Aalto University
School of Science

Tero Tyrväinen

Risk Importance Measures and Common Cause Failures in Dynamic Flowgraph Methodology

Master´s thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology in the Degree Programme in Engineering Physics and Mathematics.

Espoo, 24.11.2011

Supervisor: Professor Ahti Salo

Instructor: M.Sc. Kim Björkman, D.Sc. Jan-Erik Holmberg

Aalto University School of Science	ABSTRACT OF THE MASTER'S THESIS
Author: Tero Tyrväinen	
Title: Risk Importance Measures and Common Cause Failures in Dynamic Flowgraph Methodology	
Title in Finnish: Riskitärkeysmitat ja yhteisviat dynaamisessa vuokaaviomallinnuksessa	
Degree Programme: Degree Programme in Engineering Physics and Mathematics	
Major subject: Systems and Operations Research	Minor subject: Mathematics
Chair (code): Mat-2	
Supervisor: Prof. Ahti Salo	Instructor: M.Sc. Kim Björkman D.Sc. Jan-Erik Holmberg

Abstract:

Traditionally, fault tree analysis has been the leading method for reliability analysis of complex systems. However, dynamic properties of systems cannot always be described with adequate accuracy using fault trees. Dynamic reliability analysis has been studied widely since the 1990s. Some dynamic calculation tools have been developed but they cannot compete with fault tree analysis tools yet in reliability analysis of nuclear power plants.

Dynamic flowgraph modelling (DFM) is an approach for reliability analysis of dynamic systems. DFM models are directed graphs whose nodes can contain a finite number of states. A system's dynamics is described by discrete state transitions. As in the fault tree analysis, the essential goal of dynamic flowgraph modelling is to identify root causes that lead to a system's failure.

VTT has been developing a DFM-based reliability analysis tool called YADRAT since 2009. DFM models have been analysed previously by transforming them into sets of timed fault trees from which the root causes of the system's failure have been identified. In YADRAT, the model that describes a system is transformed into a binary decision diagram.

Risk importance measures and common cause failures are a significant part of reliability theory and fault tree analysis but they have not been studied much in relation to dynamic flowgraph modelling. Risk importance measures are used to measure how important different components are with regard to the system's reliability. In this thesis, dynamic risk importance measures based on two traditional risk importance measures are formulated so that they take the multi-valued and dynamic logic of DFM models into account. Dynamic risk importance measures can be calculated separately for different failure states of components so that they provide more detailed information on how different components contribute to the system's failure. In addition, dynamic generalisations are developed for traditional parametric common cause failure models. In these common cause failure models, the possibility that failure events can occur at different time points is considered. Dynamic risk importance measures and common cause failure models are implemented in YADRAT.

Date: 24.11.2011	Language: English	Number of pages: 87+12

Keywords: reliability analysis, dynamic flowgraph modelling/methodology, risk importance measure, common cause failure, failure event, component, fault tree, state of a component, prime implicant

Aalto-yliopisto Perustieteiden korkeakoulu	DIPLOMITYÖN TIIVISTELMÄ
Tekijä: Tero Tyrväinen	
Työn nimi: Riskitärkeysmitat ja yhteisviat dynaamisessa vuokaaviomallinnuksessa	
Title in English: Risk Importance Measures and Common Cause Failures in Dynamic Flowgraph Methodology	
Tutkinto-ohjelma: Teknillisen fysiikan ja matematiikan tutkinto-ohjelma	
Pääaine: Systeemi- ja operaatiotutkimus	Sivuaine: Matematiikka
Opetusyksikön (ent. professuuri) koodi: Mat-2	
Työn valvoja: Prof. Ahti Salo	Työn ohjaaja(t): DI Kim Björkman TkT Jan-Erik Holmberg

Tiivistelmä:

Vikapuuanalyysi on perinteisesti ollut johtava menetelmä monimutkaisten järjestelmien luotettavuusanalyysissä. Vikapuilla ei kuitenkaan aina pystytä kuvaamaan järjestelmien dynaamisia ominaisuuksia riittävän tarkasti. Dynaamista luotettavuusanalyysiä on tutkittu laajasti 1990-luvulta lähtien. Dynaamisia laskentatyökaluja on kehitetty, mutta ne eivät vielä pysty kilpailemaan vikapuuanalyysityökalujen kanssa ydinvoimaloiden luotettavuusanalyysissä.

Dynaaminen vuokaaviomallintaminen (dynamic flowgraph modelling, DFM) on menetelmä dynaamisten järjestelmien luotettavuuden analysointiin. DFM-mallit ovat suunnattuja graafeja, joiden solmuilla on äärellinen määrä tiloja. Systeemin dynamiikka kuvataan diskreetteinä tilasiirtyminä. Dynaamisen vuokaaviomallinnuksen, kuten vikapuuanalyysinkin, keskeisenä tavoitteena on selvittää järjestelmävikaan johtavat perimmäiset syyt.

VTT on kehittänyt YADRAT-nimistä DFM-pohjaista luotettavuusanalyysityökalua vuodesta 2009. DFM-malleja on aiemmin analysoitu muodostamalla niistä vikapuita, joista järjestelmävikaan johtavat syyt on voitu tunnistaa. YADRAT:ssa systeemiä kuvaavasta mallista muodostetaan sen sijaan binäärinen päätöspuu.

Riskitärkeysmitat ja yhteisviat ovat merkittävä osa luotettavuusteoriaa ja vikapuuanalyysiä, mutta niitä ei ole ennen juurikaan tutkittu dynaamisen vuokaaviomallinnuksen yhteydessä. Riskitärkeysmitat mittaavat, kuinka tärkeitä eri komponentit ovat järjestelmän luotettavuuden kannalta. Tässä diplomityössä on muodostettu kahteen perinteiseen riskitärkeysmittaan perustuvat dynaamiset riskitärkeysmitat, jotka ottavat huomioon DFM-mallien monitilaisen logiikan ja dynaamisen luonteen. Dynaamiset riskitärkeysmitat voidaan laskea erikseen komponenttien eri vikatiloille. Näin saadaan tarkempaa tietoa eri komponenttien roolista järjestelmävian synnyssä. Lisäksi työssä on kehitetty dynaamisia yleistyksiä perinteisille parametrisille yhteisvikamalleille. Näissä yhteisvikamalleissa huomioidaan mahdollisuus, että vikatapahtumat voivat tapahtua eri ajanhetkinä yhteisen syyn seurauksena. Dynaamiset riskitärkeysmitat ja yhteisvikamallit on toteutettu YADRAT:ssa.

Päivämäärä: 24.11.2011	Kieli: englanti	Sivumäärä: 87+12

Avainsanat: luotettavuusanalyysi, dynaaminen vuokaaviomallintaminen, riskitärkeysmitta, yhteisvika, vikatapahtuma, komponentti, vikapuu, komponentin tila, prime implicantti

Acknowledgments

This thesis was done in VTT Technical Research Center of Finland. The research of the thesis was part of SARANA project in SAFIR2014 research programme. I want to thank my instructor Jan-Erik Holmberg and my supervisor Kaisa Simola for this job opportunity. Thanks to my instructors Jan-Erik and Kim Björkman for guidance and several valuable advices with regard to my thesis. I also want to thank Professor Ahti Salo for supervising my thesis. In addition, thanks to my parents for support.

Espoo, November 14, 2011

Tero Tyrväinen

Contents

Chapter 1

Introduction

1.1 Background

Traditionally, fault tree analysis has been the leading method for reliability analysis of a complex system [1] [2]. The goal of fault tree analysis is to detect which basic event combinations lead to a failure of a system. There are many fault tree -based reliability analysis tools [3] that are widely used, such as in the safety analysis of nuclear power plants. However, fault trees can describe a system's dynamical behaviour only in limited manner. In addition, the fault tree analysis allows components to have only two states, a failure state and a functioning state. Thus, there is demand for competitive dynamic and multi-state reliability analysis methods. Dynamic reliability analysis has been studied extensively since the 1990s [4].

Dynamic flowgraph methodology (DFM) [5] is an approach used for analysing systems with time-dependences and feedback loops. DFM models are directed graphs. Variables of a system are represented as nodes with discrete states and the dynamic logic of a system is represented with discrete state transitions. As in the fault tree analysis, the goal of DFM analysis is to find out the root causes of a certain postulated event, called a top event. DFM models can be analysed either inductively or deductively. In inductive analysis, a model is simulated starting with initial conditions, whereas deductive analysis begins from a top event and the model is traced backwards. A set of timed fault trees is obtained using this process. The root causes can be identified from these timed fault trees.

VTT has been developing a DFM-based tool called YADRAT (Yet Another Dynamic Reliability Analysis Tool) since 2009 [6]. The main difference between YADRAT and the first known DFM implementation, Dymonda [7], is that YADRAT employs binary decision diagrams (BDD) [8] [9] to represent its dynamic graph model instead of transforming the model to a set of timed fault trees. YADRAT supports only deductive analysis to identify

sets of basic events that can cause a top event. In DFM, these sets of basic events are called prime implicants. Prime implicants are an extension to minimal cut sets of fault tree analysis.

The main alternative to dynamic flowgraph modelling is the Markov model approach [10]. The Markov model approach can also describe the dynamic and multi-valued logic of a system to a degree of accuracy that is comparable to DFM. The main difference is that every state transition is assigned with a probability in Markov models. A comparison between dynamic flowgraph methodology and Markov modelling coupled with cell-to-cell mapping [11] suggests that Markov models are more efficient in inductive analysis while DFM works better in deductive analysis. Other dynamic reliability analysis methods include Petri nets [12], event sequence diagrams [13], GO-FLOW methodology [14] and dynamic fault trees [15].

There are fields of reliability theory [16], such as risk importance measures [17] [18] and common cause failures (CCF) [19] [20], which have not been studied much in the context of DFM. Risk importance measures are used to analyse which components are most important with regard to the system's reliability. The importance of a component depends not only on the reliability of the component but also on its position in the system's structure. With the information provided by risk importance measures, the system's reliability can be improved effectively. The research for this thesis focuses on two traditional importance measures, the Fussell-Vesely measure and the risk increase factor, that are widely used in nuclear safety analysis. By using these two importance measures together, the influence of the component's reliability can be described completely.

A common cause failure means that more than one component fails due to a common cause. A common cause can be, for example, power outage, fire or extraordinary weather conditions. If common cause failures are not taken into account, the system's reliability might be overestimated. In fault trees, CCFs can be modelled explicitly or by using parametric models. In this thesis, two parametric models are considered: the β-factor model and the α-factor model.

1.2 Objectives

The aim of this thesis is to investigate two risk importance measures, the Fussell-Vesely measure and the risk increase factor, in the DFM context. Due to multi-valued logic, a component can usually fail in more than one way in DFM. Failures can also occur at different time steps. Risk importance measures are calculated from the prime implicants of a top event. To provide all the information available in prime implicants, the traditional importance measures are generalised so that the multi-valued logic and the time aspect are

taken into account. The implementation of these two dynamic risk importance measures in YADRAT is described in detail.

Another goal of this thesis is to develop dynamic generalisations for the traditional parametric common cause failure models. In fault trees, the possibility that failures can occur at different time points due to a common cause cannot be taken into account. However, in DFM, components can fail at different time steps. This makes it possible to develop CCF models in which a common cause can affect different components of a group at different time steps. The CCF models are also implemented in YADRAT.

First, in Chapter 2, relevant parts of reliability analysis theory are introduced. In Chapter 3, dynamic flowgraph methodology and YADRAT are presented in detail. Chapter 4 addresses dynamic risk importance measures, while Chapter 5 discusses common cause failure models in the DFM context. In chapters 6 and 7, two example systems are analysed with YADRAT. Possibilities for further research are discussed in Chapter 8 and the essential parts of the thesis are summarised in Chapter 9.

Chapter 2

Theory of Fault Tree Analysis

In this chapter, the theory of fault tree analysis [1] [2] is briefly presented. First, the reader is introduced to a few concepts of reliability theory [16], Boolean algebra [21], fault trees and minimal cut sets [22]. Theories of risk importance measures [17] [18] and common cause failures [19] [20] are essential with regard to this thesis. These concepts are presented in sections 2.5 and 2.6.

2.1 Concepts of Reliability Theory

The **reliability of an item** (component or system) is defined as an ability to consistently perform a required function under given environmental and operational conditions for a specified period of time. Reliability can be numerically measured, for example, as a probability that an item is functioning at a given time. This measure is called **availability** $A(t)$. Another commonly-used measure is **frequency** of failures. In this thesis, a system or component's reliability is mainly measured by **unavailability** $Q(t)$, which is a complement of availability, a probability that an item is not functioning at a given time t.

2.2 Boolean Algebra

Boolean algebra [21] is an algebra that deals with truth values 0 and 1. 0 represents truth value *false* and 1 truth value *true*. The basic operations of Boolean algebra are AND ($\cdot$), OR ($+$) and NOT ($^-$) operations.

Boolean formulae are built over Boolean variables, logical operations and constants 0 and 1. For example $F = x \cdot y + y \cdot z + 1$ is a Boolean formula. From this point forward, the notation $\cdot$ will be left out and $x \cdot y$ is written xy.

Boolean algebra has similar associativity, commutativity and distributivity laws to

linear algebra [23]. Additionally, Boolean variables satisfy absorption laws:

$$x + xy = x \tag{2.1}$$

$$x(x + y) = x \tag{2.2}$$

A **literal** is a Boolean variable v or its complement $\overline{v}$. A literal v is a positive literal and a literal $\overline{v}$ is a negative literal. Literals v and $\overline{v}$ are opposite literals of each other. Opposite literals satisfy the following laws:

$$v + \overline{v} = 1 \tag{2.3}$$

$$v\overline{v} = 0 \tag{2.4}$$

A **product** is a set of literals that does not contain both a literal and its opposite. For example, xy, $\overline{x}y$ and $\overline{x}y\overline{z}$ are products, but $xy\overline{x}$ is not. Let V be a set of variables. A product that contains a literal build over each variable of V is called a **minterm**.

A Boolean formula can always be presented as a sum of products. However, the presentation is not usually unambiguous. For example, consider the formula $F = xy + \overline{y}z$. The representation $xy + \overline{y}z$ is the reduced form, but F can also be presented as a sum of minterms: $xyz + xy\overline{z} + x\overline{y}z + \overline{x}\overline{y}z$. The reduced form and the representation as a sum of minterms are both unambiguous. A reduced form can always be obtained from other forms by applying the laws of Boolean algebra.

For sets S and T of Boolean products, a product is defined as

$$S \cdot T := \{\pi \cdot \rho \mid \pi \in S, \rho \in T\}, \tag{2.5}$$

where units π and ρ are Boolean products.

Let V be a denumerable set of variables. An **assignment** is a mapping $\sigma : V \to \{0, 1\}$. It assigns a value to each variable in V. Assignments can be inductively extended into mappings from Boolean formulae to $\{0, 1\}$, meaning that a formula inherits the values of the assignment. An assignment σ satisfies a formula F if $\sigma[F] = 1$, where $\sigma[F]$ is a value of the formula F inherited from the assignment σ.

If $\sigma[F] = 1 \Rightarrow \sigma[G] = 1$ for all assignments σ, it is said that F implies G and it can be denoted by $F \models G$.

2.3 Fault Tree Analysis

Fault tree analysis [1] [2] is a widely used method to analyse how a failure of a system occurs. The goal of fault tree analysis is to find **basic events** that cause a top event. These basic events are usually failure events of components or events that cause some kind of

extraordinary condition, while a **top event** is usually a failure of a system. The principle is to start from a top event and search for the factors that could lead to it occurring. The most essential part of fault tree analysis is to build a graphical presentation of basic events, which could cause the top event, and connections between them. As a result of this analysis, critical combinations of basic events are found and the probability of the top event can be calculated, if probabilities of basic events are known.

In fault trees, events are connected with each other by different logical gates. The most commonly used gates are OR and AND. Their logic is similar to the corresponding Boolean operations, + and $\cdot$. Symbols that describe these gates are shown in Figure 2.1. Figure 2.1 also contains gates EXCLUSIVE OR and K/N. For EXCLUSIVE OR, the output is true only if one and only one of the inputs is true. For K/N the output is true only if at least K of the inputs are true.

An example fault tree is shown in Figure 2.2. A fault tree can be represented as a Boolean formula F. The Boolean formula representation of the fault tree in Figure 2.2 is $F = (a+b) \cdot c \cdot (d+e) = acd + ace + bcd + bce$. The interpretation of this fault tree is that the top event can occur only if c occurs. In addition, at least a or b must occur and also at least d or e must occur.

2.4 Minimal Cut Sets

In fault tree analysis, a **cut set** is a set of basic events that can cause the top event. A **minimal cut set** is a cut set that contains the minimal amount of basic events that can cause the top event [22]. This means that if one of the basic events is taken away from a minimal cut set, it is not a cut set anymore. However, this definition is correct only for coherent fault trees. In a **coherent fault tree**, only component failures contribute to the system's failure, but in case of **non-coherent fault trees** [24], working components might also cause the top event. The case of non-coherent fault trees is considered in section 3.1.

Minimal cut sets are Boolean products consisting of positive literals. When considering a Boolean formula that does not contain any negative literals, the reduced form of the formula is actually a sum of its minimal cut sets.

The fault tree in Figure 2.2 can be represented as a Boolean formula $F = acd + ace + bcd + bce$ from which the minimal cut sets can easily be seen. They are acd, ace, bcd and bce.

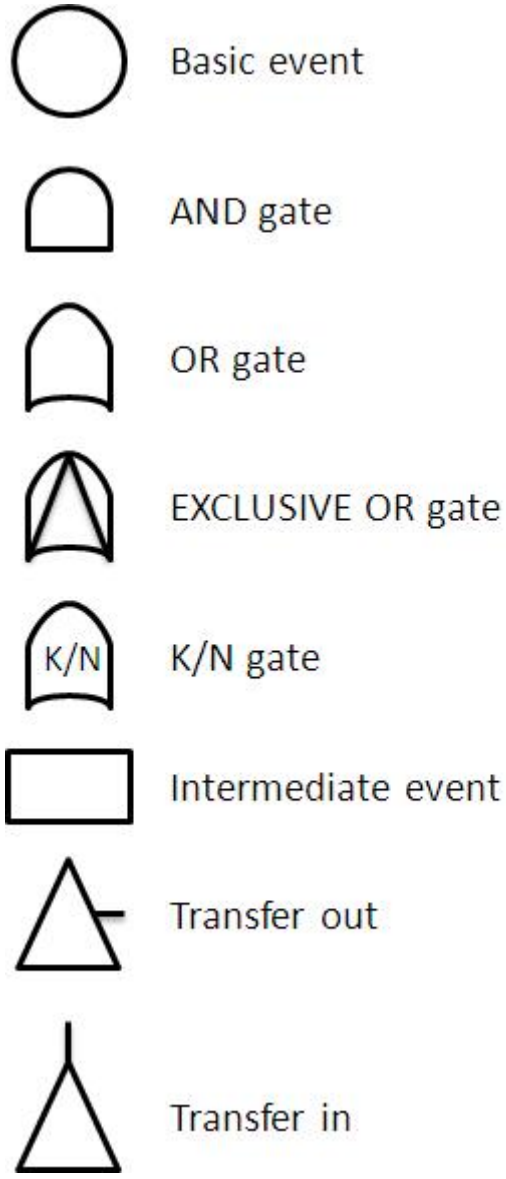

Figure 2.1: A basic event symbol, gate symbols, an intermediate event symbol and transfer symbols used in fault trees. An intermediate event is an event that occurs because of certain basic events. Transfer symbols are used when a big fault tree is divided into smaller elements. A transfer in symbol tells us that the fault tree continues from the corresponding transfer out symbol.

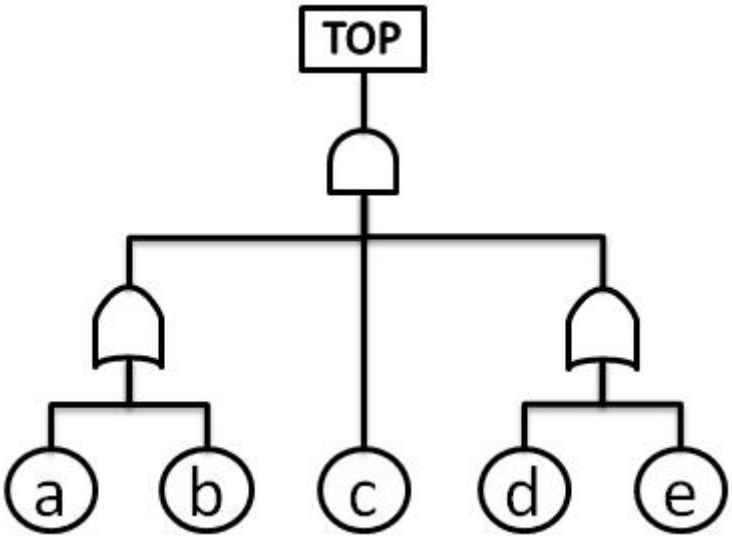

Figure 2.2: A fault tree with five basic events.

2.5 Risk Importance Measures

One of the main questions for a reliability analyst is how to improve the reliability of a system. Risk importance measures [17] [18] are used to analyse which components contribute most to the system's reliability. The importance of a component depends not only on the reliability of the component but also on its position in the system's structure. There are three different options to improve the system's reliability. First, improving the reliability of the component. Second, improving the system's design with regard to its ability to survive the failure of the component. Third, decreasing the contributing initiating event frequencies. At least two different importance measures should be used to sort out which is the best option because one measure is usually limited to describing the component's influence over the system's reliability only from one point of view, or can be high due to different reasons.

In the reliability analysis of nuclear power plants, the **Fussell-Vesely** measure of importance and the **risk increase factor** (also known as the risk achievement worth) are the most often used risk importance measures [18]. These two measures form a combination that can describe the influence of the component's unavailability completely. The Fussell-Vesely measure represents how the unavailability of the component directly affects the top event, whereas the risk increase factor only has a small dependence on the component's unavailability. It measures how much the failure of a component impacts on the system's reliability. The risk increase factor is useful when the repairing order of failed

components must be decided. The risk increase factor places more emphasis on the component's position in the system's structure and the reliability of other components, while the Fussell-Vesely measure can be high purely because of the component's high unavailability.

The Fussell-Vesely measure of importance $I^{FV}(i)$ for a component i is defined as a probability that at least one minimal cut set containing a failure of a component i has been realised assuming that the system has failed.

Definition 1. *The Fussell-Vesely measure of importance:*

$$I^{FV}(i) := \frac{Q^i_{TOP}}{Q_{TOP}}, \tag{2.6}$$

where Q_{TOP} is the probability that the system has failed and Q^i_{TOP} is the probability that the system has failed based on the minimal cut sets that include the component i.

The equation (2.7) presents a definition for the fractional contribution.

$$I^{FC}(i) := \frac{Q_{TOP} - Q_{TOP}(i=0)}{Q_{TOP}}, \tag{2.7}$$

where $Q_{TOP}(i = 0)$ is the unavailability of a system, given that the component i is working. For coherent systems, the fractional contribution is exactly the same measure as the Fussell-Vesely measure.

The difference $Q_{TOP} - Q_{TOP}(i = 0)$ can be rewritten by using availability A as the measure for reliability: $A_{TOP}(i = 0) - A_{TOP}$. Here, A_{TOP} is the probability that the top event has not occurred and $A_{TOP}(i = 0)$ is the probability that the top event has not occurred given that the component i is functioning. The fractional contribution and the Fussell-Vesely measure of importance for coherent systems have an interpretation that they measure how much the reliability of the system would increase if the component was perfect. This can be seen from the form:

$$I^{FC}(i) = \frac{A_{TOP}(i=0) - A_{TOP}}{Q_{TOP}}. \tag{2.8}$$

Definition 2. *The risk increase factor:*

$$I^{I}(i) := \frac{Q_{TOP}(i=1)}{Q_{TOP}}, \tag{2.9}$$

where $Q_{TOP}(i = 1)$ is the unavailability of the system, given that the component i has failed.

