On the Analysis of Spectrum-based Fault Localization

A thesis submitted for the degree of
Doctor of Philosophy

by

Xiaoyuan Xie

Faculty of Information and Communication Technologies
Swinburne University of Technology
Australia

2012
Abstract

Spectrum-based fault localization (SBFL) has been widely studied due to its simplicity and effectiveness. However, it still has some challenging problems. The application of SBFL in the absence of test oracle and the selection of the most effective risk evaluation formulas are amongst the most critical problems. In this thesis, we are going to address these two problems.

Currently, all existing SBFL techniques have assumed the existence of a test oracle. Otherwise, the program spectrum will not be associated with the testing result of failed or passed, and as a consequence, there will be insufficient information to perform the risk evaluation. However, in many real-world applications, it is very common that test oracles do not exist, and hence SBFL cannot be applied in such situations. Therefore, in this thesis, we propose a novel concept of metamorphic slice resulted from the integration of metamorphic testing and program slicing, to alleviate the oracle problem for SBFL. In our approach, instead of using the program slice and the testing result of failed or passed for an individual test case, metamorphic slice and the testing result of violation or non-violation of a metamorphic relation are used. Since we need not to know the execution result for an individual test case, the existence of test oracle is no longer a prerequisite to SBFL. Experimental results show that our proposed solution delivers a performance comparable to the performance of existing SBFL techniques for the situations where test oracles exist. As a consequence, our study has significantly extended the scope of the applicability of SBFL.

For the second problem of selecting the most effective risk evaluation formulas, though it has been one of the most important tasks in SBFL, there does not exist a completely satisfactory solution. It is well-known that risk evaluation is very critical in SBFL and hence many studies have been conducted to compare the performance among various risk evaluation formulas. Most of the previous studies have adopted an empirical approach, which however, can hardly be considered as sufficiently comprehensive because of the huge possible combinations of various factors in SBFL. Though there are some studies aiming at overcoming the limitations of the empirical studies through a theoretical approach, these studies were based on the most strict type of equivalence that does not properly reflect the more realistic scenario, and did not adopt the most commonly used performance metric. Therefore, in this thesis, we provide a theoretical investigation on the effectiveness of risk evaluation formulas. We define two types of relations between different formulas, namely, equivalent and
better. To identify the relations between different formulas, we develop an innovative framework for the theoretical investigation. Our framework is based on the concept that the determinant for the effectiveness of a formula is the number of statements with risk values higher than that of the faulty statement. Our framework groups all program statements into three disjoint sets with risk values higher than, equal to and lower than that of the faulty statement, respectively. For different formulas, the sizes of their sets are compared using the notion of subset. We use this framework to identify the maximal formulas which should be the only candidate formulas for use. Compared with previous studies, our conclusions are derived from a completely theoretical analysis, and hence are more robust. Besides, we adopt the most commonly used performance metric, and use a more general and intuitively appealing type of equivalence relation.
Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Tsong Yueh Chen for being the mentor in both of my PhD study and my life. During the past four years, his immense experience and knowledge have given me invaluable guidance for choosing my research topics. His patient supervision has made me more and more mature in my way of academic. And his wisdom has taught me how to face every difficulty and appreciate every single happiness in my life.

I would like to thank my parents who have provided my most reliable foundation of support, in every way, from the very beginning. They are the persons who know me the best in the world. My father, Yongda Xie, always gives me most constructive advices whenever I encounter any difficulties. His encouragement has given me so much confidence and courage, and has taught me to have faith, never give up and always persist. Thanks to my mother, Yihua Liu, who has given me most selfless love all the time. Her successful career also gives me a good example for being an independent woman and pursuing my dream.

Many thanks are given to my dear friend Shirui Liu, who has supported me so much during my PhD study. Also thanks to the members of our research group: Dr. Huai Liu, Dr. Fei-Ching Kuo, Dr. Robert G. Merkel, Dr. Arlinta Barus and Xuan Liu. Their support and kind help have given me such a pleasant working environment during my PhD study.

Finally, I would like to thank my previous supervisor Baowen Xu, who firstly led me into the fantastic world of software testing, and introduced Prof. Tsong Yueh Chen to me. I won’t forget how I read my first academic paper and drafted my first publication under his supervision. All these have given me a good foundation for my PhD study.

Xiaoyuan Xie

Melbourne, Australia

May, 2012
Declaration

I, Xiaoyuan Xie, declare that this thesis entitled:

“On the Analysis of Spectrum-based Fault Localization”

is my own work and has not been submitted previously, in whole or in part, in respect of any other academic award.

Xiaoyuan Xie

Faculty of Information and Communication Technologies
Swinburne University of Technology
Australia

June, 2012
# Contents

Abstract  

Acknowledgements  

Contents  

List of Figures  

List of Tables  

1 Introduction  
   1.1 Automatic fault localization  
   1.2 Problem statement  
      1.2.1 Application of SBFL in the absence of test oracle  
      1.2.2 Identification of the most effective risk evaluation formulas  
   1.3 Contributions  
   1.4 Thesis outline  

2 Background  
   2.1 Spectrum-based fault localization  
   2.2 Metamorphic testing  
   2.3 Program slicing  

3 SBFL in the absence of test oracle
## CONTENTS

3.1 Preliminary ................................................................. 14
3.2 Metamorphic slice: a property-based program slice ................. 15
3.3 SBFL with e-mslice ...................................................... 16
3.4 Experimental set-up ..................................................... 17
   3.4.1 Testing objects and their metamorphic relations ............ 17
   3.4.2 Source test suite generation ................................. 27
   3.4.3 Mutant generation ............................................... 27
   3.4.4 Experimental methodology .................................. 29
3.5 Results and analysis ................................................... 30
   3.5.1 EXAM score distribution ..................................... 30
   3.5.2 Statistical comparison ........................................ 37
   3.5.3 Average EXAM score comparison ............................ 44
   3.5.4 Real-life faults in Siemens Suite ........................... 46
   3.5.5 Summary .......................................................... 46
3.6 Threats to validity ...................................................... 47
   3.6.1 External validity ............................................... 47
   3.6.2 Internal validity ............................................... 47
   3.6.3 Construct validity ............................................. 48
3.7 Related works .......................................................... 48

4 A Theoretical analysis of the risk evaluation formulas in SBFL .... 49
4.1 Motivation ................................................................. 49
4.2 Our framework .......................................................... 50
4.3 Effectiveness comparison of risk evaluation formulas .............. 53
   4.3.1 Assumptions ..................................................... 53
   4.3.2 Investigated formulas ......................................... 54
   4.3.3 $S^R_B$, $S^R_F$ and $S^R_A$ for all formulas .................. 57
CONTENTS

4.3.4 Maximal risk evaluation formulas ...................................... 101
4.4 Performance refinement on risk evaluation formulas ................. 124
4.5 Discussion ............................................................................. 129
4.6 Related works ....................................................................... 130

5 Conclusion .................................................................................. 134
  5.1 Summary .................................................................................. 134
  5.2 Future works ........................................................................... 136

References ..................................................................................... 139

Publications of the Author ................................................................. 148
List of Figures

2.1 Essential information for conventional SBFL .......................... 8
2.2 An example .......................................................... 8

3.1 Essential information for SBFL with $e_{mslice}$ .......................... 16
3.2 Distribution of $EXAM$ scores (first part) ............................ 31
3.2 Distribution of $EXAM$ scores (second part) ......................... 32
3.2 Distribution of $EXAM$ scores (third part) .......................... 33
3.2 Distribution of $EXAM$ scores (fourth part) ......................... 34
3.2 Distribution of $EXAM$ scores (fifth part) .......................... 35
3.3 Average of $EXAM$ scores .......................................... 45

4.1 Sample program $PG_1$ ............................................. 123
4.2 Sample program $PG_2$ ............................................. 124
List of Tables

3.1 Executable line of codes ................................................. 18
3.2 Number of source test cases ........................................... 28
3.3 Number of mutants ....................................................... 28
3.4 Investigated formulas .................................................. 29
3.5 MS v.s. S-ST ............................................................... 38
3.6 MS v.s. S-FT ............................................................... 40
3.7 MS v.s. S-AT ............................................................... 42
3.8 Summarized results ....................................................... 44
4.1 Investigated formulas ..................................................... 55
4.2 $A_i$ for $P G_1$ with $T S_1$ .............................................. 122
4.3 $A_i$ for $P G_2$ with $T S_2$ .............................................. 122
4.4 Sets for different combinations of formula and test suite ........ 122
1

Introduction

1.1 Automatic fault localization

It is commonly recognized that testing and debugging are important but expensive activities in software engineering. Attempts to reduce the number of faults in software are estimated to consume 50% to 80% of the total development and maintenance effort [Collofello and Woodfield, 1989]. Fault localization is one of the most essential activities, which pinpoints the possible positions of the faults based on various information, and hence gives debuggers hints for fault-fixing. However, this activity always involves a great amount of manual job. For example, a typical debugging process usually begins when the failures of a program are observed. Then, a debugger will focus on a particular failed execution and manually set a series of breakpoints in the program. By inspecting and altering the internal states at these breakpoints, and iteratively re-executing the program with this test case, the debugger will locate the fault for this failure.

Obviously, such a great amount of manual involvement makes the fault localization very resource consuming and not effective. Therefore, automation of this task becomes very important, which can significantly increase its effectiveness and decrease its cost.

Currently, many techniques have been proposed towards the automatic fault localization. Some of them use various information to isolate a set of program entities that are likely to be faulty, based on different heuristics [Agrawal et al., 1995; DeMillo et al., 1996; Renieris and Reiss, 2003; Gupta et al., 2005; Zhang et al., 2005; Wong and Qi, 2006]. For example, Zhang et al. [2005] have utilized different types of dynamic slices associated with failed test cases, as the set of suspicious statements of being faulty. However, these techniques always have difficulties in compromising between the effectiveness and precision. Generally speaking, a large suspicious set of program entities usually has better precision since it has more chance to contain the faults; however, it has to sacrifice the effectiveness since more program entities need to be examined. Moreover, these techniques usually involve complicated program analysis, and hence are not efficient enough to be adopted in practice.
1.2 Problem statement

Therefore, people have proposed another promising automatic fault localization technique, spectrum-based fault localization (referred to as SBFL in this thesis). Instead of isolating the suspicious program entities, SBFL ranks program entities according to their risks of being faulty. Generally speaking, SBFL first collects the information from software testing, including various program spectra and the associated testing result, in terms of failed or passed, of each individual test case. The program spectrum can be any granularity of program entities (e.g. statements, branches, blocks, etc.), and any type of run-time information (e.g. the binary coverage status, the execution frequency, etc.) \cite{Reps1997, Harrold2000}. With these information, SBFL then uses different formulas to evaluate the risk of containing a fault for each program entity, and gives a risk ranking list. SBFL intends to highlight program entities which strongly correlate with program failures, and these entities are regarded as the likely faulty locations \cite{Abreu2007}.

Compared with the other debugging techniques, SBFL is much easier to be implemented and adopted in practice. Actually, it has received a lot of attention due to its simplicity and effectiveness. Some recent studies in SBFL were focused on proposing new approaches that are distinguished with each other in the selection of program spectrum, the choice of formula used for evaluating risk values for each program entity, etc., in order to improve the accuracy of the diagnosis. Some typical risk evaluation formulas include Pinpoint \cite{Chen2002a}, Tarantula \cite{Jones2002}, Ochiai \cite{Abreu2006}, etc. On the other hand, some studies have aimed to compare the performance of different SBFL techniques, or to investigate how can different factors (e.g. number of failed test cases, size of test suite, selection of coverage criterion, tie-breaking scheme for statements with the same risk values, etc.) affect the performance of a particular SBFL technique, and how to adjust them to obtain a better performance \cite{Jones2005, Abreu2006, Abreu2009a, Santelices2009, Jiang2010}.

1.2 Problem statement

Despite of the extensive studies, there are still some problems in SBFL, such as, dealing with the multiple faults, efficiency of locating faults for large-scaled programs, applying SBFL in the absence of test oracle, and selecting the most effective risk evaluation formulas, etc.

First, it is well-known that dealing with multiple faults is usually much more complicated than dealing with single fault, because some faults may mask or obfuscate others, and an observed failure may be resulted from the interaction among a series of faults. Recently, this problem has received more and more attention, and many techniques have been proposed to address this problem, including the clustering-based method, model-based method, etc. \cite{Zheng2006, Jones2007, Abreu2009b, Abreu2009c}.

For the problem of applying SBFL in large-scaled programs, one challenge is that for these
1.2. PROBLEM STATEMENT

programs, even there is only a small percentage of program entities ranked before the faulty entity in
the final ranking list, the absolute number of these entities is still large. Thus, the effectiveness of an
SBFL technique becomes critical, and any small improvement in the performance will significantly
reduce the cost of debugging. Besides, for the large-scaled programs, collecting and analyzing the
program spectrum are usually very resource-consuming due to the huge amount of data. This problem
has been studied in [Liblit et al., 2003, 2005; Ahn et al., 2009], where a program sampling and
instrumentation technique was used to isolate the faults.

With regard to the problem of applying SBFL in the absence of test oracle, it has never been
addressed to the best of our knowledge. And for the selection of risk formulas, though many studies
were conducted on it, there is still no completely satisfactory solution to identify which formulas are
the most effective ones. Therefore, in this thesis, we will focus on the last two problems.

1.2.1 Application of SBFL in the absence of test oracle

In the conventional SBFL techniques, both the program spectrum and testing results, in terms of
failed or passed, are required to evaluate the risk values for the program entities. In other words, the
conventional SBFL techniques have assumed that test oracle is available for the program, that is, the
correctness of the computed outputs can always be verified.

However, in many real-world applications, such as, complex computational programs, machine
learning algorithms, etc. [Baker and Thornton, 2004; Chen et al., 2009; Ho et al., 2008, 2010; Xie
et al., 2009, 2011a], it is impossible or too expensive to verify the correctness of the computed outputs.
This is known as the “oracle problem” [Weyuker, 1982]. For such applications, the program spectrum
cannot be associated with any testing result. Therefore, the information is incomplete for the risk
evaluation and hence the conventional SBFL becomes inapplicable for such applications.

As a consequence, this assumption has severely restricted the scope of the application for the
conventional SBFL techniques. In this thesis, we will present our solution to this problem in
Chapter 3.

1.2.2 Identification of the most effective risk evaluation formulas

It is well-known that risk evaluation is one of the most important tasks in SBFL. With more and more
formulas being proposed, to identify the most effective formulas becomes one popular and important
research area.

Currently, most of the related studies are empirical investigations [Jones and Harrold, 2005;
Abreu et al., 2006, 2007; Wong et al., 2007; Abreu et al., 2009a], in which various approaches
have been applied to control the threats to validity (e.g. using the established experimental set-up
1.3. CONTRIBUTIONS

and benchmark, Siemens Suite [SIR, 2005]), in order to provide a fair evaluation and comparison. However, the performance of a risk evaluation formula is strongly dependent on the experimental set-up. No matter how well the researchers standardize their experimental set-up or vary the set-up choices, their investigation could never be considered as sufficiently comprehensive because of the huge amount of possible combinations of various factors. Thus, despite of all the above attempts, the limitations of the experimental study still cannot be ignored.

Recently, some researchers have conducted a theoretical analysis on the effectiveness of the risk evaluation formulas. Lee et al. [2009a] have first proved that formulas Tarantula and \( q_e \) are equivalent. In their follow-up study, a more comprehensive investigation was conducted, where more groups of equivalent risk evaluation formulas were proved [Naish et al., 2011]. However, the type of equivalence used by both Lee et al. [2009a] and Naish et al. [2011] is the most strict type of equivalence, which requires identical ranking list for equivalent formulas. Naish et al. [2011] also proposed two optimal risk evaluation formulas, with respect to their proposed model program and performance metric that calculates the average performance over all possible multisets of execution paths. However, their metric was not commonly used by the SBFL community.

In summary, there is no completely satisfactory method to identify the most effective formulas with respect to the most commonly used metric, namely, EXAM score [Wong et al., 2010]. Thus, in this thesis, we will provide a solution to this important problem in Chapter 4.

1.3 Contributions

Our contributions in this thesis are as follows.

1. In order to apply the SBFL in programs without test oracle, we propose a novel type of program slice, namely, metamorphic slice, as a counterpart of the traditional slice used in SBFL when test oracle is available. More precisely, this concept of metamorphic slice is a combination of traditional slice and the metamorphic testing that was proposed to alleviate oracle problem [Chen et al., 1998, 2003a]. It is property-based and related to certain expected characterization of the program under testing. Thus, even in the absence of test oracle, the outcomes of dissatisfaction or satisfaction of this property can still be determined, which are referred to as violation or non-violation hereafter, and are regarded as the counterparts of failure or pass, respectively, for the situation that a test oracle exists.

In fact, our proposal of using metamorphic slice in SBFL is not only intuitively appealing and conceptually feasible, but also practically effective. To investigate its performance, an experimental study involving 9 programs and 3 risk evaluation formulas is conducted. The results show that our proposed solution delivers a performance comparable to the performance of existing
SBFL techniques for the situations where test oracles exist. As a consequence, with metamorphic slice, test oracle is no longer mandatory for SBFL, and hence the scope of the applicability of SBFL is significantly extended.

2. In order to identify the most effective risk evaluation formulas, we provide an innovative theoretical solution. It is well-known that rather than the absolute risk value, the ranking of the faulty statement is the determinant of the performance for a risk evaluation formula. To identify whether a formula is equivalent to or better than another formula, we develop an innovative theoretical framework, which compares the number of statements with risk values higher than that of the faulty statement among different formulas using the notion of subset.

Using this framework, we investigate 30 risk evaluation formulas, and are able to find five out of these 30 formulas as maximal formulas, under the single-fault scenario. And Naish et al.’s two optimal formulas are within these five maximal formulas.

1.4 Thesis outline

The rest of this thesis is organized as follows.

Chapter 2 first introduces the concept of SBFL techniques. Then, it presents the details of metamorphic testing. Finally, it briefly introduces the basic concept of program slicing and its applications.

Chapter 3 is focused on the problem of applying SBFL in the absence of test oracle. It first introduces a novel concept of metamorphic slice. In particular, two types of metamorphic slices are defined, namely, execution metamorphic slice and dynamic metamorphic slice. Then, this chapter describes how the execution metamorphic slice can be applied to alleviate the oracle problem in SBFL, and presents the experimental results to demonstrate how effective our proposal is. Finally, some related works are discussed.

Chapter 4 is focused on the problem of identifying the most effective risk evaluation formulas. It starts by giving the motivation of our study. Then, an innovative theoretical framework is proposed to conduct the performance comparison of different risk evaluation formulas, based on the intuition that the most fundamental determinant for the performance of a formula is the number of statements with risk values higher than that of the faulty statement. With this framework, 30 existing risk evaluation formulas are then investigated, from which five maximal formulas are identified. Furthermore, this chapter also investigates the intuitions behind the risk evaluation formulas, and proposes a refinement solution to improve the performance for some formulas. Finally, some related studies and their limitations are discussed.

Chapter 5 summarizes the main contributions of this thesis, discusses about their impact and
significance, gives the conclusions, and finally identifies several research topics for our future works.
2

Background

This chapter provides background and context for the work presented in this thesis. We begin in Section 2.1 by introducing the basic concept of spectrum-based fault localization. We then describe the metamorphic testing technique in details in Section 2.2, and discuss some previous works in this field. Finally, in Section 2.3, we briefly introduce the traditional program slicing techniques, which will be integrated with metamorphic testing in our proposal to alleviate the oracle problem for spectrum-based fault localization.

2.1 Spectrum-based fault localization

Spectrum-based fault localization (referred to as SBFL) is a dynamic approach, which basically utilizes two types of information collected during software testing, namely testing results and program spectrum. The testing result associated with each test case records whether a test case is passed or failed. While a program spectrum is a collection of data that provides a specific view on the dynamic behaviour of software [Reps et al., 1997; Harrold et al., 1998]. Generally speaking, it records the run-time profiles about various program entities for a specific test suite. The program entities could be statements, branches, paths, basic blocks, etc.; while the run-time information could be the binary coverage status, the execution frequency, the program state before and after executing the program entity, etc. In practice, there are many kinds of combinations [Harrold et al., 1998, 2000]. The most widely adopted combination involves statement and its binary coverage status in a test execution [Agrawal et al., 1995; Wong et al., 2005; Wong and Qi, 2006]. In this thesis, we will follow the common practice to use this combination as a representative of the program spectrum.

Let us consider a program $PG=<s_1, s_2, ..., s_n>$ with $n$ statements and executed by a test suite of $m$ test cases $TS={t_1, t_2, ..., t_m}$. Figure 2.1 shows the essential information required by SBFL. $RE$ records all the testing results associated with the test cases, in which $p$ indicates passed and $f$ indicates failed. And matrix $MS$ represents the program spectrum, where the element in the $i^{th}$ row
2.1. SPECTRUM-BASED FAULT LOCALIZATION

and \(j^{th}\) column represents the coverage information of statement \(s_i\), by test case \(t_j\), with 1 indicating \(s_i\) is executed, and 0 otherwise. In other words, the \(j^{th}\) column represents the execution slice of \(t_j\).

\[
\begin{align*}
TS : & (t_1 & t_2 & \ldots & t_m) \\
MS : & \begin{cases}
    1/0 & 1/0 & \ldots & 1/0 \\
    1/0 & 1/0 & \ldots & 1/0 \\
    \vdots & & & \vdots \\
    1/0 & 1/0 & \ldots & 1/0
\end{cases} \\
MS : & \begin{cases}
    \vdots & & & \vdots \\
    \vdots & & & \vdots \\
    \vdots & & & \vdots \\
    \vdots & & & \vdots \\
\end{cases} \\
RE : & (p/f & p/f & \ldots & p/f)
\end{align*}
\]

Figure 2.1: Essential information for conventional SBFL

![Figure 2.1](image)

For each statement \(s_i\), these data can be represented as a vector of four elements, denoted as \(A_i = \langle a_{ei}^i, a_{ep}^i, a_{nf}^i, a_{np}^i \rangle\), where \(a_{ei}^i\) and \(a_{ep}^i\) represent the number of test cases in \(TS\) that execute statement \(s_i\) and return the testing result of failure or pass, respectively; \(a_{nf}^i\) and \(a_{np}^i\) denote the number of test cases that do not execute \(s_i\), and return the testing result of failure or pass, respectively. Obviously, the sum of these four parameters for each statement should always be equal to the size of the test suite. An example is shown in Figure 2.2.

\[
\begin{align*}
TS : & (t_1 & t_2 & t_3 & t_4 & t_5 & t_6) \\
MS : & \begin{cases}
    1 & 1 & 1 & 1 & 1 & 1 \\
    0 & 0 & 0 & 1 & 0 & 0 \\
    0 & 1 & 1 & 1 & 0 & 1 \\
    1 & 1 & 0 & 1 & 1 & 1
\end{cases} \\
RE : & (p & p & p & f & f)
\end{align*}
\]

Figure 2.2: An example

![Figure 2.2](image)

In Figure 2.2, program \(PG\) has 4 statements \(\{s_1, s_2, s_3, s_4\}\), and test suite \(TS\) has 6 test cases \(\{t_1, t_2, t_3, t_4, t_5, t_6\}\). As indicated in \(RE\), \(t_5\) and \(t_6\) give rise to failed runs and the remaining four test cases give rise to passed runs. Matrix \(MS\) records the binary coverage information for each statement with respect to every test case. Matrix \(MA\) is such defined that its \(i^{th}\) row represents the corresponding \(A_i\) for \(s_i\). For instance, in this figure, \(a_{nf}^1\)=0 for \(s_1\) means that no test case in the current test suite gives a testing result of pass without executing \(s_1\); \(a_{ef}^4\)=2 for \(s_4\) represents that \(s_4\) is executed by two
test cases which can detect failure.

A risk evaluation formula $R$ is then applied on each statement $s_i$ to assign a real value that indicates its risk of being faulty. All formulas follow the same intuition that statements associated with more failed and less passed testing results should have higher risks. For example, formula Tarantula is defined as follows [Jones et al., 2002].

$$R_T(s_i) = \frac{a_{ef}^i}{a_{ef}^i + a_{nf}^i} / \left( \frac{a_{ef}^i}{a_{ef}^i + a_{nf}^i} + \frac{a_{ep}^i}{a_{ep}^i + a_{np}^i} \right)$$

Apart from Tarantula, there are many other risk evaluation formulas including Jaccard [Chen et al., 2002a], AMPLE [Zeller, 2002], Ochiai [Abreu et al., 2006], Wong [Wong et al., 2010], etc. [Pan and Spafford, 1992; Reps et al., 1997; Zeller, 2002; Liblit, 2004; Liblit et al., 2005; Liu et al., 2006; Wong et al., 2008, 2010].

A statement with higher risk value is interpreted to have a higher possibility to be faulty, which therefore should be examined with higher priority. Hence, after being assigned with the risk values, all statements are sorted descendingly according to their risk values. An effective formula should be able to make the faulty statements as top in the list as possible.

For the performance measurement of the risk evaluation formulas, majority of the SBFL community used the same metric or its equivalent, which is the percentage of the code that needs (or needs not) to be examined before the faulty statement is identified. Such a metric is used with the assumption of “perfect bug detection” that the fault can always be identified once it is examined [Wong et al., 2010]. In [Wong et al., 2010], the percentage of code that needs to be examined before the faults are identified is referred to as the EXAM score, which will be adopted in this thesis. Obviously, a lower EXAM score indicates a better performance.

### 2.2 Metamorphic testing

As explained in Section 2.1, SBFL requires both the program spectrum and testing results for the risk evaluation. Thus, the conventional SBFL cannot be applied in the programs without test oracle.

However, many real-world applications have the “oracle problem”, that is, it is impossible or too expensive to verify the correctness of the computed outputs [Weyuker, 1982]. For example, in programs computing multiple precision arithmetic, the operands involved are very large numbers, and hence, the computed results are very expensive to check. When testing a compiler, it is not easy to verify whether the generated object code is equivalent to the source code or not. Other examples include testing programs involving machine learning algorithms, simulations, combinatorial calculations, graph display in the monitor, etc. [Xie et al., 2009, 2011a].
2.2. METAMORPHIC TESTING

Actually, the oracle problem has been one of the biggest difficulties in software testing in the past decades, and several attempts have been conducted to alleviate it. One attempt is to use a “pseudo-oracle” [Davis and Weyuker, 1981], in which multiple implementations of an algorithm process the same test case input and the outputs are compared; if the outputs are not the same, then one or both of the implementations contains a fault. But this is not always feasible, since multiple implementations may not exist, or they may have been created by the same developers, or by groups of developers who are prone to making the same types of mistakes [Knight and Leveson, 1986]. However, even without multiple implementations, these applications often exhibit properties such that given a test case input and its output, if the input is modified in a certain way, it should be possible to predict some characteristics of the new output. This approach is known as metamorphic testing [Chen et al., 1998, 2003a].

Metamorphic testing (MT) uses some specific properties of the problem domain, namely metamorphic relations (MRs), to verify the relationship between multiple but related test cases and their outputs, rather than verifying the correctness of the output for each individual test case. Generally speaking, when conducting MT, we first need to identify the MRs of the program under testing, and choose a test case selection strategy to generate the source test cases, from which the corresponding follow-up test cases are constructed based on the MRs. Then, we execute both the source and the follow-up test cases on the program, and check whether their outputs satisfy the corresponding MRs. Similar to test cases, a metamorphic group violating its corresponding MR implies an incorrect program, but a satisfaction of the corresponding MR does not imply the correctness of the program [Chen et al., 1998, 2003a].

Let us use an example to illustrate MT informally. Readers who are interested in a more formal and comprehensive description of MT, may consult [Chen et al., 1998, 2003a]. Consider a program that searches for the shortest path between any two nodes in an undirected graph and reports its length. Given a weighted graph $G$, a start node $x$, and a destination node $y$ in $G$, the target program is to output the shortest path and its length. Let us denote the length of the shortest path by $d(x, y, G)$. Suppose that the computed value of $d(x, y, G)$ is 12345. It is very expensive to check whether 12345 is correct due to the combinatorially large number of possible paths between $x$ and $y$. Therefore, such a program is said to have the oracle problem. When applying MT to this program, we first need to define an MR based on some well-known properties in graph theory. One possible MR (referred as MR1) is that the length of the shortest path will remain unchanged if we swap the start node and destination node, that is, $d(x, y, G)=d(y, x, G)$. Another possible MR (referred as MR2) is that suppose $w$ is any node in the shortest path with $x$ as the start node and $y$ as the destination node, then the sum of the length of the shortest path from $x$ to $w$ and the length of the shortest path from $w$ to $y$ shall be equal to the length of the shortest path from $x$ to $y$, that is, $d(x, y, G)=d(x, w, G)+d(w, y, G)$.

The core idea is that although it is difficult to verify the correctness of the individual output, namely $d(x, y, G)$, $d(y, x, G)$, $d(x, w, G)$ and $d(w, y, G)$, it is easy to verify
whether the MR1 and MR2 are satisfied or not, that is, whether \( d(x, y, G) = d(y, x, G) \) and \( d(x, y, G) = d(x, w, G) + d(w, y, G) \). In other words, for MR1, we can run the program using \( y \) as the start node and \( x \) as the destination node, if \( d(y, x, G) \) is not equal to 12345, test cases \((x, y, G)\) and \((y, x, G)\) are said to violate MR1. Then, we can conclude that the program is incorrect. As a reminder, if \( d(y, x, G) \) is also 12345, we can neither conclude the program is correct, nor incorrect. This is due to the limitation of software testing. And the similar conclusion can be obtained by using MR2. In this example, \((x, y, G)\) is referred as the source test case, \((y, x, G)\) is the follow-up test case of MR1, and \((x, w, G)\) and \((w, y, G)\) are the follow-up test cases of MR2. As shown, follow-up test cases could be multiple and dependent on both the source test case and the relevant MR. As a reminder, the source test case involved in an MR need not be a single test case and it can be selected according to any test case selection strategies.

For convenience of reference, we will refer a source test case (or a group of source test cases if appropriate; as explained in the previous paragraph, source test cases may be multiple for a specific MR) and its related follow-up test cases as a metamorphic test group.

Obviously, metamorphic testing is simple in concept, easily implemented in practice, and independent of any particular programming language [Gotlieb and Botella, 2003; Murphy et al., 2009a,b]. Up to now, MT has been successfully applied in many application domains, such as the computational programs [Chen et al., 2002b], end-user programs [Chen et al., 2005], heuristic algorithms [Barus et al., 2008], bioinformatics applications [Chen et al., 2009], service-oriented software [Chan et al., 2005, 2007a], graphical software [Chan et al., 2007b], machine learning algorithms [Murphy et al., 2007; Xie et al., 2009, 2011a], etc. Especially in its application on the machine learning algorithms, the function of MT has been extended to validation, beyond its original purpose of verification [Xie et al., 2009, 2011a].

Apart from the application of MT, another important research direction is integrating MT with other testing, debugging and proving techniques. In these integrated methods, MT complements the original techniques, extending their scopes of applicability to programs having oracle problem. Such innovative techniques include the integration of MT and fault-based testing method [Chen et al., 2003b], the integration of MT and symbolic execution [Chen et al., 2011].

Furthermore, there are also some studies investigating the generation and selection of MRs. Hu et al. [2006] and Zhang et al. [2009] have provided empirical evidence that after a brief general training, testers can properly define MRs and effectively apply MT on the target programs. This observation is consistent with the original expectation for MT. After all, in any case, a tester must have domain knowledge on the program under test. Therefore, at least some basic properties of the program can be determined, which are usually sufficient to construct helpful MRs. Actually, almost all of the MRs in previous studies are derived from those basic properties, rather than designed sophisticatedly [Chen et al., 2004; Chan et al., 2007a; Chen et al., 2005; Murphy et al., 2008; Chen et al., 2009;
Xie et al., 2009, 2011a]. Moreover, there are methods that can facilitate the identification of MRs. For example, when testing a group of programs which serve similar purpose, an MR repository can be constructed by harnessing the domain knowledge that indicates some general anticipation in this domain. And MRs from such a repository could be generally reused in this area [Xie et al., 2011a].

2.3 Program slicing

Program slicing is a program analysis technique, which focuses on a portion of the program instead of the whole program without affecting the program’s behaviours of interest. The resulting set of program statements, called a slice, captures a projection of the semantics of the original program [Weiser, 1981; Binkley et al., 2006b]. Normally, the behaviours of interest can be related to a chosen set of variables, or a particular test execution.

The first concept of program slice, which is known as static slice, was introduced by Weiser [1981]. Given a variable \( v \) of interest, the static slice (denoted as \( s_{\text{slice}}(v) \)) is a set of statements that have potential to affect the value of \( v \). Weiser [1981] computed the static slices by considering the consecutive sets of transitively relevant statements, according to data flow and control flow dependences and removing the unrelated program statements. This concept was originally proposed for the purpose of program debugging.

Later, more types of program slices were proposed. Two major types of slices were known as dynamic slice and execution slice. Different from the static slice that considers the static information only, these two types rely on a specific test case input. Given a target variable \( v \) and a specific test case \( t \), the dynamic slice (denoted as \( d_{\text{slice}}(v, t) \)), is the set of statements that actually affect (in backward slicing), or are affected by (in forward slicing) the value of \( v \) in the execution with \( t \) [Korel and Laski, 1988; Korel and Yalananchili, 1994; Zhang and Gupta, 2004; Gupta et al., 2005; Binkley et al., 2006a]. And the execution slice considers no variables. Given a specific test case \( t \), the execution slice (denoted as \( e_{\text{slice}}(t) \)), is a set of statements that have been covered in the test execution with \( t \) [Agrawal et al., 1995; Wong et al., 2005; Wong and Qi, 2006]. Execution slice is a commonly adopted program spectrum in SBFL, which considers statement and its binary coverage status in a test execution. Compared with the static slice, dynamic slice and execution slice can capture the dynamic behaviours of a program with respect to a specific test case, and hence are much smaller in sizes. Thus, they are more popular in practice.

Program slicing has been widely applied in many areas, such as reverse engineering [Beck and Eichmann, 1993; Jackson and Rollins, 1994], software maintenance [Gallagher and Lyle, 1991], program comprehension [De Lucia et al., 1996], software testing and debugging [Weiser, 1982; Agrawal et al., 1993, 1995; Gupta et al., 2005; Wong et al., 2005; Zhang et al., 2005; Wong and Qi, 2006], etc.
In Chapter 3, we are going to provide a novel concept of program slice, namely, metamorphic slice, which is based on the integration of MT and program slicing. We will demonstrate one of its applications in SBFL, to address the first investigated problem in this thesis, namely, the unrealistic assumption of having test oracle.
3

SBFL in the absence of test oracle

3.1 Preliminary

As discussed in previous chapters, in order to evaluate the risk values for the program statements, SBFL requires information of program spectrum that associated with conventional testing results. For instance, when using the execution slice as a representative of the program spectrum, each slice must be associated with the testing result of an individual test case, in terms of failed or passed. In other words, all the existing SBFL techniques have assumed the existence of a test oracle.

