Isolation of Multiple-faults with Generalized Fault-modes

Master's thesis
performed in **Vehicular Systems**
for **Scania CV AB**

by
Dan Sune

Reg nr: LiTH-ISY-EX-3380-2002

2nd December 2002

Isolation of Multiple-faults with Generalized Fault-modes

Master's thesis

performed in **Vehicular Systems**,
Dept. of Electrical Engineering
at **Linköpings universitet**

Performed for **Scania CV AB**
by **Dan Sune**

Reg nr: LiTH-ISY-EX-3380-2002

Supervisor: **Mattias Nyberg**
Scania CV AB
Mattias Krysander
Linköpings universitet

Examiner: **Professor Lars Nielsen**
Linköpings universitet

Linköping, 2nd December 2002

Avdelning, Institution
Division, Department

Vehicular Systems,
Dept. of Electrical Engineering
581 83 Linköping

Datum
Date

2nd December 2002

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ ______________

Rapporttyp
Report category

☐ Licentiatavhandling
☒ Examensarbete
☐ C-uppsats
☐ D-uppsats
☐ Övrig rapport
☐ ______________

ISBN
—

ISRN
LITH-ISY-EX-3380-2002

Serietitel och serienummer
Title of series, numbering

ISSN
—

URL för elektronisk version

http://www.vehicular.isy.liu.se
http://www.ep.liu.se/exjobb/isy/2002/3380/

Titel Isolering av multipelfel med generella felmoder

Title Isolation of Multiple-faults with Generalized Fault-modes

Författare Dan Sune
Author

Sammanfattning
Abstract

Most AI approaches for fault isolation handle only the behavioral modes OK and $\neg OK$. To be able to isolate faults in components with generalized behavioral modes, a new framework is needed. By introducing domain logic and assigning the behavior of a component to a behavioral mode domain, efficient representation and calculation of diagnostic information is made possible.

Diagnosing components with generalized behavioral modes also requires extending familiar characterizations. The characterizations candidate, generalized kernel candidate and generalized minimal candidate are introduced and it is indicated how these are deduced.

It is concluded that neither the full candidate representation nor the generalized kernel candidate representation are conclusive enough. The generalized minimal candidate representation focuses on the interesting diagnostic statements to a large extent. If further focusing is needed, it is satisfactory to present the minimal candidates which have a probability close to the most probable minimal candidate.

The performance of the fault isolation algorithm is very good, faults are isolated as far as it is possible with the provided diagnostic information.

Nyckelord
Keywords

Behavioral mode, logic, assumption based diagnostics, Generalized minimal candidate

Abstract

Most AI approaches for fault isolation handle only the behavioral modes OK and $\neg OK$. To be able to isolate faults in components with generalized behavioral modes, a new framework is needed. By introducing domain logic and assigning the behavior of a component to a behavioral mode domain, efficient representation and calculation of diagnostic information is made possible.

Diagnosing components with generalized behavioral modes also requires extending familiar characterizations. The characterizations candidate, generalized kernel candidate and generalized minimal candidate are introduced and it is indicated how these are deduced.

It is concluded that neither the full candidate representation nor the generalized kernel candidate representation are conclusive enough. The generalized minimal candidate representation focuses on the interesting diagnostic statements to a large extent. If further focusing is needed, it is satisfactory to present the minimal candidates which have a probability close to the most probable minimal candidate.

The performance of the fault isolation algorithm is very good, faults are isolated as far as it is possible with the provided diagnostic information.

Keywords: Behavioral mode, logic, assumption based diagnostics, Generalized minimal candidate

Contents

Chapter 1

Introduction

Background

The environmental laws of tomorrow require more complex engine-systems in heavy trucks. In addition, the laws will also require more accurate diagnostic systems. Thus the demands on the diagnostic systems in heavy trucks will increase dramatically.

Because of the greater complexity of the engines it will be difficult to monitor all components with physical sensors. Instead one wishes to compare modelled values with the corresponding measured values. Provided that a model is correct, certain assumptions can be made in case the value of the model deviates too much from the measured value. Either the sensor is broken or there is something wrong with the components that the model is based on.

Each component may be involved in several models and each model may be based on several components. Especially in the AI community a number of approaches for fault isolation have been suggested. However, most of these, e.g in [1] and [2], handle only the fault modes OK and $\neg OK$. Since components in the engine can generally fail in more than one way, these approaches are inadequate. To isolate faults in components with general behavioral modes, a framework and an algorithm is needed. The method presented here handles multiple faults and multiple fault modes.

Objective

The objective was to develop a framework and a strategy for isolating faults based on the conflicts provided by a diagnostic system. The framework and the strategy should handle multiple fault and generalized fault modes.

Methods

An algorithm for isolating faults given a set of conflicts [2] and an environment for manipulating logical expressions has been implemented in **Matlab**.

Recommendations to the reader

It is recommended that the reader if familiar with the current research in the field of diagnostics. Basic knowledge of logic is also necessary.

Chapter 2

The basics of assumption based diagnostics

2.1 Behavioral modes

A behavioral mode of a component describes how component functions at the present time. The present behavioral mode of component C is denoted B_C.

The behavioral modes no-fault and unknown-fault are common to all components. The reason for the unknown-fault mode is that it can't be guaranteed that all manners in which a component can fail are known. Each component C has a set of possible behavioral modes. This set of all possible behavioral modes for C is denoted Υ_C.

A component is restricted to one behavioral mode at one point in time.

Definition 2.1. Φ_C
Component C, $\Upsilon_C = \{BM_C^1 ... BM_C^n\}$
The formula Φ_C *expresses the condition that a component is restricted to one behavior mode at one point in time.*

$$\Phi_C = \bigvee_{i=1}^{n} ((\bigwedge_{\forall BM_C^j \in \Upsilon_C \setminus \{BM_C^i\}} \neg(B_C = BM_C^j)) \wedge B_C = BM_C^i)$$

□

The following examples illustrates the formulas Υ_C and Φ_C for two components.

Example 2.1
Two components, a switch (S) and a lamp (L) assigned to the no-fault

mode, in a circuit connected to a power supply which cannot be faulty.

$$
\begin{aligned}
\Upsilon_S &= \{NF_S, SO_S, UF_S\} \\
\Upsilon_L &= \{NF_L, BO_L, UF_L\} \\
NF: & \quad \text{No fault} \\
SO: & \quad \text{Stuck open} \\
BO: & \quad \text{Burnt out} \\
UF: & \quad \text{Unknown fault} \\
\Phi_S &= \begin{cases} (B_S = NF_S \ \wedge \ \neg(B_S = SO_S) \ \wedge \ \neg(B_S = UF_S)) \vee \\ (\neg(B_S = NF_S) \ \wedge \ B_S = SO_S \ \wedge \ \neg(B_S = UF_S)) \vee \\ (\neg(B_S = NF_S) \ \wedge \ \neg(B_S = SO_S) \ \wedge B_S = UF_S) \end{cases} \\
\Phi_L &= \begin{cases} (B_L = NF_L \ \wedge \ \neg(B_L = BO_L) \ \wedge \ \neg(B_L = UF_L)) \vee \\ (\neg(B_L = NF_L) \ \wedge \ B_L = BO_L \ \wedge \ \neg(B_L = UF_L)) \vee \\ (\neg(B_L = NF_L) \ \wedge \ \neg(B_L = BO_L) \ \wedge B_L = UF_L) \end{cases} \\
B_S &= NF_S \\
B_L &= NF_L
\end{aligned}
$$

2.2 Submodels and assumptions

In control systems of modern engines, submodels are used to produce estimates of interesting values. Every submodel which relates to a physical subsystem contains important information about the condition of that subsystem. To be able to deduce relevant diagnostic statements, it is important to savor this information.

If the result of an otherwise correct submodel is unreasonable, the conclusion is that an error has occurred in the subsystem which the submodel is related to. Each submodel is valid if a sufficient condition, called the *assumption* of the submodel, is fulfilled. This condition expresses behavioral mode assignments sufficient to make the submodel valid.

Definition 2.2. Assumption
The assumption of a submodel M is a logic expressing behavioral mode assignments such that:

$$\text{ass } M \ \longrightarrow \ M$$

□

If a model is invalidated it means that the corresponding assumption does not hold. This observation is useful for constructing hypothesis tests.

2.3 Hypothesis tests

The concept hypothesis test is known from earlier works in this field [2]. In assumption based diagnostics, hypothesis tests are used to decide whether or not the assumption of a model holds which the following example will illustrate.

Example 2.2

Hypothesis test, Example 2.1 continued. The desired position of the switch is closed and the amount of light emanating form the lamp, Light_L, is measured

$$\begin{aligned} \text{Submodel, } M: &\quad \text{Light}_L^{expected} = \text{Light}_L^{measured} \\ \text{Assumption, ass } M: &\quad B_S = NF_S \ \wedge \ B_L = NF_L \\ \text{Test quantity, } T: &\quad \text{Light}_L^{expected} - \text{Light}_L^{measured} \\ \text{Hypotheses:} &\quad H_0 : \text{ass } M \\ \text{Rejection region, } R: &\quad \{x \in \mathbb{R} \mid x \geq \varepsilon\} \\ \text{Test decisions:} &\quad \begin{cases} H_0 \text{ not rejected} & \text{if } T \in R^C \\ H_0 \text{ rejected} & \text{if } T \in R \end{cases} \end{aligned}$$

Thus if T is in the rejection region H_0 will be rejected and so will ass M.

The essence of a hypothesis test such as the one in the example can be summarized as:

$$\begin{gathered} \text{ass } M \ \rightarrow \ M \ \rightarrow \ T \in R^C \\ \Longleftrightarrow \\ T \in R \ \rightarrow \ \neg M \ \rightarrow \ \neg\text{ass } M \end{gathered}$$

I.e. if the test decision is in the rejection region the assumption of the submodel will be rejected. Since a submodel may produce reasonable values even if the assumption does not hold, no conclusions can be drawn if H_0 is not rejected.

$$T \in R^C \ \nrightarrow \ \text{ass } M$$

Information about the system can therefore only be acquired when H_0 is rejected. Rejecting H_0 implies that the assumption, ass M, does not hold. Hence if H_0 is rejected, the formula representing the assumption constitutes a conflict [2]. Rejection of H_0 is illustrated in the next example.

Example 2.3

Invalidation of a submodel, Example 2.2 continued.

$$\text{Decision: Reject } H_0 \longrightarrow \neg\text{ass } M$$

$$\begin{gathered}\neg\text{ass } M \\ \simeq \\ \neg(B_S = NF_S \ \wedge \ B_L = NF_L) \\ \simeq \\ \neg(B_S = NF_S) \ \vee \ \neg(B_L = NF_L)\end{gathered}$$

Evaluating the negation ass M renders negation of behavior mode assignments of single components.

Evaluating the negation of a conflict renders a logic formula made up of negated behavioral mode assignments of single components. To evaluate this formula it is necessary to apply the restrictions Φ_{C_i} as will be illustrated in the following example.

Example 2.4
Negation of a behavioral mode assignment of one component, Example 2.3 continued.

$$\begin{gathered}\neg(B_S = NF_S) \wedge \Phi_S \\ \simeq \\ (\neg(B_S = NF_S) \wedge B_S = NF_S \wedge \neg(B_S = SO_S) \wedge \neg(B_S = UF_S)) \vee \\ (\neg(B_S = NF_S) \wedge \neg(B_S = NF_S) \wedge B_S = SO_S \wedge \neg(B_S = UF_S)) \vee \\ (\neg(B_S = NF_S) \wedge \neg(B_S = NF_S) \wedge \neg(B_S = SO_S) \wedge B_S = UF_S) \\ \simeq \\ (\neg(B_S = NF_S) \wedge B_S = SO_S \wedge \neg(B_S = UF_S)) \vee \\ (\neg(B_S = NF_S) \wedge \neg(B_S = SO_S) \wedge B_S = UF_S) \\ \simeq \\ (\textstyle\bigvee_{\forall BM_S^i \in \Upsilon_S \setminus \{NF_S\}} B_S = BM_S^i) \ \wedge \ \Phi_S\end{gathered}$$

Observe that the formula Φ_S is present in both the first and last formulas of the example.

To create a more compact representation the operator $\simeq_\Phi$ is introduced.

