gates with n input

events.

On the one hand, the behavioural model of a single Spare gate with n input events can be obtained by considering the $n!$ different failure sequences of basic events, and by denoting the dormant and active mode of each spare event by 2 substituted variables. Let us consider a Spare gate with n input events – the primary event P and $(n - 1)$ spare events S_1 to S_{n-1} – as shown in Fig. 4.6.

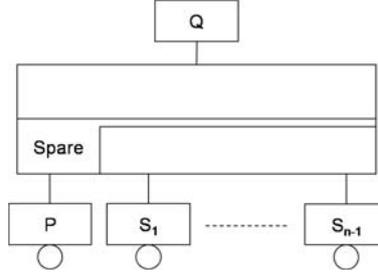


Figure 4.6: A single Spare gate with one primary event P and $(n - 1)$ spare events S_1 to S_{n-1}

The n input events of the Spare gate can occur according to $n!$ sequences, and each spare event $S_i, i \in \{1, \dots, n - 1\}$ is denoted by two variables S_{i_d} and S_{i_a} which model the dormant and active mode of the spare event, respectively. The behavioural model of the Spare gate can hence be expressed as

$$\begin{aligned}
 Q = & S_{n-1_a} \cdot (P \triangleleft S_{1_a}) \cdot (S_{1_a} \triangleleft S_{2_a}) \cdot \dots \cdot (S_{n-3_a} \triangleleft S_{n-2_a}) \cdot (S_{n-2_a} \triangleleft S_{n-1_a}) \\
 & + S_{n-2_a} \cdot (P \triangleleft S_{1_a}) \cdot (S_{1_a} \triangleleft S_{2_a}) \cdot \dots \cdot (S_{n-3_a} \triangleleft S_{n-1_d}) \cdot (S_{n-1_d} \triangleleft S_{n-2_a}) \\
 & + S_{n-3_a} \cdot (P \triangleleft S_{1_a}) \cdot (S_{1_a} \triangleleft S_{2_a}) \cdot \dots \cdot (S_{n-2_d} \triangleleft S_{n-1_d}) \cdot (S_{n-1_d} \triangleleft S_{n-3_a}) \\
 & + \dots \\
 & + S_{1_a} \cdot (S_{n-1_d} \triangleleft S_{n-2_d}) \cdot (S_{n-2_d} \triangleleft S_{n-3_d}) \cdot \dots \cdot (S_{2_d} \triangleleft P) \cdot (P \triangleleft S_{1_a}) \\
 & + P \cdot (S_{n-1_d} \triangleleft S_{n-2_d}) \cdot (S_{n-2_d} \triangleleft S_{n-3_d}) \cdot \dots \cdot (S_{2_d} \triangleleft S_{1_d}) \cdot (S_{1_d} \triangleleft P)
 \end{aligned}$$

with, $\forall i \in \{1, \dots, n - 1\}, S_{i_d} \cdot S_{i_a} = \perp$.

On the other hand, the behavioural model of p Spare gates with n input events sharing $(n - 1)$ spare events can also be determined by considering the same approach as in Section 4.3.1.2.

4.3.4 Specific case of Cold and Hot spare events

As explained above, the behavioural models presented in Sections 4.3.1 and 4.3.2 are the behavioural models of Spare gates in the general case of warm spare events. These behavioural models can be simplified in the specific cases of cold and hot spare events:

- if a spare event S is a cold spare event, it cannot fail while in a dormant state, so S_d never occurs and any expression containing S_d in the behavioural models can be removed;
- if a spare event S is a hot spare event, it has the same distribution function when in an active and in a dormant state, so $S_a \equiv S_d \equiv S$ and the behavioural models can be simplified.

Let us consider the behavioural model of a single Spare gate with 2 input events which was given in Section 4.3.1.1:

$$Q = B_a \cdot (A \triangleleft B_a) + A \cdot (B_d \triangleleft A).$$

On the one hand, if B is a cold spare event, it cannot fail while in its dormant state, so B_d never occurs and the second expression $- A \cdot (B_d \triangleleft A) -$ can be removed. The behavioural model of a single Spare gate with 1 primary event and 1 cold spare event hence becomes:

$$Q = B_a \cdot (A \triangleleft B_a).$$

On the other hand, if B is a hot spare event, it has the same distribution function when in its active and in its dormant state, so $B_a \equiv B_d \equiv B$. The behavioural model of a single Spare gate with 1 primary event and 1 hot spare event hence becomes:

$$\begin{aligned} Q &= B \cdot (A \triangleleft B) + A \cdot (B \triangleleft A) \\ &\stackrel{(3.15)}{=} B \cdot (A \triangleleft B) + A \triangle B + A \cdot (B \triangleleft A) \\ &\stackrel{(3.75)}{=} A \cdot B. \end{aligned}$$

It can hence be noted that the behavioural model of a Spare gate with hot spare events is equivalent to the behavioural model of an AND gate.

Probabilistic model of dynamic gates

Contents

5.1	A few probabilistic expressions	52
5.2	Probabilistic model of the PAND gate	53
5.3	Probabilistic model of the FDEP gate	53
5.4	Probabilistic model of Spare gates	53
5.4.1	Failure distribution of spare events	54
5.4.2	Probabilistic model of Spare gates with 2 input events	55
5.4.3	Probabilistic model of Spare gates with 3 input events	60
5.4.4	Probabilistic model of Spare gates with n input events	61

The behavioural model of dynamic gates which was presented in Chapter 4 allows to determine the structure function of DFTs, from which the cut sequences of the DFT can be extracted, as will be discussed in Chapter 6. However, the failure probability of the TE of DFTs cannot be determined directly from the structure function without a probabilistic model of dynamic gates. A few probabilistic expressions are provided in Section 5.1, and the probabilistic model of gates PAND, FDEP, and Spare which can be obtained from them is presented in Sections 5.2, 5.3, and 5.4, respectively.

5.1 A few probabilistic expressions

It is necessary to recall a few probabilistic expressions to be able to determine the probabilistic model of dynamic gates. Let us consider an event x with *cumulative distribution function* (Cdf) $F(x)$ and *probability density function* (pdf) $f(x) - f(x) = F'(x)$. The following expressions hold under the hypothesis of statistical independence [Amari 2003, Fussell 1976].

$$\begin{aligned}
 Pr \{a \cdot b\} (t) &= F_a(t) \times F_b(t) \\
 Pr \{a + b\} (t) &= F_a(t) + F_b(t) - F_a(t) \times F_b(t) \\
 Pr \{a \triangleleft b\} (t) &= \int_0^t f_a(u)(1 - F_b(u)) du \\
 Pr \{b \cdot (a \triangleleft b)\} (t) &= \int_0^t f_b(u) F_a(u) du
 \end{aligned} \tag{5.1}$$

Even though the 2 first expressions are quite common, the 2 last expressions are not. The fourth one can be obtained easily from [Fussell 1976]. Indeed,

$$\begin{aligned}
 Pr \{b \cdot (a \triangleleft b)\} (t) &= Pr \{[a, b]\} (t) \\
 &= \int_0^t f_b(u) \left(\int_0^u f_a(v) dv \right) du \\
 &= \int_0^t f_b(u) F_a(u) du
 \end{aligned}$$

The third expression can finally be deduced from the fourth expression by using theorem (3.76). Indeed, according to this theorem,

$$(a \triangleleft b) + (a \triangle b) + (a \cdot (b \triangleleft a)) = a$$

As a and b are statically independent, $a \triangle b = \perp$, and

$$(a \triangleleft b) + (a \cdot (b \triangleleft a)) = a$$

As a consequence, as $a \triangleleft b$ and $a \cdot (b \triangleleft a)$ are disjoint,

$$\begin{aligned}
 &Pr \{a \triangleleft b\} (t) + Pr \{a \cdot (b \triangleleft a)\} (t) = Pr \{a\} (t) \\
 \Leftrightarrow &Pr \{a \triangleleft b\} (t) + \int_0^t f_a(u) F_b(u) du = F_a(t) \\
 \Leftrightarrow &Pr \{a \triangleleft b\} (t) = F_a(t) - \int_0^t f_a(u) F_b(u) du \\
 \Rightarrow &\frac{dPr \{a \triangleleft b\} (t)}{dt} = f_a(t) - f_a(t) F_b(t) = f_a(t)(1 - F_b(t)) \\
 \Rightarrow &Pr \{a \triangleleft b\} (t) = \int_0^t f_a(u)(1 - F_b(u)) du
 \end{aligned}$$

5.2 Probabilistic model of the PAND gate

As explained in Section 4.1, we retain the inclusive behavioural model of the PAND gate which is

$$Q = B \cdot (A \triangleleft B)$$

The probability of occurrence of $B \cdot (A \triangleleft B)$ could be determined from the expressions presented in Section 5.1, but the probability of occurrence of $B \cdot (A \triangleleft B)$ is not known. We hence need to develop this expression to get expressions whose probability of occurrence is known. According to (3.16), $a \triangleleft b = a \triangleleft b + a \triangle b$, so

$$\begin{aligned} Q &= B \cdot (A \triangleleft B + A \triangle B) \\ &= B \cdot (A \triangleleft B) + B \cdot (A \triangle B) \\ &\stackrel{(3.35),(3.44)}{=} B \cdot (A \triangleleft B) + A \triangle B \end{aligned}$$

If A and B are two statistically independent events, $A \triangle B = \perp$ and the probabilistic model of the PAND gate can be determined as

$$\begin{aligned} F_Q(t) = Pr\{Q\}(t) &= Pr\{B \cdot (A \triangleleft B)\}(t) \\ &\stackrel{(5.1)}{=} \int_0^t f_B(u) F_A(u) du \end{aligned}$$

If A and B are two dependent events, $A \triangle B \neq \perp$ and the expression $A \triangle B$ must be developed to be able to determine the failure probability of the gate, as it will be shown later in this dissertation.

5.3 Probabilistic model of the FDEP gate

According to Section 4.2, the behavioural model of the FDEP gate is

$$\begin{cases} A_T = A + T \\ B_T = B + T \end{cases}$$

The probabilistic model of the FDEP gate can be obtained easily from this behavioural model:

$$\begin{cases} F_{A_T}(t) = Pr\{A_T\}(t) &= Pr\{A + T\}(t) \\ &\stackrel{(5.1)}{=} F_A(t) + F_T(t) - F_A(t) \times F_T(t) \\ F_{B_T}(t) = Pr\{B_T\}(t) &= Pr\{B + T\}(t) \\ &\stackrel{(5.1)}{=} F_B(t) + F_T(t) - F_B(t) \times F_T(t) \end{cases}$$

5.4 Probabilistic model of Spare gates

This section presents the probabilistic model of Spare gates. Even though the failure distribution of a spare event is quite easy to determine in its dormant mode, its

failure distribution in its active mode depends both on time and on the failure date of another event. This event can be the primary event of the Spare gate or another spare event. For this reason, the case of the failure distribution of spare events is addressed in Section 5.4.1. The probabilistic model of Spare gates will be presented in an increasing order of complexity. The model of Spare gates with 2 input events is presented in Section 5.4.2, and the model of Spare gates with 3 input events is presented in Section 5.4.3. Section 5.4.4 describes how to generalize the models presented in Sections 5.4.2 and 5.4.3 to the case of Spare gates with n input events.

5.4.1 Failure distribution of spare events

Let us consider a Spare gate with 2 input events – the primary event A and one spare event B – as shown in Fig. 5.1.

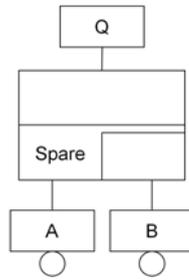


Figure 5.1: A single Spare gate with one primary event A and one spare event B

The failure distribution of the primary event A does not depend on B , so the Cdf and pdf of A are mere functions of time noted $F_A(t)$ and $f_A(t)$, respectively, as usual.

The failure distribution of the spare event B does not depend on A as long as B is dormant, so the Cdf and pdf of B_d are also mere functions of time noted $F_{B_d}(t)$ and $f_{B_d}(t)$, respectively. However, the failure distribution of the spare event B depends on A when B is active, since B becomes active at the failure date of A , which will be denoted as t_A . The Cdf and pdf of B_a hence depend both on time t and on the failure date of A (t_A). For the sake of clarity, we consider both functions as functions of the two variables t , and t_A , which will be noted $F_{B_a}(t, t_A)$ and $f_{B_a}(t, t_A)$, respectively.

Let us illustrate this aspect on the particular case of exponential distributions. If A has a failure rate λ_A , for all $t \geq 0$, its Cdf and pdf are

$$\begin{cases} F_A(t) &= 1 - e^{-\lambda_A t} \\ f_A(t) &= \lambda_A e^{-\lambda_A t} \end{cases}$$

In the same way, if B has a failure rate λ_B and a dormancy α , for all $t \geq 0$, the Cdf and pdf of B_d are

$$\begin{cases} F_{B_d}(t) &= 1 - e^{-\alpha \lambda_B t} \\ f_{B_d}(t) &= \alpha \lambda_B e^{-\alpha \lambda_B t} \end{cases}$$

Regarding the Cdf of B_a , it is known that:

- it is exponential with failure λ_B ;
- it is continuous with F_{B_d} at the failure date of A (t_A).

It can hence be assumed that $F_{B_a}(t, t_A) = 1 - e^{-\lambda_B(t-x(t_A))}$, where x is a function of t_A . By using the continuity of F_B at $t = t_A$, we have

$$\begin{aligned} F_{B_a}(t_A, t_A) &= F_{B_d}(t_A) \\ \Leftrightarrow 1 - e^{-\lambda_B(t_A-x(t_A))} &= 1 - e^{-\alpha\lambda_B t_A} \\ \Leftrightarrow \lambda_B(t_A - x(t_A)) &= \alpha\lambda_B t_A \\ \Leftrightarrow t_A - x(t_A) &= \alpha t_A \\ \Leftrightarrow x(t_A) &= (1 - \alpha)t_A \end{aligned}$$

As a consequence, for all $t \geq (1 - \alpha)t_A$,

$$\begin{cases} F_{B_a}(t, t_A) &= 1 - e^{-\lambda_B(t-(1-\alpha)t_A)} \\ f_{B_a}(t, t_A) &= \lambda_B e^{-\lambda_B(t-(1-\alpha)t_A)} \end{cases}$$

The notations used for the Cdf and pdf of the spare event B will be retained in the remainder of this dissertation, and they can be used for any spare event S by replacing t_A with the failure date of the event on which S depends, in the case of Spare gates with more than 2 input events.

5.4.2 Probabilistic model of Spare gates with 2 input events

In this section, we completely detail the probabilistic model of Spare gates with 2 input events in the configurations which were considered in Section 4.3.1. The different cases are treated in an increasing order of complexity, from a single Spare gate in Section 5.4.2.1 to 2 Spare gates sharing a spare event in Section 5.4.2.2, and even to the generalization to n Spare gates sharing a spare event in Section 5.4.2.3.

5.4.2.1 Probabilistic model of a single Spare gate

According to Section 4.3.1.1, the behavioural model of a single Spare gate with 2 input events is

$$Q = B_a \cdot (A \triangleleft B_a) + A \cdot (B_d \triangleleft A).$$

As B cannot be both in its dormant and active mode, $B_a \cdot (A \triangleleft B_a) \cdot A \cdot (B_d \triangleleft A) \stackrel{(3.62)}{=} B_a \cdot (A \triangleleft B_a) \cdot A \cdot B_d \cdot (B_d \triangleleft A) = \perp$, so the two algebraic terms $B_a \cdot (A \triangleleft B_a)$ and $A \cdot (B_d \triangleleft A)$ are disjoint and

$$Pr\{Q\}(t) = Pr\{B_a \cdot (A \triangleleft B_a)\}(t) + Pr\{A \cdot (B_d \triangleleft A)\}(t)$$

On the one hand, the Cdf and pdf of B_d do not depend on A , so the probability of occurrence of the second term – $Pr\{A \cdot (B_d \triangleleft A)\}(t)$ – can be determined by means of the expressions (5.1) as

$$Pr\{A \cdot (B_d \triangleleft A)\}(t) = \int_0^t f_A(u) F_{B_d}(u) du$$

On the other hand, the Cdf and pdf of B_a depend on the failure date of A , so $Pr\{B_a \cdot (A \triangleleft B_a)\}(t)$ cannot be determined by means of the expressions (5.1). If we respectively denote by T_A and T_{B_a} the failure dates of A and B_a , $Pr\{B_a \cdot (A \triangleleft B_a)\}(t)$ can be defined as

$$\begin{aligned} Pr\{B_a \cdot (A \triangleleft B_a)\}(t) &= Pr\{T_A \leq T_{B_a} \leq t\} \\ &= E\left[\mathbf{1}_{\{T_A \leq T_{B_a}\}} \mathbf{1}_{\{T_{B_a} \leq t\}}\right], \end{aligned}$$

where $\mathbf{1}$ is the *indicator function* [Grimmett 2001] defined as

$$\mathbf{1}_A(X) = \begin{cases} 1 & \text{if } X \in A \\ 0 & \text{if } X \notin A \end{cases}$$

and E is the *expectation value* [Grimmett 2001] defined as

$$E[\mathbf{1}_A(X)] = Pr\{X \in A\}$$

According to the *law of total expectation* [Billingsley 1995], if X is an integrable random variable and if Y is any random variable such that $E[E[X|Y]]$ has a meaning, the following relation holds:

$$E[X] = E[E[X|Y]]$$

As a consequence,

$$\begin{aligned} Pr\{B_a \cdot (A \triangleleft B_a)\}(t) &= \int_0^t \left(\int_v^t f_{T_{B_a}|T_A}(u|T_A=v) du \right) f_{T_A}(v) dv \\ &= \int_0^t \left(\int_v^t f_{B_a}(u,v) du \right) f_A(v) dv \end{aligned}$$

The probabilistic model of a single Spare gate with 2 input events hence is

$$F_Q(t) = Pr\{Q\}(t) = \int_0^t \left(\int_v^t f_{B_a}(u,v) du \right) f_A(v) dv + \int_0^t f_A(u) F_{B_d}(u) du$$

This probabilistic model does not depend on the failure distribution considered for basic events. However, in the particular case of exponential distributions, we show in Appendix B.1 that the result obtained with this probabilistic model is the same as the result obtained with Markov Chains.

5.4.2.2 Probabilistic model of 2 Spare gates sharing a spare event

According to section 4.3.1.2, the behavioural model of any of two Spare gates sharing a spare event is

$$\begin{cases} Q1 = C_a \cdot (A \triangleleft C_a) + A \cdot (C_d \triangleleft A) + A \cdot (B \triangleleft A) \\ Q2 = C_a \cdot (B \triangleleft C_a) + B \cdot (C_d \triangleleft B) + B \cdot (A \triangleleft B) \end{cases}$$

Let us first consider the expression for $Q1$:

$$Q1 = C_a \cdot (A \triangleleft C_a) + A \cdot (C_d \triangleleft A) + A \cdot (B \triangleleft A)$$

It can be noted that the two first algebraic terms $C_a \cdot (A \triangleleft C_a)$ and $A \cdot (C_d \triangleleft A)$ do not contain B while the third algebraic term $A \cdot (B \triangleleft A)$ does. These three algebraic terms are hence not disjoint. This expression for $Q1$ can be transformed into another equivalent expression containing disjoint terms only by introducing B in the two first algebraic terms.

The first algebraic term $C_a \cdot (A \triangleleft C_a)$ corresponds to the failure sequence $[A, C]$, which does not depend on B . B can fail before A (sequence $[B, A, C]$), between A and C (sequence $[A, B, C]$), after C (sequence $[A, C, B]$), or B may not fail at all (sequence $[A, C, \mathcal{B}]$ ¹). The algebraic term $C_a \cdot (A \triangleleft C_a)$ is hence equivalent to

$$\begin{aligned} C_a \cdot (A \triangleleft C_a) &= C_a \cdot (B \triangleleft A) \cdot (A \triangleleft C_a) + C_a \cdot (A \triangleleft B) \cdot (B \triangleleft C_a) \\ &\quad + B \cdot (A \triangleleft C_a) \cdot (C_a \triangleleft B) + C_a \cdot (A \triangleleft C_a) \cdot \mathcal{B} \end{aligned}$$

where the four terms represent the four possible sequences obtained by including B in the sequence $[A, C]$.

The second algebraic term $A \cdot (C_d \triangleleft A)$ corresponds to the failure sequence $[C, A]$, which does not depend on B either. B can fail before C (sequence $[B, C, A]$), between C and A (sequence $[C, B, A]$), after A (sequence $[C, A, B]$), or B may not fail at all (sequence $[C, A, \mathcal{B}]$). However, if B fails before C , C will become active, which is impossible since C is dormant in the term $A \cdot (C_d \triangleleft A)$. The algebraic term $A \cdot (C_d \triangleleft A)$ is hence equivalent to

$$A \cdot (C_d \triangleleft A) = A \cdot (C_d \triangleleft B) \cdot (B \triangleleft A) + B \cdot (C_d \triangleleft A) \cdot (A \triangleleft B) + A \cdot (C_d \triangleleft A) \cdot \mathcal{B}$$

where the three terms represent the three possible sequences obtained by including B in the sequence $[C, A]$.

The behavioural model of the gate hence becomes

$$\begin{aligned} Q1 &= C_a \cdot (B \triangleleft A) \cdot (A \triangleleft C_a) + C_a \cdot (A \triangleleft B) \cdot (B \triangleleft C_a) \\ &\quad + B \cdot (A \triangleleft C_a) \cdot (C_a \triangleleft B) + C_a \cdot (A \triangleleft C_a) \cdot \mathcal{B} \\ &\quad + A \cdot (C_d \triangleleft B) \cdot (B \triangleleft A) + B \cdot (C_d \triangleleft A) \cdot (A \triangleleft B) \\ &\quad + A \cdot (C_d \triangleleft A) \cdot \mathcal{B} + A \cdot (B \triangleleft A) \end{aligned}$$

¹ \mathcal{B} denotes the fact that B does not appear at all.

and can be transformed to

$$\begin{aligned}
Q1 &\stackrel{(3.33)}{=} A \cdot C_a \cdot (B \triangleleft A) \cdot (A \triangleleft C_a) + C_a \cdot (A \triangleleft B) \cdot (B \triangleleft C_a) \\
&\quad + B \cdot (A \triangleleft C_a) \cdot (C_a \triangleleft B) + C_a \cdot (A \triangleleft C_a) \cdot \mathcal{B} \\
&\quad + A \cdot (C_d \triangleleft B) \cdot (B \triangleleft A) + B \cdot (C_d \triangleleft A) \cdot (A \triangleleft B) \\
&\quad + A \cdot (C_d \triangleleft A) \cdot \mathcal{B} + A \cdot (B \triangleleft A),
\end{aligned}$$

in which the terms $A \cdot C_a \cdot (B \triangleleft A) \cdot (A \triangleleft C_a)$ and $A \cdot (C_d \triangleleft B) \cdot (B \triangleleft A)$ can be absorbed by the term $A \cdot (B \triangleleft A)$, thus leading to the following simplified expression

$$\begin{aligned}
Q1 &= C_a \cdot (A \triangleleft B) \cdot (B \triangleleft C_a) + B \cdot (A \triangleleft C_a) \cdot (C_a \triangleleft B) \\
&\quad + C_a \cdot (A \triangleleft C_a) \cdot \mathcal{B} + B \cdot (C_d \triangleleft A) \cdot (A \triangleleft B) \\
&\quad + A \cdot (C_d \triangleleft A) \cdot \mathcal{B} + A \cdot (B \triangleleft A).
\end{aligned}$$

It can be noted that all the terms of this expression are disjoint. The failure probability of $Q1$ can hence be expressed as

$$\begin{aligned}
Pr\{Q1\}(t) &= Pr\{C_a \cdot (A \triangleleft B) \cdot (B \triangleleft C_a)\}(t) \\
&\quad + Pr\{B \cdot (A \triangleleft C_a) \cdot (C_a \triangleleft B)\}(t) \\
&\quad + Pr\{C_a \cdot (A \triangleleft C_a) \cdot \mathcal{B}\}(t) + Pr\{B \cdot (C_d \triangleleft A) \cdot (A \triangleleft B)\}(t) \\
&\quad + Pr\{A \cdot (C_d \triangleleft A) \cdot \mathcal{B}\}(t) + Pr\{A \cdot (B \triangleleft A)\}(t)
\end{aligned}$$

By using the same approach as in Section 5.4.2.1, these six probabilities can be expressed under a form which does not depend on the failure distribution considered for basic events:

$$\begin{aligned}
Pr\{C_a \cdot (A \triangleleft B) \cdot (B \triangleleft C_a)\}(t) &= \int_0^t \left(\int_w^t \left(\int_w^u f_B(v)dv \right) f_{C_a}(u, w)du \right) f_A(w)dw \\
Pr\{B \cdot (A \triangleleft C_a) \cdot (C_a \triangleleft B)\}(t) &= \int_0^t \left(\int_0^u \left(\int_v^u f_{C_a}(w, v)dw \right) f_A(v)dv \right) f_B(u)du \\
Pr\{C_a \cdot (A \triangleleft C_a) \cdot \mathcal{B}\}(t) &= (1 - F_B(t)) \int_0^t \left(\int_v^t f_{C_a}(u, v)du \right) f_A(v)dv \\
Pr\{B \cdot (C_d \triangleleft A) \cdot (A \triangleleft B)\}(t) &= \int_0^t \left(\int_0^u f_A(v)F_{C_d}(v)dv \right) f_B(u)du \\
Pr\{A \cdot (C_d \triangleleft A) \cdot \mathcal{B}\}(t) &= (1 - F_B(t)) \int_0^t f_A(u)F_{C_d}(u)du \\
Pr\{A \cdot (B \triangleleft A)\}(t) &= \int_0^t f_A(u)F_B(u)du
\end{aligned}$$

The probabilistic model for $Q1$ can hence be deduced from these expressions, and the probabilistic model for $Q2$ can be determined by symmetry:

$$\begin{aligned}
F_{Q1}(t) = Pr\{Q1\}(t) &= \int_0^t \left(\int_w^t \left(\int_w^u f_B(v)dv \right) f_{C_a}(u, w)du \right) f_A(w)dw \\
&+ \int_0^t \left(\int_0^u \left(\int_v^u f_{C_a}(w, v)dw \right) f_A(v)dv \right) f_B(u)du \\
&+ (1 - F_B(t)) \int_0^t \left(\int_v^t f_{C_a}(u, v)du \right) f_A(v)dv \\
&+ \int_0^t \left(\int_0^u f_A(v)F_{C_d}(v)dv \right) f_B(u)du \\
&+ (1 - F_B(t)) \int_0^t f_A(u)F_{C_d}(u)du + \int_0^t f_A(u)F_B(u)du \\
F_{Q2}(t) = Pr\{Q2\}(t) &= \int_0^t \left(\int_w^t \left(\int_w^u f_A(v)dv \right) f_{C_a}(u, w)du \right) f_B(w)dw \\
&+ \int_0^t \left(\int_0^u \left(\int_v^u f_{C_a}(w, v)dw \right) f_B(v)dv \right) f_A(u)du \\
&+ (1 - F_A(t)) \int_0^t \left(\int_v^t f_{C_a}(u, v)du \right) f_B(v)dv \\
&+ \int_0^t \left(\int_0^u f_B(v)F_{C_d}(v)dv \right) f_A(u)du \\
&+ (1 - F_A(t)) \int_0^t f_B(u)F_{C_d}(u)du + \int_0^t f_B(u)F_A(u)du
\end{aligned}$$

This probabilistic model does not depend on the failure distribution considered for basic events. However, in the particular case of exponential distributions, we show in Appendix B.2 that the result obtained with this probabilistic model is the same as the result obtained with Markov Chains.

5.4.2.3 Probabilistic model of n Spare gates sharing a spare event

According to section 4.3.1.3, the behavioural model of n Spare gates sharing a Spare event is

$$Q_i = S_a \cdot (P_i \triangleleft S_a) + P_i \cdot (S_d \triangleleft P_i) + \sum_{j \neq i} P_i \cdot (P_j \triangleleft P_i)$$

It can be noted that this expression contains $(n + 1)$ algebraic terms which are not disjoint. This expression can be transformed to a sum of disjoint terms in the same way as it was done in Section 5.4.2.2, and the probability of occurrence of each one of these disjoint terms can then be determined as an expression which does not depend on the failure distribution of events. The probabilistic model of each one of the n Spare gates can then be computed from these expressions.

Even though it has been determined, the general expression of these failure probabilities will not be detailed in this dissertation.

5.4.3 Probabilistic model of Spare gates with 3 input events

According to Section 4.3.2, the behavioural model of a single Spare gate with 3 input events is

$$\begin{aligned} Q = & C_a \cdot (A \triangleleft B_a) \cdot (B_a \triangleleft C_a) + B_a \cdot (A \triangleleft C_d) \cdot (C_d \triangleleft B_a) \\ & + C_a \cdot (B_d \triangleleft A) \cdot (A \triangleleft C_a) + A \cdot (B_d \triangleleft C_d) \cdot (C_d \triangleleft A) \\ & + B_a \cdot (C_d \triangleleft A) \cdot (A \triangleleft B_a) + A \cdot (C_d \triangleleft B_d) \cdot (B_d \triangleleft A) \end{aligned}$$

It can be noted that the six terms of this expression are disjoint as B and C cannot fail both in their dormant and active mode. The failure probability of Q can hence be expressed as

$$\begin{aligned} Pr \{Q\}(t) = & Pr \{C_a \cdot (A \triangleleft B_a) \cdot (B_a \triangleleft C_a)\}(t) \\ & + Pr \{B_a \cdot (A \triangleleft C_d) \cdot (C_d \triangleleft B_a)\}(t) \\ & + Pr \{C_a \cdot (B_d \triangleleft A) \cdot (A \triangleleft C_a)\}(t) \\ & + Pr \{A \cdot (B_d \triangleleft C_d) \cdot (C_d \triangleleft A)\}(t) \\ & + Pr \{B_a \cdot (C_d \triangleleft A) \cdot (A \triangleleft B_a)\}(t) \\ & + Pr \{A \cdot (C_d \triangleleft B_d) \cdot (B_d \triangleleft A)\}(t) \end{aligned}$$

By using the same approach as in Section 5.4.2.1, these six probabilities can be expressed under a form which does not depend on the failure distribution considered for basic events:

$$\begin{aligned} Pr \{C_a \cdot (A \triangleleft B_a) \cdot (B_a \triangleleft C_a)\}(t) &= \int_0^t \left(\int_u^t \left(\int_v^t f_{C_a}(w, v) dw \right) f_{B_a}(v, u) dv \right) f_A(u) du \\ Pr \{B_a \cdot (A \triangleleft C_d) \cdot (C_d \triangleleft B_a)\}(t) &= \int_0^t \left(\int_w^t \left(\int_w^u f_{C_d}(v) dv \right) f_{B_a}(u, w) du \right) f_A(w) dw \\ Pr \{C_a \cdot (B_d \triangleleft A) \cdot (A \triangleleft C_a)\}(t) &= \int_0^t \left(\int_u^t f_{C_a}(v, u) dv \right) f_A(u) F_{B_d}(u) du \\ Pr \{A \cdot (B_d \triangleleft C_d) \cdot (C_d \triangleleft A)\}(t) &= \int_0^t \left(\int_0^u f_{C_d}(v) F_{B_d}(v) dv \right) f_A(u) du \\ Pr \{B_a \cdot (C_d \triangleleft A) \cdot (A \triangleleft B_a)\}(t) &= \int_0^t \left(\int_u^t f_{B_a}(v, u) dv \right) f_A(u) F_{C_d}(u) du \\ Pr \{A \cdot (C_d \triangleleft B_d) \cdot (B_d \triangleleft A)\}(t) &= \int_0^t \left(\int_0^u f_{B_d}(v) F_{C_d}(v) dv \right) f_A(u) du \end{aligned}$$

The probabilistic model of a single Spare gate with 3 input events can hence be

deduced from these expressions:

$$\begin{aligned}
F_Q(t) = Pr\{Q\}(t) = & \int_0^t \left(\int_u^t \left(\int_v^t f_{C_a}(w, v) dw \right) f_{B_a}(v, u) dv \right) f_A(u) du \\
& + \int_0^t \left(\int_w^t \left(\int_w^u f_{C_d}(v) dv \right) f_{B_a}(u, w) du \right) f_A(w) dw \\
& + \int_0^t \left(\int_u^t f_{C_a}(v, u) dv \right) f_A(u) F_{B_d}(u) du \\
& + \int_0^t \left(\int_0^u f_{C_d}(v) F_{B_d}(v) dv \right) f_A(u) du \\
& + \int_0^t \left(\int_u^t f_{B_a}(v, u) dv \right) f_A(u) F_{C_d}(u) du \\
& + \int_0^t \left(\int_0^u f_{B_d}(v) F_{C_d}(v) dv \right) f_A(u) du
\end{aligned}$$

This probabilistic model does not depend on the failure distribution considered for basic events. However, in the particular case of exponential distributions, we show in Appendix B.3 that the result obtained with this probabilistic model is the same as the result obtained with Markov Chains.

The probabilistic model of n Spare gates with 3 input events sharing 2 spare events could be determined by using the same approach as in Sections 5.4.2.2 and 5.4.2.3. Even though it has been determined, it will not be detailed in this dissertation.

5.4.4 Probabilistic model of Spare gates with n input events

In the same way as the behavioural model of Spare gates with 2 and 3 input events could be generalized to the case of Spare gates with n input events, the probabilistic models presented in Sections 5.4.2 and 5.4.3 can be generalized to the case of Spare gates with n input events.

As was mentioned in Section 4.3.3, the behavioural model of a single Spare gate with n input events can be obtained by considering the $n!$ different failure sequences of basic events, and by denoting the dormant and active mode of each spare event by 2 substituted variables. Given each spare event cannot be both in its dormant and active mode, it can be noted that the $n!$ algebraic terms of the behavioural model of a single Spare gate with n input events are disjoint, since these $n!$ algebraic terms precisely represent the $n!$ possible failure sequences of the n input events of the Spare gate which are, by definition, disjoint. The probability of occurrence of each term can then be determined whatever the failure distribution considered for basic events, and the probabilistic model of the Spare gate can hence be determined.

In the same way, the probabilistic model of p Spare gates with n input events sharing $(n - 1)$ spare events can be determined by using the same approach as in Sections 5.4.2.2 and 5.4.2.3.

As the number of algebraic terms grows exponentially with the number of input events of Spare gates, these probabilistic models will not be detailed in this dissertation.

Analysis of Dynamic Fault Trees

Contents

6.1	Dynamic FTA based on the structure function	64
6.1.1	Minimal canonical form of the structure function of Dynamic Fault Trees	64
6.1.2	Qualitative analysis of DFTs based on the structure function	67
6.1.3	Quantitative analysis of DFTs based on the structure function	68
6.2	Application examples	69
6.2.1	The Hypothetical Example Computer System (HECS)	70
6.2.2	The Hypothetical Cardiac Assist System (HCAS)	71
6.3	Dynamic FTA of the HECS	72
6.3.1	Minimal canonical form of the structure function of the HECS	72
6.3.2	Qualitative analysis of the HECS	76
6.3.3	Quantitative analysis of the HECS	78
6.4	Dynamic FTA of the HCAS	80
6.4.1	Minimal canonical form of the structure function of the HCAS	80
6.4.2	Qualitative analysis of the HCAS	85
6.4.3	Quantitative analysis of the HCAS	87

This chapter shows how the structure function of DFTs can be determined from the behavioural model of dynamic gates determined in Chapter 4, and how both the qualitative and quantitative analysis of DFTs can be performed directly from this structure function thanks to the probabilistic model of dynamic gates which was presented in Chapter 5. The general methodology is detailed in section 6.1, and it is illustrated on two DFTs examples from the literature which are presented in Section 6.2. The determination of the structure function of the two DFT examples and their analysis are presented in Sections 6.3 and 6.4.

6.1 Dynamic FTA based on the structure function

This section details the general methodology allowing to perform the analysis of DFTs from their structure function. Section 6.1.1 shows how the structure function of any DFT can be determined from the behavioural model of dynamic gates determined in Chapter 4, and how this structure function can be simplified to a minimal canonical form thanks to the theorems presented in Chapter 3 and to a minimization algorithm. Section 6.1.2 shows how the minimal cut sequences of the DFT can be extracted from this minimal canonical form of the structure function, and Section 6.1.3 explains how the quantitative analysis of DFTs can be performed directly from this minimal canonical form thanks to the probabilistic model of dynamic gates which was presented in Chapter 5.

6.1.1 Minimal canonical form of the structure function of Dynamic Fault Trees

Section 6.1.1.1 shows how to determine the structure function of any DFT. This structure function can be simplified to a canonical form, as presented in Section 6.1.1.2, and then minimized thanks to a minimization algorithm presented in Section 6.1.1.3.

6.1.1.1 Determination of the structure function

The behavioural models of dynamic gates presented in Chapter 4 allow us to determine the structure function of any DFT as a function of basic events – some of which may be spare events in their active and dormant modes – that can be repeated without restrictions.

Given a DFT with n basic events $\{b_i, i \in (1, \dots, n)\}$, the structure function for the TE becomes an expression containing at most the n basic events (some of which may be spare events which can be split into two substitution variables to take into account their active and dormant mode), and operators $+$, \cdot , \triangleleft , \triangle , and \trianglelefteq .

6.1.1.2 Canonical form of the structure function

The structure function can then be developed and simplified, thanks to the theorems presented in Section 3.5, to reach a standardized sum-of-product *canonical form* where each product term contains operator \cdot , and ordered pairs of variables linked by operator \triangleleft only. The steps to be followed to reach the *canonical form* are:

1. Starting from the TE , in a top down fashion, replace each dynamic gate by its behavioural model from Chapter 4.
2. Eliminate the parenthesis by applying distributivity theorems, such as theorems (3.23) to (3.30), and (3.52) to (3.59).

3. The structure function is then expressed in a sum of product terms as in (6.1):

$$TE = \sum \left(\prod_i b_i \cdot \prod_{j,k} (b_j \leq b_k) \cdot \prod_{l,m} (b_l \triangleleft b_m) \cdot \prod_{o,p} (b_o \triangle b_p) \right). \quad (6.1)$$

4. Because b_o and b_p are basic events, in virtue of theorem (3.15), (6.1) can always be simplified to the form

$$TE = \sum \left(\prod b_i \cdot \prod (b_j \leq b_k) \cdot \prod (b_l \triangleleft b_m) \right). \quad (6.2)$$

5. Because b_j and b_k are basic events, taking into account theorems (3.15) and (3.16), we can write $b_j \leq b_k = b_j \triangleleft b_k$. Hence, the expression in (6.2) becomes

$$TE = \sum \left(\prod b_i \cdot \prod (b_j \triangleleft b_k) \right).$$

6. According to theorem (3.22), $j = k \Rightarrow b_j \triangleleft b_k = \perp$, then the structure function can be simplified to

$$TE = \sum \left(\prod b_i \cdot \prod (b_j \triangleleft b_k) \right), j \neq k.$$

7. Finally, according to theorem (3.33), $i = j \Rightarrow b_i \cdot (b_j \triangleleft b_k) = b_j \triangleleft b_k$, so we get the structure function in *canonical form*

$$TE = \sum \left(\prod b_i \cdot \prod (b_j \triangleleft b_k) \right), j \notin \{i, k\}. \quad (6.3)$$

6.1.1.3 Minimal canonical form of the structure function

In the case of SFTs, a minimal form of the structure function can be determined easily thanks to the theorems of Boolean algebra, or by resorting to BDDs [Rauzy 2001], [Stamatelatos 2002]. Such minimal form provides the minimal cut sets of the SFT. In the case of DFTs, the concept of minimal cut must be refined to *minimal cut sequence* [Tang 2004], representing the minimal (ordered) failure sequence of events that causes the occurrence of the TE . The exhaustive search of the minimal cut sequences of a DFT is an open problem, in the general case. The algebraic approach for DFTs provides a sound theoretical basis for the determination of cut sequences.

In the canonical form of the structure function given in (6.3), each product term $\prod b_i \cdot \prod (b_j \triangleleft b_k)$ is not a single cut sequence, but an algebraic expression providing a sufficient condition on the order of basic event failures that leads to the TE which may contain more than one cut sequence, and actually is a *cut sequence set (CSS)*. In the remainder of this dissertation, CSS_i will represent both a set of cut sequences (like in (6.5)), and the algebraic expression that characterizes this set of cut sequences (like in (6.4)).

If there are n product terms in (6.3), the canonical form can be rewritten in the compact form

$$TE = \sum_{i=1}^n CSS_i. \quad (6.4)$$

The set \mathbb{S} of all the cut sequences of the DFT is the union of all the CSSs previously defined:

$$\mathbb{S} = \bigcup_{i=1}^n CSS_i. \quad (6.5)$$

Nevertheless, a CSS may be included in one or more CSSs and hence be redundant. To remove these redundant terms from the structure function, we use a minimization criterion which was presented in [Rauzy 2001] in the case of SFTs. In [Rauzy 2001], a cut set π is minimal if there is no cut set ρ such that $\rho \subset \pi$. We extend this concept to DFTs, and to the CSSs: CSS_j is minimal if there is no CSS_i such that $CSS_i \subset CSS_j$. If such a CSS_i exists, then we estimate that CSS_i is redundant and that it can be removed from the structure function.

From a mathematical point of view, if we consider two sets A and B , $A \subset B \Rightarrow A \cap B = A$. In the same way, $CSS_i \subset CSS_j \Rightarrow CSS_i \cdot CSS_j = CSS_i$. The generalization to all the terms of the structure function provides the following minimization criterion: CSS_i is included in one of the CSS_j if it satisfies the criterion

$$CSS_i \cdot \sum_{j \neq i} CSS_j = CSS_i. \quad (6.6)$$

If CSS_i is included in one of the CSS_j , it is redundant, and can be removed from the structure function (6.4). Iterative application of the criterion (6.6), according to Algorithm 1, removes all the redundant CSSs, and returns the minimal set \mathbb{S}_{min} of non-redundant CSSs.

Algorithm 1 Algorithm for the minimization of the canonical form of the structure function of a DFT

Require: \mathbb{S}

$\mathbb{S}_{min} \leftarrow \mathbb{S}$

for $i = 1$ to n **do**

$CSS \leftarrow \sum_{j \neq i} CSS_j, CSS_j \in \mathbb{S}_{min}$

if $CSS_i \cdot CSS = CSS_i$ **then**

$\mathbb{S}_{min} \leftarrow \mathbb{S}_{min} \setminus \{CSS_i\}$

return \mathbb{S}_{min}

If \mathbb{S}_{min} contains ($m \leq n$) cut sequence sets, the minimal canonical form of the structure function can be expressed as

$$TE = \sum_{i=1}^m CSS_i. \quad (6.7)$$

It can be noted that this minimal canonical form of the structure function is not unique, as there will be as many minimal canonical forms of the structure function as there are minimization criteria. The study of the minimization of the structure function implies many different criteria and its complexity is beyond the scope of this PhD thesis, so this topic will be addressed in future works. Besides, as the minimal canonical form of the structure function is an algebraic relation which is, by definition, not unique, a given set of minimal cut sequences may be represented by two – or more – different structure functions which can be minimal. For instance, in a DFT with 3 basic events A , B , and C , the algebraic term $(A \triangleleft B) \cdot (A \triangleleft C)$ is minimal and corresponds to the two minimal cut sequences $[A, B, C]$ and $[A, C, B]$. These two minimal cut sequences respectively correspond to the two algebraic terms $C \cdot (A \triangleleft B) \cdot (B \triangleleft C)$ and $B \cdot (A \triangleleft C) \cdot (C \triangleleft B)$, which are not redundant, so

$$(A \triangleleft B) \cdot (A \triangleleft C) = C \cdot (A \triangleleft B) \cdot (B \triangleleft C) + B \cdot (A \triangleleft C) \cdot (C \triangleleft B)$$

and these two minimal cut sequences can be contained in two different minimal canonical forms of structure functions.

6.1.2 Qualitative analysis of DFTs based on the structure function

The purpose of the qualitative analysis of SFTs is to determine the minimal cut sets of the FT, which represent the minimal combinations of basic events which will cause the system to fail and which cannot be reduced in number. However, in the case of DFTs, the concept of minimal cut must be refined to *minimal cut sequence* [Tang 2004], representing the minimal (ordered) failure sequence of events that causes the occurrence of the TE .