However, such assumption is not always true. Many real-life programs, including complex computational programs, bioinformatics applications, machine learning algorithms, etc. [Baker and Thornton, 2004; Xie et al., 2011a], do not have test oracles. Thus, this assumption has severely restricted the application of SBFL. In this chapter, we are going to focus on our first investigated problem in Section 1.2, by studying the following research questions.

1. **RQ1:** How can we apply SBFL techniques to the software application domains without test oracle?

2. **RQ2:** How effective is our proposed solution?

Since metamorphic testing has been proposed to alleviate the oracle problem, it is natural to consider how to integrate MT into SBFL, to support the application of SBFL in the programs without test oracle. Thus, in the following section, we propose a new type of program slice, namely, metamorphic slice, to implement such integration. Similar to the traditional program slice, metamorphic slice can be applied in SBFL as a type of program spectrum. More importantly, it inherits the ability of MT to alleviate the oracle problem for SBFL.
3.2 Metamorphic slice: a property-based program slice

In this section, we propose a novel concept of program slice, namely, metamorphic slice. It is based on the integration of metamorphic testing and program slicing. Different from the traditional program slice, metamorphic slice is not only data-based, but also property-based. Intuitively speaking, a metamorphic slice is a group of slices which are bound together with a specific program property (known as MR).

Corresponding to the traditional static slice, dynamic slice and execution slice, we can have static metamorphic slice, dynamic metamorphic slice and execution metamorphic slice, respectively. Since most of the current slicing-based debugging techniques have focused on dynamic slice and execution slice, in this thesis, we give the definitions of dynamic metamorphic slice and execution metamorphic slice as follows:

Definition 3.2.1. For a metamorphic relation MR, suppose \( T^S = \{t^S_1, t^S_2, \ldots, t^S_{ks}\} \) and \( T^F = \{t^F_1, t^F_2, \ldots, t^F_{kf}\} \) are its respective set of source test cases and set of follow-up test cases, such that \( T^S \) and \( T^F \) constitute a metamorphic test group \( g \).

- Given a variable \( v \), the dynamic metamorphic slice, \( d_{mslice}(v, MR, T^S) \) is the union of all \( d_{slice}(v, t) \), where \( t \in (T^S \cup T^F) \). That is,
  \[
  d_{mslice}(v, MR, T^S) = \bigcup_{i=1}^{ks} d_{slice}(v, t^S_i) \cup \bigcup_{i=1}^{kf} d_{slice}(v, t^F_i)
  \]

  Normally, all \( d_{slice}(v, t) \) involved in one \( d_{mslice}(v, MR, T^S) \) are of the same type, which is either forward or backward.

- The execution metamorphic slice, \( e_{mslice}(MR, T^S) \) is the union of all \( e_{slice}(t) \), where \( t \in (T^S \cup T^F) \). That is,
  \[
  e_{mslice}(MR, T^S) = \bigcup_{i=1}^{ks} e_{slice}(t^S_i) \cup \bigcup_{i=1}^{kf} e_{slice}(t^F_i)
  \]

Technically speaking, a metamorphic slice has bound the \( d_{slice}(v, t) \) (or \( e_{slice}(t) \)) of all test cases belonging to a metamorphic test group of MR. For a given MR, each \( d_{mslice} \) or \( e_{mslice} \) must be associated with a metamorphic testing result of violation or non-violation. More importantly, regardless of the availability of the testing result of failure or pass associated with each individual \( d_{slice} \) or \( e_{slice} \), such metamorphic testing result is always available. Therefore, if we replace the application of \( d_{slice} \) or \( e_{slice} \) in SBFL with the application of \( d_{mslice} \) or \( e_{mslice} \), respectively, we are always able to obtain sufficient information for fault localization, no matter whether the test oracle is available or not.
3.3. SBFL with $e_{\text{mslice}}$

As a consequence, for $RQ1$, we have found a solution to extend the application of SBFL to
the programs without test oracle. In next section, we will discuss about this solution in details.
Specifically, since $e_{\text{slice}}$ is one of the most widely adopted program spectra in the conventional
SBFL, we will illustrate how to utilize $e_{\text{mslice}}$ instead of $e_{\text{slice}}$ in our approach.

3.3 SBFL with $e_{\text{mslice}}$

In SBFL with $e_{\text{mslice}}$, the coverage information is provided by the $e_{\text{mslice}}$ of each metamorphic
test group $g_i$. Suppose there are $m$ metamorphic test groups in the current metamorphic test suite.
Then, there are $m$ $e_{\text{mslices}}$ and $m$ metamorphic testing results in total. With this information, as
shown in Figure 3.1, we can construct the counterparts for the matrix and vectors in the conventional
SBFL.

\[
\begin{align*}
\text{MTS} & : (g_1, g_2, \ldots, g_m) \\
\text{PG} & : \\
\begin{pmatrix}
    s_1 \\
    s_2 \\
    \vdots \\
    s_n
\end{pmatrix}
\begin{pmatrix}
    1/0 & 1/0 & \ldots & 1/0 \\
    1/0 & 1/0 & \ldots & 1/0 \\
    \vdots & \vdots & \ddots & \vdots \\
    1/0 & 1/0 & \ldots & 1/0
\end{pmatrix}
\text{MS} & : \\
\text{RE} & : (v/n, v/n, \ldots, v/n)
\end{align*}
\]

Figure 3.1: Essential information for SBFL with $e_{\text{mslice}}$

In Figure 3.1, the vector $MTS$ is the test suite containing $m$ metamorphic test groups. The $j^{th}$
column of matrix $MS$ represents the corresponding $e_{\text{mslice}}$ of $g_j$, in which the binary value of “1”
in the $i^{th}$ line denotes the membership of statement $s_i$ in this $e_{\text{mslice}}$; and 0 otherwise. Besides,
the $j^{th}$ element in vector $RE$ records the corresponding metamorphic testing result for $g_j$, with “$v$”
indicating violated and “$n$” indicating non-violated.

The transformation from Figure 2.1 (essential information for SBFL with $e_{\text{slice}}$) to Figure 3.1
(essential information for SBFL with $e_{\text{mslice}}$) basically consists of the following replacements.
Each individual test case $t_j$ is replaced by a metamorphic test group $g_j$; the $e_{\text{slice}}$ for each $t_j$
is replaced by $e_{\text{mslice}}$ for each $g_j$; and the testing result of failure or pass is replaced by the
metamorphic testing result of violation or non-violation, respectively. After such replacements, the
same procedure can be applied to reformulate the collected information into vector $A_i$ or each $s_i$, and
to evaluate the risk value $r_i$ of $s_i$ using a formula.
In the conventional SBFL using e_slice, a failed test case implies that a faulty statement is definitely included in the corresponding e_slice; while a passed test case does not provide a definite conclusion whether the corresponding e_slice is free of faulty statement. Similarly, a violated metamorphic test group implies that there is at least one failed test case within it. Even though we do not know which test cases are actually the failed ones, we still can conclude that a faulty statement must be included in the union of all the corresponding e_slices, that is, the e-mslice. On the other hand, a non-violated metamorphic test group does not provide a definite conclusion that all the involved test cases are passed, and the correctness of all statements in the current e-mslice is not guaranteed.

Conceptually speaking, it is obvious that our proposed approach is applicable in any program no matter whether the test oracle is available or not. But it is still interesting to know whether this approach is feasible in practice or not. Thus, for RQ2, we further conduct experimental studies to investigate the performance of our approach. Next section will describe all the experimental set-up.

### 3.4 Experimental set-up

#### 3.4.1 Testing objects and their metamorphic relations

In the investigation of the effectiveness of our approach, we have selected 9 programs of different sizes as our testing objects. Table 3.1 lists the number of the executable codes (eLOC), excluding blank lines, lines of left or right brace, etc. The data are collected by SLOCCount (version 2.26) from [SLOCCount, 2004].

For each program, we have enumerated three MRs. Descriptions about the programs and their MRs are as follows.

1. **grep**
   
   grep is a well-known UNIX utility written in C to perform pattern matching. Given a pattern to be matched and some input files for searching, grep searches these files for lines containing a match to the specified pattern. It supports three different versions of syntax for regular expression, namely, “Basic”, “Extended”, and “Perl”. By default, grep follows the “Basic” version, and when a match in a line is found, the whole line is printed to standard output. For example, grep can be invoked by command: `grep "[Gg]r?ep" myfile.txt`, where expression “[Gg]r?ep” is the specified regular expression to be matched, and “myfile.txt” is the input file. grep searches “myfile.txt” for lines containing a match to pattern “[Gg]r?ep”, and prints all lines containing “grep”, “Grep”, “gep” or “Gep”.

   However, testing grep is not a easy task, because it may be very difficult to verify the correctness of its output. As shown in the above example command, though we can check whether all the
3.4. EXPERIMENTAL SET-UP

Table 3.1: Executable line of codes

<table>
<thead>
<tr>
<th>Program</th>
<th>eLOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>grep</td>
<td>7309</td>
</tr>
<tr>
<td>SeqMap</td>
<td>1783</td>
</tr>
<tr>
<td>print_tokens</td>
<td>342</td>
</tr>
<tr>
<td>print_tokens2</td>
<td>355</td>
</tr>
<tr>
<td>replace</td>
<td>512</td>
</tr>
<tr>
<td>schedule</td>
<td>292</td>
</tr>
<tr>
<td>schedule2</td>
<td>269</td>
</tr>
<tr>
<td>tcas</td>
<td>135</td>
</tr>
<tr>
<td>tot_info</td>
<td>273</td>
</tr>
</tbody>
</table>

printed lines actually contain matches to the specified pattern, it is almost impossible to know whether grep has printed all the matched lines, unless we do an exhaustive examination through the entire file (namely, inspecting every single line of “myfile.txt”). Therefore, grep has the oracle problem, which makes it a suitable object for our study.

For this program, our MRs are related to the regular expression analyzer, which is one of its key components. Given the same input file to be scanned, the regular expressions in the source and follow-up test cases are denoted as \( r_s \) and \( r_f \), respectively. All the three MRs construct \( r_f \) that is equivalent to \( r_s \). As a consequence, the output of the follow-up test case (denoted as \( O_f \)) should be the same as the output of the source test case (denoted as \( O_s \)).

(1) **MR1: Completely decomposing the bracketed sub-expression**

In a bracketed regular expression \( \llbracket x_1 \ldots x_n \rrbracket \), where \( x_i \) is a single character, if these characters \( x_1, \ldots, x_n \) are continuous for the current locale\(^1\), they can be presented in a compressed way \( \llbracket x_1-x_n \rrbracket \). For such a bracketed regular expression, one of its equivalents

---

\(^1\)In our experiments, we consider the default C locale, where characters are sorted according to their ASCII codes.
is the complete decomposition of the bracket, by using the symbol “|” that means “or”. In MR1, we construct $r_f$ by completely decomposing such bracketed sub-expressions in $r_s$. For example, if $r_s$ contains a sub-expression “[abcdef]” or “[a−f]”, then we have “a|b|c|d|e|f” instead in $r_f$.

(2) **MR2: Splitting the bracketed structure**

Consider the bracketed regular expression “[x₁…xₙ]” or “[x₁−xₙ]” again. Another equivalent format is to split the bracket into 2 brackets, by using symbol “|”. In MR2, $r_f$ is constructed by replacing such sub-expression in $r_s$ with this equivalent. For example, if $r_s$ contains a sub-expression “[abcdef]” or “[a−f]”, then we have “[ab] | [c−f]” instead in $r_f$.

(3) **MR3: Bracketing simple characters**

Apart from the reserved words with special meanings, any simple character in a regular expression should be equivalent to itself enclosed by the brackets, that is, “a” is equivalent to “[a]” if a is not a reserved word. In MR3, $r_f$ is constructed by replacing some simple characters in $r_s$ with their bracketed formats. For example, if $r_s$ contains a sub-expression “abc”, then we have “[a][b][c]” instead in $r_f$.

2. **SeqMap**

Another program used in our experiment is *SeqMap*, a Short Sequence Mapping Tool in bioinformatics [Jiang and Wong, 2008]. Given a long reference string $t$ and a set of short strings $P=\{p_1, ..., p_k\}$, which consist of characters taken from the set of alphabets \{A, T, G, C\}, as well as a maximum number of mismatches $e$, *SeqMap* finds all substrings in $t$ such that each substring has an edit distance equal to or less than $e$ against some $p_i \in P$. Here edit distance refers to the number of operations required to transform one string to another. The valid edit operations include substitution, insertion and deletion. If a $p_i$ matches any substring in $t$ with not more than $e$ edit distance, it is said to be mappable, otherwise unmappable. *SeqMap* outputs all the mappable $p_i$ with optional information including the mapped location in $t$, the mapped substring of $t$, the edit distance of this mapping, etc.

Obviously, for each mappable $p_i$, it may be not difficult to verify the correctness of the printed mapping information. However, it is very expensive to check whether *SeqMap* has printed all the possible matching positions in $t$, or whether all the unmappable $p_i$ are indeed truly unmappable to $t$. In other words, soundness of the output is easy to verify, but not the completeness of the output. Therefore, *SeqMap* also has the oracle problem.

For this program, we construct MRs by modifying $t$ and $e$ while keeping $P$ unchanged. Given a set of short strings $P=\{p_1, ..., p_k\}$, a long reference string $t_s$ and the specified maximum of mismatches $e_s$ as a source test case, the output of the set of all mappable $p_i$ is denoted as $M_s$. Obviously, $M_s \subseteq P$. And the set of unmappable $p_i$ is $U_s=(P\setminus M_s)$. Let us denote the long reference string and the specified maximum of mismatches in the follow-up test case as $t_f$ and $e_f$, respectively. The sets of mappable and unmappable short strings produced by the follow-up
test case are referred to as $M_f$ and $U_f$, respectively.

(1) **MR1: Concatenating some elements of $P$ to $t_s$**
Suppose $P_1$ is a non-empty subset of $P$. $t_f$ is constructed by concatenating all elements in $P_1$ to the end of $t_s$ one by one. As a consequence,

- For any $p_i \in M_s$, we have $p_i \in M_f$. Thus, $M_s \subseteq M_f$.
- For each $p_i \in (M_s \cap P_1)$, the follow-up test case should have at least one additional mapping location in $t_f$.
- Each $p_i \in (U_s \cap P_1)$ should be mapped at least once in $t_f$, that is, we have $p_i \in M_f$.

(2) **MR2: Deleting a substring in $t_s$**
In this MR, $t_f$ is constructed from $t_s$ by deleting an arbitrary portion of strings at either the head or the end of $t_s$. As a consequence, for any $p_i \in U_s$, we have $p_i \in U_f$. Therefore, $U_s \subseteq U_f$.

(3) **MR3: Changing of $e_s$**
In this MR, $t_f = t_s$. And $e_f$ can be set to either greater or smaller than $e_s$.

- Consider the case that $0 \leq e_f < e_s$. Then, we have $M_f \subseteq M_s$.
- Consider the case that $0 \leq e_s < e_f$. Then, we have $M_s \subseteq M_f$.

3. *print_tokens* and *print_tokens2*
These two programs perform lexical parsing. They both read a sequence of strings from a file, group these strings into tokens, identify token categories and print out all the tokens and their categories in order. The main difference between these two programs is that *print_tokens* uses a hard-coded DFA; while *print_tokens2* does not.

Suppose the input files in source test case and follow-up test case are denoted as $I_s$ and $I_f$, respectively, and their respective outputs are denoted as $O_s$ and $O_f$. Each element in $O_s$ and $O_f$ has two attributes: the token category (e.g. keyword, identifier, etc.) and the string of this token.

For these two programs, we define MRs as follows.

(1) **MR1: Changing lower case into upper case**
In this MR, $I_f$ is constructed from $I_s$ by changing all characters in $I_s$ with lower cases into their upper cases. This operation does not affect the number of printed tokens, that is, we have the size of $O_f$ equal to the size of $O_s$. Since the “keywords” are case-sensitive, all the elements with categories of “keyword” in $O_s$ become “identifier”\(^2\) in $O_f$. For the non-keyword elements of $O_s$, the corresponding categories remain the same in $O_f$.

(2) **MR2: Deleting the comments**
In these two programs, any strings after a symbol “;” stand for the comment, which are

---

\(^2\)In *print_tokens* and *print_tokens2*, an identifier means any general string that is not a keyword, is not enclosed by quotes and consists of non-numeric and non-symbol characters.
3.4. EXPERIMENTAL SET-UP

not analyzed and printed as tokens. In this MR, \( I_f \) is constructed from \( I_s \) by deleting all comments in \( I_s \). Then, we have \( O_s = O_f \).

(3) **MR3: Inserting the comments**

In this MR, \( I_f \) is constructed from \( I_s \) by inserting the comment symbol “;” at the very beginning of some arbitrarily chosen lines. Then, we have \( O_f \subseteq O_s \).

4. **replace**

Program **replace** performs regular expression matching and substitution. It takes a regular expression \( r \), a replacement string \( s \) and an input file as input parameters. It produces an output file resulted from replacing any substring in the input file that is matched by \( r \), with \( s \). Instead of adopting the widely used Perl regular expression syntax, **replace** has its own syntax of regular expression.

Similar to **grep**, for **replace**, our MRs are also related to the regular expression analyzer. Given the same replacement string and input file, the regular expressions in the source and follow-up test cases are denoted as \( r_s \) and \( r_f \), respectively. All the three MRs involve \( r_s \) and \( r_f \) that are equivalent. As a consequence, the output of the follow-up test case should be the same as the output of the source test case. However, since the syntax for regular expression in **replace** does not completely comply with the Perl regular expression syntax, not every MR for **grep** can be used for **replace**.

(1) **MR1: Decompressing and permutating the expression in brackets**

The syntax of regular expression in **replace** also supports the bracketed expression \("[x_1-x_n]\) that matches any single character \( x_i \in \{x_1, \ldots, x_n\} \), where \("x_1, \ldots, x_n\) are continuous characters in the current locale. Actually, \("[x_1 \ldots x_n]\) is an equivalent of \("[x_1-x_n]\). In particular, the order of all \( x_i \) in \("[x_1 \ldots x_n]\) does not affect the interpretation of this expression. Thus, this MR constructs \( r_f \) with two types of replacements.

- The compressed sub-expression \("[x_1-x_n]\) in \( r_s \) is replaced by its completely decompressed equivalent \("[x_1 \ldots x_n]\).”
- The decompressed sub-expression \("[x_1 \ldots x_n]\) in \( r_s \) is replaced by \("[x_1 \ldots x_i_n]\), where \("<x_1, \ldots, x_i_n>\) is a permutation of \("<x_1, \ldots, x_n>\).”

For example, if \( r_s \) contains a sub-expression \("[a-c]\), then we have \("[abc]\) instead in \( r_f \). For another example, if \( r_s \) contains a sub-expression \("[abc]\), then \( r_f \) has \("[cba]\) instead.

(2) **MR2: Bracketing simple characters**

This MR is basically the same as the MR3 of **grep**. As a reminder, **grep** and **replace** have different lists of reserved characters.

(3) **MR3: Escaping reserved characters with bracketed structure**

For **replace**, symbol “@” is the escape character. Any reserved character of **replace** (e.g. “$”, “%”, etc.) after “@”, is interpreted as its original meaning. Actually there is another
way to preserve the original meaning for these reserved characters, that is, to bracket the reserved characters individually. For example “@$” and “[$]” both mean “$”. In MR3, the \( r_f \) is constructed by replacing the escaped reserved characters by “@" in \( r_s \) with the bracketed format.

5. \textit{schedule} and \textit{schedule2}

These two programs perform priority scheduling, which internally maintain 4 mutually exclusive job lists:

- Three priority job-lists \( P_1, P_2 \) and \( P_3 \), with \( P_3 \) and \( P_1 \) indicating the highest and lowest priorities, respectively. Each list contains a list of jobs with the same priority.
- One blocked job list \( P_B \) contains all jobs currently suspended.

There is one and only one job pointer pointing to the currently activated job \( p_a \), which is the first job in list \( P_i \) with the highest priority in the system. At any time there is at most one activated job in the system. Each job has two attributes: the Job\_ID and the Job\_Priority. The Job\_ID is unique for a job. For a newly created job, the Job\_ID is assigned in an incremental manner, and the Job\_Priority indicates the priority value of 1, 2 or 3.

The basic functionality of \textit{schedule} and \textit{schedule2} is the same, except that \textit{schedule} is non-preemptive and \textit{schedule2} is preemptive. Both of them accept three integers \( (a_1, a_2 \) and \( a_3) \) and an input file as the input parameters, where \( a_i \) specifies the initial number of jobs in \( P_i \). The input file contains a series of commands that specify various operations on the job lists. There are 7 types of operations including addition of job, deletion of job, increasing the priority of a job, etc. The operations are coded from 1 to 7.

Given the same \( a_i \) \((1 \leq i \leq 3)\), the input files containing a series of commands in the source test case and follow-up test case are denoted as \( C_s \) and \( C_f \), respectively. For \textit{schedule}, we define the MRs as follows.

(1) **MR1: Substituting the quantum expire command**

For \textit{schedule}, command coded in integer “5” is called “QUANTUM_EXPIRE” command, which releases the currently activated job and puts it to the end of the corresponding job list. Command coded in integer “3” is called “BLOCK” command, which puts the currently activated job to the block list \( P_B \). Its reverse command “UNBLOCK” command is coded in “4” that takes a ratio \( r \) as a parameter. The index of the selected job to be unblocked from \( P_B \) is determined by multiplying the length of \( P_B \) by the ratio \( r \). When processing command “4 \( r \)”, \textit{schedule} first releases the selected job from \( P_B \), and then puts it to the end of the corresponding job list. Therefore, command “5” can be re-interpreted as command “3”, followed by “4 1.00”. By using “3” and “4 1.00”, \textit{schedule} first blocks the currently activated job (denoted as \( p_a \)), puts it at the end of \( P_B \), then unblocks the last job in \( P_B \), that is, \( p_a \), (since the length of \( P_B \) multiplied by 1.00 indicates the last index in \( P_B \)) and puts
3.4. EXPERIMENTAL SET-UP

this job to the end of the corresponding job list. Obviously, consecutively processing these two commands has the same effect as solely processing command “5”. Thus, in MR1, \( C_f \) is constructed by replacing command “5” in \( C_s \) with commands “3” and “4 1.00”. And the output of the follow-up test case (denoted as \( O_f \)) should be the same as the output of the source test case (denoted as \( O_s \)).

(2) **MR2: Substituting the adding job command**

For \( schedule \), command coded in “1” is called “NEW_JOB” command, which takes an integer \( i (1 \leq i \leq 3) \) as its parameter to specify the priority, and adds a new job to the end of \( P_i \). And command coded in “2” is the “UPGRADE_PRIOR” command, which promotes a job from its current priority job list \( (P_i) \) into the next higher priority job list \( (P_{i+1}) \), where \( i = 1 \) or \( 2 \). This command takes two parameters. The first one is an integer \( i \) of 1 or 2, which specifies the current priority job list \( P_i \). The second parameter is a ratio \( r \), and the index of the selected job in \( P_i \) to be upgraded is determined by multiplying the length of \( P_i \) by the ratio \( r \). As a consequence, directly adding a new job with priority \( (i+1) \) has the same effect as adding a job with priority \( i \) and then promoting it to the job list with priority \( (i+1) \). Thus, in MR2, \( C_f \) is constructed by replacing command “1 \( i+1 \)” in \( C_s \) with command “1 \( i \)” followed by “2 \( i \).00” (\( i = 1 \) or 2). And \( O_f \) should be the same as \( O_s \).

(3) **MR3: Substituting the block and unblock commands**

As mentioned above, commands “3” and “4” are reversed to each other. Command “3” increases the job number in \( P_B \); while command “4” has the opposite actions. Therefore, we can make some replacements on these two types of commands, without changing the number of blocked jobs.

Suppose that a list of consecutive commands consists of commands “1”, “2”, “3”, “4” or “5”, and the numbers of commands “3” and “4” in this list are \( m \) and \( n \), respectively. Besides each of these two types of commands has actually blocked or unblocked a job, rather than operating on a “NULL” job pointer. Then, after processing this command list, there should be \( k=m-n \) jobs in \( P_B \). Therefore, we can remove all the commands “3” and “4” and insert the following commands in this command list, without changing the value of \( k \):

- If \( k>0 \), insert command “3” \( k \) times.
- If \( k<0 \), insert command “4” \( k \) times (the parameter of ratio can be any value within \([0.00, 1.00]\)).
- If \( k=0 \), insert no command.

In MR3, we search for such sub-lists of commands in \( C_s \) and apply the above removal and insertion operations to construct \( C_f \). And the number of printed jobs in \( O_f \) and \( O_s \) must be the same.

And for \( schedule_2 \), we also use the above MRs. However, since \( schedule_2 \) is a preemption scheduler, rescheduling happens in \( schedule_2 \) but not in \( schedule \), whenever a job is added,
unblocked or promoted to a higher priority list. Thus, in each MR of schedule2, only the number of printed jobs in \( O_f \) and \( O_s \), but not necessarily their contents, must be the same.

6. **tcas**

This program is a module of an on-board aircraft conflict detection and resolution system used by commercial aircraft. The entire system continuously monitors the radar information to check whether there is any neighbour aircraft (called intruder) that may give rise to a potential crash. **tcas** takes 12 integer parameters, including the altitudes of the controlled aircraft and the intruder aircraft, etc. It outputs a Resolution Advisory (RA) to advise the pilot to climb (UPWARD), descend (DOWNWARD) or remain the current trajectory (UNRESOLVEDWARD), as follows.

- if \( A \land B \land C \) is true, then \( RA = \text{UPWARD} \);
- if \( A \land \neg B \land D \) is true, then \( RA = \text{DOWNWARD} \);
- otherwise, \( RA = \text{UNRESOLVEDWARD} \).

where \( A, B, C \) and \( D \) are boolean expressions that involve input parameters. Their definitions are as follows.

\[
A = (\text{High Confidence} = 1) \land \\
(\text{Own_Tracked_Alt Rate} \leq 600) \land \\
(\text{Cur Vertical Sep} > 600) \land \\
(A_a \lor A_b) \quad (3.4.1)
\]

\[
A_a = (\text{Other Capability} = 1) \land \\
(\text{Two Of Three Report Valid} = 1) \land \\
(\text{Other RAC} = 0) \quad (3.4.2)
\]

\[
A_b = (\text{OtherCapability} \neq 1) \quad (3.4.3)
\]

\[
B = ((\text{Climb Inhibit} = 1) \land (\text{Up Separation} + 100 > \text{Down Separation})) \lor \\
((\text{Climb Inhibit} \neq 1) \land (\text{Up Separation} > \text{Down Separation})) \quad (3.4.4)
\]

\[
C = (\text{Own_Tracked_Alt} < \text{Other_Tracked_Alt}) \land \\
(\text{Down Separation} < \text{Positive RA Alt Thresh}[\text{Alt Layer Value}]) \quad (3.4.5)
\]

\[
D = (\text{Other_Tracked_Alt} < \text{Own_Tracked_Alt}) \land \\
(\text{Up Separation} \geq \text{Positive RA Alt Thresh}[\text{Alt Layer Value}]) \quad (3.4.6)
\]

For this program, we define MRs as follows.

1. **MR1: Modifying the Own_Tracked_Alt and Other_Tracked_Alt**

   These two parameters are signed integers that indicate the current altitudes of the controlled aircraft and the intruder aircraft, respectively.

   Suppose \( \text{Own_Tracked_Alt} \) and \( \text{Other_Tracked_Alt} \) in the source test case \( t_s \) are denoted as \( X_1^s \) and \( X_2^s \), respectively. Let us define \( \text{AVE}_s = (X_1^s + X_2^s)/2 \). Then, the follow-up test case
3.4. EXPERIMENTAL SET-UP

$t_f$ is constructed by replacing $X_1^s$ and $X_2^s$ in $t_s$ with $X_1^f$ and $X_2^f$ in the following ways, according to the output of $t_s$ (denoted as $O_s$).

- If $O_s=\text{UPWARD}$, we should have $A \land B \land C$ to be true. From Equation (3.4.5), $X_1^l<X_2^l$.
  Then in $t_f$, set $X_1^f=AV E_s-1$ and $X_2^f=AV E_s+1$.
- If $O_s=\text{DOWNWARD}$, we should have $A \land \lnot B \land D$ to be true. From Equation (3.4.6), $X_2^l<X_1^l$. Then in $t_f$, set $X_1^f=AV E_s+1$ and $X_2^f=AV E_s-1$.
- If $O_s=\text{UNRESOLVEDWARD}$, then in $t_f$, set $X_1^f=X_2^f=AV E_s$.

Obviously, the truth values of $A$ and $B$ are the same for $t_s$ and $t_f$. Hence, the relation between $\text{Own\_Tracked\_Alt}$ and $\text{Other\_Tracked\_Alt}$ remains unchanged, and consequently, the truth values of $C$ and $D$ are unchanged. Therefore, the output of $t_f$ (denoted as $O_f$) will be the same as $O_s$.

(2) MR2: Modifying the $\text{Up\_Separation}$ and $\text{Down\_Separation}$

These two parameters are signed integers that indicate the vertical separation between the two aircrafts when they reach the closest point of approach if the controlled aircraft performs an upward and downward maneuver, respectively.

Suppose $\text{Up\_Separation}$ and $\text{Down\_Separation}$ in the source test case $t_s$ are denoted as $Y_1^s$ and $Y_2^s$, respectively. Then, the follow-up test case $t_f$ is constructed by replacing $Y_1^s$ and $Y_2^s$ in $t_s$ with $Y_1^f$ and $Y_2^f$ in the following way:

- If $O_s=\text{UPWARD}$, then set $Y_1^f=Y_1^s$ and $Y_2^f < Y_2^s$. Then, the truth values of $A$, $B$ and $C$ are unchanged. Thus, $O_f=O_s$.
- If $O_s=\text{DOWNWARD}$, then set $Y_1^f > Y_1^s$ and $Y_2^f = Y_2^s$, such that $Y_1^f + 100 \leq Y_2^f$ if the parameter $\text{Climb\_Inhibit}$ in $t_s$ is 1; otherwise $Y_1^f \leq Y_2^f$. Then, the truth values of $A$, $B$ and $D$ are unchanged. Thus, $O_f=O_s$.
- If $O_s=\text{UNRESOLVEDWARD}$, then set $Y_1^f < Y_1^s$ and $Y_2^f > Y_2^s$, such that the relation between $Y_1^f$ and $Y_2^s - 100$ is the same as the relation between $Y_1^s$ and $Y_2^s - 100$, if $\text{Climb\_Inhibit}$ in $t_s$ is 1; otherwise, the relation between $Y_1^f$ and $Y_2^f$ is the same as the relation between $Y_1^s$ and $Y_2^s$. Then, according to Equations (3.4.1) to (3.4.6), we have $O_f=O_s$.

(3) MR3: Modifying the $\text{Alt\_Layer\_Value}$

tcas has a built-in array $\text{Positive\_RA\_Alt\_Threshold[4]} = \{400, 500, 640, 740\}$, which defines four threshold levels. The vertical separation between two aircrafts at their closest point of approach is considered adequate if it is greater than the specified threshold level. The parameter $\text{Alt\_Layer\_Value}$ is used to specify which threshold level is used for reference.

Let $Z_s$ denotes $\text{Alt\_Layer\_Value}$ in $t_s$. Then, the follow-up test case $t_f$ is constructed by replacing $Z_s$ in $t_s$ with $Z_f$ as follows:

- If $O_s=\text{UPWARD}$, set $Z_f=Z_s+1$ (0 $\leq Z_s < 3$). Then, the truth values of $A$, $B$ and $C$ are unchanged. Thus, $O_f=O_s$.
- If $O_s=\text{DOWNWARD}$, set $Z_f=Z_s-1$ (0 $< Z_s \leq 3$). Then, the truth values of $A$, $B$ and $D$ are unchanged. Thus, $O_f=O_s$.
3.4. EXPERIMENTAL SET-UP

- If $O_s =$ UNRESOLVEDWARD,
  - If in $t_s$, Own $\text{Tracked}_{\text{Alt}} \leq $ Other $\text{Tracked}_{\text{Alt}}$, set $Z_f = Z_s - 1 \ (0 < Z_s \leq 3)$.
  - Otherwise, set $Z_f = Z_s + 1 \ (0 \leq Z_s < 3)$.
According to Equations (3.4.1) to (3.4.6), we have $O_f = O_s$.

7. tot info
Given a source test case having $n$ tables $T_s = \{ t_1^s, t_2^s, \ldots, t_n^s \}$. For each table $t_i^s \in T_s$, tot info
prints $(info)_i^s$, $(df)_i^s$ and $(q)_i^s$, where $(info)_i^s$, $(df)_i^s$ and $(q)_i^s$ denote the Kullbacks information
measure, the degree of freedom and the possibility density of $\chi^2$ distribution of $t_i^s$, respectively. In addition, tot info
prints $(tot\_info)^s$, $(tot\_df)^s$ and $(q)^s$, where $(tot\_info)^s$ and $(tot\_df)^s$ are the summaries of all $(info)_i^s$ and all $(df)_i^s$, respectively, and $(q)^s$ is the possibility density of $\chi^2$
distribution calculated with $(tot\_info)^s$ and $(tot\_df)^s$. Let us denote the $m$ tables in the follow-
ap test case as $T_f = \{ t_1^f, t_2^f, \ldots, t_m^f \}$. For $t_f$, the printed results are denoted as $(info)_i^f$, $(df)_i^f$, $(q)_i^f$,
$(tot\_info)^f$, $(tot\_df)^f$ and $(q)^f$. The MRs for tot info are as follows.

1) MR1: Duplicating the whole input file
In this MR, $T_f$ is constructed by duplicating all $t_i^s \in T_s$. Then, we have $(tot\_info)^f = 2 * (tot\_info)^s$ and $(tot\_df)^f = 2 * (tot\_df)^s$.

2) MR2: Duplicating one table
For an arbitrarily chosen table $t_i^s \in T_s$, suppose $t_i^s$ has $r_i^s$ rows and $c_i^s$ columns. In $T_f$, $t_i^f$ is
defined to have $r_i^f$ rows and $c_i^f$ columns such that $r_i^f = 2 * r_i^s$ and $c_i^f = c_i^s$. And the content
from the $(r_i^s + 1)^{th}$ row to the $(2 * r_i^s)^{th}$ row in $t_i^f$ is the duplicate of its first $r_i^s$ rows. As a
consequence, we have $(info)_i^f = 2 * (info)_i^s$ and $(df)_i^f = (df)_i^s + r_i^s * (c_i^s - 1)$.

3) MR3: Scaling the value of each element in one table
For an arbitrarily chosen table $t_i^s \in T_s$, $T_f$ is constructed by defining $t_i^f$ such that each value in
t_i^f is the corresponding value in t_i^s multiplied by $k$. Then, we have $(info)_i^f = k * (info)_i^s$ and
$(df)_i^f = (df)_i^s$.  