Definition 2.3. $\simeq_\Phi$, equivalent under the restriction Φ

$$\begin{gathered}A \quad \simeq_\Phi \quad B \\ \Longleftrightarrow \\ A \ \wedge \ \Phi \ \simeq \ B \ \wedge \ \Phi\end{gathered}$$

□

After introducing the restriction operator $\simeq_\Phi$ it is more convenient to evaluate negated assumptions as is illustrated in the following example.

Example 2.5
Negation of ass M, Example 2.2 continued. Note that a restriction operator over several components such as: $\simeq_{(\Phi_S \ \wedge \ \Phi_L)}$ is written as: $\simeq_{\Phi_{S,L}}$

$$\neg(B_S = NF_S) \ \vee \ \neg(B_L = NF_L)$$
$$\simeq_{\Phi_{S,L}}$$
$$(\bigvee_{\forall BM_S^i \in \Upsilon_S \setminus \{NF_S\}} B_S = BM_S^i) \ \vee \ (\bigvee_{\forall BM_L^j \in \Upsilon_S \setminus \{NF_L\}} B_L = BM_L^j)$$
$$\simeq_{\Phi_{S,L}}$$
$$B_S = SO_S \vee B_S = UF_S \vee B_L = BO_L \vee B_L = UF_L$$

The result of evaluating the negated assumption is a disjunction of behaviorial mode assignments of single components. These are alternative explanations to what has been observed.

If one assumption is negated the result is a fairly simple formula. As the complexity of the diagnosed system increases so will the complexity of the formulas. Since logic is used to express the diagnostic information there may be several representations that are logically equivalent but very different from a readability point of view. Furthermore, and even more important is the impact that the representation has on the efficiency of the fault isolation algorithm. Therefore a strategy for representing the diagnostic information is important.

Chapter 3

Diagnostic information expressed in logic

The diagnostic information is expressed in logic. In Table 3.1 the fundamental concepts of logic formulas are listed.

Table 3.1: Fundamental concents of logic formulas according to [3]

Concept	*Operator*	*Arguments*	*Example*
Constant	-	-	a, b, 6
Functional symbol	-	-	$+, -, *$
Variable symbol	-	-	x, y, z
Connective	-	-	$\vee, \wedge, \neg$
Relational symbol	-	-	$\geq, \leq, =$
Quantifiers	-	-	$\forall, \exists$
Term	Functional symbols	Variable symbols, Constants	a + x, 2 +2
Atom	Relational symbols	Terms	a + b = 2
Formula	Quantifiers,	Atom, Connectives	$\exists y(\forall x(y \geq x))$

Logic is the language in which the diagnostic information is expressed. It is according to the rules of logic that the diagnostic statements are deduced from the conflicts.

A formula may be represented in an infinite number of equivalent forms. From a human point of view most of these forms are very difficult to make sense of. This is one of the reasons the *normal forms*, [3], are important.

3.1 Normal forms

Transforming a formula to a normal form reduces the redundant information and improves readability. A two-component subsystem is introduced to exemplify formulas in normal forms.

Example 3.1
A two-component subsystem with four submodels M_i.

$$\text{Components:} \begin{cases} C_1,\ \Upsilon_{C_1} = \{NF_{C_1}, F1_{C_1}, UF_{C_1}\} \\ C_2,\ \Upsilon_{C_2} = \{NF_{C_2}, F1_{C_2}, UF_{C_2}\} \end{cases}$$

$$\text{Assumptions:} \begin{cases} \text{ass } M_1 \simeq B_{C_1} = NF_{C_1} \wedge B_{C_2} = NF_{C_2} \\ \text{ass } M_2 \simeq B_{C_1} = F1_{C_1} \wedge B_{C_2} = F1_{C_2} \\ \text{ass } M_3 \simeq B_{C_1} = F1_{C_1} \\ \text{ass } M_4 \simeq B_{C_2} = F1_{C_2} \end{cases}$$

Assume M_1, M_2 and M_3 have been invalidated in hypothesis tests $\Longrightarrow$

$$\neg\text{ass } M_1 \wedge \neg\text{ass } M_2 \wedge \neg\text{ass } M_3$$
$$\simeq$$
$$\neg(B_{C_1} = NF_{C_1} \wedge B_{C_2} = NF_{C_2}) \wedge$$
$$\neg(B_{C_1} = F1_{C_1} \wedge B_{C_2} = F1_{C_2}) \wedge$$
$$\neg(B_{C_2} = F1_{C_2})$$
$$\simeq_{\Phi_{C_1,C_2}}$$
$$((B_{C_1} = F1_{C_1} \vee B_{C_1} = UF_{C_1}) \vee (B_{C_2} = F1_{C_2} \vee B_{C_2} = UF_{C_2})) \wedge$$
$$((B_{C_1} = NF_{C_1} \vee B_{C_1} = UF_{C_1}) \vee (B_{C_2} = NF_{C_2} \vee B_{C_2} = UF_{C_2})) \wedge$$
$$(B_{C_2} = NF_{C_2} \vee B_{C_2} = UF_{C_2})$$

The last formula is in conjunctive normal form (cnf) [3], i.e. a conjunction of disjunctions. It is not intuitively easy to make out the information contents.

The resulting formula when evaluating a conjunction of negated conflicts will often be in cnf. It will not be easy to draw conclusions directley from that formula. However, if it is transposed to disjunctive normal form (dnf) [3], i.e. a disjunction of conjunctions, drawing conclusions will be easier, as the next example will illustrate.

Example 3.2
The formula in Example 3.1 transposed to disjunctive normal form.

Refer to [3] for the principle of how to transpose a general formula to dnf.

$$
\begin{gathered}
((B_{C_1} = F1_{C_1} \vee B_{C_1} = UF_{C_1}) \vee (B_{C_2} = F1_{C_2} \vee B_{C_2} = UF_{C_2})) \wedge \\
((B_{C_1} = NF_{C_1} \vee B_{C_1} = UF_{C_1}) \vee (B_{C_2} = NF_{C_2} \vee B_{C_2} = UF_{C_2})) \wedge \\
(B_{C_2} = NF_{C_2} \vee B_{C_2} = UF_{C_2}) \\
\simeq_{\Phi_{C_1,C_2}} \\
B_{C_2} = UF_{C_2} \vee \\
(B_{C_1} = F1_{C_1} \wedge B_{C_2} = NF_{C_2}) \vee \\
(B_{C_1} = UF_{C_1} \wedge B_{C_2} = NF_{C_2})
\end{gathered}
$$

There are three behavioral mode assignments grouped in a disjunction. The first assigns a behavioral mode only to one of the components.

When transposing the diagnostic information to dnf not all conjunctions assign behavior modes to all components. By deducing the *candidates* this inconsistency is eliminated. To view the candidates, the diagnostic information should be expressed in full disjunctive normal form (full dnf) [3]. However, before proceeding it is necessary to define what is meant by a candidate.

Definition 3.1. Candidate
A behavioral mode assignment on disjunctive normal form, $\mathbf{C}$*, over the components* $C_1, ..., C_n$*, is a candidate if it assigns a behavioral mode to every* C_i *and:*

$$
\mathbf{C} \wedge \Big(\bigwedge_{\forall i, T_i \in R_i} \neg \text{ass } M_i \Big) \simeq_{\Phi_{C_1,\ldots,C_n}} \mathbf{C}
$$

□

The process of deducing the candidates which are represented by the full dnf is illustrated in the following example.

Example 3.3

Deducing the candidates by transposing the formula in Example 3.1 to full disjunctive normal form, over C_1 and C_2. Refer to [3] for details.

$$
\begin{gathered}
B_{C_2} = UF_{C_2} \vee \\
(B_{C_1} = F1_{C_1} \wedge B_{C_2} = NF_{C_2}) \vee \\
(B_{C_1} = UF_{C_1} \wedge B_{C_2} = NF_{C_2}) \\
\simeq_{\Phi_{C_1,C_2}} \\
(B_{C_1} = NF_{C_1} \wedge B_{C_2} = UF_{C_2}) \vee \\
(B_{C_1} = F1_{C_1} \wedge B_{C_2} = UF_{C_2}) \vee \\
(B_{C_1} = F1_{C_1} \wedge B_{C_2} = NF_{C_2}) \vee \\
(B_{C_1} = UF_{C_1} \wedge B_{C_2} = UF_{C_2}) \vee \\
(B_{C_1} = UF_{C_1} \wedge B_{C_2} = NF_{C_2})
\end{gathered}
$$

When transposing the diagnostic information from Example 3.1 to full dnf the result is the five candidates represented by the formula above.

As seen above, expressing the diagnostic information in Example 3.1 in ordinary logic requires ten atoms in cnf, five atoms in dnf and ten atoms in full dnf. When diagnosing more complex systems, the size of the corresponding formulas will grow exponentially. Representing these in ordinary logic may prove inefficient.

3.2 A domain expansion to ordinary logic

The basic idea for creating a more efficient representation is grouping together certain atoms (Table 3.1). Instead of representing a disjunction of behavioral mode assignments of one component with a number of atoms, one generalized atom assigns the behavioral mode of the component to a *behavioral mode domain*, as shown in the following example.

Example 3.4
Example 3.1 continued.

$$\begin{gathered}(B_{C_1} = F1_{C_1} \vee B_{C_1} = UF_{C_1}) \\ \simeq \\ B_{C_1} \in \{F1_{C_1}, UF_{C_1}\}\end{gathered}$$

Where $\{F1_{C_1}, UF_{C_1}\}$ is an example of a behavioral mode domain.

What is actually a disjunction of atoms in ordinary logic is represented by a *generalized atom* in the new logic, in this thesis called *domain logic*.

Definition 3.2. Generalized atom in domain logic

$$B_C \in \bigcup_{\forall BM_C^i \in \alpha \subseteq \Upsilon_C} \{BM_C^i\} \simeq \bigvee_{\forall BM_C^i \in \alpha \subseteq \Upsilon_C} B_C = BM_C^i$$

□

This definition is the basis for expressing formulas in cnf and dnf in domain logic, which will be shown in the next two sections.

3.2.1 Conjunctive normal form in domain logic

Definition 3.3. Conjunctive normal form in domain logic
A formula is in conjunctive normal form in domain logic, over $C_1, ..., C_n$*,*

if it is a conjunction of disjunctions of generalized atoms (definition 3.2), where every disjunction assigns behavior of every C_i to a behavioral mode domain once or not at all.
□

The following example illustrates a formula expressed in cnf in domain logic.

Example 3.5
Formula in Example 3.1 expressed in domain logic.

$$((B_{C_1} = F1_{C_1} \vee B_{C_1} = UF_{C_1}) \vee (B_{C_2} = F1_{C_2} \vee B_{C_2} = UF_{C_2})) \wedge$$
$$((B_{C_1} = NF_{C_1} \vee B_{C_1} = UF_{C_1}) \vee (B_{C_2} = NF_{C_2} \vee B_{C_2} = UF_{C_2})) \wedge$$
$$(B_{C_2} = NF_{C_2} \vee B_{C_2} = UF_{C_2})$$
$$\simeq$$
$$(B_{C_1} \in \{F1_{C_1}, UF_{C_1}\} \vee B_{C_2} \in \{F1_{C_2}, UF_{C_2}\}) \wedge$$
$$(B_{C_1} \in \{NF_{C_1}, UF_{C_1}\} \vee B_{C_2} \in \{NF_{C_2}, UF_{C_2}\}) \wedge$$
$$B_{C_2} \in \{NF_{C_2}, UF_{C_2}\}$$

The representation in domain logic is more compact.

It is possible to define full cnf in domain logic but it has no use in the framework presented in this thesis. The dnf and full dnf on the other hand are integral parts.

3.2.2 Disjunctive normal form in domain logic

Definition 3.4. Disjunctive normal form in domain logic
A formula is in disjunctive normal form in domain logic, over $C_1, ..., C_n$, if it is a disjunction of conjunctions of generalized atoms (definition 3.2), where every conjunction assigns every the behavior of every C_i to a behavioral mode domain once or not at all.
□

The following example illustrates transforming a formula expressed in cnf in domain logic to dnf in domain logic.

Example 3.6
The formula in Example 3.5 transformed to disjunctive normal form in domain logic, according to the principle presented in Appendix B. The numbers refer to the subprocesses in Appendix B.