The same risk importance measures can also be calculated for basic events other than failure events of components. Anyhow, the main focus is on component failures in this thesis.

2.6 Common Cause Failures

In reality, failure events are not always independent. A failure of one component might impact on the reliability of some other components. Failures initiated by a failure of one component are called **cascading failures**. Also, several components might fail due to a common reason. In this case, a failure is called a common cause failure (CCF) [19] [20]. A common cause failure could occur, for example, due to human error, a power outage or extreme environmental conditions. The reliability of the system is overestimated if existing common causes are not taken into account.

There are two ways of modelling common cause failures: explicit and implicit modelling. The difference between **explicit** and **implicit modelling** is illustrated in figures 2.4 and 2.5. In Figure 2.4, explicit modelling of the common cause failure of components a, b and c is added to the fault tree presented in Figure 2.3. In this case, explicit modelling means that the common cause is modelled as a node that is completely separate from the basic events included in the common cause failure and has the equal effect on the top event as the occurrence of all these basic events. The probability of a common cause node does not depend on the probabilities of the basic events.

In Figure 2.5, common cause failures are modelled implicitly by replacing a basic event of a single component failure with an OR gate which connects a basic event of the component failure and common cause failures including the failure of the component. For example, a basic event a is replaced with an OR gate connecting a basic event a and common cause failure events ab, ac and abc. In this thesis, common cause failures are modelled implicitly. In implicit modelling, it is assumed that a portion of the component's failure probability is due to a CCF of a component group. Thus, probabilities of CCFs depend on probabilities of basic events via **parametric models**.

Let F be the Boolean formula represented by the fault tree shown in Figure 2.3. Clearly, $F = abc$ and abc is the only minimal cut set. Let common cause failures from Figure 2.5 be considered. Now, $F = (a+v_{ab}+v_{ac}+v_{abc})(b+v_{ab}+v_{bc}+v_{abc})(c+v_{ac}+v_{bc}+v_{abc})$, where v_{ab} is a literal representing the CCF of a and b, etc. With idempotency and absorption laws of Boolean algebra (for example $v_{ab}v_{ab} = v_{ab}$ and $v_{ab} + av_{ab} = v_{ab}$), the formula F can be reduced to the form $F = abc + v_{ab}c + v_{ac}b + av_{bc} + v_{ab}v_{ac} + v_{ab}v_{bc} + v_{ac}v_{bc} + v_{abc}$. However, products $v_{ab}v_{ac}$, $v_{ab}v_{bc}$ and $v_{ac}v_{bc}$ are usually left out in numerical calculations because their contribution to the top event probability is insignificant. In this case, $F = abc + v_{ab}c + v_{ac}b + av_{bc} + v_{abc}$ and there are five significant minimal cut sets.

There are different parametric models for the probabilities of common cause failures. The **β-factor model** is the most commonly used. It is simpler than for example the **α-factor model** that will also be presented. Let a group contain m identical components

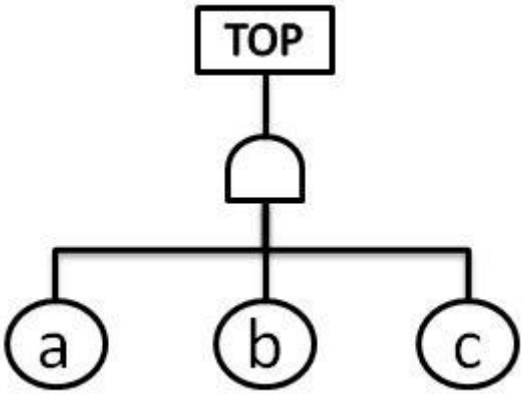

Figure 2.3: A fault tree with three basic events and an AND-operator.

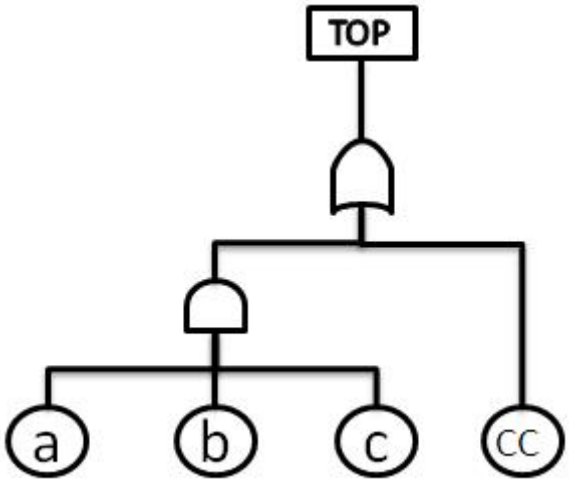

Figure 2.4: A fault tree with a common cause failure explicitly modelled as a node CC.

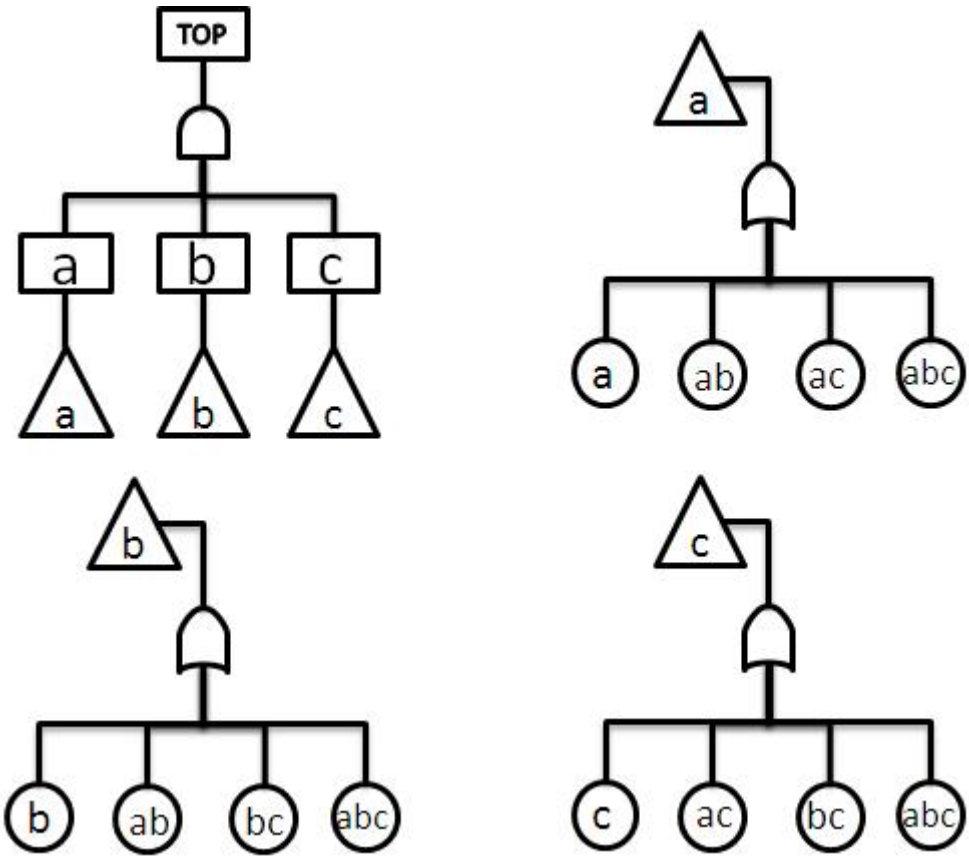

Figure 2.5: A fault tree with common cause failures implicitly modelled.

that can fail due to a common cause failure. In the β-factor model, a component has failed independently at the time t with a probability $Q^1(t) = (1-\beta) \cdot Q(t)$ and due to common cause of all m components of the group with a probability $Q^m(t) = \beta \cdot Q(t)$, where $Q(t)$ is the unavailability of a component at time t. In this model, a common cause always impacts on all components of a group.

In the generalised version of the β-factor model, components of a group do not have to be identical. Components can have different unavailabilities and only the probability of the common cause failure is same for all components. In this model, the unavailability of a component i is approximated as a sum of the independent unavailability and the common cause unavailability: $Q_i(t) \approx Q_i^1(t) + Q^m(t)$. Now, β can be defined as follows:

$$\beta := \frac{Q^m(t)}{Q^m(t) + \frac{1}{m}\sum_{j=1}^{n} Q_j^1(t)}. \tag{2.10}$$

The β-factor model can also be defined by using failure rates instead of unavailabilities.

In the α-factor model, the number of components that fail due to a common cause can vary. Components of a group are assumed to be identical. The α-factor model is presented in equations (2.11) and (2.12).

$$Q^k(t) = \frac{k}{\binom{m-1}{k-1}} \frac{\alpha_k}{\alpha_{tot}} Q(t) \tag{2.11}$$

$$\alpha_{tot} = \sum_{k=1}^{m} k\alpha_k \tag{2.12}$$

The parameter α_k is a probability that a failure event is a common cause failure of k components.

Chapter 3

Dynamic Flowgraph Modelling

In this chapter, the principles of dynamic flowgraph methodology [5] are presented. Emphasis is placed on VTT's own dynamic reliability analysis tool YADRAT [6]. In the first two sections, the reader is introduced to the concepts of prime implicants [22] and binary decision diagrams [8] [9].

3.1 Prime Implicants

In coherent fault trees, minimal cut sets are minimal sets of basic events that could cause the top event. However, in the case of non-coherent fault trees, this definition is not correct and it has to be replaced [22]. Thus, the concept of minimal cut sets is extended to the concept of prime implicants. Both minimal cut sets and prime implicants are sets of literals, in other words, products. Minimal cut sets include only positive literals, while prime implicants can also include negative literals.

Definition 3. *Let F be a Boolean formula and π be a product. The product π is an implicant of F if $\pi \models F$.*

An implicant π is a prime implicant if there is no other implicant ρ of F such that $\rho \subset \pi$.

For example, prime implicants of a formula $G = ab + c\overline{d}$ are ab and $c\overline{d}$ while the minimal cut sets are ab and c. Now, to bring in some challenge, let a formula $F = \overline{a}bc + b\overline{c}d + c\overline{d}e$ be considered. It is easy to see that $\overline{a}bc$, $b\overline{c}d$ and $c\overline{d}e$ are prime implicants. Additionally, $\overline{a}bd$ is also a prime implicant, because if $c = 1$ and $\overline{a}bd = 1$ then $\overline{a}bc = 1$ and if $c = 0$ and $\overline{a}bd = 1$ then $b\overline{c}d = 1$. The minimal cut sets of the formula F are bc, bd and ce.

A **cover** of a Boolean formula is a set of prime implicants that is logically equivalent to the formula. This means that each minterm of the Boolean formula implies at least one prime implicant of the cover. A cover is an **irredundant cover** if it ceases to be a cover

when any of the prime implicants is removed from it. A cover is a **complete cover** if it contains all possible prime implicants.

A set $\{\overline{a}bc, b\overline{c}d, c\overline{d}e\}$ is an irredundant cover of the formula $F = \overline{a}bc + b\overline{c}d + c\overline{d}e$. The complete cover of the formula F is $\{\overline{a}bc, b\overline{c}d, c\overline{d}e, \overline{a}bd\}$.

3.2 Binary Decision Diagrams

A binary decision diagram (BDD) [8] [9] is an efficient data structure for symbolic Boolean manipulation. It is a directed acyclic graph that consists of **decision nodes**, two kinds of edges, 0-edges and 1-edges, and terminal nodes, 1-terminal and 0-terminal. In a binary decision diagram, each decision node, representing a Boolean variable, has a 0-edge and a 1-edge. When a binary decision diagram represents a Boolean formula, each path from the root node to the 0-terminal or the 1-terminal represents a Boolean assignment. If a fault tree is represented as a BDD, each path ending in the 1-terminal also corresponds to a cut set. BDDs are widely used in areas of computer aided design [25] [26], fault trees [22] [27] [28] [29], event trees [30] and model checking [31].

BDDs are based on the repeated application of the Shannon expansion formula [9]:

$$F = x \cdot F|_{x=1} + \overline{x} \cdot F|_{x=0}, \tag{3.1}$$

where x is a Boolean variable.

An **ordered binary decision diagram** (OBDD) is a BDD with a constraint that the variables are ordered and the nodes are visited in increasing order in every path from a decision node to a terminal node [9]. OBDDs can be reduced by the two following rules:

1. Remove useless nodes with both edges pointing to the same node.
2. Share equivalent sub-graphs.

When these rules are applied to an OBDD, the result is called a **reduced ordered BDD** (ROBDD), which is a canonical form for BDDs. The manipulation of ROBDDs is more efficient than the manipulation of regular BDDs. In addition, the uniqueness of this representation makes testing of functional properties, such as equivalence and satisfiability, more straightforward. From this point forward, it is assumed that all the discussed BDDs are in the ROBDD form.

Another type of BDD is a **zero-suppressed BDD** (ZBDD) [32]. ZBDDs are BDDs with different semantics for nodes and they are based on the following reduction rules:

1. Remove the nodes having 1-edge pointing to the 0-terminal and connect the edges that led to the removed nodes to the nodes where the 0-edges pointed.

2. Share equivalent sub-graphs.

Basic operations, such as union ($+$) and intersection ($\cdot$), for ZBDDs, are different from the basic operations of regular BDDs. The advantages of ZBDDs become clear when very sparse Boolean formulas are handled, i.e. when a formula is 0 almost everywhere, most of the nodes can be removed with the reduction rule 1.

3.3 Dynamic Flowgraph Methodology

Dynamic flowgraph methodology (DFM) [5] is an approach for the reliability analysis of dynamic systems. In DFM models, the dynamical logic of a system is modelled as a causal relationship between variables. The time aspects of the system are represented as a series of discrete state transitions. The goal of DFM analysis is usually to identify the prime implicants of a certain postulated event. In the first known DFM approach, Dymonda, the analysis gives a set of timed fault trees from which the prime implicants can be solved in a traditional manner.

DFM models are directed graphs and in Dymonda, they consist of **variable** and **condition nodes**, **causality** and **condition edges** and **transfer** and **transition boxes**. Each transfer and transition box is associated with a **decision table**.

In DFM models, each variable, described by a node, has a finite number of predefined **states**. A state of a variable depends on the states of its input nodes and can change along with discrete time steps. A node can have several inputs, but only one output.

Transfer boxes connect nodes to each other. Dependencies between variables are described in the decision tables of transfer boxes. A decision table is an extension of a truth table.

Transition boxes are like transfer boxes, but they are associated with a **time lag**. A time lag of a transition box is a multiple of a single time step. The model can include a loop if it contains a time lag that is not 0.

Condition nodes describe conditions that can affect the system's logic. Condition nodes are connected to transfer and transition boxes with condition edges, while variable nodes are connected to transfer and transition boxes with causality edges.

An example DFM model is presented in Figure 3.1.

A DFM model can be analysed **inductively** or **deductively**. Inductive analysis proceeds from causes to effects meaning that a model is simulated with certain initial conditions. In deductive analysis, event sequences are traced backwards from effects to causes.

Deductive analysis starts from a certain postulated event of interest. The model is

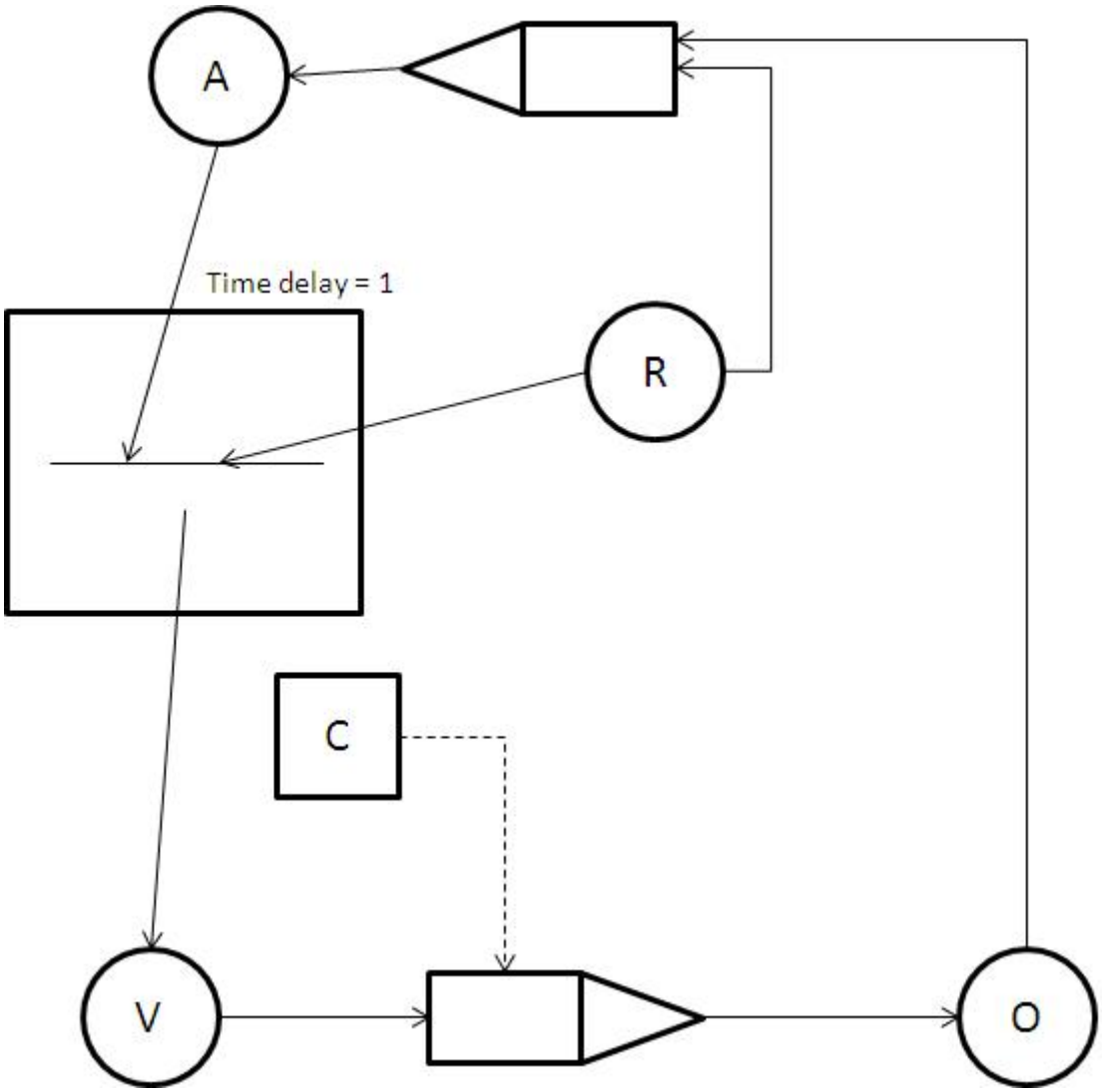

Figure 3.1: A DFM model. Nodes V, A, R and O are variable nodes, C is a condition node. The nodes V and C are connected to the variable node O via a transfer box. The variable nodes A and R are connected to the variable node V via a transition box. The variable nodes R and O are connected to the variable node A via a transfer box.

traced backwards in the cause-and-effect flow to identify what states of variables are needed to produce this postulated top event. The result of this analysis is a set of prime implicants.

In DFM analysis, an event, which is part of a sequence of events leading to a top event, is a variable in a certain state at a certain time step. Therefore, the value of a variable cannot be represented with a single Boolean literal. Thus, prime implicants cannot be defined as sets of Boolean literals as in section 3.1. Hence, the definition of a prime implicant must be extended so that a prime implicant is a set of triplets consisting of a variable, its state and the time when the variable is in that state.

In inductive analysis, all the possible consequences of the system's initial or boundary conditions are generated. The initial or boundary conditions can be either desired or undesired states. If these conditions are desired states, an inductive analysis can be used to verify system requirements, meaning that normal operation under normal conditions does not lead to undesired states. If these conditions are undesired states, inductive analysis can be used to verify the systems safety behaviour. In this thesis, only deductive analysis is considered.

3.4 YADRAT

YADRAT (Yet Another Dynamic Reliability Analysis Tool) approach [6] is a reliability analysis tool based on DFM. A YADRAT model is a directed graph and YADRAT can also be used to model the dynamical behaviour of a complex system. The main difference between YADRAT and the first known DFM implementation, Dymonda [7] appears in the computation of prime implicants. The algorithm YADRAT uses is fundamentally different from the algorithm Dymonda uses. In addition, the elements of a YADRAT model are slightly different from the ones in a Dymonda model. Only deductive analysis is possible with YADRAT.

A YADRAT model consists of two kinds of nodes, **deterministic nodes** and **stochastic nodes**, edges that connect them and decision tables of deterministic nodes. A deterministic node can have a finite number of predefined states and it must have at least one input node. The state of a deterministic node is determined by the values of the input nodes. The dependence between a deterministic variable and its input nodes is described in the corresponding decision table. An example of a decision table is shown in table 3.1.

Stochastic nodes can be divided into two groups, **failure nodes** and **random nodes**. Failure nodes represent non-decreasing Boolean variables meaning that once they are set to 1, they will remain in that state. In YADRAT, a component is defined to be failed when the failure node is in state 1. Random nodes get a random value at every time step.

Table 3.1: The decision table of a deterministic node A in YADRAT.

	Output	Inputs		
Node	A	A	B	C
Time lag	0	-1	0	0
	0	0	0	0
	1	0	0	1
	0	0	1	0
	0	0	1	1
	1	1	0	0
	2	1	0	1
	1	1	1	0
	1	1	1	1
	1	2	0	0
	1	2	0	1
	2	2	1	0
	2	2	1	1

Stochastic nodes cannot have any input nodes. Edges of a YADRAT model are always associated with a time lag. A time lag is a multiple of a single time step and it can be 0 if there is no delay in the system's logic. An example YADRAT model is presented in Figure 3.2.

For deterministic nodes, initial states can be defined or they can be left open with initial probabilities defined. If the initial state of a node is left open, the node is a random node at a predefined initial time step unless all the time lags of the input nodes are 0. If all the time lags are 0, the state of a deterministic node is determined by the inputs at the initial time step.

Deductive analysis of a YADRAT model starts with a top event. In YADRAT, a top event is defined as states of variables at particular times. Starting from a top event, the model is backtracked through the network until the predefined initial time step is reached and prime implicants are found. As defined in the previous section, a prime implicant is a set of triplets consisting of a variable name, a state and a time step. Let the concept of a literal be extended so that literals are above-mentioned triplets in YADRAT.

In a prime implicant, a literal can represent either an initial state of a deterministic node, a state of a failure node at some time step after the initial time step or a state of a random node at any time step. If there are time lags bigger than 1 (or smaller than -1 to be accurate) in a model, it is possible that a time step preceding the initial time step

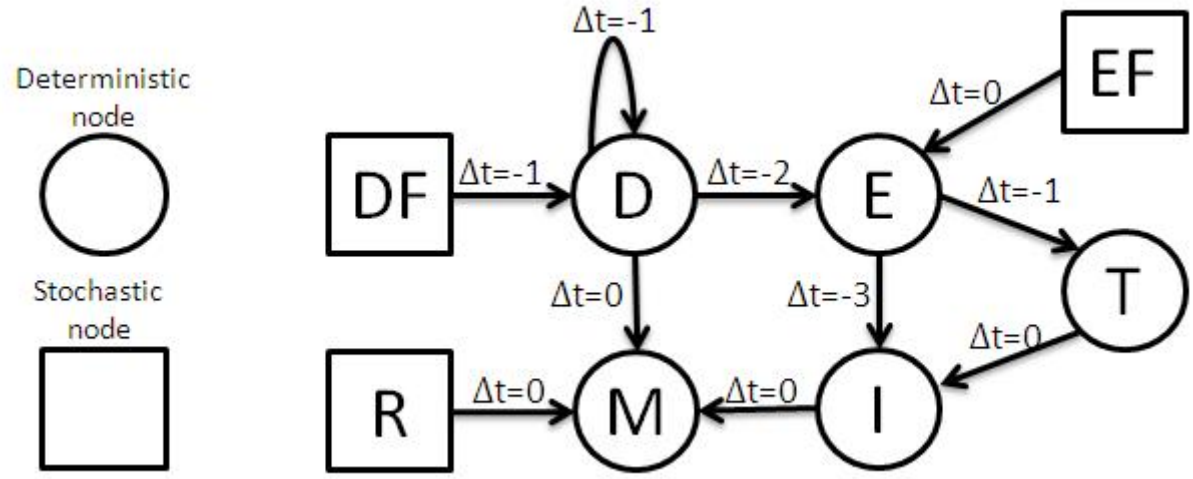

Figure 3.2: A YADRAT model.

is reached in the backtracking process, and hence, there can also be literals representing deterministic or random nodes at time steps earlier than the initial time step in prime implicants. At time steps before the initial time step, nodes are in same states as they are at the initial time step.