Among all the 9 object programs, the source code for grep used in our experiments is v0 in
grep1.2 that is obtained from the Software-artifact Infrastructure Repository (SIR) website [SIR, 2005]. The program of SeqMap used in our experiments is v1.0.8 that is downloaded from its
homepage [SeqMap, 2008]. This version contains 10 head files and 1 source file (in C++). And
we only target at the following three files: probe.h, probe_match.h, and match.cpp, which implement
the major functionalities of the program. And for the last 7 programs, the versions used in our
experiments are 2.0, which are downloaded from the website of SIR [SIR, 2005].
3.4. EXPERIMENTAL SET-UP

3.4.2 Source test suite generation

As stated in Section 2.2, the source test suite is generated independently of the construction of MRs and the follow-up test suite. In our experiments, source test suites for all the 9 programs are obtained as follows.

For grep, we use a test pool with 171,634 random test cases, from one of our previous studies. Though the SIR has 807 test cases, we do not utilize them because only quite a small portion of them are “eligible source test cases” with respect to our MRs. In other words, most of these test cases do not contain specific substrings such that our MRs are applicable. However, with our test pool, we are able to have sufficient eligible source test cases for the construction of follow-up test cases. Among all the test cases in this test pool, 2982 test cases are eligible source test cases for MR1, 5003 for MR2, and 2084 for MR3.

For SeqMap, since there is no existing test suite, we use our randomly generated test cases from a previous study as the source test cases [Chen et al., 2009]. This source test suite contains 300 test cases. Each test case consists of a set of short strings \( P \), a long reference string \( t \) and the maximum number of mismatches \( e \). \( t \) and \( e \) are the same in all the 300 test cases; while \( P \) varies in different test cases.

For the 7 programs in Siemens Suite, we simply adopt the “universe” test plans provided by SIR [SIR, 2005], as our source test suites. The numbers of the source test cases for all the 9 programs are listed in Table 3.2. As a reminder, in SeqMap and the 7 programs of Siemens Suite, all the source test cases are eligible for all their MRs.

3.4.3 Mutant generation

SIR has several released mutants for Siemens Suite and grep. However, many of them are equivalent mutants with respect to our MRs, and hence no violation will be expected. In the previous studies of SBFL with \( e_{slice} \), it has been adopted that mutants with no failure revealed were excluded. Likewise, mutants with no violation are also excluded in our investigation. Therefore, for the Siemens Suite and grep programs, apart from their feasible mutants, we randomly generate 300 extra mutants. For SeqMap, since there is no existing mutant available, we generate 300 mutants randomly.

Our mutant generation focuses on the non-omission and first-order faults (that is, single-fault mutants). Multiple-fault scenario will be investigated in future. Two types of mutant operators are used: statement mutation and operator mutation. For statement mutation, either a “continue” statement is replaced with a “break” statement (and vice versa), or the label of a “goto” statement is replaced with another valid label. For operator mutation, an arithmetic (or a logical) operator is substituted by a different arithmetic (or logical) operator. To generate a mutant, our tool first randomly
### Table 3.2: Number of source test cases

<table>
<thead>
<tr>
<th>Program</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>grep</td>
<td>2982 (MR1)</td>
</tr>
<tr>
<td></td>
<td>5003 (MR2)</td>
</tr>
<tr>
<td></td>
<td>2084 (MR3)</td>
</tr>
<tr>
<td>SeqMap</td>
<td>300</td>
</tr>
<tr>
<td>print tokens</td>
<td>4130</td>
</tr>
<tr>
<td>print tokens2</td>
<td>4115</td>
</tr>
<tr>
<td>replace</td>
<td>5542</td>
</tr>
<tr>
<td>schedule</td>
<td>2650</td>
</tr>
<tr>
<td>schedule2</td>
<td>2710</td>
</tr>
<tr>
<td>tcas</td>
<td>1608</td>
</tr>
<tr>
<td>tot info</td>
<td>1052</td>
</tr>
</tbody>
</table>

### Table 3.3: Number of mutants

<table>
<thead>
<tr>
<th>Program</th>
<th>MR1</th>
<th>MR2</th>
<th>MR3</th>
</tr>
</thead>
<tbody>
<tr>
<td>grep</td>
<td>78</td>
<td>32</td>
<td>36</td>
</tr>
<tr>
<td>SeqMap</td>
<td>49</td>
<td>27</td>
<td>21</td>
</tr>
<tr>
<td>print tokens</td>
<td>24</td>
<td>32</td>
<td>21</td>
</tr>
<tr>
<td>print tokens2</td>
<td>31</td>
<td>27</td>
<td>21</td>
</tr>
<tr>
<td>replace</td>
<td>49</td>
<td>53</td>
<td>42</td>
</tr>
<tr>
<td>schedule</td>
<td>31</td>
<td>27</td>
<td>40</td>
</tr>
<tr>
<td>schedule2</td>
<td>36</td>
<td>37</td>
<td>21</td>
</tr>
<tr>
<td>tcas</td>
<td>16</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>tot info</td>
<td>21</td>
<td>27</td>
<td>28</td>
</tr>
</tbody>
</table>
selects a line in the relevant part of the source code, and then searches systematically for possible locations where a mutant operator can be applied. One of these mutant operators is then selected randomly and applied randomly to one of the possible locations.

For each program, we exclude the following mutants: the mutants which could not be compiled successfully, the mutants without any violated metamorphic test group, and the mutants with an exceptional exit (crash-failure) because their coverage information could not be correctly collected. The numbers of the feasible mutants for each program in each MR are shown in Table 3.3.

### 3.4.4 Experimental methodology

As discussed in Section 3.3, in our approach with $e_{mslice}$, the procedure to evaluate the risk value for each statement is the same as the conventional SBFL. In our experiments, we have chosen three risk evaluation formulas, namely Tarantula [Jones et al., 2002], Jaccard [Chen et al., 2002a] and Ochiai [Abreu et al., 2006], whose definitions are listed in Table 3.4.

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula expression $R(A)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tarantula</td>
<td>$\frac{a_{ef}}{a_{ef} + a_{nf}} \left( \frac{a_{ef}}{a_{ef} + a_{nf}} + \frac{a_{ep}}{a_{ep} + a_{np}} \right)$</td>
</tr>
<tr>
<td>Jaccard</td>
<td>$\frac{a_{ef}}{a_{ef} + a_{nf} + a_{np}}$</td>
</tr>
<tr>
<td>Ochiai</td>
<td>$\frac{a_{ef}}{\sqrt{(a_{ef} + a_{nf})(a_{ef} + a_{np})}}$</td>
</tr>
</tbody>
</table>

In our experiments, we also conduct the conventional SBFL with $e_{slice}$, and use its performance as a benchmark for our approach. We do such comparison because the testing community generally regards SBFL as effective for the programs with test oracle. Thus, it is natural to investigate whether our proposed approach can deliver a similar performance as the existing SBFL techniques for the situation with test oracle. In essence, we are interested to see whether there is a significant impact on the effectiveness when the traditional $e_{slice}$ is replaced by $e_{mslice}$ and the testing result of failure or pass is replaced by the metamorphic testing result of violation or non-violation. If there is no significant difference in effectiveness, our approach can be considered as a feasible approach in practice. Overall, we conduct SBFL under the following four scenarios:

- **MS**: SBFL using $e_{mslice}$ with all eligible metamorphic test groups
- **S-ST**: SBFL using $e_{slice}$ with all eligible source test cases
3.5. RESULTS AND ANALYSIS

- S-FT: SBFL using $e_{slice}$ with all eligible follow-up test cases
- S-AT: SBFL using $e_{slice}$ with all eligible source and follow-up test cases

We conduct experiments on these four scenarios because:

1. The number of $e_{slice}$ used in S-ST or S-FT is the same as the number of $e_{mslice}$ used in MS, but the number of $e_{slice}$ used in S-AT is doubled. Therefore, S-ST, S-FT and MS have the same amount of raw data for the risk evaluation formulas, and hence their performance evaluation is expected to have the same degree of reliability.

2. MS and S-AT have the same number of test executions, which is twice the number of test executions of S-ST and S-FT. Thus the program execution overheads of MS and S-AT are the same, while the program execution overheads of S-ST and S-FT are also similar, but about half of those for either MS or S-AT.

As a reminder, S-ST, S-FT and S-AT require a test oracle. Similar to some previous studies, we use the non-mutated versions as the assumed test oracles in these three scenarios in our experiments. For statements with the same risk values, we rank them according to the tie-breaking scheme of original order in the source code. It should be noted that there are other tie-breaking schemes [Wong et al., 2008, 2010]. All experiments are conducted on a cluster of 64-bit Intel Clovertown systems running CentOS 5, and the statement coverage is collected by using gcov.

3.5 Results and analysis

In our experiments, we investigate the effectiveness of our approach from three aspects: the EXAM score distribution, the Wilcoxon-Signed-Rank Test and the average EXAM scores.

3.5.1 EXAM score distribution

We utilize the line diagram to present the EXAM distribution of MS, S-ST, S-FT and S-AT, for all possible combinations of program and risk evaluation formula. The results are presented in Figure 3.2. Each graph collects and summarizes the results over all the three MRs, for a particular combination of program and formula. In these figures, the horizontal axis means the EXAM score, while the vertical axis means the cumulative percentage of the mutants whose EXAM scores are less than or equal to the corresponding EXAM score. Obviously, the faster that the curve increases and reaches 100% of mutants, the better performance the technique has.

Generally speaking, we can observe from these graphs that the performance of MS is comparable to the other three methods. Specifically, there are following situations.
3.5. RESULTS AND ANALYSIS

Figure 3.2: Distribution of EXAM scores (first part)
3.5. RESULTS AND ANALYSIS

Figure 3.2: Distribution of EXAM scores (second part)
Figure 3.2: Distribution of EXAM scores (third part)
Figure 3.2: Distribution of EXAM scores (fourth part)
3.5. RESULTS AND ANALYSIS

Figure 3.2: Distribution of EXAM scores (fifth part)
1. In the two relatively larger programs `grep` and `SeqMap`, MS provides quite satisfactory performance. As compared with S-ST, S-FT and S-AT, MS performs no worse or even better than the other three methods.

First, it is obvious in Figure 3.2(a) that for `grep`, the curves of MS are always above the other three methods, until they all achieve 100%, with all the three formulas. This indicates that MS has more mutants with low EXAM scores than the other three, that is, MS performs better. Actually, with all the three formulas, MS has quite a few mutants with EXAM scores less than 5%. And with the increasing of EXAM, the percentage of mutants also accumulates sharply, and achieves the 100% very quickly. For example, using formula Ochiai, MS already has over 45.2% of the mutants with EXAM smaller than 5%. Under the EXAM score of 10%, there are already over 89.7% of the mutants accumulated. And before EXAM of 15%, the curve has reached the top point, which means there are no mutant whose EXAM is greater than 15%.

Similar situation can be observed in Figure 3.2(b) for `SeqMap`. Especially with Tarantula, the advantage is obvious: over 97.9% of the mutants have EXAM smaller than 40%, while under the same EXAM score, S-ST, S-FT and S-AT only have 61.8%, 63.9% and 63.9% of the mutants, respectively. With the other two formulas, the differences in the performance are less significant. However, MS still performs better than S-ST.

2. In the seven smaller programs from Siemens Suite, though MS does not always perform the best, its performance is still comparable to the other three methods.

First, from Figure 3.2(g), in program `schedule2`, the results are similar to the ones in `grep` and `SeqMap`, that is, MS has better or similar performance as the other three methods.

Secondly, in program `print_tokens2`, MS does not outperform the other three methods, but they still have the similar performance. It can be seen in Figure 3.2(d) that the curves of MS are intersecting with the other three curves, such that MS performs slightly worse at the low EXAM score, but its curves increase sharply with the increase of the EXAM scores, and finally exceed the other three curves to reach 100% of mutants at a smaller EXAM score.

Finally, for the remaining programs, `replace`, `print_tokens`, `schedule`, `tcas` and `tot_info`, MS performs similarly to at least one of the other three methods using Tarantula formula, but worse than the other three methods using formulas Ochiai and Jaccard. For example, in program `tot_info` shown in Figure 3.2(i), it can be seen that when using formula Tarantula, the performance of MS is worse than the other three methods when the EXAM score is smaller than 10%, but becomes better when the EXAM score is greater than 10%. But with the other two formulas, the curves of MS are always underneath those of S-ST, S-FT and S-AT. However, even in these worse situations, the increasing trend of the MS is satisfactory. It can be seen that with either formula Ochiai or Jaccard, MS has only 30% of mutants having smaller than 5% EXAM scores, and has nearly 90% of mutants having smaller than 15% EXAM scores. However, MS almost catches up with the
3.5. RESULTS AND ANALYSIS

other three methods at the EXAM score of 25% to 30%. Similar situations can be found in other programs.

In summary, the EXAM score distribution does not show any visually significant difference between our approach and the existing techniques, that is, as compared with S-ST, S-FT and S-AT, MS may perform either better, similarly, or worse. Therefore, in next section, we further conduct the paired Wilcoxon-Signed-Rank Test to compare the effectiveness between MS and S-ST, S-FT, S-AT from a statistical perspective, investigating how “comparable” is the performance of our method using mslice to the conventional methods using slice.

3.5.2 Statistical comparison

The EXAM score distribution only provides a visual comparison and is not rigorous enough. Thus, we further conduct a more rigorous and scientific comparison, namely, the paired Wilcoxon-Signed-Rank Test. The paired Wilcoxon-Signed-Rank test is a non-parametric statistical hypothesis test that makes use of the sign and the magnitude of the rank of the differences between pairs of measurements \( F(x) \) and \( G(y) \), which do not follow a normal distribution [Corder and Foreman, 2009]. At the given significant level \( \sigma \), there are both 2-tailed p-value and 1-tailed p-value which can be used to obtain a conclusion.

For the 2-tailed p-value, if \( p \geq \sigma \), the null hypothesis \( H_0 \) that \( F(x) \) and \( G(y) \) are not significantly different is accepted; otherwise, the alternative hypothesis \( H_1 \) that \( F(x) \) and \( G(y) \) are significantly different is accepted. For 1-tailed p-value, there are two cases, the lower case and the upper case. In the lower case, if \( p \geq \sigma \), \( H_0 \) that \( F(x) \) does not significantly tend to be greater than the \( G(y) \) is accepted; otherwise, \( H_1 \) that \( F(x) \) significantly tends to be greater than the \( G(y) \) is accepted. And in the upper case, if \( p \geq \sigma \), \( H_0 \) that \( F(x) \) does not significantly tend to be less than the \( G(y) \) is accepted; otherwise, \( H_1 \) that \( F(x) \) significantly tends to be less than the \( G(y) \) is accepted.

In our experiments, we conduct three paired Wilcoxon-Signed-Rank tests, for MS v.s.S-ST, MS v.s.S-FT as well as MS v.s.S-AT, using both the 2-tailed and 1-tailed test, at the \( \sigma \) level of 0.05. Translated into our context, for a given combination of program and formula, the list of measurements of \( F(x) \) is the list of the EXAM scores for all the mutants in MS; while the list of measurements of \( G(y) \) is the list of the EXAM scores for all the mutants in either S-ST, S-FT or S-AT. Therefore, in the 2-tailed test, \( H_0 \) being accepted means that MS has SIMILAR performance as S-ST, S-FT or S-AT, at the significant level of 0.05. And in the 1-tailed test (lower), \( H_1 \) being accepted means that MS has WORSE performance than S-ST, S-FT or S-AT, at the significant level of 0.05. Finally, in the 1-tailed test (upper), \( H_1 \) being accepted means that MS has BETTER performance than S-ST, S-FT or S-AT, at the significant level of 0.05.

We use OriginPro 8.1 developed by OriginLab for this analysis. Table 3.5 to Table 3.7 present
the results for the comparison between MS and S-ST, MS and S-FT and MS and S-AT, respectively, where T, O and J stand for formulas Tarantula, Ochiai and Jaccard, respectively. And columns 2-tailed, 1-tailed (lower) and 1-tailed (upper) record the corresponding p-values.

Table 3.5: MS v.s. S-ST

<table>
<thead>
<tr>
<th>Program</th>
<th>2-tailed</th>
<th>1-tailed (lower)</th>
<th>1-tailed (upper)</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>grep</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.00748</td>
<td>0.99628</td>
<td>0.00374</td>
<td>BETTER</td>
</tr>
<tr>
<td>O</td>
<td>0.10282</td>
<td>0.9488</td>
<td>0.05141</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>J</td>
<td>0.21888</td>
<td>0.89093</td>
<td>0.10944</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>SeqMap</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>1.46218E-06</td>
<td>1</td>
<td>7.3109E-07</td>
<td>BETTER</td>
</tr>
<tr>
<td>O</td>
<td>0.99891</td>
<td>0.50164</td>
<td>0.49945</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>J</td>
<td>0.90321</td>
<td>0.54948</td>
<td>0.45161</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>print tokens</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.00659</td>
<td>0.00329</td>
<td>0.99677</td>
<td>WORSE</td>
</tr>
<tr>
<td>O</td>
<td>8.9407E-08</td>
<td>4.47035E-08</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td>J</td>
<td>1.41342E-05</td>
<td>7.06711E-06</td>
<td>0.99999</td>
<td>WORSE</td>
</tr>
<tr>
<td>print tokens2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.96774</td>
<td>0.51724</td>
<td>0.48387</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>O</td>
<td>0.15971</td>
<td>0.07986</td>
<td>0.92055</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>J</td>
<td>0.16053</td>
<td>0.08027</td>
<td>0.92013</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>replace</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.22033</td>
<td>0.89027</td>
<td>0.11017</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>O</td>
<td>0.47632</td>
<td>0.23816</td>
<td>0.76255</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>J</td>
<td>0.30655</td>
<td>0.15328</td>
<td>0.84724</td>
<td>SIMILAR</td>
</tr>
</tbody>
</table>
### Table 3.5: MS vs. S-ST (cont.)

<table>
<thead>
<tr>
<th>Program</th>
<th>2-tailed</th>
<th>1-tailed (lower)</th>
<th>1-tailed (upper)</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>schedule</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>4.3232E-07</td>
<td>2.1616E-07</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td>O</td>
<td>2.9894E-09</td>
<td>1.4947E-09</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td>J</td>
<td>5.6112E-10</td>
<td>2.8056E-10</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td><strong>schedule2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.00097049</td>
<td>0.99952</td>
<td>0.00048524</td>
<td>BETTER</td>
</tr>
<tr>
<td>O</td>
<td>0.19735</td>
<td>0.09867</td>
<td>0.90206</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>J</td>
<td>0.22816</td>
<td>0.11408</td>
<td>0.88676</td>
<td>SIMILAR</td>
</tr>
<tr>
<td><strong>teas</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.13688</td>
<td>0.06844</td>
<td>0.93253</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>O</td>
<td>4.1329E-07</td>
<td>2.0664E-07</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td>J</td>
<td>3.91155E-08</td>
<td>1.9557E-08</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td><strong>tot_info</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.04193</td>
<td>0.97921</td>
<td>0.02097</td>
<td>BETTER</td>
</tr>
<tr>
<td>O</td>
<td>0.00168</td>
<td>0.000837554</td>
<td>0.99918</td>
<td>WORSE</td>
</tr>
<tr>
<td>J</td>
<td>0.00187</td>
<td>0.000933166</td>
<td>0.99909</td>
<td>WORSE</td>
</tr>
</tbody>
</table>
### 3.5. RESULTS AND ANALYSIS

Table 3.6: MS v.s. S-FT

<table>
<thead>
<tr>
<th>Program</th>
<th>2-tailed</th>
<th>1-tailed (lower)</th>
<th>1-tailed (upper)</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>grep</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>2.66301E-07</td>
<td>1.33151E-07</td>
<td>BETTER</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>0.000114414</td>
<td>5.72071E-05</td>
<td>BETTER</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>0.000238973</td>
<td>0.000119486</td>
<td>BETTER</td>
<td></td>
</tr>
<tr>
<td><strong>SeqMap</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.000015654</td>
<td>7.82698E-06</td>
<td>BETTER</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>0.45093</td>
<td>0.77556</td>
<td>SIMILAR</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>0.34456</td>
<td>0.82859</td>
<td>SIMILAR</td>
<td></td>
</tr>
<tr>
<td><strong>print_tokens</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.12978</td>
<td>0.93588</td>
<td>SIMILAR</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>0.01372</td>
<td>0.99334</td>
<td>WORSE</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>0.00211</td>
<td>0.99897</td>
<td>WORSE</td>
<td></td>
</tr>
<tr>
<td><strong>print_tokens2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.44396</td>
<td>0.77882</td>
<td>SIMILAR</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>0.38151</td>
<td>0.80995</td>
<td>SIMILAR</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>0.13207</td>
<td>0.9343</td>
<td>SIMILAR</td>
<td></td>
</tr>
<tr>
<td><strong>replace</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.22251</td>
<td>0.11125</td>
<td>SIMILAR</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>0.6055</td>
<td>0.69799</td>
<td>SIMILAR</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>0.6538</td>
<td>0.67387</td>
<td>SIMILAR</td>
<td></td>
</tr>
</tbody>
</table>
## Table 3.6: MS v.s. S-FT (cont.)

<table>
<thead>
<tr>
<th>Program</th>
<th>2-tailed</th>
<th>1-tailed (lower)</th>
<th>1-tailed (upper)</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>schedule</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.01871</td>
<td>0.00935</td>
<td>0.99076</td>
<td>WORSE</td>
</tr>
<tr>
<td>O</td>
<td>6.60245E-09</td>
<td>3.30123E-09</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td>J</td>
<td>4.05146E-10</td>
<td>2.02573E-10</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td><strong>schedule2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.000702801</td>
<td>0.99965</td>
<td>0.000351401</td>
<td>BETTER</td>
</tr>
<tr>
<td>O</td>
<td>0.00584</td>
<td>0.00292</td>
<td>0.99712</td>
<td>WORSE</td>
</tr>
<tr>
<td>J</td>
<td>0.12099</td>
<td>0.06049</td>
<td>0.94005</td>
<td>SIMILAR</td>
</tr>
<tr>
<td><strong>teas</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.96421</td>
<td>0.48211</td>
<td>0.52065</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>O</td>
<td>0.000546162</td>
<td>0.000273081</td>
<td>0.99973</td>
<td>WORSE</td>
</tr>
<tr>
<td>J</td>
<td>0.000173005</td>
<td>8.65023E-05</td>
<td>0.99992</td>
<td>WORSE</td>
</tr>
<tr>
<td><strong>tot_info</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.11317</td>
<td>0.94382</td>
<td>0.05659</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>O</td>
<td>7.91943E-08</td>
<td>3.95971E-08</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td>J</td>
<td>1.43003E-07</td>
<td>7.15017E-08</td>
<td>1</td>
<td>WORSE</td>
</tr>
</tbody>
</table>
### 3.5. RESULTS AND ANALYSIS

Table 3.7: MS v.s. S-AT

<table>
<thead>
<tr>
<th>Program</th>
<th>2-tailed</th>
<th>1-tailed (lower)</th>
<th>1-tailed (upper)</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>grep</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.03322</td>
<td>0.98347</td>
<td>0.01661</td>
<td>BETTER</td>
</tr>
<tr>
<td>O</td>
<td>0.98409</td>
<td>0.50875</td>
<td>0.49204</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>J</td>
<td>0.87375</td>
<td>0.43688</td>
<td>0.5639</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>SeqMap</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>1.65726E-05</td>
<td>0.99999</td>
<td>8.28628E-06</td>
<td>BETTER</td>
</tr>
<tr>
<td>O</td>
<td>0.14614</td>
<td>0.07307</td>
<td>0.92731</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>J</td>
<td>0.13568</td>
<td>0.06784</td>
<td>0.93252</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>print_tokens</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.0039</td>
<td>0.00195</td>
<td>0.99809</td>
<td>WORSE</td>
</tr>
<tr>
<td>O</td>
<td>5.58794E-08</td>
<td>2.79397E-08</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td>J</td>
<td>1.31834E-06</td>
<td>6.5917E-07</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td>print_tokens2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.84935</td>
<td>0.42468</td>
<td>0.57639</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>O</td>
<td>0.18764</td>
<td>0.09382</td>
<td>0.90664</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>J</td>
<td>0.11254</td>
<td>0.05627</td>
<td>0.94403</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>replace</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.68337</td>
<td>0.34168</td>
<td>0.65919</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>O</td>
<td>0.19999</td>
<td>0.1</td>
<td>0.90039</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>J</td>
<td>0.1096</td>
<td>0.0548</td>
<td>0.94544</td>
<td>SIMILAR</td>
</tr>
</tbody>
</table>
### 3.5. RESULTS AND ANALYSIS

Table 3.7: MS vs. S-AT (cont.)

<table>
<thead>
<tr>
<th>Program</th>
<th>2-tailed</th>
<th>1-tailed (lower)</th>
<th>1-tailed (upper)</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>schedule</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>2.28286E-06</td>
<td>1.14143E-06</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td>O</td>
<td>5.73627E-09</td>
<td>2.86814E-09</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td>J</td>
<td>3.40643E-10</td>
<td>1.70322E-10</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td><strong>schedule2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.0001375</td>
<td>0.99993</td>
<td>6.87502E-05</td>
<td>BETTER</td>
</tr>
<tr>
<td>O</td>
<td>0.13507</td>
<td>0.06754</td>
<td>0.93303</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>J</td>
<td>0.14197</td>
<td>0.07098</td>
<td>0.9296</td>
<td>SIMILAR</td>
</tr>
<tr>
<td><strong>tcas</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.14073</td>
<td>0.07036</td>
<td>0.93062</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>O</td>
<td>3.70874E-07</td>
<td>1.85437E-07</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td>J</td>
<td>5.57982E-11</td>
<td>2.78991E-11</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td><strong>totinfo</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>0.38748</td>
<td>0.80732</td>
<td>0.19374</td>
<td>SIMILAR</td>
</tr>
<tr>
<td>O</td>
<td>9.84774E-08</td>
<td>4.92387E-08</td>
<td>1</td>
<td>WORSE</td>
</tr>
<tr>
<td>J</td>
<td>7.68342E-06</td>
<td>3.84171E-06</td>
<td>1</td>
<td>WORSE</td>
</tr>
</tbody>
</table>
Table 3.8 summarizes the numbers of SIMILAR, BETTER and WORSE cases in Table 3.5 to Table 3.7. It shows that with the exception of the comparisons of MS v.s.S-FT using Ochiai and Jaccard, all other comparisons have more BETTER and SIMILAR results than WORSE results. Overall, we have over 60% BETTER and SIMILAR results. Thus, the performance of MS is statistically comparable to the performance of the other three scenarios where the test oracle is assumed to be available.

Table 3.8: Summarized results

<table>
<thead>
<tr>
<th>Comparison</th>
<th>BETTER</th>
<th>WORSE</th>
<th>SIMILAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS v.s.S-ST T</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>O</td>
<td>0</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>J</td>
<td>0</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>MS v.s.S-FT T</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>O</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>J</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>MS v.s.S-AT T</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>O</td>
<td>0</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>J</td>
<td>0</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

3.5.3 Average EXAM score comparison

In this experiment we summarize the average EXAM score over all mutants and all MRs, for each pair of program and formula. The results are shown in Figure 3.3.

It can be seen from Figure 3.3 that in the comparison between MS and S-ST, when using formula Tarantula, the average EXAM scores of MS are smaller than those of S-ST in grep, SeqMap, print_tokens2, replace, schedule2 and tot_info. When using formula Ochiai, in programs grep, SeqMap and print_tokens2, MS can have smaller average EXAM scores than S-ST, while in other programs the average EXAM scores of MS are larger, but most of the differences are actually
insignificant. Finally, formula Jaccard has similar observations as Ochiai.

The comparison between MS and S-ST indicates that the performance of MS is comparable to the performance of S-ST. Besides, in the comparisons between MS and other two methods (S-FT and S-AT), we have similar conclusions.

In summary, MS can be quite a satisfactory substitution of the conventional SBFL with e_slice. The performance of MS can be quite close to the conventional methods, or could be even better. And MS may have violated metamorphic testing groups even when S-ST has passed results for the corresponding source test cases. More importantly, we should emphasize that the most significant contribution of our approach is to extend the application of conventional SBFL to the programs without test oracle. The performance comparison is mainly to substantiate the practicability of our approach.
3.5. RESULTS AND ANALYSIS

3.5.4 Real-life faults in Siemens Suite

A by-product of our experiments is the identification of two real-life faults in the Siemens Suite, despite that these programs have been extensively used and tested. This is understandable as metamorphic testing provides testing from a new perspective that has not been considered in the previous testing of the Siemens Suite. The two faults are described as follows.

1. **Fault 1:** The first fault is at line 214 of program schedule, in function: “void upgrade_process_prio(int prio, float ratio)”. This function is for command “2 prio ratio”, which upgrades a job from \( P_{\text{prio}} \) to its next higher priority. After processing the upgrade, the priority of the specified job should be increased by 1. But in line 214, the code is: “\( \text{proc} \rightarrow \text{priority} = \text{prio} \)”, rather than the correct “\( \text{proc} \rightarrow \text{priority} = \text{prio} + 1 \)”.

2. **Fault 2:** The second fault is in program print_tokens. In print_tokens and print_tokens2, there is a buffer with a length of 80 characters, to store the read-in tokens. However in print_tokens, even a token is shorter than 80 characters, it also may be read and printed incorrectly. This is due to the way that print_tokens deals with the blanks.

   According the specification, when reading in a token, these two programs should ignore all irrelevant blanks. In print_tokens2, before storing a valid token into the buffer, the program will first delete all irrelevant blanks. However, in print_tokens, a token will be stored into the buffer with its neighbour blanks. Removal of the blanks is performed afterward. Therefore, in print_tokens, the blanks occupy some space in the buffer which is of a fixed length of 80 characters, and hence, a valid token may only be partially stored by mistake.

3.5.5 Summary

According to the above analysis, we have the following conclusions.

1. The experimental results are positive that there is no significant impact on the performance of SBFL, when \( e \_slice \) is replaced by \( e \_mslice \). In other words, the performance of our approach is comparable to that of the conventional SBFL, which has been generally regarded as effective by the testing community.

2. Despite that the investigated programs are extensively used and well tested, the identification of two real-life faults in our study is a strong evidence that metamorphic testing provides a new perspective of generating test cases that can supplement other commonly used test case generation techniques. This further supports a conjecture in our recent study that diversity plays a key role in the fault detection effectiveness of test cases [Chen et al., 2010].
3.6 Threats to validity

3.6.1 External validity

The primary threat to the external validity is the generalizability of our results acquired from the 9 testing objects, with only three MRs for each. It is well known that the 7 programs in Siemens Suite are all small in size. Although grep and SeqMap are considerably larger than these 7 programs, they are still not very large-scaled programs. Thus it is worthwhile to use more large-scaled programs to further validate the effectiveness of our approach of using \( e_{mslice} \) in SBFL.

Another threat is the type of mutants (effectively, the type of faults) used in our study. Even though these mutants are randomly generated, but each mutant only contains exactly one fault and the types of faults are also restricted. Actually, our approach can be extended to programs with multiple faults. If there is a way (e.g., the fault focused approach in [Jones et al., 2007]) to segregate or cluster violated executions together such that violated test groups in each cluster are related to the same fault, then these relevant violated test groups along with some non-violated test groups can be used to locate a specific fault.

Besides the effectiveness of the MRs is not investigated in this thesis. We understand that identifying effective MRs is not a trivial task for some programs. Though such an identification has been easily done in this thesis, without a more comprehensive experimental investigation, we cannot simply claim that this identification can always be achieved in other applications. Nevertheless, even if the selected MRs are not effective, our approach is still of significant impact because it provides a solution for programs without test oracle, which cannot be dealt with by the conventional SBFL.

3.6.2 Internal validity

The primary threat to the internal validity involves the correctness of our experimental framework which includes generation of source and follow-up test cases, generation of mutants, execution of these mutants with both source and follow-up test cases, as well as examination of the outputs of source and follow-up test cases against the corresponding MR. Our experimental framework is quite different from the conventional SBFL using \( e_{slice} \). Thus the existing experimental framework from SIR cannot be adopted straightly. In order to assure the quality of our experimentation, we have conducted a very thorough unit testing at each implementation step and a comprehensive functional testing at the system level.
3.6.3 Construct validity

The primary threat to the construct validity is the use of the EXAM score as a metric for the effectiveness of an SBFL technique. Since this metric has been extensively used [Jones and Harrold, 2005; Liblit et al., 2005; Liu et al., 2006; Wong et al., 2010], therefore this threat is acceptably mitigated.

3.7 Related works

As discussed above, our $d_\text{mslice}$ and $e_\text{mslice}$ involve multiple test executions with respect to a particular program property (MR). Actually, this is not the first attempt to involve multiple executions in one slice. Canfora et al. [1998] and Harman et al. [2001] have proposed the concept of conditioned slice, which consists of a subset of program statements preserving the behaviour of the original program with respect to a set of program executions under a particular condition. The conditioned slice has been used in partition testing [Hierons et al., 2002] and program maintenance [Beszedes et al., 2002]. Different from metamorphic slice that involves multiple executions bound with an MR, the conditioned slice involves multiple executions bound with a condition that usually defines the constraints on the interested variables. Obviously, this condition-based concept does not reflect the program properties, and hence is not able to alleviate the oracle problem for SBFL.

In this chapter, we have demonstrated how to replace the traditional $e_\text{slice}$ with $e_\text{mslice}$ in SBFL. Actually, there are other types of spectra used in fault localization. One example is the predicate-count spectrum [Liblit et al., 2005; Liu et al., 2006]. These techniques instrumented and investigated predicates at some critical positions of the program. After evaluating and ranking the risk values for these predicates, these techniques highlighted the suspicious predicates that indicate the faults. Another popular spectrum is the method-calling-sequence spectrum [Dallmeier et al., 2005; Liu et al., 2005]. Instead of recording the program execution in the statement level, these techniques consider the method level. Obviously, they are more appealing to large-scaled programs and object-oriented programs. However, no matter what spectrum is adopted, the existence of a test oracle is always assumed in these techniques. In order to alleviate the oracle problem, we can integrate them with metamorphic testing, similar to what we have done in this chapter.
4

A Theoretical analysis of the risk evaluation formulas in SBFL

4.1 Motivation

It is well-known that one of the most essential tasks in SBFL is the risk evaluation. An effective risk evaluation formula is very crucial to provide a good fault-localization performance for SBFL. With more and more formulas proposed, some people started to compare their performance, in order to identify the formulas with the “best” performance [Jones and Harrold, 2005; Abreu et al., 2006, 2007, 2009a]. In all these studies, empirical approaches were conducted to investigate and measure the effectiveness of the risk evaluation formulas. In order to make the experimental results more reliable, people have used various approaches to control the threats to validity. For example, they adopted the same performance metric or its equivalents, the standardized experimental set-up and the unified benchmarks. In addition, both the mutation analysis and real-life case studies were conducted.