1: Evaluation of negations, done in Example 3.1.
2: Moving all occurrences of $\wedge$ inside the occurrences of $\vee$:

$$
\begin{gathered}
(B_{C_1} \in \{F1_{C_1}, UF_{C_1}\} \vee B_{C_2} \in \{F1_{C_2}, UF_{C_2}\}) \wedge \\
(B_{C_1} \in \{NF_{C_1}, UF_{C_1}\} \vee B_{C_2} \in \{NF_{C_2}, UF_{C_2}\}) \wedge \\
B_{C_2} \in \{NF_{C_2}, UF_{C_2}\} \\
\simeq \\
(B_{C_1} \in \{F1_{C_1}, UF_{C_1}\} \wedge B_{C_1} \in \{NF_{C_1}, UF_{C_1}\} \wedge B_{C_2} \in \{NF_{C_2}, UF_{C_2}\}) \vee \\
(B_{C_1} \in \{F1_{C_1}, UF_{C_1}\} \wedge B_{C_2} \in \{NF_{C_2}, UF_{C_2}\} \wedge B_{C_2} \in \{NF_{C_2}, UF_{C_2}\}) \vee \\
(B_{C_2} \in \{F1_{C_2}, UF_{C_2}\} \wedge B_{C_1} \in \{NF_{C_1}, UF_{C_1}\} \wedge B_{C_2} \in \{NF_{C_2}, UF_{C_2}\}) \vee \\
(B_{C_2} \in \{F1_{C_2}, UF_{C_2}\} \wedge B_{C_2} \in \{NF_{C_2}, UF_{C_2}\} \wedge B_{C_2} \in \{NF_{C_2}, UF_{C_2}\})
\end{gathered}
$$

3: Eliminate occurrences of multiple behavior mode assignments for the same component in a conjunction:

$$
\begin{gathered}
\simeq_{\Phi_{C_1, C_2}} \\
(B_{C_1} \in \{UF_{C_1}\} \wedge B_{C_2} \in \{NF_{C_2}, UF_{C_2}\}) \vee \\
(B_{C_1} \in \{F1_{C_1}, UF_{C_1}\} \wedge B_{C_2} \in \{NF_{C_2}, UF_{C_2}\}) \vee \\
(B_{C_1} \in \{NF_{C_1}, UF_{C_1}\} \wedge B_{C_2} \in \{UF_{C_2}\}) \vee \\
B_{C_2} \in \{UF_{C_2}\}
\end{gathered}
$$

4: Removing conjunctions assigning an empty set of behavior modes. No such assignments in this case.
5: Removing multiple occurrences of the same conjunction. No such assignments in this case.
6: Removing redundant information. How this is done for formulas on full dnf in domain logic is indicated in Appendix C and Examples 5.3 and 5.4.

$$
\begin{gathered}
\simeq \\
B_{C_2} \in \{UF_{C_2}\} \vee \\
(B_{C_1} \in \{F1_{C_1}, UF_{C_1}\} \wedge B_{C_2} \in \{NF_{C_2}\})
\end{gathered}
$$

Compared to the the formula on dnf in Example 3.2 the equivalent formula above is more compact.

To deduce the domain logic representation of the candidates, the diagnostic information should be expressed in full disjunctive normal form in domain logic.

3.2.3 Full normal disjunctive form in domain logic

Definition 3.5. Full disjunctive normal form in domain logic
A formula is in full disjunctive normal form in domain logic, over $C_1, ..., C_n$*, if it is a disjunction of conjunctions of generalized atoms (definition 3.2),where every conjunction assigns behavior of every* C_i *to a behavioral mode domain exactly once.*
□

Stating that a component is in anyone of it's possible behavioral modes is always true (Rule A.5). Therefore it is easy to extend a formula in dnf to full dnf in domain logic. The following example illustrates a method for extending a formula on dnf to full dnf, thereby deducing the domain logic representation of the candidates.

Example 3.7
Extending the formula in Example 3.6 to full disjunctive normal form.

$$\begin{gathered} B_{C_2} \in \{UF_{C_2}\} \vee \\ (B_{C_1} \in \{F1_{C_1}, UF_{C_1}\} \wedge B_{C_2} \in \{NF_{C_2}\}) \\ \simeq_{\Phi_{C_1,C_2}} \\ (B_{C_1} \in \Upsilon_{C_1} \wedge B_{C_2} \in \{UF_{C_2}\}) \vee \\ (B_{C_1} \in \{F1_{C_1}, UF_{C_1}\} \wedge B_{C_2} \in \{NF_{C_2}\}) \\ \simeq \\ (B_{C_1} \in \{NF_{C_1}, F1_{C_1}, UF_{C_1}\} \wedge B_{C_2} \in \{UF_{C_2}\}) \vee \\ (B_{C_1} \in \{F1_{C_1}, UF_{C_1}\} \wedge B_{C_2} \in \{NF_{C_2}\}) \end{gathered}$$

By incorporating a behavioral mode domain assignment, assigning all possible behavioral modes to a component which had no prior assignment, the formula is extended to full dnf in domain logic.

Extending a formula on dnf in domain logic is easier than doing the same operation in ordinary logic. This is only one of the benefits of domain logic.

3.2.4 The benefits of domain logic

To illustrate the benifits of domain logic, the next example summarizes the different representations of the diagnostic information from example 3.1.

Example 3.8
The representations of the diagnostic information from Example 3.1, in ordinary logic and domain logic.

Representations in ordinary logic:

$$\begin{gathered}(B_{C_1} = F1_{C_1} \lor B_{C_1} = UF_{C_1} \lor B_{C_2} = F1_{C_2} \lor B_{C_2} = UF_{C_2}) \land \\ (B_{C_1} = NF_{C_1} \lor B_{C_1} = UF_{C_1} \lor B_{C_2} = NF_{C_2} \lor B_{C_2} = UF_{C_2}) \land \\ B_{C_2} = NF_{C_2} \lor B_{C_2} = UF_{C_2} \\ \simeq_{\Phi_{C_1,C_2}} \\ B_{C_2} = UF_{C_2} \lor \\ (B_{C_1} = F1_{C_1} \land B_{C_2} = NF_{C_2}) \lor \\ (B_{C_1} = UF_{C_1} \land B_{C_2} = NF_{C_2}) \\ \simeq_{\Phi_{C_1,C_2}} \\ (B_{C_1} = NF_{C_1} \land B_{C_2} = UF_{C_2}) \lor \\ (B_{C_1} = F1_{C_1} \land B_{C_2} = UF_{C_2}) \lor \\ (B_{C_1} = F1_{C_1} \land B_{C_2} = NF_{C_2}) \lor \\ (B_{C_1} = UF_{C_1} \land B_{C_2} = UF_{C_2}) \lor \\ (B_{C_1} = UF_{C_1} \land B_{C_2} = NF_{C_2})\end{gathered}$$

Representations in domain logic:

$$\begin{gathered}(B_{C_1} \in \{F1_{C_1}, UF_{C_1}\} \lor B_{C_2} \in \{F1_{C_2}, UF_{C_2}\}) \land \\ (B_{C_1} \in \{NF_{C_1}, UF_{C_1}\} \lor B_{C_2} \in \{NF_{C_2}, UF_{C_2}\}) \land \\ B_{C_2} \in \{NF_{C_2}, UF_{C_2}\} \\ \simeq_{\Phi_{C_1,C_2}} \\ B_{C_2} \in \{UF_{C_2}\} \lor \\ (B_{C_1} \in \{F1_{C_1}, UF_{C_1}\} \land B_{C_2} \in \{NF_{C_2}\}) \\ \simeq_{\Phi_{C_1,C_2}} \\ (B_{C_1} \in \{F1_{C_1}, NF_{C_1}, UF_{C_1}\} \land B_{C_2} \in \{UF_{C_2}\}) \lor \\ (B_{C_1} \in \{F1_{C_1}, UF_{C_1}\} \land B_{C_2} \in \{NF_{C_2}\})\end{gathered}$$

The cnf, dnf and full dnf representations in ordinary logic, of the formula in Example 3.1, required ten, five and ten atoms respectively. In domain logic these representations required five, three and four generalized atoms.

The representations of diagnostic information in domain logic are more compact than the corresponding representations in ordinary logic.

Furthermore, the deduction of the dnf representation of the diagnostic information is a complex operation and it is central to the method presented in this thesis. The computational complexity when deducing the dnf from a formula on cnf, (3.1), is highly dependant of the number of atoms needed to represent the formula on cnf.

$$\begin{gathered}\mathcal{O}(n_1 \cdot \ldots \cdot n_m) \\ m \text{ is the number of disjunctions in the formula on cnf} \\ n_i \text{ is the number of atoms or generalized atoms in each disjunction}\end{gathered} \tag{3.1}$$

Therefore the deductions of the dnf representation require less computations in domain logic. This makes domain logic an important internal representation when calculating candidates.

Chapter 4

Applying assumption based diagnostics

To visualize the process of applying the theories of assumption based diagnostics, a typical system in an engine is introduced.

4.1 Intake manifold with three components

The intake manifold, see Figure 4.1 for the basic layout, is represented by three components:

W : mass flow sensor
p : pressure sensor
T : throttle

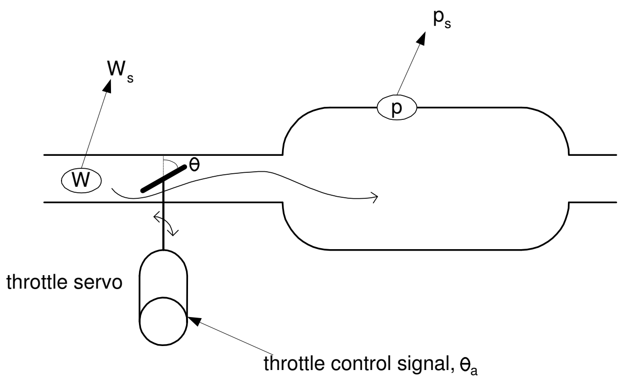

Figure 4.1: Intake manifold with throttle and two sensors

The components have the possible behavioral modes:

$$
\begin{aligned}
\Upsilon_W &= \{NF_W, SG_W, UF_W\} \\
\Upsilon_p &= \{NF_p, SG_p, UF_p\} \\
\Upsilon_T &= \{NF_T, SC_T, SO_T, S_T, UF_T\}
\end{aligned}
$$

NF : No fault

SG : Short to ground

SC : Stuck closed

SO : Stuck open

S : Stuck

UF : Unknown fault

4.2 Submodels and assumptions

The basic submodels describes the dynamics in the intake manifold and the behavior of the components.

Assumption	Basic submodel
$\emptyset$	$W - f(\theta, p) = 0$
$\emptyset$	$p - g(\theta, W) = 0$
$\emptyset$	$W_- \leq W \leq W_+$
$\emptyset$	$p_- \leq p \leq p_+$
$\emptyset$	$0° \leq \theta \leq 90°$
$B_W = NF_W$	$W_s = W$
$B_W = SG_W$	$W_s = 0\text{V}$
$B_p = NF_p$	$p_s = p$
$B_p = SG_p$	$p_s = 0\text{V}$
$B_T = NF_T$	$\theta_a = \theta$
$B_T = S_T$	$\theta = c$
$B_T = SO_T$	$\theta = 0°$
$B_T = SC_T$	$\theta = 90°$

The assumption $\emptyset$ means that the submodel always holds and c, $0° < c < 90°$, is a constant. The function f estimates the air mass flow and g estimates the air pressure in the intake manifold.

By utilizing the basic submodels a number of evaluable submodels can be deduced.

	Assumption	Evaluable submodel, M
1	$B_W = NF_W \wedge B_T = NF_T$	$p_- \leq g(\theta_a, W_s) \leq p_+$
2	$B_p = NF_p \wedge B_T = NF_T$	$W_- \leq f(\theta_a, p_s) \leq W_+$
3	$B_W = NF_W \wedge B_p = NF_p \wedge B_T = NF_T$	$W_s - f(\theta_a, p_s) = 0$
4	$B_W = NF_W \wedge B_p = NF_p \wedge B_T = S_T$	$W_s - f(c, p_s) = 0$
5	$B_W = NF_W \wedge B_p = NF_p \wedge B_T = SO_T$	$W_s - f(0, p_s) = 0$
6	$B_W = NF_W \wedge B_p = NF_p \wedge B_T = SC_T$	$W_s - f(90, p_s) = 0$

Note that M_5 and M_6 are deduced simply by manipulating the inputs to the function f. Also note that in a real application it would be wise to evaluate the submodels describing the behavior of the sensors when they are short cut. The reason for choosing not to, is that it makes the task of isolating faults more interesting.