Algorithms that YADRAT uses to identify prime implicants are based on a decomposition theorem [29] [33], which states that the set of prime implicants can be divided into three sets with regard to a Boolean variable. When it is recursively applied to a Boolean formula, all the prime implicants can be found.

Theorem 1. *The decomposition theorem: Let F be a Boolean formula, x a Boolean variable and $PI[F]$ a set of its prime implicants. Then*

$$\begin{aligned} PI[F] =& PI[F|_{x=1} \cdot F|_{x=0}] \\ & \cup \{\overline{x}\} \cdot PI[F|_{x=0}] \setminus PI[F|_{x=1} \cdot F|_{x=0}] \\ & \cup \{x\} \cdot PI[F|_{x=1}] \setminus PI[F|_{x=1} \cdot F|_{x=0}]. \end{aligned} \tag{3.2}$$

In theorem 1, $PI[F|_{x=1} \cdot F|_{x=0}]$ is the set of prime implicants that do not contain the literal x or its negative $\overline{x}$, $\{\overline{x}\} \cdot PI[F|_{x=0}] \setminus PI[F|_{x=1} \cdot F|_{x=0}]$ is the set of prime implicants that do contain $\overline{x}$ and $\{x\} \cdot PI[F|_{x=1}] \setminus PI[F|_{x=1} \cdot F|_{x=0}]$ is the set of prime implicants that do contain x. The proof of theorem 1 can be found in [29].

In section 3.1, the prime implicants of the Boolean formula $F = \overline{a}bc + b\overline{c}d + c\overline{d}e$ were identified to be $\overline{a}bc$, $b\overline{c}d$, $c\overline{d}e$ and $\overline{a}bd$. Using the Shannon expansion formula, F can be written in the following form:

$$F = c \cdot (\overline{a}b + \overline{d}e) + \overline{c} \cdot (bd). \tag{3.3}$$

Now, theorem 1 can be applied to this formula:

$$\begin{aligned} PI[F] =& PI[(\overline{a}b + \overline{d}e) \cdot bd] \\ & \cup \{\overline{c}\} \cdot PI[bd] \setminus PI[(\overline{a}b + \overline{d}e) \cdot bd] \\ & \cup \{c\} \cdot PI[\overline{a}b + \overline{d}e] \setminus PI[(\overline{a}b + \overline{d}e) \cdot bd] \\ =& \{\overline{a}bd\} \cup \{b\overline{c}d\} \cup \{\overline{a}bc, c\overline{d}e\} \end{aligned} \tag{3.4}$$

Now, the prime implicants are divided into three sets with regard to the variable c. After this, theorem 1 could again be applied to the third set to get all the prime implicants separated.

In YADRAT, prime implicants are identified by transforming a YADRAT model to a binary decision diagram. The BDD is constructed based on decision tables. For computation of prime implicants, there are two approaches. Both are based on theorem 1. The first one deals with meta-products [34] and the second uses zero-suppressed BDDs [22]. Algorithms that are used to identify prime implicants are presented in [6], along with the algorithm that is used to construct a BDD from a YADRAT model.

Both the ZBDD and the meta-product approaches calculate a complete cover of a function representing a top event. Additionally, the CUDD package [35], which provides functions for BDD manipulation, offers an algorithm to calculate an irredundant cover of a top function. The chosen order of variables affects significantly to the size and structure of a BDD. The irredundant cover depends on the order of variables due to this reason. If the order of variables is changed, the irredundant cover might change as well while the complete cover is always the same. Because of this the complete cover is preferred in qualitative analysis.

Consider the example of decision table 3.1. Let node B be a failure node, node C be a random node, the initial time step be -3 and the initial states be left open. Table 3.2 presents a set of prime implicants of a top event $A(0) = 2$.

BDDs encode Boolean formulae but in YADRAT, a variable can have more than two states. Because of this, the variable's states need to be coded into Boolean values. Let $S = \{0, 1, \ldots, k-1\}$ be the set of the variable's states. Now, let a unique Boolean vector $(v_1, v_2, \ldots, v_n)$ be assigned to each element $s \in S$. There are 2^n Boolean vectors of length n. To code the states efficiently, the length of the vector should be as small as possible. This is realised when n is chosen so that $2^{n-1} < |S| \leq 2^n$, where $|S|$ is the number of elements of S. If $2^{n-1} < |S| < 2^n$, the state space is extended so that it contains 2^n states because decision tables must be completely defined by the input variables represented as Boolean vectors. This enables the correct computation of prime implicants and reduces post-processing of prime implicants. The extended states of the variable are interpreted

Table 3.2: The prime implicants of the top event $A(0) = 2$.

	Prime implicant
1	$\{A(-3) = 0, C(-2) = 0, C(-1) = 1, B(0) = 0, C(0) = 1\}$
2	$\{A(-3) = 0, C(-2) = 1, B(-1) = 0, C(-1) = 1, B(0) = 1\}$
3	$\{C(-2) = 1, C(-1) = 0, B(0) = 0, C(0) = 1\}$
4	$\{A(-3) = 1, C(-1) = 0, B(0) = 0, C(0) = 1\}$
5	$\{A(-3) = 1, C(-2) = 1, B(0) = 0, C(0) = 1\}$
6	$\{A(-3) = 1, B(-2) = 0, C(-2) = 1, B(-1) = 1\}$
7	$\{A(-3) = 1, C(-2) = 0, B(-1) = 0, C(-1) = 1, B(0) = 1\}$
8	$\{A(-3) = 2, B(-2) = 1\}$
9	$\{A(-3) = 2, C(-1) = 0, B(0) = 0, C(0) = 1\}$
10	$\{A(-3) = 2, B(-1) = 0, C(-1) = 1, B(0) = 1\}$

as the largest value of the original state space. Because of this, duplicate prime implicants may occur but they can be filtered out in the post-processing phase. The post-processing of prime implicants is described in detail in [6].

Consider the variable A from table 3.1. It has three states, $S = \{0, 1, 2\}$. These states must be represented with a Boolean vector of length 2 because $2^1 < |S| < 2^2$. The state 0 is coded to the vector $(0, 0)$, state 1 to $(0, 1)$ and the state 2 to $(1, 0)$. In the extended state space, there will also be a fourth state coded to $(1, 1)$. It is interpreted as state 2 of the original state space.

Failure nodes of a YADRAT model represent non-decreasing Boolean variables. In the construction of a BDD from a YADRAT model, failure nodes are handled with an assistive random node and a decision table. In the decision table, the state of the assistive random node and the state of the failure node on the previous time step are inputs and the state of the failure node at the current time step is an output. This decision table provides the non-decreasing property for the failure node. This is illustrated in table 3.3.

For failure probabilities, a constant model is currently used in YADRAT. A user gives a probability q as a parameter for each failure node. The probability of a literal representing a failure node in state 1 is q regardless of the time step and the probability of a literal representing a failure node in state 0 is $1 - q$.

An upper bound for the top event probability is calculated with a following formula:

$$P_{tot} = 1 - \prod_{i=1}^{n} (1 - P_i), \tag{3.5}$$

where P_{tot} is the top event probability, P_i is the probability of the i:th prime implicant, and n is the number of prime implicants. If all prime implicants were independent, this

Table 3.3: The decision table of a failure node.

	Output	Inputs	
Node	FN	FN	H
Time lag	0	-1	0
	0	0	0
	1	0	1
	1	1	0
	1	1	1

formula would give an exact top event probability. A probability of a prime implicant is calculated as a product of probabilities of its literals.

Chapter 4

Risk Importance Measures for YADRAT

4.1 Risk Importance Measures for Dynamic and Multi-state Systems

There has been little study done on risk importance measures in dynamic reliability analysis [4]. As opposed to traditional fault tree analysis, components have multiple states in dynamic flowgraph methodology. Because of this, a component can usually fail in multiple different ways. This needs to be taken into account in the construction of risk importance measures for DFM. The time when a failure occurs also matters in models with dynamic logic. Thus, the time aspect needs somehow to be included in dynamic risk importance measures.

There are some risk importance measures developed for dynamic systems [36] [37] [38] and multi-state systems [39] [40] [41], but none of them would be useful in YADRAT as such. Markov models [10] constitute a dynamic reliability analysis approach that is comparable to DFM [11]. Markov models can be used to analyse dynamic multi-state systems as DFM models. Some studies have been carried out on risk importance measures for Markov models [36] [37] [38]. However, it is difficult to formulate similar importance measures for dynamic flowgraph methodology because it is not known how transition rates matrices should be built from DFM models.

There are two types of importance measures for multi-state systems [39]. Measures of type 1 measure the effect that a component has on a system's reliability. Type 1 measures are useful when it is analysed if the number of redundant components needs to be increased. Measures of type 2 measure how a certain state or states of a component affect the system's reliability. Type 2 measures provide guidance on how a component

should be changed so that the system's reliability would improve. Many risk importance measures of traditional fault tree analysis can be generalised for multi-state systems.

An approach of type 2 is to transform a multi-state variable into a binary variable by dividing the states into two sets with regard to a specified performance level [40]. When a multi-state variable is treated as a binary variable, traditional risk importance measures can be applied to it. This approach could be used in DFM. However, in YADRAT, component failures are defined with failure nodes. Therefore, there is no need treat any states of a component as a failure.

Another approach for importance of multi-state components is composite importance measures [41], which are weighted averages of state importances and represent type 1 measures. YADRAT would not benefit from them either because failures are well-defined with failure nodes.

In the fault tree analysis, the failure of a component is often only one basic event. However, components with multiple states can usually fail in different ways. In the fault tree analysis, this can be taken into account by defining a basic event for each failure mode of a component. In YADRAT, let a component be defined to be a deterministic node that has a failure node as an input node. Hence, the state of a component is one variable and each component is also associated with another variable which determines if the component is failed or not.

In YADRAT models used in the work of this thesis, all failures are such that a component is stuck in one of its states. In other words, after a failure node is set to state 1 for the rest of the scenario, the component that is dependent on the failure node is held in one of its states for the rest of the scenario as well. Let a state in which a component is stuck be called a **failure state**. Table 4.2 illustrates how a component from table 4.1 can be stuck in different states. It should be noted that failures can be defined similarly in all DFM-based approaches.

Table 4.3 presents some YADRAT-related concepts that are used in this and subsequent chapters.

4.2 The Dynamic Fussell-Vesely Measure of Importance

Generally speaking, the Fussell-Vesely measure of importance is calculated as

$$I^{FV}(H) := \frac{P(PI \cap H)}{P(PI)}, \tag{4.1}$$

where PI is the set of prime implicants of a top event, $PI \cap H$ is a subset of the set of prime implicants, $P(PI)$ is the probability that at least one prime implicant realises and $P(PI \cap H)$ is the probability that at least one prime implicant from the subset realises. The

Table 4.1: The decision table of a component N that depends on a failure node F.

	Output	Inputs	
Node	N	N	F
Time lag	0	-1	-1
	1	0	0
	0	0	1
	0	1	0
	1	1	1

Table 4.2: An example on how a failure affects the component N. In case 0, the component does not fail. In case 1, the failure occurs at time step -3 and the failure state is 0. In case 2, the failure occurs at time step -2 and the failure state is 1.

	Case 0			Case 1			Case 2	
Time	N	F		N	F		N	F
-4	1	0		1	0		1	0
-3	0	0		0	1		0	0
-2	1	0		0	1		1	1
-1	0	0		0	1		1	1
0	1	0		0	1		1	1

Table 4.3: Concepts of YADRAT

Concept	**Explanation**
Component	A deterministic node that has a failure node as an input node.
Component is failed	A failure node that is connected to the component is in state 1.
Failure/Failure event	The state of a failure node changes from 0 to 1.
Failure time	The time step at which a failure node is in state 1 for the first time.
Failure state	The state in which a component is stuck after a failure event.
Component is functioning	When the failure node that is connected to a component is in state 0, the component is functioning.
State probability	The probability that a node is in certain state.
Initial time	A predefined time step at which the backtracking is stopped.
Initial state	A state in which a node is at the initial time. A user can define initial states for deterministic and random nodes. The initial state of a failure node is always 0.
Initial probabilities	If the initial state of a node is left open, each state of the node is associated with a state probability at the initial time.
Time before initial time	The analysed system is assumed to be in a steady state before the initial time. No failures have occurred and deterministic nodes and random node are in the same state as they are at the initial time.
Timed node	A node at a given time step
Time lag of a failure node	The time lag of a failure node that is in the decision table of the corresponding component (this lag is defined to be negative or 0)

Fussell-Vesely measure is always calculated for some sort of subset of prime implicants. Most often, this subset is a set of prime implicants that contain a failure of a certain component. However, it can also be something else. For multi-state systems, the subset could be a set of prime implicants that contain a certain state of a component. In dynamic flowgraph methodology, there are also possibilities to limit the subset with regard to time steps.

The Fussell-Vesely measure of importance gives a probability that at least one minimal cut set containing a failure event has realised assuming that the system has failed. In YADRAT, it would be beneficial to calculate the Fussell-Vesely measure separately for each failure state of a component to understand which kind of failures of the component contribute most to a top event. The Fussell-Vesely measure for a failure state is presented in equation (4.2), where $Q_{TOP}^{i_s}$ is the probability that the system has failed based on a prime implicant including a failure of a component i to a state s.

$$I^{FV}(i_s) := \frac{Q_{TOP}^{i_s}}{Q_{TOP}} \tag{4.2}$$

In the dynamic flowgraph methodology, the order of events leading to a top event is important. Thus, it could be interesting to know when a certain failure usually occurs when it is part of a prime implicant leading to a top event. Because of this, the dynamic version of the Fussell-Vesely measure of importance should be calculated separately for each time step. In construction of the dynamic Fussell-Vesely (DFV), the interpretation of the traditional Fussell-Vesely measure should be kept in mind. In the case of coherent systems, the Fussell-Vesely can be interpreted so that it measures how much the reliability of a system would increase if a component was perfect. To maintain this idea, the dynamic Fussell-Vesely should be constructed in such way that it could be interpreted as how much the reliability of a system would increase if a component could be made not to fail in a certain state at least until a certain time step, when a system is coherent with regard to the failure node connected to the considered component. Thus, as the definition of the Fussell-Vesely deals with prime implicants that include a certain component failure, the definition of the DFV should respectively consider prime implicants that include a certain component failure in a certain state before or at a certain time step.

Let it be defined that the time step of the top event is 0 and the initial time is $-n$ ($n \in \mathbb{N}$).

Definition 4. *The dynamic Fussell-Vesely measure of importance for a failure state of a component:*

$$I^{DFV}(i_s(-t)) := \frac{Q_{TOP}^{i_s(-t)}}{Q_{TOP}}, \tag{4.3}$$

where $Q_{TOP}^{i_s(-t)}$ is a probability that at least one prime implicant, including a failure of a component i to a state s before or at a time step $-t$ $(0 \leq t < n)$, has realised.

The DFV $I^{DFV}(i_s(0))$ is basically the same measure as $I^{FV}(i_s)$ in (4.2). When the failure time is not considered interesting, all the attention can be paid to the DFV of time step 0 because it takes failures from all time steps into account.

The Dynamic Fussell-Vesely in definition 4 is a measure of type 2 as it considers the effect that a state of a component has on the system's reliability. Sometimes analysts are only interested in measures of type 1. Thus, there is demand for a dynamic Fussell-Vesely measure that takes into account all the failures without considering different failure states separately.

Definition 5. *The dynamic Fussell-Vesely measure of importance for a failure of a component:*

$$I^{DFV}(i(-t)) := \frac{Q_{TOP}^{i(-t)}}{Q_{TOP}}, \tag{4.4}$$

where $Q_{TOP}^{i(-t)}$ is a probability that at least one prime implicant, including a failure of a component i before or at a time step $-t$, has realised.

The dynamic Fussell-Vesely measure of importance for a failure of a component is, in practice, the dynamic Fussell-Vesely for the state 1 of a failure node.

When a failure node is in state 0, the corresponding component is functioning as it is meant to. It might be interesting to know if a system is non-coherent with regard to a failure of a certain component. Therefore, it could be useful to measure how much state 0 of a failure node contributes to a top event. The dynamic Fussell-Vesely measure is formulated for state 0 of a failure node in definition 6. The time dependence is inverted compared to the failure event DFV for interpretative reasons.

Definition 6. *The dynamic Fussell-Vesely measure of importance for a state 0 of a failure node:*

$$I^{DFV}(i_{f=0}(-t)) := \frac{Q_{TOP}^{i_{f=0}(-t)}}{Q_{TOP}}, \tag{4.5}$$

where $Q_{TOP}^{i_{f=0}(-t)}$ is the probability that at least one prime implicant, including a literal representing the failure node f of a component i in state 0 at time step $-t$ or later, has realised.

YADRAT models also include random nodes that typically represent randomly occurring conditions that affect the system's logic. The dynamic Fussell-Vesely can be formulated for them in similar manner as for components.

Definition 7. *The dynamic Fussell-Vesely measure of importance for a random node:*

$$I^{DFV}(r_s(-t)) := \frac{Q_{TOP}^{r(-t)=s}}{Q_{TOP}}, \tag{4.6}$$

where $Q_{TOP}^{r(-t)=s}$ is the probability that at least one prime implicant, including a random node r in state s before or at time step $-t$, has realised.

In YADRAT, the initial state of a deterministic node is not always defined. If the initial state is left open, a probability is given to each state and the node is considered to be a random node at the initial time step. The Fussell-Vesely measure can be formulated for initial states of a deterministic node to measure how much each initial state contributes to the top event probability.

Definition 8. *The Fussell-Vesely measure of importance for an initial state of a deterministic node:*

$$I^{FV}(d_s) := \frac{Q_{TOP}^{d=s}}{Q_{TOP}}, \tag{4.7}$$

where $Q_{TOP}^{d=s}$ is the probability that at least one prime implicant, including a deterministic node d in state s before or at the initial time step $-n$, has realised.

4.3 The Dynamic Risk Increase Factor

The risk increase factor measures how much the unavailability of a system increases if a component fails. Thus, in the calculation of the risk increase factor it must be assumed that a component is failed. In DFM, a component can fail at different time steps. The original idea of the risk increase factor can be maintained in formulation of the dynamic risk increase factor if a component is assumed to be failed throughout the whole scenario. In other words, a component must fail at the initial time. Again, different failure states can be considered separately if a type 2 measure is desired.

Definition 9. *The dynamic risk increase factor (DRIF) for a failure state of a component:*

$$I^{DI}(i_s) := \frac{Q_{TOP}(f(-n) = 1, i(-n+l) = s)}{Q_{TOP}}, \tag{4.8}$$

where f is a failure node connected to a component i, $-l$ is the time lag of the failure node in the decision table of the component i and $Q_{TOP}(f(-n) = 1, i(-n+l) = s)$ is the probability that the top event occurs assuming that a component i is failed to a state s at the initial time $-n$.

The dynamic risk increase factor can be defined as a type 1 measure for a component failure similarly, with an exception that the component may fail in any of its states. In

practice, the dynamic risk increase factor for a component failure is the dynamic risk increase factor for the state 1 of a failure node.

Definition 10. *The dynamic risk increase factor for a component failure:*

$$I^{DI}(i) := \frac{Q_{TOP}(f(-n) = 1)}{Q_{TOP}}, \tag{4.9}$$

where f is a failure node connected to a component i and $Q_{TOP}(f(-n) = 1)$ is the probability that the top event occurs assuming that the component i is failed at the initial time step $-n$.

In YADRAT, a random node gets a new value at each time step regardless of the earlier values. However, the dynamic risk increase factor for a random node should be consistent with the preceding idea that the failure conditions last the whole analysis time from the initial time step until the time step of a top event. Thus, in the dynamic risk increase factor for a random node, it should be assumed that the random node is in same state throughout the whole scenario.

Definition 11. *The dynamic risk increase factor for a random node:*

$$I^{DI}(r_s) := \frac{Q_{TOP}(r(-t) = s,\ \forall\, t \in \{0, 1, ..., n-1, n\})}{Q_{TOP}}, \tag{4.10}$$

where $Q_{TOP}(r(-t) = s,\ \forall\, t \in \{0, 1, ..., n-1, n\})$ is the probability that the top event occurs assuming that a random node r is in state s at every time step.

The risk increase factor can also be formulated for an initial state of a deterministic node. It measures to what extent the top event probability increases if the initial state of a deterministic node is defined to be a certain state.

Definition 12. *The risk increase factor for an initial state of a deterministic node:*

$$I^{I}(d_s) := \frac{Q_{TOP}(d = s)}{Q_{TOP}}, \tag{4.11}$$

where $Q_{TOP}(d = s)$ is the probability that the top event occurs assuming that the initial state of a deterministic node d is s.

4.4 Other Dynamic Risk Importance Measures

Most of the traditional risk importance measures [18] can be derived from the Fussell-Vesely measure (or the fractional contribution) and the risk increase factor in the case of

a coherent system. For example, the risk decrease factor (also known as the risk reduction worth), defined as

$$I^R(i) := \frac{Q_{TOP}}{Q_{TOP}(i=0)}, \tag{4.12}$$

can be derived from the Fussell-Vesely measure as follows:

$$I^R(i) = \frac{1}{1 - I^{FV}(i)}. \tag{4.13}$$

However, for non-coherent systems, the risk decrease factor or other measures that rely on the risk decrease ($Q_{TOP} - Q_{TOP}(i=0)$) cannot accurately be derived from the Fussell-Vesely. In a non-coherent case, the fractional contribution can be used instead of the Fussell-Vesely measure.

The dynamic fractional contribution (DFC) can be defined as follows:

$$I^{DFC}(i_s(-t)) := \frac{Q_{TOP} - Q_{TOP}(\{f(-t)=0\} \cup \{i(-t+l) \neq s\})}{Q_{TOP}}, \tag{4.14}$$

where f is a failure node connected to a component i, $-l$ is the time lag of the failure node in the decision table of the component i and $Q_{TOP}(\{f(-t)=0\} \cup \{i(-t+l) \neq s\})$ is the top event probability given that a component i has not failed to state s before or at a time step $-t$.

When a failure node does not appear in prime implicants in state 0, the dynamic Fussell-Vesely and the dynamic fractional contribution give basically the same value (results may differ due to different calculation methods). The dynamic fractional contribution is more demanding to calculate because a new set of prime implicants needs to be solved to compute the conditional top event probability $Q_{TOP}(\{f(-t)=0\} \cup \{i(-t+l) \neq s\})$. The dynamic fractional contribution can be negative and is always smaller than the dynamic Fussell-Vesely, which is always between 0 and 1.

The risk decrease factor can be derived from the fractional contribution in the dynamic case as well. An interpretation of the dynamic risk decrease factor for a failure state of a component would be that it measures how much the unavailability of a system would be reduced if it could be ensured that the component would not fail in a certain state before or at a certain time step. However, the translation of all other traditional risk importance measures to the dynamic case is not as straightforward. For example, the traditional Birnbaum importance, defined as

$$I^B(i) := Q_{TOP}(i=1) - Q_{TOP}(i=0), \tag{4.15}$$

can be calculated from the fractional contribution and the risk increase factor as in equation (4.16).

$$I^B(i) = Q_{TOP} \cdot (I^I(i) - 1 + I^{FC}(i)) \tag{4.16}$$

The same can be carried out in a dynamic case if the dynamic Birnbaum importance measure is defined as a difference of the top event probability, assuming that a component fails (in a certain state) at the initial time step and the top event probability assuming that a component does not fail (in a certain state) at all. Equation (4.17) presents how the dynamic Birnbaum importance can be calculated from the DFC and the DRIF.

$$I^{DB}(i_s) = Q_{TOP} \cdot (I^{DI}(i_s) - 1 + I^{DFC}(i_s(0))) \tag{4.17}$$

Therefore, in the calculation of the dynamic Birnbaum importance, only the dynamic fractional contribution of the time step 0 is used. The conclusion is that these kinds of generalisations concerning the dynamic case are usually possible but they have to be made carefully, especially in the case of those risk importance measures that are derived from both the fractional contribution and the risk increase factor.