When a theoretical analysis is intractable, an empirical approach is usually the only effective way to conduct an investigation. However, the limitations of the empirical approach shall not be ignored. In an experimental analysis, the performance of a risk evaluation formula strongly depends on the experimental set-up. Different combinations of various test suites, testing objects, faults types, etc., may affect the experimental results. Even though people have adopted the unified set-up and benchmarks, these empirical studies can hardly be considered as sufficiently comprehensive due to the huge number of combinations of all the possible variations. In other words, the experimental results are still the sampled observations and cannot conclusively identify the most effective formulas.

Therefore, some researchers have investigated the performance of risk evaluation formulas from a theoretical perspective. As the first attempt, Lee et al. [2009a] have proved that formula Tarantula always produces identical ranking list as formula $q_e$, and hence they are equivalent. This preliminary study was followed by a more comprehensive investigation [Naish et al., 2011], where over 30
4.2. OUR FRAMEWORK

Formulas were studied and more equivalence relations were identified, using the same definition of equivalence as Lee et al. [2009a]. Naish et al. [2011] proposed a model program to simulate a single-fault program, and the average performance over all possible multisets of execution paths was used to measure the performance of a formula. Based on this model program and performance metric, two optimal risk evaluation formulas were proposed and proved.

However, Lee et al. [2009a] and Naish et al. [2011] have their limitations. First, they both adopted the most strict type of equivalence, which requires that two equivalent formulas produce identical ranking list. Intuitively speaking, two formulas should be regarded as equivalent as long as, for any faulty program and test suite, the rankings of the same faulty statement in their final ranking lists are identical. Obviously, identical ranking for all statements is a sufficient condition but not a necessary condition to have the same ranking for the faulty statement. Therefore, some formulas which are intuitively equivalent would be regarded as non-equivalent according to the type of equivalence used by Lee et al. [2009a] and Naish et al. [2011]. Hence, their equivalence does not properly reflect the realistic scenarios. Secondly, though Naish et al. [2011] have provided the optimal formulas for their model and performance metric, their optimal formulas may not be optimal with respect to the most popular performance metric of EXAM score in the SBFL community.

In summary, none of the previous studies has actually provided a conclusive answer that which formula is the most effective one with respect to the metric of EXAM score, and hence should be used when SBFL is applied. Therefore, in this chapter, we will provide an innovative theoretical framework to conduct the effectiveness comparison of various risk evaluation formulas. By using our framework, we are able to identify the maximal formulas among our investigated formulas.

4.2 Our framework

As discussed in Section 2.1, in SBFL, given a program and a test suite, the matrix $MA$ can be constructed accordingly. A risk evaluation formula $R$ uses $MA$ to assess the risk of being faulty for all statements, according to which, all statements will be sorted descendingly. Such a ranking list is then used to assist debugging. Therefore, the relative risk values rather than the absolute risk values of all statements, is the key factor determining the EXAM score for a formula $R$.

Given a ranking list in descending order of the risk values evaluated by a formula $R$, we can divide all statements into three disjoint sets, $S^R_B$, $S^R_F$ and $S^R_A$ with respect to an arbitrary $s_f$, as follows.

**Definition 4.2.1.** Given a program with $n$ statements $PG=<s_1, s_2, ..., s_n>$, a test suite of $m$ test cases $TS={t_1, t_2, ..., t_m}$, and a risk evaluation formula $R$, vector $A_i=<a^i_{ef}, a^i_{ep}, a^i_{np}, a^i_{nf}>$ can be constructed for each statement $s_i$, and $R(s_i)$ can be computed accordingly. For any faulty statement $s_f$, the set of program statements $S={s_1, s_2, ..., s_n}$ can be decomposed into three mutually exclusive subsets:
4.2. OUR FRAMEWORK

(a) $S^R_B$ consists of all statements with risk values higher than the risk value of the faulty statement $s_f$, that is, $S^R_B = \{ s_i \in S | R(s_i) > R(s_f), 1 \leq i \leq n \}$. 

(b) $S^R_F$ consists of all statements with the risk values equal to the risk value of the faulty statement $s_f$, that is, $S^R_F = \{ s_i \in S | R(s_i) = R(s_f), 1 \leq i \leq n \}$. 

(c) $S^R_A$ consists of all statements with the risk values lower than the risk value of the faulty statement $s_f$, that is, $S^R_A = \{ s_i \in S | R(s_i) < R(s_f), 1 \leq i \leq n \}$. 

In the practice of SBFL, a tie-breaking scheme is always required to determine the order of the statements with same risk values, and this scheme may affect the performance of SBFL. Different tie-breaking schemes have been developed, including WORST, BEST, ORIGINAL ORDER, etc. [Wong et al., 2008, 2010; Xie et al., 2011b]. However, in terms of evaluating the EXAM score, there is no need to consider the application or impact of tie-breaking scheme on $S^R_B$ or $S^R_A$, because by definition, in the final list returned by a risk evaluation formula $R$, all $s_i \in S^R_B$ are ranked higher than $s_f$; while all $s_i \in S^R_A$ are ranked lower than $s_f$. Thus, the ordering of the statements within $S^R_B$ or $S^R_A$ does not affect the ranking of $s_f$. As a consequence, we are interested in how a tie-breaking scheme distinguishes and ranks $s_i \in S^R_F$.

However, as a theoretical analysis, our framework cannot support any arbitrary tie-breaking scheme, since some of them might be unreasonable and hence make the performance comparison intractable. Actually, a tie-breaking scheme solves the ordering problem that a risk evaluation formula cannot handle. Thus, when focusing on the comparison between different formulas, it is reasonable to expect that the tie-breaking scheme returns consistent results for all formulas. Thus, in the comparison, we require that a tie-breaking scheme preserves the relative order of any pair of statements irrespective of which formula is used. We refer such schemes as consistent tie-breaking schemes, which are defined as follows.

Definition 4.2.2. Given any two statement sets $S_1$ and $S_2$, which contain elements with the same risk values. A tie-breaking scheme returns the ordered statement lists $O_1$ and $O_2$ for $S_1$ and $S_2$, respectively. The tie-breaking scheme is said to be consistent, if all elements common to $S_1$ and $S_2$ have the same relative order in $O_1$ and $O_2$.

Let us use an simple example to further illustrate the intuition behind this requirement. Given two risk evaluation formulas $R_1$ and $R_2$ that return the same $S^R_F$ but different $S^R_B$. Suppose the size of $S^R_{B1}$ is smaller than the size of $S^R_{B2}$. Since the order of $s_i \in S^R_F$ in both $R_1$ and $R_2$ cannot be decided by these two formulas, that is, it is independent of these formulas, then to make a fair comparison, this order must be identical in $R_1$ and $R_2$. Obviously, only by adopting a consistent tie-breaking scheme can guarantee such identical order. And in this example, $R_1$ with smaller $S^R_B$ would have a lower EXAM score.

Intuitively speaking, the most straightforward approach towards the theoretical analysis of the
performance between two formulas is to compare the sizes of their $S^R_B$ and the numbers of statements that are from $S^F_R$ but ranked before $s_f$ based on the tie-breaking scheme. However, since the sizes of $S^R_B$ and $S^F_R$ depend on the program and test suite, which can be very varying, a size comparison appears to be intractable. One of the innovative contributions in our study is to make use of the subset relationships among $S^R_B$ (or $S^F_B$) of different formulas, to facilitate the analysis. It turns out that the use of the notion of subset is sufficiently strong to identify the maximal risk evaluation formulas.

Let $E_1$ and $E_2$ denote the EXAM scores for risk evaluation formulas $R_1$ and $R_2$, respectively. We define two types of relations between $R_1$ and $R_2$ as follows.

**Definition 4.2.3** (Better). $R_1$ is said to be better than $R_2$ (denoted as $R_1 \rightarrow R_2$) if for any program, faulty statement $s_f$, test suite and consistent tie-breaking scheme, we have $E_1 \leq E_2$.

Obviously the relation “$\rightarrow$” is reflexive, that is, we have $R_1 \rightarrow R_1$. Furthermore, this relation is transitive, that is, if $R_1 \rightarrow R_2$ and $R_2 \rightarrow R_3$, we have $R_1 \rightarrow R_3$.

**Definition 4.2.4** (Equivalent). $R_1$ and $R_2$ are said to be equivalent (denoted as $R_1 \leftrightarrow R_2$), if for any program, faulty statement $s_f$, test suite and consistent tie-breaking scheme, we have $E_1 = E_2$.

Our definition of equivalence is more general and intuitively appealing than the definition of equivalence used by Lee et al. [2009a] and Naish et al. [2011]. Following our Definition 4.2.4, two formulas are equivalent if and only if they have the same number of statements preceding the faulty statement in the ranking lists. In [Lee et al., 2009a] and [Naish et al., 2011], two formulas are equivalent if they have identical ranking list, which however is only a sufficient condition but not a necessary condition for having the same EXAM score. Thus, their definition is a special case of ours.

As a reminder, this relation “$\leftrightarrow$” is reflexive, symmetric and transitive, that is, $R_1 \leftrightarrow R_1$; if $R_1 \leftrightarrow R_2$, then $R_2 \leftrightarrow R_1$; and if $R_1 \leftrightarrow R_2$ and $R_2 \leftrightarrow R_3$, then $R_1 \leftrightarrow R_3$.

Immediately from Definition 4.2.3 and Definition 4.2.4, we have the following property.

**Theorem 4.2.1.** For any two risk evaluation formulas $R_1$ and $R_2$, $R_1 \leftrightarrow R_2$ if and only if $R_1 \rightarrow R_2$ and $R_2 \rightarrow R_1$.

Also, a sufficient condition of $R_1 \rightarrow R_2$ involving the notion of subset is as follows.

**Theorem 4.2.2.** Given any two risk evaluation formulas $R_1$ and $R_2$, if for any program, faulty statement $s_f$ and test suite, we have $S^R_{R_1} \subseteq S^R_{R_2}$ and $S^R_{A_2} \subseteq S^R_{A_1}$, then $R_1 \rightarrow R_2$.

**Proof.** Consider a virtual formula $R_3$, such that for any program, $s_f$ and test suite, $S^R_B = S^R_{R_1}$ and $S^R_A = S^R_{A_1}$. Let $E_3$ denote the EXAM score of $R_3$, and let $L_1$, $L_2$ and $L_3$ denote the ranking lists returned by $R_1$, $R_2$ and $R_3$, respectively. Obviously, considering $R_1$ and $R_3$, we have $S^R_B = S^R_{R_1}$, $S^R_{R_1} \subseteq S^R_{A_1}$ and $S^R_{A_3} \subseteq S^R_{A_1}$. If the tie-breaking scheme is consistent, $s_f$ can never have lower ranking in $L_1$ than in $L_3$. Therefore, we have $E_1 \leq E_3$. Now, considering $R_2$ and $R_3$, we have $S^R_{R_3} \subseteq S^R_{R_2}$,
4.3 Effectiveness comparison of risk evaluation formulas

In this section, we will investigate some popular formulas and compare their performance using the notion of subset.

4.3.1 Assumptions

Before presenting the performance comparison using our theoretical framework, we describe and discuss our assumptions.

1. We assume that the SBFL techniques are applied to programs with testing oracle. In other words, for any test case, the testing result of either fail or pass, can be decided. This assumption is adopted in all previous studies, except our recent work [Xie et al., 2011b].

2. We have the assumption of “perfect bug detection”, which is adopted by most of the previous SBFL studies. It assumes that the fault can always be identified once the faulty statement is examined [Wong et al., 2010].

3. We assume that the faults are the deterministic faults, that is, a test case will always yield the same testing result of either failed or passed. This type of faults is not affected by any run-time environment, and is also assumed in the majority of previous SBFL studies. Moreover, we will exclude the omission faults, because SBFL is designed to assign risk values to the existent statements.
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Some previous SBFL experimental studies handled the omission faults by considering the preceding or succeeding statement of the missing statement as the “faulty statement”. However, this approach is not completely satisfactory because it could lead to controversy or inconsistency. Furthermore, the “preceding or succeeding” statement may have different interpretations, such as “the line order of source code” or “the order according to the control-flow graph”. Not all the experimental studies have explicitly clarified their methods of identifying these faults. Thus in order to avoid unnecessary noises, we do not consider the omission faults in this study.

4. The test suite is assumed to have 100% statement coverage, that is, for any \( s_i \), we have \( a_{ef}^i + a_{ep}^i > 0 \). Also assumed is that the test suite contains at least one passed test case and one failed test case, that is, for any \( s_i \), we have \( a_{ep}^i + a_{np}^i > 0 \) and \( a_{ef}^i + a_{nf}^i > 0 \). Intuitively speaking, these assumptions are reasonable because even though we can never justify that a test suite has provided “sufficient” testing information for fault localization, we can at least argue that a test suite with some uncovered statements, or with either solely passed test cases or solely failed test cases, is not sufficient for debugging. More importantly, these assumptions are required to make some formulas (such as Tarantula) totally defined.

4.3.2 Investigated formulas

In this thesis, we investigate 30 risk evaluation formulas, which are selected from [Naish et al., 2011], because their study is the most comprehensive theoretical investigation. These formulas are listed in Table 4.1. Some of their formulas are excluded in our investigation, because they require specific constraints to make them totally defined, which however are not intuitively justified in the context of SBFL. For example, formula M1 in [Naish et al., 2011] is defined as \( \frac{a_{ef} + a_{np}}{a_{ef} + a_{np}} \). In M1, all statements with \( a_{nf} + a_{ep} = 0 \) would then have undefined risk values. However, there is no intuition to justify why we need to have the constraint of \( a_{nf} + a_{ep} \neq 0 \).

Besides, since some formulas are not originally designed for SBFL, they may require modifications prior to their applications in SBFL. For example, the original form of formula AMPLE2 defined in Table 4.1 is \( \left| \frac{a_{ef}}{a_{nf} + a_{ef}} - \frac{a_{ep}}{a_{np} + a_{ep}} \right| \), which was originally proposed to identify faulty classes in object-oriented software, with the assumption that there is exactly one failing run [Dullmeier et al., 2005]. Since this original form always returns an absolute value, the magnitude order of the computed signed values may be changed. Such a change would violate against the intuition of risk evaluation in the context of SBFL that statements associated with more failed and less passed testing results should have higher faulty risks. Therefore, when applying this formula in SBFL, we follow Naish et al. [2011] to use its variant defined in Table 4.1.
### 4.3. Effectiveness Comparison of Risk Evaluation Formulas

#### Table 4.1: Investigated Formulas

<table>
<thead>
<tr>
<th>Name</th>
<th>Expression</th>
</tr>
</thead>
</table>
| Op1        | \[
|            | \begin{cases}
|            | -1 & \text{if } a_{ef} < F \\
|            | P - a_{ep} & \text{if } a_{ef} = F \\
|            | \end{cases}                                                                 |
| Op2        | \[ a_{ef} - \frac{a_{ep}}{a_{ep} + a_{np} + 1} \]                           |
| Jaccard    | \[ \frac{a_{ef}}{a_{ef} + a_{nf} + a_{ep}} \]                              |
| Anderberg  | \[ \frac{a_{ef}}{a_{ef} + 2(a_{nf} + a_{ep})} \]                           |
| Sørensen-Dice | \[ \frac{2a_{ef}}{2a_{ef} + a_{nf} + a_{ep}} \]                           |
| Dice       | \[ \frac{2a_{ef}}{a_{ef} + a_{nf} + a_{ep}} \]                              |
| Goodman    | \[ \frac{2a_{ef} - a_{nf} - a_{np}}{2a_{ef} + a_{nf} + a_{np}} \]           |
| Tarantula  | \[ \frac{a_{ef}}{a_{ef} + a_{nf}} \left( \frac{a_{ef}}{a_{ef} + a_{nf}} + \frac{a_{ep}}{a_{ep} + a_{np}} \right) \] |
| qe         | \[ \frac{a_{ef}}{a_{ef} + a_{ep}} \]                                       |
| CBI Inc.   | \[ \frac{a_{ef}}{a_{ef} + a_{ep}} - \frac{a_{ef} + a_{nf}}{a_{ef} + a_{nf} + a_{ep} + a_{np}} \] |
| Wong2      | \[ a_{ef} - a_{cp} \]                                                       |
| Hamann     | \[ \frac{a_{ef} + a_{np} - a_{nf} - a_{ep}}{a_{ef} + a_{nf} + a_{ep} + a_{np}} \] |
| Simple Matching | \[ \frac{a_{ef} + a_{np}}{a_{ef} + a_{np} + a_{ep} + a_{np}} \]            |
| Sokal      | \[ \frac{2(a_{ef} + a_{np})}{2(a_{ef} + a_{np}) + a_{nf} + a_{ep}} \]       |
| Rogers&Tanimoto | \[ \frac{a_{ef} + a_{np}}{a_{ef} + a_{np} + 2(a_{nf} + a_{ep})} \]         |
### 4.3. Effectiveness Comparison of Risk Evaluation Formulas

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hamming etc. (Including Hamming, Manhattan, Lee metrics)</td>
<td>$a_{ef} + a_{np}$</td>
</tr>
<tr>
<td>Euclid</td>
<td>$\sqrt{a_{ef} + a_{np}}$</td>
</tr>
<tr>
<td>Wong1</td>
<td>$a_{ef}$</td>
</tr>
<tr>
<td>Russel &amp; Rao</td>
<td>$\frac{a_{ef}}{a_{ef} + a_{nf} + a_{ep} + a_{np}}$</td>
</tr>
<tr>
<td>Binary</td>
<td>$\begin{cases} 0 &amp; \text{if } a_{ef} &lt; F \ 1 &amp; \text{if } a_{ef} = F \end{cases}$</td>
</tr>
<tr>
<td>Scott</td>
<td>$\frac{4a_{ef}a_{np} - 4a_{ef}a_{np} - (a_{nf} - a_{ep})^2}{2(a_{ef} + a_{nf} + a_{ep})(2a_{np} + a_{nf} + a_{ep})}$</td>
</tr>
<tr>
<td>Rogot1</td>
<td>$\frac{1}{2} \left( \frac{a_{ef}}{a_{ef} + a_{np} + a_{nf} + a_{ep}} + \frac{a_{np}}{2a_{np} + a_{ef} + a_{np}} \right)$</td>
</tr>
<tr>
<td>Kulczynski2</td>
<td>$\frac{1}{2} \left( \frac{a_{ef}}{a_{ef} + a_{np}} + \frac{a_{np}}{a_{ef} + a_{np}} \right)$</td>
</tr>
<tr>
<td>M2</td>
<td>$\frac{a_{ef}}{a_{ef} + a_{np} + 2(a_{nf} + a_{ep})}$</td>
</tr>
<tr>
<td>Ochiai</td>
<td>$\frac{a_{ef}}{\sqrt{(a_{ef} + a_{nf})(a_{ef} + a_{np})}}$</td>
</tr>
<tr>
<td>AMPLE2</td>
<td>$\frac{a_{ef}}{a_{ef} + a_{nf} - \frac{a_{np}}{a_{ef} + a_{np}}}$</td>
</tr>
</tbody>
</table>
### 4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Table 4.1: Investigated formulas (cont.)

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wong3</td>
<td>( a_{ef} - h, ) where ( h = \begin{cases} a_{ep} &amp; \text{if } a_{ep} \leq 2 \ 2 + 0.1(a_{ep} - 2) &amp; \text{if } 2 &lt; a_{ep} \leq 10 \ 2.8 + 0.001(a_{ep} - 10) &amp; \text{if } a_{ep} &gt; 10 \end{cases} )</td>
</tr>
<tr>
<td>Arithmetic Mean</td>
<td>( \frac{2a_{ef}a_{np} - 2a_{ef}a_{ep}}{(a_{ef} + a_{ep})(a_{np} + a_{ef}) + (a_{ef} + a_{np})(a_{ef} + a_{ep})} )</td>
</tr>
<tr>
<td>Cohen</td>
<td>( \frac{2a_{ef}a_{np} - 2a_{ef}a_{ep}}{(a_{ef} + a_{ep})(a_{np} + a_{ep}) + (a_{ef} + a_{np})(a_{ef} + a_{ep})} )</td>
</tr>
<tr>
<td>Fleiss</td>
<td>( \frac{4a_{ef}a_{np} - 4a_{ef}a_{ep} - (a_{ef} - a_{ep})^2}{(2a_{ef} + a_{ef} + a_{ep})(a_{ef} + a_{np} + a_{ep})} )</td>
</tr>
</tbody>
</table>

### 4.3.3 \( S^R_B, S^R_F \) and \( S^R_A \) for all formulas

In this section, we will construct the \( S^R_B, S^R_F \) and \( S^R_A \) for all the investigated formulas, to facilitate our following analysis. Before constructing the relevant sets, we would like to define some notations and give some lemmas.

Given a test suite \( TS \), we denote its size as \( T \), the number of failed test cases as \( F \) and the number of passed cases as \( P \). Obviously, we have \( 1 \leq F < T, 1 \leq P < T, \) and \( P + F = T \). And we have the following lemmas of which the proofs are immediately after the definitions and the above assumptions.

**Lemma 4.3.1.** For any \( A_i = \langle a_{ef}^i, a_{ep}^i, a_{nf}^i, a_{np}^i \rangle \), we have \( a_{ef}^i + a_{ep}^i > 0, a_{ef}^i + a_{nf}^i = F, a_{ep}^i + a_{np}^i = P, a_{ef}^i \leq F \) and \( a_{ep}^i \leq P \).

**Lemma 4.3.2.** For any faulty statement \( s_f \) with \( A_f = \langle a_{ef}^f, a_{ep}^f, a_{nf}^f, a_{np}^f \rangle \), if \( s_f \) is the only faulty statement in the program, we have \( a_{ef}^f = F \) and \( a_{np}^f = 0 \).

With the above lemmas, we will analyze the formulas in Table 4.1 one by one.
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

1. Op1

As stated in Table 4.1, formula Op1 is defined as follows.

\[ R_{Op1}(s_i) = \begin{cases} 
-1 & \text{if } a_{ef}^i < F \\
 P - a_{ep}^i & \text{if } a_{ef}^i = F 
\end{cases} \]

After Definition 4.2.1, we have

\[
S_B^{Op1} = \{ s_i | (a_{ef}^i < F \text{ and } -1 > P - a_{ep}^i) \text{ or } (a_{ef}^i = F \text{ and } P - a_{ep}^i > P - a_{ep}^i), 1 \leq i \leq n \} \tag{4.3.1} \\
S_F^{Op1} = \{ s_i | (a_{ef}^i < F \text{ and } -1 = P - a_{ep}^i) \text{ or } (a_{ef}^i = F \text{ and } P - a_{ep}^i = P - a_{ep}^i), 1 \leq i \leq n \} \tag{4.3.2} \\
S_A^{Op1} = \{ s_i | (a_{ef}^i < F \text{ and } -1 < P - a_{ep}^i) \text{ or } (a_{ef}^i = F \text{ and } P - a_{ep}^i < P - a_{ep}^i), 1 \leq i \leq n \} \tag{4.3.3} \\
\]

We are going to prove that the above sets \( S_B^{Op1}, S_F^{Op1} \) and \( S_A^{Op1} \) are equal to the following sets \( X^1, Y^1 \) and \( Z^1 \), respectively.

\[
X^1 = \{ s_i | (a_{ef}^i = F \text{ and } a_{ep}^i - a_{ep}^i > 0), 1 \leq i \leq n \} \tag{4.3.4} \\
Y^1 = \{ s_i | (a_{ef}^i = F \text{ and } a_{ep}^i - a_{ep}^i = 0), 1 \leq i \leq n \} \tag{4.3.5} \\
Z^1 = \{ s_i | (a_{ef}^i < F) \text{ or } (a_{ef}^i = F \text{ and } a_{ep}^i - a_{ep}^i < 0), 1 \leq i \leq n \} \tag{4.3.6} \\
\]

First, we will prove \( S_B^{Op1} = X^1 \). \( S_B^{Op1} \) defined in (4.3.1) can be re-written as:

\[
S_B^{Op1} = \{ s_i | a_{ef}^i < F \text{ and } -1 > P - a_{ep}^i, 1 \leq i \leq n \} \cup \{ s_i | a_{ef}^i = F \text{ and } a_{ep}^i - a_{ep}^i > 0, 1 \leq i \leq n \} \\
\]

Since \((-1 < P - a_{ep}^i)\) after Lemma 4.3.1, we have

\[
\{ s_i | a_{ef}^i < F \text{ and } -1 > P - a_{ep}^i, 1 \leq i \leq n \} = \emptyset \\
\]

Therefore, \( S_B^{Op1} \) becomes

\[
S_B^{Op1} = \{ s_i | a_{ef}^i = F \text{ and } a_{ep}^i - a_{ep}^i > 0, 1 \leq i \leq n \} = X^1 \\
\]

Similarly, we can prove that \( S_F^{Op1} = Y^1 \).

Now, consider \( S_A^{Op1} \) defined in (4.3.3). Since \((-1 < P - a_{ep}^i)\) after Lemma 4.3.1,
(\(a_{ef}\leq F\) and \(-1<P-a_{ep}\)) is logically equivalent to \((a_{ef}\leq F)\). Therefore, \(S_{A}^{Op1}\) becomes

\[
S_{A}^{Op1} = \{s_i | (a_{ef}\leq F) \text{ or } (a_{ef} = F \text{ and } a_{ep} - a_{ip} < 0), 1 \leq i \leq n\} = \mathbb{Z}^1
\]

In conclusion, we have proved that \(S_{B}^{Op1} = X^1, S_{F}^{Op1} = Y^1\) and \(S_{A}^{Op1} = Z^1\).

2. **Op2**

As stated in Table 4.1, formula Op2 is defined as follows.

\[
R_{Op2}(s_i) = a_{ef}^i - \frac{a_{ip}}{a_{ep} + a_{ip} + 1}
\]

It follows from Lemmas 4.3.1 and 4.3.2 that \(R_{Op2}(s_i) = a_{ef}^i \frac{a_{ep}}{P+1}\) and \(R_{Op2}(s_i) = F - a_{ef}^i \frac{a_{ip}}{P+1}\). Then, after Definition 4.2.1, we have

\[
\begin{align*}
S_{B}^{Op2} &= \{s_i | a_{ef}^i \frac{a_{ip}}{P+1} > F - \frac{a_{ip}}{P+1}, 1 \leq i \leq n\} \quad (4.3.7) \\
S_{F}^{Op2} &= \{s_i | a_{ef}^i \frac{a_{ip}}{P+1} = F - \frac{a_{ip}}{P+1}, 1 \leq i \leq n\} \quad (4.3.8) \\
S_{A}^{Op2} &= \{s_i | a_{ef}^i \frac{a_{ip}}{P+1} < F - \frac{a_{ip}}{P+1}, 1 \leq i \leq n\} \quad (4.3.9)
\end{align*}
\]

We are going to prove that \(S_{B}^{Op2} = X^1\). For any \(s_i\), we have either \((a_{ef}^i \leq F)\) or \((a_{ef}^i = F)\). Therefore, \(S_{B}^{Op2}\) defined in (4.3.7) can be re-written as

\[
S_{B}^{Op2} = \{s_i | a_{ef}^i \leq F \text{ and } a_{ef}^i \frac{a_{ip}}{P+1} > F - \frac{a_{ip}}{P+1}, 1 \leq i \leq n\}
\]

\[
\cup \{s_i | a_{ef}^i = F \text{ and } a_{ef}^i \frac{a_{ip}}{P+1} > F - \frac{a_{ip}}{P+1}, 1 \leq i \leq n\}
\]

Consider the case that \(a_{ef}^i \leq F\). Then, we have \(F - a_{ef}^i \geq 1\). Since \(a_{ip} - a_{ep} \leq P\) after Lemma 4.3.1, we have \(a_{ef}^i \frac{a_{ip}}{P+1} < 1\). Thus, \((a_{ef}^i - a_{ip} \frac{a_{ip}}{P+1}) - (F - a_{ef}^i \frac{a_{ip}}{P+1}) = a_{ip} - a_{ip} \frac{a_{ip}}{P+1} - (F - a_{ef}^i) < 0\). Therefore, we have \(a_{ef}^i \frac{a_{ip}}{P+1} < F - a_{ef}^i \frac{a_{ip}}{P+1}\), which is contradictory to \(a_{ef}^i \frac{a_{ip}}{P+1} > F - a_{ef}^i \frac{a_{ip}}{P+1}\). Thus,

\[
\{s_i | a_{ef}^i < F \text{ and } a_{ef}^i \frac{a_{ip}}{P+1} > F - \frac{a_{ip}}{P+1}, 1 \leq i \leq n\} = \emptyset
\]
Hence, we have

\[ S_{B}^{Op2} = \{ s_i | a_{ef}^i = F \text{ and } a_{ef}^i - \frac{a_{ep}^i}{P+1} > F - \frac{a_{ep}^f}{P+1}, 1 \leq i \leq n \} \]  
(4.3.10)

- Assume that \( s_i \in S_{B}^{Op2} \). After (4.3.10), we have \( (a_{ef}^i = F \text{ and } a_{ef}^i - \frac{a_{ep}^i}{P+1} > F - \frac{a_{ep}^f}{P+1}) \). Since \( a_{ef}^i = F \), \( a_{ef}^i - \frac{a_{ep}^i}{P+1} > F - \frac{a_{ep}^f}{P+1} \) becomes \( a_{ep}^i - a_{ep}^f > 0 \). Since \( P+1 > 0 \), then \( a_{ep}^f - a_{ep}^i > 0 \) implies \( a_{ep}^i - a_{ep}^f > 0 \). Thus, \( s_i \in X^1 \) after (4.3.4). Therefore, \( S_{B}^{Op2} \subseteq X^1 \).

- Assume that \( s_i \in X^1 \). After (4.3.4), we have \( (a_{ef}^i = F \text{ and } a_{ep}^f - a_{ep}^i > 0) \). Since \( a_{ef}^i = F \), we have \( a_{ep}^f = (a_{ep}^i - a_{ep}^f) - (P+1)(F - a_{ef}^i) > 0 \). Since \( P+1 > 0 \), we have \( (a_{ef}^i - a_{ep}^f) - (P+1)(F - a_{ef}^i) > 0 \) implies \( a_{ef}^i - \frac{a_{ep}^i}{P+1} > F - \frac{a_{ep}^f}{P+1} \). Thus, \( s_i \in S_{B}^{Op2} \) after (4.3.10). Therefore, \( X^1 \subseteq S_{B}^{Op2} \).

In summary, we have proved that \( S_{B}^{Op2} = X^1 \).

Similarly, we can prove that \( S_{F}^{Op2} = Y^1 \).

Next, we are going to prove \( S_{A}^{Op2} = Z^1 \). \( S_{A}^{Op2} \) defined in (4.3.9) can be re-written as

\[ S_{A}^{Op2} = \{ s_i | a_{ef}^i < F \text{ and } a_{ef}^i - \frac{a_{ep}^i}{P+1} < F - \frac{a_{ep}^f}{P+1}, 1 \leq i \leq n \} \]

\[ \cup \{ s_i | a_{ef}^i = F \text{ and } a_{ef}^i - \frac{a_{ep}^i}{P+1} < F - \frac{a_{ep}^f}{P+1}, 1 \leq i \leq n \} \]

Consider the case that \( a_{ef}^i < F \). As shown in the above proof of \( S_{B}^{Op2} = X^1 \), \( a_{ef}^i < F \) implies \( a_{ef}^i - \frac{a_{ep}^i}{P+1} < F - \frac{a_{ep}^f}{P+1} \). Thus, \( (a_{ef}^i < F \text{ and } a_{ef}^i - \frac{a_{ep}^i}{P+1} < F - \frac{a_{ep}^f}{P+1}) \) is logically equivalent to \( a_{ef}^i < F \). Therefore, \( S_{A}^{Op2} \) becomes

\[ \{ s_i | a_{ef}^i < F, 1 \leq i \leq n \} \cup \{ s_i | a_{ef}^i = F \text{ and } a_{ef}^i - \frac{a_{ep}^i}{P+1} < F - \frac{a_{ep}^f}{P+1}, 1 \leq i \leq n \} \]

Similar to the proof of \( S_{B}^{Op2} = X^1 \), we can prove

\[ \{ s_i | a_{ef}^i = F \text{ and } a_{ef}^i - \frac{a_{ep}^i}{P+1} < F - \frac{a_{ep}^f}{P+1}, 1 \leq i \leq n \} = \{ s_i | a_{ef}^i = F \text{ and } a_{ep}^f - a_{ep}^i < 0, 1 \leq i \leq n \} \]

Therefore,

\[ S_{A}^{Op2} = \{ s_i | (a_{ef}^i < F) \text{ or } (a_{ef}^i = F \text{ and } a_{ep}^f - a_{ep}^i < 0), 1 \leq i \leq n \} = Z^1 \]

In conclusion, we have proved that \( S_{B}^{Op2} = X^1 \), \( S_{F}^{Op2} = Y^1 \) and \( S_{A}^{Op2} = Z^1 \).
3. Jaccard

As stated in Table 4.1, formula Jaccard is defined as follows.

\[ R_J(s_i) = \frac{a_{ef}^i}{a_{ef}^i + a_{nf}^i + a_{ep}^i} \]

It follows from Lemmas 4.3.1 and 4.3.2 that \( R_J(s_i) = \frac{a_{ef}^i}{F + a_{ep}^i} \) and \( R_J(s_f) = \frac{F}{F + a_{ep}^i} \). Then, after Definition 4.2.1, we have

\[
S_{JB}^J = \{ s_i | \frac{a_{ef}^i}{F + a_{ep}^i} > \frac{F}{F + a_{ep}^i}, 1 \leq i \leq n \} \quad (4.3.11)
\]

\[
S_{BF}^J = \{ s_i | \frac{a_{ef}^i}{F + a_{ep}^i} = \frac{F}{F + a_{ep}^i}, 1 \leq i \leq n \} \quad (4.3.12)
\]

\[
S_{JA}^J = \{ s_i | \frac{a_{ef}^i}{F + a_{ep}^i} < \frac{F}{F + a_{ep}^i}, 1 \leq i \leq n \} \quad (4.3.13)
\]

We are going to prove that the above sets \( S_{JB}^J \), \( S_{BF}^J \) and \( S_{JA}^J \) are equal to the following sets \( X^2 \), \( Y^2 \) and \( Z^2 \), respectively.

\[
X^2 = \{ s_i | a_{ef}^i > 0 \text{ and } 1 + \frac{a_{ep}^i}{F} - \frac{a_{ef}^i}{a_{ef}^i} > 0, 1 \leq i \leq n \} \quad (4.3.14)
\]

\[
Y^2 = \{ s_i | a_{ef}^i > 0 \text{ and } 1 + \frac{a_{ep}^i}{F} - \frac{a_{ef}^i}{a_{ef}^i} = 0, 1 \leq i \leq n \} \quad (4.3.15)
\]

\[
Z^2 = \{ s_i | (a_{ef}^i = 0) \text{ or } (a_{ef}^i > 0 \text{ and } 1 + \frac{a_{ep}^i}{F} - \frac{a_{ef}^i}{a_{ef}^i} < 0), 1 \leq i \leq n \} \quad (4.3.16)
\]

First, we will prove \( S_{JB}^J = X^2 \). For any \( s_i \), we have either \( a_{ef}^i = 0 \) or \( a_{ef}^i > 0 \). Therefore, \( S_{JB}^J \) in (4.3.11) can be re-written as

\[
S_{JB}^J = \{ s_i | a_{ef}^i = 0 \text{ and } \frac{a_{ef}^i}{F + a_{ep}^i} > \frac{F}{F + a_{ep}^i}, 1 \leq i \leq n \}
\]

\[
\cup \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i}{F + a_{ep}^i} > \frac{F}{F + a_{ep}^i}, 1 \leq i \leq n \}
\]

Consider the case that \( a_{ef}^i = 0 \). Since \( F > 0 \) and \( (F + a_{ep}^i) > 0 \) after Lemma 4.3.1, we have
\[
\frac{a^i_{ef}}{F+a^i_{ep}} = \frac{0}{F+a^i_{ep}} = 0 < \frac{F}{F+a^i_{ep}}, \text{ which is contradictory to } \frac{a^i_{ef}}{F+a^i_{ep}} \geq \frac{F}{F+a^i_{ep}}. \text{ Thus, }
\]

\[\{s_i | a^i_{ef} = 0 \text{ and } \frac{a^i_{ef}}{F+a^i_{ep}} > \frac{F}{F+a^i_{ep}}, 1 \leq i \leq n\} = \emptyset.\]

Hence, we have

\[S^I_B = \{s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{F+a^i_{ep}} > \frac{F}{F+a^i_{ep}}, 1 \leq i \leq n\}\]

(4.3.17)

- Assume that \(s_i \in S^I_B\). It follows from (4.3.17) that \((a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{F+a^i_{ep}} > \frac{F}{F+a^i_{ep}})\). Since \(a^i_{ef} > 0\), \(F > 0\), \((F+a^i_{ep}) > 0\) (after Lemma 4.3.1) and \((F+a^j_{ep}) > 0\) (after Lemma 4.3.1), \(\frac{a^i_{ef}}{F+a^i_{ep}} > \frac{F}{F+a^i_{ep}}\) implies \(\frac{F+a^i_{ep}}{a^i_{ef}} < \frac{F+a^i_{ep}}{F}\). After re-arranging the terms, we have \(1 + \frac{a^i_{ep}}{F} - \frac{F}{a^i_{ef}} - \frac{a^i_{ef}}{a^i_{ef}} > 0\). Thus, \(s_i \in X^2\) after (4.3.14). Therefore, \(S^I_B \subseteq X^2\).