According to Section 6.1.1.3, the minimal canonical form of the structure function of any DFT can be obtained as

$$TE = \sum \left(\prod b_i \cdot \prod (b_j \triangleleft b_k) \right), j \notin \{i, k\},$$

where each product term is a non-redundant term thanks to the use of Algorithm 1.

The qualitative analysis of DFTs can be performed directly from this structure function. This canonical form of the structure function is a sum-of-product form, so each product term of the structure function can provide *minimal cut sets* or *minimal cut sequences* for the DFT. Two cases may happen:

- if a product term does not contain the temporal operator BF (\triangleleft), it is static and provides a *minimal cut set* for the DFT;
- if a product term contains the temporal operator BF (\triangleleft), it is dynamic and provides *minimal cut sequences* for the DFT. In some cases, a set of minimal cut sequences may represent all the possible sequences which correspond to a minimal cut set and can hence be reduced to this minimal cut set.

This canonical form of the structure function thus provides an hybrid result for the qualitative analysis of DFTs by allowing to determine both minimal cut sets and minimal cut sequences. Furthermore, as it was said above, a set of minimal cut sequences may sometimes be equivalent to a single minimal cut set. Two cases may happen:

- if these minimal cut sequences contain spare events, they must not be reduced to their equivalent minimal cut set since the knowledge of the state in which spare events fail will be needed to perform the quantitative analysis of the DFT;
- if these minimal cut sequences do not contain spare events, they can be reduced to their equivalent minimal cut set. Indeed, even though both results are equivalent, minimal cut sets represent a more concise – and hence more useful – result to the practitioner than the corresponding set of minimal cut sequences.

In the remainder of this dissertation, failure sequences will be noted between brackets. For instance, the failure sequence in which A fails before B will be noted $[A, B]$. The minimal cut sequences can be extracted quite easily from the structure function of DFTs. Let us consider a DFT with 3 basic events A , B , C and whose structure function contains the term $B \cdot (A \triangleleft B)$. The TE of the DFT will hence appear if A fails before B , so the sequence $[A, B]$ is a cut sequence of the DFT. However, any sequence of failures of the 3 basic events in which A fails before B is a cut sequence too, so the sequences $[C, A, B]$, $[A, C, B]$, and $[A, B, C]$ also are cut sequences for the DFT. However, as these 3 sequences represent all the possible inclusions of C in the sequence $[A, B]$, $[A, B]$ can be considered as minimal with regard to the 3 other sequences. The algebraic term $B \cdot (A \triangleleft B)$ of the structure function hence provides the minimal cut sequence $[A, B]$ for the DFT.

As explained above, it may happen that a set of minimal cut sequences represents all the possible sequences which correspond to a minimal cut set and can hence be reduced to this minimal cut set. For instance, the six minimal cut sequences $[A, B, C]$, $[A, C, B]$, $[B, A, C]$, $[B, C, A]$, $[C, A, B]$, and $[C, B, A]$ represent the six possible failure sequences of A , B , and C , and are hence equivalent to the minimal cut set $A \cdot B \cdot C$.

6.1.3 Quantitative analysis of DFTs based on the structure function

The purpose of quantitative analysis is to calculate the probability of occurrence of the TE from the probability of occurrence of the basic events [Henley 2000, Shooman 1990].

According to Section 6.1.1.3, the minimal canonical form of the structure function of any DFT can be obtained as

$$TE = \sum \left(\prod b_i \cdot \prod (b_j \triangleleft b_k) \right), j \notin \{i, k\}.$$

If this minimal canonical form has m CSSs, we can compute the probability of the TE by resorting to the standard inclusion–exclusion formula [Trivedi 2001]:

$$\begin{aligned}
Pr\{TE\} &= Pr\{CSS_1 + CSS_2 + \dots + CSS_m\} \\
&= \sum_{1 \leq i \leq m} Pr\{CSS_i\} - \sum_{1 \leq i < j \leq m} Pr\{CSS_i \cdot CSS_j\} \\
&\quad + \sum_{1 \leq i < j < k \leq m} Pr\{CSS_i \cdot CSS_j \cdot CSS_k\} \\
&\quad + \dots + (-1)^{m-1} Pr\{CSS_1 \cdot CSS_2 \cdot \dots \cdot CSS_m\} \quad (6.8)
\end{aligned}$$

with $\forall i \in \{1, \dots, m\}, CSS_i \in \mathbb{S}_{min}$.

Each term of these sums contains the product of the algebraic expressions verified by the CSSs that can share the same basic events, and thus are not statistically independent. However, in these product terms, some simplifications might be possible in three cases:

- if a basic component b_i or a term $(b_i \triangleleft b_j)$ appear in two or more CSSs, we can apply the idempotence theorem $b_i \cdot b_i = b_i$ or $(b_i \triangleleft b_j) \cdot (b_i \triangleleft b_j) = (b_i \triangleleft b_j)$;
- if CSS_ℓ contains the term b_i , and CSS'_ℓ the term $(b_i \triangleleft b_j)$, by virtue of (3.33), $b_i \cdot (b_i \triangleleft b_j) = (b_i \triangleleft b_j)$; and
- if CSS_ℓ contains the term $(b_i \triangleleft b_j)$, and CSS'_ℓ the term $(b_j \triangleleft b_i)$, by virtue of (3.17), $(b_i \triangleleft b_j) \cdot (b_j \triangleleft b_i) = \perp$.

As soon as the simplification of the different terms has been performed, their failure probabilities can be calculated thanks to the expressions presented in Section 5.1 and to the probabilistic model of dynamic gates presented in Sections 5.2 to 5.4.

6.2 Application examples

This section presents the two application examples that we chose to illustrate the approach presented in Section 6.1. First, these 2 examples are presented and analyzed in an increasing order of complexity: the HECS example contains dynamic gates FDEP and CSP only, whereas the HCAS example contains all the dynamic gates (PAND, FDEP, WSP, and CSP). In both cases, CSP gates have a common spare event, so both examples allow to show how our approach addresses this complexity. Besides, the second example is quite specific since it contains an additional complexity: the input events of its WSP gate are dependent basic events of an FDEP gate and are hence not statistically independent. Finally, both examples are examples of the literature which allow to compare our results with the results from the literature.

6.2.1 The Hypothetical Example Computer System (HECS)

The first DFT example that is going to be used is the DFT of an Hypothetical Example Computer System (HECS) from [Stamatelatos 2002] which is shown in Fig. 6.1.

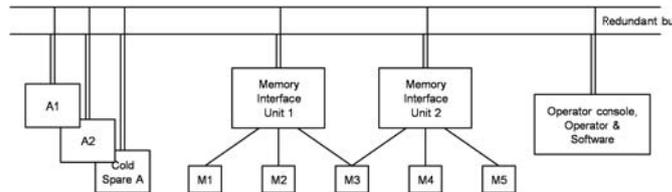


Figure 6.1: The Hypothetical Example Computer System (HECS)

The HECS includes dual-redundant processors $A1$ and $A2$ and a cold spare processor A , which can replace either upon failure. Processors $A1$, $A2$, and A are all identical processors, running the same operating system. The processors are dual-redundant and use comparison monitoring for fault tolerance. Periodic checkpoints are taken and stored to aid in recovery from errors. Comparison of results and checkpoints are used to detect errors. When one of the two active processors is determined to have failed, the cold spare processor takes its place. The system can continue to operate until all three processors have failed.

The HECS also includes five memory units of which three are required. These memory units are connected to the redundant bus via two memory interface units. If a memory interface unit fails, the memory units connected to it are unusable. Memory unit 3 ($M3$) is connected to both interfaces for redundancy; thus $M3$ is accessible as long as either interface unit is operational. A memory interface unit must hence be operational in order for the memory units which are connected to it to be accessible, thus the memory units are functionnally dependent on the interfaces.

There are two identical redundant buses, of which one is required for system operation. Thus the bus subsystem fails when both of the buses fail.

The last subsystem to be considered is the application subsystem. The application software runs on the computer system. The operator is a human who interfaces with the computer via a Graphical User Interface (GUI) that runs on an interface device. Thus an application (software (SW)) failure, GUI (hardware (HW)) failure or human operator error will lead to system failure.

The HECS requires the correct operation of the processing, memory, and bus subsystems, as well as the software application. Thus the HECS will fail if any of these subsystems fail. The DFT which models the potential failure of the HECS is shown in Fig. 6.2.

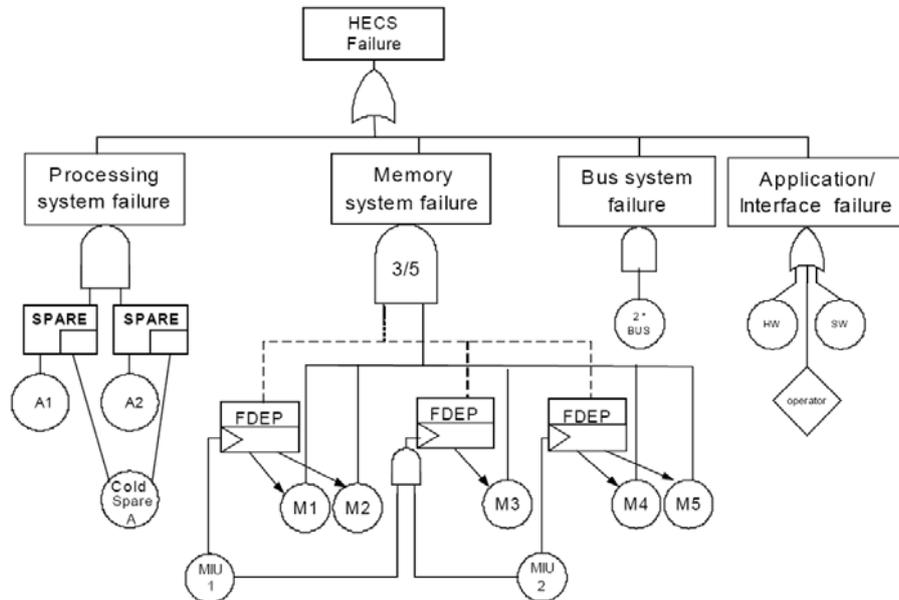


Figure 6.2: The DFT of the HECS

6.2.2 The Hypothetical Cardiac Assist System (HCAS)

The other DFT example that is going to be used is the DFT of an Hypothetical Cardiac Assist System (HCAS) from [Boudali 2005a] which was inspired from a Cardiac Assist System found in [Vemuri 1999], and whose structure can be found in [Ren 1998] and is shown in Fig. 6.3, where TEDTS stands for Transcutaneous Energy and Data Transmission System.

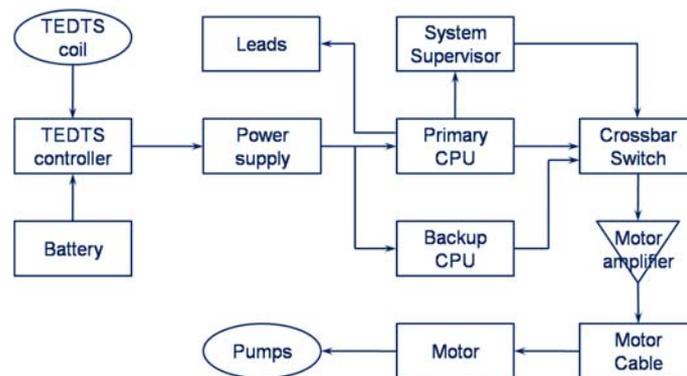


Figure 6.3: The Hypothetical Cardiac Assist System (HCAS)

The HCAS is designed to treat mechanical and electrical failures of the heart. The system can be divided into 4 modules: Trigger, CPU unit, motor section, and pumps. The crossbar switch (CS) and the system supervisor (SS) represent the

Trigger, since the failure of either CS or SS triggers the failure of both CPUs. The CPU unit can be considered as a warm spare with a primary P and a spare unit B (which corresponds to the backup CPU). For the motor section to function, either the motor (MOTOR) or the motor cable (MOTORC) need to be working. The pumps unit is comprised of two cold spares, each having a primary pump (PUMP_1 and PUMP_2), and sharing a common spare pump (Backup_PUMP). In order for the pumps unit to fail, all three pumps need to fail and CSP_1 needs to fail before (or at the same time as) CSP_2.

The DFT which models the potential failure of the HCAS is shown in Fig. 6.4.

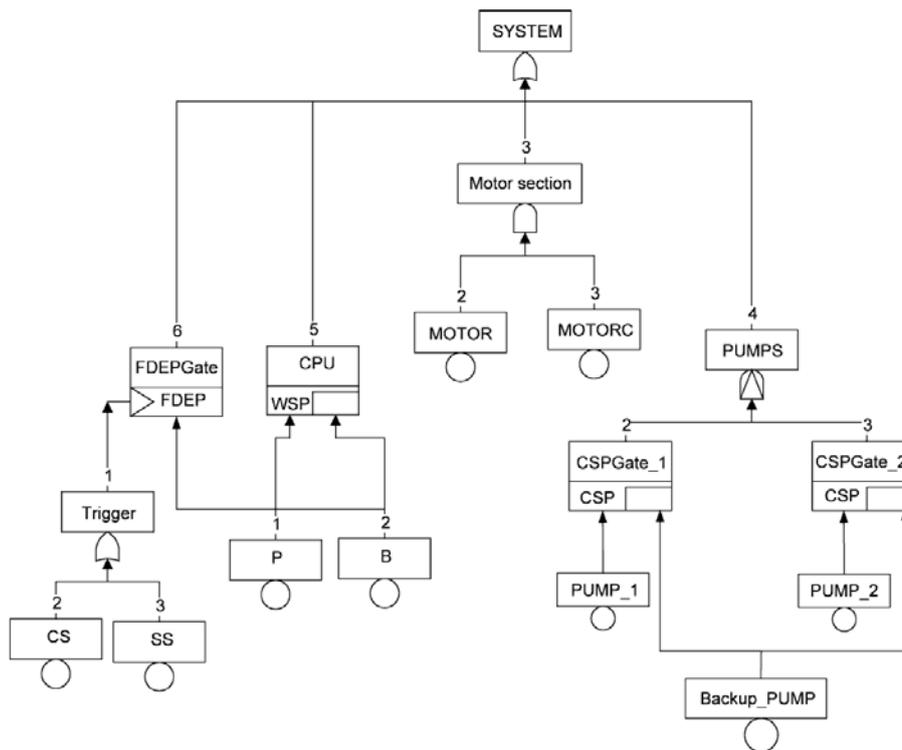


Figure 6.4: The DFT of the HCAS

6.3 Dynamic FTA of the HECS

6.3.1 Minimal canonical form of the structure function of the HECS

Let us determine the structure function of the DFT of the HECS in Fig. 6.2. This DFT can be divided into 4 independent subtrees, whose structure functions will be successively determined, by using the modularization algorithm presented in [Dutuit 1996]. These 4 subtrees are as follows:

- subtree 1 corresponds to the processing system failure. It contains one AND

gate and two Spare gates, and is hence dynamic. Its TE will be denoted by TE_1 ;

- subtree 2 corresponds to the memory system failure. It contains one 3-out-of-5 gate, one AND gate, and five FDEP gates, and is hence dynamic. Its TE will be denoted by TE_2 ;
- subtree 3 corresponds to the bus system failure. It contains a single AND gate and is hence static. Its TE will be denoted by TE_3 ;
- subtree 4 corresponds to the application/interface failure. It contains a single OR gate and is hence static. Its TE will be denoted by TE_4 .

The structure function of the DFT of the HECS can hence be expressed as

$$TE = TE_1 + TE_2 + TE_3 + TE_4.$$

6.3.1.1 Determination of the structure function of subtrees 3 and 4

The structure functions of subtrees 2 and 3 can be determined easily since both subtrees are static. Indeed, if both buses are denoted by $BUS1$ and $BUS2$, and if the operator is denoted by OP ,

$$\begin{aligned} TE_3 &= BUS1 \cdot BUS2 \\ TE_4 &= HW + SW + OP \end{aligned}$$

6.3.1.2 Determination of the structure function of subtree 2

According to the behavioural model of gate FDEP presented in Section 4.2, the structure function of subtree 2 can be determined as

$$\begin{aligned} TE_2 &= M1_{MIU1} \cdot M2_{MIU1} \cdot M3_{MIU1 \cdot MIU2} + M1_{MIU1} \cdot M2_{MIU1} \cdot M4_{MIU2} \\ &+ M1_{MIU1} \cdot M2_{MIU1} \cdot M5_{MIU2} + M1_{MIU1} \cdot M3_{MIU1 \cdot MIU2} \cdot M4_{MIU2} \\ &+ M1_{MIU1} \cdot M3_{MIU1 \cdot MIU2} \cdot M5_{MIU2} + M1_{MIU1} \cdot M4_{MIU2} \cdot M5_{MIU2} \\ &+ M2_{MIU1} \cdot M3_{MIU1 \cdot MIU2} \cdot M4_{MIU2} \\ &+ M2_{MIU1} \cdot M3_{MIU1 \cdot MIU2} \cdot M5_{MIU2} \\ &+ M2_{MIU1} \cdot M4_{MIU2} \cdot M5_{MIU2} + M3_{MIU1 \cdot MIU2} \cdot M4_{MIU2} \cdot M5_{MIU2} \end{aligned} \quad (6.9)$$

where

$$\left\{ \begin{array}{l} M1_{MIU1} = MIU1 + M1 \\ M2_{MIU1} = MIU1 + M2 \\ M3_{MIU1 \cdot MIU2} = MIU1 \cdot MIU2 + M3 \\ M4_{MIU2} = MIU2 + M4 \\ M5_{MIU2} = MIU2 + M5 \end{array} \right. \quad (6.10)$$

By replacing $M1_{MIU1}$, $M2_{MIU1}$, $M3_{MIU1 \cdot MIU2}$, $M4_{MIU2}$, and $M5_{MIU2}$ in (6.9) by their expressions in (6.10), the structure function can be developed and simplified to the following form:

$$\begin{aligned}
TE_2 = & MIU1 \cdot MIU2 + MIU1 \cdot M3 + MIU1 \cdot M4 + MIU1 \cdot M5 \\
& + MIU2 \cdot M1 + MIU2 \cdot M2 + MIU2 \cdot M3 + M1 \cdot M2 \cdot M3 \\
& + M1 \cdot M2 \cdot M4 + M1 \cdot M2 \cdot M5 + M1 \cdot M3 \cdot M4 \\
& + M1 \cdot M3 \cdot M5 + M1 \cdot M4 \cdot M5 + M2 \cdot M3 \cdot M4 \\
& + M2 \cdot M3 \cdot M5 + M2 \cdot M4 \cdot M5 + M3 \cdot M4 \cdot M5
\end{aligned}$$

6.3.1.3 Determination of the structure function of subtree 1

The behavioural model of 2 Spare gates sharing a spare event presented in Section 4.3.1.2 allows to determine the structure function of subtree 1 as

$$TE_1 = CSP1 \cdot CSP2,$$

where $CSP1$ and $CSP2$ denote the two Spare gates, with

$$\begin{cases} CSP1 = A_a \cdot (A1 \triangleleft A_a) + A1 \cdot (A2 \triangleleft A1) \\ CSP2 = A_a \cdot (A2 \triangleleft A_a) + A2 \cdot (A1 \triangleleft A2) \end{cases}$$

As a consequence,

$$\begin{aligned}
TE_1 &= [A_a \cdot (A1 \triangleleft A_a) + A1 \cdot (A2 \triangleleft A1)] \cdot [A_a \cdot (A2 \triangleleft A_a) + A2 \cdot (A1 \triangleleft A2)] \\
&= A_a \cdot (A1 \triangleleft A_a) \cdot A_a \cdot (A2 \triangleleft A_a) + A_a \cdot (A1 \triangleleft A_a) \cdot A2 \cdot (A1 \triangleleft A2) \\
&\quad + A1 \cdot (A2 \triangleleft A1) \cdot A_a \cdot (A2 \triangleleft A_a) + A1 \cdot (A2 \triangleleft A1) \cdot A2 \cdot (A1 \triangleleft A2) \\
&\stackrel{(3.17)}{=} A_a \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) + A_a \cdot A2 \cdot (A1 \triangleleft A_a) \cdot (A1 \triangleleft A2) \\
&\quad + A_a \cdot A1 \cdot (A2 \triangleleft A_a) \cdot (A2 \triangleleft A1)
\end{aligned}$$

6.3.1.4 Determination of the minimal canonical form of the structure function of the HECS

The structure function of the HECS hence is

$$\begin{aligned}
TE = & A_a \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) + A_a \cdot A2 \cdot (A1 \triangleleft A_a) \cdot (A1 \triangleleft A2) \\
& + A_a \cdot A1 \cdot (A2 \triangleleft A_a) \cdot (A2 \triangleleft A1) \\
& + MIU1 \cdot MIU2 + MIU1 \cdot M3 + MIU1 \cdot M4 + MIU1 \cdot M5 \\
& + MIU2 \cdot M1 + MIU2 \cdot M2 + MIU2 \cdot M3 + M1 \cdot M2 \cdot M3 \\
& + M1 \cdot M2 \cdot M4 + M1 \cdot M2 \cdot M5 + M1 \cdot M3 \cdot M4 + M1 \cdot M3 \cdot M5 \\
& + M1 \cdot M4 \cdot M5 + M2 \cdot M3 \cdot M4 + M2 \cdot M3 \cdot M5 + M2 \cdot M4 \cdot M5 \\
& + M3 \cdot M4 \cdot M5 + BUS1 \cdot BUS2 + HW + SW + OP
\end{aligned}$$

It can be noted that this form is the canonical form of the structure function, since it is a sum-of-product form. However, even though the expressions for the structure functions of subtrees 2, 3, and 4 cannot be further simplified, the expression for the structure function of subtree 1,

$$A_a \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) + A_a \cdot A2 \cdot (A1 \triangleleft A_a) \cdot (A1 \triangleleft A2) + A_a \cdot A1 \cdot (A2 \triangleleft A_a) \cdot (A2 \triangleleft A1),$$

contains a redundant term. This expression is composed of three cut sequence sets

$$\begin{cases} CSS1 &= A_a \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) \\ CSS2 &= A_a \cdot A2 \cdot (A1 \triangleleft A_a) \cdot (A1 \triangleleft A2) \\ CSS3 &= A_a \cdot A1 \cdot (A2 \triangleleft A_a) \cdot (A2 \triangleleft A1) \end{cases}$$

Algorithm 1 allows us to check whether one of these CSSs is included in the other one, according to criterion (6.6), and to remove it from the structure function.

Algorithm 1 starts with $\mathbb{S}_{min} = \mathbb{S} = CSS_1 \cup CSS_2 \cup CSS_3$. For $i = 1$, $CSS = CSS_2 + CSS_3$. Consequently,

$$\begin{aligned} CSS_1 \cdot CSS &= CSS_1 \cdot CSS_2 + CSS_1 \cdot CSS_3 \\ &= A_a \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) \cdot A_a \cdot A2 \cdot (A1 \triangleleft A_a) \cdot (A1 \triangleleft A2) \\ &\quad + A_a \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) \cdot A_a \cdot A1 \cdot (A2 \triangleleft A_a) \cdot (A2 \triangleleft A1) \\ &= A_a \cdot A2 \cdot (A1 \triangleleft A2) \cdot (A2 \triangleleft A_a) \cdot (A1 \triangleleft A_a) \\ &\quad + A_a \cdot A1 \cdot (A2 \triangleleft A1) \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) \\ &\stackrel{(3.33)}{=} A_a \cdot (A1 \triangleleft A2) \cdot (A2 \triangleleft A_a) \cdot (A1 \triangleleft A_a) \\ &\quad + A_a \cdot (A2 \triangleleft A1) \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) \\ &= A_a \cdot [(A1 \triangleleft A2) + (A2 \triangleleft A1)] \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) \\ &\stackrel{(3.15)}{=} A_a \cdot [(A1 \triangleleft A2) + (A1 \triangleleft A2) + (A2 \triangleleft A1)] \\ &\quad \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) \\ &\stackrel{(3.74)}{=} A_a \cdot (A1 + A2) \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) \\ &= A_a \cdot A1 \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) \\ &\quad + A_a \cdot A2 \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) \\ &\stackrel{(3.33)}{=} A_a \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) + A_a \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) \\ &= A_a \cdot (A1 \triangleleft A_a) \cdot (A2 \triangleleft A_a) = CSS_1 \end{aligned}$$

As $CSS_1 \cdot CSS = CSS_1$, CSS_1 is included in CSS and can be removed from the structure function and from \mathbb{S}_{min} : $\mathbb{S}_{min} = CSS_2 \cup CSS_3$. For $i = 2$, $CSS = CSS_3$. Consequently,

$$\begin{aligned} CSS_2 \cdot CSS &= A_a \cdot A2 \cdot (A1 \triangleleft A_a) \cdot (A1 \triangleleft A2) \\ &\quad \cdot A_a \cdot A1 \cdot (A2 \triangleleft A_a) \cdot (A2 \triangleleft A1) \\ &\stackrel{(3.17)}{=} \perp \end{aligned}$$

As $CSS_2 \cdot CSS \neq CSS_2$, CSS_2 is not included in CSS .

For $i = 3$, $CSS = CSS_2$ and the same result can be obtained, so CSS_3 is not included in CSS .

As a result of the minimization algorithm, \mathbb{S}_{min} contains two elements

$$\mathbb{S}_{min} = \{A_a \cdot A_2 \cdot (A_1 \triangleleft A_a) \cdot (A_1 \triangleleft A_2), A_a \cdot A_1 \cdot (A_2 \triangleleft A_a) \cdot (A_2 \triangleleft A_1)\}.$$

The minimal canonical form of the structure function of the DFT of the HECS finally is

$$\begin{aligned} TE = & A_a \cdot A_2 \cdot (A_1 \triangleleft A_a) \cdot (A_1 \triangleleft A_2) + A_a \cdot A_1 \cdot (A_2 \triangleleft A_a) \cdot (A_2 \triangleleft A_1) \\ & + MIU1 \cdot MIU2 + MIU1 \cdot M3 + MIU1 \cdot M4 + MIU1 \cdot M5 \\ & + MIU2 \cdot M1 + MIU2 \cdot M2 + MIU2 \cdot M3 + M1 \cdot M2 \cdot M3 \\ & + M1 \cdot M2 \cdot M4 + M1 \cdot M2 \cdot M5 + M1 \cdot M3 \cdot M4 + M1 \cdot M3 \cdot M5 \\ & + M1 \cdot M4 \cdot M5 + M2 \cdot M3 \cdot M4 + M2 \cdot M3 \cdot M5 + M2 \cdot M4 \cdot M5 \\ & + M3 \cdot M4 \cdot M5 + BUS1 \cdot BUS2 + HW + SW + OP \end{aligned} \quad (6.11)$$

6.3.2 Qualitative analysis of the HECS

According to Section 6.3.1, the minimal canonical form of the structure function of the DFT of the HECS is

$$\begin{aligned} TE = & A_a \cdot A_2 \cdot (A_1 \triangleleft A_a) \cdot (A_1 \triangleleft A_2) + A_a \cdot A_1 \cdot (A_2 \triangleleft A_a) \cdot (A_2 \triangleleft A_1) \\ & + MIU1 \cdot MIU2 + MIU1 \cdot M3 + MIU1 \cdot M4 + MIU1 \cdot M5 \\ & + MIU2 \cdot M1 + MIU2 \cdot M2 + MIU2 \cdot M3 + M1 \cdot M2 \cdot M3 \\ & + M1 \cdot M2 \cdot M4 + M1 \cdot M2 \cdot M5 + M1 \cdot M3 \cdot M4 + M1 \cdot M3 \cdot M5 \\ & + M1 \cdot M4 \cdot M5 + M2 \cdot M3 \cdot M4 + M2 \cdot M3 \cdot M5 + M2 \cdot M4 \cdot M5 \\ & + M3 \cdot M4 \cdot M5 + BUS1 \cdot BUS2 + HW + SW + OP \end{aligned}$$

This structure function contains 23 terms. On the one hand, 21 terms do not contain the temporal operator BF (\triangleleft). They are static and can hence provide minimal cut sets for the DFT:

$$\begin{aligned} & MIU1 \cdot MIU2, MIU1 \cdot M3, MIU1 \cdot M4, MIU1 \cdot M5, \\ & MIU2 \cdot M1, MIU2 \cdot M2, MIU2 \cdot M3, M1 \cdot M2 \cdot M3, \\ & M1 \cdot M2 \cdot M4, M1 \cdot M2 \cdot M5, M1 \cdot M3 \cdot M4, M1 \cdot M3 \cdot M5, \\ & M1 \cdot M4 \cdot M5, M2 \cdot M3 \cdot M4, M2 \cdot M3 \cdot M5, M2 \cdot M4 \cdot M5, \\ & M3 \cdot M4 \cdot M5, BUS1 \cdot BUS2, HW, SW, OP. \end{aligned}$$

On the other hand, 2 terms contain the temporal operator BF (\triangleleft). They are dynamic and can hence provide minimal cut sequences for the DFT. The algebraic term $A_a \cdot A_2 \cdot (A_1 \triangleleft A_a) \cdot (A_1 \triangleleft A_2)$ corresponds to the two minimal cut sequences $[A_1, A_2, A_a]$ and $[A_1, A_a, A_2]$, whereas the algebraic term $A_a \cdot A_1 \cdot (A_2 \triangleleft A_a) \cdot (A_2 \triangleleft A_1)$

corresponds to the two minimal cut sequences $[A2, A1, A_a]$ and $[A2, A_a, A1]$. Any failure sequence including one of these 4 minimal cut sequences – any failure sequence in which events A , $A1$, and $A2$ fail in the same order as in one of these 4 minimal cut sequences – thus is a cut sequence for the DFT. The minimal cut sequences of the DFT hence are

$$[A1, A2, A_a], [A1, A_a, A2], [A2, A1, A_a], [A2, A_a, A1].$$

The minimal cut sets and sequences of the DFT of the HECS can then be determined as

$$\begin{aligned} & [A1, A2, A_a], [A1, A_a, A2], [A2, A1, A_a], [A2, A_a, A1], \\ & MIU1 \cdot MIU2, MIU1 \cdot M3, MIU1 \cdot M4, MIU1 \cdot M5, \\ & MIU2 \cdot M1, MIU2 \cdot M2, MIU2 \cdot M3, M1 \cdot M2 \cdot M3, \\ & M1 \cdot M2 \cdot M4, M1 \cdot M2 \cdot M5, M1 \cdot M3 \cdot M4, M1 \cdot M3 \cdot M5, \\ & M1 \cdot M4 \cdot M5, M2 \cdot M3 \cdot M4, M2 \cdot M3 \cdot M5, M2 \cdot M4 \cdot M5, \\ & M3 \cdot M4 \cdot M5, BUS1 \cdot BUS2, HW, SW, OP. \end{aligned}$$

Galileo provides the same minimal cut sets and sequences for the DFT of the HECS. However, Galileo considers that $A1 \cdot A2 \cdot A$ is a minimal cut set whereas it is not. Indeed, $[A1, A2, A_a]$, $[A1, A_a, A2]$, $[A2, A1, A_a]$, and $[A2, A_a, A1]$ are minimal cut sequences for the DFT, but $[A_d, A1, A2]$ and $[A_d, A2, A1]$ are not since such sequences cannot occur as A is a cold spare.

This difference between the results obtained with Galileo and with our approach is caused by the ZBDD approach used to perform the qualitative analysis in Galileo. In our case, the difference of results occurs in the processing system failure subsystem. Indeed, the qualitative analysis performed in Galileo consists in

1. replacing dynamic gates in the DFT with the static gates corresponding to their logic constraints: the two Spare gates are hence replaced by two AND gates, thus making impossible the modelling of the behaviour of the spare event A , which should normally replace the first main component to fail and which should not fail while in its dormant mode as it is a cold spare;
2. generating the minimal cut sets of the resulting SFT: the SFT of the processing system failure subsystem hence has a single minimal cut set, which is $A1 \cdot A2 \cdot A$; and
3. expanding each minimal cut set to minimal cut sequences by considering the timing constraints: as there are only timing constraints between $A1$ and A and between $A2$ and A and as the behaviour of the cold spare event A was not modelled at step 1, the minimal cut sequences obtained are equivalent to the minimal cut set $A1 \cdot A2 \cdot A$. However, the sequence $[A_d, A1, A2]$, which is included in $A1 \cdot A2 \cdot A$, cannot be a minimal cut sequence as A cannot fail in its dormant mode.

6.3.3 Quantitative analysis of the HECS

According to Section 6.3.1.4, the minimal canonical form of the structure function of the DFT of the HECS is

$$\begin{aligned}
TE = & A_a \cdot A2 \cdot (A1 \triangleleft A_a) \cdot (A1 \triangleleft A2) + A_a \cdot A1 \cdot (A2 \triangleleft A_a) \cdot (A2 \triangleleft A1) \\
& + MIU1 \cdot MIU2 + MIU1 \cdot M3 + MIU1 \cdot M4 + MIU1 \cdot M5 \\
& + MIU2 \cdot M1 + MIU2 \cdot M2 + MIU2 \cdot M3 + M1 \cdot M2 \cdot M3 \\
& + M1 \cdot M2 \cdot M4 + M1 \cdot M2 \cdot M5 + M1 \cdot M3 \cdot M4 + M1 \cdot M3 \cdot M5 \\
& + M1 \cdot M4 \cdot M5 + M2 \cdot M3 \cdot M4 + M2 \cdot M3 \cdot M5 + M2 \cdot M4 \cdot M5 \\
& + M3 \cdot M4 \cdot M5 + BUS1 \cdot BUS2 + HW + SW + OP
\end{aligned}$$

All the terms of this minimal canonical form are not statistically independent since some of them contain the same basic events. To make the calculation of the failure probability of the TE easier, this structure function can be divided into the 4 structure functions of the 4 subtrees of the DFT of the HECS which were determined in Section 6.3.1.

The failure probability of the TE of the DFT can hence be determined as

$$\begin{aligned}
Pr \{TE\} (t) &= Pr \{TE_1 + TE_2 + TE_3 + TE_4\} (t) \\
&\stackrel{(6.8)}{=} Pr \{TE_1\} (t) + Pr \{TE_2\} (t) + Pr \{TE_3\} (t) + Pr \{TE_4\} (t) \\
&\quad - Pr \{TE_1\} (t) \times Pr \{TE_2\} (t) - Pr \{TE_1\} (t) \times Pr \{TE_3\} (t) \\
&\quad - Pr \{TE_1\} (t) \times Pr \{TE_4\} (t) - Pr \{TE_2\} (t) \times Pr \{TE_3\} (t) \\
&\quad - Pr \{TE_2\} (t) \times Pr \{TE_4\} (t) - Pr \{TE_3\} (t) \times Pr \{TE_4\} (t) \\
&\quad + Pr \{TE_1\} (t) \times Pr \{TE_2\} (t) \times Pr \{TE_3\} (t) \\
&\quad + Pr \{TE_1\} (t) \times Pr \{TE_2\} (t) \times Pr \{TE_4\} (t) \\
&\quad + Pr \{TE_1\} (t) \times Pr \{TE_3\} (t) \times Pr \{TE_4\} (t) \\
&\quad + Pr \{TE_2\} (t) \times Pr \{TE_3\} (t) \times Pr \{TE_4\} (t) \\
&\quad - Pr \{TE_1\} (t) \times Pr \{TE_2\} (t) \times Pr \{TE_3\} (t) \times Pr \{TE_4\} (t)
\end{aligned} \tag{6.12}$$

thanks to the inclusion–exclusion formula [Trivedi 2001], since the 4 subtrees are statistically independent. The four expressions for $Pr \{TE_1\} (t)$, $Pr \{TE_2\} (t)$, $Pr \{TE_3\} (t)$, and $Pr \{TE_4\} (t)$ can then be determined as follows.

6.3.3.1 Calculation of $Pr \{TE_1\} (t)$

$$\begin{aligned}
Pr \{TE_1\} (t) &= Pr \{A_a \cdot A2 \cdot (A1 \triangleleft A_a) \cdot (A1 \triangleleft A2) \\
&\quad + A_a \cdot A1 \cdot (A2 \triangleleft A_a) \cdot (A2 \triangleleft A1)\} (t)
\end{aligned}$$

The algebraic term $A_a \cdot A2 \cdot (A1 \triangleleft A_a) \cdot (A1 \triangleleft A2)$ corresponds to the two minimal cut sequences $[A1, A2, A_a]$ and $[A1, A_a, A2]$, whereas the algebraic term $A_a \cdot A1 \cdot$

$(A2 \triangleleft A_a) \cdot (A2 \triangleleft A1)$ corresponds to the two minimal cut sequences $[A2, A1, A_a]$ and $[A2, A_a, A1]$. The structure function for TE_1 can hence be rewritten as

$$\begin{aligned} TE_1 = & A_a \cdot (A1 \triangleleft A2) \cdot (A2 \triangleleft A_a) + A2 \cdot (A1 \triangleleft A_a) \cdot (A_a \triangleleft A2) \\ & + A_a \cdot (A2 \triangleleft A1) \cdot (A1 \triangleleft A_a) + A1 \cdot (A2 \triangleleft A_a) \cdot (A_a \triangleleft A1), \end{aligned}$$

where each algebraic term represents one of the 4 minimal cut sequences determined previously. As there four algebraic terms are disjoint,

$$\begin{aligned} Pr\{TE_1\}(t) &= Pr\{A_a \cdot (A1 \triangleleft A2) \cdot (A2 \triangleleft A_a)\}(t) \\ &+ Pr\{A2 \cdot (A1 \triangleleft A_a) \cdot (A_a \triangleleft A2)\}(t) \\ &+ Pr\{A_a \cdot (A2 \triangleleft A1) \cdot (A1 \triangleleft A_a)\}(t) \\ &+ Pr\{A1 \cdot (A2 \triangleleft A_a) \cdot (A_a \triangleleft A1)\}(t) \\ &= \int_0^t \left(\int_w^t \left(\int_w^u f_{A2}(v) dv \right) f_{A_a}(u, w) du \right) f_{A1}(w) dw \\ &+ \int_0^t \left(\int_0^w \left(\int_v^w f_{A_a}(u, v) du \right) f_{A1}(v) dv \right) f_{A2}(w) dw \\ &+ \int_0^t \left(\int_w^t \left(\int_w^u f_{A1}(v) dv \right) f_{A_a}(u, w) du \right) f_{A2}(w) dw \\ &+ \int_0^t \left(\int_0^w \left(\int_v^w f_{A_a}(u, v) du \right) f_{A2}(v) dv \right) f_{A1}(w) dw \end{aligned}$$

thanks to the expressions presented in Section 5.1 and to the probabilistic model of Spare gates presented in Section 5.4.

6.3.3.2 Calculation of $Pr\{TE_2\}(t)$

The expression for $Pr\{TE_2\}(t)$ can be determined thanks to the inclusion–exclusion formula [Trivedi 2001]. It will not be detailed here because of its size.

6.3.3.3 Calculation of $Pr\{TE_3\}(t)$

The expression for $Pr\{TE_3\}(t)$ can be determined directly as

$$\begin{aligned} Pr\{TE_3\}(t) &= Pr\{BUS1 \cdot BUS2\}(t) \\ &= F_{BUS1}(t) \times F_{BUS2}(t) \end{aligned}$$

6.3.3.4 Calculation of $Pr\{TE_4\}(t)$

The expression for $Pr\{TE_4\}(t)$ can be determined directly as

$$\begin{aligned} Pr\{TE_4\}(t) &= Pr\{HW + SW + OP\}(t) \\ &= F_{HW}(t) + F_{SW}(t) + F_{OP}(t) \\ &\quad - F_{HW}(t) \times F_{SW}(t) - F_{HW}(t) \times F_{OP}(t) - F_{SW}(t) \times F_{OP}(t) \\ &\quad + F_{HW}(t) \times F_{SW}(t) \times F_{OP}(t) \end{aligned}$$

6.3.3.5 Failure probability of the TE of the DFT of the HECS

The four expressions obtained for $Pr\{TE_1\}(t)$, $Pr\{TE_2\}(t)$, $Pr\{TE_3\}(t)$, and $Pr\{TE_4\}(t)$ allow to determine the failure probability of the TE of the DFT of the HECS thanks to the relation (6.12). As the expressions obtained in Sections 6.3.3.1 to 6.3.3.4 do not depend on the distribution considered for basic events, the failure probability of the TE does not either.

In the particular case of exponential distributions with the failure rates given in Table 6.1, relation (6.12) allows to determine an unreliability of 95.92% for the HECS at mission time $T = 100$ hours. The Galileo tool provides the same result.

Table 6.1: Failure rates of the basic events of the DFT of the HECS, from [Stamatelatos 2002]

Basic component	Failure rate
A1, A2, A	10^{-4}
Mi	6×10^{-5}
MIUi	5×10^{-5}
BUS1, BUS2	10^{-6}
HW	5×10^{-5}
SW	3×10^{-2}
OP	10^{-3}

It can be noted that the result obtained is different from the result obtained in [Stamatelatos 2002]. This difference is due to the fact that basic components were considered as repairable in [Stamatelatos 2002]. Besides, this result does not depend on the distribution considered for basic events. As the exponential distribution is not necessarily the most suitable to model the failure of components, the failure probability of the HECS could be computed by considering other more suitable distributions, such as the Weibull distribution, as well.

6.4 Dynamic FTA of the HCAS

6.4.1 Minimal canonical form of the structure function of the HCAS

Let us determine the structure function of the DFT of the HCAS in Fig. 6.4. This DFT can be divided into 3 independent subtrees, whose structure functions will be successively determined, by using the modularization algorithm presented in [Dutuit 1996]. These 3 subtrees are as follows:

- subtree 1 corresponds to the failure of the CPU unit. It contains one OR gate, one FDEP gate, and one Spare gate, and is hence dynamic. Its TE will be denoted by TE_1 ;
- subtree 2 corresponds to the failure of the motor section. It contains a single AND gate and is hence static. Its TE will be denoted by TE_2 ;

- subtree 3 corresponds to the failure of the pumps unit. It contains one PAND gate and two Spare gates, and is hence dynamic. Its TE will be denoted by TE_3 .

The structure function of the DFT of the HCAS can hence be expressed as

$$TE = TE_1 + TE_2 + TE_3.$$

6.4.1.1 Determination of the structure function of subtree 2

The structure function of subtree 2 can be determined easily since it is static:

$$TE_2 = MOTOR \cdot MOTORC.$$

6.4.1.2 Determination of the structure function of subtree 3

The behavioural model of gate PAND presented in Section 4.1 allows to express TE_3 as

$$TE_3 = CSP2 \cdot (CSP1 \trianglelefteq CSP2).$$

According to the behavioural model of a single Spare gate with 2 inputs presented in Section 4.3.1.1,

$$\begin{cases} CSP1 = BP_a \cdot (P1 \triangleleft BP_a) + P1 \cdot (P2 \triangleleft P1) \\ CSP2 = BP_a \cdot (P2 \triangleleft BP_a) + P2 \cdot (P1 \triangleleft P2) \end{cases}$$

where $P1$ and $P2$ denote $PUMP_1$ and $PUMP_2$, respectively, for the sake of clarity. $CSP1$ and $CSP2$ denote the two Spare gates $CSPGate_1$ and $CSPGate_2$.