4.5 Implementation of the Dynamic Fussell-Vesely

In the calculation of the dynamic Fussell-Vesely for a failure state of a component in YADRAT, each prime implicant is examined. First, a check is made to see if a prime implicant contains a literal representing the failure node in state 1. If it does, the failure state must be solved. If the time lag of the failure node in the decision table of the component is $-l$, the failure state is the state in which the component is at the l:th time step after the failure. If the time lag is 0, the failure state is the state of the component at the time step of the failure. Depending on the time lag, the model is worked backwards either from the time of the failure or some time step after the failure. The goal of the backtracking process is to calculate the probabilities for the component to be in each of its states at the given time step using the decision tables under the conditions set by the prime implicant.

In YADRAT, failure nodes have a property that means that once they are set to state 1, they stay there. Because of this, a literal representing a failure node in state 1 at a time step $-t$ ($t > 0$) implies a literal representing the failure node in state 1 at time step $-t + 1$. Due to this, if $\{D(-4) = 1, E(-4) = 0, F(-2) = 1\}$ is a prime implicant, then $\{D(-4) = 1, E(-4) = 0, F(-3) = 1\}$ is necessarily at least an implicant if not a prime implicant, when F is a failure node. Therefore, a set of prime implicants usually contain such pairs of prime implicants that are otherwise similar except that they contain a literal representing a failure node in state 1 at different time steps.

If a prime implicant contains a literal representing a failure node in state 1 at a later time step $-t$ than the first time step after the initial time step ($-t > -n + 1$) and does not contain a literal representing a failure node in state 0 at time step $-t - 1$, the prime implicant realises if the failure occurs at time step $-t$ or at time step $-t - 1$. However, there must also be a prime implicant of a similar type with a literal representing a failure node in state 1 at time step $-t - 1$. When considering the preceding prime implicant with the failure node in state 1 at time step $-t$ in the calculation of the Fussell-Vesely, it would be tempting to think that the possibility of the failure happening already at time step $-t - 1$ should be taken into account. If this is the case, the same contribution of the failure is calculated twice as there is also a prime implicant with failure node in state 1 at time step $-t - 1$. This would lead to overestimation of the dynamic Fussell-Vesely of time step $-t - 1$. Thus, to obtain accurate results in the calculation of the DFV from prime implicants, it must be assumed that a failure occurs exactly at the time step that is read from a prime implicant and at earlier time steps, the corresponding component is functioning.

The backtracking is done by recursively calculating the state probabilities of an output node from the state probabilities of input nodes under the conditions set by the prime implicant. During the calculation process, each node is always associated with a time step at which the state probabilities of the node must be determined. The time step of an input node is the time step of an output node minus the time lag of the input node. In the process, each input node is examined one by one. First, it is checked if the node is a failure node, a random node or a deterministic node. Each node type is treated differently. Different cases for each node type are presented in tables 4.4, 4.5 and 4.6.

State probabilities of a node depend on its appearance in the prime implicant and the time steps at which it appears in the prime implicant. Let the time step of the considered input node be $-t$ in tables 4.4, 4.5 and 4.6. Time step $-n$ is again the initial time. A time step $-t_0$ refers to the time step of a literal representing the considered failure node in state 0, a time step $-t_1$ refers to the time step of a literal representing the considered failure node in state 1 and a time step $-t_s$ refers to the time step of a literal representing the considered node in state s. In tables 4.4, 4.5 and 4.6, "PI includes" columns reveal which literals are included in the prime implicant. The symbol Y means that the prime implicant contains the literal, N means that the prime implicant does not contain the literal and I means that it is irrelevant whether the prime implicant contains the literal or not. When no literal determines the state of the node, state probabilities are obtained from the model parameters defined by the user, or in the case of a deterministic node at a later time step than the initial time, they are calculated recursively from the state probabilities of the

Table 4.4: Different cases for a failure node F in the backtracking process.

Case	Time condition	PI includes		State probabilities
		$F(-t_0) = 0$	$F(-t_1) = 1$	
F1.	$-t \geq -t_1$	I	Y	State is 1 with the probability 1
F2.	$-t_1 > -t > -n$	I	Y	State is 0 with the probability 1
F3.	$-t > -t_0$	Y	N	State probabilities from model parameters
F4.	$-t_0 \geq -t > -n$	Y	I	State is 0 with the probability 1
F5.	$-t > -n$	N	N	State probabilities from model parameters
F6.	$-t \leq -n$	I	I	State is 0 with the probability 1

Table 4.5: Different cases for a random node R in the backtracking process.

Case	Time condition	PI includes	State probabilities
		$R(-t_s) = s$	
R1.	$-t = -t_s$	Y	State is s with the probability 1
R2.	$-t \leq -n$	N	Initial state probabilities from model parameters
R3.	$-t > -n$	N	State probabilities from model parameters

input nodes of the considered node. When considering a deterministic node at the initial time, the time lags of the input nodes determine whether the backtracking is continued or not.

After the state probabilities of the input nodes are obtained, the state probabilities of the output node are calculated with the help of the decision table. Each row of the decision table, containing a combination of input states and an output state, is examined. For each combination of input states, a probability is calculated and this probability is added to the probability of the corresponding output state. The rows containing additional values that were added in the state space extension process are ignored. As a result, the state probabilities of the output node are obtained.

The probability that at least one prime implicant, including the failure of a component i to a state s before or at a time step $-t$, has realised is calculated as follows:

$$Q_{TOP}^{i_s(-t)} = 1 - \prod_{k \in K_i(-t)} (1 - P_k \cdot Q(i_s(-t)|PI_k)), \tag{4.18}$$

where P_k is the probability that a prime implicant k has realised, $K_i(-t)$ is the set of prime implicants containing a failure of a component i at time step $-t$ or earlier and $Q(i_s(-t)|PI_k)$ is the probability that a component i fails in state s at time step $-t$ or earlier assuming that a prime implicant k realises. It should be noted that the assump-

Table 4.6: Different cases for a deterministic node D in the backtracking process. The symbol Y for a literal $D(-n) = s$ implies that the prime implicant includes D at the initial time or earlier.

Case	Time condition	PI includes $D(-n) = s$	Input lags	State probabilities
D1.	$-t > -n$	I	I	From state probabilities of input nodes
D2.	$-t = -n$	Y	I	State is s with the probability 1
D3.	$-t = -n$	N	Not all 0	Initial state probabilities from model parameters
D4.	$-t = -n$	N	All 0	From state probabilities of input nodes
D5.	$-t < -n$	N	I	Initial state probabilities from model parameters

tion PI_k ($k \in K_i(-t)$) includes a failure of the component i at time step $-t$. Thus, $\sum_{s \in S} Q(i_s(-t)|PI_k) = 1$, where S is the set of states of the component i.

When the dynamic Fussell-Vesely is calculated for a failure event of a component, $Q(i_s(-t)|PI_k)$ can be replaced with 1 in equation (4.18) and the backtracking process is not needed.

In section 3.4, prime implicants from table 3.1 were calculated and they were presented in table 3.2. In this example, failure node B is connected to component A. Component A has three states in which it can get stuck. Let the calculation of the dynamic Fussell-Vesely measure be illustrated by this example. The probability that the top event ($A(0) = 2$) occurs due to a prime implicant including a failure of component A at time step -2 is

$$Q_{TOP}^{A(-2)} = P_8, \tag{4.19}$$

where P_8 is the probability of the prime implicant 8 in table 3.2. The prime implicant 8 is the only prime implicant that includes a literal representing the failure at time step -2. Respectively, the probability that the top event occurs due to a prime implicant including a literal representing the failure node B in state 1 at time step -1 or earlier is

$$Q_{TOP}^{A(-1)} = 1 - (1 - P_6) \cdot (1 - P_8) \tag{4.20}$$

and the probability that the top event occurs due to a prime implicant including a literal representing the failure node B in state 1 at the time step 0 or earlier is approximated as

$$Q_{TOP}^{A(0)} = 1 - (1 - P_2) \cdot (1 - P_6) \cdot (1 - P_7) \cdot (1 - P_8) \cdot (1 - P_{10}). \tag{4.21}$$

Prime implicants 2, 6, 7, 8 and 10 contain the failure of the component A. In these prime implicants, component A is always stuck in state 2 because the top event ($A(0) = 2$) could not occur if A was stuck in some other state. Anyhow, let the prime implicant 6

($\{A(-3) = 1, B(-2) = 0, C(-2) = 1, B(-1) = 1\}$) be used to illustrate how the failure state is solved. In this prime implicant, the failure occurs at time step -1. The time lag of the failure is 0. Hence, the state of the node A at time step -1 must be solved.

The timed node $A(-1)$ depends on $A(-2)$, $B(-1)$ and $C(-1)$. First, the state probabilities of $A(-2)$ are calculated. Node A is a deterministic node and time step -2 is not the initial time step. Thus, the state probabilities of $A(-2)$ are calculated from the state probabilities of $A(-3)$, $B(-2)$ and $C(-2)$. All of these three nodes are represented in the prime implicant with these time steps. In the prime implicant, $A(-3) = 1$, $B(-2) = 0$ and $C(-2) = 1$. From decision table 3.1, it can be seen that this combination gives $A(-2) = 2$. Second, in the prime implicant, $B(-1) = 1$. Third, $C(-1)$ is not represented in the prime implicant and C is a random node. Hence, both states are possible.

Now, the rows of decision table 3.1 are examined one by one. The probabilities of the ten first rows are 0. The probability of the 11th row is $P(C(-1) = 0)$ and the probability of the 12th row is $P(C(-1) = 1)$. Neither of the combinations in these rows lead to states 0 or 1 of A. Thus, $P(A(-1) = 0) = 0$ and $P(A(-1) = 1) = 0$. Both of the combinations lead to state 2. Hence, $P(A(-1) = 2) = P(C(-1) = 0) + P(C(-1) = 1) = 1$. Therefore, the only possible failure state is state 2. The same backtracking process is illustrated in table 4.7. Because the failure state is always 2 in the prime implicants, $Q_{TOP}^{A_2(-t)} = Q_{TOP}^{A(-t)}$ and $Q_{TOP}^{A_0(-t)} = Q_{TOP}^{A_1(-t)} = 0$ for all $t \in \{0, 1, 2\}$.

Two other examples of solving the failure state probabilities by backtracking are presented in appendixes A and B.

The calculation of the dynamic Fussell-Vesely measure for state 0 of a failure node is more straightforward. When each prime implicant is examined, a check is carried out to see if a prime implicant contains a literal representing a failure node in state 0. If it does, the prime implicant contributes to the DFV of the corresponding time step and earlier time steps. The DFV for state 0 of a failure node is approximated with the following formula:

$$I^{DFV}(i_{f=0}(-t)) = \frac{1 - \prod_{u \in U_{f=0}(-t)}(1 - P_u)}{Q_{TOP}}, \tag{4.22}$$

where $U_{f=0}(-t)$ is the set of prime implicants that contain a literal representing the state 0 of a failure node f at a time step $-t$ or later and P_u is the probability of a prime implicant u.

The calculation process of the dynamic Fussell-Vesely measure for random nodes is, in general terms, similar to the calculation of the DFV measure for the state 0 of a failure node. When each prime implicant is examined, a check is carried out to see if a prime implicant contains a literal representing a random node in certain state. If it does, the prime implicant contributes to the DFV of the corresponding time step and later time

Table 4.7: The backtracking process to obtain the state probabilities of $A(-1)$ under the conditions set by the prime implicant 6. Notations: $p_0 = P(C(-1) = 0)$ and $p_1 = P(C(-1) = 1)$. Different cases of backtracking process are presented in tables 4.4, 4.5 and 4.6.

Output		Input	State pr.			Case
			0	1	2	
$A(-1)$	$\sim$	$A(-2)$	?	?	?	D1
		$B(-1)$	0	1		F1
		$C(-1)$	p_0	p_1		R3
Result:		probs. of $A(-2)$ needed				
$A(-2)$	$\sim$	$A(-3)$	0	1	0	D2
		$B(-2)$	1	0		F4
		$C(-2)$	0	1		R1
Result:		$A(-2)$	0	0	1	
$A(-1)$	$\sim$	$A(-2)$	0	0	1	D1
		$B(-1)$	0	1		F1
		$C(-1)$	p_0	p_1		R3
Result:		$A(-1)$	0	0	1	

steps. If the state of the random node appears more than once in the prime implicant, only the first time step is relevant. If the initial state of the random node is defined, literals representing the random node at the initial time step or earlier are ignored in this calculation process. The dynamic Fussell-Vesely for a state of a random node is approximated using the following formula:

$$I^{DFV}(r_s(-t)) = \frac{1 - \prod_{k \in K_{r=s}(-t)}(1 - P_k)}{Q_{TOP}}, \tag{4.23}$$

where $K_{r=s}(-t)$ is the set of prime implicants that contain a literal representing a state s of a random node r at a time step $-t$ or earlier, if the initial state of the random node r is left open, and P_k is the probability of a prime implicant k. If the initial state of the random node r is defined, $K_{r=s}(-t)$ is the set of prime implicants that contain a literal representing a state s of the random node r at a time step $-t$ or earlier time steps after the initial time step $-n$.

The Fussell-Vesely for deterministic nodes is calculated in a similar way to the dynamic Fussell-Vesely measure for random nodes, except that time steps do not have to be considered.

4.6 Implementation of the Dynamic Risk Increase Factor

The dynamic risk increase factor for a component failure presented in equation (4.9) was defined so that a component is failed at the initial time. However, In YADRAT, a failure cannot occur at the initial time. Thus, the first possibility for a failure to occur is at the first time step after the initial time. Hence, in the calculation of the dynamic risk increase factor, it must be assumed that the failure occurs at the first time step after the initial time instead of at the initial time step. Let the dynamic risk increase factor for a component failure be taken into account first without considering in which state the component is stuck. To obtain a top event probability $Q_{TOP}(f(-n+1) = 1)$, a new set of prime implicants is derived from the original prime implicants with the following rules:

1. If a prime implicant contains a literal representing the failure node in state 1 at some time step, the literal is removed from the prime implicant because its probability is 1.

2. If a prime implicant contains a literal representing the failure node in state 0 at some time step, the prime implicant is removed from the set of prime implicants.

After these two rules are applied to the set of original prime implicants, a new set might contain duplicate prime implicants and implicants that are not prime implicants

under the failure assumption. Those needless implicants must be removed. First, those implicants that had literals removed are systematically compared to each other and all implicants that are not prime implicants and duplicate prime implicants are removed. Second, those implicants that had literals removed and survived from the previous step are compared to the implicants that remained untouched. Again, all implicants that are not prime implicants are removed. After this, the probability $Q_{TOP}(f(-n+1)=1)$ can be calculated from the new set of prime implicants.

The calculation of the dynamic risk increase factor for a failure state of a component presented in equation (4.8) is more complicated. It must be assumed that a component fails in a certain state at the first time step after the initial time. The failure state might depend on states of other nodes at previous time steps (timed nodes). Thus, first, it must be detected which timed nodes affect the failure state. This is done by applying the following steps starting from step 1. Again, it is important to keep a tally of time steps at which nodes are considered.

1. From the decision table of the component, check if the component always fails to the same state. If it does, a failure state does not depend on any node. If there are different failure states, go to step 2.

2. From the decision table of the component, check if one of inputs defines the output state under the failure condition. If there is a defining node, go to step 3. If not, the failure is not such that the component is stuck in one of its states and the dynamic risk increase factor cannot be calculated for a failure state.

3. If the defining node is a random node or a failure node, the failure is not such that the component is stuck in one of its states and the dynamic risk increase factor cannot be calculated for a failure state. If the defining node is a deterministic node, go to step 4.

4. If the defining node is the considered component, go to step 5. If not, check if the state of the node is defined by the considered component. If it is, go to step 5. If not, the failure is not such that the component is stuck in one of its states and the dynamic risk increase factor cannot be calculated for a failure state.

5. If the failure state is defined by a state of a node at a time step earlier than the initial time, a failure state depends only on this node at the initial time (because the system is at steady state before the initial time). In this case, if the initial state of the defining node is left open, add the defining node at the initial time to the list of

timed nodes that affect the failure state and end the procedure. If the failure state is defined by a state of a node at the initial time, go to step 6. If not, go to step 7.

6. If at least one of the time lags of the input nodes of the node that defines the failure state is not 0, a failure state depends only on the defining node at the initial time. In this case, if the initial state of the defining node is left open, add the defining node at the initial time to the list of timed nodes that affect the failure state and end the procedure. If all the time lags are 0, go to step 7.

7. Examine all the input nodes of the deterministic node, detect their time lags and write down the new time step for each input node. If an input node is a random node, go to step 10. If an input node is a deterministic node, go to step 8. If an input node is a failure node and its time step is some time step after the initial time step, add the corresponding timed node to the list of timed nodes that affect the failure state. If not, ignore it.

8. If the time step of the deterministic node is the initial time step, go to step 9. If the time step is a time step after the initial time, go to step 7. If the time step is an earlier time step than the initial time step, add the deterministic node at the initial time to the list of timed nodes that affect the failure state if its initial state is left open.

9. If at least one of the lags of the input nodes of the deterministic node is not 0, add the deterministic node at the initial time to the list of timed nodes that affect the failure state if its initial state is left open. If all the lags are 0 go to step 7.

10. If the time step of the random node is a time step after the initial time, add the corresponding timed node to the list of timed nodes that affect the failure state. If the time step of the random node is the initial time step or earlier, add the random node at the initial time step to the list of timed nodes that affect the failure state if its initial state is left open.

Table 4.8 presents an example of a component whose failure state does not depend on any node because it always fails in the same state.

If the list of the timed nodes that affect the failure state is empty, the component can only fail in one state at the first time step after the initial time. In this case, the dynamic risk increase factor can only be calculated for this failure state and the calculation process is similar to the case of the dynamic risk increase factor for a component failure. However, if the list is not empty, different state combinations of its timed nodes might lead to different failure states. As the list contains all the timed nodes that affect the failure

Table 4.8: The decision table of a component E that always fails in the same state.

	Output	Inputs	
Node	E	E	EF
Time lag	0	-1	0
	1	0	0
	1	0	1
	0	1	0
	1	1	1

state, each state combination can lead to only one failure state. Let a vector $\mathbf{X}_i$ contain variables represented by those timed nodes that affect the failure state of a component i. For each state combination $\mathbf{c}$, the following four tasks are performed:

1. The probability $P(\mathbf{X}_i = \mathbf{c})$ of a state combination $\mathbf{c}$ is calculated as a product of the state probabilities.

2. The failure state s of a component i is calculated by backtracking the model as in the calculation of the dynamic Fussell-Vesely.

3. New prime implicants for the top event are calculated under the conditions of the state combination $(\mathbf{X}_i = \mathbf{c})$ and the failure.

4. The top event probability $Q_{TOP}(f(-n+1) = 1, i(-n+1+l) = s, \mathbf{X}_i = \mathbf{c})$ is calculated from prime implicants.

In task 2, a failure state of a component is calculated by backtracking the model with the same rules that were used in the calculation of the dynamic Fussell-Vesely. However, in the calculation of the dynamic Fussell-Vesely, the backtracking was performed under conditions set by a prime implicant. Now, in this case, no prime implicant is used. Instead, the backtracking is performed under the conditions of the state combination and the failure at the first time step after the initial time. In the result, one of the states has a probability 1 and others 0.

In task 3, the original prime implicants are treated with the following rules:

1. If a prime implicant contains a literal representing the failure node in state 1, the literal is removed from the prime implicant because its probability is 1.

2. If a prime implicant contains a literal representing the failure node in state 0, the prime implicant is removed from the set of prime implicants.

3. If a prime implicant contains a literal representing a timed node from the list of timed nodes that affect the failure state with the state that is in the state combination, the literal is removed from the prime implicant because its probability is 1.

4. If a prime implicant contains a literal representing a timed node from the list of timed nodes that affect the failure state with a state that is not in the state combination, the prime implicant is removed from the set of prime implicants.

Again, after applying the rules, the set of prime implicants might contain duplicate prime implicants and implicants that are not prime implicants. All irrelevant implicants and duplicate prime implicants are removed.

When a new top event probability has been calculated for each state combination $\mathbf{c}$, conditional top event probabilities $Q_{TOP}(f(-n+1)=1, i(-n+1+l)=s)$ are calculated as weighted sums of these top event probabilities with state combination probabilities used as weights for each state s:

$$\begin{aligned} &Q_{TOP}(f(-n+1)=1, i(-n+1+l)=s) \\ &= \sum_{\mathbf{c}\in C(s)} P(\mathbf{X}_i=\mathbf{c}) \cdot Q_{TOP}(f(-n+1)=1, i(-n+1+l)=s, \mathbf{X}_i=\mathbf{c}), \end{aligned} \tag{4.24}$$

where $C(s)$ is a set of state combinations that lead to a failure state s.

Let the example of decision table 3.1 be continued by calculating the dynamic risk increase factor for the component A. New prime implicants for the top event must be calculated under the assumption that the node B is in state 1 at time step -2. All the prime implicants, except the prime implicant 8 ($\{A(-3)=2, B(-2)=1\}$), contain the failure node B in state 0. Thus, the prime implicant 8 is the only prime implicant that should not be removed from the set of prime implicants. Also, literal $B(-2)=1$ must be removed from the prime implicant 8. Thus, the only prime implicant for the conditional top event is $\{A(-3)=2\}$ and

$$I^{DI}(A) = \frac{P(A(-3)=2)}{Q_{TOP}}. \tag{4.25}$$

Now, let the dynamic risk increase factor for the failure states of the component A be considered. First, it must be solved which timed nodes affect the state of the node A at time step -2. It is clear that there are more than one possible failure states. The timed node $A(-3)$ defines the failure state at time step -2. Time step -3 is the initial time step and the node A depends on itself with a time lag -1. Hence, $A(-3)$ is the only timed node that affects the failure state at time step -2. According to decision table 3.1, the failure state is always the state of $A(-3)$.

The next step is to solve prime implicants for the top event assuming that the failure occurs at time step -2 and the initial state of the node A is 0, 1 or 2. Again, all the prime implicants, except the prime implicant 8, are removed and the literal $B(-2) = 1$ is removed from the prime implicant too. However, in the calculation of the dynamic risk increase factor for the state 2, also the literal $A(-3) = 2$ must be removed from the prime implicant 8. Thus, only one empty prime implicant is left. This means that the conditional top event probability is 1 and

$$I^{DI}(A_2) = \frac{1}{Q_{TOP}}. \tag{4.26}$$

Thus, if the node A is stuck in state 2 at the time step -2, the top event occurs with certainty.

In the calculation of the dynamic risk increase factor for states 0 and 1, the prime implicant 8 must also be removed because it contains $A(-3)$ in a wrong state. Thus, the top event cannot occur if the component A fails in state 0 or 1 at the first time step after the initial time.

Calculating the dynamic risk increase factor for a random node is more straightforward. If the initial state is left open for a random node, it is assumed that a random node is in a certain state at every time step. If a random node is defined to be in a different state at the initial time step, it is assumed that a random node is in a certain state at every time step after the initial time step. Prime implicants for the top event under these assumptions are derived from the original prime implicants with the following rules:

1. If a prime implicant contains a literal representing the random node with the right state at a time step later than the initial time, the literal is removed from the prime implicant because its probability is 1.