- Assume that \(s_i \in X^2\). After (4.3.14), we have \((a^i_{ef} > 0 \text{ and } 1 + \frac{a^i_{ep}}{F} - \frac{F}{a^i_{ef}} - \frac{a^i_{ef}}{a^i_{ef}} > 0)\). After re-arranging the terms, \(1 + \frac{a^i_{ep}}{F} - \frac{F}{a^i_{ef}} - \frac{a^i_{ef}}{a^i_{ef}} > 0\) becomes \(\frac{F+a^i_{ep}}{a^i_{ef}} < \frac{F+a^i_{ep}}{F}\), which implies \(\frac{a^i_{ef}}{F+a^i_{ep}} > \frac{F}{F+a^i_{ep}}\) after \(a^i_{ef} > 0\), \(F > 0\), \((F+a^i_{ep}) > 0\) and \((F+a^j_{ep}) > 0\). It follows from (4.3.17) that \(s_i \in S^I_B\). Therefore, \(X^2 \subseteq S^I_B\).

In summary, we have proved that \(S^I_B = X^2\).

Similarly, we can prove that \(S^J_B = Y^2\).

Next, we are going to prove \(S^J_A = Z^2\). \(S^J_A\) in (4.3.13) can be re-written as

\[S^J_A = \{s_i | a^i_{ef} = 0 \text{ and } \frac{a^i_{ef}}{F+a^i_{ep}} < \frac{F}{F+a^i_{ep}}, 1 \leq i \leq n\}\]

\[\cup \{s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{F+a^i_{ep}} < \frac{F}{F+a^i_{ep}}, 1 \leq i \leq n\}\]

Consider the case \((a^i_{ef} = 0)\), which implies \(\frac{a^i_{ef}}{F+a^i_{ep}} = 0 < \frac{F}{F+a^i_{ep}}\). Thus, \((a^i_{ef} = 0 \text{ and } \frac{a^i_{ef}}{F+a^i_{ep}} < \frac{F}{F+a^i_{ep}})\) is logically equivalent to \((a^i_{ef} = 0)\). Therefore, \(S^J_A\) becomes

\[\{s_i | a^i_{ef} = 0, 1 \leq i \leq n\} \cup \{s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{F+a^i_{ep}} < \frac{F}{F+a^i_{ep}}, 1 \leq i \leq n\}\]
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Similar to the proof of $S_J^B=X^2$, we can prove

$$
\{s_i|a_{ef}^i>0 \text{ and } \frac{a_{ef}^i}{F+a_{ep}^i} < \frac{F}{F+a_{ep}^i}, 1 \leq i \leq n\}
$$

$$
=\{s_i|a_{ef}^i>0 \text{ and } 1+\frac{a_{ep}^i}{F-a_{ef}^i} \frac{F}{a_{ef}^i} - \frac{a_{ep}^i}{a_{ef}^i} < 0, 1 \leq i \leq n\}
$$

Therefore,

$$
S_J^B=\{s_i|(a_{ef}^i=0) \text{ or } (a_{ef}^i>0 \text{ and } 1+\frac{a_{ep}^i}{F-a_{ef}^i} \frac{F}{a_{ef}^i} - \frac{a_{ep}^i}{a_{ef}^i}<0), 1 \leq i \leq n\}=Z^2
$$

In conclusion, we have proved that $S_J^B=X^2$, $S_J^F=Y^2$ and $S_J^A=Z^2$.

4. Anderberg

As stated in Table 4.1, formula Anderberg is defined as

$$
R_{An}(s_i) = \frac{a_{ef}^i}{a_{ef}^i + 2(a_{ef}^i + a_{ep}^i)}
$$

It follows from Lemmas 4.3.1 and 4.3.2 that $R_{An}(s_i)=\frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^i}$ and $R_{An}(s_f)=\frac{F}{F+2a_{ep}^i}$. Then, after Definition 4.2.1, we have

$$
S_{BAn} = \{s_i|\frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^i} > \frac{F}{F+2a_{ep}^i}, 1 \leq i \leq n\} \quad (4.3.18)
$$

$$
S_{FAn} = \{s_i|\frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^i} = \frac{F}{F+2a_{ep}^i}, 1 \leq i \leq n\} \quad (4.3.19)
$$

$$
S_{AAn} = \{s_i|\frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^i} < \frac{F}{F+2a_{ep}^i}, 1 \leq i \leq n\} \quad (4.3.20)
$$

We are going to prove that $S_{BAn}$, $S_{FAn}$ and $S_{AAn}$ are equal to the above sets $X^2$ in (4.3.14), $Y^2$ in (4.3.15) and $Z^2$ in (4.3.16), respectively.

First, we will prove $S_{BAn}=X^2$. For any $s_i$, we have either $(a_{ef}^i=0)$ or $(a_{ef}^i>0)$. Therefore, $S_{BAn}$ in (4.3.18) can be re-written as
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Next, we are going to prove $S$. In summary, we have proved that $S$. Similarly, we can prove that $S$. Hence we have

Consider the case that $(a_{ef}^i=0)$. Since $F > 0$ and $(F+2a_{ep}^f) > 0$ after Lemma 4.3.1, we have

$$\frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^f} < 0 \frac{F}{F+2a_{ep}^f},$$

which is contradictory to $\frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^f} > \frac{F}{F+2a_{ep}^f}$. Thus,

$$\{ s_i | a_{ef}^i = 0 \text{ and } \frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^f} > \frac{F}{F+2a_{ep}^f}, 1 \leq i \leq n \} = \emptyset$$

Hence we have

$$S_B^{An} = \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^f} > \frac{F}{F+2a_{ep}^f}, 1 \leq i \leq n \} \quad (4.3.21)$$

- Assume that $s_i \in S_B^{An}$. After (4.3.21), we have $(a_{ef}^i > 0$ and $\frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^f} > \frac{F}{F+2a_{ep}^f}$). Since $a_{ef}^i > 0$, $F > 0$, $(2F-a_{ef}^i+2a_{ep}^f) > 0$ after Lemma 4.3.1 and $(F+2a_{ep}^f) > 0$ after Lemma 4.3.1, $\frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^f} > \frac{F}{F+2a_{ep}^f}$ implies $\frac{2F-a_{ef}^i+2a_{ep}^f}{a_{ef}^i} < \frac{F+2a_{ep}^f}{F}$. After re-arranging the terms, we have $1 + \frac{a_{ep}^f - F}{a_{ef}^i} = \frac{a_{ef}^i}{a_{ef}^i} > 0$. It follows from (4.3.14) that $s_i \in X^2$. Therefore, $S_B^{An} \subseteq X^2$.

- Assume that $s_i \in X^2$. After (4.3.14), we have $(a_{ef}^i > 0$ and $(a_{ef}^i)^{-1} > 0)$ and $(a_{ep}^f)^{-1} > 0)$. After re-arranging the terms, $1 + \frac{a_{ef}^i}{a_{ef}^i} - \frac{F}{a_{ef}^i} = \frac{a_{ef}^i}{a_{ef}^i} > 0$. After (after $a_{ef}^i > 0$, $F > 0$, $(2F-a_{ef}^i+2a_{ep}^f) > 0$ and $(F+2a_{ep}^f) > 0$. Then, we have $s_i \in S_B^{An}$ after (4.3.21). Therefore, $X^2 \subseteq S_B^{An}$.

In summary, we have proved that $S_B^{An} = X^2$.

Similarly, we can prove that $S_F^{An} = Y^2$.

Next, we are going to prove $S_A^{An} = Z^2$. $S_A^{An}$ defined in (4.3.20) can be re-written as follows.

$$S_A^{An} = \{ s_i | a_{ef}^i = 0 \text{ and } \frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^f} < \frac{F}{F+2a_{ep}^f}, 1 \leq i \leq n \}$$

$$\cup \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^f} < \frac{F}{F+2a_{ep}^f}, 1 \leq i \leq n \}$$
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Consider the case \( a_{ef}^i = 0 \), which implies \( \frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^i} = 0 < \frac{F}{F+2a_{ep}^i} \). Thus, \( (a_{ef}^i = 0 \) and \( \frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^i} < \frac{F}{F+2a_{ep}^i} \)) is logically equivalent to \( (a_{ef}^i = 0) \). Therefore, \( S_A^{An} \) becomes

\[
\{ s_i | a_{ef}^i = 0, 1 \leq i \leq n \} \cup \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^i} < \frac{F}{F+2a_{ep}^i}, 1 \leq i \leq n \}
\]

Similar to the proof of \( S_A^{An} = X^2 \), we can prove

\[
\{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i}{2F-a_{ef}^i+2a_{ep}^i} < \frac{F}{F+2a_{ep}^i}, 1 \leq i \leq n \}
\]

Therefore, we have

\[
S_A^{An} = \{ s_i | (a_{ef}^i = 0) \text{ or } (a_{ef}^i > 0 \text{ and } 1 + \frac{a_{ef}^i}{F} - \frac{a_{ef}^i}{a_{ef}^i} < 0), 1 \leq i \leq n \} = Z^2
\]

In conclusion, we have proved that \( S_B^{An} = X^2 \), \( S_F^{An} = Y^2 \) and \( S_A^{An} = Z^2 \).

5. Sørensen-Dice

As stated in Table 4.1, formula Sørensen-Dice is defined as follows.

\[
R_{SD}(s_i) = \frac{2a_{ef}^i}{2a_{ef}^i + a_{nf}^i + a_{ep}^i}
\]

It follows from Lemmas 4.3.1 and 4.3.2 that \( R_{SD}(s_i) = \frac{2a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} \) and \( R_{SD}(s_j) = \frac{2F}{2F + a_{ep}^i} \). Then, after Definition 4.2.1, we have

\[
S_B^{SD} = \{ s_i | \frac{2a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} > \frac{2F}{2F + a_{ep}^i}, 1 \leq i \leq n \} \quad (4.3.22)
\]

\[
S_F^{SD} = \{ s_i | \frac{2a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} = \frac{2F}{2F + a_{ep}^i}, 1 \leq i \leq n \} \quad (4.3.23)
\]

\[
S_A^{SD} = \{ s_i | \frac{2a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} < \frac{2F}{2F + a_{ep}^i}, 1 \leq i \leq n \} \quad (4.3.24)
\]

We are going to prove that \( S_B^{SD} \), \( S_F^{SD} \) and \( S_A^{SD} \) are equal to the above sets \( X^2 \) in (4.3.14), \( Y^2 \) in (4.3.15) and \( Z^2 \) in (4.3.16), respectively.
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

First, we will prove \( S_{BD}^{SD} = X^2 \). For any \( s_i \), we have either \( (a_{ef}^i = 0) \) or \( (a_{ef}^i > 0) \). Therefore, \( S_{BD}^{SD} \) in (4.3.22) can be re-written as

\[
S_{BD}^{SD} = \{ s_i | a_{ef}^i = 0 \text{ and } \frac{2a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} > \frac{2F}{2F + a_{ep}^i}, 1 \leq i \leq n \} \\
\cup \{ s_i | a_{ef}^i > 0 \text{ and } \frac{2a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} > \frac{2F}{2F + a_{ep}^i}, 1 \leq i \leq n \}
\]

Consider the case that \( (a_{ef}^i = 0) \). Since \( F > 0 \) and \( (2F + a_{ep}^i) > 0 \) after Lemma 4.3.1, we have \( \frac{2a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} = 0 \), which is contradictory to \( \frac{2a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} > \frac{2F}{2F + a_{ep}^i} \). Thus,

\[
\{ s_i | a_{ef}^i = 0 \text{ and } \frac{2a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} > \frac{2F}{2F + a_{ep}^i}, 1 \leq i \leq n \} = \emptyset
\]

Hence, we have

\[
S_{BD}^{SD} = \{ s_i | a_{ef}^i > 0 \text{ and } \frac{2a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} > \frac{2F}{2F + a_{ep}^i}, 1 \leq i \leq n \} \quad (4.3.25)
\]

- Assume that \( s_i \in S_{BD}^{SD} \). After (4.3.25), we have \( (a_{ef}^i > 0) \) and \( \frac{2a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} > \frac{2F}{2F + a_{ep}^i} \).
  
  Since \( a_{ef}^i > 0 \), \( F > 0 \), \( (a_{ef}^i + a_{ep}^i + F) > 0 \) (after Lemma 4.3.1) and \( (2F + a_{ep}^i) > 0 \) (after Lemma 4.3.1), \( \frac{2a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} > \frac{2F}{2F + a_{ep}^i} \) implies \( \frac{a_{ef}^i + a_{ep}^i + F}{a_{ef}^i} < \frac{2F + a_{ep}^i}{a_{ep}^i} \), which can be re-arranged as \( 1 + \frac{a_{ef}^i}{F} - \frac{2F}{a_{ef}^i} \frac{a_{ep}^i}{F} > a_{ef}^i > 0 \). Then, we have \( s_i \in X^2 \) after (4.3.14). Therefore, \( S_{BD}^{SD} \subseteq X^2 \).

- Assume that \( s_i \in X^2 \). After (4.3.14), we have \( (a_{ef}^i > 0) \) and \( 1 + \frac{a_{ef}^i}{F} - \frac{2F}{a_{ef}^i} \frac{a_{ep}^i}{F} > a_{ef}^i > 0 \). Obvioulsy, \( 1 + \frac{a_{ef}^i}{F} - \frac{2F}{a_{ef}^i} \frac{a_{ep}^i}{F} > a_{ef}^i > 0 \) can be re-arranged as \( \frac{a_{ef}^i + a_{ep}^i + F}{a_{ef}^i} < \frac{2F + a_{ep}^i}{a_{ep}^i} \), which implies \( \frac{2a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} > \frac{2F}{2F + a_{ep}^i} \) after \( a_{ef}^i > 0, F > 0, (a_{ef}^i + a_{ep}^i + F) > 0 \) and \( (2F + a_{ep}^i) > 0 \). Then, we have \( s_i \in S_{BD}^{SD} \) after (4.3.25). Therefore, \( X^2 \subseteq S_{BD}^{SD} \).

In summary, we have proved that \( S_{BD}^{SD} = X^2 \).

Similarly, we can prove that \( S_{BD}^{SD} = Y^2 \).

Next, we are going to prove \( S_{AD}^{SD} = Z^2 \). \( S_{AD}^{SD} \) in (4.3.24) can be re-written as follows.

\[
S_{AD}^{SD} = \{ s_i | a_{ef}^i = 0 \text{ and } \frac{a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} < \frac{2F}{2F + a_{ep}^i}, 1 \leq i \leq n \} \\
\cup \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i}{a_{ef}^i + a_{ep}^i + F} < \frac{2F}{2F + a_{ep}^i}, 1 \leq i \leq n \}
\]
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Consider the case \( (a_{ef}^i=0) \), which implies \[ \frac{2a_{ef}^i}{a_{ef}^i+a_{np}^i+F} = 0 < \frac{2F}{2F+a_{ep}^i} \]. Thus, \( (a_{ef}^i=0 \) and \( \frac{2a_{ef}^i}{a_{ef}^i+a_{np}^i+F} < \frac{2F}{2F+a_{ep}^i} \) \) is logically equivalent to \( (a_{ef}^i=0) \). Therefore, \( S_{SD}^A \) becomes

\[
\{ s_i | a_{ef}^i = 0, 1 \leq i \leq n \} \cup \{ s_i | a_{ef}^i > 0 \text{ and } \frac{2a_{ef}^i}{a_{ef}^i+a_{np}^i+F} < \frac{2F}{2F+a_{ep}^i}, 1 \leq i \leq n \}
\]

Similar to the proof of \( S_{SD}^B = X^2 \), we can prove

\[
\{ s_i | a_{ef}^i > 0 \text{ and } \frac{2a_{ef}^i}{a_{ef}^i+a_{np}^i+F} < \frac{2F}{2F+a_{ep}^i}, 1 \leq i \leq n \}
\]

\[
= \{ s_i | a_{ef}^i > 0 \text{ and } 1 + \frac{a_{ef}^i}{F} - \frac{a_{ep}^i}{a_{ef}^i} < 0, 1 \leq i \leq n \}
\]

Therefore, we have

\[
S_{SD}^A = \{ s_i | (a_{ef}^i=0) \text{ or } (a_{ef}^i > 0 \text{ and } 1 + \frac{a_{ef}^i}{F} - \frac{a_{ep}^i}{a_{ef}^i} < 0), 1 \leq i \leq n \} = Z^2
\]

In conclusion, we have proved that \( S_{SD}^B = X^2 \), \( S_{SD}^F = Y^2 \) and \( S_{SD}^A = Z^2 \).

6. Dice

As stated in Table 4.1, formula Dice is defined as

\[
R_D(s_i) = \frac{2a_{ef}^i}{a_{ef}^i + a_{np}^i + a_{ep}^i}
\]

It follows from Lemmas 4.3.1 and 4.3.2 that \( R_D(s_i) = \frac{2a_{ef}^i}{F+a_{ep}^i} \) and \( R_D(s_j) = \frac{2F}{F+a_{ep}^i} \). Then, after Definition 4.2.1, we have

\[
S_{SD}^D_B = \{ s_i | \frac{2a_{ef}^i}{F+a_{ep}^i} > \frac{2F}{F+a_{ep}^i}, 1 \leq i \leq n \} \quad \text{(4.3.26)}
\]

\[
S_{SD}^D_F = \{ s_i | \frac{2a_{ef}^i}{F+a_{ep}^i} = \frac{2F}{F+a_{ep}^i}, 1 \leq i \leq n \} \quad \text{(4.3.27)}
\]

\[
S_{SD}^D_A = \{ s_i | \frac{2a_{ef}^i}{F+a_{ep}^i} < \frac{2F}{F+a_{ep}^i}, 1 \leq i \leq n \} \quad \text{(4.3.28)}
\]

We are going to prove that \( S_{SD}^D_B \), \( S_{SD}^D_F \) and \( S_{SD}^D_A \) are equal to the above sets \( X^2 \) in (4.3.14), \( Y^2 \) in (4.3.15) and \( Z^2 \) in (4.3.16), respectively.
First, we will prove $S_B^D = X^2$. For any $s_i$, we have either ($a_{ef}^i = 0$) or ($a_{ef}^i > 0$). Therefore, $S_B^D$ defined in (4.3.26) can be re-written as

$$S_B^D = \{s_i | a_{ef}^i = 0 \text{ and } \frac{2a_{ef}^i}{F + a_{ep}} > \frac{2F}{F + a_{ep}}, 1 \leq i \leq n \} \cup \{s_i | a_{ef}^i > 0 \text{ and } \frac{2a_{ef}^i}{F + a_{ep}} > \frac{2F}{F + a_{ep}}, 1 \leq i \leq n \}$$

Consider the case that ($a_{ef}^i = 0$). Since $F > 0$ and ($F + a_{ef}^i > 0$) after Lemma 4.3.1, we have $\frac{2a_{ef}^i}{F + a_{ep}} = 0 < \frac{2F}{F + a_{ep}}$, which is contradictory to $\frac{2a_{ef}^i}{F + a_{ep}} > \frac{2F}{F + a_{ep}}$. Thus,

$$\{s_i | a_{ef}^i = 0 \text{ and } \frac{2a_{ef}^i}{F + a_{ep}} > \frac{2F}{F + a_{ep}}, 1 \leq i \leq n \} = \emptyset$$

Hence, we have

$$S_B^D = \{s_i | a_{ef}^i > 0 \text{ and } \frac{2a_{ef}^i}{F + a_{ep}} > \frac{2F}{F + a_{ep}}, 1 \leq i \leq n \}$$

(4.3.29)

- Assume that $s_i \in S_B^D$. After (4.3.29), we have ($a_{ef}^i > 0$ and $\frac{2a_{ef}^i}{F + a_{ep}} > \frac{2F}{F + a_{ep}}$). Since $a_{ef}^i > 0$, $F > 0$, ($F + a_{ef}^i > 0$) after Lemma 4.3.1 and ($F + a_{ep}^i > 0$) after Lemma 4.3.1, we have $\frac{F + a_{ef}^i}{F + a_{ep}^i} < \frac{2a_{ef}^i}{F + a_{ep}}$ from $\frac{2a_{ef}^i}{F + a_{ep}} > \frac{2F}{F + a_{ep}}$. After re-arranging the terms, $\frac{F + a_{ef}^i}{a_{ef}^i} < \frac{F + a_{ef}^i}{a_{ep}^i}$ becomes $1 + \frac{a_{ef}^i}{F - a_{ep}^i} > 0$. It follows from (4.3.14) that $s_i \in X^2$. Therefore, $S_B^D \subseteq X^2$.

- Assume that $s_i \in X^2$. After (4.3.14), we have ($a_{ef}^i > 0$ and $1 + \frac{a_{ef}^i}{F - a_{ep}^i} - \frac{a_{ep}^i}{a_{ef}^i} > 0$). Obviously, $1 + \frac{a_{ef}^i}{F - a_{ep}^i} - \frac{a_{ep}^i}{a_{ef}^i} > 0$ can be re-arranged as $\frac{F + a_{ef}^i}{a_{ef}^i} < \frac{F + a_{ef}^i}{a_{ef}^i}$, which implies $\frac{2a_{ef}^i}{F + a_{ep}^i} < \frac{2F}{F + a_{ep}^i}$ after $a_{ef}^i > 0$, $F > 0$, ($F + a_{ep}^i > 0$) and ($F + a_{ef}^i > 0$). Then, we have $s_i \in S_B^D$ after (4.3.29). Therefore, $X^2 \subseteq S_B^D$.

In summary, we have proved that $S_B^D = X^2$.

Similarly, we can prove that $S_B^D = Y^2$.

Next, we are going to prove $S_A^D = Z^2$. $S_A^D$ in (4.3.28) can be re-written as follows

$$2S_A^D = \{s_i | a_{ef}^i = 0 \text{ and } \frac{2a_{ef}^i}{F + a_{ep}^i} < \frac{2F}{F + a_{ep}^i}, 1 \leq i \leq n \} \cup \{s_i | a_{ef}^i > 0 \text{ and } \frac{2a_{ef}^i}{F + a_{ep}^i} < \frac{2F}{F + a_{ep}^i}, 1 \leq i \leq n \}$$
Consider the case \((a^i_{ef} = 0)\), which implies \(\frac{2a^i_{ef}}{F + a^i_{ep}} = 0 < \frac{2F}{F + a^i_{ep}}\). Thus, \((a^i_{ef} = 0 \text{ and } \frac{2a^i_{ef}}{F + a^i_{ep}} < \frac{2F}{F + a^i_{ep}})\) is logically equivalent to \((a^i_{ef} = 0)\). Therefore, \(S^D_A\) becomes
\[
\{s_i|a^i_{ef}=0, 1\leq i \leq n\} \cup \{s_i|a^i_{ef}>0 \text{ and } \frac{2a^i_{ef}}{F + a^i_{ep}} \leq \frac{2F}{F + a^i_{ep}}, 1 \leq i \leq n\}
\]

Similar to the proof of \(S^D_B = X^2\), we can prove
\[
\{s_i|a^i_{ef}>0 \text{ and } \frac{2a^i_{ef}}{F + a^i_{ep}} < \frac{2F}{F + a^i_{ep}}, 1 \leq i \leq n\} = \{s_i|a^i_{ef}>0 \text{ and } 1 + \frac{a^i_{ep}}{a^i_{ef}} \frac{F}{a^i_{ep}} a^i_{ef} < 0, 1 \leq i \leq n\}
\]

Therefore, we have
\[
S^D_A = \{s_i|(a^i_{ef} = 0) \text{ or } (a^i_{ef} > 0 \text{ and } 1 + \frac{a^i_{ep}}{a^i_{ef}} \frac{F}{a^i_{ep}} a^i_{ef} < 0), 1 \leq i \leq n\} = Z^2
\]

In conclusion, we have proved that \(S^D_B = X^2, S^D_F = Y^2\) and \(S^D_A = Z^2\).

7. Goodman

As stated in Table 4.1, formula Goodman is defined as
\[
R_G(s_i) = \frac{2a^i_{ef} - a^i_{nf} - a^i_{ep}}{2a^i_{ef} + a^i_{nf} + a^i_{ep}}
\]

It follows from Lemmas 4.3.1 and 4.3.2 that \(R_G(s_i) = \frac{3a^i_{ef} - F - a^i_{ep}}{a^i_{ef} + F + a^i_{ep}}\) and \(R_G(s_f) = \frac{2F - a^i_{ef}}{2F + a^i_{ep}}\). Then, after Definition 4.2.1, we have
\[
S^G_B = \{s_i|\frac{3a^i_{ef} - F - a^i_{ep}}{a^i_{ef} + F + a^i_{ep}} > \frac{2F - a^i_{ef}}{2F + a^i_{ep}}, 1 \leq i \leq n\} \quad (4.3.30)
\]
\[
S^G_F = \{s_i|\frac{3a^i_{ef} - F - a^i_{ep}}{a^i_{ef} + F + a^i_{ep}} = \frac{2F - a^i_{ef}}{2F + a^i_{ep}}, 1 \leq i \leq n\} \quad (4.3.31)
\]
\[
S^G_A = \{s_i|\frac{3a^i_{ef} - F - a^i_{ep}}{a^i_{ef} + F + a^i_{ep}} < \frac{2F - a^i_{ef}}{2F + a^i_{ep}}, 1 \leq i \leq n\} \quad (4.3.32)
\]

We are going to prove that \(S^G_B, S^G_F\) and \(S^G_A\) are equal to the above sets \(X^2\) in (4.3.14), \(Y^2\) in (4.3.15) and \(Z^2\) in (4.3.16), respectively.
First, we will prove \( S^G_B = X^2 \). For any \( s_i \), we have either \( (a_{ef}^i = 0) \) or \( (a_{ef}^i > 0) \). Therefore, \( S^G_B \) defined in (4.3.30) can be re-written as

\[
S^G_B = \begin{cases} 
\{ s_i | a_{ef}^i = 0 \} & \text{and} \quad \frac{3a_{ef}^i - F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} > \frac{2F - a_{ep}^f}{2F + a_{ep}^f}, 1 \leq i \leq n \\
\cup \{ s_i | a_{ef}^i > 0 \} & \text{and} \quad \frac{3a_{ef}^i - F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} > \frac{2F - a_{ep}^f}{2F + a_{ep}^f}, 1 \leq i \leq n 
\end{cases}
\]

Consider the case that \( (a_{ef}^i = 0) \). Since \( F > 0 \), we have \( \frac{2F - a_{ep}^f}{2F + a_{ep}^f} > -1 \). Thus, we have

\[
\frac{3a_{ef}^i - F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} = -1 < \frac{2F - a_{ep}^f}{2F + a_{ep}^f}, \text{ which is contradictory to } \frac{3a_{ef}^i - F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} > \frac{2F - a_{ep}^f}{2F + a_{ep}^f}. \text{ Thus,}
\]

\[
\{ s_i | a_{ef}^i = 0 \} \quad \text{and} \quad \frac{3a_{ef}^i - F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} > \frac{2F - a_{ep}^f}{2F + a_{ep}^f}, 1 \leq i \leq n \} = \emptyset
\]

Hence, we have

\[
S^G_B = \{ s_i | a_{ef}^i > 0 \} \quad \text{and} \quad \frac{3a_{ef}^i - F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} > \frac{2F - a_{ep}^f}{2F + a_{ep}^f}, 1 \leq i \leq n \}
\]  

(4.3.33)

- Assume that \( s_i \in S^G_B \). After (4.3.33), we have \( (a_{ef}^i > 0) \) and \( \frac{3a_{ef}^i - F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} > \frac{2F - a_{ep}^f}{2F + a_{ep}^f} \). After re-arranging terms, \( \frac{3a_{ef}^i - F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} > \frac{2F - a_{ep}^f}{2F + a_{ep}^f} \) becomes \( \frac{4F + 4a_{ef}^i a_{ep}^i - 4F a_{ep}^i - 4F^2}{(a_{ef}^i + F + a_{ep}^i)(2F + a_{ep}^f)} > 0 \). Since \( a_{ef}^i > 0 \) and \( F > 0 \), we have \( Fa_{ef}^i > 0 \). Furthermore, since \( (a_{ef}^i + F + a_{ep}^i)(2F + a_{ep}^f) > 0 \) after Lemma 4.3.1, then, \( \frac{4F + 4a_{ef}^i a_{ep}^i - 4F a_{ep}^i - 4F^2}{(a_{ef}^i + F + a_{ep}^i)(2F + a_{ep}^f)} > 0 \) implies \( \frac{-F + a_{ef}^i a_{ep}^i - F a_{ep}^i - F^2}{a_{ef}^i + F + a_{ep}^i} > 0 \). Then, we have \( s_i \in X^2 \) after (4.14). Therefore, \( S^G_B \subseteq X^2 \).

- Assume that \( s_i \in X^2 \). After (4.14), we have \( (a_{ef}^i > 0) \) and \( 1 + \frac{a_{ef}^i - F}{a_{ef}^i} - \frac{a_{ef}^i}{a_{ef}^i} > 0 \). Obviously, \( 1 + \frac{a_{ef}^i - F}{a_{ef}^i} - \frac{a_{ef}^i}{a_{ef}^i} > 0 \) can be re-arranged as \( \frac{Fa_{ef}^i + a_{ep}^i a_{ef}^i - Fa_{ep}^f - F^2}{a_{ef}^i} > 0 \). Since \( a_{ef}^i > 0 \), we have \( Fa_{ef}^i > 0 \) because \( F > 0 \). Furthermore, since \( (a_{ef}^i + F + a_{ep}^i)(2F + a_{ep}^f) > 0 \) after Lemma 4.3.1, then, \( \frac{Fa_{ef}^i + a_{ep}^i a_{ef}^i - Fa_{ep}^f - F^2}{a_{ef}^i + F + a_{ep}^i} > 0 \) implies \( \frac{4F a_{ef}^i + a_{ep}^i a_{ef}^i - Fa_{ep}^f - F^2}{(a_{ef}^i + F + a_{ep}^i)(2F + a_{ep}^f)} > 0 \), which can be re-arranged as \( \frac{3a_{ef}^i - F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} > \frac{2F - a_{ep}^f}{2F + a_{ep}^f} \). Then, we have \( s_i \in S^G_B \) after (4.3.33). Therefore, \( X^2 \subseteq S^G_B \).

In summary, we have proved that \( S^G_B = X^2 \).

Similarly, we can prove that \( S^G_E = Y^2 \).
Next, we are going to prove $S_A^G = Z^2$. $S_A^G$ in (4.3.32) can be re-written as follows

$$S_A^G = \{ s_i | a_{ef}^i = 0 \text{ and } \frac{3a_{ef}^i F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} < \frac{2F - a_{ef}^i}{2F + a_{ep}^i}, 1 \leq i \leq n \}$$

$$\cup \{ s_i | a_{ef}^i > 0 \text{ and } \frac{3a_{ef}^i F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} < \frac{2F - a_{ef}^i}{2F + a_{ep}^i}, 1 \leq i \leq n \}$$

Consider the case that $(a_{ef}^i = 0)$. As shown in the above proof of $S_B^G = X^2$, $(a_{ef}^i = 0)$ implies $\frac{3a_{ef}^i F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} < \frac{2F - a_{ef}^i}{2F + a_{ep}^i}$. Thus, $(a_{ef}^i = 0)$ and $\frac{3a_{ef}^i F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} < \frac{2F - a_{ef}^i}{2F + a_{ep}^i}$ is logically equivalent to $(a_{ef}^i = 0)$.

Therefore, $S_A^G$ becomes

$$\{ s_i | a_{ef}^i = 0, 1 \leq i \leq n \} \cup \{ s_i | a_{ef}^i > 0 \text{ and } \frac{3a_{ef}^i F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} < \frac{2F - a_{ef}^i}{2F + a_{ep}^i}, 1 \leq i \leq n \}$$

Similar to the proof of $S_B^G = X^2$, we can prove

$$\{ s_i | a_{ef}^i > 0 \text{ and } \frac{3a_{ef}^i F - a_{ep}^i}{a_{ef}^i + F + a_{ep}^i} < \frac{2F - a_{ef}^i}{2F + a_{ep}^i}, 1 \leq i \leq n \}$$

$$= \{ s_i | a_{ef}^i > 0 \text{ and } 1 + \frac{a_{ef}^i}{a_{ef}^i + a_{ep}^i} - \frac{a_{ep}^i}{a_{ef}^i} < 0, 1 \leq i \leq n \}$$

Therefore, we have

$$S_A^G = \{ s_i | (a_{ef}^i = 0) \text{ or } (a_{ef}^i > 0 \text{ and } 1 + \frac{a_{ef}^i}{a_{ef}^i + a_{ep}^i} - \frac{a_{ep}^i}{a_{ef}^i} < 0), 1 \leq i \leq n \} = Z^2$$

In conclusion, we have proved that $S_B^G = X^2$, $S_F^G = Y^2$ and $S_A^G = Z^2$.