4.3 Simulating faults

When a fault occurs, a number of evaluable submodels will be invalidated. This implies that the corresponding assumptions constitute conflicts. The conjunction of the negated conflicts represent the knowledge of the system-condition. The following example illustrates what diagnostic information could be attained if a certain fault occurred.

Example 4.1
The fault $B_p = SG_p$ is present and the subsystem is exited in a way such that the models M_2, M_3, M_4, M_5, M_6 are invalidated.

$$\begin{gathered}\neg\text{ass } M_2 \wedge \neg\text{ass } M_3 \wedge \neg\text{ass } M_4 \wedge \neg\text{ass } M_5 \wedge \neg\text{ass } M_6 \\ \simeq_{\Phi_{W,T,p}} \\ (B_T \in \{S_T, SC_T, SO_T, UF_T\} \vee B_p \in \{SG_p, UF_p\}) \wedge \\ (B_T \in \{S_T, SC_T, SO_T, UF_T\} \vee B_W \in \{SG_W, UF_W\} \vee B_p \in \{SG_p, UF_p\}) \wedge \\ (B_T \in \{NF_T, SC_T, SO_T, UF_T\} \vee B_W \in \{SG_W, UF_W\} \vee B_p \in \{SG_p, UF_p\}) \wedge \\ (B_T \in \{NF_T, S_T, SC_T, UF_T\} \vee B_W \in \{SG_W, UF_W\} \vee B_p \in \{SG_p, UF_p\}) \wedge \\ (B_T \in \{NF_T, S_T, SO_T, UF_T\} \vee B_W \in \{SG_W, UF_W\} \vee B_p \in \{SG_p, UF_p\})\end{gathered}$$

In the following example a double fault has occurred.

Example 4.2
The fault $B_W = SG_W \wedge B_p = SG_p$ is present and the subsystem is exited in a way such that the models M_1, M_2, M_3, M_4, M_5, M_6 are invalidated.

$$\begin{gathered}\neg\text{ass } M_1 \wedge \neg\text{ass } M_2 \wedge \neg\text{ass } M_3 \wedge \neg\text{ass } M_4 \wedge \neg\text{ass } M_5 \wedge \neg\text{ass } M_6 \\ \simeq_{\Phi_{W,T,p}} \\ (B_T \in \{S_T, SC_T, SO_T, UF_T\} \vee B_W \in \{SG_W, UF_W\}) \wedge \\ (B_T \in \{S_T, SC_T, SO_T, UF_T\} \vee B_p \in \{SG_p, UF_p\}) \wedge \\ (B_T \in \{S_T, SC_T, SO_T, UF_T\} \vee B_W \in \{SG_W, UF_W\} \vee B_p \in \{SG_p, UF_p\}) \wedge \\ (B_T \in \{NF_T, SC_T, SO_T, UF_T\} \vee B_W \in \{SG_W, UF_W\} \vee B_p \in \{SG_p, UF_p\}) \wedge \\ (B_T \in \{NF_T, S_T, SC_T, UF_T\} \vee B_W \in \{SG_W, UF_W\} \vee B_p \in \{SG_p, UF_p\}) \wedge \\ (B_T \in \{NF_T, S_T, SO_T, UF_T\} \vee B_W \in \{SG_W, UF_W\} \vee B_p \in \{SG_p, UF_p\})\end{gathered}$$

Finding different candidate representations is a task of refining the formulas in examples 4.1 and 4.2.

Chapter 5

Candidate representations

Throughout this chapter, results from Chapter 4 will be used to exemplify deduction of candidate representations. Also note that throughout this chapter a behavioral mode assignment will be written: α_{C_i} or BM_{C_i} instead of $B_{C_i} \in \alpha_{C_i}$ or $B_{C_i} = BM_{C_i}$, due to practical reasons.

As indicated earlier the deduction of candidates commence with the evaluation of the negated conflicts.

Example 5.1
Formula representing the evaluated negated conflicts, Example 4.1 continued.

$$\begin{gathered}(\{S_T, SC_T, SO_T, UF_T\} \vee \{SG_p, UF_p\}) \wedge \\ (\{NF_T, S_T, SC_T, UF_T\} \vee \{SG_W, UF_W\} \vee \{SG_p, UF_p\}) \wedge \\ (\{NF_T, S_T, SO_T, UF_T\} \vee \{SG_W, UF_W\} \vee \{SG_p, UF_p\}) \wedge \\ (\{NF_T, SC_T, SO_T, UF_T\} \vee \{SG_W, UF_W\} \vee \{SG_p, UF_p\}) \wedge \\ (\{S_T, SC_T, SO_T, UF_T\} \vee \{SG_W, UF_W\} \vee \{SG_p, UF_p\})\end{gathered}$$

Viewing the above formula is not easy to draw any conclusions.

Viewing the conjunction of negated conflicts it is not easy to draw relevant conclusions. To reveal the actual information content, the formula is transformed to disjunctive normal form in domain logic. This is done according to the principle presented in Appendix B and it is illustrated in the next example.

Example 5.2
Formula in Example 5.1 transformed to dnf in domain logic and ordi-

nary logic.

$$\begin{array}{c} \{UF_T\} \vee \\ \{SG_p, UF_p\} \vee \\ (\{S_T, SC_T, SO_T\} \wedge \{SG_W, UF_W\}) \\ \simeq \\ UF_T \vee \\ SG_p \vee \\ UF_p \vee \\ (SC_T \wedge SG_W) \vee \\ (SC_T \wedge UF_W) \vee \\ (SO_T \wedge SG_W) \vee \\ (SO_T \wedge UF_W) \vee \\ (S_T \wedge SG_W) \vee \\ (S_T \wedge UF_W) \end{array}$$

It is clear that there are three single and faults and six double faults that explain the conflicts.

Transforming the formula representing the negated conflicts to dnf in domain logic is crucial to this diagnostics approach. It is also the operation which requires the most computations. After the transformation, a number of representations can be deduced.

5.1 Candidates

The candidates are represented by the full dnf representation of the diagnostic information, as earlier stated in Section 3.2.3.

Since the observations often contain too little information to focus the search for the interesting diagnostic statements, there are often many candidates.

Example 5.3
Deducing full candidates in domain logic, according to the method

indicated in Section 3.2.3.

$$
\begin{gathered}
\{UF_T\} \vee \\
\{SG_p, UF_p\} \vee \\
(\{S_T, SC_T, SO_T\} \wedge \{SG_W, UF_W\}) \\
\simeq_{\Phi_{W,T,p}} \\
(\{UF_T\} \wedge \Upsilon_W \wedge \Upsilon_p) \\
(\Upsilon_T \wedge \Upsilon_W \wedge \{SG_p, UF_p\}) \vee \\
(\{S_T, SC_T, SO_T\} \wedge \{SG_W, UF_W\} \wedge \Upsilon_p) \ \vee \\
\simeq \\
(\{UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{NF_p, SG_p, UF_p\}) \vee \\
(\{NF_T, S_T, SC_T, SO_T, UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{SG_p, UF_p\}) \vee \\
(\{S_T, SC_T, SO_T\} \wedge \{SG_W, UF_W\} \wedge \{NF_p, SG_p, UF_p\})
\end{gathered}
$$

The last formula represents the candidates, however, it contains some redundant information. When two of the conjunctions are transformed to ordinary logic it is apparent that some candidates are represented more than once, for example:

$$(\{\underline{UF_T}\} \wedge \{\underline{NF_W}, \underline{SG_W}, \underline{UF_W}\} \wedge \{NF_p, \underline{SG_p}, \underline{UF_p}\})$$
$$\simeq$$
$$\begin{array}{c}
(B_T = UF_T \wedge B_W = NF \wedge B_p = NF_p) \vee \\
\underline{(B_T = UF_T \wedge B_W = NF_W \wedge B_p = S_pG)} \vee \\
\underline{(B_T = UF_T \wedge B_W = NF_W \wedge B_p = UF_p)} \vee \\
(B_T = UF_T \wedge B_W = SG \wedge B_p = NF_p) \vee \\
\underline{(B_T = UF_T \wedge B_W = SG_W \wedge B_p = SG_p)} \vee \\
\underline{(B_T = UF_T \wedge B_W = SG_W \wedge B_p = UF_p)} \vee \\
\underline{(B_T = UF_T \wedge B_W = UF_W \wedge B_p = NF_p)} \vee \\
\underline{(B_T = UF_T \wedge B_W = UF_W \wedge B_p = SG_p)} \vee \\
\underline{(B_T = UF_T \wedge B_W = UF_W \wedge B_p = UF_p)} \vee
\end{array}$$

$$(\{NF_T, S_T, SC_T, SO_T, \underline{UF}\} \wedge \{\underline{NF_W}, \underline{SG_W}, \underline{UF_W}\} \wedge \{\underline{SG_p}, \underline{UF_p}\})$$
$$\simeq$$
$$\begin{array}{c}
(B_T = NF_T \wedge B_W = NF_W \wedge B_p = SG_p) \vee \\
(B_T = NF_T \wedge B_W = NF_W \wedge B_p = SG_p) \vee \\
(B_T = NF_T \wedge B_W = NF_W \wedge B_p = SG_p) \vee \\
(B_T = NF_T \wedge B_W = SG_W \wedge B_p = UF_p) \vee \\
(B_T = NF_T \wedge B_W = SG_W \wedge B_p = UF_p) \vee \\
(B_T = NF_T \wedge B_W = SG_W \wedge B_p = UF_p) \vee \\
\vdots \\
\underline{(B_T = UF_T \wedge B_W = NF_W \wedge B_p = SG_p)} \vee \\
\underline{(B_T = UF_T \wedge B_W = NF_W \wedge B_p = UF_p)} \vee \\
\underline{(B_T = UF_T \wedge B_W = SG_W \wedge B_p = SG_p)} \vee \\
\underline{(B_T = UF_T \wedge B_W = SG_W \wedge B_p = UF_p)} \vee \\
\underline{(B_T = UF_T \wedge B_W = UF_W \wedge B_p = SG_p)} \vee \\
\underline{(B_T = UF_T \wedge B_W = UF_W \wedge B_p = UF_p)}
\end{array}$$

The underlined candidates are represented more than once.

When deducing the candidates according to the method indicated in Section 3.2.3 certain candidates will be represented twice. Therefore it is necessary to eliminate this redundancy. This process will be illustrated in the next example.

Example 5.4

There are two equivalent redundance free representations of the formula in Example 5.1. The algorithm used to find and eliminate redundancies in formulas on dnf in domain logic is presented in Appendix C. The algorithm takes a conjunction, $\mathbb{C}$, and checks whether there is any candidates represented twice that can be removed form the other conjunctions, $\mathbb{C}'$. This process then repeated so all conjunctions is chosen to be $\mathbb{C}$.

$$
\begin{gathered}
\mathbb{C}: \quad (\{UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{NF_p, SG_p, UF_p\}) \vee \\
\mathbb{C}': \quad (\{NF_T, S_T, SC_T, SO_T, \underline{UF_T}\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{SG_p, UF_p\}) \vee \\
(\{S_T, SC_T, SO_T\} \wedge \overline{\{SG_W}, UF_W\} \wedge \{NF_p, SG_p, UF_p\}) \\
\simeq \\
\mathbb{C}: \quad (\{UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{NF_p, SG_p, UF_p\}) \vee \\
(\{NF_T, S_T, SC_T, SO_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{SG_p, UF_p\}) \vee \\
\mathbb{C}': \quad (\{S_T, SC_T, SO_T\} \wedge \{SG_W, UF_W\} \wedge \{NF_p, SG_p, UF_p\}) \\
\simeq \\
\mathbb{C}': \quad (\{UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{NF_p, SG_p, UF_p\}) \vee \\
\mathbb{C}: \quad (\{NF_T, S_T, SC_T, SO_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{SG_p, UF_p\}) \vee \\
(\{S_T, SC_T, SO_T\} \wedge \{SG_W, UF_W\} \wedge \{NF_p, SG_p, UF_p\}) \\
\simeq \\
(\{UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{NF_p, SG_p, UF_p\}) \vee \\
\mathbb{C}: \quad (\{NF_T, S_T, SC_T, SO_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{SG_p, UF_p\}) \vee \\
\mathbb{C}': \quad (\{S_T, SC_T, SO_T\} \wedge \{SG_W, UF_W\} \wedge \{NF_p, \underline{SG_p}, \underline{UF_p}\}) \\
\simeq \\
\vdots \\
\text{No more redundancy will be found} \\
\simeq \\
(\{UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{NF_p, SG_p, UF_p\}) \vee \\
(\{NF_T, S_T, SC_T, SO_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{SG_p, UF_p\}) \vee \\
(\{S_T, SC_T, SO_T\} \wedge \{SG_W, UF_W\} \wedge \{NF_p\})
\end{gathered}
$$

If $\mathbb{C}$ and $\mathbb{C}$' is chosen differently in the beginning, the result will be different.

$$
\begin{gathered}
\mathbb{C}': \quad (\{UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{NF_p, \underline{SG_p}, \underline{UF_p}\}) \vee \\
\mathbb{C}: \quad (\{NF_T, S_T, SC_T, SO_T, UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{SG_p, UF_p\}) \vee \\
(\{S_T, SC_T, SO_T\} \wedge \{SG_W, UF_W\} \wedge \{NF_p, SG_p, UF_p\}) \\
\simeq \\
\vdots \\
\simeq \\
(\{UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{NF_p\}) \vee \\
(\{NF_T, S_T, SC_T, SO_T, UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{SG_p, UF_p\}) \vee \\
(\{S_T, SC_T, SO_T\} \wedge \{SG_W, UF_W\} \wedge \{NF_p\})
\end{gathered}
$$

Hence there are two equivalent redundancy free full dnf representations.