The following result will be exploited to determine the structure function:

$$\begin{aligned} A \trianglelefteq ((A \cdot B) + C) &\stackrel{(3.52)}{=} (A \trianglelefteq (A \cdot B)) \cdot (A \trianglelefteq C) \\ &\stackrel{(3.53)}{=} ((A \trianglelefteq A) + (A \trianglelefteq B)) \cdot (A \trianglelefteq C) \\ &\stackrel{(3.51)}{=} (A + (A \trianglelefteq B)) \cdot (A \trianglelefteq C) \\ &\stackrel{(3.60)}{=} A \cdot (A \trianglelefteq C) \stackrel{(3.62)}{=} A \trianglelefteq C \end{aligned} \tag{6.13}$$

$CSP1 \trianglelefteq CSP2$ can now be expressed as

$$\begin{aligned}
CSP1 \trianglelefteq CSP2 &= [BP_a \cdot (P1 \triangleleft BP_a) + P1 \cdot (P2 \triangleleft P1)] \trianglelefteq CSP2 \\
&\stackrel{(3.56)}{=} (BP_a \cdot (P1 \triangleleft BP_a)) \trianglelefteq CSP2 \\
&\quad + (P1 \cdot (P2 \triangleleft P1)) \trianglelefteq CSP2 \\
&\stackrel{(3.57)}{=} (BP_a \trianglelefteq CSP2) \cdot ((P1 \triangleleft BP_a) \trianglelefteq CSP2) \\
&\quad + (P1 \trianglelefteq CSP2) \cdot ((P2 \triangleleft P1) \trianglelefteq CSP2) \\
&\stackrel{(3.59)}{=} (BP_a \trianglelefteq CSP2) \cdot (P1 \triangleleft BP_a) \cdot (P1 \trianglelefteq CSP2) \\
&\quad + (P1 \trianglelefteq CSP2) \cdot (P2 \triangleleft P1) \cdot (P2 \trianglelefteq CSP2) \\
&= (P1 \triangleleft BP_a) \cdot (BP_a \trianglelefteq CSP2) \cdot (P1 \trianglelefteq CSP2) \\
&\quad + (P2 \triangleleft P1) \cdot (P1 \trianglelefteq CSP2) \cdot (P2 \trianglelefteq CSP2) \\
&\stackrel{(3.15),(3.16)}{=} (P1 \trianglelefteq BP_a) \cdot (BP_a \trianglelefteq CSP2) \cdot (P1 \trianglelefteq CSP2) \\
&\quad + (P2 \trianglelefteq P1) \cdot (P1 \trianglelefteq CSP2) \cdot (P2 \trianglelefteq CSP2) \\
&\stackrel{(3.66)}{=} (P1 \trianglelefteq BP_a) \cdot (BP_a \trianglelefteq CSP2) \\
&\quad + (P2 \trianglelefteq P1) \cdot (P1 \trianglelefteq CSP2) \\
&\stackrel{(3.15),(3.16)}{=} (P1 \triangleleft BP_a) \cdot (BP_a \trianglelefteq CSP2) \\
&\quad + (P2 \triangleleft P1) \cdot (P1 \trianglelefteq CSP2)
\end{aligned}$$

On the one hand,

$$\begin{aligned}
BP_a \trianglelefteq CSP2 &= BP_a \trianglelefteq [BP_a \cdot (P2 \triangleleft BP_a) + P2 \cdot (P1 \triangleleft P2)] \\
&\stackrel{(6.13)}{=} BP_a \trianglelefteq [P2 \cdot (P1 \triangleleft P2)] \\
&\stackrel{(3.53)}{=} (BP_a \trianglelefteq P2) + (BP_a \trianglelefteq (P1 \triangleleft P2)) \\
&\stackrel{(3.15),(3.54)}{=} (BP_a \trianglelefteq P2) + (BP_a \triangleleft P1) + BP_a \cdot P1 \cdot (P2 \trianglelefteq P1) \\
&\stackrel{(3.15),(3.16)}{=} (BP_a \triangleleft P2) + (BP_a \triangleleft P1) + BP_a \cdot P1 \cdot (P2 \triangleleft P1)
\end{aligned}$$

On the other hand,

$$\begin{aligned}
P1 \trianglelefteq CSP2 &= P1 \trianglelefteq [BP_a \cdot (P2 \triangleleft BP_a) + P2 \cdot (P1 \triangleleft P2)] \\
&\stackrel{(3.33)}{=} P1 \trianglelefteq [BP_a \cdot (P2 \triangleleft BP_a) + P1 \cdot P2 \cdot (P1 \triangleleft P2)] \\
&\stackrel{(6.13)}{=} P1 \trianglelefteq [BP_a \cdot (P2 \triangleleft BP_a)] \\
&\stackrel{(3.53)}{=} (P1 \trianglelefteq BP_a) + (P1 \trianglelefteq (P2 \triangleleft BP_a)) \\
&\stackrel{(3.15),(3.54)}{=} (P1 \trianglelefteq BP_a) + (P1 \triangleleft P2) + P1 \cdot P2 \cdot (BP_a \trianglelefteq P2) \\
&\stackrel{(3.15),(3.16)}{=} (P1 \triangleleft BP_a) + (P1 \triangleleft P2) + P1 \cdot P2 \cdot (BP_a \triangleleft P2)
\end{aligned}$$

Consequently,

$$\begin{aligned}
CSP1 \trianglelefteq CSP2 &= (P1 \triangleleft BP_a) \cdot [(BP_a \triangleleft P2) + (BP_a \triangleleft P1)] \\
&\quad + BP_a \cdot P1 \cdot (P2 \triangleleft P1) + (P2 \triangleleft P1) \cdot [(P1 \triangleleft BP_a) \\
&\quad + (P1 \triangleleft P2) + P1 \cdot P2 \cdot (BP_a \triangleleft P2)] \\
&\stackrel{(3.17)}{=} (P1 \triangleleft BP_a) \cdot [(BP_a \triangleleft P2) + BP_a \cdot P1 \cdot (P2 \triangleleft P1)] \\
&\quad + (P2 \triangleleft P1) \cdot [(P1 \triangleleft BP_a) + P1 \cdot P2 \cdot (BP_a \triangleleft P2)] \\
&= (P1 \triangleleft BP_a) \cdot (BP_a \triangleleft P2) \\
&\quad + BP_a \cdot P1 \cdot (P1 \triangleleft BP_a) \cdot (P2 \triangleleft P1) \\
&\quad + (P2 \triangleleft P1) \cdot (P1 \triangleleft BP_a) \\
&\quad + P1 \cdot P2 \cdot (P2 \triangleleft P1) \cdot (BP_a \triangleleft P2) \\
&\stackrel{(3.13)}{=} (P1 \triangleleft BP_a) \cdot (BP_a \triangleleft P2) + (P2 \triangleleft P1) \cdot (P1 \triangleleft BP_a) \\
&\quad + P1 \cdot (BP_a \triangleleft P2) \cdot (P2 \triangleleft P1)
\end{aligned}$$

Since BP_a cannot fail before $P1$ and $P2$,

$$CSP1 \trianglelefteq CSP2 = (P1 \triangleleft BP_a) \cdot (BP_a \triangleleft P2) + (P2 \triangleleft P1) \cdot (P1 \triangleleft BP_a)$$

Finally,

$$\begin{aligned}
TE_3 &= [BP_a \cdot (P2 \triangleleft BP_a) + P2 \cdot (P1 \triangleleft P2)] \\
&\quad \cdot [(P1 \triangleleft BP_a) \cdot (BP_a \triangleleft P2) + (P2 \triangleleft P1) \cdot (P1 \triangleleft BP_a)] \\
&\stackrel{(3.17)}{=} P2 \cdot (P1 \triangleleft P2) \cdot (P1 \triangleleft BP_a) \cdot (BP_a \triangleleft P2) \\
&\quad + BP_a \cdot (P2 \triangleleft BP_a) \cdot (P2 \triangleleft P1) \cdot (P1 \triangleleft BP_a) \\
&\stackrel{(3.34)}{=} P2 \cdot (P1 \triangleleft BP_a) \cdot (BP_a \triangleleft P2) + BP_a \cdot (P2 \triangleleft P1) \cdot (P1 \triangleleft BP_a).
\end{aligned}$$

6.4.1.3 Determination of the structure function of subtree 1

The model of Spare gates presented in Section 4.3 is valid when the input events of the Spare gate are independent basic events which can consequently not occur simultaneously. However, in the case of subtree 1, basic events P and B are basic events which have a common cause failure represented by the Trigger, and they can hence occur simultaneously when the trigger occurs. This particular aspect can be taken into account in the model by introducing an additional term related to the simultaneous occurrence of P_T and $B_T - P_T \triangleleft B_T$ - in the algebraic model of the Spare gate. TE_1 can hence first be expressed as

$$TE_1 = B_{a_T} \cdot (P_T \triangleleft B_{a_T}) + P_T \cdot (B_{d_T} \triangleleft P_T) + P_T \triangleleft B_T,$$

where B_a and B_d denote the active and dormant state of the spare unit B, according to the behavioural model of the Spare gate presented in Section 4.3.1.1. As explained

in Section 4.2, the substituted variables B_{aT} , B_{dT} , B_T , and P_T explicitly indicate the effect of trigger T and denote the global failure of basic events B and P . Thus we have:

$$\begin{cases} B_T = B + T \\ B_{aT} = B_a + T \\ B_{dT} = B_d + T \\ P_T = P + T \end{cases}$$

The additional term $P_T \triangle B_T$ can first be determined since $T = CS + SS$:

$$\begin{aligned} P_T \triangle B_T &= (P + CS + SS) \triangle (B + CS + SS) \\ &\stackrel{(3.39)}{=} ((P + CS + SS) \triangle B) \cdot (B \triangle (CS + SS)) \\ &\quad + ((P + CS + SS) \triangle (CS + SS)) \cdot ((CS + SS) \triangle B) \\ &\stackrel{(3.39)}{=} (B \triangle P) \cdot (P \triangle (CS + SS)) \cdot (B \triangle (CS + SS)) \\ &\quad + (B \triangle (CS + SS)) \cdot ((CS + SS) \triangle P) \cdot (B \triangle (CS + SS)) \\ &\quad + ((CS + SS) \triangle P) \cdot (P \triangle (CS + SS)) \cdot ((CS + SS) \triangle B) \\ &\quad + ((CS + SS) \triangle (CS + SS)) \cdot ((CS + SS) \triangle P) \\ &\quad \cdot ((CS + SS) \triangle B) \end{aligned}$$

According to (3.15), $B \triangle P = \perp$, and in the same way, $B \triangle (CS + SS) = (CS + SS) \triangle P = \perp$. Consequently,

$$\begin{aligned} P_T \triangle B_T &= ((CS + SS) \triangle (CS + SS)) \cdot ((CS + SS) \triangle P) \cdot ((CS + SS) \triangle B) \\ &\stackrel{(3.38)}{=} (CS + SS) \cdot ((CS + SS) \triangle P) \cdot ((CS + SS) \triangle B) \\ &\stackrel{(3.33)}{=} ((CS + SS) \triangle B) \cdot ((CS + SS) \triangle P) \\ &= ((CS + SS) \triangle B_d) \cdot ((CS + SS) \triangle P) \\ &\quad + ((CS + SS) \triangle P) \cdot ((CS + SS) \triangle B_a) \end{aligned}$$

As a result,

$$\begin{aligned} TE_1 &= (B_a + CS + SS) \cdot ((P + CS + SS) \triangle (B_a + CS + SS)) \\ &\quad + (P + CS + SS) \cdot ((B_d + CS + SS) \triangle (P + CS + SS)) \\ &\quad + ((CS + SS) \triangle B_d) \cdot ((CS + SS) \triangle P) \\ &\quad + ((CS + SS) \triangle P) \cdot ((CS + SS) \triangle B_a). \end{aligned}$$

Finally, this structure function can be developed thanks to the use of theorems

(3.23) and (3.27), and simplified to the following form:

$$\begin{aligned}
TE_1 = & (P \triangleleft B_a) \cdot (B_a \triangleleft (CS + SS)) \\
& + (P \triangleleft (CS + SS)) \cdot ((CS + SS) \triangleleft B_a) \\
& + (B_d \triangleleft P) \cdot (P \triangleleft (CS + SS)) \\
& + (B_d \triangleleft (CS + SS)) \cdot ((CS + SS) \triangleleft P) \\
& + ((CS + SS) \triangleleft B_d) \cdot ((CS + SS) \triangleleft P) \\
& + ((CS + SS) \triangleleft P) \cdot ((CS + SS) \triangleleft B_a)
\end{aligned}$$

Some of the terms of this structure function can be grouped to obtain the final simplified following form:

$$TE_1 = CS + SS + P \cdot (B_d \triangleleft P) + B_a \cdot (P \triangleleft B_a).$$

6.4.1.4 Determination of the minimal canonical form of the structure function of the HCAS

The structure function of the HCAS can finally be determined as

$$\boxed{TE = CS + SS + MOTOR \cdot MOTORC + P \cdot (B_d \triangleleft P) + B_a \cdot (P \triangleleft B_a) + BP_a \cdot (P2 \triangleleft P1) \cdot (P1 \triangleleft BP_a) + P2 \cdot (P1 \triangleleft BP_a) \cdot (BP_a \triangleleft P2).}$$

It can be noted that this form is the canonical form of the structure function, since it is a sum-of-product form. Besides, application of algorithm 1 allows us to determine that this form is minimal according to criterion (6.6), and hence is the minimal canonical form of the structure function of the HCAS.

6.4.2 Qualitative analysis of the HCAS

According to Section 6.4.1, the minimal canonical form of the structure function of the DFT of the HCAS is

$$\begin{aligned}
TE = & CS + SS + MOTOR \cdot MOTORC + P \cdot (B_d \triangleleft P) + B_a \cdot (P \triangleleft B_a) \\
& + BP_a \cdot (P2 \triangleleft P1) \cdot (P1 \triangleleft BP_a) + P2 \cdot (P1 \triangleleft BP_a) \cdot (BP_a \triangleleft P2)
\end{aligned}$$

This structure function contains 7 terms. On the one hand, 3 terms do not contain the temporal operator BF (\triangleleft). They are static and can hence provide minimal cut sets for the DFT:

$$CS, SS, MOTOR \cdot MOTORC.$$

On the other hand, 4 terms contain the temporal operator BF (\triangleleft). They are dynamic and can hence provide minimal cut sequences for the DFT. For instance, the algebraic term $P2 \cdot (P1 \triangleleft BP_a) \cdot (BP_a \triangleleft P2)$ indicates that $P1$, BP_a , and $P2$ must fail in this order, so the sequence $[P1, BP_a, P2]$ is a cut sequence for the DFT. As any failure sequence including the sequence $[P1, BP_a, P2]$ is a cut sequence for

the DFT, the sequence $[P1, BP_a, P2]$ is a minimal cut sequence for the DFT. Each dynamic algebraic term of this structure function hence provides a single minimal cut sequence for the DFT, and the minimal cut sequences of the DFT are as follows:

$$[B_d, P], [P, B_a], [P2, P1, BP_a], [P1, BP_a, P2].$$

The minimal cut sets and sequences of the DFT of the HCAS can then be determined as

$$[B_d, P], [P, B_a], [P2, P1, BP_a], [P1, BP_a, P2], \\ CS, SS, MOTOR \cdot MOTORC.$$

In this case, it can be noted that the two minimal cut sequences $[B_d, P]$ and $[P, B_a]$ are logically equivalent to the single minimal cut set $P \cdot B$. However, the minimal cut set does not render the two states of the basic event B which will be needed to perform the quantitative analysis of the DFT. This is the reason why these two minimal cut sequences were not reduced to the equivalent minimal cut set $B \cdot P$.

Galileo provides the same minimal cut sets for the DFT, but it does not consider $[P2, P1, BP_a]$ and $[P1, BP_a, P2]$ as minimal cut sequences. However, Galileo considers that $P1 \cdot P2 \cdot BP$ is a minimal cut set whereas it is not. Indeed, $[P2, P1, BP_a]$ and $[P1, BP_a, P2]$ are minimal cut sequences for the DFT (even though Galileo does not recognize them as such), but $[P1, P2, BP_a]$, $[P2, BP_a, P1]$, $[BP_d, P1, P2]$, and $[BP_d, P2, P1]$ are not.

Once again, this difference between the results obtained with Galileo and with our approach is caused by the ZBDD approach used to perform the qualitative analysis in Galileo. In our case, the difference of results occurs in the pumps unit. The qualitative analysis performed in Galileo consists in

1. replacing dynamic gates in the DFT with the static gates corresponding to their logic constraints: the two CSP gates and the PAND gate are hence replaced by three AND gates, thus making impossible the modelling of the behaviour of the spare event BP , which should normally replace the first main component to fail and which should not fail while in its dormant mode as it is a cold spare;
2. generating the minimal cut sets of the resulting SFT: the SFT of the pumps unit hence has a single minimal cut set, which is $P1 \cdot P2 \cdot BP$; and
3. expanding each minimal cut set to minimal cut sequences by considering the timing constraints: as there are only timing constraints between $P1$ and BP and between $P2$ and BP and as the behaviour of the cold spare event BP was not modelled at step 1, the timing constraints allow to obtain minimal cut sequences which are equivalent to the single minimal cut set $P1 \cdot P2 \cdot BP$. However, the sequence $[BP_d, P1, P2]$, which is included in $P1 \cdot P2 \cdot BP$, cannot be a minimal cut sequence as BP cannot fail in its dormant mode.

6.4.3 Quantitative analysis of the HCAS

According to Section 6.4.1.4, the minimal canonical form of the structure function of the DFT of the HCAS is

$$TE = CS + SS + MOTOR \cdot MOTORC + P \cdot (B_d \triangleleft P) + B_a \cdot (P \triangleleft B_a) + BP_a \cdot (P_2 \triangleleft P_1) \cdot (P_1 \triangleleft BP_a) + P_2 \cdot (P_1 \triangleleft BP_a) \cdot (BP_a \triangleleft P_2)$$

All the terms of this minimal canonical form are not statistically independent since some of them contain the same basic events. To make the calculation of the failure probability of the TE easier, this structure function can be divided into the 3 structure functions of the 3 subtrees of the DFT of the HCAS which were determined in Section 6.4.2.

The failure probability of the TE of the DFT can hence be determined as

$$\begin{aligned} Pr\{TE\}(t) &= Pr\{TE_1 + TE_2 + TE_3\}(t) \\ &\stackrel{(6.8)}{=} Pr\{TE_1\}(t) + Pr\{TE_2\}(t) + Pr\{TE_3\}(t) \\ &\quad - Pr\{TE_1\}(t) \times Pr\{TE_2\}(t) - Pr\{TE_1\}(t) \times Pr\{TE_3\}(t) \\ &\quad - Pr\{TE_2\}(t) \times Pr\{TE_3\}(t) \\ &\quad + Pr\{TE_1\}(t) \times Pr\{TE_2\}(t) \times Pr\{TE_3\}(t) \end{aligned} \quad (6.14)$$

thanks to the inclusion–exclusion formula [Trivedi 2001], since the 3 subtrees are statistically independent. The expressions for $Pr\{TE_1\}(t)$, $Pr\{TE_2\}(t)$, and $Pr\{TE_3\}(t)$ can then be determined as follows.

6.4.3.1 Calculation of $Pr\{TE_1\}(t)$

$$Pr\{TE_1\}(t) = Pr\{CS + SS + P \cdot (B_d \triangleleft P) + B_a \cdot (P \triangleleft B_a)\}(t)$$

According to the probabilistic model of a single Spare gate with 2 inputs presented in Section 5.4.2.1,

$$\begin{aligned} Pr\{P \cdot (B_d \triangleleft P) + B_a \cdot (P \triangleleft B_a)\}(t) &= \int_0^t \left(\int_v^t f_{B_a}(u, v) du \right) f_P(v) dv \\ &\quad + \int_0^t f_P(u) F_{B_d}(u) du \end{aligned}$$

As a consequence, according to the inclusion–exclusion formula [Trivedi 2001],

$$\begin{aligned} Pr\{TE_1\}(t) &= F_{CS}(t) + F_{SS}(t) - F_{CS}(t) \times F_{SS}(t) \\ &\quad + (1 - F_{CS}(t) - F_{SS}(t) + F_{CS}(t) \times F_{SS}(t)) \\ &\quad \times \left(\int_0^t \left(\int_v^t f_{B_a}(u, v) du \right) f_P(v) dv + \int_0^t f_P(u) F_{B_d}(u) du \right) \end{aligned}$$

6.4.3.2 Calculation of $Pr\{TE_2\}(t)$

The expression for $Pr\{TE_2\}(t)$ can be determined directly as

$$\begin{aligned} Pr\{TE_2\}(t) &= Pr\{MOTOR \cdot MOTORC\}(t) \\ &= F_{MOTOR}(t) \times F_{MOTORC}(t) \end{aligned}$$

6.4.3.3 Calculation of $Pr\{TE_3\}(t)$

The expression for $Pr\{TE_3\}(t)$ can be determined as

$$\begin{aligned} Pr\{TE_3\}(t) &= Pr\{BP_a \cdot (P2 \triangleleft P1) \cdot (P1 \triangleleft BP_a)\}(t) \\ &\quad + Pr\{P2 \cdot (P1 \triangleleft BP_a) \cdot (BP_a \triangleleft P2)\}(t) \\ &= \int_0^t \left(\int_w^t \left(\int_w^u f_{P1}(v) dv \right) f_{BP_a}(u, w) du \right) f_{P2}(w) dw \\ &\quad + \int_0^t \left(\int_0^w \left(\int_v^w f_{BP_a}(u, v) du \right) f_{P1}(v) dv \right) f_{P2}(w) dw \end{aligned}$$

thanks to the expressions presented in Section 5.1 and to the probabilistic model of Spare gates presented in Section 5.4.

6.4.3.4 Failure probability of the TE of the DFT of the HCAS

The expressions obtained for $Pr\{TE_1\}(t)$, $Pr\{TE_2\}(t)$, and $Pr\{TE_3\}(t)$ allow to determine the failure probability of the TE of the DFT of the HCAS thanks to the relation (6.14). As the expressions obtained in Sections 6.4.3.1 to 6.4.3.3 do not depend on the distribution considered for basic events, the failure probability of the TE does not either.

In the particular case of exponential distributions with the failure rates given in Table 6.2 and with a dormancy of 0.5 for the spare event B , relation (6.14) allows to determine an unreliability of 36.35% for the HCAS at mission time $T = 100000$ hours. The Galileo tool provides the same result.

Table 6.2: Failure rates of the basic events of the DFT of the HCAS, from [Boudali 2005a]

Basic component	Failure rate (10^{-6})
CS	1
SS	2
P, B	4
P1, P2, BP	5
MOTOR	5
MOTORC	1

Once more, this result does not depend on the distribution considered for basic events. As the exponential distribution is not necessarily the most suitable to model

the failure of components, the failure probability of the HCAS could be computed by considering other more suitable distributions, such as the Weibull distribution, as well.

Conclusions

This dissertation has presented an algebraic framework based on a temporal model of events and on the introduction of temporal operators BEFORE (BF) and SIMULTANEOUS (SM). This framework allows to extend the analytical approaches commonly used to analyze SFTs to DFTs. The technical contributions of this dissertation can be summarized as follows:

- the algebraic framework introduced allows to provide a behavioural model of all dynamic gates (PAND, FDEP, and Spare);
- this behavioural model of dynamic gates allows to determine the structure function of any DFT;
- the structure function of DFTs can always be simplified to a sum-of-product canonical form thanks to a list of theorems that we provide;
- such a canonical form of the structure function may contain redundant terms, which may be removed from the structure function thanks to the use of an algorithm that we provide, and which is based on a minimization criterion;
- the qualitative analysis of any DFT can be performed directly from its structure function;
- a probabilistic model of dynamic gates that we provide also allows to perform the quantitative analysis of any DFT from its structure function. One of the advantages of this probabilistic model is that it does not depend on the failure distribution considered for basic events.

Much work remains to be done. We showed that a set of minimal cut sequences can be extracted directly from the minimal canonical form of the structure function of DFTs. However, nothing ensures that this set of minimal cut sequences is minimal as it may contain minimal cut sequences from different cut sequence sets but which may be redundant. Indeed, the set of minimal cut sequences can be minimized in many ways, depending on the objective. On the one hand, it can be minimized to make qualitative analysis easier by keeping the shortest sequences only, making the assumption that longer sequences may have a lower probability of occurrence and may hence be less relevant. On the other hand, it can be minimized to make quantitative analysis easier by keeping the sequences with the highest probability of occurrence only. The determination of the *minimal set of minimal cut sequences*

hence is a more complex issue than a mere minimization issue which depends on the targeted objective.

Besides, the work presented in this dissertation allowed to propose a formal background for the determination of the structure function of any DFT. The theorems provided allow to develop and simplify this structure function to a minimal canonical form. However, one should be aware of the complexity of the calculation of this canonical form. The few structure function developments presented in this dissertation show that the simplification of structure functions can be quite complex and depends on the order in which the theorems are used. The expertise of the analyst can thus be quite useful to make the determination of the canonical form of the structure function easier. The probabilistic models of dynamic gates can also become quite complex, in the case of Spare gates, for instance. Future work will hence be dedicated to the elaboration of efficient algorithms allowing to automatically perform the calculation of this structure function and the qualitative and quantitative analysis of DFTs. Such an implementation will require the definition of temporal operators, as well as the definition of the theorems which will be useful to simplify the structure function, making sure that the theorems defined will not be redundant. Algorithms would allow to automatically extract the cut sequences of the DFT from the minimal canonical form of its structure function, and to determine which sequences include the other ones and are hence minimal. The application of the probabilistic model of dynamic gates to specific non necessarily exponential failure distributions would also be simplified by the use of some algorithms.

Finally, the hypotheses of this algebraic framework were that events are non-repairable and that basic events are statistically independent. Finally, we made the hypothesis that events are non-repairable, in accordance with [Vesely 1981]. However, repairable events allow to model cyclic behaviours [Bobbio 2004] and would hence be worth being taken into account as many users aim at modeling repairable systems. The algebraic framework presented in this dissertation will no longer be valid because of its hypotheses. A new set of repairable events will have to be defined, and the temporal model of repairable events will have to be defined, each repairable event having not only one date of appearance, but many dates of appearance and disappearance. A new definition of temporal operators BEFORE and SIMULTANEOUS will have to be determined on the set of repairable events, and the behavioural model of dynamic gates will be the same as the behavioural model presented in this dissertation. Even though the unary operator NOT ($\bar{}$) commonly used in the Boolean algebra of Boolean variables could not be defined on the set of non-repairable events because it was not stable, it could be defined on the set of repairable events and may thus provide a Boolean algebra structure to the set of repairable events. Some of the theorems which were presented in this dissertation may no longer be valid, and the theorems which will still be valid on the set of repairable events will have to be demonstrated by exploiting the new model of temporal operators. Such a change in the hypotheses of the framework will hence result in a sizeable work.

Bibliography

- [Amari 2003] S. Amari, G. Dill and E. Howals. *A new approach to solve dynamic fault-trees*. In Proceedings IEEE Annual Reliability and Maintainability Symposium (RAMS 2003), pages 374–379, Tampa, FL, USA, 2003. 16, 52
- [Banks 1984] J. Banks and J. Carson. Discrete-event system simulation. Prentice-Hall INC, 1984. 9
- [Billingsley 1995] P. Billingsley. Probability and measure. John Wiley & Sons, New York, USA, 1995. 56
- [Bobbio 2001] A. Bobbio, L. Portinale, M. Minichino and E. Ciancamerla. *Improving the analysis of dependable systems by mapping fault trees into Bayesian networks*. Reliability Engineering and System Safety, vol. 71, no. 3, pages 249–260, 2001. 22
- [Bobbio 2004] A. Bobbio and D. Codetta Raiteri. *Parametric Fault Trees with Dynamic Gates and Repair Boxes*. In Proceedings of the Annual Reliability and Maintainability Symposium (RAMS 2004), pages 459–465, Los Angeles, CA, USA, 2004. vii, 18, 19, 20, 92
- [Boudali 2005a] H. Boudali and J. B. Dugan. *A discrete-time Bayesian network reliability modeling and analysis framework*. Reliability Engineering & System Safety, vol. 87, no. 3, pages 337–349, 2005. vii, ix, 21, 22, 71, 88
- [Boudali 2005b] H. Boudali and J. B. Dugan. *A New Bayesian Network Approach to Solve Dynamic Fault Trees*. In Proceedings of the Annual Reliability and Maintainability Symposium (RAMS 2005), pages 451–456, Alexandria, VA, USA, 2005. 22
- [Boudali 2006] H. Boudali and J. B. Dugan. *A Continuous-Time Bayesian Network Reliability Modeling, and Analysis Framework*. IEEE Transactions on Reliability, vol. 55, no. 1, pages 86–97, 2006. 21, 22, 44
- [Boudali 2007a] H. Boudali, P. Crouzen and M. Stoelinga. *A compositional semantics for Dynamic Fault Trees in terms of Interactive Markov Chains*. In Proceedings of the 5th International Symposium on Automated Technology for Verification and Analysis (ATVA 2007), pages 441–456, Tokyo, Japan, 2007. 12
- [Boudali 2007b] H. Boudali, P. Crouzen and M. Stoelinga. *Dynamic Fault Tree analysis through input/output interactive Markov chains*. In Proceedings of the International Conference on Dependable Systems and Networks (DSN 2007), pages 25–38, Edinburgh, UK, 2007. 31

- [Bouissou 2003] M. Bouissou and J.-L. Bon. *A new formalism that combines advantages of fault-trees and Markov models: Boolean logic driven Markov processes*. Reliability Engineering and System Safety, vol. 82, no. 2, pages 149–163, 2003. 15
- [Chiola 1993] G. Chiola, C. Dutheillet, G. Franceschinis and S. Haddad. *Stochastic Well-Formed Colored Nets and Symmetric Modeling Applications*. IEEE Transactions on Computers, vol. 42, no. 11, pages 1343–1360, 1993. 18
- [Chiola 1995] G. Chiola, G. Franceschinis, R. Gaeta and M. Ribaud. *GreatSPN 1.7: Graphical Editor and Analyzer for Timed and Stochastic Petri Nets*. Performance Evaluation, vol. 24, no. 1-2, pages 47–68, 1995. 20
- [Clarhaut 2009] J. Clarhaut, B. Conrard, S. Hayat and V. Cocquempot. *Optimal Design of Dependable Control System Architectures Using Temporal Sequences of Failures*. IEEE Transactions on Reliability, vol. 58, no. 3, pages 511–522, 2009. 15
- [Coppit 2000] D. Coppit, K. J. Sullivan and J.B. Dugan. *Formal Semantics of Models for Computational Engineering: A Case Study on Dynamic Fault Trees*. In Proceedings of the 11th International Symposium on Software Reliability Engineering (ISSRE 2000), pages 270–282, San Jose, CA, USA, 2000. vii, 16
- [Coppit 2003] D. Coppit. *Engineering Modeling and Analysis: Sound Methods and Effective Tools*. PhD thesis, University of Virginia, 2003. 16
- [Delamare 2003] C. Delamare, Y. Gardan and P. Moreaux. *Efficient Implementation for Performance Evaluation of Synchronous Decomposition of High Level Stochastic Petri Nets*. In Proceedings of the ICALP2003 Workshop on Stochastic Petri Nets and Related Formalisms, pages 164–183, Eindhoven, Holland, 2003. 20
- [Dugan 1990] J.B. Dugan, S. Bavuso and M. Boyd. *Fault Trees and Sequence Dependencies*. In Proceedings of the Annual Reliability and Maintainability Symposium (RAMS 1990), pages 286–293, 1990. 12, 13
- [Dugan 1992] J.B. Dugan, S. Bavuso and M. Boyd. *Dynamic fault-tree models for fault-tolerant computer systems*. IEEE Transactions on Reliability, vol. 41, no. 3, pages 363–377, 1992. 2, 16, 24
- [Dugan 1993] J.B. Dugan, S. Bavuso and M. Boyd. *Fault trees and Markov models for reliability analysis of fault tolerant systems*. Reliability Engineering and System Safety, vol. 39, no. 3, pages 291–307, 1993. 16
- [Dugan 2000] J.B. Dugan, K.J. Sullivan and D. Coppit. *Developing a low-cost high-quality software tool for Dynamic fault-tree analysis*. IEEE Transactions on Reliability, vol. 49, no. 1, pages 49–59, 2000. 2, 6, 12, 16, 18, 25, 41

- [Dutuit 1996] Y. Dutuit and A. Rauzy. *A Linear-Time Algorithm to Find Modules of Fault Trees*. IEEE Transactions on Reliability, vol. 45, no. 3, pages 422–425, 1996. 2, 18, 72, 80
- [ECSS 2001] ECSS. Failure modes, effects and criticality analysis (FMECA). European Cooperation for Space Standardization, ESA Publication, 2001. 11
- [Ejlali 2003] A. Ejlali and S.G. Miremadi. *Time-to-Failure Tree*. In Proceedings of the Annual Reliability and Maintainability Symposium (RAMS 2003), pages 148–152, Tampa, FL, USA, 2003. 15
- [Emerson 1990] E.A. Emerson. *Temporal and modal logic*. In Elsevier Science Publishers, Handbook of Theoretical Computer Science, pages 996–1072, Amsterdam, The Netherlands, 1990. 11
- [Fenelon 1985] P. Fenelon, J. McDermid, A. Nicholson and D. Pumfrey. *Experience with the Application of HAZOP to Computer-Based Systems*. In Proceedings of the 10th Annual Conference on Computer Assurance, pages 37–48, Gaithersburg, MD, USA, 1985. 1
- [Fussell 1976] J.B. Fussell, E.F. Aber and R.G. Rahl. *On the Quantitative Analysis of Priority-AND Failure Logic*. IEEE Transactions on Reliability, vol. R-25, no. 5, pages 324–326, 1976. 2, 12, 52
- [Grimaldi 2004] R.P. Grimaldi. Discrete and combinatorial mathematics: an applied introduction. Addison Wesley, 5th edition, 2004. 30
- [Grimmett 2001] G.R. Grimmett and D.R. Stirzaker. Probability and Random Processes. Oxford University Press, USA, 3rd edition, 2001. 56
- [Gulati 1997] R. Gulati and J.B. Dugan. *A modular approach for analyzing static and dynamic fault trees*. In Proceedings of the Annual Reliability and Maintainability Symposium (RAMS 1997), pages 57–63, Philadelphia, PA, USA, 1997. 16
- [Henley 2000] E.J. Henley and H. Kumamoto. Probabilistic Risk Assessment and Management for Engineers and Scientists. Wiley-IEEE Press, 2nd edition, 2000. 1, 68
- [Jensen 1996] F.V. Jensen. An introduction to Bayesian networks. Springer, New York, 1996. 20
- [Lang 2005] S. Lang. Algebra. Springer-Verlag New York Inc., 3rd edition, 2005. 34
- [Lauritzen 2001] S.L. Lauritzen and F. Jensen. *Stable local computation with conditional Gaussian distributions*. Statistics and Computing, vol. 11, no. 2, pages 191–203, 2001. 22

- [Leveson 1987] N.G. Leveson. *Safety analysis using Petri nets*. IEEE Transactions on Software Engineering, vol. SE-13, pages 386–397, 1987. 1
- [Malhotra 1994] M. Malhotra and K. Trivedi. *Power-Hierarchy of Dependability-Model Types*. IEEE Transactions on Reliability, vol. 43, no. 3, pages 493–502, 1994. 30
- [McDermott 1996] R.E. McDermott, R.J. Mikulak and M.R. Beauregard. The Basics of FMEA. Quality Resources, 1996. 11
- [McMillan 1990] K.L. McMillan. Symbolic Model Checking. Kluwer Academic Publishers, 1990. 11
- [Merle 2007a] G. Merle and J.-M. Roussel. *Algebraic modelling of Fault Trees with Priority AND gates*. In Proceedings of the 1st IFAC Workshop on Dependable Control of Discrete Systems (DCDS'07), pages 175–180, Paris, France, 2007. 2, 24, 35
- [Merle 2007b] G. Merle and J.-M. Roussel. *Modèle algébrique des arbres de défaillance dynamiques intégrant des contraintes sur l'ordre d'occurrence des événements*. In Proceedings of the 2nd Doctoriales / MACS National Event (JD-JN-MACS'07), Reims, France, 2007. CD Rom paper no. 40, 6 p. 24, 35
- [Merle 2009] G. Merle, J.-M. Roussel, J.-J. Lesage and A. Bobbio. *Algebraic Expression of the Structure Function of a subclass of Dynamic Fault Trees*. In Proceedings of the 2nd IFAC Workshop on Dependable Control of Discrete Systems (DCDS'09), pages 129–134, Bari, Italy, 2009. 24, 35
- [Merle 2010a] G. Merle, J.-M. Roussel and J.-J. Lesage. *Improving the Efficiency of Dynamic Fault Tree Analysis by Considering Gates FDEP as Static*. In Proceedings of the European Safety & Reliability Conference 2010 (ESREL 2010), Rhodes, Greece, 2010. To be published. 45
- [Merle 2010b] G. Merle, J.-M. Roussel, J.-J. Lesage and A. Bobbio. *Probabilistic Algebraic Analysis of Fault Trees with Priority Dynamic Gates and Repeated Events*. IEEE Transactions on Reliability, vol. 59, no. 1, pages 250–261, 2010. 24, 25
- [Merle 2010c] G. Merle, J.-M. Roussel, J.-J. Lesage and N. Vayatis. *Analytical Calculation of Failure Probabilities in Dynamic Fault Trees including Spare Gates*. In Proceedings of the European Safety & Reliability Conference 2010 (ESREL 2010), Rhodes, Greece, 2010. To be published. 24
- [Minato 1993] S. Minato. *Zero-suppressed BDDs for set manipulation in combinatorial problems*. In Proceedings of the 30th ACM/IEEE Design Automation Conference (DAC'93), pages 272–277, Hamburg, Germany, 1993. 10

- [Minato 2001] S. Minato. *Zero-suppressed BDDs and their applications*. International Journal on Software Tools for Technology Transfer (STTT), vol. 3, pages 156–170, 2001. 10
- [Montani 2005] S. Montani, L. Portinale, A. Bobbio and D. Codetta Raiteri. *DB-Net, a tool to convert Dynamic Fault Trees into Dynamic Bayesian Networks*. Technical report, Dipartimento di Informatica, Università del Piemonte Orientale, 2005. 22, 25
- [Moral 2001] S. Moral, R. Rumí and A. Salmerón. *Mixtures of truncated exponentials in hybrid Bayesian networks*. In Proceedings of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning With Uncertainty, vol. 2143, Lecture Notes in Artificial Intelligence, pages 145–167, 2001. 22
- [Ortmeier 2005] F. Ortmeier, W. Reif and G. Schellhorn. *Deductive Cause-Consequence Analysis (DCCA)*. In Proceedings of the 16th IFAC World Congress, Prague, Czech Republic, 2005. 8, 11
- [Papadopoulos 2001] Y. Papadopoulos, J. McDermid, R. Sasse and G. Heiner. *Analysis and synthesis of the behaviour of complex programmable electronic systems in conditions of failure*. Reliability Engineering and System Safety, vol. 71, no. 3, pages 229–247, 2001. 25
- [Raiteri 2005] D. Codetta Raiteri. *The Conversion of Dynamic Fault Trees to Stochastic Petri Nets, as a case of Graph Transformation*. Electronic Notes in Theoretical Computer Science, vol. 127, no. 2, pages 45–60, 2005. 20
- [Rao 2009] K. Durga Rao, V. Gopika, V.V.S. Sanyasi Rao, H.S. Kushwaha, A.K. Verma and A. Srividya. *Dynamic fault tree analysis using Monte Carlo simulation in probabilistic safety assessment*. Reliability Engineering and System Safety, vol. 94, no. 4, pages 872–883, 2009. 16
- [Rauzy 1997] A. Rauzy and Y. Dutuit. *Exact and truncated computations of prime implicants of coherent and non-coherent fault trees within Aralia*. Reliability Engineering and System Safety, vol. 58, no. 2, pages 127–144, 1997. 10
- [Rauzy 2001] A. Rauzy. *Mathematical Foundations of Minimal Cutsets*. IEEE Transactions on Reliability, vol. 50, no. 4, pages 389–396, 2001. 9, 65, 66
- [Reifer 1979] D. Reifer. *Software Failure Modes and Effects Analysis*. IEEE Transactions on Reliability, vol. R-28, no. 3, pages 247–249, 1979. 1
- [Ren 1998] Y. Ren and J.B. Dugan. *Design of Reliable Systems Using Static & Dynamic Fault Trees*. IEEE Transactions on Reliability, vol. 47, no. 3, pages 234–244, 1998. 71

- [Schellhorn 2002] G. Schellhorn, A. Thums and W. Reif. *Formal fault tree semantics*. In Proceedings of the 6th World Conference on Integrated Design & Process Technology, Pasadena, CA, USA, 2002. 11
- [Shooman 1990] M.L. Shooman. Probabilistic Reliability: An Engineering Approach. McGraw-Hill, New York, 2nd edition, 1990. 1, 68
- [Stamatelatos 2002] M. Stamatelatos and W. Vesely. *Fault Tree Handbook with Aerospace Applications*. volume 1.1, pages 1–205. NASA Office of Safety and Mission Assurance, 2002. ix, 11, 13, 23, 29, 45, 46, 48, 65, 70, 80
- [Tang 2004] Z. Tang and J.B. Dugan. *Minimal cut set/sequence generation for dynamic fault trees*. In Proceedings of the Annual Reliability and Maintainability Symposium (RAMS 2004), pages 207–213, Los Angeles, CA, USA, 2004. vii, 10, 11, 17, 65, 67
- [Trivedi 2001] K. Trivedi. Probability & statistics with reliability, queueing & computer science applications. Wiley, 2 edition, 2001. 8, 9, 69, 78, 79, 87
- [Vemuri 1999] K.K. Vemuri, J.B. Dugan and K.J. Sullivan. *Automatic synthesis of fault trees for computer-based systems*. IEEE Transactions on Reliability, vol. 48, no. 4, pages 394–402, 1999. 71
- [Vesely 1981] W. Vesely, F. Goldberg, N. Roberts and D. Haasl. Fault Tree Handbook NUREG-0492. U.S. Nuclear Regulatory Commission, 1981. vii, 1, 2, 6, 12, 13, 23, 29, 44, 92
- [Vittorini 2002] V. Vittorini, G. Franceschinis, M. Gribaudo, M. Iacono and N. Mazzocca. *Drawnet: Model objects to support performance analysis and simulation of systems*. 12th International Conference on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation, Springer Verlag - LNCS, vol. 2324, pages 233–238, 2002. 20, 25
- [Walker 2006] M. Walker and Y. Papadopoulos. *Pandora: The time of priority-AND gates*. In Proceedings of the 12th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2006), pages 237–242, Saint-Etienne, France, 2006. 22, 41
- [Walker 2007a] M. Walker, L. Bottaci and Y. Papadopoulos. *Compositional temporal fault tree analysis*. In Proceedings of the 26th International Conference on Computer Safety, Reliability and Security (SAFECOMP 2007), pages 106–119, Nuremberg, Germany, 2007. 24
- [Walker 2007b] M. Walker and Y. Papadopoulos. *Pandora 2: The time of priority-OR gates*. In Proceedings of the 1st IFAC Workshop on Dependable Control of Discrete Systems (DCDS'07), pages 169–174, Paris, France, 2007. 23

-
- [Walker 2009] M. Walker and Y. Papadopoulos. *Qualitative temporal analysis: Towards a full implementation of the Fault Tree Handbook*. Control Engineering Practice, vol. 17, no. 10, pages 1115–1125, 2009. [vii](#), [23](#), [24](#), [42](#)
- [Watson 1961] H.A. Watson and Bell Telephone Laboratories. *Launch control safety study*. Technical report, Bell Telephone Laboratories, Murray Hill, NJ, 1961. [1](#)
- [Yuge 2008] T. Yuge and S. Yanagi. *Quantitative analysis of a fault tree with priority AND gates*. Reliability Engineering and System Safety, vol. 93, no. 11, pages 1577–1583, 2008. [30](#)

Proofs of the theorems

A.1 Principle of the proof of the theorems

A.1.1 Theorems involving 2 events

When theorems involve 2 events a and b , 3 cases are possible:

- a and b occur simultaneously ($d(a) = d(b)$);
- a occurs before b ($d(a) < d(b)$);
- a occurs after b ($d(a) > d(b)$).

Such theorems were hence demonstrated by considering the 3 possible orderings of the failure dates of a and b :

- Case 1: the two dates are equal: $d(a) = d(b)$
- Case 2: the two dates are different
 - Case 2a: $d(a) < d(b)$
 - Case 2b: $d(a) > d(b)$

A.1.2 Theorems involving 3 events

When theorems involve 3 events a , b , and c , 13 cases are possible:

- a , b , and c occur simultaneously ($d(a) = d(b) = d(c)$);
- 2 events out of the 3 events occur simultaneously ($d(a) = d(b)$ or $d(a) = d(c)$ or $d(b) = d(c)$). As the third event can occur before or after the 2 others, we consider 6 cases;
- the 3 events occur at different dates (6 cases).

Such theorems were hence demonstrated by considering the 13 possible orderings of the failure dates of a , b , and c :

- Case 1: the three dates are equal: $d(a) = d(b) = d(c)$
- Case 2: two dates out of three are equal
 - Case 2a: $d(a) = d(b) < d(c)$

- Case 2b: $d(a) = d(b) > d(c)$
- Case 2c: $d(a) = d(c) < d(b)$
- Case 2d: $d(a) = d(c) > d(b)$
- Case 2e: $d(b) = d(c) < d(a)$
- Case 2f: $d(b) = d(c) > d(a)$
- Case 3: the three dates are different
 - Case 3a: $d(a) < d(b) < d(c)$
 - Case 3b: $d(a) < d(c) < d(b)$
 - Case 3c: $d(b) < d(a) < d(c)$
 - Case 3d: $d(b) < d(c) < d(a)$
 - Case 3e: $d(c) < d(a) < d(b)$
 - Case 3f: $d(c) < d(b) < d(a)$

A.2 Proofs

In many cases, a theorem T_i will be proved by using other theorems T_j . However, none of theorems T_j is proved by using theorem T_i nor is proved by using theorems which are proved by using theorem T_i .