8. Tarantula

As stated in Table 4.1, formula Tarantula is defined as follows.

$$R_T(s_i) = \frac{a_{ef}^i}{a_{ef}^i + a_{ep}^i} / \left( \frac{a_{ef}^i}{a_{ef}^i + a_{n^f}^i} + \frac{a_{ep}^i}{a_{ep}^i + a_{n^p}^i} \right)$$

It follows from Lemmas 4.3.1 and 4.3.2 that $R_T(s_i) = \frac{a_{ef}^i}{a_{ef}^i + a_{ep}^i}$ and $R_T(s_f) = 1 / (1 + \frac{a_{ef}^i}{a_{ep}^i})$.

Then, after Definition 4.2.1, we have
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

$$S_B^T = \{ s_i | \frac{a_{ef}^i}{F} / (\frac{a_{ef}^i}{F} + \frac{a_{ep}^i}{P}) > 1/(1 + \frac{a_{ef}^i}{P}), 1 \leq i \leq n \}$$  \hspace{1cm} (4.3.34)$$

$$S_F^T = \{ s_i | \frac{a_{ef}^i}{F} / (\frac{a_{ef}^i}{F} + \frac{a_{ep}^i}{P}) = 1/(1 + \frac{a_{ef}^i}{P}), 1 \leq i \leq n \}$$  \hspace{1cm} (4.3.35)$$

$$S_A^T = \{ s_i | \frac{a_{ef}^i}{F} / (\frac{a_{ef}^i}{F} + \frac{a_{ep}^i}{P}) < 1/(1 + \frac{a_{ef}^i}{P}), 1 \leq i \leq n \}$$  \hspace{1cm} (4.3.36)$$

We are going to prove that $S_B^T$, $S_F^T$ and $S_A^T$ are equal to the following sets $X^3$, $Y^3$ and $Z^3$, respectively.

$$X^3 = \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_f^i}{F} - \frac{a_{ep}^i}{a_{ef}^i} > 0, 1 \leq i \leq n \}$$  \hspace{1cm} (4.3.37)$$

$$Y^3 = \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_f^i}{F} - \frac{a_{ep}^i}{a_{ef}^i} = 0, 1 \leq i \leq n \}$$  \hspace{1cm} (4.3.38)$$

$$Z^3 = \{ s_i | (a_{ef}^i = 0) \text{ or } (a_{ef}^i > 0 \text{ and } \frac{a_f^i}{F} - \frac{a_{ep}^i}{a_{ef}^i} < 0), 1 \leq i \leq n \}$$  \hspace{1cm} (4.3.39)$$

First, we will prove $S_B^T = X^3$. For any $s_i$, we have either $(a_{ef}^i = 0)$ or $(a_{ef}^i > 0)$. Therefore, $S_B^T$ defined in (4.3.34) can be re-written as

$$S_B^T = \{ s_i | a_{ef}^i = 0 \text{ and } \frac{a_f^i}{F} / (\frac{a_{ef}^i}{F} + \frac{a_{ep}^i}{P}) > 1/(1 + \frac{a_{ef}^i}{P}), 1 \leq i \leq n \}$$

$$\cup \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_f^i}{F} / (\frac{a_{ef}^i}{F} + \frac{a_{ep}^i}{P}) > 1/(1 + \frac{a_{ef}^i}{P}), 1 \leq i \leq n \}$$

Consider the case that $(a_{ef}^i = 0)$. Since $(1 + \frac{a_{ef}^i}{P}) > 0$ after Lemma 4.3.1, we have

$$\frac{a_f^i}{F} / (\frac{a_{ef}^i}{F} + \frac{a_{ep}^i}{P}) = 0 < 1/(1 + \frac{a_{ef}^i}{P}),$$

which is contradictory to $\frac{a_f^i}{F} / (\frac{a_{ef}^i}{F} + \frac{a_{ep}^i}{P}) > 1/(1 + \frac{a_{ef}^i}{P})$. Thus,

$$\{ s_i | a_{ef}^i = 0 \text{ and } \frac{a_f^i}{F} / (\frac{a_{ef}^i}{F} + \frac{a_{ep}^i}{P}) > 1/(1 + \frac{a_{ef}^i}{P}), 1 \leq i \leq n \} = \emptyset$$

Hence, we have

$$S_B^T = \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_f^i}{F} / (\frac{a_{ef}^i}{F} + \frac{a_{ep}^i}{P}) > 1/(1 + \frac{a_{ef}^i}{P}), 1 \leq i \leq n \}$$  \hspace{1cm} (4.3.40)$$

- Assume that $s_i \in S_B^T$. After (4.3.40), we have $(a_{ef}^i > 0$ and $(\frac{a_f^i}{F} / (\frac{a_{ef}^i}{F} + \frac{a_{ep}^i}{P}) > 1/(1 + \frac{a_{ef}^i}{P}))).$

Since $a_{ep}^i > 0$, we have $\frac{F}{a_{ef}^i} > 0$ because $F > 0$. Then, $\frac{a_f^i}{F} / (\frac{a_{ef}^i}{F} + \frac{a_{ep}^i}{P}) > 1/(1 + \frac{a_{ef}^i}{P})$ implies

$$1/(1 + \frac{a_{ep}^i}{a_{ef}^i}) > 1/(1 + \frac{a_{ef}^i}{F}).$$

Furthermore, it follows from $F/a_{ef}^i > 0$ and Lemma 4.3.1 that
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Next, we are going to prove \( S \) = \( X^3 \).

In summary, we have proved that \( S^T_B \subseteq X^3 \).

Assume that \( s_i \in X^3 \). After (4.3.37), we have \((a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{P} > a^i_{ep} > 0)\). Since \( \frac{F}{P} > 0 \), after re-arranging the terms and multiplying each side by \( P \), \( \frac{a^i_{ef}}{P} - \frac{a^i_{ef}}{a^i_{ep}} > 0 \) becomes \( a^i_{ep} \frac{F}{a^i_{ef}} < \frac{a^i_{ef}}{P} \).

Then, we have \( s_i \in S^T_B \) after (4.3.40). Therefore, \( X^3 \subseteq S^T_B \).

In conclusion, we have proved that \( S^T_B = X^3 \).

Similarly, we can prove that \( S^T_F = Y^3 \).

Next, we are going to prove \( S^T_A = Z^3 \). \( S^T_A \) in (4.3.36) can be re-written as follows

\[
S^T_A = \{ s_i | a^i_{ef} = 0 \text{ and } \frac{a^i_{ef}}{F} / (\frac{a^i_{ef}}{F} + \frac{a^i_{ep}}{P}) < 1 / (1 + \frac{a^i_{ef}}{P}), 1 \leq i \leq n \} \\
\cap \{ s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{F} / (\frac{a^i_{ef}}{F} + \frac{a^i_{ep}}{P}) < 1 / (1 + \frac{a^i_{ef}}{P}), 1 \leq i \leq n \}
\]

Consider the case \((a^i_{ef} = 0)\), which implies \( \frac{a^i_{ef}}{F} / (\frac{a^i_{ef}}{F} + \frac{a^i_{ep}}{P}) = 0 < 1 / (1 + \frac{a^i_{ef}}{P}) \). Thus, \((a^i_{ef} = 0 \text{ and } \frac{a^i_{ef}}{F} / (\frac{a^i_{ef}}{F} + \frac{a^i_{ep}}{P}) < 1 / (1 + \frac{a^i_{ef}}{P}))\) is logically equivalent to \((a^i_{ef} = 0)\). Therefore, \( S^T_A \) becomes

\[
\{ s_i | a^i_{ef} = 0, 1 \leq i \leq n \} \cup \{ s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{F} / (\frac{a^i_{ef}}{F} + \frac{a^i_{ep}}{P}) < 1 / (1 + \frac{a^i_{ef}}{P}), 1 \leq i \leq n \}
\]

Similar to the proof of \( S^T_B = X^3 \), we can prove

\[
\{ s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{F} / (\frac{a^i_{ef}}{F} + \frac{a^i_{ep}}{P}) < 1 / (1 + \frac{a^i_{ef}}{P}), 1 \leq i \leq n \}
\]

\[
= \{ s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{F} - \frac{a^i_{ep}}{a^i_{ef}} < 0, 1 \leq i \leq n \}
\]

Therefore, we have

\[
S^T_A = \{ s_i | (a^i_{ef} = 0) \text{ or } (a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{F} - \frac{a^i_{ep}}{a^i_{ef}} < 0), 1 \leq i \leq n \} = Z^3
\]

In conclusion, we have proved that \( S^T_B = X^3 \), \( S^T_F = Y^3 \) and \( S^T_A = Z^3 \).
9. **qe**

As stated in Table 4.1, formula \( q_e \) is defined as follows.

\[
R_{QE}(s_i) = \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}}
\]

It follows from Lemma 4.3.2 that \( R_{QE}(s_f) = F \). Then, after Definition 4.2.1, we have

\[
S^{QE}_B = \{ s_i | a^i_{ef} > F, 1 \leq i \leq n \} \tag{4.3.41}
\]

\[
S^{QE}_F = \{ s_i | a^i_{ef} = F, 1 \leq i \leq n \} \tag{4.3.42}
\]

\[
S^{QE}_A = \{ s_i | a^i_{ef} < F, 1 \leq i \leq n \} \tag{4.3.43}
\]

We are going to prove that \( S^{QE}_B, S^{QE}_F \) and \( S^{QE}_A \) are equal to the above sets \( X^3 \) in (4.3.37), \( Y^3 \) in (4.3.38) and \( Z^3 \) in (4.3.39), respectively.

First, we will prove \( S^{QE}_B = X^3 \). For any \( s_i \), we have either \( a^i_{ef} = 0 \) or \( a^i_{ef} > 0 \). Therefore, \( S^{QE}_B \) in (4.3.41) can be re-written as

\[
S^{QE}_B = \{ s_i | a^i_{ef} = 0 \text{ and } \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} > \frac{F}{F + a^i_{ep}}, 1 \leq i \leq n \}
\]

\[
\cup \{ s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} > \frac{F}{F + a^i_{ep}}, 1 \leq i \leq n \}
\]

Consider the case that \( a^i_{ef} = 0 \). Since \( F > 0 \) and \( (F + a^i_{ep}) > 0 \) after Lemma 4.3.1, we have

\[
\frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} = 0 < \frac{F}{F + a^i_{ep}},
\]

which is contradictory to \( \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} > \frac{F}{F + a^i_{ep}} \). Thus,

\[
\{ s_i | a^i_{ef} = 0 \text{ and } \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} > \frac{F}{F + a^i_{ep}}, 1 \leq i \leq n \} = \emptyset
\]

Hence, we have

\[
S^{QE}_B = \{ s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} > \frac{F}{F + a^i_{ep}}, 1 \leq i \leq n \} \tag{4.3.44}
\]

- Assume that \( s_i \in S^{QE}_B \). After (4.3.44), we have \( a^i_{ef} > 0 \) and \( \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} > \frac{F}{F + a^i_{ep}} \). Since \( a^i_{ef} > 0 \)}
and $F>0$, we have $1/(1+a_{ef}^i/a_{ep}^i)>1/(1+a_{ef}^i/F)$ from $a_{ef}^i/F > a_{ef}^i + a_{ep}^i$. Furthermore, since $a_{ef}^i > 0$ and $a_{ep}^i > 0$ after $a_{ef}^i > 0$, $F > 0$ and Lemma 4.3.1, $1/(1+a_{ef}^i/a_{ep}^i)>1/(1+a_{ef}^i/F)$ implies $a_{ef}^i/F - a_{ef}^i > 0$. Then, we have $s_i \in X^3$ after (4.3.37). Therefore, $S_B^{QE} \subseteq X^3$.

- Assume that $s_i \in X^3$. After (4.3.37), we have $(a_{ef}^i > 0$ and $a_{ep}^i > a_{ef}^i > 0$). Obviously, $a_{ef}^i/F - a_{ef}^i > 0$ can be re-written as $a_{ef}^i/a_{ep}^i < a_{ef}^i/F$. Since $a_{ef}^i > 0$ and $a_{ep}^i > 0$ after $a_{ef}^i > 0$, $F > 0$ and Lemma 4.3.1, $a_{ef}^i/a_{ep}^i < a_{ef}^i/F$ implies $1/(1+a_{ef}^i/a_{ep}^i)>1/(1+a_{ef}^i/F)$. Furthermore, we have $a_{ef}^i + a_{ep}^i > F$ because $a_{ef}^i > 0$ and $F > 0$. Then, we have $s_i \in S_B^{QE}$ after (4.3.44). Therefore, $X^3 \subseteq S_B^{QE}$.

In summary, we have proved that $S_B^{QE} = X^3$.

Similarly, we can prove that $S_F^{QE} = Y^3$.

Next, we are going to prove $S_A^{QE} = Z^3$. $S_A^{QE}$ in (4.3.43) can be re-written as follows

$$S_A^{QE} = \{ s_i | a_{ef}^i = 0 \text{ and } a_{ef}^i < a_{ef}^i + a_{ep}^i, 1 \leq i \leq n \}$$

$$\cup \{ s_i | a_{ef}^i > 0 \text{ and } a_{ef}^i < a_{ef}^i + a_{ep}^i, 1 \leq i \leq n \}$$

Consider the case $(a_{ef}^i = 0)$, which implies $a_{ef}^i/a_{ep}^i = 0 < F/(F+a_{ep}^i)$. Thus, $(a_{ef}^i = 0$ and $a_{ef}^i > 0$ and $a_{ef}^i < a_{ef}^i + a_{ep}^i$) is logically equivalent to $(a_{ef}^i = 0)$. Therefore, $S_A^{QE}$ becomes

$$\{ s_i | a_{ef}^i = 0, 1 \leq i \leq n \} \cup \{ s_i | a_{ef}^i > 0 \text{ and } a_{ef}^i/a_{ep}^i < F/(F+a_{ep}^i), 1 \leq i \leq n \}$$

Similar to the proof of $S_B^{QE} = X^3$, we can prove

$$\{ s_i | a_{ef}^i > 0 \text{ and } a_{ef}^i/F < a_{ef}^i/a_{ep}^i, 1 \leq i \leq n \} = \{ s_i | a_{ef}^i > 0 \text{ and } a_{ep}^i/a_{ef}^i < 0, 1 \leq i \leq n \}$$

Therefore, we have

$$S_A^{QE} = \{ s_i | (a_{ef}^i = 0) \text{ or } (a_{ef}^i > 0 \text{ and } a_{ep}^i/a_{ef}^i < 0), 1 \leq i \leq n \} = Z^3$$
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

In conclusion, we have proved that $S_B^{QE} = X^3$, $S_F^{QE} = Y^3$ and $S_A^{QE} = Z^3$.

10. CBI Inc.

As stated in Table 4.1, formula CBI Inc. is defined as follows.

$$R_C(s_i) = \frac{a_{ef}^i}{a_{ef}^i + a_{ep}^i} - \frac{a_{ef}^i + a_{nf}^i}{a_{ef}^i + a_{nf}^i + a_{np}^i}$$

It follows from Lemmas 4.3.1 and 4.3.2 that $R_C(s_i) = \frac{a_{ef}^i}{a_{ef}^i + a_{ep}^i} - \frac{F}{F + a_{ep}^i}$ and $R_C(s_f) = \frac{F}{F + a_{ep}^i} - \frac{F}{F + P}$. Then, after Definition 4.2.1, we have

$$S_B^C = \{s_i| \frac{a_{ef}^i}{a_{ef}^i + a_{ep}^i} > \frac{F}{F + P}, 1 \leq i \leq n\}$$

$$S_F^C = \{s_i| \frac{a_{ef}^i}{a_{ef}^i + a_{ep}^i} = \frac{F}{F + P}, 1 \leq i \leq n\}$$

$$S_A^C = \{s_i| \frac{a_{ef}^i}{a_{ef}^i + a_{ep}^i} < \frac{F}{F + P}, 1 \leq i \leq n\}$$

Obviously, the above sets defined in (4.3.45), (4.3.46) and (4.3.47) are the same as $S_B^{QE}$ in (4.3.41), $S_F^{QE}$ in (4.3.42) and $S_A^{QE}$ in (4.3.43), respectively, and hence are equal to $X^3$ in (4.3.37), $Y^3$ in (4.3.38) and $Z^3$ in (4.3.39), respectively.

11. Wong2

As stated in Table 4.1, formula Wong2 is defined as follows.

$$R_{W2}(s_i) = a_{ef}^i - a_{ep}^i$$

After Lemma 4.3.2 and Definition 4.2.1, by re-arranging the terms, we have

$$S_B^{W2} = \{s_i| a_{ef}^i - F + (a_{ep}^i - a_{ep}^i) > 0, 1 \leq i \leq n\}$$

$$S_F^{W2} = \{s_i| a_{ef}^i - F + (a_{ep}^i - a_{ep}^i) = 0, 1 \leq i \leq n\}$$

$$S_A^{W2} = \{s_i| a_{ef}^i - F + (a_{ep}^i - a_{ep}^i) < 0, 1 \leq i \leq n\}$$
12. **Hamann**

As stated in Table 4.1, formula Hamann is defined as follows.

\[ R_{HN}(s_i) = \frac{a_{ef}^i + a_{np}^i - a_{nf}^i - a_{ep}^i}{a_{ef}^i + a_{np}^i + a_{ep}^i + a_{np}^i} \]


It follows from Lemmas 4.3.1 and 4.3.2 that \( R_{HN}(s_i) = \frac{P-F+2a_{ef}^i-2a_{ep}^i}{F+P} \) and \( R_{HN}(s_f) = \frac{P-F-2a_{ep}^f}{F+P} \). Then, after Definition 4.2.1, we have

\[ S_B^{HN} = \{ s_i | \frac{P-F+2a_{ef}^i-2a_{ep}^i}{F+P} > \frac{P+F-2a_{ef}^i}{F+P}, 1 \leq i \leq n \} \]  
\[ S_F^{HN} = \{ s_i | \frac{P-F+2a_{ef}^i-2a_{ep}^i}{F+P} = \frac{P+F-2a_{ef}^i}{F+P}, 1 \leq i \leq n \} \]  
\[ S_A^{HN} = \{ s_i | \frac{P-F+2a_{ef}^i-2a_{ep}^i}{F+P} < \frac{P+F-2a_{ef}^i}{F+P}, 1 \leq i \leq n \} \]

We are going to prove that \( S_B^{HN}, S_F^{HN} \) and \( S_A^{HN} \) are equal to the above sets \( S_B^{W2} \) in (4.3.48), \( S_F^{W2} \) in (4.3.49) and \( S_A^{W2} \) in (4.3.50), respectively.

First, we will prove \( S_B^{HN} = S_B^{W2} \).

- Assume that \( s_i \in S_B^{HN} \). After (4.3.51), we have \( \frac{P-F+2a_{ef}^i-2a_{ep}^i}{F+P} > \frac{P+F-2a_{ef}^i}{F+P} \). Since \( F+P > 0 \), we have \( P-F+2a_{ef}^i-2a_{ep}^i > P+F-2a_{ef}^i \). Thus, \( (a_{ef}^i-F) + (a_{ef}^i-a_{ep}^i) > 0 \), and hence \( s_i \in S_B^{W2} \) after (4.3.48). Therefore, \( S_B^{HN} \subseteq S_B^{W2} \).

- Assume that \( s_i \notin S_B^{HN} \). After (4.3.48), we have \( (a_{ef}^i-F) + (a_{ef}^i-a_{ep}^i) > 0 \), which implies \( P-F+2a_{ef}^i-2a_{ep}^i > P+F-2a_{ep}^i \). Since \( F+P > 0 \), we have \( \frac{P-F+2a_{ef}^i-2a_{ep}^i}{F+P} > \frac{P+F-2a_{ep}^i}{F+P} \). Then, we have \( s_i \in S_B^{HN} \) after (4.3.51). Therefore, \( S_B^{W2} \subseteq S_B^{HN} \).

In summary, we have proved that \( S_B^{HN} = S_B^{W2} \).

Similarly, we can prove \( S_F^{HN} = S_F^{W2} \) and \( S_A^{HN} = S_A^{W2} \).

13. **Simple Matching**

As stated in Table 4.1, formula Simple Matching is defined as follows.

\[ R_{SM}(s_i) = \frac{a_{ef}^i + a_{np}^i}{a_{ef}^i + a_{np}^i + a_{ep}^i + a_{np}^i} \]

It follows from Lemmas 4.3.1 and 4.3.2 that \( R_{SM}(s_i) = \frac{P+a_{ef}^i-a_{ep}^i}{F+P} \) and \( R_{SM}(s_f) = \frac{P-F+a_{ef}^f}{F+P} \).
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Then, after Definition 4.2.1, we have

\[ S_{SM}^{B} = \{ s_i \frac{P + a_{ef}^i - a_{ep}^i}{F + P} > \frac{P + F - a_{ep}^i}{F + P}, 1 \leq i \leq n \} \]  \hspace{1cm} (4.3.54)

\[ S_{SM}^{F} = \{ s_i \frac{P + a_{ef}^i - a_{ep}^i}{F + P} = \frac{P + F - a_{ep}^i}{F + P}, 1 \leq i \leq n \} \]  \hspace{1cm} (4.3.55)

\[ S_{SM}^{A} = \{ s_i \frac{P + a_{ef}^i - a_{ep}^i}{F + P} < \frac{P + F - a_{ep}^i}{F + P}, 1 \leq i \leq n \} \]  \hspace{1cm} (4.3.56)

We are going to prove that \( S_{SM}^{B} \), \( S_{SM}^{F} \) and \( S_{SM}^{A} \) are equal to the above sets \( S_{BW}^{W} \) in (4.3.49), \( S_{FW}^{W} \) in (4.3.44) and \( S_{AW}^{W} \) in (4.3.50), respectively.

First, we will prove \( S_{SM}^{B} = S_{BW}^{W} \).

- Assume that \( s_i \in S_{SM}^{B} \). After (4.3.54), we have \( \frac{P + a_{ef}^i - a_{ep}^i}{F + P} > \frac{P + F - a_{ep}^i}{F + P} \). Since \( F + P > 0 \), we have \( P + a_{ef}^i - a_{ep}^i > P + F - a_{ep}^i \), and hence \( (a_{ef}^i - F) + (a_{ep}^i - a_{ep}^i) > 0 \). After (4.3.48), \( s_i \in S_{BW}^{W} \). Therefore, \( S_{SM}^{B} \subseteq S_{BW}^{W} \).

- Assume that \( s_i \in S_{BW}^{W} \). After (4.3.48), we have \( (a_{ef}^i - F) + (a_{ep}^i - a_{ep}^i) > 0 \), and hence \( P + a_{ef}^i - a_{ep}^i > P + F - a_{ep}^i \). Since \( F + P > 0 \), we have \( \frac{P + a_{ef}^i - a_{ep}^i}{F + P} > \frac{P + F - a_{ep}^i}{F + P} \). Then, after (4.3.54), \( s_i \in S_{SM}^{B} \). Therefore, \( S_{BW}^{W} \subseteq S_{SM}^{B} \).

In summary, we have proved that \( S_{SM}^{B} = S_{BW}^{W} \).

Similarly, we can prove \( S_{SM}^{F} = S_{FW}^{W} \) and \( S_{SM}^{A} = S_{AW}^{W} \).

14. Sokal

As stated in Table 4.1, formula Sokal is defined as follows.

\[ R_{S}(s_i) = \frac{2(a_{ef}^i + a_{np}^i)}{2(a_{ef}^i + a_{np}^i) + a_{nf}^i + a_{ep}^i} \]

It follows from Lemmas 4.3.1 and 4.3.2 that \( R_{S}(s_i) = \frac{2P + 2a_{ef}^i - 2a_{ep}^i}{2P + F + a_{ef}^i - a_{ep}^i} \) and \( R_{S}(s_f) = \frac{2P + 2F - 2a_{ep}^i}{2P + 2F - a_{ep}^i} \). Then, after Definition 4.2.1, we have

\[ S_{SM}^{B} = \{ s_i \frac{2P + 2a_{ef}^i - 2a_{ep}^i}{2P + F + a_{ef}^i - a_{ep}^i} > \frac{2P + 2F - 2a_{ep}^i}{2P + 2F - a_{ep}^i}, 1 \leq i \leq n \} \]  \hspace{1cm} (4.3.57)

\[ S_{SM}^{F} = \{ s_i \frac{2P + 2a_{ef}^i - 2a_{ep}^i}{2P + F + a_{ef}^i - a_{ep}^i} = \frac{2P + 2F - 2a_{ep}^i}{2P + 2F - a_{ep}^i}, 1 \leq i \leq n \} \]  \hspace{1cm} (4.3.58)

\[ S_{SM}^{A} = \{ s_i \frac{2P + 2a_{ef}^i - 2a_{ep}^i}{2P + F + a_{ef}^i - a_{ep}^i} < \frac{2P + 2F - 2a_{ep}^i}{2P + 2F - a_{ep}^i}, 1 \leq i \leq n \} \]  \hspace{1cm} (4.3.59)
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

We are going to prove that \( S^S_B \), \( S^S_F \) and \( S^S_A \) are equal to the above sets \( S^{W2}_B \) in (4.3.48), \( S^{W2}_F \) in (4.3.49) and \( S^{W2}_A \) in (4.3.50), respectively.

First, we will prove \( S^S_B = S^{W2}_B \).

- Assume that \( s_i \in S^S_B \). After (4.3.57), we have
  \[
  \frac{2P+2a^i_{ef}−2a^i_{ep}}{2P+P+α_f^i−α_{ep}^i} > \frac{2P+2F−2a^i_{ep}}{2P+2F−α_{ep}^i}.
  \]
  Since \( (2P+F+α_{ef}^i−α_{ep}^i) > 0 \) and \( (2P+2F−α_{ep}^i) > 0 \) after Lemma 4.3.1, we have
  \[
  (2P+2a^i_{ef}−2a^i_{ep})(2P+2F−α_{ep}^i) > (2P+2F−2a^i_{ep})(2P+F+α^i_{ef}−α_{ep}^i).
  \]
  After simplification, we have \( (F+P)(α^i_{ef}−F+α^i_{ep}−α_{ef}^i) > 0 \). Since \( F+P > 0 \), we have
  \[
  (α^i_{ef}−F+(α^i_{ep}−α_{ef}^i) > 0.
  \]
  Therefore, \( S^S_B \subseteq S^{W2}_B \).

- Assume that \( s_i \in S^{W2}_B \). After (4.48), we have \( (α^i_{ef}−F+(α^i_{ep}−α_{ef}^i) > 0 \), which implies
  \[
  2(F+P)(α^i_{ef}−F+α^i_{ep}−α_{ef}^i) > 0 \] because \( F+P > 0 \). It can be proved that
  \[
  2(F+P)(α^i_{ef}−F+α^i_{ep}−α_{ef}^i) = (2P+2a^i_{ef}−2a^i_{ep})(2P+2F−α_{ep}^i) − (2P+2F−2a^i_{ep})(2P+F+α^i_{ef}−α_{ep}^i)
  \]
  Thus, we have
  \[
  (2P+2a^i_{ef}−2a^i_{ep})(2P+2F−α_{ep}^i) > (2P+2F−2a^i_{ep})(2P+F+α^i_{ef}−α_{ep}^i)
  \]
  It follows from Lemma 4.3.1 that \( (2P+F+α_{ef}^i−α_{ep}^i) > 0 \) and \( (2P+2F−α_{ep}^i) > 0 \). Thus, we have
  \[
  \frac{2P+2a^i_{ef}−2a^i_{ep}}{2P+P+α_f^i−α_{ep}^i} > \frac{2P+2F−2a^i_{ep}}{2P+2F−α_{ep}^i}.
  \]
  After (4.3.57), \( s_i \in S^S_B \). Therefore, \( S^{W2}_B \subseteq S^S_B \).

In summary, we have proved that \( S^S_B = S^{W2}_B \).

Similarly, we can prove \( S^S_F = S^{W2}_F \) and \( S^S_A = S^{W2}_A \).

15. Rogers & Tanimoto

As stated in Table 4.1, formula Rogers & Tanimoto is defined as follows.

\[
R_{RT}(s_i) = \frac{α^i_{ef} + α^i_{np}}{α^i_{ef} + α^i_{np} + 2(α^i_{ef} + α^i_{np})}.
\]

It follows from Lemmas 4.3.1 and 4.3.2 that \( R_{RT}(s_i) = \frac{P+α^i_{ef}−α^i_{ep}}{2P+F−α_{ef}^i+α_{ep}^i} \) and \( R_{RT}(s_f) = \frac{F+P−α^i_{ep}}{F+P+α^i_{ep}} \).

Then, after Definition 4.2.1, we have
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

\[ S_{RT}^B = \{ s_i \mid \frac{P + a_{ef}^i - a_{ep}^i}{2F + P - a_{ef}^i + a_{ep}^i} > \frac{P - a_{ep}^i}{F + P + a_{ep}^i}, 1 \leq i \leq n \} \quad (4.3.60) \]

\[ S_{RT}^F = \{ s_i \mid \frac{P + a_{ef}^i - a_{ep}^i}{2F + P - a_{ef}^i + a_{ep}^i} = \frac{F - P - a_{ep}^i}{F + P + a_{ep}^i}, 1 \leq i \leq n \} \quad (4.3.61) \]

\[ S_{RT}^A = \{ s_i \mid \frac{P + a_{ef}^i - a_{ep}^i}{2F + P - a_{ef}^i + a_{ep}^i} < \frac{P - a_{ep}^i}{F + P + a_{ep}^i}, 1 \leq i \leq n \} \quad (4.3.62) \]

We are going to prove that \( S_{RT}^B \), \( S_{RT}^F \) and \( S_{RT}^A \) are equal to the above sets \( S_B^{W2} \) in (4.3.48), \( S_F^{W2} \) in (4.3.49) and \( S_A^{W2} \) in (4.3.50), respectively.

First, we will prove \( S_{RT}^B = S_B^{W2} \).

- Assume that \( s_i \in S_{RT}^B \). After (4.3.60), we have \( \frac{P + a_{ef}^i - a_{ep}^i}{2F + P - a_{ef}^i + a_{ep}^i} > \frac{P - a_{ep}^i}{F + P + a_{ep}^i} \).
  Since \( (2F + P - a_{ef}^i + a_{ep}^i) > 0 \) and \( (F + P + a_{ep}^i) > 0 \) after Lemma 4.3.1, we have \( (P + a_{ef}^i - a_{ep}^i)(F + P + a_{ep}^i) > (F + P - a_{ep}^i)(2F + P - a_{ef}^i + a_{ep}^i) \). After simplification, we have \( (F + P)(a_{ef}^i - F + a_{ef}^i - a_{ep}^i) > 0 \) because \( F + P > 0 \). It can be proved that

  \[
  (F + P)(a_{ef}^i - F + a_{ep}^i > a_{ep}^i) = (P + a_{ef}^i - a_{ep}^i)(F + P + a_{ep}^i) - (F + P - a_{ep}^i)(2F + P - a_{ef}^i + a_{ep}^i)
  \]

  Thus, we have

  \[
  (P + a_{ef}^i - a_{ep}^i)(F + P + a_{ep}^i) > (F + P - a_{ep}^i)(2F + P - a_{ef}^i + a_{ep}^i)
  \]

  Since \( (2F + P - a_{ef}^i + a_{ep}^i) > 0 \) and \( (F + P + a_{ep}^i) > 0 \) after Lemma 4.3.1, we have \( \frac{P + a_{ef}^i - a_{ep}^i}{2F + P - a_{ef}^i + a_{ep}^i} > \frac{P - a_{ep}^i}{F + P + a_{ep}^i} \). After (4.3.60), \( s_i \in S_{RT}^B \). Therefore, \( S_B^{W2} \subseteq S_B^{RT} \).

In summary, we have proved that \( S_{RT}^B = S_B^{W2} \).

Similarly, we can prove \( S_{RT}^F = S_F^{W2} \) and \( S_{RT}^A = S_A^{W2} \).

16. Hamming etc.

As stated in Table 4.1, Hamming etc. is defined as follows.

\[ R_{HM}(s_i) = a_{ef}^i + a_{np}^i \]
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

After Lemma 4.3.1, 4.3.2 and Definition 4.2.1, we have

\[ S_{HM}^B = \{ s_i | a_{ef}^i + P - a_{ep}^i > F + P - a_{ep}^f, 1 \leq i \leq n \} \]  
(4.3.63)

\[ S_{HM}^F = \{ s_i | a_{ef}^i + P - a_{ep}^i = F + P - a_{ep}^f, 1 \leq i \leq n \} \]  
(4.3.64)

\[ S_{HM}^A = \{ s_i | a_{ef}^i + P - a_{ep}^i < F + P - a_{ep}^f, 1 \leq i \leq n \} \]  
(4.3.65)

It is obvious that the above sets defined in (4.3.63), (4.3.64) and (4.3.65) are equal to \( S_{W2}^B \) in (4.3.48), \( S_{W2}^F \) in (4.3.49) and \( S_{W2}^A \) in (4.3.50), respectively.

17. **Euclid**

As stated in Table 4.1, formula Euclid is defined as follows.

\[ R_E(s_i) = \sqrt{a_{ef}^i + a_{ep}^i} \]

After Lemma 4.3.1, 4.3.2 and Definition 4.2.1, we have

\[ S_{E}^B = \{ s_i | \sqrt{a_{ef}^i + P - a_{ep}^i} > \sqrt{F + P - a_{ep}^f}, 1 \leq i \leq n \} \]  
(4.3.66)

\[ S_{E}^F = \{ s_i | \sqrt{a_{ef}^i + P - a_{ep}^i} = \sqrt{F + P - a_{ep}^f}, 1 \leq i \leq n \} \]  
(4.3.67)

\[ S_{E}^A = \{ s_i | \sqrt{a_{ef}^i + P - a_{ep}^i} < \sqrt{F + P - a_{ep}^f}, 1 \leq i \leq n \} \]  
(4.3.68)

Since \( a_{ef}^i + P - a_{ep}^i \geq 0 \) and \( F + P - a_{ep}^f > 0 \) after Lemma 4.3.1, obviously,

\[ \sqrt{a_{ef}^i + P - a_{ep}^i} \Theta \sqrt{F + P - a_{ep}^f} \]  
if and only if \( (a_{ef}^i + P - a_{ep}^i) \Theta (F + P - a_{ep}^f) \)

where “\( \Theta \)” is “\(<\)”, “\(=\)” or “\(>\)”. As a consequence, sets defined in (4.3.66), (4.3.67) and (4.3.68) are equal to sets in (4.3.63), (4.3.64) and (4.3.65), respectively. Therefore, \( S_E^B \), \( S_E^F \) and \( S_E^A \) are equal to \( S_{W2}^B \), \( S_{W2}^F \) and \( S_{W2}^A \), respectively.