2. If a prime implicant contains a literal representing the random node with a wrong state at a time step later than the initial time, the prime implicant is removed from the set of prime implicants.

3. If a prime implicant contains a literal representing the random node at the initial time step or earlier and the initial state is defined, the literal is removed from the prime implicant because its probability is 1.

4. If a prime implicant contains a literal representing the random node in the right state at the initial time step or earlier and the initial state is left open, the literal is removed from the prime implicant because its probability is 1.

5. If a prime implicant contains a literal representing the random node in a wrong state at the initial time step or earlier and the initial state is left open, the prime implicant is removed from the set of prime implicants.

After applying these rules, all implicants that are not prime implicants and duplicate prime implicants are removed by systematically comparing implicants to each other. The conditional top event probability $Q_{TOP}(r(-t) = s, \ \forall \ t \in \{0, 1, ..., n-1, n\})$ can be calculated from the new prime implicants and the dynamic risk increase factor is obtained when it is divided by the original top event probability Q_{TOP}.

In the calculation of the risk increase factor for an initial state of a deterministic node, prime implicants for the top event with an assumption, that the initial state of the deterministic node is a certain state, need to be identified. These prime implicants are derived from the original prime implicants with two rules:

1. If a prime implicant contains a literal representing the deterministic node with the right state, the literal is removed from the prime implicant because its probability is 1.

2. If a prime implicant contains a literal representing the deterministic node with a wrong state, the prime implicant is removed from the set of prime implicants.

After applying these rules, new prime implicants are obtained by removing all implicants that are not prime implicants and duplicate prime implicants from the list. The conditional top event probability can be calculated from these new prime implicants.

Chapter 5

Common Cause Failures in YADRAT

5.1 Dynamic Common Cause Failure Models

In the field of dynamic reliability analysis, some research on common cause failures [42] [43] [44] can be found. CCF modelling in Markov models [42] is difficult to apply in DFM because of the fundamental differences of these approaches [11]. Some other dynamic common cause failure modelling has also been performed, for example, in dynamic hierarchical systems [43] and GO-Flow methodology [44].

In fault tree analysis, a common cause failure means that a group of components fails at some time interval due to a common cause. Exact failure times are not considered. DFM provides a possibility for making the common cause failure analysis more accurate as the time interval between the initial time $(-n)$ and the time step of a top event (0) has been divided into smaller time intervals by time steps. In DFM, the possibility that failures may occur at different time steps due to a common cause can be taken into account. A common cause failure should always be associated with a combination of time steps at which the components fail.

When parametric models are applied to common cause failures in DFM, it must be noted that CCFs can occur with different time step combinations. The problem can be addressed with similar β and α parameters as in the fault tree analysis. In the β-factor model, β is the probability that a component fails due to a common cause given that it fails. In DFM, there are several different CCF events that include a failure of a component at a certain time step. Now, let each CCF event with a combination of failure times have its own parameter $\beta(-t_1, -t_2, ..., -t_m)$ that depends on failure times $-t_1, -t_2, ..., -t_m$, when m is the number of components in the CCF group. Let $P(F(-t) = 1)$ be the probability

that a component fails at a time step $-t$. The probability that a component fails at a time step $-t$ due to a CCF event should be $\beta \cdot P(F(-t) = 1)$. When $-n$ is the initial time, there are n^{m-1} different failure time step combinations that include a failure of a certain component at a certain time step. Thus, assuming that the i:th component fails at a time step $-t$, the probability that it fails due to a CCF event is

$$\beta = \sum_{t_1=0}^{n-1} \cdots \sum_{t_{i-1}=0}^{n-1} \sum_{t_{i+1}=0}^{n-1} \cdots \sum_{t_m=0}^{n-1} \beta(-t_1, ..., -t_{i-1}, -t, -t_{i+1}, -t_m). \tag{5.1}$$

The question of how a parameter $\beta(-t_1, -t_2, ..., -t_m)$ should be estimated is an open research problem. In this thesis, it assumed that all failure time combinations are equally probable meaning that

$$\beta(-t_1, -t_2, ..., -t_m) = \frac{1}{n^{m-1}}\beta \tag{5.2}$$

for all $(-t_1, -t_2, ..., -t_m) \in \{-n+1, ..., 0\}^m$.

The parametrisation of the α-factor model can be carried out in a similar way. The only difference is that β is replaced with $\frac{k}{\binom{m-1}{k-1}}\frac{\alpha_k}{\alpha_{tot}}$.

If a non-constant reliability model was used meaning that failure probability would be different at different time steps, the CCF model parametrisations would become more complex. The models might also be more realistic, if the parameter β were to depend on time, but there is no research data for estimation of such parameters yet.

5.2 Implementation of Common Cause Failure Models

Common cause failure models could be implemented in YADRAT by attaching common cause failure nodes to a model, but there is a more efficient way of doing it. All prime implicants that include CCFs can be constructed based on those prime implicant that contain only independent failures. Thus, prime implicants can be determined first without considering common cause failures and those prime implicants that include CCFs can be added in the post-processing phase.

After the set of prime implicant has been identified without considering common cause failures, the set of prime implicants is examined separately with each CCF group. In the case of the α-factor model, prime implicants are examined separately with each combination that contains at least two components. For each CCF, all prime implicants are gone through one by one. There are three different cases of how prime implicants are treated:

1. If a prime implicant contains failures of all components in the CCF group, a new prime implicant based on it is created. This new prime implicant is similar except that the failures of the components in the group are replaced by a CCF with a time step combination of the failure times in the original prime implicant.

2. If a prime implicant contains only failures of some components in the group but not all, a set of new prime implicants based on it is created. New prime implicants are similar to the original prime implicant except that the failures of the components in the group are replaced by CCFs. Time step combinations of these CCFs include the same failure time steps as the failures in the original prime implicant. However, time steps of those failures that were not in the original prime implicant differ. For a failure that was not in the original prime implicant, each time step after the initial time is possible, unless the failure node appears in the original prime implicant in state 0. In that case, possible failure times are limited to the time steps later than the time step of the literal representing the failure node in state 0. A prime implicant with a CCF is created for each possible failure time step combination.

3. If a prime implicant does not contain a failure of any component in the CCF group, no new prime implicants are constructed.

After the prime implicants have been examined with a certain CCF group, all new prime implicants are systematically compared to each other and all duplicate prime implicants and implicants that are not prime implicants are removed from the set.

All the new prime implicants that are created are also examined similarly with other CCFs so that there can also be prime implicants with more than one common cause failure.

In prime implicants, CCFs are presented so that the time step of a literal representing a CCF is the earliest time step in the failure time step combination. A time lag combination is attached to the name of each CCF event. Together, the time step of the literal and the time lag combination imply the failure time step combination as the failure time of a component is the time step of the literal plus the corresponding time lag.

Let F_1 and F_2 be failure nodes that are associated with a β-factor group. Let a set of literals $\{A(-3) = 1, B(-3) = 0, F_1(-1) = 1, F_2(-2) = 1\}$ be a prime implicant. A new prime implicant based on this prime implicant must be created. This new prime implicant contains a CCF literal $F^{1,0}(-2) = 1$ instead of literals representing independent failures: $\{A(-3) = 1, B(-3) = 0, F^{1,0}(-2) = 1\}$. The probability of this CCF event is $P(F^{1,0}(-2) = 1) = \frac{1}{3}\beta \cdot P(F_1(-1) = 1) = \frac{1}{3}\beta \cdot P(F_2(-2) = 1)$ if the initial time is -3.

Now, let $\{C(-3) = 0, D(-3) = 2, F_1(-1) = 1, E(-3) = 2\}$ be a prime implicant. New prime implicants based on this prime implicant must be created as well. In those new prime implicants, a CCF event must include the failure of the first component at time step -1. However, the failure time of the second component can be -2, -1 or 0. Thus, three new prime implicants are created and they are presented in table 5.1. All these CCF events have the same probability $\frac{1}{3}\beta \cdot P(F_1(-1) = 1)$.

Let $\{B(-3) = 2, C(-3) = 1, F_1(-1) = 1, F_2(-2) = 0\}$ be a prime implicant. Again, new prime implicants based on this prime implicant must be created. In those new prime implicants, a CCF event must include the failure of the first component at time step -1. The second component cannot fail at time step -2 but it can fail at time steps -1 or 0. Thus, two new prime implicants are created. They are presented in table 5.2.

Let V_1, V_2 and V_3 be failure nodes that are associated with a α-factor group. Let $\{Z(-3) = 0, V_1(-1) = 1, V_2(-2) = 1\}$ be a prime implicant. This prime implicant must be considered with each combination of V_1, V_2 and V_3 that contains at least two failure nodes. All the prime implicants that are created based on this prime implicant are presented in table 5.3. The probability of a CCF event that includes failures of all three components would be $\frac{1}{3^2} \cdot 3 \cdot \frac{\alpha_3}{\alpha_{tot}} \cdot P(V_1(-1) = 1)$.

5.3 Common Cause Failures and Risk Importance Measures

If a model includes common cause failures, they must be taken into account in the calculation of risk importance measures. The calculation of risk importance measures for common cause events does not differ much from the case of failure events. However, when the failure states of components included in a common cause group are considered, the calculation process is more complex. In that case, risk importance measures are calculated for different failure state combinations of components in a common cause group.

The calculation of the dynamic Fussell-Vesely is largely similar to the case of an independent component failure. Anyhow, the failure state must be solved for each component included in a common cause group. To obtain the starting time step for the backtracking process, the time lag of the failure node and the time lag of the failure event in the common cause group are added to the time step of the CCF literal.

When a failure node is considered in the backtracking process, the possibility of a common cause failure must be taken into account. If the prime implicant includes a literal representing a common cause failure which includes a failure of the component that has the considered failure node as an input node, the time step is checked to see if the failure has already occurred or not. The failure time is obtained by adding the time lag of the failure event in the CCF literal to the time step of the CCF literal. If the considered time step is earlier than the failure time, the failure node is in state 0 at the considered time step. If the considered time step is the same or later than the failure time, the failure node is in state 1 at the considered time step. If the prime implicant does not include a literal representing a common cause failure which includes a failure of the component that has the considered failure node as an input node, the rules F1-F6 presented in table 4.4 are applied.

Table 5.1: Prime implicants with CCF literals created based on the prime implicant $\{C(-3) = 0, D(-3) = 2, F_1(-1) = 1, E(-3) = 2\}$.

	Prime implicant
1	$\{C(-3) = 0, D(-3) = 2, F^{1,0}(-2) = 1, E(-3) = 2\}$
2	$\{C(-3) = 0, D(-3) = 2, F^{0,0}(-1) = 1, E(-3) = 2\}$
3	$\{C(-3) = 0, D(-3) = 2, F^{0,1}(-1) = 1, E(-3) = 2\}$

Table 5.2: Prime implicants with CCF literals created based on the prime implicant $\{B(-3) = 2, C(-3) = 1, F_1(-1) = 1, F_2(-2) = 0\}$.

	Prime implicant
1	$\{B(-3) = 2, C(-3) = 1, F^{0,0}(-1) = 1, F_2(-2) = 0\}$
2	$\{B(-3) = 2, C(-3) = 1, F^{0,1}(-1) = 1, F_2(-2) = 0\}$

Table 5.3: Prime implicants with CCF literals created based on the prime implicant $\{Z(-3) = 0, V_1(-1) = 1, V_2(-2) = 1\}$.

Combination	Prime implicant
V_1 and V_2	$\{Z(-3) = 0, V_{1,2}^{1,0}(-2) = 1\}$
V_1 and V_3	$\{Z(-3) = 0, V_{1,3}^{1,0}(-2) = 1, V_2(-2) = 1\}$
	$\{Z(-3) = 0, V_{1,3}^{0,0}(-1) = 1, V_2(-2) = 1\}$
	$\{Z(-3) = 0, V_{1,3}^{0,1}(-1) = 1, V_2(-2) = 1\}$
V_2 and V_3	$\{Z(-3) = 0, V_1(-1) = 1, V_{2,3}^{0,0}(-2) = 1\}$
	$\{Z(-3) = 0, V_1(-1) = 1, V_{2,3}^{0,1}(-2) = 1\}$
	$\{Z(-3) = 0, V_1(-1) = 1, V_{2,3}^{0,2}(-2) = 1\}$
V_1, V_2 and V_3	$\{Z(-3) = 0, V^{1,0,0}(-2) = 1\}$
	$\{Z(-3) = 0, V^{1,0,1}(-2) = 1\}$
	$\{Z(-3) = 0, V^{1,0,2}(-2) = 1\}$

When the dynamic Fussell-Vesely is calculated for a CCF event, the time step of the CCF event is assumed to be the time step of the first failure. Other failure times are considered only when the failure state probabilities are solved.

When CCFs are included in a model, a component can fail independently or due to a common cause. The dynamic Fussell-Vesely measure that takes into account both independent failures and common cause failures can also be calculated. It is calculated as a sum of independent failure DFV and CCF DFVs. However, the CCF DFV used here is a little different from what was described above. Here, the failure time of the considered component is treated as the time step of the CCF event instead of the time step of the first failure.

In calculation of the dynamic risk increase factor for a CCF, it is assumed that all the failures occur at the first time step after the initial time. When the dynamic risk increase factor is calculated for a combination of failure states of components in a CCF group, the first step is to solve which timed nodes affect these failure states. This is done in a similar way to the case of an independent failure as each component failure is considered separately. As a result of this step, a list of timed nodes that affect at least one failure state is obtained. Next, all state combinations of these timed nodes are examined and a solution is achieved as to which failure state combination each state combination of timed nodes leads. New prime implicants for the conditional top event are identified by applying rules that are similar to the case of an independent failure except that those prime implicants that contain literals representing failure nodes from the considered CCF group and those prime implicants that contain literals representing CCFs of the same group with time lag combinations in which all the lags are not 0 are removed. Finally, the dynamic risk increase factor for a combination of failure states of components in a CCF group is obtained by calculating a weighted sum as in (4.24).

In addition to the DRIF for a CCF and an independent failure, it might also be beneficial to calculate the dynamic risk increase factor that takes into account both the possibility of a common cause failure and the possibility of an independent failure. This total dynamic risk increase factor (TDRIF) is calculated as a weighted sum of CCF DRIFs and DRIF of the independent failure. The weights come directly from the parametrisation of the CCF model. For example, for the β-factor model, $TDRIF = \beta \cdot DRIF_{CCF} + (1 - \beta) \cdot DRIF_{ind}$.

Chapter 6

A Feed Water Tank System

6.1 The Example System

A feed water tank system [45] is presented in Figure 6.1. In the feed water tank system, the water flow to the tank is constant. A control valve controls how much water is released out of the tank. The valve is operated based on the values of the water level measurement. The goal is that the water level does neither get too high nor too low.

6.2 A YADRAT Model

A YADRAT model based on the feed water tank system is presented in Figure 6.2 and tables 6.1, 6.2 and 6.3. The model contains two components: a valve V and a sensor WLM that measures the water level. The valve can be stuck in state 1 (open) or 0 (close). The water level measurement can be frozen in state -1 (low), 0 (middle) or 1 (high).

The initial time of the analysis is $-n = -3$ and the top event is that the water level

Table 6.1: The decision table of the tank water level WL.

	Output	Inputs	
Node	WL	V	WL
Time lag	0	-1	-1
	0	0	-1
	1	0	0
	1	0	1
	-1	1	-1
	-1	1	0
	0	1	1

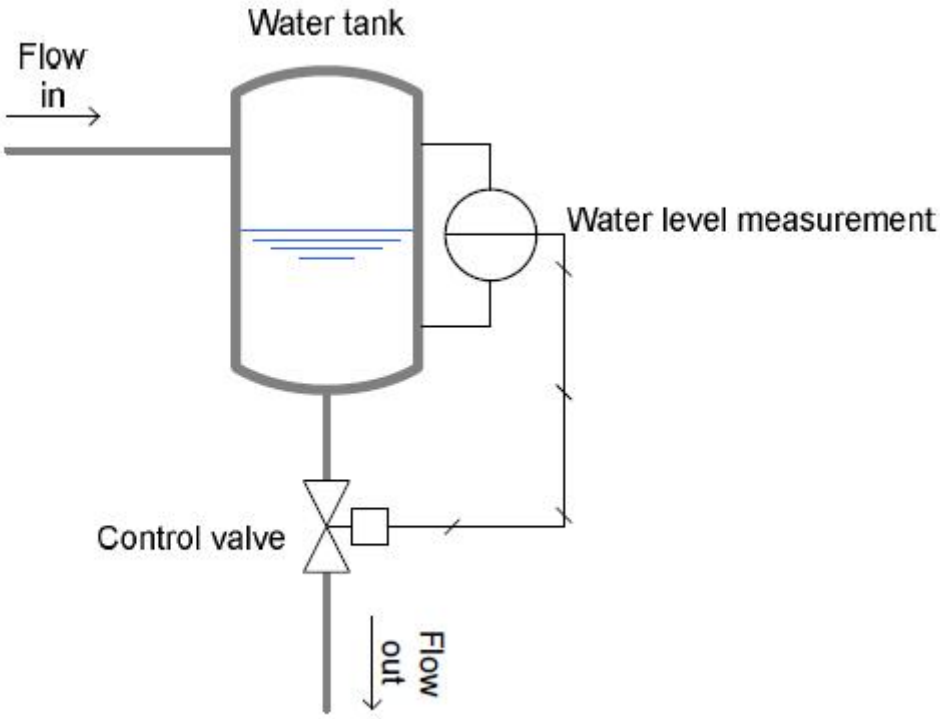

Figure 6.1: A feed water tank system.

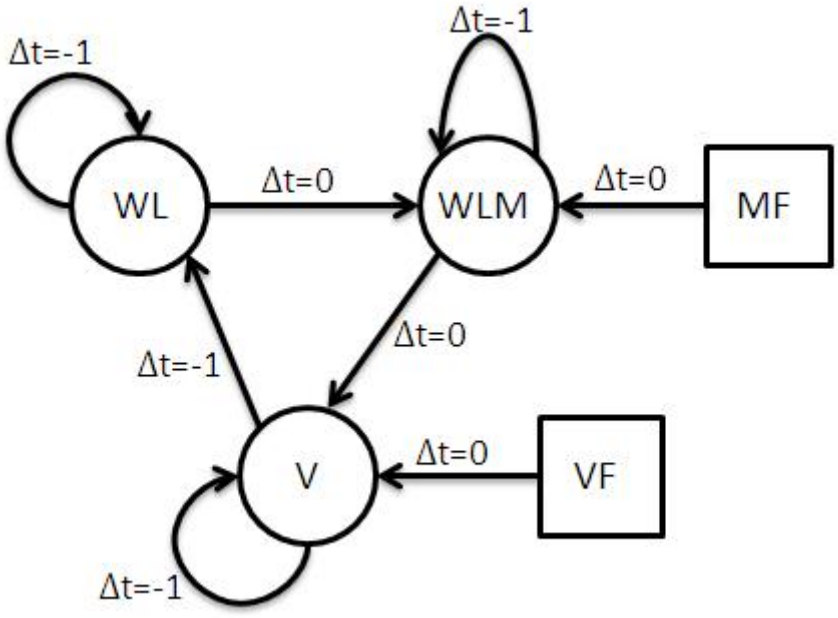

Figure 6.2: A YADRAT model based on the feed water tank system.

Table 6.2: The decision table of the tank water level measurement WLM.

	Output	Inputs		
Node	WLM	MF	WLM	WL
Time lag	0	0	−1	0
	−1	0	−1	−1
	0	0	−1	0
	1	0	−1	1
	−1	0	0	−1
	0	0	0	0
	1	0	0	1
	−1	0	1	−1
	0	0	1	0
	1	0	1	1
	−1	1	−1	−1
	−1	1	−1	0
	−1	1	−1	1
	0	1	0	−1
	0	1	0	0
	0	1	0	1
	1	1	1	−1
	1	1	1	0
	1	1	1	1

Table 6.3: The decision table of the valve V.

	Output	Inputs		
Node	V	VF	WLM	V
Time lag	0	0	0	−1
	0	0	−1	0
	0	0	−1	1
	0	0	0	0
	1	0	0	1
	1	0	1	0
	1	0	1	1
	0	1	−1	0
	1	1	−1	1
	0	1	0	0
	1	1	0	1
	0	1	1	0
	1	1	1	1

node WL is in state 1 (high) at time steps −1 and 0. The initial states of the deterministic nodes are left open with the probabilities that are presented in table 6.4. The valve fails in time unit with a probability of 0.1 and the water level measurement fails in time unit with a probability of 0.05.

6.3 Results

Both the complete cover and an irredundant cover were determined for the top event $\{WL(-1) = 1, WL(0) = 1\}$. The complete cover contains ten prime implicants that are presented in Figure 6.3. An irredundant cover consists of the first six of those prime implicants.

The complete cover gave an approximation 0.0873 for the top event probability, while the approximation calculated from the irredundant was 0.0751. As the number of prime implicants is so small, the accurate top event probability is relatively easy to calculate with help of a tree diagram. A tree diagram that is built of prime implicants is presented in Figure 6.4. All the paths of the tree are mutually exclusive. Hence, the top event probability can be calculated as a sum of path probabilities. The accurate top event probability is 0.0740.

Accurate risk importance measure values can also be calculated with help of tree

Num	Tot Prob	%	Prob		Variable	Time	Value
1	4,00E-02	45,80 %	4,00E-01		V	-3	0
			1,00E-01		VF	-2	1
2	1,25E-02	14,36 %	4,00E-01		V	-3	0
			3,30E-01		WL	-3	-1
			9,50E-01		MF	-2	0
			1,00E-01		VF	-1	1
3	6,80E-03	7,79 %	4,00E-01		V	-3	0
			3,40E-01		WLM	-3	0
			5,00E-02		MF	-2	1
4	6,60E-03	7,56 %	4,00E-01		V	-3	0
			3,30E-01		WLM	-3	-1
			5,00E-02		MF	-2	1
5	6,27E-03	7,18 %	4,00E-01		V	-3	0
			3,30E-01		WL	-3	-1
			9,50E-01		MF	-2	0
			5,00E-02		MF	-1	1
6	4,90E-03	5,61 %	9,00E-01		VF	-2	0
			3,30E-01		WL	-3	1
			3,30E-01		WLM	-3	-1
			5,00E-02		MF	-2	1
7	4,49E-03	5,14 %	4,00E-01		V	-3	0
			3,30E-01		WL	-3	-1
			3,40E-01		WLM	-3	0
			1,00E-01		VF	-1	1
8	4,36E-03	4,99 %	4,00E-01		V	-3	0
			3,30E-01		WL	-3	-1
			3,30E-01		WLM	-3	-1
			1,00E-01		VF	-1	1
9	2,24E-03	2,57 %	4,00E-01		V	-3	0
			3,30E-01		WL	-3	-1
			3,40E-01		WLM	-3	0
			5,00E-02		MF	-1	1
10	2,18E-03	2,49 %	4,00E-01		V	-3	0
			3,30E-01		WL	-3	-1
			3,30E-01		WLM	-3	-1
			5,00E-02		MF	-1	1

Figure 6.3: The complete cover of the top event $\{WL(-1) = 1, WL(0) = 1\}$.

Table 6.4: The initial probabilities of the deterministic nodes of the feed water tank system model.