As shown above, the full dnf representation of the diagnostic information is generally not unique.

The next example shows the candidates represented in ordinary logic.

Example 5.5
Full candidates in ordinary logic, Example 5.4 continued.

$$
\begin{gathered}
(NF_T \wedge NF_W \wedge SG_p) \vee \\
(NF_T \wedge NF_W \wedge UF_p) \vee \\
(NF_T \wedge SG_W \wedge SG_p) \vee \\
(NF_T \wedge SG_W \wedge UF_p) \vee \\
(NF_T \wedge UF_W \wedge SG_p) \vee \\
(NF_T \wedge UF_W \wedge UF_p) \vee \\
(SC_T \wedge NF_W \wedge SG_p) \vee \\
(SC_T \wedge NF_W \wedge UF_p) \vee \\
(SC_T \wedge SG_W \wedge NF_p) \vee \\
(SC_T \wedge SG_W \wedge SG_p) \vee \\
(SC_T \wedge SG_W \wedge UF_p) \vee \\
(SC_T \wedge UF_W \wedge NF_p) \vee \\
(SC_T \wedge UF_W \wedge SG_p) \vee \\
(SC_T \wedge UF_W \wedge UF_p) \vee \\
(SO_T \wedge NF_W \wedge SG_p) \vee \\
(SO_T \wedge NF_W \wedge UF_p) \vee \\
(SO_T \wedge SG_W \wedge NF_p) \vee \\
(SO_T \wedge SG_W \wedge SG_p) \vee \\
(SO_T \wedge SG_W \wedge UF_p) \vee \\
(SO_T \wedge UF_W \wedge NF_p) \vee \\
(SO_T \wedge UF_W \wedge SG_p) \vee \\
(SO_T \wedge UF_W \wedge UF_p) \vee \\
(S_T \wedge NF_W \wedge SG_p) \vee \\
(S_T \wedge NF_W \wedge UF_p) \vee \\
(S_T \wedge SG_W \wedge NF_p) \vee \\
(S_T \wedge SG_W \wedge SG_p) \vee \\
(S_T \wedge SG_W \wedge UF_p) \vee \\
(S_T \wedge UF_W \wedge NF_p) \vee \\
(S_T \wedge UF_W \wedge SG_p) \vee \\
(S_T \wedge UF_W \wedge UF_p) \vee \\
(UF_T \wedge NF_W \wedge NF_p) \vee \\
(UF_T \wedge NF_W \wedge SG_p) \vee \\
(UF_T \wedge NF_W \wedge UF_p) \vee \\
(UF_T \wedge SG_W \wedge NF_p) \vee \\
(UF_T \wedge SG_W \wedge SG_p) \vee \\
(UF_T \wedge SG_W \wedge UF_p) \vee \\
(UF_T \wedge UF_W \wedge NF_p) \vee \\
(UF_T \wedge UF_W \wedge SG_p) \vee \\
(UF_T \wedge UF_W \wedge UF_p)
\end{gathered}
$$

There are 39 candidates. For a system with only three components this is not very conclusive.

The candidate representation does not focus on the interesting diagnostic statements to a satisfactory level. It is obvious that more focused representations are needed.

5.2 Generalized kernel candidates

The first step in focusing on more interesting diagnostic statements is to search for the kernel candidates. For a definition of kernel candidates please refer to [2]. The basic idea is to omit to define the behaviorial modes for those components that may be in any mode.

Finding the kernel candidates in the general case is hard. Furthermore the kernel candidate representation is not conclusive enough to focus on the interesting diagnostic statements in the general case. The example below illustrates this problem.

Example 5.6
Generalized kernel candidates. In this case these are found by reverting to the first formula of Example 5.4.

$$\begin{array}{c}
(\{UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{NF_p, SG_p, UF_p\}) \vee \\
(\{NF_T, S_T, SC_T, SO_T, UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{SG_p, UF_p\}) \vee \\
(\{S_T, SC_T, SO_T\} \wedge \{SG_W, UF_W\} \wedge \{NF_p, SG_p, UF_p\}) \\
\simeq \\
(\{UF_T\} \wedge \Upsilon_W \wedge \Upsilon_p) \vee \\
(\Upsilon_T \wedge \Upsilon_W \wedge \{SG_p, UF_p\}) \vee \\
(\{S_T, SC_T, SO_T\} \wedge \{SG_W, UF_W\} \wedge \Upsilon_p) \\
\simeq \text{ (rule A.5)} \\
\{UF_T\} \vee \\
\{SG_p, UF_p\} \vee \\
(\{S_T, SC_T, SO_T\} \wedge \{SG_W, UF_W\})
\end{array}$$

There are nine true kernel candidates and these are unique. In this case the kernel candidates are represented by the same formula as the dnf representation of the negated conflicts (Example 5.2), this is not true in the general case. However, nine candidates is not conclusive enough for this small system.

Finding the generalized kernel candidates is hard. An automated process would require extensive computations. Furthermore, the kernel representation is often inconclusive. Therefore it is concluded that the deduction of generalized kernel candidates has a low priority.

5.3 Generalized minimal candidates

The concept of minimal diagnosis with components that have only two behavioral modes is well known [1], [2].

In this approach, deducing the generalized minimal candidates implies a differential prioritization between fault modes. Intuitively the behavioral mode NF is preferred before all other modes and all behavioral modes are preferred before UF, as shown in Equation (5.1). The reason for the prioritization of the unknown-fault mode is that the most common faults should be known to get good performance of a diagnostic system.

$$NF_{C_i} \prec F^{j}_{C_i} \prec UF_{C_i} \tag{5.1}$$

Definition 5.1. Generalized minimal candidates
If $\mathbf{C}$ *is candidate over the components* $C_1, ..., C_n$*, assigning the behavioral modes* $BM_{C_1}, ..., BM_{C_n}$*. Then* $\mathbf{C}$ *is a generalized minimal candidate if there exists no candidate* $\mathbf{C}' \neq \mathbf{C}$ *for which:*

$$BM'_{C_i} \prec BM_{C_i} \text{ or } BM'_{C_i} \equiv BM_{C_i},\ \forall i$$

□

Generally a candidate is minimal if it assigns the least possible number of fault behavioral modes. However, since the unknown-fault mode has a special status it is a bit more complicated than that. The process of finding the generalized minimal candidates is illustrated in the following example.

Example 5.7
Generalized minimal candidates deduced from the full candidates in Example 5.4. This is done according to the principle presented in presented in Appendix D.
First the the candidates minimal for each conjunction are found. These candidates are then transformed to ordinary logic and compared to each

other to determine which are minimal.

$$
\begin{gathered}
(\{UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{NF_p\}) \vee \\
(\{NF_T, S_T, SC_T, SO_T, UF_T\} \wedge \{NF_W, SG_W, UF_W\} \wedge \{SG_p, UF_p\}) \vee \\
(\{S_T, SC_T, SO_T\} \wedge \{SG_W, UF_W\} \wedge \{NF_p\}) \\
\simeq_{NF_{C_i} \prec F^j_{C_i} \prec UF_{C_i}} \\
(\{UF_T\} \wedge \{NF_W\} \wedge \{NF_p\}) \vee \\
(\{NF_T\} \wedge \{NF_W\} \wedge \{SG_p\} \vee \\
(\{S_T, SC_T, SO_T\} \wedge \{SG_W\} \wedge \{NF_p\}) \\
\simeq \\
(NF_T \wedge NF_W \wedge SG_p) \vee \\
(SC_T \wedge SG_W \wedge NF_p) \vee \\
(SO_T \wedge SG_W \wedge NF_p) \vee \\
(S_T \wedge SG_W \wedge NF_p) \vee \\
(UF_T \wedge NF_W \wedge NF_p)
\end{gathered}
$$

There are five generalized minimal candidates, two of which assign the no fault mode to two components.

The generalized minimal candidate representation seems to focus on the interesting candidates to a certain level. However, presenting five alternative candidates would not be satisfactory in this case, even narrower focusing is needed. Two methods are represented in the following sections.

5.4 Probability prioritization

One method for narrower focusing is to present the most probable candidates. This requires estimates of probabilities for each fault mode.

The probability of a candidate, over $C_1, ..., C_n$, assigning the behavioral modes $BM_{C_1}, ..., BM_{C_n}$ is calculated as:

$$\prod_{i=1}^{n} p(BM_{C_i})$$

provided that the each behavior mode assignment is independent.

Presenting only the most probable candidate may prove hazardous. A better approach may be to present all generalized minimal candidates that have a probability higher than some probability threshold. E.g. present all minimal candidates that have at least 50% of the probability of the most probable minimal candidate.

Example 5.8

Probability prioritized candidates, Example 5.7 continued. Assume that $p(NF_{C_i}) \gg p(F^j_{C_i}) > p(UF_{C_i})$.

$$NF_T \wedge NF_W \wedge SG_p$$

The most probable candidate is in fact the candidate which describes the condition of the system.

The probability prioritization gives a satisfactory result.

5.5 Behavioral mode grading prioritization

Another method for prioritizing candidates is to give each behavioral mode a grading. For example:

$$G(NF_{Ci}) = 0$$
$$G(F^j_{Ci}) = 1$$
$$G(UF_{Ci}) = 2$$

The grading of a candidate, over $C_1, ..., C_n$, is calculated as:

$$\Sigma_{i=1}^{n} G(BM_{C_i})$$

The candidates with the lowest grading should then be presented. This is illustrated in the following example.

Example 5.9
Grading prioritized candidates, Example 5.7 continued. Assuming the grading: $G(NF_{Ci}) = 0, G(F^j_{Ci}) = 1, G(UF_{Ci}) = 2$, the behavioral mode grading prioritized candidate is:

$$NF_T \wedge NF_W \wedge SG_p$$

The suggested candidate describe the condition of the system.

5.6 The representation of candidates in Scania Diagnos

The current implementation of Scania Diagnos[1] shows a conjunction of fault codes, i.e negated conflicts. It can not show a disjunction of alternative candidates. If the framework presented in this thesis, or some similar principle, was to be used for deducing candidates, this restriction would prove problematic.

The information content in the observations often is not large enough to focus only on the correct candidate. The best guess will often be a

[1] For further information please refer to: http://www.scania.se/services/diagnos/

few alternative candidates. If it would be possible to present only one of the candidates it most certainly be the one with the highest probability. This would render that the presented candidate sometimes would not describe the actual system condition. This problem is illustrated in the following examples.