A.2.1 Proofs of the theorems satisfied by Boolean operators

Proof of Theorem (3.1) : $a + b = b + a$

1. Case 1: the two dates are equal: $d(a) = d(b)$, meaning that $a = b$

By definition:

$$a + b = a \qquad b + a = b$$

Given $a = b$, $a + b = b + a$.

2. Case 2: the two dates are different

- (a) Case 2a: $d(a) < d(b)$

By definition:

$$a + b = a \qquad b + a = a$$

- (b) Case 2b: $d(a) > d(b)$

By definition:

$$a + b = b \qquad b + a = b$$

For all the cases of the domain of our study, $a + b = b + a$. Thus:

$$a + b = b + a$$

End of the proof of Theorem (3.1).

Proof of Theorem (3.2) : $a \cdot b = b \cdot a$

1. Case 1:
- $d(a) = d(b)$
- , meaning that
- $a = b$

By definition:

$$a \cdot b = a \qquad b \cdot a = b$$

Given $a = b$, $a \cdot b = b \cdot a$.

2. Case 2: the two dates are different

- (a) Case 2a:
- $d(a) < d(b)$

By definition:

$$a \cdot b = b \qquad b \cdot a = b$$

- (b) Case 2b:
- $d(a) > d(b)$

By definition:

$$a \cdot b = a \qquad b \cdot a = a$$

For all the cases of the domain of our study, $a \cdot b = b \cdot a$. Thus:

$$a \cdot b = b \cdot a$$

*End of the proof of Theorem (3.2).***Proof of Theorem (3.3) : $a + (b + c) = (a + b) + c$**

1. Case 1: the three dates are equal:
- $d(a) = d(b) = d(c)$
- , meaning that
- $a = b = c$

By definition:

$$\begin{array}{ll} b + c = b & a + b = a \\ a + (b + c) = a & (a + b) + c = a \end{array}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a:
- $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{ll} b + c = b & a + b = a \\ a + (b + c) = a & (a + b) + c = a \end{array}$$

- (b) Case 2b:
- $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{ll} b + c = c & a + b = a \\ a + (b + c) = c & (a + b) + c = c \end{array}$$

(c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{ll} b + c = c & a + b = a \\ a + (b + c) = a & (a + b) + c = a \end{array}$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{ll} b + c = b & a + b = b \\ a + (b + c) = b & (a + b) + c = b \end{array}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{ll} b + c = b & a + b = b \\ a + (b + c) = b & (a + b) + c = b \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{ll} b + c = b & a + b = a \\ a + (b + c) = a & (a + b) + c = a \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{ll} b + c = b & a + b = a \\ a + (b + c) = a & (a + b) + c = a \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{ll} b + c = c & a + b = a \\ a + (b + c) = a & (a + b) + c = a \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{ll} b + c = b & a + b = b \\ a + (b + c) = b & (a + b) + c = b \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{ll} b + c = b & a + b = b \\ a + (b + c) = b & (a + b) + c = b \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{ll} b + c = c & a + b = a \\ a + (b + c) = c & (a + b) + c = c \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{ll} b + c = c & a + b = b \\ a + (b + c) = c & (a + b) + c = c \end{array}$$

For all the cases of the domain of our study, $a + (b + c) = (a + b) + c$. Thus:

$$a + (b + c) = (a + b) + c$$

End of the proof of Theorem (3.3).

Proof of Theorem (3.4) : $a \cdot (b \cdot c) = (a \cdot b) \cdot c$

1. Case 1: the three dates are different: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{array}{ll} b \cdot c = b & a \cdot b = a \\ a \cdot (b \cdot c) = a & (a \cdot b) \cdot c = a \end{array}$$

2. Case 2: two dates out of three are different

(a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{ll} b \cdot c = c & a \cdot b = a \\ a \cdot (b \cdot c) = c & (a \cdot b) \cdot c = c \end{array}$$

(b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{ll} b \cdot c = b & a \cdot b = a \\ a \cdot (b \cdot c) = a & (a \cdot b) \cdot c = a \end{array}$$

(c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{ll} b \cdot c = b & a \cdot b = b \\ a \cdot (b \cdot c) = b & (a \cdot b) \cdot c = b \end{array}$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{ll} b \cdot c = c & a \cdot b = a \\ a \cdot (b \cdot c) = a & (a \cdot b) \cdot c = a \end{array}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{ll} b \cdot c = b & a \cdot b = a \\ a \cdot (b \cdot c) = a & (a \cdot b) \cdot c = a \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{ll} b \cdot c = b & a \cdot b = b \\ a \cdot (b \cdot c) = b & (a \cdot b) \cdot c = b \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{ll} b \cdot c = c & a \cdot b = b \\ a \cdot (b \cdot c) = c & (a \cdot b) \cdot c = c \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{ll} b \cdot c = b & a \cdot b = b \\ a \cdot (b \cdot c) = b & (a \cdot b) \cdot c = b \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{ll} b \cdot c = c & a \cdot b = a \\ a \cdot (b \cdot c) = c & (a \cdot b) \cdot c = c \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{ll} b \cdot c = c & a \cdot b = a \\ a \cdot (b \cdot c) = a & (a \cdot b) \cdot c = a \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{ll} b \cdot c = b & a \cdot b = b \\ a \cdot (b \cdot c) = b & (a \cdot b) \cdot c = b \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{ll} b \cdot c = b & a \cdot b = a \\ a \cdot (b \cdot c) = a & (a \cdot b) \cdot c = a \end{array}$$

For all the cases of the domain of our study, $a \cdot (b \cdot c) = (a \cdot b) \cdot c$. Thus:

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

End of the proof of Theorem (3.4).

Proof of Theorem (3.5) : $a + a = a$

By definition, $a + b = a$ if $d(a) = d(b)$. Thus, if $b = a$, $d(b) = d(a)$ and:

$$a + a = a$$

End of the proof of Theorem (3.5).

Proof of Theorem (3.6) : $a \cdot a = a$

By definition, $a \cdot b = a$ if $d(a) = d(b)$. Thus, if $b = a$, $d(b) = d(a)$ and:

$$a \cdot a = a$$

End of the proof of Theorem (3.6).

Proof of Theorem (3.7) : $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$
By definition:

$$\begin{array}{l} b + c = b \\ a \cdot (b + c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \cdot b = a \\ a \cdot c = a \end{array} \right. \\ (a \cdot b) + (a \cdot c) = a \end{array}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$
By definition:

$$\begin{array}{l} b + c = b \\ a \cdot (b + c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \cdot b = a \\ a \cdot c = c \end{array} \right. \\ (a \cdot b) + (a \cdot c) = a \end{array}$$

- (b) Case 2b: $d(a) = d(b) > d(c)$
By definition:

$$\begin{array}{l} b + c = c \\ a \cdot (b + c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \cdot b = a \\ a \cdot c = a \end{array} \right. \\ (a \cdot b) + (a \cdot c) = a \end{array}$$

- (c) Case 2c: $d(a) = d(c) < d(b)$
By definition:

$$\begin{array}{l} b + c = c \\ a \cdot (b + c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \cdot b = b \\ a \cdot c = a \end{array} \right. \\ (a \cdot b) + (a \cdot c) = a \end{array}$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{ll} b + c = b & \begin{cases} a \cdot b = a \\ a \cdot c = a \end{cases} \\ a \cdot (b + c) = a & (a \cdot b) + (a \cdot c) = a \end{array}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{ll} b + c = b & \begin{cases} a \cdot b = a \\ a \cdot c = a \end{cases} \\ a \cdot (b + c) = a & (a \cdot b) + (a \cdot c) = a \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{ll} b + c = b & \begin{cases} a \cdot b = b \\ a \cdot c = c \end{cases} \\ a \cdot (b + c) = b & (a \cdot b) + (a \cdot c) = b \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{ll} b + c = b & \begin{cases} a \cdot b = b \\ a \cdot c = c \end{cases} \\ a \cdot (b + c) = b & (a \cdot b) + (a \cdot c) = b \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{ll} b + c = c & \begin{cases} a \cdot b = b \\ a \cdot c = c \end{cases} \\ a \cdot (b + c) = c & (a \cdot b) + (a \cdot c) = c \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{ll} b + c = b & \begin{cases} a \cdot b = a \\ a \cdot c = c \end{cases} \\ a \cdot (b + c) = a & (a \cdot b) + (a \cdot c) = a \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{ll} b + c = b & \begin{cases} a \cdot b = a \\ a \cdot c = a \end{cases} \\ a \cdot (b + c) = a & (a \cdot b) + (a \cdot c) = a \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{l} b + c = c \\ a \cdot (b + c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \cdot b = b \\ a \cdot c = a \end{array} \right. \\ (a \cdot b) + (a \cdot c) = a \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{l} b + c = c \\ a \cdot (b + c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \cdot b = a \\ a \cdot c = a \end{array} \right. \\ (a \cdot b) + (a \cdot c) = a \end{array}$$

For all the cases of the domain of our study, $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$. Thus:

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

End of the proof of Theorem (3.7).

Proof of Theorem (3.8) : $a + \perp = a$

1. Case 1: $d(a) = d(\perp) = +\infty$

By definition:

$$a + \perp = a$$

2. Case 2: $d(a) \neq d(\perp)$, meaning that $d(a) < d(\perp)$

By definition:

$$a + \perp = a$$

For all the cases of the domain of our study, $a + \perp = a$. Thus:

$$a + \perp = a$$

End of the proof of Theorem (3.8).

Proof of Theorem (3.9) : $a \cdot \top = a$

1. Case 1: $d(a) = d(\top) = 0$

By definition:

$$a \cdot \top = a$$

2. Case 2: $d(a) \neq d(\top)$, meaning that $d(a) > d(\top)$

By definition:

$$a \cdot \top = a$$

For all the cases of the domain of our study, $a \cdot \top = a$. Thus:

$$a \cdot \top = a$$

End of the proof of Theorem (3.9).

Proof of Theorem (3.10) : $a \cdot \perp = \perp$

1. Case 1: $d(a) = d(\perp) = +\infty$

By definition:

$$a \cdot \perp = \perp$$

2. Case 2: $d(a) \neq d(\perp)$, meaning that $d(a) < d(\perp)$

By definition:

$$a \cdot \perp = \perp$$

For all the cases of the domain of our study, $a \cdot \perp = \perp$. Thus:

$$a \cdot \perp = \perp$$

End of the proof of Theorem (3.10).

A.2.2 Proofs of the theorems satisfied by operator non-inclusive BEFORE

Proof of Theorem (3.17) : $(a \triangleleft b) \cdot (b \triangleleft a) = \perp$

1. Case 1: the two dates are equal: $d(a) = d(b)$, meaning that $a = b$

By definition:

$$\begin{cases} a \triangleleft b = \perp \\ b \triangleleft a = \perp \end{cases} \\ (a \triangleleft b) \cdot (b \triangleleft a) = \perp$$

2. Case 2: the two dates are different

- (a) Case 2a: $d(a) < d(b)$

By definition:

$$\begin{cases} a \triangleleft b = a \\ b \triangleleft a = \perp \end{cases} \\ (a \triangleleft b) \cdot (b \triangleleft a) = \perp$$

- (b) Case 2b: $d(a) > d(b)$

By definition:

$$\begin{cases} a \triangleleft b = \perp \\ b \triangleleft a = b \end{cases} \\ (a \triangleleft b) \cdot (b \triangleleft a) = \perp$$

For all the cases of the domain of our study, $(a \triangleleft b) \cdot (b \triangleleft a) = \perp$. Thus:

$$(a \triangleleft b) \cdot (b \triangleleft a) = \perp$$

End of the proof of Theorem (3.17).

Proof of Theorem (3.18) :

$$a \triangleleft (b \triangleleft c) = (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = (a \triangleleft b) + (a \cdot b \cdot (c \leq b))$$

This proof will be done in 2 steps:

- Step 1: $a \triangleleft (b \triangleleft c) = (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)))$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{array}{l} b \triangleleft c = \perp \\ a \triangleleft (b \triangleleft c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} c \triangleleft b = \perp \\ c \triangle b = c \end{array} \right. \\ \left\{ \begin{array}{l} a \triangleleft b = \perp \\ (c \triangleleft b) + (c \triangle b) = c \end{array} \right. \\ \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)) = a \end{array} \right. \\ (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = a \end{array}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{l} b \triangleleft c = b \\ a \triangleleft (b \triangleleft c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} c \triangleleft b = \perp \\ c \triangle b = \perp \end{array} \right. \\ \left\{ \begin{array}{l} a \triangleleft b = \perp \\ (c \triangleleft b) + (c \triangle b) = \perp \end{array} \right. \\ \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)) = \perp \end{array} \right. \\ (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = \perp \end{array}$$

- (b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{l} b \triangleleft c = \perp \\ a \triangleleft (b \triangleleft c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} c \triangleleft b = c \\ c \triangle b = \perp \end{array} \right. \\ \left\{ \begin{array}{l} a \triangleleft b = \perp \\ (c \triangleleft b) + (c \triangle b) = c \end{array} \right. \\ \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)) = a \end{array} \right. \\ (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = a \end{array}$$

- (c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{l}
 b \triangleleft c = \perp \\
 a \triangleleft (b \triangleleft c) = a
 \end{array}
 \quad
 \begin{array}{l}
 \left\{ \begin{array}{l} c \triangleleft b = c \\ c \triangle b = \perp \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = a \\ (c \triangleleft b) + (c \triangle b) = c \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = a \\ a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)) = b \end{array} \right. \\
 (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = a
 \end{array}$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{l}
 b \triangleleft c = b \\
 a \triangleleft (b \triangleleft c) = \perp
 \end{array}
 \quad
 \begin{array}{l}
 \left\{ \begin{array}{l} c \triangleleft b = \perp \\ c \triangle b = \perp \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = \perp \\ (c \triangleleft b) + (c \triangle b) = \perp \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)) = \perp \end{array} \right. \\
 (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = \perp
 \end{array}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{l}
 b \triangleleft c = \perp \\
 a \triangleleft (b \triangleleft c) = a
 \end{array}
 \quad
 \begin{array}{l}
 \left\{ \begin{array}{l} c \triangleleft b = \perp \\ c \triangle b = c \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = \perp \\ (c \triangleleft b) + (c \triangle b) = c \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)) = a \end{array} \right. \\
 (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = a
 \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{l}
 b \triangleleft c = \perp \\
 a \triangleleft (b \triangleleft c) = a
 \end{array}
 \quad
 \begin{array}{l}
 \left\{ \begin{array}{l} c \triangleleft b = \perp \\ c \triangle b = c \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = a \\ (c \triangleleft b) + (c \triangle b) = c \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = a \\ a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)) = b \end{array} \right. \\
 (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = a
 \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{l}
 b \triangleleft c = b \\
 a \triangleleft (b \triangleleft c) = a
 \end{array}
 \quad
 \begin{array}{l}
 \left\{ \begin{array}{l} c \triangleleft b = \perp \\ c \triangle b = \perp \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = a \\ (c \triangleleft b) + (c \triangle b) = \perp \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = a \\ a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)) = \perp \end{array} \right. \\
 (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = a
 \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{l}
 b \triangleleft c = \perp \\
 a \triangleleft (b \triangleleft c) = a
 \end{array}
 \quad
 \begin{array}{l}
 \left\{ \begin{array}{l} c \triangleleft b = c \\ c \triangle b = \perp \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = a \\ (c \triangleleft b) + (c \triangle b) = c \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = a \\ a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)) = b \end{array} \right. \\
 (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = a
 \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{l}
 b \triangleleft c = b \\
 a \triangleleft (b \triangleleft c) = \perp
 \end{array}
 \quad
 \begin{array}{l}
 \left\{ \begin{array}{l} c \triangleleft b = \perp \\ c \triangle b = \perp \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = \perp \\ (c \triangleleft b) + (c \triangle b) = \perp \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)) = \perp \end{array} \right. \\
 (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = \perp
 \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{l}
 b \triangleleft c = b \\
 a \triangleleft (b \triangleleft c) = \perp
 \end{array}
 \quad
 \begin{array}{l}
 \left\{ \begin{array}{l} c \triangleleft b = \perp \\ c \triangle b = \perp \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = \perp \\ (c \triangleleft b) + (c \triangle b) = \perp \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)) = \perp \end{array} \right. \\
 (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = \perp
 \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{l}
 b \triangleleft c = \perp \\
 a \triangleleft (b \triangleleft c) = a
 \end{array}
 \quad
 \begin{array}{l}
 \left\{ \begin{array}{l} c \triangleleft b = c \\ c \triangle b = \perp \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = a \\ (c \triangleleft b) + (c \triangle b) = c \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = a \\ a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)) = b \end{array} \right. \\
 (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = a
 \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{l}
 b \triangleleft c = \perp \\
 a \triangleleft (b \triangleleft c) = a
 \end{array}
 \quad
 \begin{array}{l}
 \left\{ \begin{array}{l} c \triangleleft b = c \\ c \triangle b = \perp \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = \perp \\ (c \triangleleft b) + (c \triangle b) = c \end{array} \right. \\
 \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)) = a \end{array} \right. \\
 (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = a
 \end{array}$$

For all the cases of the domain of our study,

$$a \triangleleft (b \triangleleft c) = (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))).$$

Thus:

$$a \triangleleft (b \triangleleft c) = (a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b)))$$

- Step 2: $(a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) = (a \triangleleft b) + (a \cdot b \cdot (c \triangle b))$

For this proof, we will use the following theorem:

$$(3.16) : a \triangle b = a \triangleleft b + a \triangle b$$

$$(a \triangleleft b) + (a \cdot b \cdot ((c \triangleleft b) + (c \triangle b))) \stackrel{(3.16)}{=} (a \triangleleft b) + (a \cdot b \cdot (c \triangle b))$$

End of the proof of Theorem (3.18).

Proof of Theorem (3.19) : $(a \triangleleft b) \triangleleft c = (a \triangleleft b) \cdot (a \triangleleft c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{array}{l}
 a \triangleleft b = \perp \\
 (a \triangleleft b) \triangleleft c = \perp
 \end{array}
 \quad
 \begin{array}{l}
 \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\
 (a \triangleleft b) \cdot (a \triangleleft c) = \perp
 \end{array}$$

2. Case 2: two dates out of three are equal

(a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{ll} a \triangleleft b = \perp & \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) \triangleleft c = \perp & (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

(b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{ll} a \triangleleft b = \perp & \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \triangleleft c = \perp & (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

(c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{ll} a \triangleleft b = a & \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \triangleleft c = \perp & (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{ll} a \triangleleft b = \perp & \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \triangleleft c = \perp & (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{ll} a \triangleleft b = \perp & \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \triangleleft c = \perp & (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{ll} a \triangleleft b = a & \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) \triangleleft c = a & (a \triangleleft b) \cdot (a \triangleleft c) = a \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{l} a \triangleleft b = a \\ (a \triangleleft b) \triangleleft c = a \end{array} \quad \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = a$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{l} a \triangleleft b = a \\ (a \triangleleft b) \triangleleft c = a \end{array} \quad \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = a$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{l} a \triangleleft b = \perp \\ (a \triangleleft b) \triangleleft c = \perp \end{array} \quad \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = \perp$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{l} a \triangleleft b = \perp \\ (a \triangleleft b) \triangleleft c = \perp \end{array} \quad \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = \perp$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{l} a \triangleleft b = a \\ (a \triangleleft b) \triangleleft c = \perp \end{array} \quad \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = \perp$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{l} a \triangleleft b = \perp \\ (a \triangleleft b) \triangleleft c = \perp \end{array} \quad \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = \perp$$

For all the cases of the domain of our study, $(a \triangleleft b) \triangleleft c = (a \triangleleft b) \cdot (a \triangleleft c)$. Thus:

$$(a \triangleleft b) \triangleleft c = (a \triangleleft b) \cdot (a \triangleleft c)$$

End of the proof of Theorem (3.19).

Proof of Theorem (3.20) : $\perp \triangleleft a = \perp$

1. Case 1: $d(a) = d(\perp) = +\infty$
By definition:

$$\perp \triangleleft a = \perp$$

2. Case 2: $d(a) \neq d(\perp)$, meaning that $d(a) < d(\perp)$
By definition:

$$\perp \triangleleft a = \perp$$

For all the cases of the domain of our study, $\perp \triangleleft a = \perp$. Thus:

$$\perp \triangleleft a = \perp$$

End of the proof of Theorem (3.20).

Proof of Theorem (3.21) : $a \triangleleft \perp = a$

1. Case 1: $d(a) = d(\perp) = +\infty$
By definition:

$$a \triangleleft \perp = \perp = a$$

2. Case 2: $d(a) \neq d(\perp)$, meaning that $d(a) < d(\perp)$
By definition:

$$a \triangleleft \perp = a$$

For all the cases of the domain of our study, $a \triangleleft \perp = a$. Thus:

$$a \triangleleft \perp = a$$

End of the proof of Theorem (3.21).

Proof of Theorem (3.22) : $a \triangleleft a = \perp$

By definition, $a \triangleleft b = \perp$ if $d(a) = d(b)$. Thus, if $b = a$, $d(b) = d(a)$ and:

$$a \triangleleft a = \perp$$

End of the proof of Theorem (3.22).

Proof of Theorem (3.23) : $a \triangleleft (b + c) = (a \triangleleft b) \cdot (a \triangleleft c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$
By definition:

$$\begin{array}{l} b + c = b \\ a \triangleleft (b + c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{l} b + c = b \\ a \triangleleft (b + c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

- (b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{l} b + c = c \\ a \triangleleft (b + c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

- (c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{l} b + c = c \\ a \triangleleft (b + c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

- (d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{l} b + c = b \\ a \triangleleft (b + c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

- (e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{l} b + c = b \\ a \triangleleft (b + c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{l} b + c = b \\ a \triangleleft (b + c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = a \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{l} b + c = b \\ a \triangleleft (b + c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = a \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{l} b + c = c \\ a \triangleleft (b + c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = a \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{l} b + c = b \\ a \triangleleft (b + c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{l} b + c = b \\ a \triangleleft (b + c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{l} b + c = c \\ a \triangleleft (b + c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{l} b + c = c \\ a \triangleleft (b + c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (a \triangleleft c) = \perp \end{array}$$

For all the cases of the domain of our study, $a \triangleleft (b + c) = (a \triangleleft b) \cdot (a \triangleleft c)$. Thus:

$$a \triangleleft (b + c) = (a \triangleleft b) \cdot (a \triangleleft c)$$

End of the proof of Theorem (3.23).

Proof of Theorem (3.24) : $a \triangleleft (b \cdot c) = (a \triangleleft b) + (a \triangleleft c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$
By definition:

$$\begin{array}{l} b \cdot c = b \\ a \triangleleft (b \cdot c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) + (a \triangleleft c) = \perp \end{array}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$
By definition:

$$\begin{array}{l} b \cdot c = c \\ a \triangleleft (b \cdot c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) + (a \triangleleft c) = a \end{array}$$

- (b) Case 2b: $d(a) = d(b) > d(c)$
By definition:

$$\begin{array}{l} b \cdot c = b \\ a \triangleleft (b \cdot c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) + (a \triangleleft c) = \perp \end{array}$$

- (c) Case 2c: $d(a) = d(c) < d(b)$
By definition:

$$\begin{array}{l} b \cdot c = b \\ a \triangleleft (b \cdot c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) + (a \triangleleft c) = a \end{array}$$

- (d) Case 2d: $d(a) = d(c) > d(b)$
By definition:

$$\begin{array}{l} b \cdot c = c \\ a \triangleleft (b \cdot c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) + (a \triangleleft c) = \perp \end{array}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{l} b \cdot c = b \\ a \triangleleft (b \cdot c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) + (a \triangleleft c) = \perp \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{l} b \cdot c = b \\ a \triangleleft (b \cdot c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) + (a \triangleleft c) = a \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{l} b \cdot c = c \\ a \triangleleft (b \cdot c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) + (a \triangleleft c) = a \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{l} b \cdot c = b \\ a \triangleleft (b \cdot c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) + (a \triangleleft c) = a \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{l} b \cdot c = c \\ a \triangleleft (b \cdot c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = a \end{array} \right. \\ (a \triangleleft b) + (a \triangleleft c) = a \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{l} b \cdot c = c \\ a \triangleleft (b \cdot c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) + (a \triangleleft c) = \perp \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{l} b \cdot c = b \\ a \triangleleft (b \cdot c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = a \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) + (a \triangleleft c) = a \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{l} b \cdot c = b \\ a \triangleleft (b \cdot c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) + (a \triangleleft c) = \perp \end{array}$$

For all the cases of the domain of our study, $a \triangleleft (b \cdot c) = (a \triangleleft b) + (a \triangleleft c)$. Thus:

$$a \triangleleft (b \cdot c) = (a \triangleleft b) + (a \triangleleft c)$$

End of the proof of Theorem (3.24).

Proof of Theorem (3.25) :

$$a \triangleleft (b \triangleleft c) = (a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c)$$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{array}{l} b \triangleleft c = b \\ a \triangleleft (b \triangleleft c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} b \triangleleft c = \perp \\ c \triangleleft b = \perp \\ a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \\ \left\{ \begin{array}{l} a \cdot (b \triangleleft c) = \perp \\ a \cdot (c \triangleleft b) = \perp \\ a \triangleleft b = \perp \\ a \triangleleft c = \perp \end{array} \right. \end{array}$$

$$(a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) = \perp$$

2. Case 2: two dates out of three are equal

(a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{l} b \triangleleft c = \perp \\ a \triangleleft (b \triangleleft c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} b \triangleleft c = b \\ c \triangleleft b = \perp \\ a \triangleleft b = \perp \\ a \triangleleft c = a \end{array} \right. \\ \left\{ \begin{array}{l} a \cdot (b \triangleleft c) = a \\ a \cdot (c \triangleleft b) = \perp \\ a \triangleleft b = \perp \\ a \triangleleft c = a \end{array} \right. \end{array}$$

$$(a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) = a$$

(b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{l}
 b \triangle c = \perp \\
 a \triangleleft (b \triangle c) = a
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 b \triangleleft c = \perp \\
 c \triangleleft b = c \\
 a \triangleleft b = \perp \\
 a \triangleleft c = \perp \\
 a \cdot (b \triangleleft c) = \perp \\
 a \cdot (c \triangleleft b) = a \\
 a \triangleleft b = \perp \\
 a \triangleleft c = \perp
 \end{array}
 \right.$$

$$(a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) = a$$

(c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{l}
 b \triangle c = \perp \\
 a \triangleleft (b \triangle c) = a
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 b \triangleleft c = \perp \\
 c \triangleleft b = c \\
 a \triangleleft b = a \\
 a \triangleleft c = \perp \\
 a \cdot (b \triangleleft c) = \perp \\
 a \cdot (c \triangleleft b) = a \\
 a \triangleleft b = a \\
 a \triangleleft c = \perp
 \end{array}
 \right.$$

$$(a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) = a$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{l}
 b \triangle c = \perp \\
 a \triangleleft (b \triangle c) = a
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 b \triangleleft c = b \\
 c \triangleleft b = \perp \\
 a \triangleleft b = \perp \\
 a \triangleleft c = \perp \\
 a \cdot (b \triangleleft c) = a \\
 a \cdot (c \triangleleft b) = \perp \\
 a \triangleleft b = \perp \\
 a \triangleleft c = \perp
 \end{array}
 \right.$$

$$(a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) = a$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{l}
 b \triangle c = b \\
 a \triangleleft (b \triangle c) = \perp
 \end{array}
 \quad
 \begin{array}{l}
 \left\{ \begin{array}{l}
 b \triangleleft c = \perp \\
 c \triangleleft b = \perp \\
 a \triangleleft b = \perp \\
 a \triangleleft c = \perp
 \end{array} \right. \\
 \left\{ \begin{array}{l}
 a \cdot (b \triangleleft c) = \perp \\
 a \cdot (c \triangleleft b) = \perp \\
 a \triangleleft b = \perp \\
 a \triangleleft c = \perp
 \end{array} \right.
 \end{array}$$

$$(a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) = \perp$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{l}
 b \triangle c = b \\
 a \triangleleft (b \triangle c) = a
 \end{array}
 \quad
 \begin{array}{l}
 \left\{ \begin{array}{l}
 b \triangleleft c = \perp \\
 c \triangleleft b = \perp \\
 a \triangleleft b = a \\
 a \triangleleft c = a
 \end{array} \right. \\
 \left\{ \begin{array}{l}
 a \cdot (b \triangleleft c) = \perp \\
 a \cdot (c \triangleleft b) = \perp \\
 a \triangleleft b = a \\
 a \triangleleft c = a
 \end{array} \right.
 \end{array}$$

$$(a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) = a$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{l}
 b \triangle c = \perp \\
 a \triangleleft (b \triangle c) = a
 \end{array}
 \quad
 \begin{array}{l}
 \left\{ \begin{array}{l}
 b \triangleleft c = b \\
 c \triangleleft b = \perp \\
 a \triangleleft b = a \\
 a \triangleleft c = a
 \end{array} \right. \\
 \left\{ \begin{array}{l}
 a \cdot (b \triangleleft c) = b \\
 a \cdot (c \triangleleft b) = \perp \\
 a \triangleleft b = a \\
 a \triangleleft c = a
 \end{array} \right.
 \end{array}$$

$$(a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) = a$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{l}
 b \triangle c = \perp \\
 a \triangleleft (b \triangle c) = a
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 b \triangleleft c = \perp \\
 c \triangleleft b = c \\
 a \triangleleft b = a \\
 a \triangleleft c = a
 \end{array}
 \right.$$

$$\left\{
 \begin{array}{l}
 a \cdot (b \triangleleft c) = \perp \\
 a \cdot (c \triangleleft b) = c \\
 a \triangleleft b = a \\
 a \triangleleft c = a
 \end{array}
 \right.$$

$$(a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) = a$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{l}
 b \triangle c = \perp \\
 a \triangleleft (b \triangle c) = a
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 b \triangleleft c = b \\
 c \triangleleft b = \perp \\
 a \triangleleft b = \perp \\
 a \triangleleft c = a
 \end{array}
 \right.$$

$$\left\{
 \begin{array}{l}
 a \cdot (b \triangleleft c) = a \\
 a \cdot (c \triangleleft b) = \perp \\
 a \triangleleft b = \perp \\
 a \triangleleft c = a
 \end{array}
 \right.$$

$$(a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) = a$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{l}
 b \triangle c = \perp \\
 a \triangleleft (b \triangle c) = a
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 b \triangleleft c = b \\
 c \triangleleft b = \perp \\
 a \triangleleft b = \perp \\
 a \triangleleft c = \perp
 \end{array}
 \right.$$

$$\left\{
 \begin{array}{l}
 a \cdot (b \triangleleft c) = a \\
 a \cdot (c \triangleleft b) = \perp \\
 a \triangleleft b = \perp \\
 a \triangleleft c = \perp
 \end{array}
 \right.$$

$$(a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) = a$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{l}
 b \triangle c = \perp \\
 a \triangleleft (b \triangle c) = a
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 b \triangleleft c = \perp \\
 c \triangleleft b = c \\
 a \triangleleft b = a \\
 a \triangleleft c = \perp
 \end{array}
 \right.$$

$$\left\{
 \begin{array}{l}
 a \cdot (b \triangleleft c) = \perp \\
 a \cdot (c \triangleleft b) = a \\
 a \triangleleft b = a \\
 a \triangleleft c = \perp
 \end{array}
 \right.$$

$$(a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) = a$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{l}
 b \triangle c = \perp \\
 a \triangleleft (b \triangle c) = a
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 b \triangleleft c = \perp \\
 c \triangleleft b = c \\
 a \triangleleft b = \perp \\
 a \triangleleft c = \perp
 \end{array}
 \right.$$

$$\left\{
 \begin{array}{l}
 a \cdot (b \triangleleft c) = \perp \\
 a \cdot (c \triangleleft b) = a \\
 a \triangleleft b = \perp \\
 a \triangleleft c = \perp
 \end{array}
 \right.$$

$$(a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) = a$$

For all the cases of the domain of our study,

$$a \triangleleft (b \triangle c) = (a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c)$$

Thus:

$$a \triangleleft (b \triangle c) = (a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c)$$

End of the proof of Theorem (3.25).

Proof of Theorem (3.26) : $a \triangleleft (b \trianglelefteq c) = (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b))$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.2) : a \cdot b = b \cdot a$$

$$(3.4) : a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

$$(3.6) : a \cdot a = a$$

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.8) : a + \perp = a$$

$$(3.10) : a \cdot \perp = \perp$$

$$(3.13) : a + (a \cdot b) = a$$

$$(3.14) : a \cdot (a + b) = a$$

$$(3.16) : a \trianglelefteq b = a \triangleleft b + a \triangle b$$

$$(3.18) : a \triangleleft (b \triangleleft c) = (a \triangleleft b) + (a \cdot b \cdot (c \trianglelefteq b))$$

$$(3.23) : a \triangleleft (b + c) = (a \triangleleft b) \cdot (a \triangleleft c)$$

$$(3.25) : a \triangleleft (b \triangle b c) = (a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c)$$

$$(3.33) : a \cdot (a \triangleleft b) = a \triangleleft b$$

$$(3.35) : a \triangle b = b \triangle a$$

$$(3.44) : a \cdot (a \triangle b) = a \triangle b$$

$$(3.70) : (a \triangleleft b) \cdot (b \triangle c) = (a \triangleleft c) \cdot (b \triangle c)$$

$$(3.71) : (a \trianglelefteq b) \cdot (a \triangleleft b) = a \triangleleft b$$

$$(3.72) : (a \triangleleft b) \cdot (b \trianglelefteq a) = \perp$$

$$\begin{aligned}
 a \triangleleft (b \trianglelefteq c) &\stackrel{(3.16)}{=} a \triangleleft ((b \triangleleft c) + (b \triangle c)) \\
 &\stackrel{(3.23)}{=} (a \triangleleft (b \triangleleft c)) \cdot (a \triangleleft (b \triangle c)) \\
 &\stackrel{(3.18), (3.25)}{=} ((a \triangleleft b) + (a \cdot b \cdot (c \trianglelefteq b))) \cdot ((a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) \\
 &\quad + (a \triangleleft b) + (a \triangleleft c)) \\
 &\stackrel{(3.7)}{=} (a \triangleleft b) \cdot ((a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) \\
 &\quad + (a \triangleleft c)) + (a \cdot b \cdot (c \trianglelefteq b)) \cdot ((a \cdot (b \triangleleft c)) \\
 &\quad + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c)) \\
 &\stackrel{(3.1)}{=} (a \triangleleft b) \cdot ((a \triangleleft b) + (a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) \\
 &\quad + (a \triangleleft c)) + (a \cdot b \cdot (c \trianglelefteq b)) \cdot ((a \cdot (b \triangleleft c)) \\
 &\quad + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c))
 \end{aligned}$$

$$\begin{aligned}
a \triangleleft (b \trianglelefteq c) &\stackrel{(3.7),(3.14)}{=} (a \triangleleft b) + (a \cdot b \cdot (c \trianglelefteq b)) \cdot (a \cdot (b \triangleleft c)) \\
&\quad + (a \cdot b \cdot (c \trianglelefteq b)) \cdot (a \cdot (c \triangleleft b)) + (a \cdot b \cdot (c \trianglelefteq b)) \\
&\quad \cdot (a \triangleleft b) + (a \cdot b \cdot (c \trianglelefteq b)) \cdot (a \triangleleft c) \\
&\stackrel{(3.2),(3.4)}{=} (a \triangleleft b) + (a \cdot a \cdot b) \cdot ((b \triangleleft c) \cdot (c \trianglelefteq b)) \\
&\quad + (a \cdot a \cdot b) \cdot ((c \triangleleft b) \cdot (c \trianglelefteq b)) + (b \cdot (c \trianglelefteq b)) \\
&\quad \cdot (a \cdot (a \triangleleft b)) + (b \cdot (c \trianglelefteq b)) \cdot (a \cdot (a \triangleleft c)) \\
&\stackrel{(3.33),(3.71),(3.72)}{=} (a \triangleleft b) + (a \cdot a \cdot b) \cdot \perp + (a \cdot a \cdot b) \cdot (c \triangleleft b) \\
&\quad + (b \cdot (c \trianglelefteq b)) \cdot (a \triangleleft b) + (b \cdot (c \trianglelefteq b)) \cdot (a \triangleleft c) \\
&\stackrel{(3.6),(3.10)}{=} (a \triangleleft b) + \perp + (a \cdot b) \cdot (c \triangleleft b) + (b \cdot (c \trianglelefteq b)) \cdot (a \triangleleft b) \\
&\quad + (b \cdot (c \trianglelefteq b)) \cdot (a \triangleleft c) \\
&\stackrel{(3.4)}{=} (a \triangleleft b) + \perp + (a \cdot b \cdot (c \triangleleft b)) + (b \cdot (c \trianglelefteq b)) \cdot (a \triangleleft b) \\
&\quad + (b \cdot (c \trianglelefteq b)) \cdot (a \triangleleft c) \\
&\stackrel{(3.1),(3.2)}{=} (a \triangleleft b) + (a \triangleleft b) \cdot (b \cdot (c \trianglelefteq b)) + (a \cdot b \cdot (c \triangleleft b)) + \\
&\quad + (b \cdot (c \trianglelefteq b)) \cdot (a \triangleleft c) + \perp \\
&\stackrel{(3.8),(3.13)}{=} (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b)) + (b \cdot (c \trianglelefteq b)) \cdot (a \triangleleft c) \\
&\stackrel{(3.2)}{=} (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b)) + (a \triangleleft c) \cdot (b \cdot (c \trianglelefteq b)) \\
&\stackrel{(3.16)}{=} (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b)) \\
&\quad + (a \triangleleft c) \cdot (b \cdot ((c \triangleleft b) + (c \triangleleft b))) \\
&\stackrel{(3.7)}{=} (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b)) + (a \triangleleft c) \cdot (b \cdot (c \triangleleft b)) \\
&\quad + (a \triangleleft c) \cdot (b \cdot (c \triangleleft b)) \\
&\stackrel{(3.35),(3.44)}{=} (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b)) + (a \triangleleft c) \cdot (b \cdot (c \triangleleft b)) \\
&\quad + (a \triangleleft c) \cdot (b \triangleleft c) \\
&\stackrel{(3.33)}{=} (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b)) + (a \cdot (a \triangleleft c)) \cdot (b \cdot (c \triangleleft b)) \\
&\quad + (a \triangleleft c) \cdot (b \triangleleft c) \\
&\stackrel{(3.2),(3.4)}{=} (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b)) + (a \cdot b \cdot (c \triangleleft b)) \cdot (a \triangleleft c) \\
&\quad + (a \triangleleft c) \cdot (b \triangleleft c) \\
&\stackrel{(3.8),(3.13)}{=} (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b)) + (a \triangleleft c) \cdot (b \triangleleft c) \\
&\stackrel{(3.70)}{=} (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b)) + (a \triangleleft b) \cdot (b \triangleleft c) \\
&\stackrel{(3.8),(3.13)}{=} (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b))
\end{aligned}$$

End of the proof of Theorem (3.26).

Proof of Theorem (3.27) : $(a + b) \triangleleft c = (a \triangleleft c) + (b \triangleleft c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{array}{ll} a + b = a & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a + b) \triangleleft c = \perp & (a \triangleleft c) + (b \triangleleft c) = \perp \end{array}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{ll} a + b = a & \begin{cases} a \triangleleft c = a \\ b \triangleleft c = b \end{cases} \\ (a + b) \triangleleft c = a & (a \triangleleft c) + (b \triangleleft c) = a \end{array}$$

- (b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{ll} a + b = a & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a + b) \triangleleft c = \perp & (a \triangleleft c) + (b \triangleleft c) = \perp \end{array}$$

- (c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{ll} a + b = a & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a + b) \triangleleft c = \perp & (a \triangleleft c) + (b \triangleleft c) = \perp \end{array}$$

- (d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{ll} a + b = b & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = b \end{cases} \\ (a + b) \triangleleft c = b & (a \triangleleft c) + (b \triangleleft c) = b \end{array}$$

- (e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{ll} a + b = b & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a + b) \triangleleft c = \perp & (a \triangleleft c) + (b \triangleleft c) = \perp \end{array}$$

- (f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{ll} a + b = a & \begin{cases} a \triangleleft c = a \\ b \triangleleft c = \perp \end{cases} \\ (a + b) \triangleleft c = a & (a \triangleleft c) + (b \triangleleft c) = a \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{ll} a + b = a & \begin{cases} a \triangleleft c = a \\ b \triangleleft c = b \end{cases} \\ (a + b) \triangleleft c = a & (a \triangleleft c) + (b \triangleleft c) = a \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{ll} a + b = a & \begin{cases} a \triangleleft c = a \\ b \triangleleft c = \perp \end{cases} \\ (a + b) \triangleleft c = a & (a \triangleleft c) + (b \triangleleft c) = a \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{ll} a + b = b & \begin{cases} a \triangleleft c = a \\ b \triangleleft c = b \end{cases} \\ (a + b) \triangleleft c = b & (a \triangleleft c) + (b \triangleleft c) = b \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{ll} a + b = b & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = b \end{cases} \\ (a + b) \triangleleft c = b & (a \triangleleft c) + (b \triangleleft c) = b \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{ll} a + b = a & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a + b) \triangleleft c = \perp & (a \triangleleft c) + (b \triangleleft c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{ll} a + b = b & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a + b) \triangleleft c = \perp & (a \triangleleft c) + (b \triangleleft c) = \perp \end{array}$$

For all the cases of the domain of our study, $(a + b) \triangleleft c = (a \triangleleft c) + (b \triangleleft c)$. Thus:

$$(a + b) \triangleleft c = (a \triangleleft c) + (b \triangleleft c)$$

End of the proof of Theorem (3.27).

Proof of Theorem (3.28) : $(a \cdot b) \triangleleft c = (a \triangleleft c) \cdot (b \triangleleft c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$
By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a \cdot b) \triangleleft c = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \triangleleft c = a \\ b \triangleleft c = b \end{cases} \\ (a \cdot b) \triangleleft c = a & (a \triangleleft c) \cdot (b \triangleleft c) = a \end{array}$$

- (b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a \cdot b) \triangleleft c = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

- (c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{ll} a \cdot b = b & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a \cdot b) \triangleleft c = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

- (d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = b \end{cases} \\ (a \cdot b) \triangleleft c = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

- (e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a \cdot b) \triangleleft c = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{ll} a \cdot b = b & \begin{cases} a \triangleleft c = a \\ b \triangleleft c = \perp \end{cases} \\ (a \cdot b) \triangleleft c = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{ll} a \cdot b = b & \begin{cases} a \triangleleft c = a \\ b \triangleleft c = b \end{cases} \\ (a \cdot b) \triangleleft c = b & (a \triangleleft c) \cdot (b \triangleleft c) = b \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{ll} a \cdot b = b & \begin{cases} a \triangleleft c = a \\ b \triangleleft c = \perp \end{cases} \\ (a \cdot b) \triangleleft c = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \triangleleft c = a \\ b \triangleleft c = b \end{cases} \\ (a \cdot b) \triangleleft c = a & (a \triangleleft c) \cdot (b \triangleleft c) = a \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = b \end{cases} \\ (a \cdot b) \triangleleft c = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{ll} a \cdot b = b & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a \cdot b) \triangleleft c = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a \cdot b) \triangleleft c = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

For all the cases of the domain of our study, $(a \cdot b) \triangleleft c = (a \triangleleft c) \cdot (b \triangleleft c)$. Thus:

$$(a \cdot b) \triangleleft c = (a \triangleleft c) \cdot (b \triangleleft c)$$

End of the proof of Theorem (3.28).