Node	Initial probabilities		
V	0: 0.4	1: 0.6	
WL	−1: 0.33	0: 0.34	1: 0.33
WLM	−1: 0.33	0: 0.34	1: 0.33

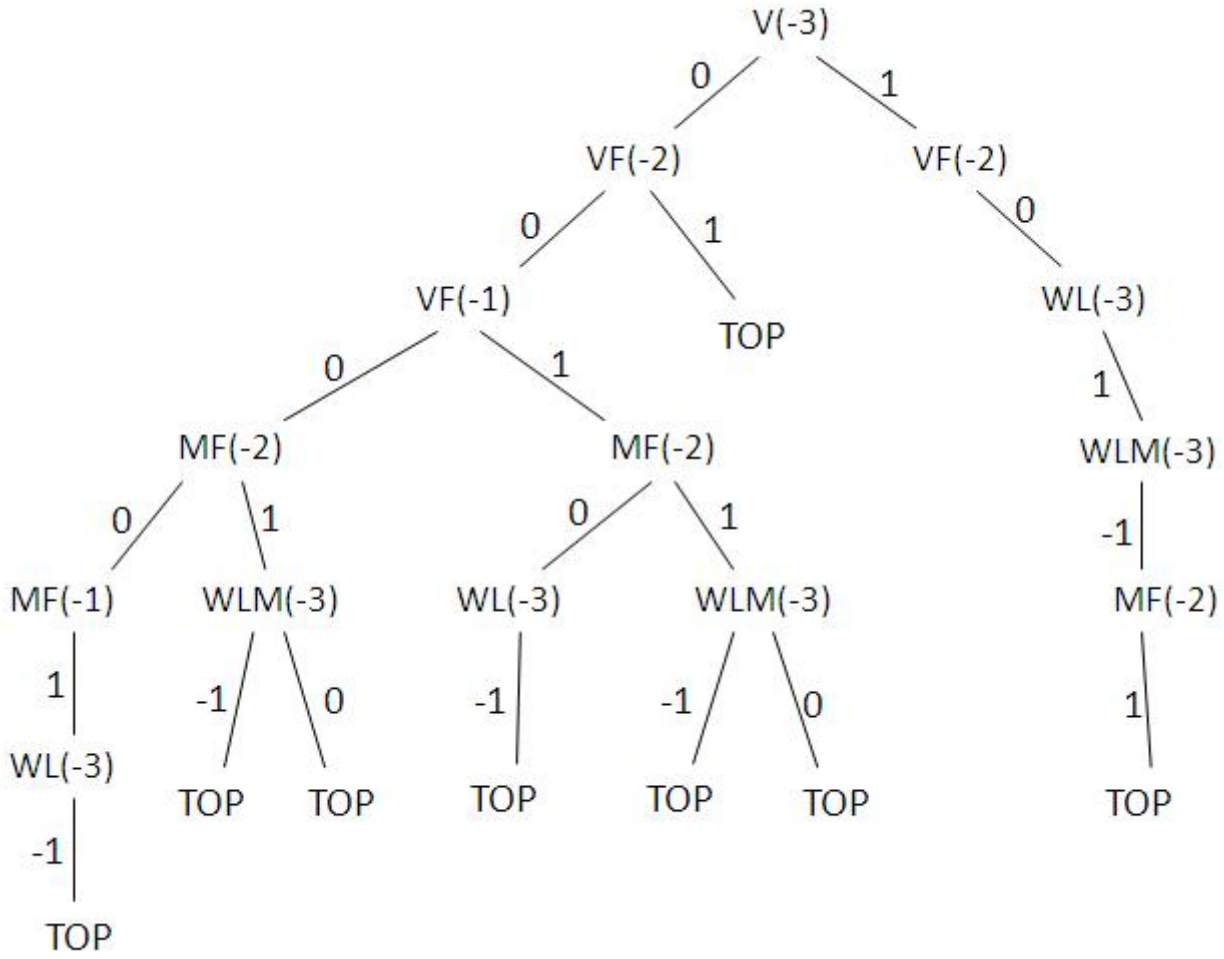

Figure 6.4: A tree diagram built of prime implicants. All the paths lead to the top event. This same diagram can be constructed both from the complete cover and an irredundant cover. The number of paths does not need to be same as the number of prime implicants. Some paths can satisfy multiple prime implicants. For each prime implicant, there is at least one path that satisfies it.

diagrams. The results are presented in figures 6.5 and 6.6. Both dynamic Fussell-Vesely values and dynamic risk increase factor values indicate that the valve is a more important component than the sensor that measures the water level. This is logical because the valve affects the water level directly while the water level measurement affects the water level only via the valve.

In the prime implicants, the valve fails only to state 0, which means that the valve is closed at every time step after the failure and no water flows out of the tank causing the water level to rise. If the valve is stuck in state 0 at time step -2, the top event occurs with certainty. On the other hand, the failure to state 1 cannot cause the water level to rise because the valve is open at every time step. If the valve is stuck in state 1 at time step -2, the top event cannot occur. The prime implicant 6 contains a literal $VF(-2) = 0$ because to the failure would actually prevent the top event from occurring.

The dynamic Fussell-Vesely values indicate that failure states -1 and 0 of the water level measurement contribute to the top event while failure state 1 does not. This is logical because state 1 means that the measurement gives a high value. When the measurement value is high, the valve is opened and the water level lowers. On the other hand, state -1 means that the measurement value is low. In this case, no water is let out of the tank and the water level rises. When the measurement gives a value from the middle level (state 0), nothing is done to the valve which means that the valve stays open or closed. Thus, the failure in state 0 can contribute to the top event if the valve is closed earlier. Some prime implicants contain MF in state 0. This is because the measure failure would cause the valve to stay open which would prevent the top event from occurring.

The risk importance measure values for the initial states might first look surprising as the initial state -1 of the water level and the initial state -1 of the water level measurement contribute to the top event more than other initial states of the same components. The reason for this is that with these initial states, it is more likely that the water level measurement is stuck in state -1 or that the valve fails to state 0 as these initial conditions indicate that the water level needs to be raised by closing the valve.

Each prime implicant contain either the failure of the valve or the failure of the water level measurement. It should be noted that this does not mean that the dynamic Fussell-Vesely values of the valve failure and the measurement failure should sum up to 1 because prime implicants containing different failures are not all mutually exclusive. This can be seen from the tree diagram of figure 6.4. The tree diagram includes two pats leading to the top event that contain failures of the both components.

It is not evident which cover gives more accurate risk importance measure values. In many cases, the value calculated from the irredundant cover is more accurate because the

The dynamic Fussell-Vesely for component failures				
Component	Time	FV (comp)	FV (irre)	FV (accu)
V	0	0,692	0,693	0,709
	-1	0,692	0,693	0,709
	-2	0,458	0,532	0,54
WLM	0	0,328	0,324	0,31
	-1	0,328	0,324	0,31
	-2	0,208	0,242	0,22
The dynamic Fussell-Vesely for 0-states of failure nodes				
Failure Node	Time	FV (comp)	FV (irre)	FV (accu)
VF	-2	0,056	0,065	0,066
MF	-2	0,214	0,249	0,246
The dynamic Fussell-Vesely for initial states of deterministic nodes				
Deterministic node	State	FV (comp)	FV (irre)	FV (accu)
V	0	0,949	0,939	0,96
WL	-1	0,363	0,249	0,305
	1	0,056	0,065	0,066
WLM	-1	0,205	0,153	0,21
	0	0,154	0,091	0,175
The dynamic risk increase factor for component failures				
Component	RIF (comp)	RIF (irre)	RIF (accu)	
V	4,58	5,32	5,4	
WLM	4,01	4,67	4,05	
The dynamic risk increase factor for initial states of deterministic nodes				
Deterministic node	State	RIF (comp)	RIF (irre)	RIF (accu)
V	0	2,26	2,27	2,31
	1	0,06	0,07	0,07
WL	-1	1,62	1,41	1,37
	0	0,6	0,7	0,68
	1	0,77	0,89	0,8
WLM	-1	1,05	1,21	1,12
	0	0,89	1,02	1
	1	0,66	0,77	0,76

Figure 6.5: Accurate risk importance measure values for failure events, random nodes and initial states and values calculated from both the complete and the irredundant cover.

The dynamic Fussell-Vesely for failure states					
Component	State	Time	FV (comp)	FV (irre)	FV (accu)
V	0	0	0,692	0,693	0,709
	0	-1	0,692	0,693	0,709
	0	-2	0,458	0,532	0,54
WLM	-1	0	0,131	0,153	0,129
	-1	-1	0,131	0,153	0,129
	-1	-2	0,131	0,153	0,129
	0	0	0,199	0,173	0,177
	0	-1	0,199	0,173	0,177
	0	-2	0,078	0,091	0,092
The dynamic risk increase factor for failure states					
Component	State	RIF (comp)	RIF (irre)	RIF (accu)	
V	0	TOP Pr. 1	TOP Pr. 1	TOP Pr. 1	
	1	0	0	0	
WLM	-1	6,62	7,7	7,81	
	0	4,58	5,32	5,4	
	1	0,46	0,53	0,54	

Figure 6.6: Accurate risk importance measure values for failure states and values calculated from both the complete and the irredundant cover. "TOP Pr. 1" means that the conditional top event probability is 1.

corresponding top event probability approximation is more accurate. However, the top event probability approximation affects every value in a similar way. Thus, the accuracy of the top event probability approximation does not affect the importance order.

In some cases, dynamic Fussell-Vesely values are overestimated. This is because those prime implicants that contain the considered condition (a literal representing a node at some state) include other common literals and are not mutually exclusive.

The value calculated from the complete cover can be greater than the value calculated from the irredundant cover if some of those prime implicants that were left out of the irredundant cover include the considered condition. For example, all the prime implicants that were left out of the irredundant cover contain a literal $WL(-3) = -1$. This can be seen in the dynamic Fussell-Vesely value.

Dynamic Fussell-Vesely values can also be underestimated if those prime implicants that contain the considered condition are mutually exclusive. In some cases, more accurate values are clearly obtained from the complete cover as the irredundant cover leaves important prime implicants out.

In dynamic risk increase factor calculation, the same prime implicants for the conditional top event are often obtained from both covers. In those cases, dynamic risk increase

factor values calculated from different covers only differ due to the approximations of the original top event probability. Because of this, the quotient of an irredundant cover value and a complete cover value is often the same as the quotient of the top event probability approximation from the complete cover and the top event probability approximation from the irredundant cover. The set of prime implicants of the conditional top event solved from an irredundant cover can never be greater than the set of prime implicants solved from the complete cover. Thus, the quotient of an irredundant cover dynamic risk increase factor value and a complete cover DRIF value can never be larger than the quotient of the top event probability approximation from the complete cover and the top event probability approximation from the irredundant cover. In this case, the upper bound for the quotient of dynamic risk increase factor values is $\frac{0.0873}{0.0751} \approx 1.16$. For example, for component failures, $\frac{5.32}{4.58} \approx 1.16$ and $\frac{4.67}{4.01} \approx 1.16$. The risk increase factor value calculated from the complete cover can also be bigger than the value calculated from an irredundant cover if the irredundant cover left out prime implicants that contained the considered condition. Again, this is seen in the case of $WL(-3) = -1$.

New prime implicants of the conditional top event are not always identified perfectly. When new prime implicants are identified from an irredundant cover, some of the new prime implicants might not be real prime implicants but only implicants. For example, when the dynamic risk increase factor is calculated for the initial state -1 of WLM from the irredundant cover, an implicant $\{V(-3) = 0, WL(-3) = -1, MF(-2) = 0, MF(-1) = 1\}$ is left in the set of new prime implicants. However, this is not really a prime implicant for the top event with the condition $WLM(-3) = -1$. When new prime implicants are solved from the complete cover, a prime implicant $\{V(-3) = 0, WL(-3) = -1, MF(-1) = 1\}$ is obtained from the 10th of the original prime implicants. This is a real prime implicant of the conditional top event while $\{V(-3) = 0, WL(-3) = -1, MF(-2) = 0, MF(-1) = 1\}$ is not. Due to this, the risk increase factor value calculated from the irredundant cover is a little smaller than it is supposed to be.

As with dynamic Fussell-Vesely values, dynamic risk increase factor values can be either overestimated or underestimated. When new prime implicants of the conditional top event probability are contain common literals and are not mutually exclusive, the dynamic risk increase factor value is overestimated. When prime implicants of the conditional top event probability are mutually exclusive, the dynamic risk increase factor value is underestimated.

Chapter 7

An Emergency Core Cooling System

7.1 The Example System

The operational principle of an emergency core cooling system of a boiling water reactor [45] is presented in Figure 7.1. The purpose of this system is to provide adequate water cooling of a reactor core if the ordinary cooling system is not functioning. An on-off control system controls the water level in the pressure vessel by controlling pumps and valves. Sensors measure the water level. The water level can decrease due to evaporation or leagage. If the water level is too low, more water is pumped into the pressure vessel until an upper limit is reached. This cycle is repeated for as long as necessary.

7.2 A YADRAT Model

A YADRAT model based on the emergency core cooling system is presented in figure 7.2. This model contains two pump lines. The water level will not get too low unless both pump lines fail. Both pump lines contain four components: a sensor that measures the water level, a sensor that measures the pressure, a regulation valve and a pump. The pump can only fail in the "off" state which means that it does not pump any water. A valve can be stuck in "close" or "open" states. The water level measurement can be frozen in "low", "medium" or "high" states, while the pressure measurement can be frozen in "low" or "high" states. A pump line also contains a random node that represents a pump leakage signal. It closes the valve and stops the pumping if it turns to the "true" state. Deterministic nodes of the model include a water flow node (WF), a reactor water level node (WL), some clock nodes and some nodes that are links between sensors and the

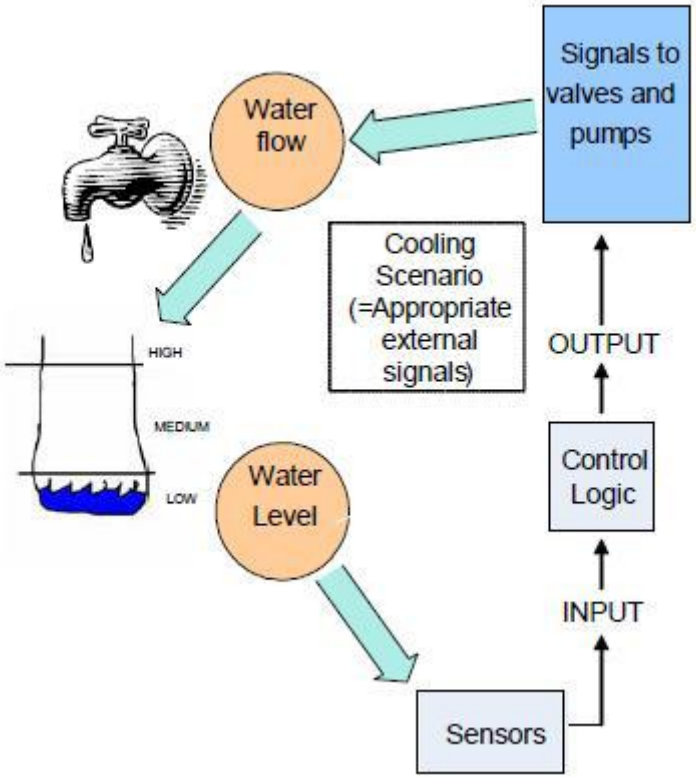

Figure 7.1: An operational principle of an emergency core cooling system of a boiling water reactor.

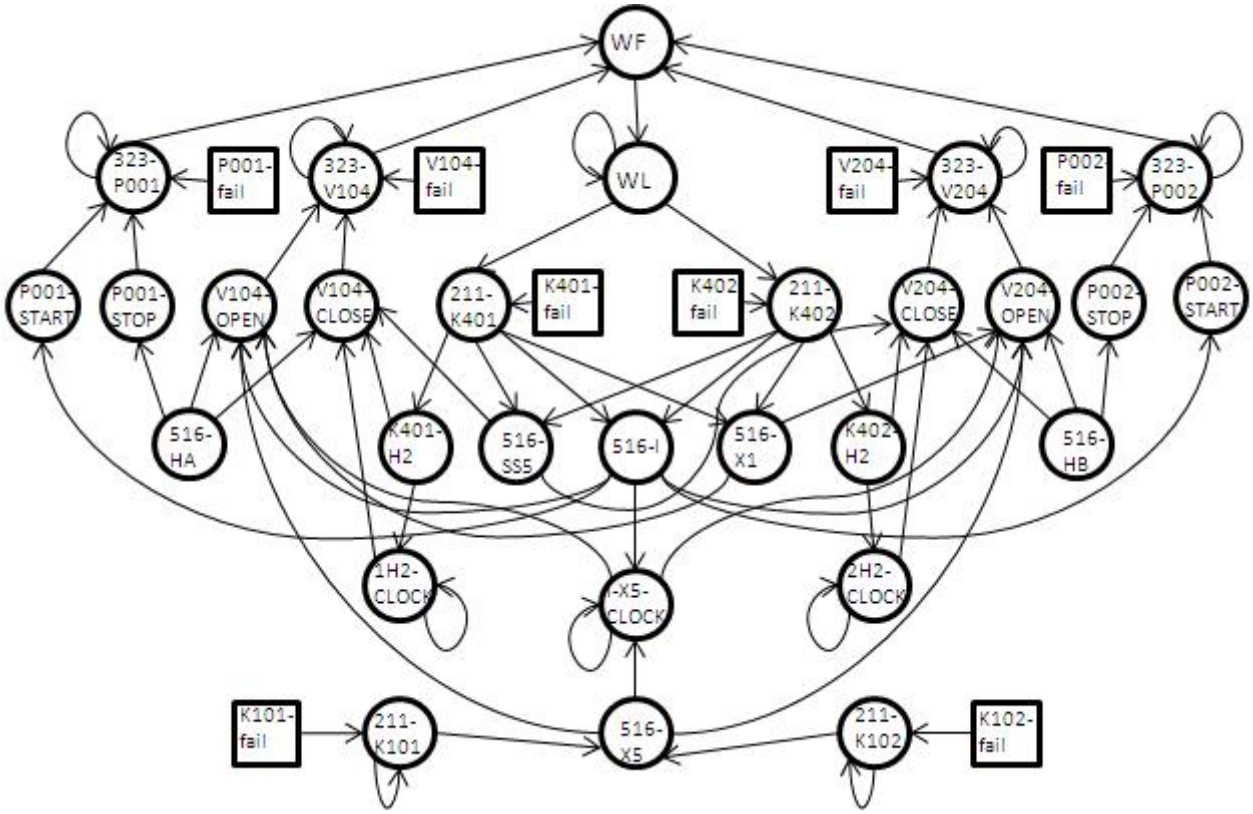

Figure 7.2: A YADRAT model based on the emergency core cooling system.

pump and the valve. More details about the model can be found in [45].

The analysis is performed with the following settings: The initial time of the analysis is -5 and the top event is that the water level node is in the "low" state at time steps -1 and 0. The initial states are defined for all nodes. The model contains eight failure nodes: 323-P001-fail, 323-V104-fail, 211-K401-fail, 211-K101-fail, 323-P002-fail, 323-V204-fail, 211-K402-fail and 211-K102-fail. All the components fail in time unit with a probability 0.05. Random nodes 516-HA and 516-HB are in the "false" state with a probability of 0.9 and in the "true" state with a probability of 0.1.

7.3 Results

Figure 7.3 presents the dynamic Fussell-Vesely values for failure states of the components in the first pump line. The dynamic Fussell-Vesely is also calculated for the state 0 of the failure node 211-K101-fail (0-states of other failure nodes do appear in the prime implicants) and for states of a random node. The results for the second pump line are exactly the same due to symmetry.

The dynamic Fussell-Vesely for failure states			
Component	State	Time	FV
323-P001	off	0	0,331
	off	-1	0,331
	off	-2	0,331
	off	-3	0,321
	off	-4	0,162
323-V104	close	0	0,331
	close	-1	0,331
	close	-2	0,331
	close	-3	0,321
	close	-4	0,162
211-K401	medium	0	0,025
	medium	-1	0,025
	medium	-2	0,025
	medium	-3	0,025
	medium	-4	0,025
211-K101	high	0	0,05
	high	-1	0,05
	high	-2	0,05
	high	-3	0,05
	high	-4	0,05
The dynamic Fussell-Vesely for 0-states of failure nodes			
Failure node	Time	FV	
211-K101-fail	-3	0,473	
	-4	0,473	
The dynamic Fussell-Vesely for random nodes			
Random node	State	Time	FV
516-HA	true	0	0,344
	true	-1	0,344
	true	-2	0,344
	true	-3	0,344
516-HB	true	0	0,344
	true	-1	0,344
	true	-2	0,344
	true	-3	0,344

Figure 7.3: The dynamic Fussell-Vesely for failure states, state 0 of a failure node and states of a random node in the emergency core cooling system.

For each of the components, there is only one failure state that appears in prime implicants. The valve 323-V104 is only stuck in the "close" state in the prime implicants. This is logical because the "open" state cannot really cause the water level to decrease. The water level measurement 211-K401 fails only in the "medium" state in the prime implicants. The measurement state "low" causes more water to be pumped into the pressure vessel. Hence, it cannot be a root cause for the top event. However, it would be logical that the "high" state would also contribute to the top event. The failure state "high" does not appear in the prime implicants because with these initial conditions 211-K401 can only fail in the "high" state if the water level rises to the "high" state at time step -2 or later and if that happens it is already too late for the top event to occur. The pressure measurement 211-K101 fails in the "high" state in the prime implicants. The reason why the failure state "low" does not appear in the prime implicants is the same as with the water level measurement. The pump 323-P001 can only fail in the "off" state.

According to the dynamic Fussell-Vesely, pumps 323-P001 and 323-P002 and valves 323-V104 and 323-V204 are the most important components. It is logical that pumps and valves are more important regarding the top event than the sensors because they directly affect the water flow.

The measurement failures appear in the prime implicants only at the time step -4, while pump and valve failures also appear at time steps -3 and -2. The measurement failures do not appear at time step -3 because the sensors can only fail in "low" state at that time step. The reason why they do not appear at time step -2 is that it is too late for them to affect the water level at time steps -1 and 0.

State 0 of the failure node of the pressure measurement 211-K101 has a high dynamic Fussell-Vesely value at time step -3. This is because with these initial conditions the pressure measurement can only be frozen in the "low" state at that time step and that would cause more water to be pumped into the reactor containment.

The "true" state of the random node 516-HA has the dynamic Fussell-Vesely values of the same level as the pumps and the valves. The "true" state of 516-HA causes pump 323-P001 to stop pumping water and the valve 323-V104 to close, hence the results seem to be consistent.

Figure 7.4 presents the dynamic risk increase factor values for failure states of the components and states of the random nodes in the emergency core cooling system. For each component there is only one possible failure state at the first time step after the initial time because of the initial conditions. The failures of the pumps and valves increase the top event probability substantially more that the measurement failures. The reason is that the failure of a pump or a valve causes the water flow from the pump line to stop

The dynamic risk increase factor for failure states		
Component	State	RIF
323-P001	off	2,89
323-V104	close	2,89
211-K401	medium	1,44
211-K101	high	1,42
323-P002	off	2,89
323-V204	close	2,89
211-K402	medium	1,44
211-K102	high	1,42
The dynamic risk increase factor for random nodes		
Random node	State	RIF
516-HA	true	2,89
	false	0,68
516-HB	true	2,89
	false	0,68

Figure 7.4: The dynamic risk increase factor for failure states and states of random nodes in the emergency core cooling system.

completely. A measurement failure cannot cause that alone. Also, if the random node 516-HA stays in the "true" state throughout the whole analysis time, no water will be pumped in the corresponding pump line. Because of this, the dynamic risk increase factor for the state "true" of 516-HA is the same as for the pumps and valves. If the random node 516-HA remains in the "false" state throughout the whole analysis time, the top event probability decreases.

7.4 Results with Common Cause Failures

Now, let common cause failures be added to the model of the Emergency Core Cooling System. Let there be possible common cause failures for pumps, valves, sensors that measure the water level and sensors that measure the pressure. The CCF models are β-factor models with $\beta = 0.1$.

Due to common cause failures the number of prime implicants increases from 78 to 544. In addition, the top event probability increases from 0.0983 to 0.0992. The large number of prime implicants originates from the fact that there are 25 possible failure time combinations for each CCF. A large portion of prime implicants include such common cause failures in which only one failure really contributes to the top event.