Example 5.10
Example 4.2 continued. Deduction of minimal candidates when the fault $SG_W \wedge SG_p$ is present i the subsystem presented in Chapter 4.2. Conjunction of negated conflicts:

$$
\begin{gathered}
(\{S_T, SC_T, SO_T, UF_T\} \vee \{SG_W, UF_W\}) \wedge \\
(\{S_T, SC_T, SO_T, UF_T\} \vee \{SG_p, UF_p\}) \wedge \\
(\{NF_T, S_T, SC_T, UF_T\} \vee \{SG_W, UF_W\} \vee \{SG_p, UF_p\}) \wedge \\
(\{NF_T, S_T, SO_T, UF_T\} \vee \{SG_W, UF_W\} \vee \{SG_p, UF_p\}) \wedge \\
(\{NF_T, SC_T, SO_T, UF_T\} \vee \{SG_W, UF_W\} \vee \{SG_p, UF_p\}) \wedge \\
(\{S_T, SC_T, SO_T, UF_T\} \vee \{SG_W, UF_W\} \vee \{SG_p, UF_p\})
\end{gathered}
$$

$$\Longrightarrow$$

Minimal candidates:

$$
\begin{gathered}
(NF_T \wedge SG_W \wedge SG_p) \vee \\
(S_T \wedge NF_W \wedge SG_p) \vee \\
(SC_T \wedge NF_W \wedge SG_p) \vee \\
(SO_T \wedge NF_W \wedge SG_p) \vee \\
(S_T \wedge SG_W \wedge NF_p) \vee \\
(SC_T \wedge SG_W \wedge NF_p) \vee \\
(SO_T \wedge SG_W \wedge NF_p) \vee \\
(UF_T \wedge NF_W \wedge NF_p)
\end{gathered}
$$

In this specific case there are eight generalized minimal candidates. If the intention was to show a candidate in today's version of Scania Diagnos, the one with the highest probability would be chosen.

Example 5.11
Example 5.10 continued. Candidate with the highest probability assuming that: $p(NF_{C_i}) \gg p(S_T) > p(SG_W) > p(SG_p) > p(SC_T) > p(SO_T) > p(UF_{C_i})$.

$$S_T \wedge SG_W \wedge NF_p$$

The most probable candidate does in fact not describe the actual system condition.

If instead that the technician in the workshop was allowed to view the alternative candidates or some selection there of, the process of finding the actual fault could be facilitated.

5.7 Concluding remarks

When designing a diagnostic system using the approach presented in this thesis, evaluating models based on fault behavior of components is crucial. Since the diagnostic statements are deduced from the conflicts, fault behavior modes must be part of these to be invalidated if components are non faulty.

As shown in the subsystem presented in Section 4.2, modelling faulty behavior may be quite easy. By changing some values, the same models may be used as when modelling normal behavior.

The candidate and generalized kernel candidate representations are inconclusive in the general case. The generalized minimal candidate representation, which is deduced from the candidate representation, seems to focus better on the interesting diagnostic statements. Therefore it is concluded that deduction the generalized kernel candidates has a low priority.

For further focusing either one of the two methods of probability prioritization or behavioral mode grading prioritization works well.

Chapter 6

Diagnosing an EGR system

6.1 Diagnostic tests used in the industry

Today in industry, the more recent theories of diagnostics are not widely incorporated. However, the tests used and the conclusions drawn bear some resemblance to the important concepts of assumption based diagnostics. A major difference is that in industry conclusions may be drawn even when models are not rejected, as opposed to the principle presented in Section 2.3.

Example 6.1

Example of diagnosing a sensor in industry and corresponding interpretation in assumption based diagnostics. There is also a suggestion of how it should ideally have been implemented in assumption based diagnostics.

Industry diagnostics

T: Temperature sensor, with behavioral modes:

$$\begin{aligned} \Upsilon_T &= \{NF_T, RH_T, RL_T\} \\ NF: &\quad \text{No fault} \\ RH: &\quad \text{Sensor value is out of range, high} \\ RL: &\quad \text{Sensor value is out of range, low} \end{aligned}$$

Note that the unknown-fault mode is not included.
There are two in-range tests for the temperature sensor:

Test 1

$$\text{Test decisions}, \delta_1 : \quad \begin{cases} 0: & NF_T \vee RL_T & \text{if } T_s \leq T_+ \\ 1: & RH_T & \text{if } T_+ < T_s \end{cases}$$

Test 2

$$\text{Test decisions}, \delta_2 : \quad \begin{cases} 0: & NF_T \vee RH_T & \text{if } T_- \leq T_s \\ 1: & RL_T & \text{if } T_s < T_- \end{cases}$$

$$\text{Diagnostic statement} : \quad \delta_1 \wedge \delta_2$$

Diagnostic statements for different test decisions δ_1 and δ_2:

δ_1	δ_2	$\delta_1 \wedge \delta_2$	Diagnostic statement
0	0	$(NF_T \vee RL_T) \wedge (NF_T \vee RH_T)$	NF_T
0	1	$(NF_T \vee RL_T) \wedge RL_T$	RL_T
1	0	$RH_T \wedge (NF_T \vee RH_T)$	RH_T
1	1	$RH_T \wedge RL_T$	$\emptyset$

In a deterministic sense, the case $\delta_1 = 1$ and $\delta_2 = 1$ cannot occur. However, in real life applications this could happened, e.g. due to noise or settings of thresholds.

Interpretation in assumption based diagnostics

$$\Upsilon_T = \{NF_T, RH_T, RL_T, UF_T\}$$

Assumption	**Basic submodel**
$\emptyset$	$T_- \leq T \leq T_+$
NF_T	$T_s = T$
RL_T	$T_s < T_-$
RH_T	$T_+ < T_s$

	Assumption	**Evaluable submodel**M
1	$NF_T \vee RL_T$	$T_s \leq T_+$
2	RH_T	$T_+ < T_s$
3	$NF_T \vee RH_T$	$T_- \leq T_s$
4	RL_T	$T_s < T_-$

$$\text{Diagnostic statement : } \bigwedge_{\forall i, T_i \in R_i} \neg \text{ass } M_i$$

(T_i and R_i are test decisions and rejection regions)

Diagnostic statements for different sets of invalidated assumptions:

$\neg M_1$	$\neg M_2$	$\neg M_3$	$\neg M_4$	Diagnostic statement
1	0	0	1	$RH_T \vee UF_T$
0	1	0	1	$NF_T \vee UF_T$
1	0	1	0	$RL_T \vee UF_T$

A number of cases cannot occur in the deterministic sense:

$\neg M_1$	$\neg M_2$	$\neg M_3$	$\neg M_4$	Diagnostic statement
0	0	0	0	Υ_T
0	0	0	1	$RH_T \vee UF_T$
0	0	1	0	$NF_T \vee RL_T \vee UF_T$
0	1	0	0	$RL_T \vee UF_T$
1	0	0	0	$NF_T \vee RH_T \vee UF_T$
⋮	⋮	⋮	⋮	UF_T

These may occur in real life applications, e.g. due to noise or settings of thresholds.

Ideal implementation in assumption based diagnostics

$$\Upsilon_T = \{NF_T, RH_T, RL_T, UF_T\}$$

Assumption	**Basic submodel**
$\emptyset$	$T_- \leq T \leq T_+$
NF_T	$T_s = T$
RL_T	$T_s < T_-$
RH_T	$T_+ < T_s$

	Assumption	**Evaluable submodel**M
1	NF_T	$T_- \leq T_s \leq T_+$
2	RH_T	$T_+ < T_s$
3	RL_T	$T_s < T_-$

$$\text{Diagnostic statement : } \bigwedge_{\forall i, T_i \in R_i} \neg \text{ass } M_i$$

(T_i and R_i are test decisions and rejection regions)

Diagnostic statements for different sets of invalidated assumptions:

$\neg M_1$	$\neg M_2$	$\neg M_3$	Diagnostic statement
0	1	1	$NF_T \vee UF_T$
1	0	1	$RH_T \vee UF_T$
1	1	0	$RL_T \vee UF_T$

A number of cases cannot occur in the deterministic sense:

$\neg M_1$	$\neg M_2$	$\neg M_3$	Diagnostic statement
0	0	0	Υ_T
0	0	1	$RH_T \vee RL_T \vee UF_T$
0	1	0	$NF_T \vee RL_T \vee UF_T$
1	0	0	$NF_T \vee RH_T \vee UF_T$
1	1	1	UF_T

These may occur in real life applications, e.g. due to noise or settings of thresholds.

Note that the unknown fault mode will never be invalidated since it can not be modelled.

In a typical diagnostic system used in the industry today, each sensor and component is diagnosed separately as far as possible. Unfortunately, as the number of components in the engine increase, this will be ineffective. By utilizing knowledge about how the components interact and available system-performance information, it will be possible to draw relevant conclusions.

An example of complex interaction between components is an EGR system, see Figure 6.1 for the basic layout.

6.2 EGR system

Future environmental laws will impose graver restrictions on the exhausts of engines in heavy trucks. Among other things, the level of NO_x must be lowered. One approach to solve this problem is to incorporate an EGR system.

A problem is that in most operating points the gas-pressure on the exhaust side of the engine is lower than on the intake side. One solution may be to incorporate a venturi.

The primary components such an EGR system are:

W: Massflow sensor
Tb: Boost temperature sensor

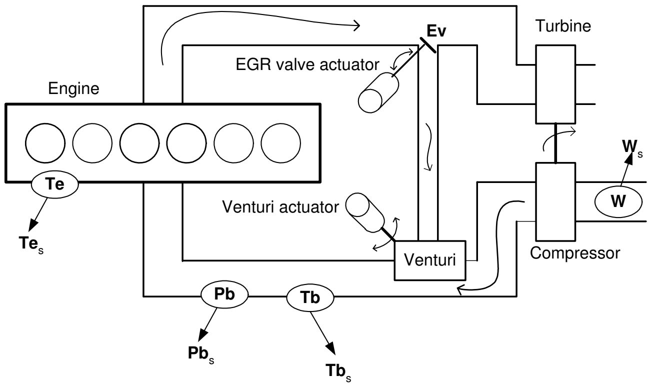

Figure 6.1: EGR-system with an EGR-valve, venturi and four sensors.

Te: Water temperature sensor
Pb: Boost pressure sensor
Ev: EGR valve
V: Venturi

The components have the possible behavioral modes:

$$
\begin{aligned}
\Upsilon_W &= \{NF_W, RH_W, RH_W, UF_W\} \\
\Upsilon_{Tb} &= \{NF_{Tb}, RH_{Tb}, RL_{Tb}, UF_{Tb}\} \\
\Upsilon_{Te} &= \{NF_{Te}, RH_{Te}, RL_{Te}, UF_{Te}\} \\
\Upsilon_{Pb} &= \{NF_{Pb}, RH_{Pb}, RL_{Pb}, UF_{Pb}\} \\
\Upsilon_{Ev} &= \{NF_{Ev}, S_{Ev}, SC_{Ev}, SO_{Ev}, UF_{Ev}\} \\
\Upsilon_V &= \{NF_V, SC_V, SO_V, UF_V\}
\end{aligned}
$$

NF : No fault
RH : Sensor value is out of range, high
RL : Sensor value is out of range, low
SC : Stuck closed
SO : Stuck open
S : Stuck
UF : Unknown fault

6.2.1 Submodels and assumptions

The submodels describe both normal and faulty behavior of the system. The engine speed, N, is assumed to originate from a fault-free sensor.

Assumption	**Basic submodel**
$\emptyset$	$Tb_- \leq Tb \leq Tb_+$
$\emptyset$	$Te_- \leq Te \leq Te_+$
$\emptyset$	$Pb_- \leq Pb \leq Pb_+$
$\emptyset$	$W_- \leq W_{air} \leq W_+$
$\emptyset$	$\%_{EGR} = \dfrac{W_{EGR}}{W_{cyl}} \cdot 100$
$\emptyset$	$g(\eta_{vol}, N, Pb, Tb) = W_{cyl}$
$\emptyset$	$h(Tb, Te, Pb, N) = \eta_{vol}$
$\emptyset$	$W_{EGR} = W_{cyl} - W_{air}$
NF_W	$f_- \leq f(\eta_{vol}, W_s, N, Pb, Tb) \leq f_+$
NF_{Tb}	$Tb_s = Tb$
NF_{Te}	$Te_s = Te$
NF_{Pb}	$Pb_s = Pb$
NF_W	$W_s \cdot f(\eta_{vol}, W_s, N, Pb, Tb) = W_{air}$
$(NF_{Ev} \vee SO_{Ev}) \wedge (NF_V \vee SC_V)$	$\%^d_{EGR} \leq \%_{EGR}$
$(NF_{Ev} \vee SC_{Ev}) \wedge (NF_V \vee SO_V)$	$\%_{EGR} \leq \%^d_{EGR}$

Including behavior-models of the EGR-valve and venturi is outside the scope of this thesis. Therefore the last two basic submodels are deduced intuitively by the author.