Proof of Theorem (3.29) :

$$(a \triangle b) \triangleleft c = (a \triangle b) \cdot (a \triangleleft c) = (a \triangle b) \cdot (b \triangleleft c) = (a \triangleleft c) \triangle (b \triangleleft c)$$

This proof will be done in 3 steps:

- Step 1: $(a \triangle b) \triangleleft c = (a \triangle b) \cdot (a \triangleleft c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{array}{ll} a \triangle b = a & a \triangleleft c = \perp \\ (a \triangle b) \triangleleft c = \perp & (a \triangle b) \cdot (a \triangleleft c) = \perp \end{array}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{ll} a \triangle b = a & a \triangleleft c = a \\ (a \triangle b) \triangleleft c = a & (a \triangle b) \cdot (a \triangleleft c) = a \end{array}$$

- (b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{ll} a \triangle b = a & a \triangleleft c = \perp \\ (a \triangle b) \triangleleft c = \perp & (a \triangle b) \cdot (a \triangleleft c) = \perp \end{array}$$

- (c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & a \triangleleft c = \perp \\ (a \triangle b) \triangleleft c = \perp & (a \triangle b) \cdot (a \triangleleft c) = \perp \end{array}$$

- (d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & a \triangleleft c = \perp \\ (a \triangle b) \triangleleft c = \perp & (a \triangle b) \cdot (a \triangleleft c) = \perp \end{array}$$

- (e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & a \triangleleft c = \perp \\ (a \triangle b) \triangleleft c = \perp & (a \triangle b) \cdot (a \triangleleft c) = \perp \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{aligned} a \triangle b &= \perp & a \triangleleft c &= a \\ (a \triangle b) \triangleleft c &= \perp & (a \triangle b) \cdot (a \triangleleft c) &= \perp \end{aligned}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{aligned} a \triangle b &= \perp & a \triangleleft c &= a \\ (a \triangle b) \triangleleft c &= \perp & (a \triangle b) \cdot (a \triangleleft c) &= \perp \end{aligned}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{aligned} a \triangle b &= \perp & a \triangleleft c &= a \\ (a \triangle b) \triangleleft c &= \perp & (a \triangle b) \cdot (a \triangleleft c) &= \perp \end{aligned}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{aligned} a \triangle b &= \perp & a \triangleleft c &= a \\ (a \triangle b) \triangleleft c &= \perp & (a \triangle b) \cdot (a \triangleleft c) &= \perp \end{aligned}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{aligned} a \triangle b &= \perp & a \triangleleft c &= \perp \\ (a \triangle b) \triangleleft c &= \perp & (a \triangle b) \cdot (a \triangleleft c) &= \perp \end{aligned}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{aligned} a \triangle b &= \perp & a \triangleleft c &= \perp \\ (a \triangle b) \triangleleft c &= \perp & (a \triangle b) \cdot (a \triangleleft c) &= \perp \end{aligned}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{aligned} a \triangle b &= \perp & a \triangleleft c &= \perp \\ (a \triangle b) \triangleleft c &= \perp & (a \triangle b) \cdot (a \triangleleft c) &= \perp \end{aligned}$$

For all the cases of the domain of our study, $(a \triangle b) \triangleleft c = (a \triangle b) \cdot (a \triangleleft c)$.
Thus:

$$(a \triangle b) \triangleleft c = (a \triangle b) \cdot (a \triangleleft c)$$

- Step 2: $(a \triangle b) \triangleleft c = (a \triangle b) \cdot (b \triangleleft c)$

For this proof, we will use the following theorems:

$$(3.29) \text{ part 1 : } (a \triangle b) \triangleleft c = (a \triangle b) \cdot (a \triangleleft c)$$

$$(3.35) : a \triangle b = b \triangle a$$

$$(a \triangle b) \triangleleft c \stackrel{(3.35)}{=} (b \triangle a) \triangleleft c \stackrel{(3.29)}{=} (b \triangle a) \cdot (b \triangleleft c) \stackrel{(3.35)}{=} (a \triangle b) \cdot (b \triangleleft c)$$

- Step 3: $(a \triangle b) \triangleleft c = (a \triangleleft c) \triangle (b \triangleleft c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{array}{ll} a \triangle b = a & \left\{ \begin{array}{l} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangle b) \triangleleft c = \perp & (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{ll} a \triangle b = a & \left\{ \begin{array}{l} a \triangleleft c = a \\ b \triangleleft c = b \end{array} \right. \\ (a \triangle b) \triangleleft c = a & (a \triangleleft c) \triangle (b \triangleleft c) = a \end{array}$$

- (b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{ll} a \triangle b = a & \left\{ \begin{array}{l} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangle b) \triangleleft c = \perp & (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

- (c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & \left\{ \begin{array}{l} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangle b) \triangleleft c = \perp & (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

- (d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & \left\{ \begin{array}{l} a \triangleleft c = \perp \\ b \triangleleft c = b \end{array} \right. \\ (a \triangle b) \triangleleft c = \perp & (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{l} a \triangle b = \perp \\ (a \triangle b) \triangleleft c = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{l} a \triangle b = \perp \\ (a \triangle b) \triangleleft c = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft c = a \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{l} a \triangle b = \perp \\ (a \triangle b) \triangleleft c = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft c = a \\ b \triangleleft c = b \end{array} \right. \\ (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{l} a \triangle b = \perp \\ (a \triangle b) \triangleleft c = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft c = a \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{l} a \triangle b = \perp \\ (a \triangle b) \triangleleft c = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft c = a \\ b \triangleleft c = b \end{array} \right. \\ (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{l} a \triangle b = \perp \\ (a \triangle b) \triangleleft c = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft c = \perp \\ b \triangleleft c = b \end{array} \right. \\ (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{l} a \triangle b = \perp \\ (a \triangle b) \triangleleft c = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{l} a \triangle b = \perp \\ (a \triangle b) \triangleleft c = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

For all the cases of the domain of our study, $(a \triangle b) \triangleleft c = (a \triangleleft c) \triangle (b \triangleleft c)$.
Thus:

$$(a \triangle b) \triangleleft c = (a \triangleleft c) \triangle (b \triangleleft c)$$

End of the proof of Theorem (3.29).

Proof of Theorem (3.30) : $(a \triangleleft b) \triangleleft c = (a \triangleleft b) \cdot (a \triangleleft c)$

For this proof, we will use the following theorems:

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.16) : a \triangleleft b = a \triangleleft b + a \triangle b$$

$$(3.19) : (a \triangleleft b) \triangleleft c = (a \triangleleft b) \cdot (a \triangleleft c)$$

$$(3.27) : (a + b) \triangleleft c = (a \triangleleft c) + (b \triangleleft c)$$

$$(3.29) : (a \triangle b) \triangleleft c = (a \triangle b) \cdot (a \triangleleft c)$$

$$\begin{aligned} (a \triangleleft b) \triangleleft c &\stackrel{(3.16)}{=} ((a \triangleleft b) + (a \triangle b)) \triangleleft c \\ &\stackrel{(3.27)}{=} ((a \triangleleft b) \triangleleft c) + ((a \triangle b) \triangleleft c) \\ &\stackrel{(3.19), (3.29)}{=} (a \triangleleft b) \cdot (a \triangleleft c) + (a \triangle b) \cdot (a \triangleleft c) \\ &\stackrel{(3.7)}{=} ((a \triangleleft b) + (a \triangle b)) \cdot (a \triangleleft c) \\ &\stackrel{(3.16)}{=} (a \triangleleft b) \cdot (a \triangleleft c) \end{aligned}$$

End of the proof of Theorem (3.30).

Proof of Theorem (3.31) : $a + (a \triangleleft b) = a$

1. Case 1: the two dates are equal: $d(a) = d(b)$, meaning that $a = b$

By definition:

$$\begin{aligned} a \triangleleft b &= \perp \\ a + (a \triangleleft b) &= a \end{aligned}$$

2. Case 2: the two dates are different

(a) Case 2a: $d(a) < d(b)$

By definition:

$$\begin{aligned} a \triangleleft b &= a \\ a + (a \triangleleft b) &= a \end{aligned}$$

(b) Case 2b: $d(a) > d(b)$

By definition:

$$\begin{aligned} a \triangleleft b &= \perp \\ a + (a \triangleleft b) &= a \end{aligned}$$

For all the cases of the domain of our study, $a + (a \triangleleft b) = a$. Thus:

$$a + (a \triangleleft b) = a$$

End of the proof of Theorem (3.31).

Proof of Theorem (3.32) : $(a \triangleleft b) + b = a + b$

1. Case 1: the two dates are equal: $d(a) = d(b)$, meaning that $a = b$

By definition:

$$\begin{aligned} a \triangleleft b &= \perp & a + b &= a \\ (a \triangleleft b) + b &= b \end{aligned}$$

Given $a = b$, $(a \triangleleft b) + b = a + b$.

2. Case 2: the two dates are different

(a) Case 2a: $d(a) < d(b)$

By definition:

$$\begin{aligned} a \triangleleft b &= a & a + b &= a \\ (a \triangleleft b) + b &= a \end{aligned}$$

(b) Case 2b: $d(a) > d(b)$

By definition:

$$\begin{aligned} a \triangleleft b &= \perp & a + b &= b \\ (a \triangleleft b) + b &= b \end{aligned}$$

For all the cases of the domain of our study, $(a \triangleleft b) + b = a + b$. Thus:

$$(a \triangleleft b) + b = a + b$$

End of the proof of Theorem (3.32).

Proof of Theorem (3.33) : $a \cdot (a \triangleleft b) = a \triangleleft b$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.2) : a \cdot b = b \cdot a$$

$$(3.14) : a \cdot (a + b) = a$$

$$(3.31) : a + (a \triangleleft b) = a$$

$$a \cdot (a \triangleleft b) \stackrel{(3.31)}{=} (a + (a \triangleleft b)) \cdot (a \triangleleft b) \stackrel{(3.1),(3.2)}{=} (a \triangleleft b) \cdot ((a \triangleleft b) + a) \stackrel{(3.14)}{=} a \triangleleft b$$

End of the proof of Theorem (3.33).

Proof of Theorem (3.34) : $(a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = (a \triangleleft b) \cdot (b \triangleleft c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{array}{cc} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = \perp \\ a \triangleleft c = \perp \end{array} \right. & \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = \perp & (a \triangleleft b) \cdot (b \triangleleft c) = \perp \end{array}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{cc} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = b \\ a \triangleleft c = a \end{array} \right. & \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = b \end{array} \right. \\ (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = \perp & (a \triangleleft b) \cdot (b \triangleleft c) = \perp \end{array}$$

- (b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{cc} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = \perp \\ a \triangleleft c = \perp \end{array} \right. & \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = \perp & (a \triangleleft b) \cdot (b \triangleleft c) = \perp \end{array}$$

- (c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{cc} \left\{ \begin{array}{l} a \triangleleft b = a \\ b \triangleleft c = \perp \\ a \triangleleft c = \perp \end{array} \right. & \left\{ \begin{array}{l} a \triangleleft b = a \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = \perp & (a \triangleleft b) \cdot (b \triangleleft c) = \perp \end{array}$$

- (d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{cc} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = b \\ a \triangleleft c = \perp \end{array} \right. & \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = b \end{array} \right. \\ (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = \perp & (a \triangleleft b) \cdot (b \triangleleft c) = \perp \end{array}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{ccc} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = \perp \\ a \triangleleft c = \perp \end{array} \right. & & \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = \perp & & (a \triangleleft b) \cdot (b \triangleleft c) = \perp \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{ccc} \left\{ \begin{array}{l} a \triangleleft b = a \\ b \triangleleft c = \perp \\ a \triangleleft c = a \end{array} \right. & & \left\{ \begin{array}{l} a \triangleleft b = a \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = \perp & & (a \triangleleft b) \cdot (b \triangleleft c) = \perp \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{ccc} \left\{ \begin{array}{l} a \triangleleft b = a \\ b \triangleleft c = b \\ a \triangleleft c = a \end{array} \right. & & \left\{ \begin{array}{l} a \triangleleft b = a \\ b \triangleleft c = b \end{array} \right. \\ (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = b & & (a \triangleleft b) \cdot (b \triangleleft c) = b \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{ccc} \left\{ \begin{array}{l} a \triangleleft b = a \\ b \triangleleft c = \perp \\ a \triangleleft c = a \end{array} \right. & & \left\{ \begin{array}{l} a \triangleleft b = a \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = \perp & & (a \triangleleft b) \cdot (b \triangleleft c) = \perp \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{ccc} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = b \\ a \triangleleft c = a \end{array} \right. & & \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = b \end{array} \right. \\ (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = \perp & & (a \triangleleft b) \cdot (b \triangleleft c) = \perp \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{ccc} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = b \\ a \triangleleft c = \perp \end{array} \right. & & \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = b \end{array} \right. \\ (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = \perp & & (a \triangleleft b) \cdot (b \triangleleft c) = \perp \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{ccc} \left\{ \begin{array}{l} a \triangleleft b = a \\ b \triangleleft c = \perp \\ a \triangleleft c = \perp \end{array} \right. & & \left\{ \begin{array}{l} a \triangleleft b = a \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = \perp & & (a \triangleleft b) \cdot (b \triangleleft c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{ccc} \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = \perp \\ a \triangleleft c = \perp \end{array} \right. & & \left\{ \begin{array}{l} a \triangleleft b = \perp \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = \perp & & (a \triangleleft b) \cdot (b \triangleleft c) = \perp \end{array}$$

For all the cases of the domain of our study, $(a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = (a \triangleleft b) \cdot (b \triangleleft c)$.
Thus:

$$(a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = (a \triangleleft b) \cdot (b \triangleleft c)$$

End of the proof of Theorem (3.34).

A.2.3 Proofs of the theorems satisfied by operator SIMULTANEOUS

Proof of Theorem (3.35) : $a \triangle b = b \triangle a$

1. Case 1: the two dates are equal: $d(a) = d(b)$, meaning that $a = b$

By definition:

$$a \triangle b = a \qquad b \triangle a = b$$

Given $a = b$, $a \triangle b = b \triangle a$.

2. Case 2: the two dates are different

(a) Case 2a: $d(a) < d(b)$

By definition:

$$a \triangle b = \perp \qquad b \triangle a = \perp$$

(b) Case 2b: $d(a) > d(b)$

By definition:

$$a \triangle b = \perp \qquad b \triangle a = \perp$$

For all the cases of the domain of our study, $a \triangle b = b \triangle a$. Thus:

$$a \triangle b = b \triangle a$$

End of the proof of Theorem (3.35).

Proof of Theorem (3.36) :

$$a \triangle (b \triangle c) = (a \triangle b) \triangle c = (a \triangle b) \cdot (b \triangle c) = (a \triangle c) \cdot (c \triangle b)$$

This proof will be done in 3 steps:

- Step 1: $a \triangle (b \triangle c) = (a \triangle b) \triangle c$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{array}{ll} a \triangle b = a & b \triangle c = b \\ (a \triangle b) \triangle c = a & a \triangle (b \triangle c) = a \end{array}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{ll} a \triangle b = a & b \triangle c = \perp \\ (a \triangle b) \triangle c = \perp & a \triangle (b \triangle c) = \perp \end{array}$$

- (b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{ll} a \triangle b = a & b \triangle c = \perp \\ (a \triangle b) \triangle c = \perp & a \triangle (b \triangle c) = \perp \end{array}$$

- (c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & b \triangle c = \perp \\ (a \triangle b) \triangle c = \perp & a \triangle (b \triangle c) = \perp \end{array}$$

- (d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & b \triangle c = \perp \\ (a \triangle b) \triangle c = \perp & a \triangle (b \triangle c) = \perp \end{array}$$

- (e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & b \triangle c = b \\ (a \triangle b) \triangle c = \perp & a \triangle (b \triangle c) = \perp \end{array}$$

- (f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & b \triangle c = b \\ (a \triangle b) \triangle c = \perp & a \triangle (b \triangle c) = \perp \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & b \triangle c = \perp \\ (a \triangle b) \triangle c = \perp & a \triangle (b \triangle c) = \perp \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & b \triangle c = \perp \\ (a \triangle b) \triangle c = \perp & a \triangle (b \triangle c) = \perp \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & b \triangle c = \perp \\ (a \triangle b) \triangle c = \perp & a \triangle (b \triangle c) = \perp \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & b \triangle c = \perp \\ (a \triangle b) \triangle c = \perp & a \triangle (b \triangle c) = \perp \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & b \triangle c = \perp \\ (a \triangle b) \triangle c = \perp & a \triangle (b \triangle c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{ll} a \triangle b = \perp & b \triangle c = \perp \\ (a \triangle b) \triangle c = \perp & a \triangle (b \triangle c) = \perp \end{array}$$

For all the cases of the domain of our study, $(a \triangle b) \triangle c = a \triangle (b \triangle c)$. Thus:

$$(a \triangle b) \triangle c = a \triangle (b \triangle c)$$

• Step 2: $a \triangle (b \triangle c) = (a \triangle b) \cdot (b \triangle c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{array}{ll} b \triangle c = b & \begin{cases} a \triangle b = a \\ b \triangle c = b \end{cases} \\ a \triangle (b \triangle c) = a & (a \triangle b) \cdot (b \triangle c) = a \end{array}$$

2. Case 2: two dates out of three are equal

(a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{aligned} b \triangle c &= \perp & \begin{cases} a \triangle b = a \\ b \triangle c = \perp \end{cases} \\ a \triangle (b \triangle c) &= \perp & (a \triangle b) \cdot (b \triangle c) = \perp \end{aligned}$$

(b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{aligned} b \triangle c &= \perp & \begin{cases} a \triangle b = a \\ b \triangle c = \perp \end{cases} \\ a \triangle (b \triangle c) &= \perp & (a \triangle b) \cdot (b \triangle c) = \perp \end{aligned}$$

(c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{aligned} b \triangle c &= \perp & \begin{cases} a \triangle b = \perp \\ b \triangle c = \perp \end{cases} \\ a \triangle (b \triangle c) &= \perp & (a \triangle b) \cdot (b \triangle c) = \perp \end{aligned}$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{aligned} b \triangle c &= \perp & \begin{cases} a \triangle b = \perp \\ b \triangle c = \perp \end{cases} \\ a \triangle (b \triangle c) &= \perp & (a \triangle b) \cdot (b \triangle c) = \perp \end{aligned}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{aligned} b \triangle c &= b & \begin{cases} a \triangle b = \perp \\ b \triangle c = b \end{cases} \\ a \triangle (b \triangle c) &= \perp & (a \triangle b) \cdot (b \triangle c) = \perp \end{aligned}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{aligned} b \triangle c &= b & \begin{cases} a \triangle b = \perp \\ b \triangle c = b \end{cases} \\ a \triangle (b \triangle c) &= \perp & (a \triangle b) \cdot (b \triangle c) = \perp \end{aligned}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{aligned} b \triangle c &= \perp & \begin{cases} a \triangle b = \perp \\ b \triangle c = \perp \end{cases} \\ a \triangle (b \triangle c) &= \perp & (a \triangle b) \cdot (b \triangle c) = \perp \end{aligned}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{l} b \triangle c = \perp \\ a \triangle (b \triangle c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{l} b \triangle c = \perp \\ a \triangle (b \triangle c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{l} b \triangle c = \perp \\ a \triangle (b \triangle c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{l} b \triangle c = \perp \\ a \triangle (b \triangle c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{l} b \triangle c = \perp \\ a \triangle (b \triangle c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

For all the cases of the domain of our study, $a \triangle (b \triangle c) = (a \triangle b) \cdot (b \triangle c)$.

Thus:

$$a \triangle (b \triangle c) = (a \triangle b) \cdot (b \triangle c)$$

- Step 3: $(a \triangle b) \cdot (b \triangle c) = (a \triangle c) \cdot (c \triangle b)$

For this proof, we will use the following theorems:

$$(3.2) : a \cdot b = b \cdot a$$

$$(3.35) : a \triangle b = b \triangle a$$

$$(3.45) : (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = (a \triangle b) \cdot (b \triangle c)$$

$$\begin{aligned} (a \triangle b) \cdot (b \triangle c) &\stackrel{(3.45)}{=} (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) \\ &\stackrel{(3.2)}{=} (a \triangle c) \cdot (b \triangle c) \cdot (a \triangle b) \\ &\stackrel{(3.35)}{=} (a \triangle c) \cdot (c \triangle b) \cdot (a \triangle b) \stackrel{(3.45)}{=} (a \triangle c) \cdot (c \triangle b) \end{aligned}$$

End of the proof of Theorem (3.36).

Proof of Theorem (3.37) : $a \triangle \perp = \perp$

1. Case 1: $d(a) = d(\perp) = +\infty$

By definition:

$$a \triangle \perp = a = \perp$$

2. Case 2: $d(a) \neq d(\perp)$, meaning that $d(a) < d(\perp)$

By definition:

$$a \triangle \perp = \perp$$

For all the cases of the domain of our study, $a \triangle \perp = \perp$. Thus:

$$a \triangle \perp = \perp$$

End of the proof of Theorem (3.37).

Proof of Theorem (3.38) : $a \triangle a = a$

By definition, $a \triangle b = a$ if $d(a) = d(b)$. Thus, if $b = a$, $d(b) = d(a)$ and:

$$a \triangle a = a$$

End of the proof of Theorem (3.38).

Proof of Theorem (3.39) :

$$\begin{aligned} a \triangle (b + c) &= (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) + (a \triangle c) \cdot (c \triangleleft b) \\ &= (a \triangle b) \cdot (b \trianglelefteq c) + (a \triangle c) \cdot (c \trianglelefteq b) \end{aligned}$$

This proof will be done in 2 steps:

• Step 1: $a \triangle (b + c) = (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) + (a \triangle c) \cdot (c \triangleleft b)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{aligned} b + c &= b \\ a \triangle (b + c) &= a \end{aligned} \quad \left\{ \begin{array}{l} a \triangle b = a \\ b \triangleleft c = \perp \\ a \triangle c = a \\ c \triangleleft b = \perp \\ b \triangle c = b \end{array} \right. \begin{aligned} &(a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) \\ &+ (a \triangle c) \cdot (c \triangleleft b) = a \end{aligned}$$

2. Case 2: two dates out of three are equal

(a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{l}
 b + c = b \\
 a \triangle (b + c) = a
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = a \\
 b \triangleleft c = b \\
 a \triangle c = \perp \\
 c \triangleleft b = \perp \\
 b \triangle c = \perp
 \end{array}
 \right.
 \quad
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) \\
 + (a \triangle c) \cdot (c \triangleleft b) = a
 \end{array}$$

(b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{l}
 b + c = c \\
 a \triangle (b + c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = a \\
 b \triangleleft c = \perp \\
 a \triangle c = \perp \\
 c \triangleleft b = c \\
 b \triangle c = \perp
 \end{array}
 \right.
 \quad
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) \\
 + (a \triangle c) \cdot (c \triangleleft b) = \perp
 \end{array}$$

(c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{l}
 b + c = c \\
 a \triangle (b + c) = a
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 b \triangleleft c = \perp \\
 a \triangle c = a \\
 c \triangleleft b = c \\
 b \triangle c = \perp
 \end{array}
 \right.
 \quad
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) \\
 + (a \triangle c) \cdot (c \triangleleft b) = a
 \end{array}$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{l}
 b + c = b \\
 a \triangle (b + c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 b \triangleleft c = b \\
 a \triangle c = a \\
 c \triangleleft b = \perp \\
 b \triangle c = \perp
 \end{array}
 \right.
 \quad
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) \\
 + (a \triangle c) \cdot (c \triangleleft b) = \perp
 \end{array}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{l}
 b + c = b \\
 a \triangle (b + c) = \perp
 \end{array}
 \quad
 \left\{ \begin{array}{l}
 a \triangle b = \perp \\
 b \triangleleft c = \perp \\
 a \triangle c = \perp \\
 c \triangleleft b = \perp \\
 b \triangle c = b
 \end{array} \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) \\
 + (a \triangle c) \cdot (c \triangleleft b) = \perp
 \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{l}
 b + c = b \\
 a \triangle (b + c) = \perp
 \end{array}
 \quad
 \left\{ \begin{array}{l}
 a \triangle b = \perp \\
 b \triangleleft c = \perp \\
 a \triangle c = \perp \\
 c \triangleleft b = \perp \\
 b \triangle c = b
 \end{array} \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) \\
 + (a \triangle c) \cdot (c \triangleleft b) = \perp
 \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{l}
 b + c = b \\
 a \triangle (b + c) = \perp
 \end{array}
 \quad
 \left\{ \begin{array}{l}
 a \triangle b = \perp \\
 b \triangleleft c = b \\
 a \triangle c = \perp \\
 c \triangleleft b = \perp \\
 b \triangle c = \perp
 \end{array} \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) \\
 + (a \triangle c) \cdot (c \triangleleft b) = \perp
 \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{l}
 b + c = c \\
 a \triangle (b + c) = \perp
 \end{array}
 \quad
 \left\{ \begin{array}{l}
 a \triangle b = \perp \\
 b \triangleleft c = \perp \\
 a \triangle c = \perp \\
 c \triangleleft b = c \\
 b \triangle c = \perp
 \end{array} \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) \\
 + (a \triangle c) \cdot (c \triangleleft b) = \perp
 \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{l}
 b + c = b \\
 a \triangle (b + c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 b \triangleleft c = b \\
 a \triangle c = \perp \\
 c \triangleleft b = \perp \\
 b \triangle c = \perp
 \end{array}
 \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) \\
 + (a \triangle c) \cdot (c \triangleleft b) = \perp
 \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{l}
 b + c = b \\
 a \triangle (b + c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 b \triangleleft c = b \\
 a \triangle c = \perp \\
 c \triangleleft b = \perp \\
 b \triangle c = \perp
 \end{array}
 \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) \\
 + (a \triangle c) \cdot (c \triangleleft b) = \perp
 \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{l}
 b + c = c \\
 a \triangle (b + c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 b \triangleleft c = \perp \\
 a \triangle c = \perp \\
 c \triangleleft b = c \\
 b \triangle c = \perp
 \end{array}
 \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) \\
 + (a \triangle c) \cdot (c \triangleleft b) = \perp
 \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{l}
 b + c = c \\
 a \triangle (b + c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 b \triangleleft c = \perp \\
 a \triangle c = \perp \\
 c \triangleleft b = c \\
 b \triangle c = \perp
 \end{array}
 \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) \\
 + (a \triangle c) \cdot (c \triangleleft b) = \perp
 \end{array}$$

For all the cases of the domain of our study,

$$a \triangle (b + c) = (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) + (a \triangle c) \cdot (c \triangleleft b)$$

Thus:

$$a \triangle (b + c) = (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) + (a \triangle c) \cdot (c \triangleleft b)$$

- Step 2: $a \triangle (b + c) = (a \triangle b) \cdot (b \trianglelefteq c) + (a \triangle c) \cdot (c \trianglelefteq b)$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.3) : a + (b + c) = (a + b) + c$$

$$(3.5) : a + a = a$$

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.16) : a \trianglelefteq b = a \triangleleft b + a \triangle b$$

$$(3.36) : (a \triangle b) \cdot (b \triangle c) = (a \triangle c) \cdot (c \triangle b)$$

$$(3.39) \text{ part 1 : } a \triangle (b + c) = (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) + (a \triangle c) \cdot (c \triangleleft b)$$

$$\begin{aligned} a \triangle (b + c) &\stackrel{(3.39)}{=} (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c) + (a \triangle c) \cdot (c \triangleleft b) \\ &\stackrel{(3.5)}{=} ((a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c)) \\ &\quad + (a \triangle b) \cdot (b \triangleleft c) + (a \triangle c) \cdot (c \triangleleft b) \\ &\stackrel{(3.3)}{=} ((a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c)) \\ &\quad + ((a \triangle b) \cdot (b \triangle c) + (a \triangle c) \cdot (c \triangleleft b)) \\ &\stackrel{(3.36)}{=} ((a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangleleft c)) \\ &\quad + ((a \triangle c) \cdot (c \triangle b) + (a \triangle c) \cdot (c \triangleleft b)) \\ &\stackrel{(3.7)}{=} (a \triangle b) \cdot ((b \triangle c) + (b \triangleleft c)) + (a \triangle c) \cdot ((c \triangle b) + (c \triangleleft b)) \\ &\stackrel{(3.1)}{=} (a \triangle b) \cdot ((b \triangleleft c) + (b \triangle c)) + (a \triangle c) \cdot ((c \triangleleft b) + (c \triangle b)) \\ &\stackrel{(3.16)}{=} (a \triangle b) \cdot (b \trianglelefteq c) + (a \triangle c) \cdot (c \trianglelefteq b) \end{aligned}$$

End of the proof of Theorem (3.39).

Proof of Theorem (3.40) :

$$\begin{aligned} a \triangle (b \cdot c) &= (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) + (a \triangle c) \cdot (b \triangleleft c) \\ &= (a \triangle b) \cdot (c \trianglelefteq b) + (a \triangle c) \cdot (b \trianglelefteq c) \end{aligned}$$

This proof will be done in 2 steps:

- Step 1: $a \triangle (b \cdot c) = (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) + (a \triangle c) \cdot (b \triangleleft c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{array}{l}
 b \cdot c = b \\
 a \triangle (b \cdot c) = a
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = a \\
 c \triangleleft b = \perp \\
 a \triangle c = a \\
 b \triangleleft c = \perp \\
 b \triangle c = b
 \end{array}
 \right.
 \quad
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) \\
 + (a \triangle c) \cdot (b \triangleleft c) = a
 \end{array}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{l}
 b \cdot c = c \\
 a \triangle (b \cdot c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = a \\
 c \triangleleft b = \perp \\
 a \triangle c = \perp \\
 b \triangleleft c = b \\
 b \triangle c = \perp
 \end{array}
 \right.
 \quad
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) \\
 + (a \triangle c) \cdot (b \triangleleft c) = \perp
 \end{array}$$

- (b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{l}
 b \cdot c = b \\
 a \triangle (b \cdot c) = a
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = a \\
 c \triangleleft b = c \\
 a \triangle c = \perp \\
 b \triangleleft c = \perp \\
 b \triangle c = \perp
 \end{array}
 \right.
 \quad
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) \\
 + (a \triangle c) \cdot (b \triangleleft c) = a
 \end{array}$$

- (c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{l}
 b \cdot c = b \\
 a \triangle (b \cdot c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 c \triangleleft b = c \\
 a \triangle c = a \\
 b \triangleleft c = \perp \\
 b \triangle c = \perp
 \end{array}
 \right.
 \quad
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) \\
 + (a \triangle c) \cdot (b \triangleleft c) = \perp
 \end{array}$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{l}
 b \cdot c = c \\
 a \triangle (b \cdot c) = a
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 c \triangleleft b = \perp \\
 a \triangle c = a \\
 b \triangleleft c = b \\
 b \triangle c = \perp
 \end{array}
 \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) \\
 + (a \triangle c) \cdot (b \triangleleft c) = a
 \end{array}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{l}
 b \cdot c = b \\
 a \triangle (b \cdot c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 c \triangleleft b = \perp \\
 a \triangle c = \perp \\
 b \triangleleft c = \perp \\
 b \triangle c = b
 \end{array}
 \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) \\
 + (a \triangle c) \cdot (b \triangleleft c) = \perp
 \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{l}
 b \cdot c = b \\
 a \triangle (b \cdot c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 c \triangleleft b = \perp \\
 a \triangle c = \perp \\
 b \triangleleft c = \perp \\
 b \triangle c = b
 \end{array}
 \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) \\
 + (a \triangle c) \cdot (b \triangleleft c) = \perp
 \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{l}
 b \cdot c = c \\
 a \triangle (b \cdot c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 c \triangleleft b = \perp \\
 a \triangle c = \perp \\
 b \triangleleft c = b \\
 b \triangle c = \perp
 \end{array}
 \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) \\
 + (a \triangle c) \cdot (b \triangleleft c) = \perp
 \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{l}
 b \cdot c = b \\
 a \triangle (b \cdot c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 c \triangleleft b = c \\
 a \triangle c = \perp \\
 b \triangleleft c = \perp \\
 b \triangle c = \perp
 \end{array}
 \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) \\
 + (a \triangle c) \cdot (b \triangleleft c) = \perp
 \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{l}
 b \cdot c = c \\
 a \triangle (b \cdot c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 c \triangleleft b = \perp \\
 a \triangle c = \perp \\
 b \triangleleft c = b \\
 b \triangle c = \perp
 \end{array}
 \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) \\
 + (a \triangle c) \cdot (b \triangleleft c) = \perp
 \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{l}
 b \cdot c = c \\
 a \triangle (b \cdot c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 c \triangleleft b = \perp \\
 a \triangle c = \perp \\
 b \triangleleft c = b \\
 b \triangle c = \perp
 \end{array}
 \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) \\
 + (a \triangle c) \cdot (b \triangleleft c) = \perp
 \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{l}
 b \cdot c = b \\
 a \triangle (b \cdot c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 c \triangleleft b = c \\
 a \triangle c = \perp \\
 b \triangleleft c = \perp \\
 b \triangle c = \perp
 \end{array}
 \right.
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) \\
 + (a \triangle c) \cdot (b \triangleleft c) = \perp
 \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{l}
 b \cdot c = b \\
 a \triangle (b \cdot c) = \perp
 \end{array}
 \quad
 \left\{
 \begin{array}{l}
 a \triangle b = \perp \\
 c \triangleleft b = c \\
 a \triangle c = \perp \\
 b \triangleleft c = \perp \\
 b \triangle c = \perp
 \end{array}
 \right.
 \quad
 \begin{array}{l}
 (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) \\
 + (a \triangle c) \cdot (b \triangleleft c) = \perp
 \end{array}$$

For all the cases of the domain of our study,

$$a \triangle (b \cdot c) = (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) + (a \triangle c) \cdot (b \triangleleft c)$$

Thus:

$$a \triangle (b \cdot c) = (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) + (a \triangle c) \cdot (b \triangleleft c)$$

- Step 2: $a \triangle (b \cdot c) = (a \triangle b) \cdot (c \triangleleft b) + (a \triangle c) \cdot (b \triangleleft c)$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.3) : a + (b + c) = (a + b) + c$$

$$(3.5) : a + a = a$$

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.16) : a \triangleleft b = a \triangleleft b + a \triangle b$$

$$(3.35) : a \triangle b = b \triangle a$$

$$(3.36) : (a \triangle b) \cdot (b \triangle c) = (a \triangle c) \cdot (c \triangle b)$$

$$(3.40) \text{ part 1} : a \triangle (b \cdot c) = (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) + (a \triangle c) \cdot (b \triangleleft c)$$

$$\begin{aligned}
 a \triangle (b \cdot c) &\stackrel{(3.40)}{=} (a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b) + (a \triangle c) \cdot (b \triangleleft c) \\
 &\stackrel{(3.5)}{=} ((a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (b \triangle c)) \\
 &\quad + (a \triangle b) \cdot (c \triangleleft b) + (a \triangle c) \cdot (b \triangleleft c) \\
 &\stackrel{(3.3)}{=} ((a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b)) \\
 &\quad + ((a \triangle b) \cdot (b \triangle c) + (a \triangle c) \cdot (b \triangleleft c)) \\
 &\stackrel{(3.36)}{=} ((a \triangle b) \cdot (b \triangle c) + (a \triangle b) \cdot (c \triangleleft b)) \\
 &\quad + ((a \triangle c) \cdot (c \triangle b) + (a \triangle c) \cdot (b \triangleleft c))
 \end{aligned}$$

$$\begin{aligned}
a \triangle (b \cdot c) &\stackrel{(3.35)}{=} ((a \triangle b) \cdot (c \triangle b) + (a \triangle b) \cdot (c \triangleleft b)) \\
&\quad + ((a \triangle c) \cdot (b \triangle c) + (a \triangle c) \cdot (b \triangleleft c)) \\
&\stackrel{(3.7)}{=} (a \triangle b) \cdot ((c \triangle b) + (c \triangleleft b)) + (a \triangle c) \cdot ((b \triangle c) + (b \triangleleft c)) \\
&\stackrel{(3.1)}{=} (a \triangle b) \cdot ((c \triangleleft b) + (c \triangle b)) + (a \triangle c) \cdot ((b \triangleleft c) + (b \triangle c)) \\
&\stackrel{(3.16)}{=} (a \triangle b) \cdot (c \trianglelefteq b) + (a \triangle c) \cdot (b \trianglelefteq c)
\end{aligned}$$

End of the proof of Theorem (3.40).

Proof of Theorem (3.41) : $a \triangle (b \triangleleft c) = (a \triangle b) \cdot (b \triangleleft c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$
By definition:

$$\begin{array}{ll}
b \triangleleft c = \perp & a \triangle b = a \\
a \triangle (b \triangleleft c) = \perp & (a \triangle b) \cdot (b \triangleleft c) = \perp
\end{array}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{ll}
b \triangleleft c = b & a \triangle b = a \\
a \triangle (b \triangleleft c) = a & (a \triangle b) \cdot (b \triangleleft c) = a
\end{array}$$

- (b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{ll}
b \triangleleft c = \perp & a \triangle b = a \\
a \triangle (b \triangleleft c) = \perp & (a \triangle b) \cdot (b \triangleleft c) = \perp
\end{array}$$

- (c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{ll}
b \triangleleft c = \perp & a \triangle b = \perp \\
a \triangle (b \triangleleft c) = \perp & (a \triangle b) \cdot (b \triangleleft c) = \perp
\end{array}$$

- (d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{ll}
b \triangleleft c = b & a \triangle b = \perp \\
a \triangle (b \triangleleft c) = \perp & (a \triangle b) \cdot (b \triangleleft c) = \perp
\end{array}$$

- (e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{ll}
b \triangleleft c = \perp & a \triangle b = \perp \\
a \triangle (b \triangleleft c) = \perp & (a \triangle b) \cdot (b \triangleleft c) = \perp
\end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{ll} b \triangleleft c = \perp & a \triangle b = \perp \\ a \triangle (b \triangleleft c) = \perp & (a \triangle b) \cdot (b \triangleleft c) = \perp \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{ll} b \triangleleft c = b & a \triangle b = \perp \\ a \triangle (b \triangleleft c) = \perp & (a \triangle b) \cdot (b \triangleleft c) = \perp \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{ll} b \triangleleft c = \perp & a \triangle b = \perp \\ a \triangle (b \triangleleft c) = \perp & (a \triangle b) \cdot (b \triangleleft c) = \perp \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{ll} b \triangleleft c = b & a \triangle b = \perp \\ a \triangle (b \triangleleft c) = \perp & (a \triangle b) \cdot (b \triangleleft c) = \perp \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{ll} b \triangleleft c = b & a \triangle b = \perp \\ a \triangle (b \triangleleft c) = \perp & (a \triangle b) \cdot (b \triangleleft c) = \perp \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{ll} b \triangleleft c = \perp & a \triangle b = \perp \\ a \triangle (b \triangleleft c) = \perp & (a \triangle b) \cdot (b \triangleleft c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{ll} b \triangleleft c = \perp & a \triangle b = \perp \\ a \triangle (b \triangleleft c) = \perp & (a \triangle b) \cdot (b \triangleleft c) = \perp \end{array}$$

For all the cases of the domain of our study, $a \triangle (b \triangleleft c) = (a \triangle b) \cdot (b \triangleleft c)$. Thus:

$$a \triangle (b \triangleleft c) = (a \triangle b) \cdot (b \triangleleft c)$$

End of the proof of Theorem (3.41).

Proof of Theorem (3.42) : $a \triangle (b \triangleleft c) = (a \triangle b) \cdot (b \triangleleft c)$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.2) : a \cdot b = b \cdot a$$

$$(3.4) : a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

$$(3.6) : a \cdot a = a$$

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.8) : a + \perp = a$$

$$(3.14) : a \cdot (a + b) = a$$

$$(3.16) : a \triangleleft b = a \triangleleft b + a \triangle b$$

$$(3.22) : a \triangleleft a = \perp$$

$$(3.35) : a \triangle b = b \triangle a$$

$$(3.36) : a \triangle (b \triangle c) = (a \triangle b) \cdot (b \triangle c)$$

$$(3.38) : a \triangle a = a$$

$$(3.39) : a \triangle (b + c) = (a \triangle b) \cdot (b \triangleleft c) + (a \triangle c) \cdot (c \triangleleft b)$$

$$(3.41) : a \triangle (b \triangleleft c) = (a \triangle b) \cdot (b \triangleleft c)$$

$$(3.44) : a \cdot (a \triangle b) = a \triangle b$$

$$(3.54) : a \triangleleft (b \triangleleft c) = (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b)) + (a \triangle b) \cdot (b \triangleleft c)$$

$$(3.55) : a \triangleleft (b \triangle c) = (a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) \\ + (a \triangle b) \cdot (b \triangle c)$$

$$(3.58) : (a \triangle b) \triangleleft c = (a \triangle b) \cdot (a \triangleleft c)$$

$$(3.59) : (a \triangleleft b) \triangleleft c = (a \triangleleft b) \cdot (a \triangleleft c)$$

$$(3.69) : (a \triangleleft b) \cdot (a \triangle b) = \perp$$

$$(3.73) : (a \triangleleft b) \cdot (a \triangle b) = a \triangle b$$

$$\begin{aligned} a \triangle (b \triangleleft c) &\stackrel{(3.16)}{=} a \triangle ((b \triangleleft c) + (b \triangle c)) \\ &\stackrel{(3.39)}{=} (a \triangle (b \triangleleft c)) \cdot ((b \triangleleft c) \triangleleft (b \triangle c)) \\ &\quad + (a \triangle (b \triangle c)) \cdot ((b \triangle c) \triangleleft (b \triangleleft c)) \end{aligned}$$

$$\begin{aligned} a \triangle (b \triangleleft c) &\stackrel{(3.41)}{=} (a \triangle b) \cdot (b \triangleleft c) \\ (b \triangleleft c) \triangleleft (b \triangle c) &\stackrel{(3.59)}{=} (b \triangleleft c) \cdot (b \triangleleft (b \triangle c)) \\ &\stackrel{(3.55)}{=} (b \triangleleft c) \cdot ((b \cdot (b \triangleleft c)) + (b \cdot (c \triangleleft b)) + (b \triangleleft b) + (b \triangleleft c) \\ &\quad + (b \triangle b) \cdot (b \triangle c)) \\ &\stackrel{(3.1)}{=} (b \triangleleft c) \cdot ((b \triangleleft c) + (b \cdot (b \triangleleft c)) + (b \cdot (c \triangleleft b)) + (b \triangleleft b) \\ &\quad + (b \triangle b) \cdot (b \triangle c)) \stackrel{(3.14)}{=} b \triangleleft c \end{aligned}$$

$$\begin{aligned}
(a \triangle (b \triangleleft c)) \cdot ((b \triangleleft c) \trianglelefteq (b \triangle c)) &= ((a \triangle b) \cdot (b \triangleleft c)) \cdot (b \triangleleft c) \\
&\stackrel{(3.4)}{=} (a \triangle b) \cdot ((b \triangleleft c) \cdot (b \triangleleft c)) \\
&\stackrel{(3.6)}{=} (a \triangle b) \cdot (b \triangleleft c)
\end{aligned}$$

$$\begin{aligned}
a \triangle (b \triangle c) &\stackrel{(3.36)}{=} (a \triangle b) \cdot (b \triangle c) \\
(b \triangle c) \trianglelefteq (b \triangleleft c) &\stackrel{(3.58)}{=} (b \triangle c) \cdot (b \trianglelefteq (b \triangleleft c)) \\
&\stackrel{(3.54)}{=} (b \triangle c) \cdot ((b \triangleleft b) + (b \cdot b \cdot (c \trianglelefteq b)) \\
&\quad + (b \triangle b) \cdot (b \triangleleft c)) \\
&\stackrel{(3.6),(3.22)}{=} (b \triangle c) \cdot (\perp + (b \cdot (c \trianglelefteq b)) + (b \triangle b) \cdot (b \triangleleft c)) \\
&\stackrel{(3.1)}{=} (b \triangle c) \cdot ((b \cdot (c \trianglelefteq b)) + (b \triangle b) \cdot (b \triangleleft c) + \perp) \\
&\stackrel{(3.8)}{=} (b \triangle c) \cdot ((b \cdot (c \trianglelefteq b)) + (b \triangle b) \cdot (b \triangleleft c)) \\
&\stackrel{(3.38)}{=} (b \triangle c) \cdot ((b \cdot (c \trianglelefteq b)) + (b \cdot (b \triangleleft c))) \\
&\stackrel{(3.7),(3.44)}{=} (b \triangle c) \cdot ((c \trianglelefteq b) + (b \triangleleft c)) \\
&\stackrel{(3.2),(3.7),(3.35)}{=} (c \trianglelefteq b) \cdot (c \triangle b) + (b \triangleleft c) \cdot (b \triangle c) \\
&\stackrel{(3.69),(3.73)}{=} (c \triangle b) + \perp \\
&\stackrel{(3.8),(3.35)}{=} b \triangle c
\end{aligned}$$

$$\begin{aligned}
(a \triangle (b \triangle c)) \cdot ((b \triangle c) \trianglelefteq (b \triangleleft c)) &= ((a \triangle b) \cdot (b \triangle c)) \cdot (b \triangle c) \\
&\stackrel{(3.4)}{=} (a \triangle b) \cdot ((b \triangle c) \cdot (b \triangle c)) \\
&\stackrel{(3.6)}{=} (a \triangle b) \cdot (b \triangle c)
\end{aligned}$$

$$\begin{aligned}
a \triangle (b \trianglelefteq c) &= (a \triangle b) \cdot (b \triangleleft c) + (a \triangle b) \cdot (b \triangle c) \\
&\stackrel{(3.7)}{=} (a \triangle b) \cdot ((b \triangleleft c) + (b \triangle c)) \\
&\stackrel{(3.16)}{=} (a \triangle b) \cdot (b \trianglelefteq c)
\end{aligned}$$

End of the proof of Theorem (3.42).