Figure 7.5 presents some prime implicants that include such common cause failures of

Num	Tot Prob	%	Prob		Variable	Time	Value
28	1,00E-03	1,01 %	1,00E-03		323-V04-group#00	-4	true
29	1,00E-03	1,01 %	1,00E-03		323-V04-group#01	-4	true
30	1,00E-03	1,01 %	1,00E-03		323-V04-group#10	-4	true
80	1,00E-04	0,10 %	1,00E-03		323-V04-group#20	-4	true
			1,00E-01		516-HA	-3	true
81	1,00E-04	0,10 %	1,00E-03		323-V04-group#10	-3	true
			1,00E-01		516-HA	-3	true
82	1,00E-04	0,10 %	1,00E-03		323-V04-group#02	-4	true
			1,00E-01		516-HB	-3	true

Figure 7.5: Examples of prime implicants that include such common cause failures of the valves in which both failures are significant. In this example, the "true" state represents the failure.

the valves in which both failures are significant. Figure 7.6 presents some prime implicants that include such common cause failures in which only single failure is significant. In Figure 7.7, some prime implicants that include two common cause failure events are presented. In all the prime implicants that include two common cause failures, CCFs are such that only single failure is significant.

The dynamic Fussell-Vesely results are presented in figures 7.8, 7.9 and 7.10. Common cause failures bring about some changes in the total dynamic Fussell-Vesely values even though independent failures still mostly dominate the system's reliability. Those components that had biggest dynamic Fussell-Vesely values earlier also have biggest dynamic Fussell-Vesely values for CCFs.

Failure state combinations "open-close" and "close-open" of the valves play a small contribution to the top event probability. However, it would seem logical that the "open" state of a valve would rather prevent the top event from occurring. Regardless of this, this type of failure state combination can be a root cause for the top event if some other failure also appears in that pump line that has the valve in the "open" state. The total Fussell-Vesely values reveal that the "open" state appears in the prime implicants only at time steps -2, -1 and 0. If a valve is stuck in the "open" state this late, it cannot prevent the top event from occurring if the pump from the same line has already failed. Thus, when a failure state combination "open-close" appears in a prime implicant, only the state "close" is really partly causing the top event and the state "open" cannot prevent the top

Num	Tot Prob	%	Prob		Variable	Time	Value
90	9,03E-05	0,09 %	1,00E-03		323-V04-group#30	-4	true
			1,00E-01		516-HA	-3	true
			9,50E-01		211-K101-fail	-3	false
			9,50E-01		211-K102-fail	-3	false
91	9,03E-05	0,09 %	1,00E-03		323-V04-group#30	-3	true
			1,00E-01		516-HA	-3	true
			9,50E-01		211-K101-fail	-3	false
			9,50E-01		211-K102-fail	-3	false
92	9,03E-05	0,09 %	1,00E-03		323-V04-group#20	-3	true
			1,00E-01		516-HA	-3	true
			9,50E-01		211-K101-fail	-3	false
			9,50E-01		211-K102-fail	-3	false
191	4,50E-06	0,00 %	1,00E-03		323-P00-group#04	-4	true
			1,00E-01		516-HB	-3	true
			4,50E-02		323-V204-fail	-2	true
192	4,50E-06	0,00 %	1,00E-03		323-P00-group#03	-4	true
			1,00E-01		516-HB	-3	true
			4,50E-02		323-V204-fail	-2	true
193	4,50E-06	0,00 %	1,00E-03		323-P00-group#03	-3	true
			1,00E-01		516-HB	-3	true
			4,50E-02		323-V204-fail	-2	true

Figure 7.6: Examples of prime implicants that include such common cause failures in which only single failure is significant. In this example, the "true" state of a failure node represents the failure and the "false" state of a failure node means that the component is not failed.

Num	Tot Prob	%	Prob		Variable	Time	Value
355	1,00E-06	0,00 %	1,00E-03		323-P00-group#10	-3	true
			1,00E-03		323-V04-group#03	-4	true
356	1,00E-06	0,00 %	1,00E-03		323-P00-group#30	-3	true
			1,00E-03		323-V04-group#02	-4	true
357	1,00E-06	0,00 %	1,00E-03		323-P00-group#20	-3	true
			1,00E-03		323-V04-group#02	-4	true
488	9,50E-08	0,00 %	1,00E-03		323-P00-group#02	-3	true
			1,00E-03		211-K10-group#20	-4	true
			9,50E-01		211-K101-fail	-3	false
			1,00E-01		516-HB	-3	true
489	9,50E-08	0,00 %	1,00E-03		211-K10-group#04	-4	true
			1,00E-03		323-V04-group#40	-4	true
			1,00E-01		516-HA	-3	true
			9,50E-01		211-K102-fail	-3	false
490	9,50E-08	0,00 %	1,00E-03		211-K10-group#03	-4	true
			1,00E-03		323-V04-group#40	-4	true
			1,00E-01		516-HA	-3	true
			9,50E-01		211-K102-fail	-3	false

Figure 7.7: Examples of prime implicants that include two common cause failure events. In this example, the "true" state represents the failure.

The dynamic Fussell-Vesely for failure state combinations of CCFs						
CCF	States		Time	FV		
323-V04-group	open	close	0	0,007		
			-1	0,007		
			-2	0,007		
			-3	0,007		
			-4	0,003		
	close	open	0	0,007		
			-1	0,007		
			-2	0,007		
			-3	0,007		
			-4	0,003		
	close	close	0	0,052		
			-1	0,052		
			-2	0,052		
			-3	0,052		
			-4	0,036		
211-K40-group	medium	medium	0	0,01		
			-1	0,01		
			-2	0,01		
			-3	0,01		
			-4	0,01		
211-K10-group	high	high	0	0,012		
			-1	0,012		
			-2	0,012		
			-3	0,012		
			-4	0,012		
	high	low	0	0,0005		
			-1	0,0005		
			-2	0,0005		
			-3	0,0005		
			-4	0,0005		
	low	high	0	0,0005		
			-1	0,0005		
			-2	0,0005		
			-3	0,0005		
			-4	0,0005		
323-P00-group	off	off	0	0,066		
			-1	0,066		
			-2	0,066		
			-3	0,065		
			-4	0,043		

Figure 7.8: The dynamic Fussell-Vesely values for failure state combinations of CCFs.

The dynamic Fussell-Vesely for failure states (independent failures)			
Component	State	Time	FV
323-P001	off	0	0,282
	off	-1	0,282
	off	-2	0,282
	off	-3	0,273
	off	-4	0,137
323-V104	close	0	0,282
	close	-1	0,282
	close	-2	0,282
	close	-3	0,273
	close	-4	0,137
211-K401	medium	0	0,02
	medium	-1	0,02
	medium	-2	0,02
	medium	-3	0,02
	medium	-4	0,02
211-K101	high	0	0,041
	high	-1	0,041
	high	-2	0,041
	high	-3	0,041
	high	-4	0,041
The dynamic Fussell-Vesely for 0-states of failure nodes			
Failure node	Time	FV	
211-K101-fail	-3	0,446	
	-4	0,446	

Figure 7.9: The dynamic Fussell-Vesely values for failure states of independently failed components and 0-states of failure nodes.

The total dynamic Fussell-Vesely for failure states			
Component	**State**	**Time**	**FV**
323-P001	off	0	0,348
	off	-1	0,343
	off	-2	0,339
	off	-3	0,326
	off	-4	0,164
323-V104	open	0	0,007
	open	-1	0,003
	open	-2	0,002
	close	0	0,341
	close	-1	0,34
	close	-2	0,337
	close	-3	0,325
	close	-4	0,164
211-K401	medium	0	0,03
	medium	-1	0,03
	medium	-2	0,03
	medium	-3	0,03
	medium	-4	0,03
211-K101	high	0	0,053
	high	-1	0,053
	high	-2	0,053
	high	-3	0,052
	high	-4	0,052
	low	0	0,0005
	low	-1	0,0005
Fussell-Vesely for random nodes			
Random node	**State**	**Time**	**FV**
516-HA	true	0	0,328
	true	-1	0,328
	true	-2	0,328
	true	-3	0,328

Figure 7.10: The total dynamic Fussell-Vesely values for failure states and the dynamic Fussell-Vesely values for states of a random node.

event from occurring because of some other failure in the same pump line. The cases of "high-low" and "low-high" combinations of the pressure measurement sensors are similar.

The dynamic risk increase factor results are presented in Figure 7.11. All common cause failures that occur at time step -4 imply the top event. The dynamic risk increase factor values for independent failures are smaller than earlier only because the top event probability is greater when the common cause failures are taken into account. The total risk increase factor values for failure states are significantly greater than when CCFs were not taken into account. The dynamic risk increase factor is now slightly larger for the "true" state of the random nodes than for independent failures of the pumps and the valves, even though they were equal when CCFs were not considered. The reason for this is that an independent failure prevents the possibility of a CCF that includes the same failure.

The dynamic risk increase factor for failure state combinations of CCFs						
CCF	States		RIF			
323-V04-group	close	close	TOP Pr. 1			
211-K40-group	medium	medium	TOP Pr. 1			
211-K10-group	high	high	TOP Pr. 1			
323-P00-group	off	off	TOP Pr. 1			
The dynamic risk increase factor for failure states (independent failures)						
Component	State	RIF				
323-P001	off	2,83				
323-V104	close	2,83				
211-K401	medium	1,39				
211-K101	high	1,37				
The total dynamic risk increase factor for failure states						
Component	State	RIF				
323-P001	off	3,55				
323-V104	close	3,55				
211-K401	medium	2,26				
211-K101	high	2,24				
The dynamic risk increase factor for random nodes						
Random node	State	RIF				
516-HA	true	2,86				
	false	0,69				

Figure 7.11: The dynamic risk increase factor values for failure state combinations, failure states and random nodes. “TOP Pr. 1” means that the conditional top event probability is 1.

Chapter 8

Discussion

8.1 Dynamic Risk Importance Measures

In this thesis, two traditional risk importance measures, the Fussell-Vesely and the risk increase factor, were generalised to take into account the multi-valued logic and the time aspect of dynamic flowgraph methodology. These new generalisations were implemented in YADRAT. Two example cases were analysed with YADRAT and it was observed that the dynamic risk importance measure results were logically consistent.

The information that is provided by dynamic risk importance measures is more detailed than the information that traditional risk importance measures provide. In addition to an importance order of components, dynamic risk importance measures describe more specifically how components contribute to the top event. As components can usually fail in different states, often, only specific failure states connive with the top event and some failure states might even prevent the considered top event from occurring. With dynamic risk importance measures calculated for different failure states of components and 0-states of failure nodes, a component's influence on the system's reliability can be analysed more comprehensively than with just risk importance measures calculated for failure events. The system's reliability can be improved if the probability that a component fails in a state that has high risk importance measure values is reduced. This can be done, for example by changing the system so that the critical component is less frequently in the critical state. Additionally, as the dynamic Fussell-Vesely is calculated for different failure time steps, it can be judged at which points of the time line certain failures need to occur so that they contribute to the top event.

The Fussell-Vesely measure is the most often used importance measure because it is simple to compute and it encapsulates the information from minimal cut sets or prime implicants purely. The Fussell-Vesely takes into account both the failure probability of

a component and how a component interconnects with other components while the risk increase factor gives a more restricted view on how a component contributes to a top event. The risk increase factor is a more complicated measure to calculate as a new set of prime implicants always has to be identified if the definition is followed strictly. The Fussell-Vesely can be useful alone while the risk increase factor always needs a complementing pair if the system's reliability has to be analysed comprehensively. On the other hand, the risk increase factor values are usually easier to interpret.

A few aspects in dynamic risk importance measure calculation need to be improved before YADRAT can be applied to larger real life systems. The current backtracking technique does not always solve the failure state probabilities perfectly when the dynamic Fussell-Vesely is calculated because all dependencies between timed nodes cannot be taken into account. One way to address this problem is to use similar approach as with the dynamic risk increase factor. All the timed nodes that affect the failure state could be solved and different state combinations of those timed nodes could be considered separately. This would be computationally more demanding. Another option is to build some sort of tree diagram in the backtracking process and to solve accurate probabilities from it. The failure state probability solving should be definitely studied in more depth, at least to understand how large errors can be in the worst scenario with the current technique.

The dynamic Fussell-Vesely is fast to compute compared to identification of prime implicants. However, the dynamic risk increase factor is computationally more demanding. The computation time depends on the number of prime implicants and the system's complexity. When there were circa 5,000 prime implicants, the calculation of all risk increase factor values took over five minutes. Computation time depends on the number of prime implicants quadratically. Thus, with 100,000 prime implicants, the computation would probably last more than a day. The majority of calculation time passes during the post-processing of new prime implicants. One possibility would be to calculate approximations without doing all the post-processing. It is unlikely that reliable approximations could be computed in this way but at least values calculated from non-post-processed implicants would be suggestive. When the dynamic risk increase factor is calculated for a failure state, new prime implicants are identified separately with each state combination of timed nodes that affect the failure state. The computation time would reduce significantly if new prime implicants could be solved only once for each dynamic risk increase factor value. This should be possible to implement by manipulating the state probabilities of the timed nodes that affect the failure state.

The major reason for the inaccuracy of risk importance measure values was the approximation of the top event probability. Prime implicants contain often common literals.

Accurate results can be obtained only if this can be taken into account in the top event probability calculation. The top event probability is difficult to calculate directly from the BDD that represents the model because of the multi-valued logic of YADRAT. One option is to group prime implicants regarding to some literals and take the common literals into account inside groups. However, the inaccuracy of results should not be overly emphasised because the results are already reliable regarding the importance order.

In YADRAT, the initial state can be defined or left open. If all initial states are defined, the failure state probabilities can usually be solved correctly. The risk of errors is bigger if initial states are left open. In the calculation of the dynamic risk increase factor, it is assumed that the failure occurs at the first time step after the initial time. If initial states are defined, a component can often fail in only one state at that time step, hence the dynamic risk increase factor can be calculated for only one failure state. The case where initial states are defined is computationally less demanding but when initial states are left open, more information is obtained.

In YADRAT, the time lag of a failure node can be any negative integer. The time lag of a failure node plays important role in calculation of the dynamic risk increase factor for a failure state as it defines if the failure state is the state of the component at the first time step after the initial time or some later time step. If the time lag is 0, the backtracking can be started from the first time step after the initial time and because of that, the number of timed nodes that affect the failure state is usually smaller than with other time lags. Thus, the time lag 0 is best regarding to the computation of the dynamic risk increase factor for a failure state. To make things more simple, it could be defined that the time lag of a failure node should always be 0. This would not be a significant restriction as a delay of a failure could be modelled with helping nodes. Also, there is no reason why the relation between a failure event and a component could not be defined so that the failure time is the time when the component is affected.

The calculation of the dynamic Fussell-Vesely relied on the assumption that when a literal representing a failure node in state 1 at some time step appears in a prime implicant, the failure must really occur at that time step. The assumption was made because the set of prime implicants usually contains prime implicants that are otherwise perfectly similar or almost similar except that the failure time is different. However, if the definition of a prime implicant was changed so that a prime implicant is an implicant that is not implied by any other implicant, the set of prime implicants would be reduced so that there would not be prime implicants that are otherwise perfectly similar except that the failure time is different. Only the prime implicant with the latest failure time would be left in the set of prime implicants. The objective of this change would be to improve the top event

probability calculation and qualitative properties of YADRAT. In this case, the possibility that a failure can also occur at an earlier time step should be taken into account in the calculation of the dynamic Fussell-Vesely. This would make the calculation process more complicated. It should be studied whether the assumption that the failure might occur at an earlier time step should be made in all cases as there would still be prime implicants that are otherwise almost similar (meaning that one prime implicant would contain one literal more) except that the failure time is different. If the assumption were not made in all cases, different cases should be somehow identified.

In this thesis, the Fussell-Vesely and the risk increase factor were the risk importance measures that were investigated in the domain of dynamic flowgraph methodology. However, there are many other importance measures that could be generalised in dynamic and multi-valued cases as well. Other often-used importance measures include Birnbaum importance, the risk decrease factor (also known as risk reduction worth), the criticality importance and the partial derivative. It is possible to derive some dynamic generalisations of traditional risk importance measures from the dynamic risk increase factor. However, none of them can be derived from the dynamic Fussell-Vesely accurately except when a system is coherent regarding to the considered failure event. The dynamic fractional contribution could be used instead of the dynamic Fussell-Vesely. Anyhow, the dynamic fractional contribution is computationally more demanding. Also, for some risk importance measures, there is more than one way in which the generalisation to the dynamic case can be made. Hence, there are several possibilities for further research on risk importance measures in the DFM domain.

8.2 Dynamic Common Cause Failures

Dynamic generalisations for parametric common cause failure models were also developed in this thesis and they were implemented in YADRAT. In DFM, component failures that occur at different time steps can jointly cause a top event. As components can fail at different time points due to a common cause in real life, it is natural that this is also made possible in DFM. It is important that all the CCFs are taken into account so that the top event probability will not be underestimated.

In the simplest case, which is considered in this thesis, the CCF model parameters are exactly same as in the fault tree analysis. Hence, DFM analysis can be performed based on the same CCF data used in fault tree analysis. However, some more complex versions of the β-factor model and the α-factor model might prove more realistic. In this thesis, it was assumed that all failure time combinations are equally probable but this is usually not true in real life. One possibility would be to use a model in which the CCF

probability depends on the difference between the time of the first failure and the time of the last failure. An even more complex model would take into account all the failure times. The portion of CCF events could also depend on time. More complex models make more realistic reliability analysis possible but a complex model cannot be used realistically before all the parameters are estimated. No research on the parameter estimation for the above-mentioned more complex models has been performed yet.

In this thesis, a constant model was used for failure probabilities. In the future, other reliability models will be implemented in YADRAT. If the component failure probabilities depended on time, the determination of CCF probabilities should also be reconsidered. One possibility would be to use an average of the failure probabilities of different components. Another possibility would be to consider the time step of the first failure as the starting time of a CCF event and only use its failure probability.

The prime implicants that include CCFs are created in the post-processing phase, meaning that CCFs can be included without making the BDD more complicated. This is an quite efficient and simple way to take CCFs into account. However, with these dynamic CCF models, the number of prime implicants can easily become so high that a personal computer runs out of memory. The number of prime implicants with CCFs depends on the number of original prime implicants, the number of time steps, the number of CCF groups and the CCF models. It also depends on the number of components in a CCF group exponentially. The number of prime implicants with CCFs can be dozens of times bigger than the number of original prime implicants when the α-factor model is used. With a personal computer, memory problems can occur if the number of prime implicants is over 500,000. If the definition of a prime implicant was changed so that a prime implicant is an implicant that is not implied by any other implicant, the number of prime implicants with CCFs would be significantly reduced because the direct dependence on the number of time steps and the exponential dependence on the number of components in a CCF group would be removed.

Two parametric common cause failure models, β-factor model and α-factor model, were considered in this thesis. Other parametric CCF models can be implemented in YADRAT similarly. The explicit modelling of CCFs could be performed alike. In addition, DFM provides good options for modelling other types of dependencies between failure events, for example cascading failures.

Chapter 9

Conclusion

In this thesis, risk importance measures were studied in the domain of dynamic flowgraph modelling. Two traditional risk importance measures, the Fussell-Vesely measure and the risk increase factor, were generalised so that they take into account the multi-valued logic and the time aspect of DFM. These risk importance measures encapsulate the information of prime implicants into values that represent importances of different components, random events and initial conditions. Dynamic risk importance measures provide detailed information on how components contribute to the top event.

New dynamic risk importance measures and a new approach to handling the component's failure state were developed. The dynamic Fussell-Vesely was formulated for different time steps. These new dynamic risk importance measures are clearly beneficial in dynamic flowgraph modelling as they provide more information than the traditional importance measures. In theory, they could be applied in all dynamic methods that rely on variables with a finite number of states. The time aspect of dynamic risk importance measures would also be easy to generalise to the case of continuous time.

New dynamic generalisations of traditional parametric common cause failure models were formulated. These dynamic common cause failure models take into account the possibility that failures can occur at different time steps due to a common cause. These models make more realistic modelling of common cause failures possible. They could be applied in other dynamic reliability analysis approaches with discrete time steps.

The dynamic risk importance measures and common cause failure models were implemented in YADRAT and two example cases were analysed. Dynamic risk importance measures provided reliable information about an importance order of components and random events, even when only an irredundant cover of prime implicants was identified. The information about failure states was logically consistent. Dynamic risk importance measures clearly facilitated the analysis of results. Dynamic common cause failure models

worked as they were supposed to and made more realistic analysis of dynamic systems possible.

The goal of the YADRAT development is to provide a dynamic reliability analysis tool for large real applications. The DFM analysis is computationally more demanding than the fault tree analysis because DFM models are more complex. Thus, the biggest challenge is how to provide trustworthy results in a reasonable calculation time. In the dynamic risk importance measure calculation, this means that it is usually better to compute approximations rather than to try to aim at accurate values. Also, even though there are many possibilities for dynamic risk importance measure development in DFM, it is important to consider what information is really useful as the main objective of risk importance measures is to provide guidance for the system's design. The amount of importance measure information have to be reasonable enough so that analysts can interpret it. Similarly, common cause failure models have to be kept simple enough so that they can be applied in practice. The dynamic risk importance measures and common cause failure models developed in this thesis ensure a good basis for further research.

Bibliography

[1] Vesely, W.E., Goldberg, F.F., Roberts, N.H. & Haasl, D.F. Fault Tree Handbook. Washingon D.C.: U.S. Nuclear Regulatory Commission, 1981. 202 p. NUREG-0492.

[2] Clements, P.L. Fault Tree Analysis, 4th Edition. Massachusetts: Sverdrup Technology, Inc., 1993. 96 p.

[3] Willis, R. Survey of Support Software for Reliability Engineering. Washington D.C.: Society of Reliability Engineers, 2006. 20 p.

[4] Labeau, P.E., Smidts, C. & Swaminathan, S. Dynamic Reliability: Towards an Integrated Platform for Probabilistic Risk Assessment. Reliability Engineering and System Safety, Vol. 68 (2000) 3, pp. 219-254.

[5] Garrett, C.J., Guarro, S.B. & Apostolakis, G.E. The Dynamic Flowgraph Methodology for Assessing the Dependability of Embedded Software Systems. Systems, Man and Cybernetics, Vol. 25 (1995) 5, pp. 824-840.

[6] Björkman, K. YADRAT - Concepts and Algorithms. Espoo: VTT Technical Research Centre of Finland, 2010. 26 p. VTT-R-07658-10.

[7] Dymonda [software]. California: Asca, inc., 2010. [refered 18th November 2011] web site: http://www.ascainc.com/dymonda/dymonda.html.

[8] Bryant, R.E. Graph-Based Algorithms for Boolean Function Manipulation. Computers, Vol. C-35 (1986) 8, pp. 677-691.

[9] Bryant, R.E. Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams. ACM Computing Surveys, Vol. 24 (1992) 3, pp. 293-318.

[10] Bucci, P., Kirschenbaum, J., Aldemir, T., Smith, C. & Wood, T. Constructing Dynamic Event Trees from Markov Models. Proceedings of the 8th International Conference on Probabilistic Safety Assessment and Management, New Orleans, Louisiana, USA, 14-18 May 2006. New York: ASME Press. 9 p. ISBN 0-7918-0244-2. PSAM-0369.

[11] Aldemir, T., Guarro, S., Mandelli, D., Kirschenbaum, J., Mangan, L.A., Bucci, P., Yau, M., Ekici, E., Miller, D.W., Sun, X. & Arndt, S.A. Probabilistic Risk Assessment Modeling of Digital Instrumentation and Control Systems Using Two Dynamic Methodologies. Reliability Engineering and System Safety, Vol. 95 (2010) 10, pp. 1011-1039.

[12] Sadou, N. & Demmou, H. Reliability Analysis of Discrete Event Dynamic Systems with Petri Nets. Reliability Engineering and System Safety, Vol. 94 (2009) 11, pp. 1848-1861.

[13] Swaminathan, S. & Smidts, C. The Mathematical Formulation for the Event Sequence Diagram Framework. Reliability Engineering and System Safety, Vol. 65 (1999) 2, pp. 103-118.