Some evaluable submodels are based on limit-checking and are derived in the way described in Example 6.1. These are grouped together and marked ia, ib and ic.

	Assumption	Evaluable submodel, M
1a	NF_W	$W_- \cdot f_- \leq W_s \leq W_+ \cdot f_+$
1b	RH_W	$W_+ < W_s$
1c	RL_W	$W_s < W_-$
2a	NF_{Tb}	$Tb_- \leq Tb_s \leq Tb_+$
2b	RH_{Tb}	$Tb_+ < Tb_s$
2c	RL_{Tb}	$Tb_s < Tb_-$
3a	NF_{Te}	$Te_- \leq Te_s \leq Te_+$
3b	RH_{Te}	$Te_+ < Te_s$
3c	RL_{Te}	$Te_s < Te_-$
4a	NF_{Pb}	$Pb_- \leq Pb_s \leq Pb_+$
4b	RH_{Pb}	$Pb_+ < Pb_s$
4c	RL_{Pb}	$Pb_s < Pb_-$
5	$NF_W \wedge NF_{Te} \wedge NF_{Pb} \wedge NF_{Tb}$	$f_- \leq f_a(h', W_s, N, Pb_s, Tb_s) \leq f_+$
6	$NF_W \wedge NF_{Te} \wedge NF_{Pb} \wedge NF_{Tb} \wedge$ $(NF_{Ev} \vee SO_{Ev}) \wedge (NF_V \vee SC_V)$	$\%^d_{EGR} \leq \frac{g' - W_s \cdot f'}{g'}$
7	$NF_W \wedge NF_{Te} \wedge NF_{Pb} \wedge NF_{Tb} \wedge$ $(NF_{Ev} \vee SC_{Ev}) \wedge (NF_V \vee SO_V)$	$\frac{g' - W_s \cdot f'}{g'} \leq \%^d_{EGR}$

Note that the sensors are assumed to be fault-free or in the unknown-fault mode as long as they are in-range. If the generalized minimal candidates are derived, the unknown-fault mode will be filtered out. This means that an in-range response that is faulty will not give rise to an alarm. For compliance with future environmental laws this is not good enough. To be able to find e.g. bias faults in a certain sensor, it is necessary to compare the measured value with some predicted value, derived without using the output of the sensor.

Also note the weakness of the modelling of this subsystem, there are few evaluable submodels for determining if the EGR-valve or the venturi is stuck.

6.2.2 Simulating and diagnosing faults

Sensor faults are the easiest to isolate in this system, as will be illustrated in the next example.

Example 6.2

There is a fault present, $RH_{Pb} \wedge RL_{Te}$. Provided that the system is optimally exited, the models: $M_{1b},\ M_{1c},\ M_{2b},\ M_{2c},\ M_{3a},\ M_{3b},\ M_{4a},\ M_{4c},\ M_5,\ M_7$ are invalidated.

$$\neg\text{ass } M_{1b} \wedge \neg\text{ass } M_{1c} \wedge \neg\text{ass } M_{2b} \wedge \neg\text{ass } M_{2c} \wedge \neg\text{ass } M_{3a} \wedge \\ \neg\text{ass } M_{3b} \wedge \neg\text{ass} M_{4a} \wedge \neg\text{ass } M_{4c} \wedge \neg\text{ass } M_5 \wedge \neg\text{ass } M_7$$

$$\simeq_{\Phi_{W,Te,Tb,Pb,V,Ev}}$$

$$\{NF_W, RL_W, UF_W\} \wedge \\ \{NF_W, RH_W, UF_W\} \wedge \\ \{NF_{Tb}, RL_{Tb}, UF_{Tb}\} \wedge \\ \{NF_{Tb}, RH_{Tb}, UF_{Tb}\} \wedge \\ \{RH_{Te}, RL_{Te}, UF_{Te}\} \wedge \\ \{NF_{Te}, RL_{Te}, UF_{Te}\} \wedge \\ \{RL_{Pb}, RH_{Pb}, UF_{Pb}\} \wedge \\ \{NF_{Pb}, RH_{Pb}, UF_{Pb}\} \wedge \\ (\{RH_{Pb}, RL_{Pb}, UF_{Pb}\} \vee \{RH_{Tb}, RL_{Tb}, UF_{Tb}\} \vee \\ \{RH_{Te}, RL_{Te}, UF_{Te}\} \vee \{RH_W, RL_W, UF_W\}) \wedge \\ (\{S_{Ev}, SO_{Ev}, UF_{Ev}\} \vee \{RH_{Pb}, RL_{Pb}, UF_{Pb}\} \vee \{RH_{Tb}, RL_{Tb}, UF_{Tb}\} \vee \\ \{RH_{Te}, RL_{Te}, UF_{Te}\} \vee \{SC_V, UF_V\} \vee \{RH_W, RL_W, UF_W\})$$

$$\simeq_{\Phi_{W,Te,Tb,Pb,V,Ev}}$$

Disjunctive normal form:

$$\{RH_{Pb}, UF_{Pb}\} \wedge \{NF_{Tb}, UF_{Tb}\} \wedge \{RL_{Te}, UF_{Te}\} \wedge \{NF_W, UF_W\}$$

$$\simeq_{\Phi_{W,Te,Tb,Pb,V,Ev}}$$

Full disjunctive normal form:

$$\{NF_{Ev}, S_{Ev}, SC_{Ev}, SO_{Ev}, UF_{Ev}\} \wedge \{RH_{Pb}, UF_{Pb}\} \wedge \{NF_{Tb}, UF_{Tb}\} \wedge \\ \{RL_{Te}, UF_{Te}\} \wedge \{NF_V, SC_V, SO_V, UF_V\} \wedge \{NF_W, UF_W\}$$

$$\Longrightarrow$$

Minimal candidate:

$$NF_{Ev} \wedge RH_{Pb} \wedge NF_{Tb} \wedge RL_{Te} \wedge NF_V \wedge NF_W$$

The generalized minimal candidate describe the actual system condition.

As shown in the above example isolating an out-of-range fault in a sensor is quite easy, provided that the models are ideally invalidated. The next example illustrates what happens when models are not ideally invalidated.

Example 6.3
The fault is the same as in Example 6.2, i.e. $RH_{Pb} \wedge RL_{Te}$. Ideally the same models should be invalidated. However, in real life applications invalidations of e.g. M_{2b} and M_{6a} could be missed due to noice.

$$\neg \text{ass } M_{1b} \wedge \neg \text{ass } M_{3b} \wedge \neg \text{ass } M_{4b} \wedge \neg \text{ass } M_{5b} \wedge$$
$$\neg \text{ass } M_{10a} \wedge \neg \text{ass } M_{8b} \wedge \neg \text{ass } M_{9} \wedge \neg \text{ass } M_{12}$$

$$\simeq_{\Phi_{W,Te,Tb,Pb,V,Ev}}$$
$$\vdots$$
$$\simeq_{\Phi_{W,Te,Tb,Pb,V,Ev}}$$

Disjunctive normal form:
$$\{RH_{Pb}, UF_{Pb}\} \wedge \{NF_{Tb}, UF_{Tb}\} \wedge \{NF_{Te}, RL_{Te}, UF_{Te}\} \wedge \{NF_{W}, RL_{W}, UF_{W}\}$$

$$\Longrightarrow$$

Minimal candidate:
$$NF_{Ev} \wedge RH_{Pb} \wedge NF_{Tb} \wedge \; NF_{Te} \wedge NF_{V} \wedge \; NF_{W}$$

The generalized minimal candidate include only one of the faults. Since the model M_{6a} was not invalidated, the fault in Te is not decisively indicated. The missed invalidation of model M_{2b} has no impact on the minimal candidate. The reason is that the behavioral mode NF_W is chosen before RL_W.

Not all falsely invalidated assumptions affect the result of isolating the faults. Missed invalidations have a greater impact.

The next example illustrates the task of isolating a fault in the EGR-valve.

Example 6.4
The fault S_{Ev} is present. Provided that the system is optimally exited, the models: $M_{1b},\ M_{1c},\ M_{2b},\ M_{2c},\ M_{3b},\ M_{3c}, M_{4b},\ M_{4c},\ M_{6},\ M_{7}$ are

invalidated.

$$\neg\text{ass } M_{1b} \wedge \neg\text{ass } M_{1c} \wedge \neg\text{ass } M_{2b} \wedge \neg\text{ass } M_{2c} \wedge \neg\text{ass } M_{3b} \wedge$$
$$\neg\text{ass } M_{3c} \wedge \neg\text{ass} M_{4b} \wedge \neg\text{ass } M_{4c} \wedge \neg\text{ass } M_{6} \wedge \neg\text{ass } M_{7}$$

$$\simeq_{\Phi_{W,Te,Tb,Pb,V,Ev}}$$
$$\vdots$$
$$\simeq_{\Phi_{W,Te,Tb,Pb,V,Ev}}$$

Disjunctive normal form:

$$((\{NF_{Pb}, UF_{Pb}\} \wedge \{NF_{Tb}, UF_{Tb}\} \wedge \{NF_{Te}, UF_{Te}\} \wedge \{UF_W\}) \vee$$
$$(\{NF_{Pb}, UF_{Pb}\} \wedge \{NF_{Tb}, UF_{Tb}\} \wedge \{UF_{Te}\} \wedge \{NF_W\}) \vee$$
$$(\{NF_{Pb}, UF_{Pb}\} \wedge \{UF_{Tb}\} \wedge \{NF_{Te}\} \wedge \{NF_W\}) \vee$$
$$(\{UF_{Pb}\} \wedge \{NF_{Tb}\} \wedge \{NF_{Te}\} \wedge \{NF_W\}) \vee$$
$$(\{S_{Ev}, UF_{Ev}\} \wedge \{NF_{Pb}\} \wedge \{NF_{Tb}\} \wedge \{NF_{Te}\} \wedge \{NF_W\}) \vee$$
$$(\{NF_{Pb}\} \wedge \{NF_{Tb}\} \wedge \{NF_{Te}\} \wedge \{UF_V\} \wedge \{NF_W\}) \vee$$
$$(\{SC_{Ev}\} \wedge \{NF_{Pb}\} \wedge \{NF_{Tb}\} \wedge \{NF_{Te}\} \wedge \{SC_V\} \wedge \{NF_W\}) \vee$$
$$(\{SO_{Ev}\} \wedge \{NF_{Pb}\} \wedge \{NF_{Tb}\} \wedge \{NF_{Te}\} \wedge \{SO_V\} \wedge \{NF_W\})$$

$$\Longrightarrow$$

Minimal candidates:

$$(NF_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge NF_V \wedge UF_W) \vee$$
$$(NF_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge UF_{Te} \wedge NF_V \wedge NF_W) \vee$$
$$(NF_{Ev} \wedge NF_{Pb} \wedge UF_{Tb} \wedge NF_{Te} \wedge NF_V \wedge NF_W) \vee$$
$$(NF_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge UF_V \wedge NF_W) \vee$$
$$(NF_{Ev} \wedge UF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge NF_V \wedge NF_W) \vee$$
$$(S_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge NF_V \wedge NF_W) \vee$$
$$(SC_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge SC_V \wedge NF_W) \vee$$
$$(SO_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge SO_V \wedge NF_W)$$

$$\Longrightarrow$$

Probability prioritized candidate, assuming that: $p(NF_{C_i}) \gg p(F^j_{C_i}) > p(UF_{C_i})$.

$$S_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge NF_V \wedge NF_W$$

By utilizing the probability prioritization the remaining candidate describe the actual system condition. Since the probabilities of the different behavioral modes is not known, it is assumed that: $p(NF_{C_i}) \gg p(F^j_{C_i}) > p(UF_{C_i})$. Using these probabilities, the result when using probability prioritization is actually the same as the result of the behavioral mode grading prioritization.

As indicated earlier, isolating faults in the EGR-valve is more complicated than isolating faults in the sensors. The reason is the lack of models based only on the behavior of the EGR-valve. This gives that a fault large enough in a sensor will often cover up a fault in the EGR-valve. This problem is illustrated in the following example.