18. **Wong1**

As stated in Table 4.1, formula Wong1 is defined as follows.

\[ R_W^1(s_i) = a_{ef}^i \]

It follows from Lemma 4.3.2 that \( R_W^1(s_f) = F \). Then, after Definition 4.2.1, we have

\[ S_{W1}^B = \{ s_i | a_{ef}^i > F, 1 \leq i \leq n \} \]  
(4.3.69)

\[ S_{W1}^F = \{ s_i | a_{ef}^i = F, 1 \leq i \leq n \} \]  
(4.3.70)

\[ S_{W1}^A = \{ s_i | a_{ef}^i < F, 1 \leq i \leq n \} \]  
(4.3.71)
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Since $a_{ef}^i \leq F$ after Lemma 4.3.1, we have $S_{B}^{W1} = \emptyset$.

19. Russell & Rao

As stated in Table 4.1, formula Russell & Rao is defined as follows.

$$R_{RR}(s_i) = \frac{a_{ef}^i}{a_{ef}^i + a_{nf}^i + a_{ep}^i + a_{np}^i}$$

It follows from Lemmas 4.3.1 and 4.3.2 that $R_{RR}(s_i) = \frac{a_{ef}^i}{F + P}$ and $R_{RR}(s_f) = \frac{F}{F + P}$. Then, after Definition 4.2.1, we have

$$S_{BB}^{RR} = \{ s_i | a_{ef}^i > \frac{F}{F + P}, 1 \leq i \leq n \}$$  \hspace{1cm} (4.3.72)

$$S_{BF}^{RR} = \{ s_i | a_{ef}^i = \frac{F}{F + P}, 1 \leq i \leq n \}$$  \hspace{1cm} (4.3.73)

$$S_{BA}^{RR} = \{ s_i | a_{ef}^i < \frac{F}{F + P}, 1 \leq i \leq n \}$$  \hspace{1cm} (4.3.74)

Since $F + P > 0$, obviously, $(\frac{a_{ef}^i}{F + P}, \frac{F}{F + P})$ if and only if $(a_{ef}^i, F)$, where “Θ” is <, = or >. As a consequence, sets defined in (4.3.72), (4.3.73) and (4.3.74) are equal to sets defined in (4.3.69), (4.3.70) and (4.3.71), respectively.

20. Binary

As stated in Table 4.1, formula Binary is defined as follows.

$$R_B(s_i) = \begin{cases} 0 & \text{if } a_{ef}^i < F \\ 1 & \text{if } a_{ef}^i = F \end{cases}$$

It follows from Lemma 4.3.2 that $R_B(s_f) = 1$. Then, after Definition 4.2.1, we have

$$S_B^B = \{ s_i | (a_{ef}^i < F \text{ and } 0 > 1) \text{ or } (a_{ef}^i = F \text{ and } 1 > 1), 1 \leq i \leq n \}$$  \hspace{1cm} (4.3.75)

$$S_B^F = \{ s_i | (a_{ef}^i < F \text{ and } 0 = 1) \text{ or } (a_{ef}^i = F \text{ and } 1 = 1), 1 \leq i \leq n \}$$  \hspace{1cm} (4.3.76)

$$S_B^A = \{ s_i | (a_{ef}^i < F \text{ and } 0 < 1) \text{ or } (a_{ef}^i = F \text{ and } 1 < 1), 1 \leq i \leq n \}$$  \hspace{1cm} (4.3.77)

We are going to prove that $S_B^B$, $S_B^F$ and $S_B^A$ are equal to sets $S_{B}^{W1}$ in (4.3.69), $S_{B}^{W1}$ in (4.3.70) and $S_{A}^{W1}$ in (4.3.71), respectively.
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

First, we will prove $S_B^B = S_B^{W_1}$. It is obvious that neither $(0>1)$ nor $(1>1)$ is possible. Thus, $S_B^B = \emptyset = S_B^{W_1}$.

Secondly, we will prove $S_F^B = S_F^{W_1}$. $S_F^B$ defined in (4.3.76) can be re-written as:

$$S_F^B = \{ s_i | a_{ef}^i < F \text{ and } 0=1, 1 \leq i \leq n \} \cup \{ s_i | a_{ef}^i = F \text{ and } 1=1, 1 \leq i \leq n \}$$

Obviously, $(0=1)$ is false and $(a_{ef}^i = F \text{ and } 1=1)$ is logically equivalent to $(a_{ef}^i = F)$. Thus, $S_F^B$ becomes $\{ s_i | a_{ef}^i = F, 1 \leq i \leq n \} = S_F^{W_1}$.

Similarly, we can prove that $S_A^B = \{ s_i | a_{ef}^i < F, 1 \leq i \leq n \} = S_A^{W_1}$.

In conclusion, $S_F^B$ in (4.3.75), $S_F^B$ in (4.3.76) and $S_A^B$ in (4.3.77) are equal to sets defined in (4.3.69), (4.3.70) and (4.3.71), respectively.

21. Scott

As stated in Table 4.1, formula Scott is defined as follows.

$$R_{SC}(s_i) = \frac{4a_{ef}^i a_{np}^i - 4a_{ef}^i a_{ep}^i - (a_{ef}^i - a_{ep}^i)^2}{(2a_{ef}^i + a_{np}^i + a_{ep}^i)(2a_{np}^i + a_{nfp}^i + a_{ep}^i)}$$

From Definition 4.2.1 and Lemmas 4.3.1 and 4.3.2, after simplification, we have

$$S_B^{SC} = \{ s_i | -F^2 + 4a_{ef}^i P + 2Fa_{ef}^i - 2Fa_{ep}^i - (a_{ef}^i + a_{ep}^i)^2 > \frac{4PF - 4Fa_{ef}^i - (a_{ef}^i)^2}{(2F + a_{ep}^i)(2P - a_{ep}^i)}, 1 \leq i \leq n \}$$

(4.3.78)

$$S_F^{SC} = \{ s_i | -F^2 + 4a_{ef}^i P + 2Fa_{ef}^i - 2Fa_{ep}^i - (a_{ef}^i + a_{ep}^i)^2 > \frac{4PF - 4Fa_{ef}^i - (a_{ef}^i)^2}{(2F + a_{ep}^i)(2P - a_{ep}^i)}, 1 \leq i \leq n \}$$

(4.3.79)

$$S_A^{SC} = \{ s_i | -F^2 + 4a_{ef}^i P + 2Fa_{ef}^i - 2Fa_{ep}^i - (a_{ef}^i + a_{ep}^i)^2 > \frac{4PF - 4Fa_{ep}^i - (a_{ef}^i)^2}{(2F + a_{np}^i)(2P - a_{ep}^i)}, 1 \leq i \leq n \}$$

(4.3.80)

22. Rogot1

As stated in Table 4.1, formula Rogot1 is defined as follows.

$$R_{RO}(s_i) = \frac{1}{2} \left( \frac{a_{ef}^i}{2a_{ef}^i + a_{np}^i + a_{ep}^i} + \frac{a_{np}^i}{2a_{np}^i + a_{nfp}^i + a_{ep}^i} \right)$$
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

It follows from Lemmas 4.3.1 and 4.3.2 that \( R_{RO}(s_f) = \frac{1}{2} \left( \frac{F - a_{ep}^f}{2F + a_{ep}^f} + \frac{P - a_{ep}^f}{2P - a_{ep}^f} \right) \) and \( R_{RO}(s_i) = \frac{1}{2} \left( \frac{a_{ef}^i}{2a_{ef}^i + F - a_{ep}^i + a_{ep}^i} + \frac{P - a_{ep}^i}{2(P - a_{ep}^i) + F - a_{ef}^i + a_{ep}^i} \right) \). Then, after Definition 4.2.1, we have

\[
S_{RO}^B = \left\{ s_i \left| \frac{a_{ef}^i}{2a_{ef}^i + F - a_{ep}^i + a_{ep}^i} + \frac{P - a_{ep}^i}{2(P - a_{ep}^i) + F - a_{ef}^i + a_{ep}^i} \right| \geq \frac{1}{2} \left( \frac{F - a_{ep}^f}{2F + a_{ep}^f} + \frac{P - a_{ep}^f}{2P - a_{ep}^f} \right) \right\}
\]

(4.3.81)

\[
S_{RO}^F = \left\{ s_i \left| \frac{a_{ef}^i}{2a_{ef}^i + F - a_{ep}^i + a_{ep}^i} + \frac{P - a_{ep}^i}{2(P - a_{ep}^i) + F - a_{ef}^i + a_{ep}^i} \right| \geq \frac{1}{2} \left( \frac{F - a_{ep}^f}{2F + a_{ep}^f} + \frac{P - a_{ep}^f}{2P - a_{ep}^f} \right) \right\}
\]

(4.3.82)

\[
S_{RO}^A = \left\{ s_i \left| \frac{a_{ef}^i}{2a_{ef}^i + F - a_{ep}^i + a_{ep}^i} + \frac{P - a_{ep}^i}{2(P - a_{ep}^i) + F - a_{ef}^i + a_{ep}^i} \right| \geq \frac{1}{2} \left( \frac{F - a_{ep}^f}{2F + a_{ep}^f} + \frac{P - a_{ep}^f}{2P - a_{ep}^f} \right) \right\}
\]

(4.3.83)

We are going to prove that the sets \( S_{RO}^B \), \( S_{RO}^F \) and \( S_{RO}^A \) are equal to \( S_{SC}^B \) in (4.3.78), \( S_{SC}^F \) in (4.3.79) and \( S_{SC}^A \) in (4.3.80), respectively.

First, we will prove \( S_{RO}^B = S_{SC}^B \).

- Assume that \( s_i \in S_{RO}^B \). After (4.3.81), we have

\[
\frac{1}{2} \left( \frac{a_{ef}^i}{2a_{ef}^i + F - a_{ep}^i + a_{ep}^i} + \frac{P - a_{ep}^i}{2(P - a_{ep}^i) + F - a_{ef}^i + a_{ep}^i} \right) \geq \frac{1}{2} \left( \frac{F - a_{ep}^f}{2F + a_{ep}^f} + \frac{P - a_{ep}^f}{2P - a_{ep}^f} \right)
\]

which implies

\[
\frac{a_{ef}^i}{2a_{ef}^i + F - a_{ep}^i + a_{ep}^i} + \frac{P - a_{ep}^i}{2(P - a_{ep}^i) + F - a_{ef}^i + a_{ep}^i} \geq \frac{1}{2} \left( \frac{F - a_{ep}^f}{2F + a_{ep}^f} + \frac{P - a_{ep}^f}{2P - a_{ep}^f} \right)
\]

After re-arranging the terms, we have

\[
\frac{-F^2 + 4a_{ef}^iF + 2F0_{ef}^i - 2Fa_{ep}^i - (a_{ep}^i + a_{ef}^i)^2}{(F + 2P - a_{ep}^i - a_{ef}^i)(F + a_{ep}^i + a_{ep}^i)} \geq \frac{4P - 4Fa_{ep}^i - (a_{ep}^i)^2}{(2F + a_{ep}^i)(2P - a_{ep}^i)}
\]

Thus, \( s_i \in S_{SC}^B \) after (4.3.78). Therefore, \( S_{RO}^B \subseteq S_{SC}^B \).

- Assume that \( s_i \in S_{SC}^B \). After (4.3.78), we have

\[
\frac{-F^2 + 4a_{ef}^iF + 2F0_{ef}^i - 2Fa_{ep}^i - (a_{ep}^i + a_{ef}^i)^2}{(F + 2P - a_{ep}^i - a_{ef}^i)(F + a_{ep}^i + a_{ep}^i)} \geq \frac{4P - 4Fa_{ep}^i - (a_{ep}^i)^2}{(2F + a_{ep}^i)(2P - a_{ep}^i)}
\]
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

In summary, we have proved that $S$. As stated in Table 4.1, formula Kulczynski2 is defined as follows. Similarly, we can prove $S$. We are going to prove that the above sets $K$ and $Y$ follow from Lemmas 4.3.1 and 4.3.2 that $R$. It follows from (4.3.81) that $s_i \in S_{RO}$. Therefore, $S_{SC} \subseteq S_{RO}$. In summary, we have proved that $S_{RO} = S_{SC}$. Similarly, we can prove $S_{F} = S_{F}^{SC}$ and $S_{A} = S_{A}^{SC}$.

23. Kulczynski2

As stated in Table 4.1, formula Kulczynski2 is defined as follows.

$$R_{K2}(s_i) = \frac{1}{2} \left( \frac{a_{ef}^{i}}{a_{ef}^{i} + a_{n}^{i}} + \frac{a_{f}^{i}}{a_{ef}^{i} + a_{ep}^{i}} \right)$$

It follows from Lemmas 4.3.1 and 4.3.2 that $R_{K2}(s_i) = \frac{1}{2} \left( \frac{a_{ef}^{i}}{a_{ef}^{i} + a_{n}^{i}} + \frac{a_{f}^{i}}{a_{ef}^{i} + a_{ep}^{i}} \right)$ and $R_{K2}(s_f) = \frac{1}{2} (1 + \frac{F}{F + a_{ep}^{i}})$. Then, after Definition 4.2.1, we have

$$S_{K2}^{B} = \{ s_i | \frac{1}{2} (\frac{a_{ef}^{i}}{a_{ef}^{i} + a_{n}^{i}} + \frac{a_{f}^{i}}{a_{ef}^{i} + a_{ep}^{i}}) > \frac{1}{2} (1 + \frac{F}{F + a_{ep}^{i}}), 1 \leq i \leq n \} \quad (4.3.84)$$

$$S_{K2}^{F} = \{ s_i | \frac{1}{2} (\frac{a_{ef}^{i}}{a_{ef}^{i} + a_{n}^{i}} + \frac{a_{f}^{i}}{a_{ef}^{i} + a_{ep}^{i}}) = \frac{1}{2} (1 + \frac{F}{F + a_{ep}^{i}}), 1 \leq i \leq n \} \quad (4.3.85)$$

$$S_{K2}^{A} = \{ s_i | \frac{1}{2} (\frac{a_{ef}^{i}}{a_{ef}^{i} + a_{n}^{i}} + \frac{a_{f}^{i}}{a_{ef}^{i} + a_{ep}^{i}}) < \frac{1}{2} (1 + \frac{F}{F + a_{ep}^{i}}), 1 \leq i \leq n \} \quad (4.3.86)$$

We are going to prove that the above sets $S_{K2}^{B}$, $S_{K2}^{F}$ and $S_{K2}^{A}$ are equal to the following sets $X_{K2}$, $Y_{K2}$ and $Z_{K2}$, respectively.

$$X_{K2} = \{ s_i | a_{ef}^{i} > 0 \ and \ \frac{a_{f}^{i} F + a_{ef}^{i} a_{ep}^{i} F^{2}}{F^{2} + (F + a_{ep}^{i}) (F - a_{ef}^{i})} - \frac{a_{ef}^{i}}{a_{ef}^{i}} > 0, 1 \leq i \leq n \} \quad (4.3.87)$$

$$Y_{K2} = \{ s_i | a_{ef}^{i} > 0 \ and \ \frac{a_{f}^{i} F + a_{ef}^{i} a_{ep}^{i} F^{2}}{F^{2} + (F + a_{ep}^{i}) (F - a_{ef}^{i})} - \frac{a_{ef}^{i}}{a_{ef}^{i}} = 0, 1 \leq i \leq n \} \quad (4.3.88)$$
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

\[ Z^K2 = \{ s_i | (a^i_{ef} = 0) \text{ or } (a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}F + a^i_{ef}a^i_{ep} - F^2}{F^2 + (F + a^i_{cp})(F - a^i_{ef})} - \frac{a^i_{ep}}{a^i_{ef}} < 0), 1 \leq i \leq n \} \quad (4.3.89) \]

First, we will prove \( S^K2_B = X^K2 \). For any \( s_i \), we have either \( (a^i_{ef} = 0) \) or \( (a^i_{ef} > 0) \). Therefore, \( S^K2_B \) defined in (4.3.84) can be re-written as

\[ S^K2_B = \{ s_i | a^i_{ef} = 0 \} \cup \{ s_i | a^i_{ef} > 0 \text{ and } \frac{1}{2} \left( \frac{a^i_{ef}}{F} + \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} \right) > \frac{1}{2} \left( 1 + \frac{F}{F + a^i_{cp}} \right), 1 \leq i \leq n \} \]

Consider the case that \( (a^i_{ef} = 0) \). Since \( F > 0 \) and \( F + a^i_{cp} > 0 \) after Lemma 4.3.1, we have

\[ \frac{1}{2} \left( \frac{a^i_{ef}}{F} + \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} \right) = 0 < \frac{1}{2} \left( 1 + \frac{F}{F + a^i_{cp}} \right), \]

which is contradictory to \( \frac{1}{2} \left( \frac{a^i_{ef}}{F} + \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} \right) > \frac{1}{2} \left( 1 + \frac{F}{F + a^i_{cp}} \right) \). Thus,

\[ \{ s_i | a^i_{ef} = 0 \} \cup \{ s_i | a^i_{ef} > 0 \text{ and } \frac{1}{2} \left( \frac{a^i_{ef}}{F} + \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} \right) > \frac{1}{2} \left( 1 + \frac{F}{F + a^i_{cp}} \right), 1 \leq i \leq n \} = \emptyset \]

Hence, we have

\[ S^K2_B = \{ s_i | a^i_{ef} > 0 \text{ and } \frac{1}{2} \left( \frac{a^i_{ef}}{F} + \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} \right) > \frac{1}{2} \left( 1 + \frac{F}{F + a^i_{cp}} \right), 1 \leq i \leq n \} \quad (4.3.90) \]

- Assume that \( s_i \in S^K2_B \). After (4.3.90), we have \( (a^i_{ef} > 0 \text{ and } \frac{1}{2} \left( \frac{a^i_{ef}}{F} + \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} \right) > \frac{1}{2} \left( 1 + \frac{F}{F + a^i_{cp}} \right)) \).

By simplification and re-arrangement of the terms, \( \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} > \frac{F(a^i_{ef} + a^i_{ep})}{F(F + a^i_{cp})} \) implies \( \frac{a^i_{ef} + a^i_{ep}}{a^i_{ef}} > \frac{F}{a^i_{cp}} \). It follows from \( F > 0 \) and Lemma 4.3.1 that \( a^i_{ef} > 0 \), \( F(F + a^i_{cp}) > 0 \) and \( F(2F + a^i_{cp}) - a^i_{ef}(F + a^i_{cp}) > 0 \). Since \( a^i_{ef} > 0 \), we have \( \frac{a^i_{ef} + a^i_{ep}}{a^i_{ef}} > \frac{F}{a^i_{cp}} \) implies \( a^i_{ep} > 0 \). After (4.3.87), \( s_i \in X^K2 \). Therefore, \( S^K2_B \subseteq X^K2 \).

- Assume that \( s_i \in X^K2 \). After (4.3.87), we have \( (a^i_{ef} > 0 \text{ and } \frac{a^i_{ef} + a^i_{ep}}{a^i_{ef}} > \frac{F}{a^i_{cp}}) \).

Since \( a^i_{ef} > 0 \) and \( a^i_{ef} + a^i_{ep} > 0 \), \( F(F + a^i_{ep}) > 0 \) and \( F(2F + a^i_{cp}) - a^i_{ef}(F + a^i_{cp}) > 0 \) after Lemma 4.3.1, \( \frac{a^i_{ef}}{a^i_{ep}} < \frac{F}{F(F + a^i_{cp}) - a^i_{ef}(F + a^i_{cp})} - 1 \) implies \( \frac{1}{2} \left( \frac{a^i_{ef}}{F} + \frac{a^i_{ef}}{a^i_{ef} + a^i_{ep}} \right) < \frac{1}{2} \left( 1 + \frac{F}{F + a^i_{cp}} \right) \).

Then, we have \( s_i \in S^K2_B \) after (4.3.90). Therefore, \( X^K2 \subseteq S^K2_B \).

In summary, we have proved that \( S^K2_B = X^K2 \).
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Similarly, we can prove that \( S^K F = Y^K 2 \).

Next, we are going to prove \( S^K F = Z^K 2 \). \( S^K F \) in (4.3.86) can be re-written as follows:

\[
S^K F = \{ s_i | a_{ef}^i = 0 \text{ and } \frac{a_{ef}^i}{F} + \frac{a_{ef}^i}{a_{ef} + a_{ep}} < \frac{1}{2} (1 + \frac{F}{F + a_{ep}}), 1 \leq i \leq n \} \\
\cup \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i}{F} + \frac{a_{ef}^i}{a_{ef} + a_{ep}} < \frac{1}{2} (1 + \frac{F}{F + a_{ep}}), 1 \leq i \leq n \}
\]

Consider the case \( (a_{ef}^i = 0) \), which implies \( \frac{a_{ef}^i}{F} + \frac{a_{ef}^i}{a_{ef} + a_{ep}} = 0 < \frac{1}{2} (1 + \frac{F}{F + a_{ep}}) \). Thus, \( (a_{ef}^i = 0 \text{ and } \frac{a_{ef}^i}{F} + \frac{a_{ef}^i}{a_{ef} + a_{ep}} < \frac{1}{2} (1 + \frac{F}{F + a_{ep}}) \) is logically equivalent to \( (a_{ef}^i = 0) \). Therefore, \( S^K F \) becomes

\[
\{ s_i | a_{ef}^i = 0, 1 \leq i \leq n \} \cup \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i}{F} + \frac{a_{ef}^i}{a_{ef} + a_{ep}} < \frac{1}{2} (1 + \frac{F}{F + a_{ep}}), 1 \leq i \leq n \}
\]

Similar to the proof of \( S^K B = X^K 2 \), we can prove

\[
\{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i}{F} + \frac{a_{ef}^i}{a_{ef} + a_{ep}} < \frac{1}{2} (1 + \frac{F}{F + a_{ep}}), 1 \leq i \leq n \}
\]

\[
= \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i F + a_{ef}^i a_{ef} - F^2}{F^2 + (F + a_{ep})(F - a_{ef})} \frac{a_{ep}^i}{a_{ef}^i} < 0, 1 \leq i \leq n \}
\]

Therefore, we have

\[
S^K A = \{ s_i | (a_{ef}^i = 0) \text{ or } (a_{ef}^i > 0 \text{ and } \frac{a_{ef} F + a_{ef} a_{ef} - F^2}{F^2 + (F + a_{ep})(F - a_{ef})} \frac{a_{ep}^i}{a_{ef}^i} < 0), 1 \leq i \leq n \} = Z^K 2
\]

In conclusion, we have proved that \( S^K F = X^K 2 \), \( S^K F = Y^K 2 \) and \( S^K A = Z^K 2 \).

24. M2

As stated in Table 4.1, formula M2 is defined as follows.

\[
R_{M2}(s_i) = \frac{a_{ef}^i}{a_{ef}^i + a_{en}^i + 2(a_{ef}^i + a_{ep}^i)}
\]

It follows from Lemmas 4.3.1 and 4.3.2 that \( R_{M2}(s_i) = \frac{a_{ef}^i}{2F + F - a_{ef}^i + a_{ep}^i} \) and \( R_{M2}(s_F) = \frac{F}{F + F + a_{ep}^i} \).
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Then, after Definition 4.2.1, we have

\[ S_B^{M2} = \{ s_i | \frac{a^i_{ef}}{2F+P-a^i_{ef} + a^i_{ep}} > \frac{F}{F+P+a^i_{ep}}, 1 \leq i \leq n \} \] \tag{4.3.91}

\[ S_F^{M2} = \{ s_i | \frac{a^i_{ef}}{2F+P-a^i_{ef} + a^i_{ep}} = \frac{F}{F+P+a^i_{ep}}, 1 \leq i \leq n \} \] \tag{4.3.92}

\[ S_A^{M2} = \{ s_i | \frac{a^i_{ef}}{2F+P-a^i_{ef} + a^i_{ep}} < \frac{F}{F+P+a^i_{ep}}, 1 \leq i \leq n \} \] \tag{4.3.93}

We are going to prove that the above sets \( S_B^{M2}, S_F^{M2} \) and \( S_A^{M2} \) are equal to the following sets \( X^{M2}, Y^{M2} \) and \( Z^{M2} \), respectively.

\[ X^{M2} = \{ s_i | a^i_{ef} > 0 \text{ and } \frac{P+a^i_{ep}}{F} - \frac{2F+P}{a^i_{ef}} + 2 - \frac{a^i_{ep}}{a^i_{ef}} > 0, 1 \leq i \leq n \} \] \tag{4.3.94}

\[ Y^{M2} = \{ s_i | a^i_{ef} > 0 \text{ and } \frac{P+a^i_{ep}}{F} - \frac{2F+P}{a^i_{ef}} + 2 - \frac{a^i_{ep}}{a^i_{ef}} = 0, 1 \leq i \leq n \} \] \tag{4.3.95}

\[ Z^{M2} = \{ s_i | (a^i_{ef} = 0) \text{ or } (a^i_{ef} > 0 \text{ and } \frac{P+a^i_{ep}}{F} - \frac{2F+P}{a^i_{ef}} + 2 - \frac{a^i_{ep}}{a^i_{ef}} < 0), 1 \leq i \leq n \} \] \tag{4.3.96}

First, we will prove \( S_B^{M2} = X^{M2} \). For any \( s_i \), we have either \( a^i_{ef} = 0 \) or \( a^i_{ef} > 0 \). Therefore, \( S_B^{M2} \) defined in (4.3.91) can be re-written as

\[ S_B^{M2} = \{ s_i | a^i_{ef} = 0 \text{ and } \frac{a^i_{ef}}{2F+P-a^i_{ef} + a^i_{ep}} > \frac{F}{F+P+a^i_{ep}}, 1 \leq i \leq n \} \]

\[ \cup \{ s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{2F+P-a^i_{ef} + a^i_{ep}} > \frac{F}{F+P+a^i_{ep}}, 1 \leq i \leq n \} \]

Consider the case that \( (a^i_{ef} = 0) \). Since \( F > 0 \) and \( F + P + a^i_{ep} > 0 \) after Lemma 4.3.1, we have

\[ \frac{a^i_{ef}}{2F+P-a^i_{ef} + a^i_{ep}} = 0 < \frac{F}{F+P+a^i_{ep}} \], which is contradictory to \( \frac{a^i_{ef}}{2F+P-a^i_{ef} + a^i_{ep}} > \frac{F}{F+P+a^i_{ep}} \). Thus,

\[ \{ s_i | a^i_{ef} = 0 \text{ and } \frac{a^i_{ef}}{2F+P-a^i_{ef} + a^i_{ep}} > \frac{F}{F+P+a^i_{ep}}, 1 \leq i \leq n \} = \emptyset \]

Then, we have

\[ S_B^{M2} = \{ s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{2F+P-a^i_{ef} + a^i_{ep}} > \frac{F}{F+P+a^i_{ep}}, 1 \leq i \leq n \} \] \tag{4.3.97}

- Assume that \( s_i \in S_B^{M2} \). After (4.3.97), we have \( (a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{2F+P-a^i_{ef} + a^i_{ep}} > \frac{F}{F+P+a^i_{ep}}) \).
Since \(a_{ef}^i > 0\), \(F > 0\), \((2F + P - a_{ef}^i + a_{ep}^i) > 0\) (after Lemma 4.3.1) and \((F + P + a_{ep}^i) > 0\) (after Lemma 4.3.1), we have \(\frac{P + a_{ep}^i}{a_{ef}^i} - \frac{2F + P - a_{ef}^i + a_{ep}^i}{P} < \frac{F}{P + a_{ep}^i}\). After re-arranging the terms, we have \(\frac{P + a_{ep}^i}{a_{ef}^i} - \frac{2F + P - a_{ef}^i + a_{ep}^i}{P} > 0\). After (4.3.94), \(s_i \in X^{M^2}\). Therefore, \(S_B^{M^2} \subset X^{M^2}\).

- Assume that \(s_i \in X^{M^2}\). After (4.3.94), we have \((a_{ef}^i > 0\) and \(\frac{P + a_{ep}^i}{a_{ef}^i} - \frac{2F + P - a_{ef}^i + a_{ep}^i}{P} > 0\)). \(\frac{P + a_{ep}^i}{a_{ef}^i} - \frac{2F + P - a_{ef}^i + a_{ep}^i}{P} > 0\) can be re-written as \(\frac{2F + P - a_{ef}^i + a_{ep}^i}{a_{ef}^i} < \frac{F + P + a_{ep}^i}{a_{ef}^i}\). Since \(a_{ef}^i > 0\), \(F > 0\), \((2F + P - a_{ef}^i + a_{ep}^i) > 0\) and \((F + P + a_{ep}^i) > 0\), we have \(\frac{a_{ep}^i}{a_{ef}^i} > \frac{F}{F + P + a_{ep}^i}\). Then, we have \(s_i \in S_B^{M^2}\) after (4.3.97). Therefore, \(X^{M^2} \subset S_B^{M^2}\).

In summary, we have proved that \(S_B^{M^2} = X^{M^2}\).

Similarly, we can prove that \(S_F^{M^2} = Y^{M^2}\).

Next, we are going to prove \(S_A^{M^2} = Z^{M^2}\). \(S_A^{M^2}\) in (4.3.93) can be re-written as follows

\[
S_A^{M^2} = \left\{ s_i | a_{ef}^i = 0 \right\} \cup \left\{ s_i | a_{ef}^i > 0 \right\} = \left\{ s_i | a_{ef}^i = 0 \right\} \cup \left\{ s_i | a_{ef}^i > 0 \right\}
\]

\[
\cup \left\{ s_i | a_{ef}^i > 0 \right\} \quad \cup \left\{ s_i | a_{ef}^i > 0 \right\} = \left\{ s_i | a_{ef}^i = 0 \right\} \cup \left\{ s_i | a_{ef}^i > 0 \right\}
\]

Consider the case \((a_{ef}^i = 0\), which implies \(\frac{a_{ep}^i}{2F + P - a_{ef}^i + a_{ep}^i} = 0 < \frac{F}{F + P + a_{ep}^i}\). Thus, \((a_{ef}^i = 0\) and \(\frac{a_{ep}^i}{2F + P - a_{ef}^i + a_{ep}^i} < \frac{F}{F + P + a_{ep}^i}\) is logically equivalent to \((a_{ef}^i = 0\). Therefore, \(S_A^{M^2}\) becomes

\[
\left\{ s_i | a_{ef}^i = 0, 1 \leq i \leq n \right\} \cup \left\{ s_i | a_{ef}^i > 0 \right\} \cup \left\{ s_i | a_{ef}^i > 0 \right\} = \left\{ s_i | a_{ef}^i = 0 \right\} \cup \left\{ s_i | a_{ef}^i > 0 \right\}
\]

Similar to the proof of \(S_B^{M^2} = X^{M^2}\), we can prove

\[
\left\{ s_i | a_{ef}^i > 0 \right\} \cup \left\{ s_i | a_{ef}^i > 0 \right\} = \left\{ s_i | a_{ef}^i > 0 \right\}
\]

\[
\left\{ s_i | a_{ef}^i > 0 \right\} \cup \left\{ s_i | a_{ef}^i > 0 \right\} = \left\{ s_i | a_{ef}^i = 0 \right\} \cup \left\{ s_i | a_{ef}^i = 0 \right\}
\]

Therefore, we have

\[
S_A^{M^2} = \left\{ s_i | (a_{ef}^i = 0) \right\} \cup \left\{ s_i | (a_{ef}^i = 0) \right\} = Z^{M^2}
\]
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

In conclusion, we have proved that $S_B^M = X^M$, $S_F^M = Y^M$ and $S_A^M = Z^M$.

25. Ohicai

As stated in Table 4.1, formula Ochiai is defined as follows.

$$R_O(s_i) = \frac{a^i_{ef}}{\sqrt{(a^i_{ef} + a^i_{nf})(a^i_{ef} + a^i_{ep})}}$$

It follows from Lemmas 4.3.1 and 4.3.2 that $R_O(s_i) = \frac{a^i_{ef}}{\sqrt{F(a^i_{ef} + a^i_{ep})}}$ and $R_O(s_f) = \frac{F}{\sqrt{F(a^f_{ef} + a^f_{ep})}}$.

Then, after Definition 4.2.1, we have

$$S_B^O = \left\{ s_i \mid \frac{a^i_{ef}}{\sqrt{F(a^i_{ef} + a^i_{ep})}} > \frac{F}{\sqrt{F(a^f_{ef} + a^f_{ep})}}, 1 \leq i \leq n \right\}$$ (4.3.98)

$$S_F^O = \left\{ s_i \mid \frac{a^i_{ef}}{\sqrt{F(a^i_{ef} + a^i_{ep})}} = \frac{F}{\sqrt{F(a^f_{ef} + a^f_{ep})}}, 1 \leq i \leq n \right\}$$ (4.3.99)

$$S_A^O = \left\{ s_i \mid \frac{a^i_{ef}}{\sqrt{F(a^i_{ef} + a^i_{ep})}} < \frac{F}{\sqrt{F(a^f_{ef} + a^f_{ep})}}, 1 \leq i \leq n \right\}$$ (4.3.100)

We are going to prove that the above sets $S_B^O$, $S_F^O$ and $S_A^O$ are equal to the following sets $X^O$, $Y^O$ and $Z^O$, respectively.

$$X^O = \{ s_i \mid a^i_{ef} > 0 \text{ and } (1 + \frac{a^f_{ep}}{F})\frac{a^i_{ef}}{F} - 1 - \frac{a^i_{ep}}{a^i_{ef}} > 0, 1 \leq i \leq n \}$$ (4.3.101)

$$Y^O = \{ s_i \mid a^i_{ef} > 0 \text{ and } (1 + \frac{a^f_{ep}}{F})\frac{a^i_{ef}}{F} - 1 - \frac{a^i_{ep}}{a^i_{ef}} = 0, 1 \leq i \leq n \}$$ (4.3.102)

$$Z^O = \{ s_i \mid (a^i_{ef} = 0) \text{ or } (a^i_{ef} > 0 \text{ and } (1 + \frac{a^f_{ep}}{F})\frac{a^i_{ef}}{F} - 1 - \frac{a^i_{ep}}{a^i_{ef}} < 0), 1 \leq i \leq n \}$$ (4.3.103)

First, we will prove $S_B^O = X^O$. For any $s_i$, we have either $(a^i_{ef} = 0)$ or $(a^i_{ef} > 0)$. Therefore, $S_B^O$ defined in (4.3.98) can be re-written as

$$S_B^O = \{ s_i \mid a^i_{ef} = 0 \text{ and } \frac{a^i_{ef}}{\sqrt{F(a^i_{ef} + a^i_{ep})}} > \frac{F}{\sqrt{F(a^f_{ef} + a^f_{ep})}}, 1 \leq i \leq n \} \cup \{ s_i \mid a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{\sqrt{F(a^i_{ef} + a^i_{ep})}} > \frac{F}{\sqrt{F(a^f_{ef} + a^f_{ep})}}, 1 \leq i \leq n \}$$
Consider the case that \( a_{ef}^i = 0 \). Since \( F > 0 \) and \( F + a_{ef}^i > 0 \) after Lemma 4.3.1, we have
\[
\frac{a_{ef}^i}{\sqrt{F(a_{ef}^i + a_{ep}^i)}} = 0 < \frac{F}{\sqrt{F(F + a_{ep}^i)}} \text{, which is contradictory to } \frac{a_{ef}^i}{\sqrt{F(a_{ef}^i + a_{ep}^i)}} > \frac{F}{\sqrt{F(F + a_{ep}^i)}} \text{. Thus,}
\]
\[
\{ s_i | a_{ef}^i = 0 \text{ and } \frac{a_{ef}^i}{\sqrt{F(a_{ef}^i + a_{ep}^i)}} > \frac{F}{\sqrt{F(F + a_{ep}^i)}} 1 \leq i \leq n \} = \emptyset
\]

Then, we have
\[
S_B^O = \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i}{\sqrt{F(a_{ef}^i + a_{ep}^i)}} > \frac{F}{\sqrt{F(F + a_{ep}^i)}} 1 \leq i \leq n \} \quad (4.3.104)
\]

- Assume that \( s_i \in S_B^O \). After (4.3.104), we have \( (a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i}{\sqrt{F(a_{ef}^i + a_{ep}^i)}} > \frac{F}{\sqrt{F(F + a_{ep}^i)}} \). Since \( a_{ef}^i > 0 \) and \( F > 0 \), we have \( \frac{\sqrt{F(a_{ef}^i + a_{ep}^i)}}{(a_{ef}^i)^2} < \frac{\sqrt{F(F + a_{ep}^i)}}{F} \) after \( \frac{a_{ef}^i}{\sqrt{F(a_{ef}^i + a_{ep}^i)}} > \frac{F}{\sqrt{F(F + a_{ep}^i)}} \).