[14] Matsuoka, T. Reliability Analyses of a Self-holding Type Relay System by a Dynamical Event Tree and the GO-FLOW Methodology. Proceedings of the 8th International Conference on Probabilistic Safety Assessment and Management, New Orleans, Louisiana, USA, 14-18 May 2006. New York: ASME Press. 7 p. ISBN 0-7918-0244-2. PSAM-0214.

[15] Cepin, M. & Mavko, B. A Dynamic Fault Tree. Reliability Engineering and System Safety, Vol. 75 (2002) 1, pp. 83-91.

[16] Høyland, A. & Rausand, M. System Reliability Theory: Models and Statistical Methods. New York: Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics Section, 1994. 518 p. ISBN 0-471-59397-4.

[17] Mankamo, T., Pörn, K. & Holmberg J.E. Uses of Risk Importance Measures. Technical Report. Espoo: VTT Technical Research Centre of Finland, 1991. 36 p. ISBN 951-38-3877-3.

[18] Van Der Borst, M., & Schoonakker, H. An Overview of PSA Importance Measures. Reliability Engineering and System Safety, Vol. 72 (2001) 3, pp. 241-245.

[19] Mosleh A., Fleming, K.N., Parry, G.W., Paula, H.M., Worledge, D.H. & Rasmuson, D.M. Procedures for Treating Common Cause Failures in Safety and Reliability Studies: Procedular Framework and Examples. Newport Beach, CA: Pickard, Lowe, and Garrick, inc., 1988. 202 p. NUREG/CR-4780 EPRI NP-5613 Vol. 1.

[20] Stott, J.E., Britton, B.T., Ring, R.W., Hark, F. & Hatfield, G.S. Common Cause Failure Modeling: Aerospace vs. Nuclear. Proceedings of the 10th International Conference on Probabilistic Safety Assessment and Management, Seattle, Washington, USA, 7-11 June 2010. IAPSAM. 12 p. Paper 371.

[21] Halmos, P. & Givant, S. Introduction to Boolean Algebras. New York: Springer, Undergraduate Texts in Mathematics, 2009. 446 p. ISBN 978-0-387-40293-2.

[22] Rauzy, A. Mathematical Foundation of Minimal Cutsets. IEEE transactions on Reliability, Vol. 50 (2001) 4, pp. 389-396.

[23] Roman, S. Advanced Linear Algebra, Second Edition. New York: Springer, Graduate Texts in Mathematics, 2005. 495 p. ISBN 0-387-24766-1.

[24] Contini, S., Cojazzi, G.G.M. & Renda, G. On the Use of Non-coherent Fault Trees in Safety and Security Studies. Reliability Engineering and System Safety, Vol. 93 (2008) 12, pp. 1886-1895.

[25] Brace, K.S., Rudell, R.L. & Bryant, R.E. Efficient Implementation of a BDD Package. DAC '90 Proceedings of the 27th ACM/IEEE Design Automation Conference, Orlando, FL, USA, 24-28 June 1990. New York: ACM. pp. 40-45. ISBN 0-89791-363-9.

[26] Harlow III, J.E. & Brglez, F. Design of Experiments and Evaluation of BDD Ordering Heuristics. International Journal on Software Tools for Techonology Transfer (STTT), Vol. 3 (2001) 2, pp. 193-206.

[27] Du, S. & Sun, Y. Comparison of Progressive Variable Ordering Methods with Fixed Ordering Heuristics for Binary Decision Diagrams. Proceedings of 2007 International Conference on Wireless Communications, Networking and Mobile Computing. Shanghai, China, 21-25 September 2007. Piscataway, NJ: IEEE Operations Center. pp. 4581-4584. ISBN 978-1-4244-1311-9.

[28] Rauzy, A. New Algorithms for Fault Trees Analysis. Reliability Engineering and System Safety, Vol. 40 (1993) 3, pp. 203-211.

[29] Rauzy, A. & Dutuit, Y. Exact and Truncated Computation of Prime Implicants of Coherent and Non-coherent Fault Trees within Aralia. Reliability Engineering and System Safety, Vol. 58 (1997) 2, pp. 127-144.

[30] Andrews, J.D. & Dunnett, S.J. Event-Tree Analysis Using Binary Decision Diagrams. IEEE transactions on Reliability, Vol. 49 (2000) 2, pp. 230-238.

[31] Bartzis, C. & Bultan, T. Efficient BDDs for Bounded Arithmetic Constraints. International Journal on Software Tools for Technology Transfer (STTT), Vol. 8 (2006) 1, pp. 26-36.

[32] Minato, S. Zero-suppressed BDDs for Set Manipulation in Combinatorial Problems. DAC '93 Proceedings of the 30th ACM/IEEE Design Automation Conference, Dallas, TX, USA, 14-18 June 1993. New York: ACM. pp. 272-277. ISBN 0-89791-577-1.

[33] Morreale, E. Recursive Operators for Prime Implicant and Irredundant Normal Form Determination. Computers, Vol. C-19 (1970) 6, pp. 504-509.

[34] Coudert, O. & Madre, J.C. Implicit and Incremental Computation of Primes and Essential Primes of Boolean Functions. DAC '92 Proceedings of the 29th ACM/IEEE Design Automation Conference, Anaheim, CA, USA, 8-12 June 1992. Los Alamitos, CA, USA: IEEE Computer Society Press. pp. 36-39. ISBN 0-8186-2822-7.

[35] Somenzi, F. CUDD: CU Decision Diagram Package [public software]. Colorado: University of Colorado, 1997. [refered 7th July 2011] web site: http://vlsi.colorado.edu/ fabio/CUDD/.

[36] Huang, C.Y. & Chang, Y.R. An Improved Decomposition Scheme for Assessing the Reliability of Embedded Systems by Using Dynamic Fault Trees. Reliability Engineering and System Safety, Vol. 92 (2007) 10, pp. 1403-1412.

[37] Do Van, P., Barros, A. & Bérenguer, C. Reliability Importance Analysis of Markovian Systems at Steady State Using Perturbation Analysis. Reliability Engineering and System Safety, Vol. 93 (2008) 11, pp. 1605-1615.

[38] Do Van, P., Barros, A. & Bérenguer, C. From Differential to Difference Importance Measures for Markov Reliability Models. European Journal on Operational Research, Vol. 204 (2010) 3, pp. 513-521.

[39] Ramirez-Marquez, J.E., Rocco, C.M., Gebre, B.A., Coit, D.W. & Tortorella, M. New Insights on Multi-State Component Criticality and Importance. Reliability Engineering and System Safety, Vol. 91 (2006) 8, pp. 894-904.

[40] Levitin, G., Podofillini, L. & Zio, E. Generalised Importance Measures for Multi-State Elements Based on Performance Level Restrictions. Reliability Engineering and System Safety, Vol. 82 (2003) 3, pp. 287-298.

[41] Ramirez-Marquez, J.E. & Coit, D.W. Composite Importance Measures for Multi-State Systems with Multi-State Components. IEEE transactions on Reliability, Vol. 54 (2005) 3, pp. 517-529.

[42] Guo, H. & Yang, X. Automatic Creation of Markov Models for Reliability Assessment of Safety Instrumented Systems. Reliability Engineering and System Safety, Vol. 93 (2008) 6, pp. 807-815.

[43] Xing, L., Shrestha, A., Meshkat, L. & Wang, W. Incorporating Common-Cause Failures Into the Modular Hierarchical Systems Analysis. IEEE transactions on Reliability, Vol. 58 (2009) 1, pp. 10-19.

[44] Matsuoka, T. & Kobayashi, M. The GO-FLOW Reliability Analysis Methodology Analysis of Common Cause Failures with Uncertainty. Nuclear Engineering and Design, Vol. 175 (1997) 3, pp. 205-214.

[45] Björkman, K. & Holmberg, J.E. Comparison of Two Dynamic Reliability Analysis Tools to Solve Dynamic Flowgraph Method Models. Espoo: VTT Technical Research Centre of Finland, 2011. 30 p. VTT-R-00775-10.

Appendix A

The First Example of Solving Failure State Probabilities

In this chapter, an example YADRAT model presented in Figure A.1 and decision tables A.1-A.4, is used to illustrate the failure state probability calculation process. Let the initial time be -4 and the top event $M(0) = 2$. A set of literals $\{R(-3) = 1, T(-4) = 1, TF(-3) = 0, E(-4) = 0, D(-4) = 1, DF(-2) = 0, DF(-1) = 1\}$ is a prime implicant of this top event. Let it be considered in which state a component D is stuck when a failure node DF turns to state 1 at time step -1. The time lag of DF is -1. Thus, the state probabilities of $D(0)$ must be solved. The backtracking process is presented in tables A.5-A.9.

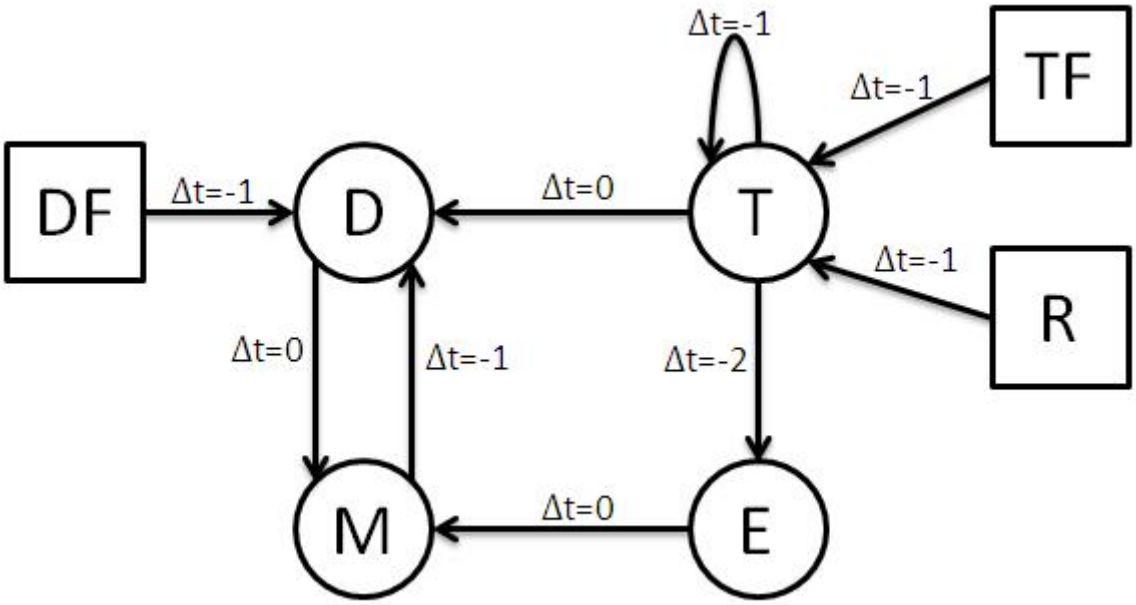

Figure A.1: A YADRAT model.

Table A.1: The decision table of the component D.

	Output	Inputs		
Node	D	M	T	DF
Time lag	0	-1	0	-1
0	1	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	2	1	0	0
5	1	1	0	1
6	2	1	1	0
7	1	1	1	1
8	1	2	0	0
9	2	2	0	1
10	1	2	1	0
11	2	2	1	1

Table A.2: The decision table of the component T.

	Output	Inputs		
Node	T	R	T	TF
Time lag	0	-1	-1	-1
0	1	0	0	0
1	1	0	0	1
2	0	0	1	0
3	1	0	1	1
4	0	1	0	0
5	1	1	0	1
6	0	1	1	0
7	1	1	1	1

Table A.3: The decision table of the deterministic node M.

	Output	Inputs	
Node	M	E	D
Time lag	0	0	0
0	0	0	0
1	1	0	1
2	2	0	2
3	0	1	0
4	0	1	1
5	1	1	2

Table A.4: The decision table of the deterministic node E.

	Output	Input
Node	E	T
Time lag	0	-2
0	0	0
1	1	1

Table A.5: Steps 1-7 of the backtracking process to obtain the state probabilities of $D(0)$.

Step	Output		Input	State pr.			Case
				0	1	2	
1	$D(0)$	$\sim$	$M(-1)$	?	?	?	D1
			$T(0)$	?	?		D1
			$DF(-1)$	0	1		F1
	Result:		probs. of $M(-1)$ needed				
2	$M(-1)$	$\sim$	$D(-1)$	?	?	?	D1
			$E(-1)$	?	?		D1
	Result:		probs. of $D(-1)$ needed				
3	$D(-1)$	$\sim$	$M(-2)$	?	?	?	D1
			$T(-1)$	?	?		D1
			$DF(-2)$	1	0		F4
	Result:		probs. of $M(-2)$ needed				
4	$M(-2)$	$\sim$	$D(-2)$	?	?	?	D1
			$E(-2)$	?	?		D1
	Result:		probs. of $D(-2)$ needed				
5	$D(-2)$	$\sim$	$M(-3)$	?	?	?	D1
			$T(-2)$	?	?		D1
			$DF(-3)$	1	0		F4
	Result:		probs. of $M(-3)$ needed				
6	$M(-3)$	$\sim$	$D(-3)$	?	?	?	D1
			$E(-3)$	?	?		D1
	Result:		probs. of $D(-3)$ needed				
7	$D(-3)$	$\sim$	$M(-4)$	?	?	?	D4
			$T(-3)$	?	?		D1
			$DF(-4)$	1	0		F6
	Result:		probs. of $M(-4)$ needed				

Table A.6: Steps 8-14 of the backtracking process to obtain the state probabilities of $D(0)$. The probability r_i is the probability that the initial state of R is 1.

Step	Output		Input	State pr.			Case
				0	1	2	
8	$M(-4)$	$\sim$	$D(-4)$	0	1	0	D2
			$E(-4)$	1	0		D2
	Result:		$M(-4)$	0	1	0	
9	$D(-3)$	$\sim$	$M(-4)$	0	1	0	D4
			$T(-3)$	?	?		D1
			$DF(-4)$	1	0		F6
	Result:		probs. of $T(-3)$ needed				
10	$T(-3)$	$\sim$	$T(-4)$	0	1		D2
			$TF(-4)$	1	0		F6
			$R(-4)$	$1-r_i$	r_i		R2
	Result:		$T(-3)$	1	0		
11	$D(-3)$	$\sim$	$M(-4)$	0	1	0	D4
			$T(-3)$	1	0		D1
			$DF(-4)$	1	0		F6
	Result:		$D(-3)$	0	0	1	
12	$M(-3)$	$\sim$	$D(-3)$	0	0	1	D1
			$E(-3)$	?	?		D1
	Result:		probs. of $E(-3)$ needed				
13	$E(-3)$	$\sim$	$T(-5)$	0	1		D2
	Result:		$E(-3)$	0	1		
14	$M(-3)$	$\sim$	$D(-3)$	0	0	1	D1
			$E(-3)$	0	1		D1
	Result:		$M(-3)$	0	1	0	

Table A.7: Steps 15-21 of the backtracking process to obtain the state probabilities of $D(0)$. A number in parentheses in the case column refers to a step number.

Step	Output		Input	State pr.			Case
				0	1	2	
15	$D(-2)$	$\sim$	$M(-3)$	0	1	0	D1
			$T(-2)$	?	?		D1
			$DF(-3)$	1	0		F4
	Result:		probs. of $T(-2)$ needed				
16	$T(-2)$	$\sim$	$T(-3)$	1	0		D1(10)
			$TF(-3)$	1	0		F4
			$R(-3)$	0	1		R1
	Result:		$T(-2)$	1	0		
17	$D(-2)$	$\sim$	$M(-3)$	0	1	0	D1
			$T(-2)$	1	0		D1
			$DF(-3)$	1	0		F4
	Result:		$D(-2)$	0	0	1	
18	$M(-2)$	$\sim$	$D(-2)$	0	0	1	D1
			$E(-2)$	?	?		D1
	Result:		probs. of $E(-2)$ needed				
19	$E(-2)$	$\sim$	$T(-4)$	0	1		D2
	Result:		$E(-2)$	0	1		
20	$M(-2)$	$\sim$	$D(-2)$	0	0	1	D1
			$E(-2)$	0	1		D1
	Result:		$M(-2)$	0	0	1	
21	$D(-1)$	$\sim$	$M(-2)$	0	1	0	D1
			$T(-1)$	?	?		D1
			$DF(-2)$	1	0		F4
	Result:		probs. of $T(-1)$ needed				

Table A.8: Steps 22-28 of the backtracking process to obtain the state probabilities of $D(0)$. The probability t is the failure probability of TF and r_l is the probability that R is in state 1 at a time step after the initial time. A number in parentheses in the case column refers to a step number.

Step	Output		Input	State pr.			Case
				0	1	2	
22	$T(-1)$	$\sim$	$T(-2)$	1	0		D1(16)
			$TF(-2)$	$1-t$	t		F3
			$R(-2)$	$1-r_l$	r_l		R3
	Result:		$T(-1)$	$(1-t)\cdot r_l$	$1-(1-t)\cdot r_l$		
23	$D(-1)$	$\sim$	$M(-2)$	0	1	0	D1
			$T(-1)$	$(1-t)\cdot r_l$	$1-(1-t)\cdot r_l$		D1
			$DF(-2)$	1	0		F4
	Result:		$D(-1)$	0	0	1	
24	$M(-1)$	$\sim$	$D(-1)$	0	0	1	D1
			$E(-1)$	?	?		D1
	Result:		probs. of $E(-1)$ needed				
25	$E(-1)$	$\sim$	$T(-3)$	1	0		D1(10)
	Result:		$E(-1)$	1	0		
26	$M(-1)$	$\sim$	$D(-1)$	0	0	1	D1
			$E(-1)$	1	0		D1
	Result:		$M(-1)$	0	0	1	
27	$D(0)$	$\sim$	$M(-1)$	0	0	1	D1
			$T(0)$	?	?		D1
			$DF(-1)$	0	1		F1
	Result:		probs. of $T(0)$ needed				
28	$T(0)$	$\sim$	$T(-1)$	1	0		D1(22)
			$TF(-1)$	$1-t$	t		F3
			$R(-1)$	$1-r_l$	r_l		R3
	Result:		$T(0)$	$(1-t)\cdot r_l$	$1-(1-t)\cdot r_l$		

Table A.9: Step 29 of the backtracking process to obtain the state probabilities of $D(0)$. The probability t is the failure probability of TF and r_l is the probability that R is in state 1 at a time step after the initial time.

Step	Output		Input	State pr.			Case
				0	1	2	
29	$D(0)$	$\sim$	$M(-1)$	0	0	1	D1
			$T(0)$	$(1-t)\cdot r_l$	$1-(1-t)\cdot r_l$		D1
			$DF(-1)$	0	1		F1
	Result:		$D(0)$	0	0	1	

Appendix B

The Second Example of Solving Failure State Probabilities

In this chapter, an example YADRAT model, presented in Figure B.1 and decision tables B.1-B.3, is used to illustrate the failure state probability calculation process. Table B.4 presents the initial probabilities of the components I and H and the state probabilities of the failure node IF. Let the initial time be -4 and the top event $T(0) = 1$. A set $\{HF(-1) = 1\}$ is a prime implicant of this top event. Let it be considered in which state a component H is stuck when a failure node HF turns to state 1 at time step -1. The time lag of HF is 0. Thus, the state probabilities of $H(-1)$ must be solved. The backtracking process is presented in table B.5.

Besides illustrating the backtracking process, this example also shows the weakness of this backtracking technique. This technique cannot take all the dependencies between timed nodes into account. In this case, timed nodes $I(-4)$ and $I(-5)$ should be in same state because the system is in a steady state before the initial time. However, in the backtracking, it is assumed that these timed nodes are independent. The only way for H to be in state 0 the time step -2 is if the initial states of H and I are both 0. Hence,

Table B.1: The decision table of the deterministic node T.

	Output	Input
Node	T	H
Time lag	0	0
0	0	0
1	1	1
2	1	2

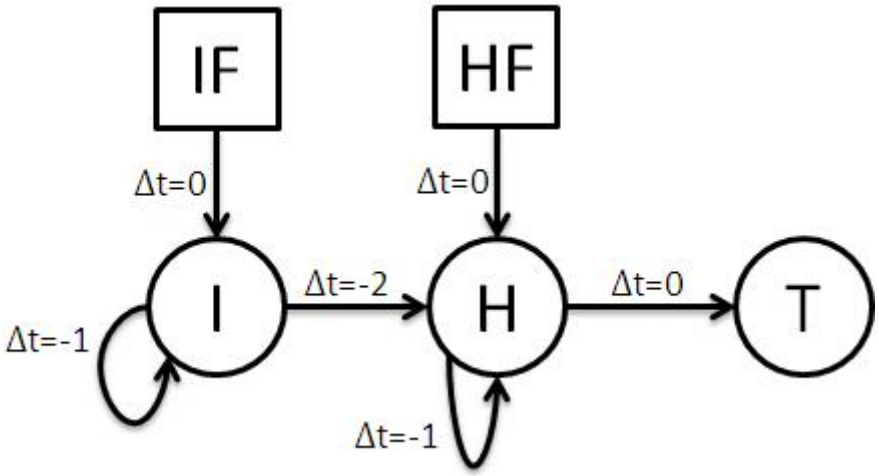

Figure B.1: A YADRAT model.

Table B.2: The decision table of the component H.

	Output	Inputs		
Node	H	H	HF	I
Time lag	0	-1	0	-2
0	1	0	0	0
1	1	0	0	1
2	1	0	1	0
3	1	0	1	1
4	0	1	0	0
5	2	1	0	1
6	1	1	1	0
7	1	1	1	1
8	0	2	0	0
9	1	2	0	1
10	2	2	1	0
11	2	2	1	1

Table B.3: The decision table of the component I.

	Output	Inputs	
Node	I	I	IF
Time lag	0	-1	0
0	1	0	0
1	0	0	1
2	0	1	0
3	1	1	1

Table B.4: The initial probabilities of the components I and H and the state probabilities of the failure node IF.

Node	State probabilities		
	0	1	2
I	0.5	0.5	
H	0.6	0.2	0.2
IF	0.9	0.1	

the real probability for H being in state 0 at time step -2 is 0.3. The component H can be in state 2 at time step -2 only if the initial state of I is 1 and the initial state of H is not 1. Hence, the real probability for H being in state 2 at time step -2 is 0.4. Therefore, the real state probabilities of $H(-2)$ should be 0.3, 0.3 and 0.4 and the failure state probabilities should be 0, 0.6 and 0.4. Thus, there is a small error in the results.

Table B.5: Steps 1-7 of the backtracking process to obtain the state probabilities of $H(-1)$.

Step	Output		Input	State pr.			Case
				0	1	2	
1	$H(-1)$	$\sim$	$H(-2)$	?	?	?	D1
			$HF(-1)$	0	1		F1
			$I(-3)$	?	?		D1
	Result:		probs. of $H(-2)$ needed				
2	$H(-2)$	$\sim$	$H(-3)$	?	?	?	D1
			$HF(-2)$	1	0		F2
			$I(-4)$	0.5	0.5		D3
	Result:		probs. of $H(-3)$ needed				
3	$H(-3)$	$\sim$	$H(-4)$	0.6	0.2	0.2	D3
			$HF(-3)$	1	0		F2
			$I(-5)$	0.5	0.5		D5
	Result:		$H(-3)$	0.2	0.7	0.1	
4	$H(-2)$	$\sim$	$H(-3)$	0.2	0.7	0.1	D1
			$HF(-2)$	1	0		F2
			$I(-4)$	0.5	0.5		D3
	Result:		$H(-2)$	0.4	0.25	0.35	
5	$H(-1)$	$\sim$	$H(-2)$	0.4	0.25	0.35	D1
			$HF(-1)$	0	1		F1
			$I(-3)$	?	?		D1
	Result:		probs. of $I(-3)$ needed				
6	$I(-3)$	$\sim$	$I(-4)$	0.5	0.5		D3
			$IF(-3)$	0.9	0.1		F5
	Result:		$I(-3)$	0.5	0.5		
7	$H(-1)$	$\sim$	$H(-2)$	0.4	0.25	0.35	D1
			$HF(-1)$	0	1		F1
			$I(-3)$	0.5	0.5		D1
	Result:		$H(-1)$	0	0.65	0.35	