Example 6.5
The fault $S_{Ev} \wedge RH_{Pb}$ is present. Provided that the system is optimally exited, the models: M_{1b}, M_{1c}, M_{2b}, M_{2c}, M_{3b}, M_{3c}, M_{4b}, M_{4c}, M_5, M_6, M_7 are invalidated.

$$\neg\text{ass } M_{1b} \wedge \neg\text{ass } M_{1c} \wedge \neg\text{ass } M_{2b} \wedge \neg\text{ass } M_{2c} \wedge \neg\text{ass } M_{3b} \wedge$$
$$\neg\text{ass } M_{3c} \wedge \neg\text{ass} M_{4b} \wedge \neg\text{ass } M_{4c} \wedge \neg\text{ass } M_5 \wedge \neg\text{ass } M_6 \wedge \neg\text{ass } M_7$$

$$\simeq_{\Phi_{W,Te,Tb,Pb,V,Ev}}$$
$$\vdots$$
$$\simeq_{\Phi_{W,Te,Tb,Pb,V,Ev}}$$

Disjunctive normal form:
$$(\{RH_{Pb}, UF_{Pb}\} \wedge \{NF_{Tb}, UF_{Tb}\} \wedge \{NF_{Te}, UF_{Te}\} \wedge \{NF_W, UF_W\})$$
$$\Longrightarrow$$

Minimal candidates:
$$(NF_{Ev} \wedge RH_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge NF_V \wedge NF_W)$$

The generalized minimal candidate include only the sensor fault.

If a sensor fault occurs at the the same time as a fault in the EGR-valve or venturi, only the sensor fault will be isolated.

The next example illustrates the process of isolating a fault in the venturi.

Example 6.6
The fault $B_V = SC_V$ is present. Provided that the system is optimally exited, the models: M_{1b}, M_{2b}, M_{3b}, M_{4b}, M_{5b}, M_{6b}, M_{7b}, M_{8b}, M_{12} are invalidated.

$$\neg\text{ass } M_{1b} \wedge \neg\text{ass } M_{2b} \wedge \neg\text{ass } M_{3b} \wedge \neg\text{ass } M_{4b} \wedge \neg\text{ass } M_{5b} \wedge$$
$$\neg\text{ass } M_{6b} \wedge \neg\text{ass } M_{7b} \wedge \neg\text{ass } M_{8b} \wedge \neg\text{ass } M_{12}$$

$$\simeq_{\Phi_{W,Te,Tb,Pb,V,Ev}}$$
$$\vdots$$
$$\simeq_{\Phi_{W,Te,Tb,Pb,V,Ev}}$$

Disjunctive normal form:

$$(\{NF_{Pb}, UF_{Pb}\} \wedge \{NF_{Tb}, UF_{Tb}\} \wedge \{NF_{Te}, UF_{Te}\} \wedge \{UF_W\}) \vee$$
$$(\{NF_{Pb}, UF_{Pb}\} \wedge \{NF_{Tb}, UF_{Tb}\} \wedge \{UF_{Te}\} \wedge \{NF_W\}) \vee$$
$$(\{NF_{Pb}, UF_{Pb}\} \wedge \{UF_{Tb}\} \wedge \{NF_{Te}\} \wedge \{NF_W\}) \vee$$
$$(\{UF_{Pb}\} \wedge \{NF_{Tb}\} \wedge \{NF_{Te}\} \wedge \{NF_W\}) \vee$$
$$(\{S_{Ev}, SO_{Ev}, UF_{Ev}\} \wedge \{NF_{Pb}\} \wedge \{NF_{Tb}\} \wedge \{NF_{Te}\} \wedge \{NF_W\}) \vee$$
$$(\{NF_{Pb}\} \wedge \{NF_{Tb}\} \wedge \{NF_{Te}\} \wedge \{SC_V, UF_V\} \wedge \{NF_W\})$$

$$\Longrightarrow$$

Minimal candidates:

$$(NF_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge NF_V \wedge UF_W) \vee$$
$$(NF_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge SC_V \wedge NF_W) \vee$$
$$(NF_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge UF_{Te} \wedge NF_V \wedge NF_W) \vee$$
$$(NF_{Ev} \wedge NF_{Pb} \wedge UF_{Tb} \wedge NF_{Te} \wedge NF_V \wedge NF_W) \vee$$
$$(NF_{Ev} \wedge UF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge NF_V \wedge NF_W) \vee$$
$$(SO_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge NF_V \wedge NF_W) \vee$$
$$(S_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge NF_V \wedge NF_W)$$

$$\Longrightarrow$$

Probability prioritized candidates, $p(NF) \gg p(F^i) > p(UF)$:

$$(NF_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge SC_V \wedge NF_W) \vee$$
$$(SO_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge NF_V \wedge NF_W) \vee$$
$$(S_{Ev} \wedge NF_{Pb} \wedge NF_{Tb} \wedge NF_{Te} \wedge NF_V \wedge NF_W)$$

One of the probability prioritized candidates describe the actual system condition, the other two indicate fault in the EGR-valve.

The performance of the fault isolation algorithm is very good, faults are isolated as far as it is possible with the provided diagnostic information. Because of the weak modelling of the behavior of the venturi and EGR-valve, it will in some cases be impossible to isolate faults in the venturi from faults in the EGR-valve.

Chapter 7

Conclusions

The performance of the fault isolation algorithm is very good, faults are isolated as far as it is possible with the provided diagnostic information. It is concluded that neither the full candidate representation nor the generalized kernel candidate representation are conclusive enough. The generalized minimal candidate representation focuses on the interesting diagnostic statements to a large extent. If further focusing is needed, it is satisfactory to present the minimal candidates which have a probability close to the most probable minimal candidate.

By utilizing domain logic a more compact representation of diagnostic information is made possible. Furthermore, the fault isolation algorithm requires less computations if the diagnostic information is expressed in domain logic.

References

[1] A.K. Mackworth J. de Kleer and R. Reiter. Charactherizing diagnoses and systems. *Readings in model based diagnosis*, pages 54–65.

[2] M. Nyberg and E. Frisk. Diagnosis and supervision of technical processes. Linköping, Sweden, 2001. Course material, Linköpings Universitet, Sweden.

[3] Tillämpad logik. Linköping, Sweden, 2001. Course material, Linköpings Universitet, Sweden.

Notation

Symbols used in the report.

Behavioral modes

NF	No fault
BO	Burnt out
SG	Short to ground
RH	Sensor value is out of range, high
RL	Sensor value is out of range, low
SC	Stuck closed
SO	Stuck open
S	Stuck
UF	Unknown fault

Operators

$\prec$	Precedes
$\simeq$	Logical equivalence
$\bigvee$	Or
$\bigwedge$	And
$\setminus$	Domain difference
$\neg$	Not
p	Probability

Compound expressions and abbreviations

BM^i	One of the possible behavior modes
$\simeq_{\Phi_{C_1..C_n}}$	$\simeq_{(\Phi_{C_1} \wedge \ldots \wedge \Phi_{C_n})}$
$\perp$	False
EGR	Exhaust gas recirculation
η_{vol}	Volumetric efficiency
W_{cyl}	Total mass flow
W_{air}	Air mass flow
W_{EGR}	EGR mass flow
$\%_{EGR}$	EGR mass flow fraction
$\%^d_{EGR}$	Demanded EGR mass flow fraction
f'	$f(h', W_s, N, Pb_s, Tb_s)$
g'	$g(h', N, Pb_s, Tb_s)$
h'	$h(Tb_s, Te_s, Pb_s, N)$

Appendix A

Rules of logic

A.1 Rules of domain logic

$$\neg(B_C \in \alpha_C) \simeq_{\Phi_C} B_C \in \Upsilon_C \backslash \alpha_C \tag{A.1}$$

$$B_C \in \alpha_C \wedge B_C \in \beta_C \simeq_{\Phi_C} B_C \in \alpha_C \bigcap \beta_C \tag{A.2}$$

$$B_C \in \alpha_C \vee B_C \in \beta_C \simeq_{\Phi_C} B_C \in \alpha_C \bigcup \beta_C \tag{A.3}$$

$$B_C \in \emptyset \simeq \bot \tag{A.4}$$

$$B_C \in \Upsilon_C \simeq \neg\bot \tag{A.5}$$

$$B_C \in \bigcup_{\forall BM_C^i \in \alpha_C} \{BM_C^i\} \simeq (\bigvee_{\forall BM_C^i \in \alpha_C} B_C = BM_C^i) \tag{A.6}$$

A.2 Rules of ordinary logic

p, q and r are general formulas.

$$
\begin{aligned}
p \wedge p &\simeq p && \text{(A.7)}\\
p \vee p &\simeq p && \text{(A.8)}\\
p \wedge q &\simeq q \wedge p && \text{(A.9)}\\
p \vee q &\simeq q \vee p && \text{(A.10)}\\
p \wedge (q \wedge r) &\simeq (p \wedge q) \wedge r && \text{(A.11)}\\
p \vee (q \vee r) &\simeq (p \vee q) \vee r && \text{(A.12)}\\
p \wedge (q \vee r) &\simeq (p \wedge q) \vee (p \wedge r) && \text{(A.13)}\\
p \vee (q \wedge r) &\simeq (p \vee q) \wedge (p \vee r) && \text{(A.14)}\\
\neg(p \vee q) &\simeq (\neg p \wedge \neg q) && \text{(A.15)}\\
\neg(p \wedge q) &\simeq (\neg p \vee \neg q) && \text{(A.16)}\\
p \wedge (p \vee q) &\simeq p && \text{(A.17)}\\
p \vee (p \wedge q) &\simeq p && \text{(A.18)}
\end{aligned}
$$

Appendix B

Transforming a formula in domain logic to dnf

Principle for transposing a general formula in domain logic to disjunctive normal form in domain logic.

1. Evaluate all occurrences of $\neg$ according to rules A.1, A.15 and A.16.
2. According to rule A.13 move all occurrences of $\wedge$ inside the occurrences of $\vee$.
3. Use rules A.9, A.11 and A.2 to eliminate occurrences of multiple behavior mode assignments for the same component in a conjunction.
4. Remove conjunctions assigning an empty set of behavior modes according to rule A.4.
5. Use rules A.8, A.10 and A.12 to remove multiple occurrences of the same conjunction.
6. Remove redundant information. How this is done is indicated in Appendix C.

Appendix C

Eliminating redundant information

Principle for eliminating redundant information contained in a formula on full dnf in domain logic. The details of how this is done for a formula on dnf is left out. Each formula is made up by a number of conjunctions $\mathbb{C}$, over $C_1, ..., C_n$, assigning the behavioral mode domains α_i.

For all conjunctions $\mathbb{C}$

For all conjunctions $\mathbb{C}' \neq \mathbb{C}$

If $\alpha'_i \cap \alpha_i \neq \emptyset,\ \forall i$

- If $\alpha'_i \subseteq \alpha_i,\ \forall i$

  Remove $\mathbb{C}'$

- Elseif $\alpha'_i \subseteq \alpha_i,\ \forall i \in \{1..n\} \backslash \{j\}$

  Set $\alpha'_j \ = \ \alpha'_j \backslash \alpha_j$

Appendix D

Deduction of the generalized minimal candidates

Principle for deducing the generalized minimal candidates from the full candidates expressed in domain logic, over $C_1, ..., C_n$, assigning the behavioral mode domains α_{C_i}.

1. For all full candidates

   For all components C_i

   - If $NF_{C_i} \in \alpha_{C_i}$
     Set $B_{C_i} \in NF_{C_i}$
   - Elseif $\alpha_{C_i} \backslash \{UF_{C_i}\} \neq \emptyset$
     Set $B_{C_i} \in \alpha_{C_i} \backslash \{UF_{C_i}\}$
   - Else
     Set $B_{C_i} \in \{UF_{C_i}\}$

2. Apply rule A.6 to represent the formula in ordinary logic

3. Use the principle presented in B to deduce the dnf representation in ordinary logic

4. For all full candidates $\mathbf{C}$

   For all full candidates, $\mathbf{C}' \neq \mathbf{C}$

   - If $\mathbf{C} \prec \mathbf{C}'$
     Remove $\mathbf{C}'$
   - Elseif $\mathbf{C} \succ \mathbf{C}'$
     Remove $\mathbf{C}$