Proof of Theorem (3.43) : $a + (a \triangle b) = a$

1. Case 1: the two dates are equal: $d(a) = d(b)$, meaning that $a = b$
By definition:

$$\begin{aligned}
a \triangle b &= a \\
a + (a \triangle b) &= a
\end{aligned}$$

2. Case 2: the two dates are different

(a) Case 2a: $d(a) < d(b)$

By definition:

$$\begin{aligned} a \triangle b &= \perp \\ a + (a \triangle b) &= a \end{aligned}$$

(b) Case 2b: $d(a) > d(b)$

By definition:

$$\begin{aligned} a \triangle b &= \perp \\ a + (a \triangle b) &= a \end{aligned}$$

For all the cases of the domain of our study, $a + (a \triangle b) = a$. Thus:

$$a + (a \triangle b) = a$$

End of the proof of Theorem (3.43).

Proof of Theorem (3.44) : $a \cdot (a \triangle b) = a \triangle b$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.2) : a \cdot b = b \cdot a$$

$$(3.14) : a \cdot (a + b) = a$$

$$(3.43) : a + (a \triangle b) = a$$

$$a \cdot (a \triangle b) \stackrel{(3.43)}{=} (a + (a \triangle b)) \cdot (a \triangle b) \stackrel{(3.1),(3.2)}{=} (a \triangle b) \cdot ((a \triangle b) + a) \stackrel{(3.14)}{=} a \triangle b$$

End of the proof of Theorem (3.44).

Proof of Theorem (3.45) : $(a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = (a \triangle b) \cdot (b \triangle c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{aligned} \left\{ \begin{array}{l} a \triangle b = a \\ b \triangle c = b \\ a \triangle c = a \end{array} \right. & \qquad \left\{ \begin{array}{l} a \triangle b = a \\ b \triangle c = b \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = a & \qquad (a \triangle b) \cdot (b \triangle c) = a \end{aligned}$$

2. Case 2: two dates out of three are equal

(a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{ccc} \left\{ \begin{array}{l} a \triangle b = a \\ b \triangle c = \perp \\ a \triangle c = \perp \end{array} \right. & & \left\{ \begin{array}{l} a \triangle b = a \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = \perp & & (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

(b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{ccc} \left\{ \begin{array}{l} a \triangle b = a \\ b \triangle c = \perp \\ a \triangle c = \perp \end{array} \right. & & \left\{ \begin{array}{l} a \triangle b = a \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = \perp & & (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

(c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{ccc} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \\ a \triangle c = a \end{array} \right. & & \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = \perp & & (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{ccc} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \\ a \triangle c = a \end{array} \right. & & \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = \perp & & (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{ccc} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = b \\ a \triangle c = \perp \end{array} \right. & & \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = b \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = \perp & & (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{ccc} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = b \\ a \triangle c = \perp \end{array} \right. & & \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = b \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = \perp & & (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \\ a \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = \perp \end{array} \qquad \begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \\ a \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = \perp \end{array} \qquad \begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \\ a \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = \perp \end{array} \qquad \begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \\ a \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = \perp \end{array} \qquad \begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \\ a \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = \perp \end{array} \qquad \begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \\ a \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = \perp \end{array} \qquad \begin{array}{l} \left\{ \begin{array}{l} a \triangle b = \perp \\ b \triangle c = \perp \end{array} \right. \\ (a \triangle b) \cdot (b \triangle c) = \perp \end{array}$$

For all the cases of the domain of our study,

$$(a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = (a \triangle b) \cdot (b \triangle c).$$

Thus:

$$(a \triangle b) \cdot (b \triangle c) \cdot (a \triangle c) = (a \triangle b) \cdot (b \triangle c)$$

End of the proof of Theorem (3.45).

A.2.4 Proofs of the theorems satisfied by operator Inclusive BEFORE

Proof of Theorem (3.46) : $(a \triangleleft b) \cdot (b \triangleleft a) = a \triangle b$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.2) : a \cdot b = b \cdot a$$

$$(3.6) : a \cdot a = a$$

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.8) : a + \perp = a$$

$$(3.13) : a + (a \cdot b) = a$$

$$(3.16) : a \triangleleft b = a \triangleleft b + a \triangle b$$

$$(3.17) : (a \triangleleft b) \cdot (b \triangleleft a) = \perp$$

$$(3.35) : a \triangle b = b \triangle a$$

$$\begin{aligned}
(a \triangleleft b) \cdot (b \triangleleft a) &\stackrel{(3.16)}{=} ((a \triangleleft b) + (a \triangle b)) \cdot ((b \triangleleft a) + (b \triangle a)) \\
&\stackrel{(3.35)}{=} ((a \triangleleft b) + (a \triangle b)) \cdot ((b \triangleleft a) + (a \triangle b)) \\
&\stackrel{(3.7)}{=} (a \triangleleft b) \cdot (b \triangleleft a) + (a \triangleleft b) \cdot (a \triangle b) + (a \triangle b) \cdot (b \triangleleft a) \\
&\quad + (a \triangle b) \cdot (a \triangle b) \\
&\stackrel{(3.6)}{=} (a \triangleleft b) \cdot (b \triangleleft a) + (a \triangleleft b) \cdot (a \triangle b) + (a \triangle b) \cdot (b \triangleleft a) \\
&\quad + (a \triangle b) \\
&\stackrel{(3.1),(3.2)}{=} (a \triangleleft b) \cdot (b \triangleleft a) + (a \triangle b) + (a \triangle b) \cdot (a \triangleleft b) \\
&\quad + (a \triangle b) \cdot (b \triangleleft a) \\
&\stackrel{(3.13)}{=} (a \triangleleft b) \cdot (b \triangleleft a) + (a \triangle b) \\
&\stackrel{(3.17)}{=} \perp + (a \triangle b) \stackrel{(3.1)}{=} (a \triangle b) + \perp \stackrel{(3.8)}{=} a \triangle b
\end{aligned}$$

End of the proof of Theorem (3.46).

Proof of Theorem (3.47) :

$$a \sqsubseteq (b \sqsubseteq c) = (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b)) + (a \triangle b) \cdot (b \sqsubseteq c)$$

For this proof, we will use the following theorems:

$$(3.16) : a \sqsubseteq b = a \triangleleft b + a \triangle b$$

$$(3.26) : a \triangleleft (b \sqsubseteq c) = (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b))$$

$$(3.42) : a \triangle (b \sqsubseteq c) = (a \triangle b) \cdot (b \sqsubseteq c)$$

$$\begin{aligned} a \sqsubseteq (b \sqsubseteq c) &\stackrel{(3.16)}{=} a \triangleleft (b \sqsubseteq c) + a \triangle (b \sqsubseteq c) \\ &\stackrel{(3.26), (3.42)}{=} (a \triangleleft b) + (a \cdot b \cdot (c \triangleleft b)) + (a \triangle b) \cdot (b \sqsubseteq c) \end{aligned}$$

End of the proof of Theorem (3.47).

Proof of Theorem (3.48) : $(a \sqsubseteq b) \sqsubseteq c = (a \sqsubseteq b) \cdot (a \triangleleft c)$

For this proof, we will use the following theorems:

$$(3.2) : a \cdot b = b \cdot a$$

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.16) : a \sqsubseteq b = a \triangleleft b + a \triangle b$$

$$(3.30) : (a \sqsubseteq b) \triangleleft c = (a \sqsubseteq b) \cdot (a \triangleleft c)$$

$$(3.35) : a \triangle b = b \triangle a$$

$$(3.42) : a \triangle (b \sqsubseteq c) = (a \triangle b) \cdot (b \sqsubseteq c)$$

$$\begin{aligned} (a \sqsubseteq b) \sqsubseteq c &\stackrel{(3.16)}{=} ((a \sqsubseteq b) \triangleleft c) + ((a \sqsubseteq b) \triangle c) \\ &\stackrel{(3.35)}{=} ((a \sqsubseteq b) \triangleleft c) + (c \triangle (a \sqsubseteq b)) \\ &\stackrel{(3.30), (3.42)}{=} (a \sqsubseteq b) \cdot (a \triangleleft c) + (c \triangle a) \cdot (a \sqsubseteq b) \\ &\stackrel{(3.2), (3.35)}{=} (a \sqsubseteq b) \cdot (a \triangleleft c) + (a \sqsubseteq b) \cdot (a \triangle c) \\ &\stackrel{(3.7)}{=} (a \sqsubseteq b) \cdot ((a \triangleleft c) + (a \triangle c)) \\ &\stackrel{(3.16)}{=} (a \sqsubseteq b) \cdot (a \triangleleft c) \end{aligned}$$

End of the proof of Theorem (3.48).

Proof of Theorem (3.49) : $\perp \sqsubseteq a = \perp$

For this proof, we will use the following theorems:

$$(3.5) : a + a = a$$

$$(3.16) : a \sqsubseteq b = a \triangleleft b + a \triangle b$$

$$(3.20) : \perp \triangleleft a = \perp$$

$$(3.35) : a \triangle b = b \triangle a$$

$$(3.37) : a \triangle \perp = \perp$$

$$\perp \trianglelefteq a \stackrel{(3.16)}{=} \perp \triangleleft a + \perp \triangleleft a \stackrel{(3.35)}{=} \perp \triangleleft a + a \triangleleft \perp \stackrel{(3.20),(3.37)}{=} \perp + \perp \stackrel{(3.5)}{=} \perp$$

End of the proof of Theorem (3.49).

Proof of Theorem (3.50) : $a \triangleleft \perp = a$

For this proof, we will use the following theorems:

$$(3.8) : a + \perp = a$$

$$(3.16) : a \trianglelefteq b = a \triangleleft b + a \triangleleft b$$

$$(3.21) : a \triangleleft \perp = a$$

$$(3.37) : a \triangleleft \perp = \perp$$

$$a \trianglelefteq \perp \stackrel{(3.16)}{=} a \triangleleft \perp + a \triangleleft \perp \stackrel{(3.21),(3.37)}{=} a + \perp \stackrel{(3.8)}{=} a$$

End of the proof of Theorem (3.50).

Proof of Theorem (3.51) : $a \trianglelefteq a = a$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.8) : a + \perp = a$$

$$(3.16) : a \trianglelefteq b = a \triangleleft b + a \triangleleft b$$

$$(3.22) : a \triangleleft a = \perp$$

$$(3.38) : a \triangleleft a = a$$

$$a \trianglelefteq a \stackrel{(3.16)}{=} a \triangleleft a + a \triangleleft a \stackrel{(3.22),(3.38)}{=} \perp + a \stackrel{(3.1)}{=} a + \perp \stackrel{(3.8)}{=} a$$

End of the proof of Theorem (3.51).

Proof of Theorem (3.52) : $a \trianglelefteq (b + c) = (a \trianglelefteq b) \cdot (a \trianglelefteq c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$
By definition:

$$\begin{array}{l} b + c = b \\ a \trianglelefteq (b + c) = a \end{array} \quad \left\{ \begin{array}{l} a \trianglelefteq b = a \\ a \trianglelefteq c = a \end{array} \right. \\ (a \trianglelefteq b) \cdot (a \trianglelefteq c) = a$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{l} b + c = b \\ a \trianglelefteq (b + c) = a \end{array} \quad \left\{ \begin{array}{l} a \trianglelefteq b = a \\ a \trianglelefteq c = a \end{array} \right. \\ (a \trianglelefteq b) \cdot (a \trianglelefteq c) = a$$

(b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{l} b + c = c \\ a \sqsubseteq (b + c) = \perp \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = a \\ a \sqsubseteq c = \perp \end{array} \right. \\ (a \sqsubseteq b) \cdot (a \sqsubseteq c) = \perp$$

(c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{l} b + c = c \\ a \sqsubseteq (b + c) = a \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = a \\ a \sqsubseteq c = a \end{array} \right. \\ (a \sqsubseteq b) \cdot (a \sqsubseteq c) = a$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{l} b + c = b \\ a \sqsubseteq (b + c) = \perp \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = \perp \\ a \sqsubseteq c = a \end{array} \right. \\ (a \sqsubseteq b) \cdot (a \sqsubseteq c) = \perp$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{l} b + c = b \\ a \sqsubseteq (b + c) = \perp \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = \perp \\ a \sqsubseteq c = \perp \end{array} \right. \\ (a \sqsubseteq b) \cdot (a \sqsubseteq c) = \perp$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{l} b + c = b \\ a \sqsubseteq (b + c) = a \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = a \\ a \sqsubseteq c = a \end{array} \right. \\ (a \sqsubseteq b) \cdot (a \sqsubseteq c) = a$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{l} b + c = b \\ a \sqsubseteq (b + c) = a \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = a \\ a \sqsubseteq c = a \end{array} \right. \\ (a \sqsubseteq b) \cdot (a \sqsubseteq c) = a$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{l} b + c = c \\ a \sqsubseteq (b + c) = a \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = a \\ a \sqsubseteq c = a \end{array} \right. \\ (a \sqsubseteq b) \cdot (a \sqsubseteq c) = a$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{l} b + c = b \\ a \sqsubseteq (b + c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \sqsubseteq b = \perp \\ a \sqsubseteq c = a \end{array} \right. \\ (a \sqsubseteq b) \cdot (a \sqsubseteq c) = \perp \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{l} b + c = b \\ a \sqsubseteq (b + c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \sqsubseteq b = \perp \\ a \sqsubseteq c = \perp \end{array} \right. \\ (a \sqsubseteq b) \cdot (a \sqsubseteq c) = \perp \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{l} b + c = c \\ a \sqsubseteq (b + c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \sqsubseteq b = a \\ a \sqsubseteq c = \perp \end{array} \right. \\ (a \sqsubseteq b) \cdot (a \sqsubseteq c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{l} b + c = c \\ a \sqsubseteq (b + c) = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \sqsubseteq b = \perp \\ a \sqsubseteq c = \perp \end{array} \right. \\ (a \sqsubseteq b) \cdot (a \sqsubseteq c) = \perp \end{array}$$

For all the cases of the domain of our study, $a \sqsubseteq (b + c) = (a \sqsubseteq b) \cdot (a \sqsubseteq c)$. Thus:

$$a \sqsubseteq (b + c) = (a \sqsubseteq b) \cdot (a \sqsubseteq c)$$

End of the proof of Theorem (3.52).

Proof of Theorem (3.53) : $a \sqsubseteq (b \cdot c) = (a \sqsubseteq b) + (a \sqsubseteq c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{array}{l} b \cdot c = b \\ a \sqsubseteq (b \cdot c) = a \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \sqsubseteq b = a \\ a \sqsubseteq c = a \end{array} \right. \\ (a \sqsubseteq b) + (a \sqsubseteq c) = a \end{array}$$

2. Case 2: two dates out of three are equal

(a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{l} b \cdot c = c \\ a \sqsubseteq (b \cdot c) = a \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = a \\ a \sqsubseteq c = a \end{array} \right. \\ (a \sqsubseteq b) + (a \sqsubseteq c) = a$$

(b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{l} b \cdot c = b \\ a \sqsubseteq (b \cdot c) = a \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = a \\ a \sqsubseteq c = \perp \end{array} \right. \\ (a \sqsubseteq b) + (a \sqsubseteq c) = a$$

(c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{l} b \cdot c = b \\ a \sqsubseteq (b \cdot c) = a \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = a \\ a \sqsubseteq c = a \end{array} \right. \\ (a \sqsubseteq b) + (a \sqsubseteq c) = a$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{l} b \cdot c = c \\ a \sqsubseteq (b \cdot c) = a \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = \perp \\ a \sqsubseteq c = a \end{array} \right. \\ (a \sqsubseteq b) + (a \sqsubseteq c) = a$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{l} b \cdot c = b \\ a \sqsubseteq (b \cdot c) = \perp \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = \perp \\ a \sqsubseteq c = \perp \end{array} \right. \\ (a \sqsubseteq b) + (a \sqsubseteq c) = \perp$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{l} b \cdot c = b \\ a \sqsubseteq (b \cdot c) = a \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = a \\ a \sqsubseteq c = a \end{array} \right. \\ (a \sqsubseteq b) + (a \sqsubseteq c) = a$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{l} b \cdot c = c \\ a \sqsubseteq (b \cdot c) = a \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = a \\ a \sqsubseteq c = a \end{array} \right. \\ (a \sqsubseteq b) + (a \sqsubseteq c) = a$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{l} b \cdot c = b \\ a \sqsubseteq (b \cdot c) = a \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = a \\ a \sqsubseteq c = a \end{array} \right. \\ (a \sqsubseteq b) + (a \sqsubseteq c) = a$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{l} b \cdot c = c \\ a \sqsubseteq (b \cdot c) = a \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = \perp \\ a \sqsubseteq c = a \end{array} \right. \\ (a \sqsubseteq b) + (a \sqsubseteq c) = a$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{l} b \cdot c = c \\ a \sqsubseteq (b \cdot c) = \perp \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = \perp \\ a \sqsubseteq c = \perp \end{array} \right. \\ (a \sqsubseteq b) + (a \sqsubseteq c) = \perp$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{l} b \cdot c = b \\ a \sqsubseteq (b \cdot c) = a \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = a \\ a \sqsubseteq c = \perp \end{array} \right. \\ (a \sqsubseteq b) + (a \sqsubseteq c) = a$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{l} b \cdot c = b \\ a \sqsubseteq (b \cdot c) = \perp \end{array} \quad \left\{ \begin{array}{l} a \sqsubseteq b = \perp \\ a \sqsubseteq c = \perp \end{array} \right. \\ (a \sqsubseteq b) + (a \sqsubseteq c) = \perp$$

For all the cases of the domain of our study, $a \sqsubseteq (b \cdot c) = (a \sqsubseteq b) + (a \sqsubseteq c)$. Thus:

$$a \sqsubseteq (b \cdot c) = (a \sqsubseteq b) + (a \sqsubseteq c)$$

End of the proof of Theorem (3.53).

Proof of Theorem (3.54) :

$$a \sqsubseteq (b \triangleleft c) = (a \triangleleft b) + (a \cdot b \cdot (c \sqsubseteq b)) + (a \triangle b) \cdot (b \triangleleft c)$$

For this proof, we will use the following theorems:

$$(3.16) : a \sqsubseteq b = a \triangleleft b + a \triangle b$$

$$(3.18) : a \triangleleft (b \triangleleft c) = (a \triangleleft b) + (a \cdot b \cdot (c \sqsubseteq b))$$

$$(3.41) : a \triangle (b \triangleleft c) = (a \triangle b) \cdot (b \triangleleft c)$$

$$\begin{aligned}
a \trianglelefteq (b \triangleleft c) &\stackrel{(3.16)}{=} (a \triangleleft (b \triangleleft c)) + (a \triangle (b \triangleleft c)) \\
&\stackrel{(3.18),(3.41)}{=} (a \triangleleft b) + (a \cdot b \cdot (c \trianglelefteq b)) + (a \triangle b) \cdot (b \triangleleft c)
\end{aligned}$$

End of the proof of Theorem (3.54).

Proof of Theorem (3.55) :

$$a \trianglelefteq (b \triangle c) = (a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) + (a \triangle b) \cdot (b \triangle c)$$

For this proof, we will use the following theorems:

$$(3.16) : a \trianglelefteq b = a \triangleleft b + a \triangle b$$

$$(3.25) : a \triangleleft (b \triangle c) = (a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c)$$

$$(3.36) : a \triangle (b \triangle c) = (a \triangle b) \cdot (b \triangle c)$$

$$\begin{aligned}
a \trianglelefteq (b \triangle c) &\stackrel{(3.16)}{=} (a \triangleleft (b \triangle c)) + (a \triangle (b \triangle c)) \\
&\stackrel{(3.25),(3.36)}{=} (a \cdot (b \triangleleft c)) + (a \cdot (c \triangleleft b)) + (a \triangleleft b) + (a \triangleleft c) \\
&\quad + (a \triangle b) \cdot (b \triangle c)
\end{aligned}$$

End of the proof of Theorem (3.55).

Proof of Theorem (3.56) : $(a + b) \trianglelefteq c = (a \trianglelefteq c) + (b \trianglelefteq c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$
By definition:

$$\begin{aligned}
a + b &= a & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{cases} \\
(a + b) \trianglelefteq c &= a & (a \trianglelefteq c) + (b \trianglelefteq c) = a
\end{aligned}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{aligned}
a + b &= a & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{cases} \\
(a + b) \trianglelefteq c &= a & (a \trianglelefteq c) + (b \trianglelefteq c) = a
\end{aligned}$$

- (b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{aligned}
a + b &= a & \begin{cases} a \trianglelefteq c = \perp \\ b \trianglelefteq c = \perp \end{cases} \\
(a + b) \trianglelefteq c &= \perp & (a \trianglelefteq c) + (b \trianglelefteq c) = \perp
\end{aligned}$$

(c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{l} a + b = a \\ (a + b) \trianglelefteq c = a \end{array} \quad \left\{ \begin{array}{l} a \trianglelefteq c = a \\ b \trianglelefteq c = \perp \end{array} \right. \\ (a \trianglelefteq c) + (b \trianglelefteq c) = a$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{l} a + b = b \\ (a + b) \trianglelefteq c = b \end{array} \quad \left\{ \begin{array}{l} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{array} \right. \\ (a \trianglelefteq c) + (b \trianglelefteq c) = b$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{l} a + b = b \\ (a + b) \trianglelefteq c = b \end{array} \quad \left\{ \begin{array}{l} a \trianglelefteq c = \perp \\ b \trianglelefteq c = b \end{array} \right. \\ (a \trianglelefteq c) + (b \trianglelefteq c) = b$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{l} a + b = a \\ (a + b) \trianglelefteq c = a \end{array} \quad \left\{ \begin{array}{l} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{array} \right. \\ (a \trianglelefteq c) + (b \trianglelefteq c) = a$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{l} a + b = a \\ (a + b) \trianglelefteq c = a \end{array} \quad \left\{ \begin{array}{l} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{array} \right. \\ (a \trianglelefteq c) + (b \trianglelefteq c) = a$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{l} a + b = a \\ (a + b) \trianglelefteq c = a \end{array} \quad \left\{ \begin{array}{l} a \trianglelefteq c = a \\ b \trianglelefteq c = \perp \end{array} \right. \\ (a \trianglelefteq c) + (b \trianglelefteq c) = a$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{l} a + b = b \\ (a + b) \trianglelefteq c = b \end{array} \quad \left\{ \begin{array}{l} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{array} \right. \\ (a \trianglelefteq c) + (b \trianglelefteq c) = b$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{ll} a + b = b & \begin{cases} a \trianglelefteq c = \perp \\ b \trianglelefteq c = b \end{cases} \\ (a + b) \trianglelefteq c = b & (a \trianglelefteq c) + (b \trianglelefteq c) = b \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{ll} a + b = a & \begin{cases} a \trianglelefteq c = \perp \\ b \trianglelefteq c = \perp \end{cases} \\ (a + b) \trianglelefteq c = \perp & (a \trianglelefteq c) + (b \trianglelefteq c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{ll} a + b = b & \begin{cases} a \trianglelefteq c = \perp \\ b \trianglelefteq c = \perp \end{cases} \\ (a + b) \trianglelefteq c = \perp & (a \trianglelefteq c) + (b \trianglelefteq c) = \perp \end{array}$$

For all the cases of the domain of our study, $(a + b) \trianglelefteq c = (a \trianglelefteq c) + (b \trianglelefteq c)$. Thus:

$$(a + b) \trianglelefteq c = (a \trianglelefteq c) + (b \trianglelefteq c)$$

End of the proof of Theorem (3.56).

Proof of Theorem (3.57) : $(a \cdot b) \trianglelefteq c = (a \trianglelefteq c) \cdot (b \trianglelefteq c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{cases} \\ (a \cdot b) \trianglelefteq c = a & (a \trianglelefteq c) \cdot (b \trianglelefteq c) = a \end{array}$$

2. Case 2: two dates out of three are equal

(a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{cases} \\ (a \cdot b) \trianglelefteq c = a & (a \trianglelefteq c) \cdot (b \trianglelefteq c) = a \end{array}$$

(b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \trianglelefteq c = \perp \\ b \trianglelefteq c = \perp \end{cases} \\ (a \cdot b) \trianglelefteq c = \perp & (a \trianglelefteq c) \cdot (b \trianglelefteq c) = \perp \end{array}$$

(c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{ll} a \cdot b = b & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = \perp \end{cases} \\ (a \cdot b) \trianglelefteq c = \perp & (a \trianglelefteq c) \cdot (b \trianglelefteq c) = \perp \end{array}$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{cases} \\ (a \cdot b) \trianglelefteq c = a & (a \trianglelefteq c) \cdot (b \trianglelefteq c) = a \end{array}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \trianglelefteq c = \perp \\ b \trianglelefteq c = b \end{cases} \\ (a \cdot b) \trianglelefteq c = \perp & (a \trianglelefteq c) \cdot (b \trianglelefteq c) = \perp \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{ll} a \cdot b = b & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{cases} \\ (a \cdot b) \trianglelefteq c = b & (a \trianglelefteq c) \cdot (b \trianglelefteq c) = b \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{ll} a \cdot b = b & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{cases} \\ (a \cdot b) \trianglelefteq c = b & (a \trianglelefteq c) \cdot (b \trianglelefteq c) = b \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{ll} a \cdot b = b & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = \perp \end{cases} \\ (a \cdot b) \trianglelefteq c = \perp & (a \trianglelefteq c) \cdot (b \trianglelefteq c) = \perp \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{cases} \\ (a \cdot b) \trianglelefteq c = a & (a \trianglelefteq c) \cdot (b \trianglelefteq c) = a \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \trianglelefteq c = \perp \\ b \trianglelefteq c = b \end{cases} \\ (a \cdot b) \trianglelefteq c = \perp & (a \trianglelefteq c) \cdot (b \trianglelefteq c) = \perp \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{ll} a \cdot b = b & \begin{cases} a \trianglelefteq c = \perp \\ b \trianglelefteq c = \perp \end{cases} \\ (a \cdot b) \trianglelefteq c = \perp & (a \trianglelefteq c) \cdot (b \trianglelefteq c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{ll} a \cdot b = a & \begin{cases} a \trianglelefteq c = \perp \\ b \trianglelefteq c = \perp \end{cases} \\ (a \cdot b) \trianglelefteq c = \perp & (a \trianglelefteq c) \cdot (b \trianglelefteq c) = \perp \end{array}$$

For all the cases of the domain of our study, $(a \cdot b) \trianglelefteq c = (a \trianglelefteq c) \cdot (b \trianglelefteq c)$. Thus:

$$(a \cdot b) \trianglelefteq c = (a \trianglelefteq c) \cdot (b \trianglelefteq c)$$

*End of the proof of Theorem (3.57).***Proof of Theorem (3.58) :**

$$(a \triangle b) \trianglelefteq c = (a \triangle b) \cdot (a \trianglelefteq c) = (a \triangle b) \cdot (b \trianglelefteq c) = (a \trianglelefteq c) \triangle (b \trianglelefteq c)$$

This proof will be done in 3 steps:

- Step 1: $(a \triangle b) \trianglelefteq c = (a \triangle b) \cdot (a \trianglelefteq c)$

For this proof, we will use the following theorems:

$$(3.2) : a \cdot b = b \cdot a$$

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.16) : a \trianglelefteq b = a \triangleleft b + a \triangle b$$

$$(3.29) : (a \triangle b) \triangleleft c = (a \triangle b) \cdot (a \triangleleft c)$$

$$(3.35) : a \triangle b = b \triangle a$$

$$(3.36) : a \triangle (b \triangle c) = (a \triangle b) \cdot (b \triangle c)$$

$$\begin{aligned}
(a \triangle b) \trianglelefteq c &\stackrel{(3.16)}{=} ((a \triangle b) \triangleleft c) + ((a \triangle b) \triangle c) \\
&\stackrel{(3.35)}{=} ((a \triangle b) \triangleleft c) + (c \triangle (a \triangle b)) \\
&\stackrel{(3.29),(3.36)}{=} (a \triangle b) \cdot (a \triangleleft c) + (c \triangle a) \cdot (a \triangle b) \\
&\stackrel{(3.35)}{=} (a \triangle b) \cdot (a \triangleleft c) + (a \triangle c) \cdot (a \triangle b) \\
&\stackrel{(3.2)}{=} (a \triangle b) \cdot (a \triangleleft c) + (a \triangle b) \cdot (a \triangle c) \\
&\stackrel{(3.7)}{=} (a \triangle b) \cdot ((a \triangleleft c) + (a \triangle c)) \\
&\stackrel{(3.16)}{=} (a \triangle b) \cdot (a \trianglelefteq c)
\end{aligned}$$

- Step 2: $(a \triangle b) \trianglelefteq c = (a \triangle b) \cdot (b \trianglelefteq c)$

For this proof, we will use the following theorems:

$$(3.35) : a \triangle b = b \triangle a$$

$$(3.58) \text{ part 1 : } (a \triangle b) \trianglelefteq c = (a \triangle b) \cdot (a \trianglelefteq c)$$

$$(a \triangle b) \trianglelefteq c \stackrel{(3.35)}{=} (b \triangle a) \trianglelefteq c \stackrel{(3.58)}{=} (b \triangle a) \cdot (b \trianglelefteq c) \stackrel{(3.35)}{=} (a \triangle b) \cdot (b \trianglelefteq c)$$

- Step 3: $(a \triangle b) \trianglelefteq c = (a \trianglelefteq c) \triangle (b \trianglelefteq c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{aligned}
a \triangle b = a & \quad \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{cases} \\
(a \triangle b) \trianglelefteq c = a & \quad (a \trianglelefteq c) \triangle (b \trianglelefteq c) = a
\end{aligned}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\begin{aligned}
a \triangle b = a & \quad \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{cases} \\
(a \triangle b) \trianglelefteq c = a & \quad (a \trianglelefteq c) \triangle (b \trianglelefteq c) = a
\end{aligned}$$

- (b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{aligned}
a \triangle b = a & \quad \begin{cases} a \trianglelefteq c = \perp \\ b \trianglelefteq c = \perp \end{cases} \\
(a \triangle b) \trianglelefteq c = \perp & \quad (a \trianglelefteq c) \triangle (b \trianglelefteq c) = \perp
\end{aligned}$$

(c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{aligned} a \triangle b &= \perp & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = \perp \end{cases} \\ (a \triangle b) \trianglelefteq c &= \perp & (a \trianglelefteq c) \triangle (b \trianglelefteq c) = \perp \end{aligned}$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{aligned} a \triangle b &= \perp & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{cases} \\ (a \triangle b) \trianglelefteq c &= \perp & (a \trianglelefteq c) \triangle (b \trianglelefteq c) = \perp \end{aligned}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{aligned} a \triangle b &= \perp & \begin{cases} a \trianglelefteq c = \perp \\ b \trianglelefteq c = b \end{cases} \\ (a \triangle b) \trianglelefteq c &= \perp & (a \trianglelefteq c) \triangle (b \trianglelefteq c) = \perp \end{aligned}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{aligned} a \triangle b &= \perp & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{cases} \\ (a \triangle b) \trianglelefteq c &= \perp & (a \trianglelefteq c) \triangle (b \trianglelefteq c) = \perp \end{aligned}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{aligned} a \triangle b &= \perp & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{cases} \\ (a \triangle b) \trianglelefteq c &= \perp & (a \trianglelefteq c) \triangle (b \trianglelefteq c) = \perp \end{aligned}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{aligned} a \triangle b &= \perp & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = \perp \end{cases} \\ (a \triangle b) \trianglelefteq c &= \perp & (a \trianglelefteq c) \triangle (b \trianglelefteq c) = \perp \end{aligned}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{aligned} a \triangle b &= \perp & \begin{cases} a \trianglelefteq c = a \\ b \trianglelefteq c = b \end{cases} \\ (a \triangle b) \trianglelefteq c &= \perp & (a \trianglelefteq c) \triangle (b \trianglelefteq c) = \perp \end{aligned}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{l} a \triangle b = \perp \\ (a \triangle b) \triangleleft c = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft c = \perp \\ b \triangleleft c = b \end{array} \right. \\ (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{l} a \triangle b = \perp \\ (a \triangle b) \triangleleft c = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{l} a \triangle b = \perp \\ (a \triangle b) \triangleleft c = \perp \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{array} \right. \\ (a \triangleleft c) \triangle (b \triangleleft c) = \perp \end{array}$$

For all the cases of the domain of our study, $(a \triangle b) \triangleleft c = (a \triangleleft c) \triangle (b \triangleleft c)$.

Thus:

$$(a \triangle b) \triangleleft c = (a \triangleleft c) \triangle (b \triangleleft c)$$

End of the proof of Theorem (3.58).

Proof of Theorem (3.59) : $(a \triangleleft b) \triangleleft c = (a \triangleleft b) \cdot (a \triangleleft c)$

For this proof, we will use the following theorems:

$$(3.2) : a \cdot b = b \cdot a$$

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.16) : a \triangleleft b = a \triangleleft b + a \triangle b$$

$$(3.19) : (a \triangleleft b) \triangleleft c = (a \triangleleft b) \cdot (a \triangleleft c)$$

$$(3.35) : a \triangle b = b \triangle a$$

$$(3.41) : a \triangle (b \triangleleft c) = (a \triangle b) \cdot (b \triangleleft c)$$

$$\begin{aligned} (a \triangleleft b) \triangleleft c &\stackrel{(3.16)}{=} ((a \triangleleft b) \triangleleft c) + ((a \triangleleft b) \triangle c) \\ &\stackrel{(3.35)}{=} ((a \triangleleft b) \triangleleft c) + (c \triangle (a \triangleleft b)) \\ &\stackrel{(3.19),(3.41)}{=} (a \triangleleft b) \cdot (a \triangleleft c) + (c \triangle a) \cdot (a \triangleleft b) \\ &\stackrel{(3.35)}{=} (a \triangleleft b) \cdot (a \triangleleft c) + (a \triangle c) \cdot (a \triangleleft b) \\ &\stackrel{(3.2)}{=} (a \triangleleft b) \cdot (a \triangleleft c) + (a \triangleleft b) \cdot (a \triangle c) \\ &\stackrel{(3.7)}{=} (a \triangleleft b) \cdot ((a \triangleleft c) + (a \triangle c)) \\ &\stackrel{(3.16)}{=} (a \triangleleft b) \cdot (a \triangleleft c) \end{aligned}$$

End of the proof of Theorem (3.59).

Proof of Theorem (3.60) : $a + (a \leq b) = a$

For this proof, we will use the following theorems:

$$(3.3) : a + (b + c) = (a + b) + c$$

$$(3.16) : a \leq b = a \triangleleft b + a \triangle b$$

$$(3.31) : a + (a \triangleleft b) = a$$

$$(3.43) : a + (a \triangle b) = a$$

$$a + (a \leq b) \stackrel{(3.16)}{=} a + ((a \triangleleft b) + (a \triangle b)) \stackrel{(3.3)}{=} (a + (a \triangleleft b)) + (a \triangle b) \stackrel{(3.31)}{=} a + (a \triangle b) \stackrel{(3.43)}{=} a$$

End of the proof of Theorem (3.60).

Proof of Theorem (3.61) : $b + (a \leq b) = a + b$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.3) : a + (b + c) = (a + b) + c$$

$$(3.16) : a \leq b = a \triangleleft b + a \triangle b$$

$$(3.32) : (a \triangleleft b) + b = a + b$$

$$(3.43) : a + (a \triangle b) = a$$

$$\begin{aligned} b + (a \leq b) &\stackrel{(3.16)}{=} b + ((a \triangleleft b) + (a \triangle b)) \stackrel{(3.3)}{=} (b + (a \triangleleft b)) + (a \triangle b) \\ &\stackrel{(3.1)}{=} ((a \triangleleft b) + b) + (a \triangle b) \stackrel{(3.32)}{=} (a + b) + (a \triangle b) \\ &\stackrel{(3.1),(3.3)}{=} (a + (a \triangle b)) + b \stackrel{(3.43)}{=} a + b \end{aligned}$$

End of the proof of Theorem (3.61).

Proof of Theorem (3.62) : $a \cdot (a \leq b) = a \leq b$

For this proof, we will use the following theorems:

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.16) : a \leq b = a \triangleleft b + a \triangle b$$

$$(3.33) : a \cdot (a \triangleleft b) = a \triangleleft b$$

$$(3.44) : a \cdot (a \triangle b) = a \triangle b$$

$$\begin{aligned} a \cdot (a \leq b) &\stackrel{(3.16)}{=} a \cdot ((a \triangleleft b) + (a \triangle b)) \stackrel{(3.7)}{=} (a \cdot (a \triangleleft b)) + (a \cdot (a \triangle b)) \\ &\stackrel{(3.33),(3.44)}{=} (a \triangleleft b) + (a \triangle b) \stackrel{(3.16)}{=} a \leq b \end{aligned}$$

End of the proof of Theorem (3.62).

Proof of Theorem (3.63) : $(a \trianglelefteq b) + (b \trianglelefteq a) = a + b$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.3) : a + (b + c) = (a + b) + c$$

$$(3.5) : a + a = a$$

$$(3.16) : a \trianglelefteq b = a \triangleleft b + a \triangle b$$

$$(3.35) : a \triangle b = b \triangle a$$

$$(3.74) : (a \triangleleft b) + (a \triangle b) + (b \triangleleft a) = a + b$$

$$\begin{aligned} (a \trianglelefteq b) + (b \trianglelefteq a) &\stackrel{(3.16)}{=} ((a \triangleleft b) + (a \triangle b)) + ((b \triangleleft a) + (b \triangle a)) \\ &\stackrel{(3.1),(3.3)}{=} (a \triangleleft b) + ((a \triangle b) + (b \triangle a)) + (b \triangleleft a) \\ &\stackrel{(3.35)}{=} (a \triangleleft b) + ((a \triangle b) + (a \triangle b)) + (b \triangleleft a) \\ &\stackrel{(3.5)}{=} (a \triangleleft b) + (a \triangle b) + (b \triangleleft a) \stackrel{(3.74)}{=} a + b \end{aligned}$$

End of the proof of Theorem (3.63).

Proof of Theorem (3.64) : $(a \cdot (b \trianglelefteq a)) + (b \cdot (a \trianglelefteq b)) = a \cdot b$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.3) : a + (b + c) = (a + b) + c$$

$$(3.5) : a + a = a$$

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.16) : a \trianglelefteq b = a \triangleleft b + a \triangle b$$

$$(3.35) : a \triangle b = b \triangle a$$

$$(3.44) : a \cdot (a \triangle b) = a \triangle b$$

$$(3.75) : (a \cdot (b \triangleleft a)) + (a \triangle b) + (b \cdot (a \triangleleft b)) = a \cdot b$$

$$\begin{aligned} (a \cdot (b \trianglelefteq a)) + (b \cdot (a \trianglelefteq b)) &\stackrel{(3.16)}{=} (a \cdot ((b \triangleleft a) + (b \triangle a))) \\ &\quad + (b \cdot ((a \triangleleft b) + (a \triangle b))) \\ &\stackrel{(3.7)}{=} (a \cdot (b \triangleleft a)) + (a \cdot (b \triangle a)) \\ &\quad + (b \cdot (a \triangleleft b)) + (b \cdot (a \triangle b)) \\ &\stackrel{(3.35)}{=} (a \cdot (b \triangleleft a)) + (a \cdot (a \triangle b)) \\ &\quad + (b \cdot (a \triangleleft b)) + (b \cdot (b \triangle a)) \end{aligned}$$

$$\begin{aligned}
(a \cdot (b \trianglelefteq a)) + (b \cdot (a \trianglelefteq b)) &\stackrel{(3.44)}{=} (a \cdot (b \triangleleft a)) + (a \triangle b) + (b \cdot (a \triangleleft b)) + (b \triangle a) \\
&\stackrel{(3.1),(3.3)}{=} (a \cdot (b \triangleleft a)) + ((a \triangle b) + (b \triangle a)) \\
&\quad + (b \cdot (a \triangleleft b)) \\
&\stackrel{(3.35)}{=} (a \cdot (b \triangleleft a)) + ((a \triangle b) + (a \triangle b)) \\
&\quad + (b \cdot (a \triangleleft b)) \\
&\stackrel{(3.5)}{=} (a \cdot (b \triangleleft a)) + (a \triangle b) + (b \cdot (a \triangleleft b)) \stackrel{(3.75)}{=} a \cdot b
\end{aligned}$$

End of the proof of Theorem (3.64).

Proof of Theorem (3.65) : $(a \trianglelefteq b) + (a \cdot (b \trianglelefteq a)) = a$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.3) : a + (b + c) = (a + b) + c$$

$$(3.5) : a + a = a$$

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.16) : a \trianglelefteq b = a \triangleleft b + a \triangle b$$

$$(3.35) : a \triangle b = b \triangle a$$

$$(3.44) : a \cdot (a \triangle b) = a \triangle b$$

$$(3.76) : (a \triangleleft b) + (a \triangle b) + (a \cdot (b \triangleleft a)) = a$$

$$\begin{aligned}
(a \trianglelefteq b) + (a \cdot (b \trianglelefteq a)) &\stackrel{(3.16)}{=} ((a \triangleleft b) + (a \triangle b)) + (a \cdot ((b \triangleleft a) + (b \triangle a))) \\
&\stackrel{(3.7)}{=} ((a \triangleleft b) + (a \triangle b)) + (a \cdot (b \triangleleft a)) + (a \cdot (b \triangle a)) \\
&\stackrel{(3.35)}{=} ((a \triangleleft b) + (a \triangle b)) + (a \cdot (b \triangleleft a)) + (a \cdot (a \triangle b)) \\
&\stackrel{(3.44)}{=} ((a \triangleleft b) + (a \triangle b)) + (a \cdot (b \triangleleft a)) + (a \triangle b) \\
&\stackrel{(3.1),(3.3)}{=} (a \triangleleft b) + ((a \triangle b) + (a \triangle b)) + (a \cdot (b \triangleleft a)) \\
&\stackrel{(3.5)}{=} (a \triangleleft b) + (a \triangle b) + (a \cdot (b \triangleleft a)) \stackrel{(3.76)}{=} a
\end{aligned}$$

End of the proof of Theorem (3.65).

Proof of Theorem (3.66) : $(a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = (a \trianglelefteq b) \cdot (b \trianglelefteq c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$

By definition:

$$\begin{aligned}
&\left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = b \\ a \trianglelefteq c = a \end{array} \right. & \left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = b \end{array} \right. \\
(a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = a & \quad (a \trianglelefteq b) \cdot (b \trianglelefteq c) = a
\end{aligned}$$

2. Case 2: two dates out of three are equal

(a) Case 2a: $d(a) = d(b) < d(c)$

By definition:

$$\left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = b \\ a \trianglelefteq c = a \end{array} \right. \quad \left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = b \end{array} \right.$$

$$(a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = a \quad (a \trianglelefteq b) \cdot (b \trianglelefteq c) = a$$

(b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = \perp \\ a \trianglelefteq c = \perp \end{array} \right. \quad \left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = \perp \end{array} \right.$$

$$(a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = \perp \quad (a \trianglelefteq b) \cdot (b \trianglelefteq c) = \perp$$

(c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = \perp \\ a \trianglelefteq c = a \end{array} \right. \quad \left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = \perp \end{array} \right.$$

$$(a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = \perp \quad (a \trianglelefteq b) \cdot (b \trianglelefteq c) = \perp$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\left\{ \begin{array}{l} a \trianglelefteq b = \perp \\ b \trianglelefteq c = b \\ a \trianglelefteq c = a \end{array} \right. \quad \left\{ \begin{array}{l} a \trianglelefteq b = \perp \\ b \trianglelefteq c = b \end{array} \right.$$

$$(a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = \perp \quad (a \trianglelefteq b) \cdot (b \trianglelefteq c) = \perp$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\left\{ \begin{array}{l} a \trianglelefteq b = \perp \\ b \trianglelefteq c = b \\ a \trianglelefteq c = \perp \end{array} \right. \quad \left\{ \begin{array}{l} a \trianglelefteq b = \perp \\ b \trianglelefteq c = b \end{array} \right.$$

$$(a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = \perp \quad (a \trianglelefteq b) \cdot (b \trianglelefteq c) = \perp$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = b \\ a \trianglelefteq c = a \end{array} \right. \quad \left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = b \end{array} \right.$$

$$(a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = b \quad (a \trianglelefteq b) \cdot (b \trianglelefteq c) = b$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{cc} \left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = b \\ a \trianglelefteq c = a \end{array} \right. & \left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = b \end{array} \right. \\ (a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = b & (a \trianglelefteq b) \cdot (b \trianglelefteq c) = b \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{cc} \left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = \perp \\ a \trianglelefteq c = a \end{array} \right. & \left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = \perp \end{array} \right. \\ (a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = \perp & (a \trianglelefteq b) \cdot (b \trianglelefteq c) = \perp \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{cc} \left\{ \begin{array}{l} a \trianglelefteq b = \perp \\ b \trianglelefteq c = b \\ a \trianglelefteq c = a \end{array} \right. & \left\{ \begin{array}{l} a \trianglelefteq b = \perp \\ b \trianglelefteq c = b \end{array} \right. \\ (a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = \perp & (a \trianglelefteq b) \cdot (b \trianglelefteq c) = \perp \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{cc} \left\{ \begin{array}{l} a \trianglelefteq b = \perp \\ b \trianglelefteq c = b \\ a \trianglelefteq c = \perp \end{array} \right. & \left\{ \begin{array}{l} a \trianglelefteq b = \perp \\ b \trianglelefteq c = b \end{array} \right. \\ (a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = \perp & (a \trianglelefteq b) \cdot (b \trianglelefteq c) = \perp \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{cc} \left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = \perp \\ a \trianglelefteq c = \perp \end{array} \right. & \left\{ \begin{array}{l} a \trianglelefteq b = a \\ b \trianglelefteq c = \perp \end{array} \right. \\ (a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = \perp & (a \trianglelefteq b) \cdot (b \trianglelefteq c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{cc} \left\{ \begin{array}{l} a \trianglelefteq b = \perp \\ b \trianglelefteq c = \perp \\ a \trianglelefteq c = \perp \end{array} \right. & \left\{ \begin{array}{l} a \trianglelefteq b = \perp \\ b \trianglelefteq c = \perp \end{array} \right. \\ (a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = \perp & (a \trianglelefteq b) \cdot (b \trianglelefteq c) = \perp \end{array}$$

For all the cases of the domain of our study,

$$(a \preceq b) \cdot (b \preceq c) \cdot (a \preceq c) = (a \preceq b) \cdot (b \preceq c).$$

Thus:

$$(a \preceq b) \cdot (b \preceq c) \cdot (a \preceq c) = (a \preceq b) \cdot (b \preceq c)$$

End of the proof of Theorem (3.66).

A.2.5 Proofs of simplification theorems

Proof of Theorem (3.67) : $(a \preceq b) + (a \triangleleft b) = a \preceq b$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.3) : a + (b + c) = (a + b) + c$$

$$(3.5) : a + a = a$$

$$(3.16) : a \preceq b = a \triangleleft b + a \triangle b$$

$$\begin{aligned} (a \preceq b) + (a \triangleleft b) &\stackrel{(3.16)}{=} ((a \triangleleft b) + (a \triangle b)) + (a \triangleleft b) \\ &\stackrel{(3.1),(3.3)}{=} ((a \triangleleft b) + (a \triangleleft b)) + (a \triangle b) \\ &\stackrel{(3.5)}{=} (a \triangleleft b) + (a \triangle b) \stackrel{(3.16)}{=} a \preceq b \end{aligned}$$

End of the proof of Theorem (3.67).

Proof of Theorem (3.68) : $(a \preceq b) + (a \triangle b) = a \preceq b$

For this proof, we will use the following theorems:

$$(3.3) : a + (b + c) = (a + b) + c$$

$$(3.5) : a + a = a$$

$$(3.16) : a \preceq b = a \triangleleft b + a \triangle b$$

$$\begin{aligned} (a \preceq b) + (a \triangle b) &\stackrel{(3.16)}{=} ((a \triangleleft b) + (a \triangle b)) + (a \triangle b) \\ &\stackrel{(3.3)}{=} (a \triangleleft b) + ((a \triangle b) + (a \triangle b)) \\ &\stackrel{(3.5)}{=} (a \triangleleft b) + (a \triangle b) \stackrel{(3.16)}{=} a \preceq b \end{aligned}$$

End of the proof of Theorem (3.68).

Proof of Theorem (3.69) : $(a \triangleleft b) \cdot (a \triangle b) = \perp$

1. Case 1: the two dates are equal: $d(a) = d(b)$, meaning that $a = b$
By definition:

$$\begin{cases} a \triangleleft b = \perp \\ a \triangle b = a \end{cases}$$

$$(a \triangleleft b) \cdot (a \triangle b) = \perp$$

2. Case 2: the two dates are different

- (a) Case 2a: $d(a) < d(b)$
By definition:

$$\begin{cases} a \triangleleft b = a \\ a \triangle b = \perp \end{cases}$$

$$(a \triangleleft b) \cdot (a \triangle b) = \perp$$

- (b) Case 2b: $d(a) > d(b)$
By definition:

$$\begin{cases} a \triangleleft b = \perp \\ a \triangle b = \perp \end{cases}$$

$$(a \triangleleft b) \cdot (a \triangle b) = \perp$$

For all the cases of the domain of our study, $(a \triangleleft b) \cdot (a \triangle b) = \perp$. Thus:

$$(a \triangleleft b) \cdot (a \triangle b) = \perp$$

End of the proof of Theorem (3.69).

Proof of Theorem (3.70) : $(a \triangleleft b) \cdot (b \triangle c) = (a \triangleleft c) \cdot (b \triangle c)$

1. Case 1: the three dates are equal: $d(a) = d(b) = d(c)$, meaning that $a = b = c$
By definition:

$$\begin{array}{cc} \begin{cases} a \triangleleft b = \perp \\ b \triangle c = b \end{cases} & \begin{cases} a \triangleleft c = \perp \\ b \triangle c = b \end{cases} \\ (a \triangleleft b) \cdot (b \triangle c) = \perp & (a \triangleleft c) \cdot (b \triangle c) = \perp \end{array}$$

2. Case 2: two dates out of three are equal

- (a) Case 2a: $d(a) = d(b) < d(c)$
By definition:

$$\begin{array}{cc} \begin{cases} a \triangleleft b = \perp \\ b \triangle c = \perp \end{cases} & \begin{cases} a \triangleleft c = a \\ b \triangle c = \perp \end{cases} \\ (a \triangleleft b) \cdot (b \triangle c) = \perp & (a \triangleleft c) \cdot (b \triangle c) = \perp \end{array}$$

(b) Case 2b: $d(a) = d(b) > d(c)$

By definition:

$$\begin{array}{ll} \begin{cases} a \triangleleft b = \perp \\ b \triangleleft c = \perp \end{cases} & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a \triangleleft b) \cdot (b \triangleleft c) = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

(c) Case 2c: $d(a) = d(c) < d(b)$

By definition:

$$\begin{array}{ll} \begin{cases} a \triangleleft b = a \\ b \triangleleft c = \perp \end{cases} & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a \triangleleft b) \cdot (b \triangleleft c) = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

(d) Case 2d: $d(a) = d(c) > d(b)$

By definition:

$$\begin{array}{ll} \begin{cases} a \triangleleft b = \perp \\ b \triangleleft c = \perp \end{cases} & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a \triangleleft b) \cdot (b \triangleleft c) = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

(e) Case 2e: $d(b) = d(c) < d(a)$

By definition:

$$\begin{array}{ll} \begin{cases} a \triangleleft b = \perp \\ b \triangleleft c = b \end{cases} & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = b \end{cases} \\ (a \triangleleft b) \cdot (b \triangleleft c) = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

(f) Case 2f: $d(b) = d(c) > d(a)$

By definition:

$$\begin{array}{ll} \begin{cases} a \triangleleft b = a \\ b \triangleleft c = b \end{cases} & \begin{cases} a \triangleleft c = a \\ b \triangleleft c = b \end{cases} \\ (a \triangleleft b) \cdot (b \triangleleft c) = b & (a \triangleleft c) \cdot (b \triangleleft c) = b \end{array}$$

3. Case 3: the three dates are different

(a) Case 3a: $d(a) < d(b) < d(c)$

By definition:

$$\begin{array}{ll} \begin{cases} a \triangleleft b = a \\ b \triangleleft c = \perp \end{cases} & \begin{cases} a \triangleleft c = a \\ b \triangleleft c = \perp \end{cases} \\ (a \triangleleft b) \cdot (b \triangleleft c) = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

(b) Case 3b: $d(a) < d(c) < d(b)$

By definition:

$$\begin{array}{ll} \begin{cases} a \triangleleft b = a \\ b \triangleleft c = \perp \end{cases} & \begin{cases} a \triangleleft c = a \\ b \triangleleft c = \perp \end{cases} \\ (a \triangleleft b) \cdot (b \triangleleft c) = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

(c) Case 3c: $d(b) < d(a) < d(c)$

By definition:

$$\begin{array}{ll} \begin{cases} a \triangleleft b = \perp \\ b \triangleleft c = \perp \end{cases} & \begin{cases} a \triangleleft c = a \\ b \triangleleft c = \perp \end{cases} \\ (a \triangleleft b) \cdot (b \triangleleft c) = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

(d) Case 3d: $d(b) < d(c) < d(a)$

By definition:

$$\begin{array}{ll} \begin{cases} a \triangleleft b = \perp \\ b \triangleleft c = \perp \end{cases} & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a \triangleleft b) \cdot (b \triangleleft c) = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

(e) Case 3e: $d(c) < d(a) < d(b)$

By definition:

$$\begin{array}{ll} \begin{cases} a \triangleleft b = a \\ b \triangleleft c = \perp \end{cases} & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a \triangleleft b) \cdot (b \triangleleft c) = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

(f) Case 3f: $d(c) < d(b) < d(a)$

By definition:

$$\begin{array}{ll} \begin{cases} a \triangleleft b = \perp \\ b \triangleleft c = \perp \end{cases} & \begin{cases} a \triangleleft c = \perp \\ b \triangleleft c = \perp \end{cases} \\ (a \triangleleft b) \cdot (b \triangleleft c) = \perp & (a \triangleleft c) \cdot (b \triangleleft c) = \perp \end{array}$$

For all the cases of the domain of our study, $(a \triangleleft b) \cdot (b \triangleleft c) = (a \triangleleft c) \cdot (b \triangleleft c)$.
Thus:

$$(a \triangleleft b) \cdot (b \triangleleft c) = (a \triangleleft c) \cdot (b \triangleleft c)$$

End of the proof of Theorem (3.70).

Proof of Theorem (3.71) : $(a \triangleleft b) \cdot (a \triangleleft b) = a \triangleleft b$

For this proof, we will use the following theorems:

$$(3.2) : a \cdot b = b \cdot a$$

$$(3.14) : a \cdot (a + b) = a$$

$$(3.16) : a \triangleleft b = a \triangleleft b + a \triangleleft b$$

$$(a \triangleleft b) \cdot (a \triangleleft b) \stackrel{(3.16)}{=} ((a \triangleleft b) + (a \triangleleft b)) \cdot (a \triangleleft b) \stackrel{(3.2)}{=} (a \triangleleft b) \cdot ((a \triangleleft b) + (a \triangleleft b)) \stackrel{(3.14)}{=} a \triangleleft b$$

End of the proof of Theorem (3.71).

Proof of Theorem (3.72) : $(a \triangleleft b) \cdot (b \triangleleft a) = \perp$

For this proof, we will use the following theorems:

$$(3.5) : a + a = a$$

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.16) : a \triangleleft b = a \triangleleft b + a \triangle b$$

$$(3.17) : (a \triangleleft b) \cdot (b \triangleleft a) = \perp$$

$$(3.35) : a \triangle b = b \triangle a$$

$$(3.69) : (a \triangleleft b) \cdot (a \triangle b) = \perp$$

$$\begin{aligned} (a \triangleleft b) \cdot (b \triangleleft a) &\stackrel{(3.16)}{=} (a \triangleleft b) \cdot ((b \triangleleft a) + (b \triangle a)) \\ &\stackrel{(3.7)}{=} (a \triangleleft b) \cdot (b \triangleleft a) + (a \triangleleft b) \cdot (b \triangle a) \\ &\stackrel{(3.35)}{=} (a \triangleleft b) \cdot (b \triangleleft a) + (a \triangleleft b) \cdot (a \triangle b) \stackrel{(3.17),(3.69)}{=} \perp + \perp \stackrel{(3.5)}{=} \perp \end{aligned}$$

End of the proof of Theorem (3.72).

Proof of Theorem (3.73) : $(a \triangleleft b) \cdot (a \triangle b) = a \triangle b$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.2) : a \cdot b = b \cdot a$$

$$(3.14) : a \cdot (a + b) = a$$

$$(3.16) : a \triangleleft b = a \triangleleft b + a \triangle b$$

$$(a \triangleleft b) \cdot (a \triangle b) \stackrel{(3.16)}{=} ((a \triangleleft b) + (a \triangle b)) \cdot (a \triangle b) \stackrel{(3.1),(3.2)}{=} (a \triangle b) \cdot ((a \triangle b) + (a \triangleleft b)) \stackrel{(3.14)}{=} a \triangle b$$

End of the proof of Theorem (3.73).

Proof of Theorem (3.74) : $(a \triangleleft b) + (a \triangle b) + (b \triangleleft a) = a + b$

1. Case 1: the two dates are equal: $d(a) = d(b)$, meaning that $a = b$

By definition:

$$\begin{cases} a \triangleleft b = \perp \\ a \triangle b = a \\ b \triangleleft a = \perp \end{cases} \quad a + b = a$$

$$(a \triangleleft b) + (a \triangle b) + (b \triangleleft a) = a$$

2. Case 2: the two dates are different

(a) Case 2a: $d(a) < d(b)$

By definition:

$$\begin{cases} a \triangleleft b = a \\ a \triangle b = \perp \\ b \triangleleft a = \perp \end{cases} \quad a + b = a$$

$$(a \triangleleft b) + (a \triangle b) + (b \triangleleft a) = a$$

(b) Case 2b: $d(a) > d(b)$

By definition:

$$\begin{cases} a \triangleleft b = \perp \\ a \triangle b = \perp \\ b \triangleleft a = b \end{cases} \quad a + b = b$$

$$(a \triangleleft b) + (a \triangle b) + (b \triangleleft a) = b$$

For all the cases of the domain of our study, $(a \triangleleft b) + (a \triangle b) + (b \triangleleft a) = a + b$.

Thus:

$$(a \triangleleft b) + (a \triangle b) + (b \triangleleft a) = a + b$$

End of the proof of Theorem (3.74).

Proof of Theorem (3.75) : $(a \cdot (b \triangleleft a)) + (a \triangle b) + (b \cdot (a \triangleleft b)) = a \cdot b$

For this proof, we will use the following theorems:

$$(3.1) : a + b = b + a$$

$$(3.2) : a \cdot b = b \cdot a$$

$$(3.4) : a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.14) : a \cdot (a + b) = a$$

$$(3.33) : a \cdot (a \triangleleft b) = a \triangleleft b$$

$$(3.44) : a \cdot (a \triangle b) = a \triangle b$$

$$(3.74) : (a \triangleleft b) + (a \triangle b) + (b \triangleleft a) = a + b$$

$$\begin{aligned} (a \cdot (b \triangleleft a)) + (a \triangle b) + (b \cdot (a \triangleleft b)) &\stackrel{(3.33),(3.44)}{=} (a \cdot (b \cdot (b \triangleleft a))) + (a \cdot b \cdot (a \triangle b)) \\ &\quad + (b \cdot (a \cdot (a \triangleleft b))) \\ &\stackrel{(3.2),(3.4)}{=} (a \cdot b \cdot (b \triangleleft a)) + (a \cdot b \cdot (a \triangle b)) \\ &\quad + (a \cdot b \cdot (a \triangleleft b)) \\ &\stackrel{(3.1),(3.7)}{=} a \cdot b \cdot ((a \triangleleft b) + (a \triangle b) + (b \triangleleft a)) \\ &\stackrel{(3.74)}{=} a \cdot b \cdot (a + b) \\ &\stackrel{(3.2),(3.4)}{=} (a \cdot (a + b)) \cdot b \stackrel{(3.14)}{=} a \cdot b \end{aligned}$$

End of the proof of Theorem (3.75).

Proof of Theorem (3.76) : $(a \triangleleft b) + (a \triangle b) + (a \cdot (b \triangleleft a)) = a$

For this proof, we will use the following theorems:

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.14) : a \cdot (a + b) = a$$

$$(3.33) : a \cdot (a \triangleleft b) = a \triangleleft b$$

$$(3.44) : a \cdot (a \triangle b) = a \triangle b$$

$$(3.74) : (a \triangleleft b) + (a \triangle b) + (b \triangleleft a) = a + b$$

$$\begin{aligned} (a \triangleleft b) + (a \triangle b) + (a \cdot (b \triangleleft a)) &\stackrel{(3.33),(3.44)}{=} (a \cdot (a \triangleleft b)) + (a \cdot (a \triangle b)) \\ &\quad + (a \cdot (b \triangleleft a)) \\ &\stackrel{(3.7)}{=} a \cdot ((a \triangleleft b) + (a \triangle b) + (b \triangleleft a)) \\ &\stackrel{(3.74)}{=} a \cdot (a + b) \stackrel{(3.14)}{=} a \end{aligned}$$

End of the proof of Theorem (3.76).

Proof of Theorem (3.77) : $(a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \trianglelefteq c) = (a \triangleleft b) \cdot (b \triangleleft c)$

For this proof, we will use the following theorems:

$$(3.7) : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$$

$$(3.13) : a + (a \cdot b) = a$$

$$(3.16) : a \trianglelefteq b = a \triangleleft b + a \triangle b$$

$$(3.34) : (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) = (a \triangleleft b) \cdot (b \triangleleft c)$$

$$\begin{aligned} (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \trianglelefteq c) &\stackrel{(3.16)}{=} (a \triangleleft b) \cdot (b \triangleleft c) \cdot ((a \triangleleft c) + (a \triangle b)) \\ &\stackrel{(3.7)}{=} (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangleleft c) + (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangle b) \\ &\stackrel{(3.34)}{=} (a \triangleleft b) \cdot (b \triangleleft c) + (a \triangleleft b) \cdot (b \triangleleft c) \cdot (a \triangle b) \\ &\stackrel{(3.13)}{=} (a \triangleleft b) \cdot (b \triangleleft c) \end{aligned}$$

End of the proof of Theorem (3.77).

Application of probabilistic models to the case of exponential distributions

B.1 Case of a single Spare gate with 2 input events

B.1.1 Result obtained with Markov Chains

The state transition diagram of a single Spare gate with 2 input events is shown in Fig. B.1. States 4 and 5 represent the failure of the Spare gate: state 4 represents the occurrence of the failure sequence $[A, B]$ whereas state 5 represents the occurrence of the failure sequence $[B, A]$.

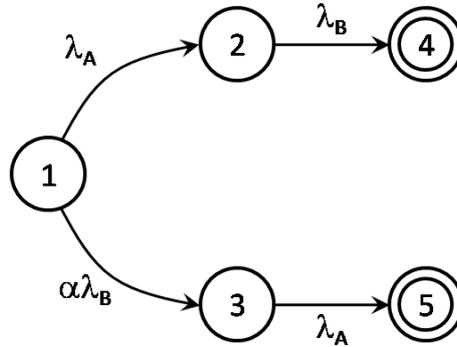


Figure B.1: State transition diagram of a single Spare gate with 2 input events

The corresponding differential system is

$$\begin{cases} \frac{dP_1(t)}{dt} = -(\lambda_A + \alpha\lambda_B)P_1(t) \\ \frac{dP_2(t)}{dt} = \lambda_A P_1(t) - \lambda_B P_2(t) \\ \frac{dP_3(t)}{dt} = \alpha\lambda_B P_1(t) - \lambda_A P_3(t) \\ \frac{dP_4(t)}{dt} = \lambda_B P_2(t) \\ \frac{dP_5(t)}{dt} = \lambda_A P_3(t) \end{cases} \quad (\text{B.1})$$

Solving this differential system (B.1) allows to determine the failure probability

of the Spare gate

$$\begin{aligned} Pr \{Q\} (t) &= P_4(t) + P_5(t) \\ &= 1 - e^{-\lambda_A t} - \frac{\lambda_A}{\lambda_A - (1 - \alpha)\lambda_B} e^{-\lambda_B t} + \frac{\lambda_A}{\lambda_A - (1 - \alpha)\lambda_B} e^{-(\lambda_A + \alpha\lambda_B)t} \end{aligned}$$

B.1.2 Result obtained with our probabilistic model

According to Section 5.4.2.1, the probabilistic model of a single Spare gate with 2 input events is

$$F_Q(t) = Pr \{Q\} (t) = \int_0^t \left(\int_v^t f_{B_a}(u, v) du \right) f_A(v) dv + \int_0^t f_A(u) F_{B_d}(u) du.$$

In the case of exponential distributions,

$$\begin{cases} F_A(t) = 1 - e^{-\lambda_A t}, \forall t \geq 0 \\ F_{B_d}(t) = 1 - e^{-\alpha\lambda_B t}, \forall t \geq 0 \\ F_{B_a}(t, t_A) = 1 - e^{-\lambda_B(t - (1 - \alpha)t_A)}, \forall t \geq (1 - \alpha)t_A \end{cases}$$

On the one hand,

$$\begin{aligned} Pr \{B_a \cdot (A \triangleleft B_a)\} (t) &= \int_0^t \left(\int_v^t f_{B_a}(u, v) du \right) f_A(v) dv \\ &= \int_0^t \left(\int_v^t \lambda_B e^{-\lambda_B(u - (1 - \alpha)v)} du \right) \lambda_A e^{-\lambda_A v} dv \\ &= \int_0^t \left(e^{-\lambda_B(v - (1 - \alpha)v)} - e^{-\lambda_B(t - (1 - \alpha)v)} \right) \lambda_A e^{-\lambda_A v} dv \\ &= \int_0^t \left(e^{-\alpha\lambda_B v} - e^{-\lambda_B(t - (1 - \alpha)v)} \right) \lambda_A e^{-\lambda_A v} dv \\ &= \int_0^t \left(\lambda_A e^{-(\lambda_A + \alpha\lambda_B)v} - \lambda_A e^{-\lambda_B t} e^{-(\lambda_A - (1 - \alpha)\lambda_B)v} \right) dv \\ &= \int_0^t \lambda_A e^{-(\lambda_A + \alpha\lambda_B)v} dv - e^{-\lambda_B t} \int_0^t \lambda_A e^{-(\lambda_A - (1 - \alpha)\lambda_B)v} dv \\ &= \frac{\lambda_A}{\lambda_A + \alpha\lambda_B} (1 - e^{-(\lambda_A + \alpha\lambda_B)t}) \\ &\quad - \frac{\lambda_A}{\lambda_A - (1 - \alpha)\lambda_B} e^{-\lambda_B t} (1 - e^{-(\lambda_A - (1 - \alpha)\lambda_B)t}) \\ &= \frac{\lambda_A}{\lambda_A + \alpha\lambda_B} (1 - e^{-(\lambda_A + \alpha\lambda_B)t}) \\ &\quad - \frac{\lambda_A}{\lambda_A - (1 - \alpha)\lambda_B} (e^{-\lambda_B t} - e^{-(\lambda_A + \alpha\lambda_B)t}) \end{aligned}$$

$$\begin{aligned}
Pr\{B_a \cdot (A \triangleleft B_a)\}(t) &= \frac{\lambda_A}{\lambda_A + \alpha\lambda_B} - \frac{\lambda_A}{\lambda_A - (1 - \alpha)\lambda_B} e^{-\lambda_B t} \\
&+ \left(\frac{\lambda_A}{\lambda_A - (1 - \alpha)\lambda_B} - \frac{\lambda_A}{\lambda_A + \alpha\lambda_B} \right) e^{-(\lambda_A + \alpha\lambda_B)t} \\
&= \frac{\lambda_A}{\lambda_A + \alpha\lambda_B} - \frac{\lambda_A}{\lambda_A - (1 - \alpha)\lambda_B} e^{-\lambda_B t} \\
&+ \frac{\lambda_A(\lambda_A + \alpha\lambda_B) - \lambda_A(\lambda_A - (1 - \alpha)\lambda_B)}{(\lambda_A + \alpha\lambda_B)(\lambda_A - (1 - \alpha)\lambda_B)} e^{-(\lambda_A + \alpha\lambda_B)t} \\
&= \frac{\lambda_A}{\lambda_A + \alpha\lambda_B} - \frac{\lambda_A}{\lambda_A - (1 - \alpha)\lambda_B} e^{-\lambda_B t} \\
&+ \frac{\lambda_A\lambda_B}{(\lambda_A + \alpha\lambda_B)(\lambda_A - (1 - \alpha)\lambda_B)} e^{-(\lambda_A + \alpha\lambda_B)t}
\end{aligned}$$

On the other hand,

$$\begin{aligned}
Pr\{A \cdot (B_d \triangleleft A)\}(t) &= \int_0^t F_{B_d}(u) f_A(u) du \\
&= \int_0^t (1 - e^{-\alpha\lambda_B u}) \lambda_A e^{-\lambda_A u} du \\
&= \int_0^t \lambda_A e^{-\lambda_A u} du - \int_0^t \lambda_A e^{-(\lambda_A + \alpha\lambda_B)u} du \\
&= 1 - e^{-\lambda_A t} - \frac{\lambda_A}{\lambda_A + \alpha\lambda_B} (1 - e^{-(\lambda_A + \alpha\lambda_B)t}) \\
&= \left(1 - \frac{\lambda_A}{\lambda_A + \alpha\lambda_B} \right) - e^{-\lambda_A t} + \frac{\lambda_A}{\lambda_A + \alpha\lambda_B} e^{-(\lambda_A + \alpha\lambda_B)t} \\
&= \frac{\alpha\lambda_B}{\lambda_A + \alpha\lambda_B} - e^{-\lambda_A t} + \frac{\lambda_A}{\lambda_A + \alpha\lambda_B} e^{-(\lambda_A + \alpha\lambda_B)t}
\end{aligned}$$

The failure probability obtained with the probabilistic model hence is

$$Pr\{Q\}(t) = 1 - e^{-\lambda_A t} - \frac{\lambda_A}{\lambda_A - (1 - \alpha)\lambda_B} e^{-\lambda_B t} + \frac{\lambda_A}{\lambda_A - (1 - \alpha)\lambda_B} e^{-(\lambda_A + \alpha\lambda_B)t},$$

and it can be noted that this failure probability is the same as the failure probability obtained in Section B.1.1. However, the probabilistic model provided in Section 5.4.2.1 does not depend on the failure distribution considered for basic events.

B.2 Case of 2 Spare gates sharing a spare event

B.2.1 Result obtained with Markov Chains

The state transition diagram of 2 Spare gates sharing a spare event is shown in Fig. B.2.

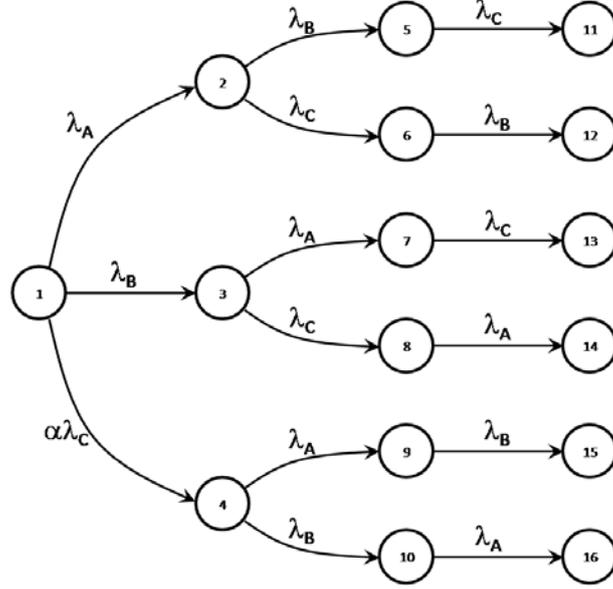


Figure B.2: State transition diagram of 2 Spare gates sharing a spare event

$Q1$ fails if $C_a \cdot (A \triangleleft C_a)$ holds (states 6, 11, 12, and 13), if $A \cdot (C_d \triangleleft A)$ (states 9, 15, and 16), or if $A \cdot (B \triangleleft A)$ holds (states 7, 13, 14, and 16). States 6, 7, 9, and 11 to 16 hence represent the failure of $Q1$.

$Q2$ fails if $C_a \cdot (B \triangleleft C_a)$ holds (states 8, 11, 13, and 14), if $B \cdot (C_d \triangleleft B)$ holds (states 10, 15, and 16), or if $B \cdot (A \triangleleft B)$ holds (states 5, 11, 12, and 15). States 5, 8, 10, and 11 to 16 hence represent the failure of $Q2$.

Solving the differential system which is equivalent to the state transition diagram in Fig. B.2 allows to determine the failure probability of $Q1$, and $Q2$:

$$\begin{aligned}
 Pr \{Q1\} (t) &= P_6(t) + P_7(t) + P_9(t) + P_{11}(t) + P_{12}(t) \\
 &\quad + P_{13}(t) + P_{14}(t) + P_{15}(t) + P_{16}(t) \\
 Pr \{Q2\} (t) &= P_5(t) + P_8(t) + P_{10}(t) + P_{11}(t) + P_{12}(t) \\
 &\quad + P_{13}(t) + P_{14}(t) + P_{15}(t) + P_{16}(t)
 \end{aligned}$$

The expressions for both probabilities will not be detailed here because of their size.

B.2.2 Result obtained with our probabilistic model

According to Section 5.4.2.2, the probabilistic model of 2 Spare gates sharing a spare event is

$$\begin{aligned}
F_{Q1}(t) = Pr\{Q1\}(t) &= \int_0^t \left(\int_w^t \left(\int_w^u f_B(v)dv \right) f_{C_a}(u, w)du \right) f_A(w)dw \\
&+ \int_0^t \left(\int_0^u \left(\int_v^u f_{C_a}(w, v)dw \right) f_A(v)dv \right) f_B(u)du \\
&+ (1 - F_B(t)) \int_0^t \left(\int_v^t f_{C_a}(u, v)du \right) f_A(v)dv \\
&+ \int_0^t \left(\int_0^u f_A(v)F_{C_d}(v)dv \right) f_B(u)du \\
&+ (1 - F_B(t)) \int_0^t f_A(u)F_{C_d}(u)du + \int_0^t f_A(u)F_B(u)du \\
F_{Q2}(t) = Pr\{Q2\}(t) &= \int_0^t \left(\int_w^t \left(\int_w^u f_A(v)dv \right) f_{C_a}(u, w)du \right) f_B(w)dw \\
&+ \int_0^t \left(\int_0^u \left(\int_v^u f_{C_a}(w, v)dw \right) f_B(v)dv \right) f_A(u)du \\
&+ (1 - F_A(t)) \int_0^t \left(\int_v^t f_{C_a}(u, v)du \right) f_B(v)dv \\
&+ \int_0^t \left(\int_0^u f_B(v)F_{C_d}(v)dv \right) f_A(u)du \\
&+ (1 - F_A(t)) \int_0^t f_B(u)F_{C_d}(u)du + \int_0^t f_B(u)F_A(u)du
\end{aligned}$$

In the case of exponential distributions,

$$\begin{cases}
F_A(t) = 1 - e^{-\lambda_A t}, \forall t \geq 0 \\
F_B(t) = 1 - e^{-\lambda_B t}, \forall t \geq 0 \\
F_{C_d}(t) = 1 - e^{-\alpha \lambda_C t}, \forall t \geq 0 \\
F_{C_a}(t, v) = 1 - e^{-\lambda_C (t - (1-\alpha)v)}, \forall t \geq (1-\alpha)v
\end{cases}$$

where v is the failure date of the event on which C depends. Indeed, C becomes active as soon as A or B fails, so the failure distribution of C depends on t_A if A fails first, and on t_B if B fails first.

Besides, the terms of the behavioural model can be identified to the states of the state transition diagram in Fig. B.2:

$$\begin{aligned}
Pr\{C_a \cdot (A \triangleleft B) \cdot (B \triangleleft C_a)\}(t) &= P_{11}(t) \\
Pr\{B \cdot (A \triangleleft C_a) \cdot (C_a \triangleleft B)\}(t) &= P_{12}(t) \\
Pr\{C_a \cdot (A \triangleleft C_a) \cdot \mathcal{B}\}(t) &= P_6(t) \\
Pr\{B \cdot (C_d \triangleleft A) \cdot (A \triangleleft B)\}(t) &= P_{15}(t) \\
Pr\{A \cdot (C_d \triangleleft A) \cdot \mathcal{B}\}(t) &= P_9(t) \\
Pr\{A \cdot (B \triangleleft A)\}(t) &= P_7(t) + P_{13}(t) + P_{14}(t) + P_{16}(t)
\end{aligned}$$

Let us apply the probabilistic model for $Q1$ to the case of exponential distributions.

- Validation of the expression for $Pr \{C_a \cdot (A \triangleleft B) \cdot (B \triangleleft C_a)\} (t)$

$$Pr \{C_a \cdot (A \triangleleft B) \cdot (B \triangleleft C_a)\} (t) = \int_0^t \left(\int_w^t \left(\int_w^u f_B(v) dv \right) f_{C_a}(u, w) du \right) f_A(w) dw$$

$$\int_w^u f_B(v) dv = \int_w^u \lambda_B e^{-\lambda_B v} dv = e^{-\lambda_B w} - e^{-\lambda_B u}$$

$$\begin{aligned} & \int_w^t \left(\int_w^u f_B(v) dv \right) f_{C_a}(u, w) du \\ &= \int_w^t \left(e^{-\lambda_B w} - e^{-\lambda_B u} \right) \lambda_C e^{-\lambda_C(u-(1-\alpha)w)} du \\ &= e^{-\lambda_B w} \int_w^t \lambda_C e^{-\lambda_C(u-(1-\alpha)w)} du - \int_w^t \lambda_C e^{-\lambda_B u} e^{-\lambda_C(u-(1-\alpha)w)} du \\ &= e^{-\lambda_B w} \int_w^t \lambda_C e^{-\lambda_C(u-(1-\alpha)w)} du - \int_w^t \lambda_C e^{-((\lambda_B+\lambda_C)u-\lambda_C(1-\alpha)w)} du \\ &= e^{-\lambda_B w} \left(e^{-\lambda_C(w-(1-\alpha)w)} - e^{-\lambda_C(t-(1-\alpha)w)} \right) \\ &\quad - \frac{\lambda_C}{\lambda_B + \lambda_C} \left(e^{-((\lambda_B+\lambda_C)w-\lambda_C(1-\alpha)w)} - e^{-((\lambda_B+\lambda_C)t-\lambda_C(1-\alpha)w)} \right) \\ &= e^{-(\lambda_B+\alpha\lambda_C)w} - e^{-(\lambda_C t+(\lambda_B-(1-\alpha)\lambda_C)w)} \\ &\quad - \frac{\lambda_C}{\lambda_B + \lambda_C} \left(e^{-(\lambda_B+\alpha\lambda_C)w} - e^{-((\lambda_B+\lambda_C)t-\lambda_C(1-\alpha)w)} \right) \\ &= \frac{\lambda_B}{\lambda_B + \lambda_C} e^{-(\lambda_B+\alpha\lambda_C)w} - e^{-(\lambda_C t+(\lambda_B-(1-\alpha)\lambda_C)w)} \\ &\quad + \frac{\lambda_C}{\lambda_B + \lambda_C} e^{-((\lambda_B+\lambda_C)t-\lambda_C(1-\alpha)w)} \end{aligned}$$

$$\begin{aligned}
& Pr \{C_a \cdot (A \triangleleft B) \cdot (B \triangleleft C_a)\} (t) \\
&= \int_0^t \left(\int_w^t \left(\int_w^u f_B(v) dv \right) f_{C_a}(u, w) du \right) f_A(w) dw \\
&= \int_0^t \frac{\lambda_B}{\lambda_B + \lambda_C} e^{-(\lambda_B + \alpha \lambda_C)w} \lambda_A e^{-\lambda_A w} dw \\
&\quad - \int_0^t e^{-(\lambda_C t + (\lambda_B - (1-\alpha)\lambda_C)w)} \lambda_A e^{-\lambda_A w} dw \\
&\quad + \int_0^t \frac{\lambda_C}{\lambda_B + \lambda_C} e^{-((\lambda_B + \lambda_C)t - \lambda_C(1-\alpha)w)} \lambda_A e^{-\lambda_A w} dw \\
&= \frac{\lambda_A \lambda_B}{\lambda_B + \lambda_C} \int_0^t e^{-(\lambda_A + \lambda_B + \alpha \lambda_C)w} dw - \lambda_A \int_0^t e^{-(\lambda_C t + (\lambda_A + \lambda_B - (1-\alpha)\lambda_C)w)} dw \\
&\quad + \frac{\lambda_A \lambda_C}{\lambda_B + \lambda_C} \int_0^t e^{-((\lambda_B + \lambda_C)t + (\lambda_A - (1-\alpha)\lambda_C)w)} dw \\
&= \frac{\lambda_A \lambda_B}{(\lambda_B + \lambda_C)(\lambda_A + \lambda_B + \alpha \lambda_C)} (1 - e^{-(\lambda_A + \lambda_B + \alpha \lambda_C)t}) \\
&\quad - \frac{\lambda_A}{\lambda_A + \lambda_B - (1-\alpha)\lambda_C} (e^{-\lambda_C t} - e^{-(\lambda_C t + (\lambda_A + \lambda_B - (1-\alpha)\lambda_C)t)}) \\
&\quad + \frac{\lambda_A \lambda_C (e^{-(\lambda_B + \lambda_C)t} - e^{-((\lambda_B + \lambda_C)t + (\lambda_A - (1-\alpha)\lambda_C)t})})}{(\lambda_B + \lambda_C)(\lambda_A - (1-\alpha)\lambda_C)} \\
&= \frac{\lambda_A \lambda_B}{(\lambda_B + \lambda_C)(\lambda_A + \lambda_B + \alpha \lambda_C)} (1 - e^{-(\lambda_A + \lambda_B + \alpha \lambda_C)t}) \\
&\quad - \frac{\lambda_A}{\lambda_A + \lambda_B - (1-\alpha)\lambda_C} (e^{-\lambda_C t} - e^{-(\lambda_A + \lambda_B + \alpha \lambda_C)t}) \\
&\quad + \frac{\lambda_A \lambda_C}{(\lambda_B + \lambda_C)(\lambda_A - (1-\alpha)\lambda_C)} (e^{-(\lambda_B + \lambda_C)t} - e^{-(\lambda_A + \lambda_B + \alpha \lambda_C)t}) \\
&= \frac{\lambda_A \lambda_B}{(\lambda_B + \lambda_C)(\lambda_A + \lambda_B + \alpha \lambda_C)} - \frac{\lambda_A}{\lambda_A + \lambda_B - (1-\alpha)\lambda_C} e^{-\lambda_C t} \\
&\quad + \frac{\lambda_A \lambda_C}{(\lambda_B + \lambda_C)(\lambda_A - (1-\alpha)\lambda_C)} e^{-(\lambda_B + \lambda_C)t} \\
&\quad - \frac{\lambda_A \lambda_B \lambda_C e^{-(\lambda_A + \lambda_B + \alpha \lambda_C)t}}{(\lambda_A - (1-\alpha)\lambda_C)(\lambda_A + \lambda_B - (1-\alpha)\lambda_C)(\lambda_A + \lambda_B + \alpha \lambda_C)} \\
&= P_{11}(t)
\end{aligned}$$

- Validation of the expression for $Pr \{B \cdot (A \triangleleft C_a) \cdot (C_a \triangleleft B)\} (t)$

$$Pr \{B \cdot (A \triangleleft C_a) \cdot (C_a \triangleleft B)\} (t) = \int_0^t \left(\int_0^u \left(\int_v^u f_{C_a}(w, v) dw \right) f_A(v) dv \right) f_B(u) du$$

$$\begin{aligned}
\int_v^u f_{C_a}(w, v) dw &= \int_v^u \lambda_C e^{-\lambda_C(w - (1-\alpha)v)} dw \\
&= e^{-\lambda_C(v - (1-\alpha)v)} - e^{-\lambda_C(u - (1-\alpha)v)} = e^{-\alpha \lambda_C v} - e^{-\lambda_C(u - (1-\alpha)v)}
\end{aligned}$$

$$\begin{aligned}
 & \int_0^u \left(\int_v^u f_{C_a}(w, v) dw \right) f_A(v) dv \\
 &= \int_0^u \left(e^{-\alpha\lambda_C v} - e^{-\lambda_C(u-(1-\alpha)v)} \right) \lambda_A e^{-\lambda_A v} dv \\
 &= \lambda_A \int_0^u e^{-\alpha\lambda_C v} e^{-\lambda_A v} dv - \lambda_A \int_0^u e^{-\lambda_C(u-(1-\alpha)v)} e^{-\lambda_A v} dv \\
 &= \lambda_A \int_0^u e^{-(\lambda_A + \alpha\lambda_C)v} dv - \lambda_A \int_0^u e^{-(\lambda_C u + (\lambda_A - (1-\alpha)\lambda_C)v)} dv \\
 &= \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} (1 - e^{-(\lambda_A + \alpha\lambda_C)u}) \\
 &\quad - \frac{\lambda_A}{\lambda_A - (1-\alpha)\lambda_C} (e^{-\lambda_C u} - e^{-(\lambda_C u + (\lambda_A - (1-\alpha)\lambda_C)u)}) \\
 &= \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} (1 - e^{-(\lambda_A + \alpha\lambda_C)u}) - \frac{\lambda_A}{\lambda_A - (1-\alpha)\lambda_C} (e^{-\lambda_C u} - e^{-(\lambda_A + \alpha\lambda_C)u}) \\
 &= \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} - \frac{\lambda_A}{\lambda_A - (1-\alpha)\lambda_C} e^{-\lambda_C u} \\
 &\quad + \frac{\lambda_A \lambda_C}{(\lambda_A - (1-\alpha)\lambda_C)(\lambda_A + \alpha\lambda_C)} e^{-(\lambda_A + \alpha\lambda_C)u}
 \end{aligned}$$

$$\begin{aligned}
 & Pr \{B \cdot (A \triangleleft C_a) \cdot (C_a \triangleleft B)\} (t) \\
 &= \int_0^t \left(\int_0^u \left(\int_v^u f_{C_a}(w, v) dw \right) f_A(v) dv \right) f_B(u) du \\
 &= \int_0^t \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} \lambda_B e^{-\lambda_B u} du - \int_0^t \frac{\lambda_A}{\lambda_A - (1-\alpha)\lambda_C} e^{-\lambda_C u} \lambda_B e^{-\lambda_B u} du \\
 &\quad + \int_0^t \frac{\lambda_A \lambda_C}{(\lambda_A - (1-\alpha)\lambda_C)(\lambda_A + \alpha\lambda_C)} e^{-(\lambda_A + \alpha\lambda_C)u} \lambda_B e^{-\lambda_B u} du \\
 &= \frac{\lambda_A \lambda_B}{\lambda_A + \alpha\lambda_C} \int_0^t e^{-\lambda_B u} du - \frac{\lambda_A \lambda_B}{\lambda_A - (1-\alpha)\lambda_C} \int_0^t e^{-(\lambda_B + \lambda_C)u} du \\
 &\quad + \frac{\lambda_A \lambda_B \lambda_C}{(\lambda_A - (1-\alpha)\lambda_C)(\lambda_A + \alpha\lambda_C)} \int_0^t e^{-(\lambda_A + \lambda_B + \alpha\lambda_C)u} du \\
 &= \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} (1 - e^{-\lambda_B t}) - \frac{\lambda_A \lambda_B}{(\lambda_B + \lambda_C)(\lambda_A - (1-\alpha)\lambda_C)} (1 - e^{-(\lambda_B + \lambda_C)t}) \\
 &\quad + \frac{\lambda_A \lambda_B \lambda_C (1 - e^{-(\lambda_A + \lambda_B + \alpha\lambda_C)t})}{(\lambda_A - (1-\alpha)\lambda_C)(\lambda_A + \alpha\lambda_C)(\lambda_A + \lambda_B + \alpha\lambda_C)} \\
 &= \frac{\lambda_A \lambda_C}{(\lambda_B + \lambda_C)(\lambda_A + \lambda_B + \alpha\lambda_C)} - \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} e^{-\lambda_B t} \\
 &\quad + \frac{\lambda_A \lambda_B}{(\lambda_B + \lambda_C)(\lambda_A - (1-\alpha)\lambda_C)} e^{-(\lambda_B + \lambda_C)t} \\
 &\quad - \frac{\lambda_A \lambda_B \lambda_C}{(\lambda_A - (1-\alpha)\lambda_C)(\lambda_A + \alpha\lambda_C)(\lambda_A + \lambda_B + \alpha\lambda_C)} e^{-(\lambda_A + \lambda_B + \alpha\lambda_C)t} \\
 &= P_{12}(t)
 \end{aligned}$$

- Validation of the expression for $Pr \{C_a \cdot (A \triangleleft C_a) \cdot \mathcal{B}\} (t)$

$$Pr \{C_a \cdot (A \triangleleft C_a) \cdot \mathcal{B}\} (t) = (1 - F_B(t)) \int_0^t \left(\int_v^t f_{C_a}(u, v) du \right) f_A(v) dv$$

$$\begin{aligned} \int_v^t f_{C_a}(u, v) du &= \int_v^t \lambda_C e^{-\lambda_C(u-(1-\alpha)v)} du \\ &= e^{-\lambda_C(v-(1-\alpha)v)} - e^{-\lambda_C(t-(1-\alpha)v)} = e^{-\alpha\lambda_C v} - e^{-\lambda_C(t-(1-\alpha)v)} \end{aligned}$$

$$\begin{aligned} &\int_0^t \left(\int_v^t f_{C_a}(u, v) du \right) f_A(v) dv \\ &= \int_0^t \left(e^{-\alpha\lambda_C v} - e^{-\lambda_C(t-(1-\alpha)v)} \right) \lambda_A e^{-\lambda_A v} dv \\ &= \lambda_A \int_0^t e^{-\alpha\lambda_C v} e^{-\lambda_A v} dv - \lambda_A \int_0^t e^{-\lambda_C(t-(1-\alpha)v)} e^{-\lambda_A v} dv \\ &= \lambda_A \int_0^t e^{-(\lambda_A + \alpha\lambda_C)v} dv - \lambda_A \int_0^t e^{-(\lambda_C t + (\lambda_A - (1-\alpha)\lambda_C)v)} dv \\ &= \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} (1 - e^{-(\lambda_A + \alpha\lambda_C)t}) \\ &\quad - \frac{\lambda_A}{\lambda_A - (1-\alpha)\lambda_C} (e^{-\lambda_C t} - e^{-(\lambda_C t + (\lambda_A - (1-\alpha)\lambda_C)t)}) \\ &= \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} (1 - e^{-(\lambda_A + \alpha\lambda_C)t}) - \frac{\lambda_A}{\lambda_A - (1-\alpha)\lambda_C} (e^{-\lambda_C t} - e^{-(\lambda_A + \alpha\lambda_C)t}) \\ &= \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} - \frac{\lambda_A}{\lambda_A - (1-\alpha)\lambda_C} e^{-\lambda_C t} \\ &\quad + \frac{\lambda_A \lambda_C}{(\lambda_A - (1-\alpha)\lambda_C)(\lambda_A + \alpha\lambda_C)} e^{-(\lambda_A + \alpha\lambda_C)t} \end{aligned}$$

$$\begin{aligned} Pr \{C_a \cdot (A \triangleleft C_a) \cdot \mathcal{B}\} (t) &= (1 - F_B(t)) \int_0^t \left(\int_v^t f_{C_a}(u, v) du \right) f_A(v) dv \\ &= e^{-\lambda_B t} \int_0^t \left(\int_v^t f_{C_a}(u, v) du \right) f_A(v) dv \\ &= \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} e^{-\lambda_B t} - \frac{\lambda_A}{\lambda_A - (1-\alpha)\lambda_C} e^{-(\lambda_B + \lambda_C)t} \\ &\quad + \frac{\lambda_A \lambda_C}{(\lambda_A - (1-\alpha)\lambda_C)(\lambda_A + \alpha\lambda_C)} e^{-(\lambda_A + \lambda_B + \alpha\lambda_C)t} \\ &= P_6(t) \end{aligned}$$

- Validation of the expression for $Pr \{B \cdot (C_d \triangleleft A) \cdot (A \triangleleft B)\} (t)$

$$Pr \{B \cdot (C_d \triangleleft A) \cdot (A \triangleleft B)\} (t) = \int_0^t \left(\int_0^u f_A(v) F_{C_d}(v) dv \right) f_B(u) du$$

$$\begin{aligned}
 \int_0^u f_A(v)F_{C_d}(v)dv &= \int_0^u \lambda_A e^{-\lambda_A v} (1 - e^{-\alpha\lambda_C v}) dv \\
 &= \int_0^u \lambda_A e^{-\lambda_A v} dv - \int_0^u \lambda_A e^{-(\lambda_A + \alpha\lambda_C)v} dv \\
 &= 1 - e^{-\lambda_A u} - \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} (1 - e^{-(\lambda_A + \alpha\lambda_C)u}) \\
 &= \frac{\alpha\lambda_C}{\lambda_A + \alpha\lambda_C} - e^{-\lambda_A u} + \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} e^{-(\lambda_A + \alpha\lambda_C)u}
 \end{aligned}$$

$$\begin{aligned}
 &Pr \{B \cdot (C_d \triangleleft A) \cdot (A \triangleleft B)\} (t) \\
 &= \int_0^t \left(\int_0^u f_A(v)F_{C_d}(v)dv \right) f_B(u) du \\
 &= \frac{\alpha\lambda_C}{\lambda_A + \alpha\lambda_C} \int_0^t \lambda_B e^{-\lambda_B u} du - \int_0^t e^{-\lambda_A u} \lambda_B e^{-\lambda_B u} du \\
 &\quad + \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} \int_0^t e^{-(\lambda_A + \alpha\lambda_C)u} \lambda_B e^{-\lambda_B u} du \\
 &= \frac{\alpha\lambda_C}{\lambda_A + \alpha\lambda_C} \int_0^t \lambda_B e^{-\lambda_B u} du - \lambda_B \int_0^t e^{-(\lambda_A + \lambda_B)u} du \\
 &\quad + \frac{\lambda_A \lambda_B}{\lambda_A + \alpha\lambda_C} \int_0^t e^{-(\lambda_A + \lambda_B + \alpha\lambda_C)u} du \\
 &= \frac{\alpha\lambda_C}{\lambda_A + \alpha\lambda_C} (1 - e^{-\lambda_B t}) - \frac{\lambda_B}{\lambda_A + \lambda_B} (1 - e^{-(\lambda_A + \lambda_B)t}) \\
 &\quad + \frac{\lambda_A \lambda_B}{(\lambda_A + \alpha\lambda_C)(\lambda_A + \lambda_B + \alpha\lambda_C)} (1 - e^{-(\lambda_A + \lambda_B + \alpha\lambda_C)t}) \\
 &= \frac{\alpha\lambda_A \lambda_C}{(\lambda_A + \lambda_B)(\lambda_A + \lambda_B + \alpha\lambda_C)} - \frac{\alpha\lambda_C}{\lambda_A + \alpha\lambda_C} e^{-\lambda_B t} \\
 &\quad + \frac{\lambda_B}{\lambda_A + \lambda_B} e^{-(\lambda_A + \lambda_B)t} - \frac{\lambda_A \lambda_B e^{-(\lambda_A + \lambda_B + \alpha\lambda_C)t}}{(\lambda_A + \alpha\lambda_C)(\lambda_A + \lambda_B + \alpha\lambda_C)} \\
 &= P_{15}(t)
 \end{aligned}$$

- Validation of the expression for $Pr \{A \cdot (C_d \triangleleft A) \cdot \mathcal{B}\} (t)$

$$Pr \{A \cdot (C_d \triangleleft A) \cdot \mathcal{B}\} (t) = (1 - F_B(t)) \int_0^t f_A(u)F_{C_d}(u)du$$

$$\begin{aligned}
\int_0^t f_A(u)F_{C_d}(u)du &= \int_0^t \lambda_A e^{-\lambda_A u} (1 - e^{-\alpha\lambda_C u}) du \\
&= \int_0^t \lambda_A e^{-\lambda_A u} du - \int_0^t \lambda_A e^{-(\lambda_A + \alpha\lambda_C)u} du \\
&= 1 - e^{-\lambda_A t} - \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} (1 - e^{-(\lambda_A + \alpha\lambda_C)t}) \\
&= \frac{\alpha\lambda_C}{\lambda_A + \alpha\lambda_C} - e^{-\lambda_A t} + \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} e^{-(\lambda_A + \alpha\lambda_C)t}
\end{aligned}$$

$$\begin{aligned}
&Pr \{A \cdot (C_d \triangleleft A) \cdot \mathcal{B}\} (t) \\
&= (1 - F_B(t)) \int_0^t f_A(u)F_{C_d}(u)du \\
&= e^{-\lambda_B t} \int_0^t f_A(u)F_{C_d}(u)du \\
&= \frac{\alpha\lambda_C}{\lambda_A + \alpha\lambda_C} e^{-\lambda_B t} - e^{-(\lambda_A + \lambda_B)t} + \frac{\lambda_A}{\lambda_A + \alpha\lambda_C} e^{-(\lambda_A + \lambda_B + \alpha\lambda_C)t} \\
&= P_9(t)
\end{aligned}$$

- Validation of the expression for $Pr \{A \cdot (B \triangleleft A)\} (t)$

$$\begin{aligned}
Pr \{A \cdot (B \triangleleft A)\} (t) &= \int_0^t f_A(u)F_B(u)du \\
&= \int_0^t \lambda_A e^{-\lambda_A u} (1 - e^{-\lambda_B u}) du \\
&= \lambda_A \int_0^t e^{-\lambda_A u} du - \lambda_A \int_0^t e^{-(\lambda_A + \lambda_B)u} du \\
&= 1 - e^{-\lambda_A t} - \frac{\lambda_A}{\lambda_A + \lambda_B} (1 - e^{-(\lambda_A + \lambda_B)t}) \\
&= \frac{\lambda_B}{\lambda_A + \lambda_B} - e^{-\lambda_A t} + \frac{\lambda_A}{\lambda_A + \lambda_B} e^{-(\lambda_A + \lambda_B)t} \\
&= P_7(t) + P_{13}(t) + P_{14}(t) + P_{16}(t)
\end{aligned}$$

The expressions obtained for the probability of occurrence of the 6 disjoint terms of the behavioural model of Section 5.4.2.2 are the same as the expressions obtained with Markov Chains. The failure probability of $Q1$, and $Q2$ (by symmetry), can be determined as

$$\begin{aligned}
Pr \{Q1\} (t) &= 1 - e^{-\lambda_A t} - \frac{\lambda_A}{\lambda_A + \lambda_B - (1 - \alpha)\lambda_C} e^{-\lambda_C t} \\
&\quad + \frac{\lambda_A}{\lambda_A + \lambda_B - (1 - \alpha)\lambda_C} e^{-(\lambda_A + \lambda_B + \alpha\lambda_C)t} \\
Pr \{Q2\} (t) &= 1 - e^{-\lambda_B t} - \frac{\lambda_B}{\lambda_A + \lambda_B - (1 - \alpha)\lambda_C} e^{-\lambda_C t} \\
&\quad + \frac{\lambda_B}{\lambda_A + \lambda_B - (1 - \alpha)\lambda_C} e^{-(\lambda_A + \lambda_B + \alpha\lambda_C)t}
\end{aligned}$$

and these failure probabilities are the same as those obtained with Markov Chains.

B.3 Case of a single Spare gate with 3 input events

B.3.1 Result obtained with Markov Chains

The state transition diagram of a single Spare gate with 3 input events is shown in Fig. B.3. States 11 to 16 represent the failure of the Spare gate.

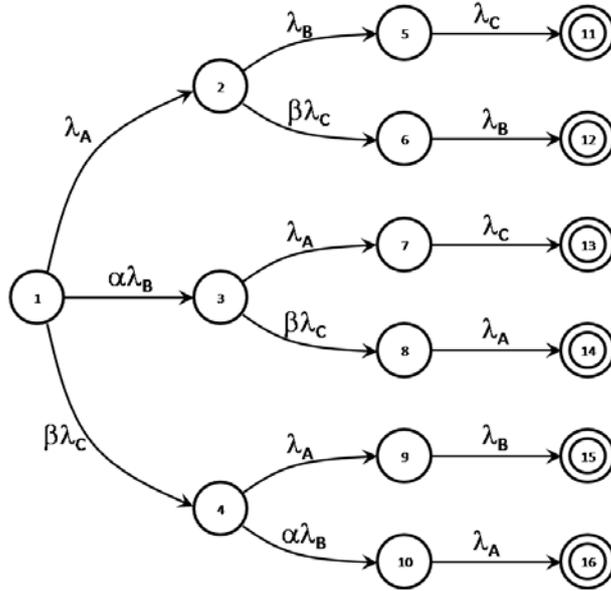


Figure B.3: State transition diagram of a single Spare gate with 3 input events

Solving the differential system which corresponds to the state transition diagram in Fig. B.3 allows to determine the failure probability of the Spare gate:

$$\begin{aligned}
 Pr \{Q\} (t) &= P_{11}(t) + P_{12}(t) + P_{13}(t) + P_{14}(t) + P_{15}(t) + P_{16}(t) \\
 &= 1 - e^{-\lambda_A t} - \frac{\lambda_A}{\lambda_A - (1 - \alpha)\lambda_B} e^{-\lambda_B t} \\
 &\quad - \frac{\lambda_A \lambda_B (\lambda_A + \alpha \lambda_B - (1 + \alpha)(1 - \beta)\lambda_C)}{(\lambda_A - (1 - \beta)\lambda_C)(\lambda_B - (1 - \beta)\lambda_C)(\lambda_A + \alpha \lambda_B - (1 - \beta)\lambda_C)} e^{-\lambda_C t} \\
 &\quad + \frac{\lambda_A}{\lambda_A - (1 - \alpha)\lambda_B} e^{-(\lambda_A + \alpha \lambda_B)t} + \frac{\lambda_A}{\lambda_A - (1 - \beta)\lambda_C} e^{-(\lambda_A + \beta \lambda_C)t} \quad (B.2) \\
 &\quad + \frac{\lambda_A \lambda_B}{(\lambda_A - (1 - \alpha)\lambda_B)(\lambda_B - (1 - \beta)\lambda_C)} e^{-(\lambda_B + \beta \lambda_C)t} \\
 &\quad - \frac{\lambda_A (\lambda_A + \alpha \lambda_B)}{(\lambda_A - (1 - \alpha)\lambda_B)(\lambda_A + \alpha \lambda_B - (1 - \beta)\lambda_C)} e^{-(\lambda_A + \alpha \lambda_B + \beta \lambda_C)t}
 \end{aligned}$$

B.3.2 Result obtained with our probabilistic model

According to Section 5.4.3, the probabilistic model of a single Spare gate with 3 input events is

$$\begin{aligned}
 Pr\{Q\}(t) = & \int_0^t \left(\int_u^t \left(\int_v^t f_{C_a}(w, v) dw \right) f_{B_a}(v, u) dv \right) f_A(u) du \\
 & + \int_0^t \left(\int_w^t \left(\int_v^u f_{C_d}(v) dv \right) f_{B_a}(u, w) du \right) f_A(w) dw \\
 & + \int_0^t \left(\int_u^t f_{C_a}(v, u) dv \right) f_A(u) F_{B_d}(u) du \\
 & + \int_0^t \left(\int_0^u f_{C_d}(v) F_{B_d}(v) dv \right) f_A(u) du \\
 & + \int_0^t \left(\int_u^t f_{B_a}(v, u) dv \right) f_A(u) F_{C_d}(u) du \\
 & + \int_0^t \left(\int_0^u f_{B_d}(v) F_{C_d}(v) dv \right) f_A(u) du
 \end{aligned}$$

In the case of exponential distributions,

$$\begin{cases}
 F_A(t) = 1 - e^{-\lambda_A t}, \forall t \geq 0 \\
 F_{B_d}(t) = 1 - e^{-\alpha \lambda_B t}, \forall t \geq 0 \\
 F_{B_a}(t, t_A) = 1 - e^{-\lambda_B (t - (1-\alpha)t_A)}, \forall t \geq (1-\alpha)t_A \\
 F_{C_d}(t) = 1 - e^{-\beta \lambda_C t}, \forall t \geq 0 \\
 F_{C_a}(t, v) = 1 - e^{-\lambda_C (t - (1-\beta)v)}, \forall t \geq (1-\beta)v
 \end{cases}$$

where v is the failure date of the event on which C depends. Indeed, C becomes active as soon as A fails if B has failed before A , or as soon as B fails, so the failure distribution of C depends on t_A or t_{B_a} .

- Validation of the expression for $Pr \{C_a \cdot (A \triangleleft B_a) \cdot (B_a \triangleleft C_a)\} (t)$

$$Pr \{C_a \cdot (A \triangleleft B_a) \cdot (B_a \triangleleft C_a)\} (t) = \int_0^t \left(\int_u^t \left(\int_v^t f_{C_a}(w, v) dw \right) f_{B_a}(v, u) dv \right) f_A(u) du$$

$$\int_v^t f_{C_a}(w, v) dw = \int_v^t \lambda_C e^{-\lambda_C(w-(1-\beta)v)} dw = e^{-\beta\lambda_C v} - e^{-\lambda_C(t-(1-\beta)v)}$$

$$\begin{aligned} & \int_u^t \left(\int_v^t f_{C_a}(w, v) dw \right) f_{B_a}(v, u) dv \\ &= \lambda_B \int_u^t \left(e^{-\beta\lambda_C v} - e^{-\lambda_C(t-(1-\beta)v)} \right) e^{-\lambda_B(v-(1-\alpha)u)} dv \\ &= \lambda_B \int_u^t e^{-((\lambda_B+\beta\lambda_C)v-(1-\alpha)\lambda_B u)} dv - \lambda_B \int_u^t e^{-(\lambda_C t+(\lambda_B-(1-\beta)\lambda_C)v-(1-\alpha)\lambda_B u)} dv \\ &= \frac{\lambda_B}{\lambda_B + \beta\lambda_C} \left(e^{-(\alpha\lambda_B+\beta\lambda_C)u} - e^{-((\lambda_B+\beta\lambda_C)t-(1-\alpha)\lambda_B u)} \right) \\ &\quad - \frac{\lambda_B}{\lambda_B - (1-\beta)\lambda_C} \left(e^{-(\lambda_C t+(\alpha\lambda_B-(1-\beta)\lambda_C)u)} - e^{-((\lambda_B+\beta\lambda_C)t-(1-\alpha)\lambda_B u)} \right) \\ &= \frac{\lambda_B}{\lambda_B + \beta\lambda_C} e^{-(\alpha\lambda_B+\beta\lambda_C)u} - \frac{\lambda_B}{\lambda_B - (1-\beta)\lambda_C} e^{-(\lambda_C t+(\alpha\lambda_B-(1-\beta)\lambda_C)u)} \\ &\quad + \frac{\lambda_B \lambda_C}{(\lambda_B - (1-\beta)\lambda_C)(\lambda_B + \beta\lambda_C)} e^{-((\lambda_B+\beta\lambda_C)t-(1-\alpha)\lambda_B u)} \end{aligned}$$

$$\begin{aligned} & Pr \{C_a \cdot (A \triangleleft B_a) \cdot (B_a \triangleleft C_a)\} (t) \\ &= \int_0^t \left(\int_u^t \left(\int_v^t f_{C_a}(w, v) dw \right) f_{B_a}(v, u) dv \right) f_A(u) du \\ &= \frac{\lambda_A \lambda_B}{\lambda_B + \beta\lambda_C} \int_0^t e^{-(\lambda_A+\alpha\lambda_B+\beta\lambda_C)u} du \\ &\quad - \frac{\lambda_A \lambda_B}{\lambda_B - (1-\beta)\lambda_C} \int_0^t e^{-(\lambda_C t+(\lambda_A+\alpha\lambda_B-(1-\beta)\lambda_C)u)} du \\ &\quad + \frac{\lambda_A \lambda_B \lambda_C}{(\lambda_B - (1-\beta)\lambda_C)(\lambda_B + \beta\lambda_C)} \int_0^t e^{-((\lambda_B+\beta\lambda_C)t+(\lambda_A-(1-\alpha)\lambda_B)u)} du \\ &= \frac{\lambda_A \lambda_B}{(\lambda_B + \beta\lambda_C)(\lambda_A + \alpha\lambda_B + \beta\lambda_C)} (1 - e^{-(\lambda_A+\alpha\lambda_B+\beta\lambda_C)t}) \\ &\quad - \frac{\lambda_A \lambda_B (e^{-\lambda_C t} - e^{-(\lambda_A+\alpha\lambda_B+\beta\lambda_C)t})}{(\lambda_B - (1-\beta)\lambda_C)(\lambda_A + \alpha\lambda_B - (1-\beta)\lambda_C)} \\ &\quad + \frac{\lambda_A \lambda_B \lambda_C (e^{-(\lambda_B+\beta\lambda_C)t} - e^{-(\lambda_A+\alpha\lambda_B+\beta\lambda_C)t})}{(\lambda_A - (1-\alpha)\lambda_B)(\lambda_B - (1-\beta)\lambda_C)(\lambda_B + \beta\lambda_C)} \end{aligned}$$

$$\begin{aligned}
&= \frac{\lambda_A \lambda_B}{(\lambda_B + \beta \lambda_C)(\lambda_A + \alpha \lambda_B + \beta \lambda_C)} \\
&\quad - \frac{\lambda_A \lambda_B}{(\lambda_B - (1 - \beta) \lambda_C)(\lambda_A + \alpha \lambda_B - (1 - \beta) \lambda_C)} e^{-\lambda_C t} \\
&\quad + \frac{\lambda_A \lambda_B \lambda_C}{(\lambda_A - (1 - \alpha) \lambda_B)(\lambda_B - (1 - \beta) \lambda_C)(\lambda_B + \beta \lambda_C)} e^{-(\lambda_B + \beta \lambda_C) t} \\
&\quad - \frac{\lambda_A \lambda_B \lambda_C e^{-(\lambda_A + \alpha \lambda_B + \beta \lambda_C) t}}{(\lambda_A - (1 - \alpha) \lambda_B)(\lambda_A + \alpha \lambda_B - (1 - \beta) \lambda_C)(\lambda_A + \alpha \lambda_B + \beta \lambda_C)} \\
&= P_{11}(t)
\end{aligned}$$

- Validation of the expression for $Pr \{B_a \cdot (A \triangleleft C_d) \cdot (C_d \triangleleft B_a)\} (t)$

$$Pr \{B_a \cdot (A \triangleleft C_d) \cdot (C_d \triangleleft B_a)\} (t) = \int_0^t \left(\int_w^t \left(\int_w^u f_{C_d}(v) dv \right) f_{B_a}(u, w) du \right) f_A(w) dw$$

$$\int_w^u f_{C_d}(v) dv = \int_w^u \beta \lambda_C e^{-\beta \lambda_C v} dv = e^{-\beta \lambda_C w} - e^{-\beta \lambda_C u}$$

$$\begin{aligned}
&\int_w^t \left(\int_w^u f_{C_d}(v) dv \right) f_{B_a}(u, w) du \\
&= \lambda_B \int_w^t \left(e^{-\beta \lambda_C w} - e^{-\beta \lambda_C u} \right) e^{-\lambda_B(u - (1 - \alpha)w)} du \\
&= \lambda_B \int_w^t e^{-(\lambda_B u + (-1 - \alpha) \lambda_B + \beta \lambda_C) w} du - \lambda_B \int_w^t e^{-((\lambda_B + \beta \lambda_C) u - (1 - \alpha) \lambda_B w)} du \\
&= e^{-(\alpha \lambda_B + \beta \lambda_C) w} - e^{-(\lambda_B t + (-1 - \alpha) \lambda_B + \beta \lambda_C) w} \\
&\quad - \frac{\lambda_B}{\lambda_B + \beta \lambda_C} \left(e^{-(\alpha \lambda_B + \beta \lambda_C) w} - e^{-((\lambda_B + \beta \lambda_C) t - (1 - \alpha) \lambda_B w)} \right) \\
&= \frac{\beta \lambda_C}{\lambda_B + \beta \lambda_C} e^{-(\alpha \lambda_B + \beta \lambda_C) w} - e^{-(\lambda_B t + (-1 - \alpha) \lambda_B + \beta \lambda_C) w} \\
&\quad + \frac{\lambda_B}{\lambda_B + \beta \lambda_C} e^{-((\lambda_B + \beta \lambda_C) t - (1 - \alpha) \lambda_B w)}
\end{aligned}$$

$$\begin{aligned}
&Pr \{B_a \cdot (A \triangleleft C_d) \cdot (C_d \triangleleft B_a)\} (t) \\
&= \int_0^t \left(\int_w^t \left(\int_w^u f_{C_d}(v) dv \right) f_{B_a}(u, w) du \right) f_A(w) dw \\
&= \frac{\beta \lambda_A \lambda_C}{\lambda_B + \beta \lambda_C} \int_0^t e^{-(\lambda_A + \alpha \lambda_B + \beta \lambda_C) w} dw - \lambda_A \int_0^t e^{-(\lambda_B t + (\lambda_A - (1 - \alpha) \lambda_B + \beta \lambda_C) w)} dw \\
&\quad + \frac{\lambda_A \lambda_B}{\lambda_B + \beta \lambda_C} \int_0^t e^{-((\lambda_B + \beta \lambda_C) t + (\lambda_A - (1 - \alpha) \lambda_B) w)} dw
\end{aligned}$$

$$\begin{aligned}
&= \frac{\beta\lambda_A\lambda_C}{(\lambda_B + \beta\lambda_C)(\lambda_A + \alpha\lambda_B + \beta\lambda_C)}(1 - e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)t}) \\
&\quad - \frac{\lambda_A}{\lambda_A - (1 - \alpha)\lambda_B + \beta\lambda_C}(e^{-\lambda_B t} - e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)t}) \\
&\quad + \frac{\lambda_A\lambda_B}{(\lambda_A - (1 - \alpha)\lambda_B)(\lambda_B + \beta\lambda_C)}(e^{-(\lambda_B + \beta\lambda_C)t} - e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)t}) \\
&= \frac{\beta\lambda_A\lambda_C}{(\lambda_B + \beta\lambda_C)(\lambda_A + \alpha\lambda_B + \beta\lambda_C)} - \frac{\lambda_A}{\lambda_A - (1 - \alpha)\lambda_B + \beta\lambda_C}e^{-\lambda_B t} \\
&\quad + \frac{\lambda_A\lambda_B}{(\lambda_A - (1 - \alpha)\lambda_B)(\lambda_B + \beta\lambda_C)}e^{-(\lambda_B + \beta\lambda_C)t} \\
&\quad - \frac{\beta\lambda_A\lambda_B\lambda_C e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)t}}{(\lambda_A - (1 - \alpha)\lambda_B)(\lambda_A - (1 - \alpha)\lambda_B + \beta\lambda_C)(\lambda_A + \alpha\lambda_B + \beta\lambda_C)} \\
&= P_{12}(t)
\end{aligned}$$

- Validation of the expression for $Pr\{C_a \cdot (B_d \triangleleft A) \cdot (A \triangleleft C_a)\}(t)$

$$Pr\{C_a \cdot (B_d \triangleleft A) \cdot (A \triangleleft C_a)\}(t) = \int_0^t \left(\int_u^t f_{C_a}(v, u) dv \right) f_A(u) F_{B_d}(u) du$$

$$\int_u^t f_{C_a}(v, u) dv = \int_u^t \lambda_C e^{-\lambda_C(v - (1 - \beta)u)} dv = e^{-\beta\lambda_C u} - e^{-\lambda_C(t - (1 - \beta)u)}$$

$$\begin{aligned}
&Pr\{C_a \cdot (B_d \triangleleft A) \cdot (A \triangleleft C_a)\}(t) \\
&= \int_0^t \left(\int_u^t f_{C_a}(v, u) dv \right) f_A(u) F_{B_d}(u) du \\
&= \lambda_A \int_0^t \left(e^{-\beta\lambda_C u} - e^{-\lambda_C(t - (1 - \beta)u)} \right) e^{-\lambda_A u} (1 - e^{-\alpha\lambda_B u}) du \\
&= \lambda_A \int_0^t \left(e^{-\beta\lambda_C u} - e^{-\lambda_C(t - (1 - \beta)u)} \right) (e^{-\lambda_A u} - e^{-(\lambda_A + \alpha\lambda_B)u}) du \\
&= \lambda_A \int_0^t e^{-\beta\lambda_C u} e^{-\lambda_A u} du - \lambda_A \int_0^t e^{-\beta\lambda_C u} e^{-(\lambda_A + \alpha\lambda_B)u} du \\
&\quad - \lambda_A \int_0^t e^{-\lambda_C(t - (1 - \beta)u)} e^{-\lambda_A u} du + \lambda_A \int_0^t e^{-\lambda_C(t - (1 - \beta)u)} e^{-(\lambda_A + \alpha\lambda_B)u} du \\
&= \lambda_A \int_0^t e^{-(\lambda_A + \beta\lambda_C)u} du - \lambda_A \int_0^t e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)u} du \\
&\quad - \lambda_A \int_0^t e^{-(\lambda_C t + (\lambda_A - (1 - \beta)\lambda_C)u)} du + \lambda_A \int_0^t e^{-(\lambda_C t + (\lambda_A + \alpha\lambda_B - (1 - \beta)\lambda_C)u)} du
\end{aligned}$$

$$\begin{aligned}
&= \frac{\lambda_A}{\lambda_A + \beta\lambda_C}(1 - e^{-(\lambda_A + \beta\lambda_C)t}) - \frac{\lambda_A}{\lambda_A + \alpha\lambda_B + \beta\lambda_C}(1 - e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)t}) \\
&\quad - \frac{\lambda_A}{\lambda_A - (1 - \beta)\lambda_C}(e^{-\lambda_C t} - e^{-(\lambda_A + \beta\lambda_C)t}) \\
&\quad + \frac{\lambda_A}{\lambda_A + \alpha\lambda_B - (1 - \beta)\lambda_C}(e^{-\lambda_C t} - e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)t}) \\
&= \frac{\alpha\lambda_A\lambda_B}{(\lambda_A + \beta\lambda_C)(\lambda_A + \alpha\lambda_B + \beta\lambda_C)} \\
&\quad - \frac{\alpha\lambda_A\lambda_B}{(\lambda_A - (1 - \beta)\lambda_C)(\lambda_A + \alpha\lambda_B - (1 - \beta)\lambda_C)}e^{-\lambda_C t} \\
&\quad + \frac{\lambda_A\lambda_C}{(\lambda_A - (1 - \beta)\lambda_C)(\lambda_A + \beta\lambda_C)}e^{-(\lambda_A + \beta\lambda_C)t} \\
&\quad - \frac{\lambda_A\lambda_C e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)t}}{(\lambda_A + \alpha\lambda_B - (1 - \beta)\lambda_C)(\lambda_A + \alpha\lambda_B + \beta\lambda_C)} \\
&= P_{13}(t)
\end{aligned}$$

- Validation of the expression for $Pr \{A \cdot (B_d \triangleleft C_d) \cdot (C_d \triangleleft A)\} (t)$

$$Pr \{A \cdot (B_d \triangleleft C_d) \cdot (C_d \triangleleft A)\} (t) = \int_0^t \left(\int_0^u f_{C_d}(v) F_{B_d}(v) dv \right) f_A(u) du$$

$$\begin{aligned}
\int_0^u f_{C_d}(v) F_{B_d}(v) dv &= \beta\lambda_C \int_0^u e^{-\beta\lambda_C v} (1 - e^{-\alpha\lambda_B v}) dv \\
&= \beta\lambda_C \int_0^u e^{-\beta\lambda_C v} dv - \beta\lambda_C \int_0^u e^{-(\alpha\lambda_B + \beta\lambda_C)v} dv \\
&= 1 - e^{-\beta\lambda_C u} - \frac{\beta\lambda_C}{\alpha\lambda_B + \beta\lambda_C} (1 - e^{-(\alpha\lambda_B + \beta\lambda_C)u}) \\
&= \frac{\alpha\lambda_B}{\alpha\lambda_B + \beta\lambda_C} - e^{-\beta\lambda_C u} + \frac{\beta\lambda_C}{\alpha\lambda_B + \beta\lambda_C} e^{-(\alpha\lambda_B + \beta\lambda_C)u}
\end{aligned}$$

$$\begin{aligned}
&Pr \{A \cdot (B_d \triangleleft C_d) \cdot (C_d \triangleleft A)\} (t) \\
&= \int_0^t \left(\int_0^u f_{C_d}(v) F_{B_d}(v) dv \right) f_A(u) du \\
&= \frac{\alpha\lambda_B}{\alpha\lambda_B + \beta\lambda_C} \int_0^t \lambda_A e^{-\lambda_A u} du - \lambda_A \int_0^t e^{-(\lambda_A + \beta\lambda_C)u} du \\
&\quad + \frac{\beta\lambda_A\lambda_C}{\alpha\lambda_B + \beta\lambda_C} \int_0^t e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)u} du \\
&= \frac{\alpha\lambda_B}{\alpha\lambda_B + \beta\lambda_C} (1 - e^{-\lambda_A t}) - \frac{\lambda_A}{\lambda_A + \beta\lambda_C} (1 - e^{-(\lambda_A + \beta\lambda_C)t}) \\
&\quad + \frac{\beta\lambda_A\lambda_C}{(\alpha\lambda_B + \beta\lambda_C)(\lambda_A + \alpha\lambda_B + \beta\lambda_C)} (1 - e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)t})
\end{aligned}$$

$$\begin{aligned}
&= \frac{\alpha\beta\lambda_B\lambda_C}{(\lambda_A + \beta\lambda_C)(\lambda_A + \alpha\lambda_B + \beta\lambda_C)} - \frac{\alpha\lambda_B}{\alpha\lambda_B + \beta\lambda_C} e^{-\lambda_A t} \\
&+ \frac{\lambda_A}{\lambda_A + \beta\lambda_C} e^{-(\lambda_A + \beta\lambda_C)t} - \frac{\beta\lambda_A\lambda_C e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)t}}{(\alpha\lambda_B + \beta\lambda_C)(\lambda_A + \alpha\lambda_B + \beta\lambda_C)} \\
&= P_{14}(t)
\end{aligned}$$

- Validation of the expression for $Pr\{B_a \cdot (C_d \triangleleft A) \cdot (A \triangleleft B_a)\}(t)$

$$Pr\{B_a \cdot (C_d \triangleleft A) \cdot (A \triangleleft B_a)\}(t) = \int_0^t \left(\int_u^t f_{B_a}(v, u) dv \right) f_A(u) F_{C_d}(u) du$$

$$\int_u^t f_{B_a}(v, u) dv = \int_u^t \lambda_B e^{-\lambda_B(v - (1-\alpha)u)} dv = e^{-\alpha\lambda_B u} - e^{-\lambda_B(t - (1-\alpha)u)}$$

$$\begin{aligned}
&Pr\{B_a \cdot (C_d \triangleleft A) \cdot (A \triangleleft B_a)\}(t) \\
&= \int_0^t \left(\int_u^t f_{B_a}(v, u) dv \right) f_A(u) F_{C_d}(u) du \\
&= \lambda_A \int_0^t \left(e^{-\alpha\lambda_B u} - e^{-\lambda_B(t - (1-\alpha)u)} \right) e^{-\lambda_A u} (1 - e^{-\beta\lambda_C u}) du \\
&= \lambda_A \int_0^t \left(e^{-\alpha\lambda_B u} - e^{-\lambda_B(t - (1-\alpha)u)} \right) (e^{-\lambda_A u} - e^{-(\lambda_A + \beta\lambda_C)u}) du \\
&= \lambda_A \int_0^t e^{-(\lambda_A + \alpha\lambda_B)u} du - \lambda_A \int_0^t e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)u} du \\
&\quad - \lambda_A \int_0^t e^{-(\lambda_B t + (\lambda_A - (1-\alpha)\lambda_B)u)} du \\
&\quad + \lambda_A \int_0^t e^{-(\lambda_B t + (\lambda_A - (1-\alpha)\lambda_B + \beta\lambda_C)u)} du \\
&= \frac{\lambda_A}{\lambda_A + \alpha\lambda_B} (1 - e^{-(\lambda_A + \alpha\lambda_B)t}) \\
&\quad - \frac{\lambda_A}{\lambda_A + \alpha\lambda_B + \beta\lambda_C} (1 - e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)t}) \\
&\quad - \frac{\lambda_A}{\lambda_A - (1-\alpha)\lambda_B} (e^{-\lambda_B t} - e^{-(\lambda_A + \alpha\lambda_B)t}) \\
&\quad + \frac{\lambda_A}{\lambda_A - (1-\alpha)\lambda_B + \beta\lambda_C} (e^{-\lambda_B t} - e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)t}) \\
&= \frac{\beta\lambda_A\lambda_C}{(\lambda_A + \alpha\lambda_B)(\lambda_A + \alpha\lambda_B + \beta\lambda_C)} \\
&\quad - \frac{\beta\lambda_A\lambda_C}{(\lambda_A - (1-\alpha)\lambda_B)(\lambda_A - (1-\alpha)\lambda_B + \beta\lambda_C)} e^{-\lambda_B t} \\
&\quad + \frac{\lambda_A\lambda_B}{(\lambda_A - (1-\alpha)\lambda_B)(\lambda_A + \alpha\lambda_B)} e^{-(\lambda_A + \alpha\lambda_B)t} \\
&\quad - \frac{\lambda_A\lambda_B e^{-(\lambda_A + \alpha\lambda_B + \beta\lambda_C)t}}{(\lambda_A - (1-\alpha)\lambda_B + \beta\lambda_C)(\lambda_A + \alpha\lambda_B + \beta\lambda_C)} \\
&= P_{15}(t)
\end{aligned}$$

- Validation of the expression for $Pr \{A \cdot (C_d \triangleleft B_d) \cdot (B_d \triangleleft A)\} (t)$

$$Pr \{A \cdot (C_d \triangleleft B_d) \cdot (B_d \triangleleft A)\} (t) = \int_0^t \left(\int_0^u f_{B_d}(v) F_{C_d}(v) dv \right) f_A(u) du$$

$$\begin{aligned} \int_0^u f_{B_d}(v) F_{C_d}(v) dv &= \alpha \lambda_B \int_0^u e^{-\alpha \lambda_B v} (1 - e^{-\beta \lambda_C v}) dv \\ &= \alpha \lambda_B \int_0^u e^{-\alpha \lambda_B v} dv - \alpha \lambda_B \int_0^u e^{-(\alpha \lambda_B + \beta \lambda_C) v} dv \\ &= 1 - e^{-\alpha \lambda_B u} - \frac{\alpha \lambda_B}{\alpha \lambda_B + \beta \lambda_C} (1 - e^{-(\alpha \lambda_B + \beta \lambda_C) u}) \\ &= \frac{\beta \lambda_C}{\alpha \lambda_B + \beta \lambda_C} - e^{-\alpha \lambda_B u} + \frac{\alpha \lambda_B}{\alpha \lambda_B + \beta \lambda_C} e^{-(\alpha \lambda_B + \beta \lambda_C) u} \end{aligned}$$

$$\begin{aligned} &Pr \{A \cdot (C_d \triangleleft B_d) \cdot (B_d \triangleleft A)\} (t) \\ &= \int_0^t \left(\int_0^u f_{B_d}(v) F_{C_d}(v) dv \right) f_A(u) du \\ &= \frac{\beta \lambda_C}{\alpha \lambda_B + \beta \lambda_C} \int_0^t \lambda_A e^{-\lambda_A u} du - \lambda_A \int_0^t e^{-(\lambda_A + \alpha \lambda_B) u} du \\ &\quad + \frac{\alpha \lambda_A \lambda_B}{\alpha \lambda_B + \beta \lambda_C} \int_0^t e^{-(\lambda_A + \alpha \lambda_B + \beta \lambda_C) u} du \\ &= \frac{\beta \lambda_C}{\alpha \lambda_B + \beta \lambda_C} (1 - e^{-\lambda_A t}) - \frac{\lambda_A}{\lambda_A + \alpha \lambda_B} (1 - e^{-(\lambda_A + \alpha \lambda_B) t}) \\ &\quad + \frac{\alpha \lambda_A \lambda_B}{(\alpha \lambda_B + \beta \lambda_C)(\lambda_A + \alpha \lambda_B + \beta \lambda_C)} (1 - e^{-(\lambda_A + \alpha \lambda_B + \beta \lambda_C) t}) \\ &= \frac{\alpha \beta \lambda_B \lambda_C}{(\lambda_A + \alpha \lambda_B)(\lambda_A + \alpha \lambda_B + \beta \lambda_C)} - \frac{\beta \lambda_C}{\alpha \lambda_B + \beta \lambda_C} e^{-\lambda_A t} \\ &\quad + \frac{\lambda_A}{\lambda_A + \alpha \lambda_B} e^{-(\lambda_A + \alpha \lambda_B) t} - \frac{\alpha \lambda_A \lambda_B e^{-(\lambda_A + \alpha \lambda_B + \beta \lambda_C) t}}{(\alpha \lambda_B + \beta \lambda_C)(\lambda_A + \alpha \lambda_B + \beta \lambda_C)} \\ &= P_{16}(t) \end{aligned}$$

The expressions obtained for the probability of occurrence of the 6 disjoint terms of the behavioural model of Section 5.4.3 are the same as the expressions obtained with Markov Chains, as well as the failure probability of the Spare gate.

Résumé – Dans le contexte de la sûreté de fonctionnement des systèmes critiques, nous nous intéressons aux analyses par arbres de défaillance dynamiques (AdDD). Notre contribution est la définition d'un cadre algébrique permettant de déterminer la fonction de structure des AdDD et d'étendre les méthodes analytiques communément utilisées pour analyser les arbres statiques aux arbres dynamiques. Dans un premier temps, nous passons en revue les principales approches utilisées pour analyser les arbres de défaillance dynamiques, ainsi que leurs limites respectives. Le cadre algébrique permettant la modélisation des AdDD est ensuite présenté. Ce cadre algébrique est fondé sur un modèle temporel des événements et sur la définition de trois opérateurs temporels permettant de traduire la séquentialité d'apparition des événements. Ces opérateurs temporels permettent de définir algébriquement le comportement des portes dynamiques, et donc la fonction de structure des AdDD. Un modèle probabiliste de ces portes dynamiques est ensuite donné afin de pouvoir déterminer la probabilité de défaillance de l'événement sommet des arbres à partir de cette fonction de structure. Nous montrons enfin comment la fonction de structure des AdDD peut être ramenée à une forme canonique grâce à des théorèmes de réécriture, puis à une forme minimale grâce à la définition d'un critère de minimisation, et comment les AdDD peuvent être analysés de manière analytique et directe à partir de cette forme canonique minimale de la fonction de structure. Nous illustrons cette approche avec deux exemples d'AdDD issus de la littérature.

Mots clés – Approche algébrique, modèle comportemental, modèle probabiliste, fonction de structure, arbres de défaillance dynamiques, analyse qualitative, ensembles de séquences de coupe, analyse quantitative.

Abstract – In the context of the reliability of critical systems, we focus on Dynamic Fault Tree (DFT) analysis. Our contribution is the definition of an algebraic framework allowing to determine the structure function of DFTs and to extend the analytical methods commonly used to analyze Static Fault Trees to DFTs. First, we review the main approaches which allow to analyze DFTs, as well as their limits. Then, the algebraic framework allowing the modelling of DFTs is presented. This algebraic framework is based on a temporal model of events, and on the definition of three temporal operators allowing to model the sequences of appearance of events. These temporal operators allow to algebraically define the behaviour of dynamic gates, and hence the structure function of DFTs. A probabilistic model of these dynamic gates is given to determine the failure probability of the top event of DFTs from this structure function. Finally, we show how the structure function of DFTs can be simplified to a canonical form thanks to some theorems and to a minimal form thanks to the definition of a minimization criterion. Last, we show how DFTs can be analyzed analytically and directly from this minimal canonical form of the structure function. We illustrate this approach on two DFT examples from the literature.

Keywords – Algebraic approach, behavioural model, probabilistic model, structure function, dynamic fault trees, qualitative analysis, cut sequence sets, quantitative analysis.