Then, we have \( \frac{F(a_{ef}^i + a_{ep}^i)}{(a_{ef}^i)^2} < \frac{F(F + a_{ep}^i)}{F^2} \). After re-arranging the terms, we have
\[
(1 + \frac{a_{ep}^i}{F}) - \frac{F}{a_{ef}^i}(1 + \frac{a_{ep}^i}{a_{ef}^i}) > 0, \text{ which implies } (1 + \frac{a_{ep}^i}{F}) - \frac{F}{a_{ef}^i}(1 + \frac{a_{ep}^i}{a_{ef}^i}) > 0 \text{ because } \frac{a_{ef}^i}{F} > 0. \text{ Then, we have } s_i \in X^O \text{ after (4.3.101). Therefore, } S_B^O \subseteq X^O.
\]

- Assume that \( s_i \in X^O \). After (4.3.101), we have \( (a_{ef}^i > 0 \text{ and } (1 + \frac{a_{ep}^i}{F}) - \frac{F}{a_{ef}^i}(1 + \frac{a_{ep}^i}{a_{ef}^i}) > 0 \). Since \( a_{ef}^i > 0 \) and \( F > 0 \), we have \( (1 + \frac{a_{ep}^i}{F}) - \frac{F}{a_{ef}^i}(1 + \frac{a_{ep}^i}{a_{ef}^i}) > 0 \) through multiplying \( \frac{F}{a_{ef}^i} \) by \( (1 + \frac{a_{ep}^i}{F}) - \frac{F}{a_{ef}^i}(1 + \frac{a_{ep}^i}{a_{ef}^i}) > 0 \). After re-arranging the terms, we have \( \frac{F(a_{ef}^i + a_{ep}^i)}{(a_{ef}^i)^2} < \frac{F(F + a_{ep}^i)}{F^2} \). Since \( a_{ef}^i > 0, F > 0, F + a_{ep}^i > 0 \) (after Lemma 4.3.1) and \( a_{ef}^i + a_{ep}^i > 0 \) (after Lemma 4.3.1), \( \frac{F(a_{ef}^i + a_{ep}^i)}{(a_{ef}^i)^2} < \frac{F(F + a_{ep}^i)}{F^2} \) implies \( \frac{a_{ef}^i}{\sqrt{F(a_{ef}^i + a_{ep}^i)}} > \frac{F}{\sqrt{F(F + a_{ep}^i)}} \). Then, we have \( s_i \in S_B^O \) after (4.3.104). Therefore, \( X^O \subseteq S_B^O \).

In summary, we have proved that \( S_B^O = X^O \).

Similarly, we can prove that \( S_F^O = Y^O \).

Next, we are going to prove \( S_A^O = Z^O \). \( S_A^O \) defined in (4.3.100) can be re-writen as follows
\[
S_A^O = \{ s_i | a_{ef}^i = 0 \text{ and } \frac{a_{ef}^i}{\sqrt{F(a_{ef}^i + a_{ep}^i)}} < \frac{F}{\sqrt{F(F + a_{ep}^i)}} 1 \leq i \leq n \}
\]
\[
\cup \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i}{\sqrt{F(a_{ef}^i + a_{ep}^i)}} < \frac{F}{\sqrt{F(F + a_{ep}^i)}} 1 \leq i \leq n \}
\]
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Consider the case \( a_{i}^{ef} = 0 \), which implies \( \frac{a_{i}^{ef}}{\sqrt{F(a_{i}^{ef}+a_{i}^{ep})}} = 0 < \frac{F}{\sqrt{F(F+a_{i}^{ep})}} \). Thus, \( a_{i}^{ef} = 0 \) and \( \frac{a_{i}^{ef}}{\sqrt{F(a_{i}^{ef}+a_{i}^{ep})}} < \frac{F}{\sqrt{F(F+a_{i}^{ep})}} \) is logically equivalent to \( a_{i}^{ef} = 0 \). Therefore, \( S_{A} \) becomes

\[
\{ s_{i} | a_{i}^{ef} = 0, 1 \leq i \leq n \} \cup \{ s_{i} | a_{i}^{ef} > 0 \text{ and } \frac{a_{i}^{ef}}{\sqrt{F(a_{i}^{ef}+a_{i}^{ep})}} < \frac{F}{\sqrt{F(F+a_{i}^{ep})}}, 1 \leq i \leq n \}
\]

Similar to the proof of \( S_{B}^{O} = X^{O} \), we can prove

\[
\{ s_{i} | a_{i}^{ef} > 0 \text{ and } \frac{a_{i}^{ef}}{\sqrt{F(a_{i}^{ef}+a_{i}^{ep})}} < \frac{F}{\sqrt{F(F+a_{i}^{ep})}}, 1 \leq i \leq n \}
\]

\[
= \{ s_{i} | a_{i}^{ef} > 0 \text{ and } (1 + \frac{a_{i}^{ep}}{F}) \frac{a_{i}^{ef}}{F} - 1 - \frac{a_{i}^{ep}}{a_{i}^{ef}} < 0, 1 \leq i \leq n \}
\]

Therefore, we have

\[
S_{A}^{O} = \{ s_{i} | (a_{i}^{ef} = 0) \text{ or } (a_{i}^{ef} > 0 \text{ and } (1 + \frac{a_{i}^{ep}}{F}) \frac{a_{i}^{ef}}{F} - 1 - \frac{a_{i}^{ep}}{a_{i}^{ef}} < 0), 1 \leq i \leq n \} = Z^{O}
\]

In conclusion, we have proved that \( S_{B}^{O} = X^{O} \), \( S_{F}^{O} = Y^{O} \) and \( S_{A}^{O} = Z^{O} \).

26. AMPLE2

As stated in Table 4.1, formula AMPLE2 is defined as follows.

\[
R_{A}(s_{i}) = \frac{a_{i}^{ef}}{a_{i}^{ef} + a_{i}^{nf}} - \frac{a_{i}^{ep}}{a_{i}^{ep} + a_{i}^{np}}
\]

It follows from Lemmas 4.3.1 and 4.3.2 that \( R_{A}(s_{i}) = \frac{a_{i}^{ef}}{F} - \frac{a_{i}^{ep}}{P} \) and \( R_{A}(s_{f}) = 1 - \frac{a_{i}^{ep}}{F} \). Then, after Definition 4.2.1, we have

\[
S_{B}^{A} = \{ s_{i} | \frac{a_{i}^{ef}}{F} > 1 - \frac{a_{i}^{ep}}{P}, 1 \leq i \leq n \} \quad (4.3.105)
\]

\[
S_{F}^{A} = \{ s_{i} | \frac{a_{i}^{ef}}{F} = 1 - \frac{a_{i}^{ep}}{P}, 1 \leq i \leq n \} \quad (4.3.106)
\]

\[
S_{A}^{A} = \{ s_{i} | \frac{a_{i}^{ef}}{F} < 1 - \frac{a_{i}^{ep}}{P}, 1 \leq i \leq n \} \quad (4.3.107)
\]
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

We are going to prove that the above sets $S_B^A$, $S_F^A$ and $S_A^A$ are equal to the following sets $X^A$, $Y^A$ and $Z^A$, respectively.

\[ X^A = \{ s_i | a^i_{ef} > 0 \text{ and } \frac{Pa^i_{ef} - PF + Fa^i_{fp}}{Fa^i_{ef}} - \frac{a^i_{ep}}{a^i_{ef}} > 0, 1 \leq i \leq n \} \quad (4.3.108) \]

\[ Y^A = \{ s_i | a^i_{ef} > 0 \text{ and } \frac{Pa^i_{ef} - PF + Fa^i_{fp}}{Fa^i_{ef}} - \frac{a^i_{ep}}{a^i_{ef}} > 0, 1 \leq i \leq n \} \quad (4.3.109) \]

\[ Z^A = \{ s_i | (a^i_{ef} = 0) \text{ or } (a^i_{ef} > 0 \text{ and } \frac{Pa^i_{ef} - PF + Fa^i_{fp}}{Fa^i_{ef}} - \frac{a^i_{ep}}{a^i_{ef}} < 0, 1 \leq i \leq n \} \quad (4.3.110) \]

First, we will prove $S_B^A = X^A$. For any $s_i$, we have either $(a^i_{ef} = 0)$ or $(a^i_{ef} > 0)$. Therefore, $S_B^A$ defined in (4.3.105) can be re-written as

\[ S_B^A = \{ s_i | a^i_{ef} = 0 \text{ and } \frac{a^i_{ef}}{F} - \frac{a^i_{ep}}{P} > 1 - \frac{a^i_{ep}}{P}, 1 \leq i \leq n \} \]

\[ \cup \{ s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{F} - \frac{a^i_{ep}}{P} > 1 - \frac{a^i_{ep}}{P}, 1 \leq i \leq n \} \]

Consider the case $(a^i_{ef} = 0)$, which implies $a^i_{ep} > 0$ because $a^i_{ef} + a^i_{ep} > 0$ after Lemma 4.3.1. Then, we have $\frac{a^i_{ef}}{P} - \frac{a^i_{ep}}{P} = -\frac{a^i_{ep}}{P} < 0$. Besides, since $a^i_{ep} \leq P$, we have $1 - \frac{a^i_{ep}}{P} \geq 0$. As a consequence, $\frac{a^i_{ef}}{P} - \frac{a^i_{ep}}{P} < 1 - \frac{a^i_{ep}}{P}$, which is contradictory to $\frac{a^i_{ef}}{P} - \frac{a^i_{ep}}{P} > 1 - \frac{a^i_{ep}}{P}$. Thus,

\[ \{ s_i | a^i_{ef} = 0 \text{ and } \frac{a^i_{ef}}{F} - \frac{a^i_{ep}}{P} > 1 - \frac{a^i_{ep}}{P}, 1 \leq i \leq n \} = \emptyset \]

Hence, we have

\[ S_B^A = \{ s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{F} - \frac{a^i_{ep}}{P} > 1 - \frac{a^i_{ep}}{P}, 1 \leq i \leq n \} \quad (4.3.111) \]

- Assume that $s_i \in S_B^A$. After (4.3.111), we have $(a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{F} - \frac{a^i_{ep}}{P} > 1 - \frac{a^i_{ep}}{P})$. Since $a^i_{ef} > 0$ and $P > 0$, we have $\frac{P - \frac{a^i_{ep}}{P}}{a^i_{ef}} > \frac{1 - \frac{a^i_{ep}}{P}}{a^i_{ef}}$ after $\frac{a^i_{ef}}{F} - \frac{a^i_{ep}}{P} > 1 - \frac{a^i_{ep}}{P}$. After simplification, we have $\frac{Pa^i_{ef} - PF + Fa^i_{fp}}{Fa^i_{ef}} - \frac{a^i_{ep}}{a^i_{ef}} > 0$. After (4.3.108), $s_i \in X^A$. Therefore, $S_B^A \subseteq X^A$.

- Assume that $s_i \in X^A$. After (4.3.108), we have $(a^i_{ef} > 0 \text{ and } \frac{Pa^i_{ef} - PF + Fa^i_{fp}}{Fa^i_{ef}} - \frac{a^i_{ep}}{a^i_{ef}} > 0)$. $\frac{Pa^i_{ef} - PF + Fa^i_{fp}}{Fa^i_{ef}} - \frac{a^i_{ep}}{a^i_{ef}} > 0$ can be re-arranged as $\frac{F}{F - \frac{a^i_{ep}}{a^i_{ef}}} > \frac{1 - \frac{a^i_{ep}}{a^i_{ef}}}{a^i_{ef}}$. Since $a^i_{ef} > 0$ and $P > 0$, we have $\frac{a^i_{fp}}{P} - \frac{a^i_{ef}}{a^i_{ef}} > \frac{a^i_{fp}}{P} - \frac{a^i_{ef}}{a^i_{ef}}$. After simplification, we have $\frac{a^i_{fp}}{P} - \frac{a^i_{ep}}{P} > 1 - \frac{a^i_{ep}}{P}$. After (4.3.111), $s_i \in S_B^A$. Therefore, $X^A \subseteq S_B^A.$
In summary, we have proved that \( S^A_B = X^A \).

Similarly, we can prove that \( S^A_F = Y^A \).

Next, we are going to prove \( S^A_A = Z^A \). \( S^A_A \) in (4.3.107) can be re-written as follows

\[
S^A_A = \{ s_i | a^i_{ef} = 0 \text{ and } \frac{a^i_{ef}}{P} < 1 - \frac{a^f_{ef}}{P}, 1 \leq i \leq n \} \\
\cup \{ s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{P} < 1 - \frac{a^f_{ef}}{P}, 1 \leq i \leq n \}
\]

Consider the case that \( (a^i_{ef} = 0) \). As shown in the above proof of \( S^A_B = X^A \), \( (a^i_{ef} = 0) \) implies \( \frac{a^i_{ef}}{P} < 1 - \frac{a^f_{ef}}{P} \). Thus, \( (a^i_{ef} = 0 \text{ and } \frac{a^i_{ef}}{P} < 1 - \frac{a^f_{ef}}{P}) \) is logically equivalent to \( (a^i_{ef} = 0) \). Therefore, \( S^A_A \) becomes

\[
\{ s_i | a^i_{ef} = 0, 1 \leq i \leq n \} \cup \{ s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{P} < 1 - \frac{a^f_{ef}}{P}, 1 \leq i \leq n \}
\]

Similar to the proof of \( S^A_B = X^A \), we can prove

\[
\{ s_i | a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}}{P} < 1 - \frac{a^f_{ef}}{P}, 1 \leq i \leq n \}
\]

\[
= \{ s_i | a^i_{ef} > 0 \text{ and } \frac{P a^i_{ef} - PF + F a^f_{ef}}{a^i_{ef}} - \frac{a^f_{ef}}{a^i_{ef}} < 0, 1 \leq i \leq n \}
\]

Therefore, we have

\[
S^A_A = \{ s_i | (a^i_{ef} = 0) \text{ or } (a^i_{ef} > 0 \text{ and } \frac{P a^i_{ef} - PF + F a^f_{ef}}{a^i_{ef}} - \frac{a^f_{ef}}{a^i_{ef}} < 0), 1 \leq i \leq n \} \}
\]

In conclusion, we have proved that \( S^A_B = X^A \), \( S^A_F = Y^A \) and \( S^A_A = Z^A \).
27. Wong3

As stated in Table 4.1, formula Wong3 is defined as \( R_{W3}(s_i) = a_{ef}^i - h \), where

\[
h = \begin{cases} 
a_{ep}^i & \text{if } a_{ep}^i \leq 2 \\
2 + 0.1(a_{ep}^i - 2) & \text{if } 2 < a_{ep}^i \leq 10 \\
2.8 + 0.001(a_{ep}^i - 10) & \text{if } a_{ep}^i > 10
\end{cases}
\]

(1) Assume that \( a_{ep}^i \leq 2 \). Then, \( R_{W3}(s_f) = F - a_{ep}^i \). After Definition 4.2.1, and re-arranging the terms, we have

\[
S_{W3}^{B} = \{s_i | a_{ep}^i \leq 2 \text{ and } (a_{ep}^i - F) + (a_{ep}^i - a_{ep}^i) > 0, 1 \leq i \leq n \} \\
\cup \{s_i | 2 < a_{ep}^i \leq 10 \text{ and } (a_{ep}^i - F) + (a_{ep}^i - 0.1a_{ep}^i) - 1.8 > 0, 1 \leq i \leq n \} \\
\cup \{s_i | a_{ep}^i > 10 \text{ and } (a_{ep}^i - F) + (a_{ep}^i - 0.001a_{ep}^i) - 2.79 > 0, 1 \leq i \leq n \} \quad (4.3.112)
\]

\[
S_{W3}^{F} = \{s_i | a_{ep}^i \leq 2 \text{ and } (a_{ep}^i - F) + (a_{ep}^i - a_{ep}^i) = 0, 1 \leq i \leq n \} \\
\cup \{s_i | 2 < a_{ep}^i \leq 10 \text{ and } (a_{ep}^i - F) + (a_{ep}^i - 0.1a_{ep}^i) - 1.8 = 0, 1 \leq i \leq n \} \\
\cup \{s_i | a_{ep}^i > 10 \text{ and } (a_{ep}^i - F) + (a_{ep}^i - 0.001a_{ep}^i) - 2.79 = 0, 1 \leq i \leq n \} \quad (4.3.113)
\]

\[
S_{W3}^{A} = \{s_i | a_{ep}^i \leq 2 \text{ and } (a_{ep}^i - F) + (a_{ep}^i - a_{ep}^i) < 0, 1 \leq i \leq n \} \\
\cup \{s_i | 2 < a_{ep}^i \leq 10 \text{ and } (a_{ep}^i - F) + (a_{ep}^i - 0.1a_{ep}^i) - 1.8 < 0, 1 \leq i \leq n \} \\
\cup \{s_i | a_{ep}^i > 10 \text{ and } (a_{ep}^i - F) + (a_{ep}^i - 0.001a_{ep}^i) - 2.79 < 0, 1 \leq i \leq n \} \quad (4.3.114)
\]

We are going to prove that the above sets in (4.3.112), (4.3.113) and (4.3.114) are equal to the following sets \( X_{W3}^1, Y_{W3}^1 \) and \( Z_{W3}^1 \), respectively.

\[
X_{W3}^1 = \{s_i | a_{ep}^i \leq 2 \text{ and } (a_{ep}^i - F) + (a_{ep}^i - a_{ep}^i) > 0, 1 \leq i \leq n \} \quad (4.3.115)
\]

\[
Y_{W3}^1 = \{s_i | a_{ep}^i \leq 2 \text{ and } (a_{ep}^i - F) + (a_{ep}^i - a_{ep}^i) = 0, 1 \leq i \leq n \} \quad (4.3.116)
\]

\[
Z_{W3}^1 = \{s_i | (a_{ep}^i - F) > 0, 1 \leq i \leq n \} \quad (4.3.117)
\]

First, we will prove that \( S_{W3}^{B} \) defined in (4.3.112) is equal to \( X_{W3}^1 \) in (4.3.115).

- Consider the case that \( 2 < a_{ep}^i \leq 10 \). Since \( a_{ep}^i \leq 2 < a_{ep}^i \leq 10 \), \( (a_{ep}^i - 0.1a_{ep}^i) - 1.8 < 0 \). And since \( a_{ep}^i - F \leq 0 \) after Lemma 4.3.1, we have \( (a_{ep}^i - F) + (a_{ep}^i - 0.1a_{ep}^i) - 1.8 < 0 \), which is
contradictory to \((a_{ef}^i - F) + (a_{ep}^i - 0.1a_{ep}^i) - 1.8 > 0\). Thus,

\[
\{ s_i \mid 2 < a_{ep}^i \leq 10 \text{ and } (a_{ef}^i - F) + (a_{ep}^i - 0.1a_{ep}^i) - 1.8 > 0, 1 \leq i \leq n \} = \emptyset
\]

- Consider the case that \((a_{ep}^i > 10)\). We have \((a_{ep}^i - 0.001a_{ep}^i) - 2.79 < 0\) after \(a_{ep}^i \leq 2\) and \(a_{ep}^i > 10\). Since \(a_{ef}^i - F \leq 0\), we have \((a_{ef}^i - F) + (a_{ep}^i - 0.001a_{ep}^i) - 2.79 < 0\), which is contradictory to \((a_{ef}^i - F) + (a_{ep}^i - 0.001a_{ep}^i) - 2.79 > 0\). Thus,

\[
\{ s_i \mid a_{ep}^i > 10 \text{ and } (a_{ef}^i - F) + (a_{ep}^i - 0.001a_{ep}^i) - 2.79 > 0, 1 \leq i \leq n \} = \emptyset
\]

As a consequence, \(S^{W3}_B\) in (4.3.112) becomes

\[
S^{W3}_B = \{ s_i \mid a_{ep}^i \leq 2 \text{ and } (a_{ef}^i - F) + (a_{ep}^i - a_{ep}^i) > 0, 1 \leq i \leq n \} = X^{W3}_1
\]

Similarly, we can prove that \(S^{W3}_F\) defined in (4.3.113) is equal to \(Y^{W3}_1\) in (4.3.116).

Next, we are going to prove that \(S^{W3}_A\) defined in (4.3.114) is equal to \(Z^{W3}_1\) in (4.3.117). As shown in the above proof of \(S^{W3}_B = X^{W3}_1\), \((2 < a_{ep}^i \leq 10)\) implies \((a_{ef}^i - F) + (a_{ep}^i - 0.1a_{ep}^i) - 1.8 < 0\) and \((a_{ep}^i > 10)\) implies \((a_{ef}^i - F) + (a_{ep}^i - 0.001a_{ep}^i) - 2.79 < 0\). Thus, \((2 < a_{ep}^i \leq 10)\) and \((a_{ef}^i - F) + (a_{ep}^i - 0.1a_{ep}^i) - 1.8 < 0\) is logically equivalent to \((a_{ep}^i > 10)\) and \((a_{ef}^i - F) + (a_{ep}^i - 0.001a_{ep}^i) - 2.79 < 0\) is logically equivalent to \((a_{ep}^i > 10)\). Therefore, \(S^{W3}_A\) in (4.3.114) becomes

\[
\{ s_i \mid (a_{ep}^i > 2) \text{ or } (a_{ef}^i - F) + (a_{ep}^i - a_{ep}^i) < 0, 1 \leq i \leq n \} = Z^{W3}_1
\]

(2) Assume that \(2 < a_{ep}^i \leq 10\). Then, \(R^{W3}(s_f) = F - 2 - 0.1(a_{ef}^i - 2) = F - 0.1a_{ep}^i - 1.8\). After Definition 4.2.1 and re-arranging the terms, we have

\[
S^{W3}_B = \{ s_i \mid a_{ep}^i \leq 2 \text{ and } (a_{ef}^i - F) + (0.1a_{ep}^i - a_{ep}^i) + 1.8 > 0, 1 \leq i \leq n \}
\]

\[
\cup \{ s_i \mid 2 < a_{ep}^i \leq 10 \text{ and } (a_{ef}^i - F) + (0.1a_{ep}^i - 0.1a_{ep}^i) > 0, 1 \leq i \leq n \}
\]

\[
\cup \{ s_i \mid a_{ep}^i > 10 \text{ and } (a_{ef}^i - F) + (0.1a_{ep}^i - 0.001a_{ep}^i) < 0.99 > 0, 1 \leq i \leq n \} (4.3.118)
\]

\[
S^{W3}_F = \{ s_i \mid a_{ep}^i \leq 2 \text{ and } (a_{ef}^i - F) + (0.1a_{ep}^i - a_{ep}^i) + 1.8 = 0, 1 \leq i \leq n \}
\]

\[
\cup \{ s_i \mid 2 < a_{ep}^i \leq 10 \text{ and } (a_{ef}^i - F) + (0.1a_{ep}^i - 0.1a_{ep}^i) = 0, 1 \leq i \leq n \}
\]

\[
\cup \{ s_i \mid a_{ep}^i > 10 \text{ and } (a_{ef}^i - F) + (0.1a_{ep}^i - 0.001a_{ep}^i) = 0.99 = 0, 1 \leq i \leq n \} (4.3.119)
\]
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Then, we will prove that (4.3.121). Consider the case that \( 0 < X < W \).

We are going to prove that the above sets defined in (4.3.118), (4.3.119) and (4.3.120) are equal to the following sets \( X_2^{W3}, Y_2^{W3} \) and \( Z_2^{W3} \), respectively.

\[
S_A^{W3} = \{ s_i | a_{ep}^{i} \leq 2 \text{ and } (a_{ef}^{i} - F) + (0.1a_{ep}^{i} - a_{ep}^{i}) + 1.8 < 0, 1 \leq i \leq n \} \\
\cup \{ s_i | 2 < a_{ep}^{i} \leq 10 \text{ and } (a_{ef}^{i} - F) + (0.1a_{ep}^{i} - 0.1a_{ep}^{i}) < 0, 1 \leq i \leq n \} \\
\cup \{ s_i | a_{ep}^{i} > 10 \text{ and } (a_{ef}^{i} - F) + (0.1a_{ep}^{i} - 0.001a_{ep}^{i}) - 0.99 < 0, 1 \leq i \leq n \} \tag{4.3.120}
\]

First, we will prove that \( S_B^{W3} \) defined in (4.3.118) is equal to \( X_2^{W3} \) in (4.3.121). Consider the case that \( a_{ep}^{i} > 10 \). Thus, we have \( 2 < a_{ep}^{i} \leq 10 < a_{ep}^{i} \), which implies \( (0.1a_{ep}^{i} - 0.001a_{ep}^{i}) - 0.99 < 0 \). And since \( a_{ef}^{i} - F \leq 0 \) after Lemma 4.3.1, we have \( (a_{ef}^{i} - F) + (0.1a_{ep}^{i} - 0.001a_{ep}^{i}) - 0.99 < 0 \), which is contradictory to \( (a_{ef}^{i} - F) + (0.1a_{ep}^{i} - 0.001a_{ep}^{i}) - 0.99 > 0 \). Thus,

\[
\{ s_i | a_{ep}^{i} > 10 \text{ and } (a_{ef}^{i} - F) + (0.1a_{ep}^{i} - 0.001a_{ep}^{i}) - 0.99 > 0, 1 \leq i \leq n \} = \emptyset
\]

Then, \( S_B^{W3} \) in (4.3.118) becomes

\[
S_B^{W3} = \{ s_i | a_{ep}^{i} \leq 2 \text{ and } (a_{ef}^{i} - F) + (0.1a_{ep}^{i} - a_{ep}^{i}) + 1.8 > 0 \} \\
(2 < a_{ep}^{i} \leq 10 \text{ and } (a_{ef}^{i} - F) + (0.1a_{ep}^{i} - 0.1a_{ep}^{i}) > 0, 1 \leq i \leq n \} = X_2^{W3}
\]

Secondly, we will prove that \( S_f^{W3} \) defined in (4.3.119) is equal to \( Y_2^{W3} \) in (4.3.122).

- Consider the case that \( a_{ep}^{i} > 10 \). Thus, we have \( 2 < a_{ep}^{i} \leq 10 < a_{ep}^{i} \), which implies \( (0.1a_{ep}^{i} - 0.001a_{ep}^{i}) - 0.99 < 0 \). Since \( a_{ef}^{i} - F \leq 0 \) after Lemma 4.3.1, we have \( (a_{ef}^{i} - F) + (0.1a_{ep}^{i} - 0.001a_{ep}^{i}) - 0.99 < 0 \), which is contradictory to \( (a_{ef}^{i} - F) + (0.1a_{ep}^{i} - 0.001a_{ep}^{i}) - 0.99 = 0 \). Thus,

\[
\{ s_i | a_{ep}^{i} > 10 \text{ and } (a_{ef}^{i} - F) + (0.1a_{ep}^{i} - 0.001a_{ep}^{i}) - 0.99 = 0, 1 \leq i \leq n \} = \emptyset
\]
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

- Consider the case that $a_{ep}^i \leq 2$. Thus, we have $a_{ep}^i \leq 2 < a_{ep}^f \leq 10$. Then, $0.1a_{ep}^f$ will be within the range of $(0, 2)$. Therefore, $(0.1a_{ep}^f + 1.8)$ cannot be integer. As a consequence, $(a_{ep}^i - F) + (0.1a_{ep} - a_{ep}^i) + 1.8 = (a_{ep}^i - F - a_{ep}^i) + (0.1a_{ep} + 1.8) \neq 0$ because $a_{ep}^i, F$ and $a_{ep}^f$ are all integers. Thus,

$$\{ s_i | a_{ep}^i \leq 2 \text{ and } (a_{ep}^i - F) + (0.1a_{ep} - a_{ep}^i) + 1.8 = 0, 1 \leq i \leq n \} = \emptyset$$

Therefore, $S^W_F$ in (4.3.119) becomes

$$S^W_F = \{ s_i | 2 < a_{ep}^i \leq 10 \text{ and } (a_{ep}^i - F) + (0.1a_{ep} - a_{ep}^i) = 0, 1 \leq i \leq n \} = Y^W_F$$

Next, we are going to prove that $S^W_A$ defined in (4.3.120) is equal to $Z^W_A$ in (4.3.123). As shown in the above proof of $S_B^W = X^W_B$, $(a_{ep}^i > 10)$ implies $(a_{ep}^i - F) + (0.1a_{ep} - 0.001a_{ep}^i) - 0.99 < 0$. Thus, $(a_{ep}^i > 10)$ and $(a_{ep}^i - F) + (0.1a_{ep} - 0.001a_{ep}^i) - 0.99 < 0)$ is logically equivalent to $(a_{ep}^i > 10)$. Therefore, $S^W_A$ in (4.3.120) becomes

$$S^W_A = \{ s_i | a_{ep}^i \leq 2 \text{ and } (a_{ep}^i - F) + (0.1a_{ep} - a_{ep}^i) + 1.8 < 0 \text{ or } (2 < a_{ep}^i \leq 10 \text{ and } (a_{ep}^i - F) + (0.1a_{ep} - a_{ep}^i) = 0) \text{ or } (a_{ep}^i > 10), 1 \leq i \leq n \}$$

$$= Z^W_A$$

(3) Assume that $a_{ep}^f > 10$. Then, $R_{W3}(s_f) = F - 2.8 - 0.001(a_{ep}^f - 10) = F - 0.001a_{ep}^f - 2.79$. After Definition 4.2.1, we have

$$S^W_B = \{ s_i | a_{ep}^i \leq 2 \text{ and } a_{ep}^i - a_{ep}^f > F - 0.001a_{ep}^f - 2.79, 1 \leq i \leq n \}$$

$$\cup \{ s_i | 2 < a_{ep}^i \leq 10 \text{ and } a_{ep}^i - 2 - 0.1(a_{ep} - 2) > F - 0.001a_{ep} - 2.79, 1 \leq i \leq n \}$$

$$\cup \{ s_i | a_{ep}^i > 10 \text{ and } a_{ep}^i - 2.8 - 0.001(a_{ep}^i - 10) > F - 0.001a_{ep} - 2.79, 1 \leq i \leq n \}$$

(4.3.124)

$$S^W_F = \{ s_i | a_{ep}^i \leq 2 \text{ and } a_{ep}^i - a_{ep}^f = F - 0.001a_{ep}^f - 2.79, 1 \leq i \leq n \}$$

$$\cup \{ s_i | 2 < a_{ep}^i \leq 10 \text{ and } a_{ep}^i - 2 - 0.1(a_{ep} - 2) = F - 0.001a_{ep} - 2.79, 1 \leq i \leq n \}$$

$$\cup \{ s_i | a_{ep}^i > 10 \text{ and } a_{ep}^i - 2.8 - 0.001(a_{ep}^i - 10) = F - 0.001a_{ep} - 2.79, 1 \leq i \leq n \}$$

(4.3.125)
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

\[ S^W_A = \{ s_i | a^i_{ep} \leq 2 \text{ and } a^i_{ef} - a^i_{ep} < F - 0.001a^i_{ep} - 2.79, 1 \leq i \leq n \} \]
\[ \cup \{ s_i | 2 < a^i_{ep} \leq 10 \text{ and } a^i_{ef} - 2 - 0.1(a^i_{ep} - 2) < F - 0.001a^i_{ep} - 2.79, 1 \leq i \leq n \} \]
\[ \cup \{ s_i | a^i_{ep} > 10 \text{ and } a^i_{ef} - 2.8 - 0.001(a^i_{ep} - 10) < F - 0.001a^i_{ep} - 2.79, 1 \leq i \leq n \} \]

(4.3.126)

It is obvious that through re-arranging the terms and merging the subsets, the above sets defined in (4.3.124), (4.3.125) and (4.3.126) are equal to the following sets \( X^W_3, Y^W_3 \) and \( Z^W_3 \), respectively.

\[ X^W_3 = \{ s_i | (a^i_{ep} \leq 2 \text{ and } (a^i_{ef} - F) + (0.001a^i_{ep} - a^i_{ef}) + 2.79 > 0) \text{ or } \]
\[ (2 < a^i_{ep} \leq 10 \text{ and } (a^i_{ef} - F) + (0.001a^i_{ep} - 0.1a^i_{ep}) + 0.99 > 0) \text{ or } \]
\[ (a^i_{ep} > 10 \text{ and } (a^i_{ef} - F) + (0.001a^i_{ep} - 0.001a^i_{ep}) > 0), 1 \leq i \leq n \} \]

(4.3.127)

\[ Y^W_3 = \{ s_i | (a^i_{ep} \leq 2 \text{ and } (a^i_{ef} - F) + (0.001a^i_{ep} - a^i_{ef}) + 2.79 = 0) \text{ or } \]
\[ (2 < a^i_{ep} \leq 10 \text{ and } (a^i_{ef} - F) + (0.001a^i_{ep} - 0.1a^i_{ep}) + 0.99 = 0) \text{ or } \]
\[ (a^i_{ep} > 10 \text{ and } (a^i_{ef} - F) + (0.001a^i_{ep} - 0.001a^i_{ep}) = 0), 1 \leq i \leq n \} \]

(4.3.128)

\[ Z^W_3 = \{ s_i | (a^i_{ep} \leq 2 \text{ and } (a^i_{ef} - F) + (0.001a^i_{ep} - a^i_{ef}) + 2.79 < 0) \text{ or } \]
\[ (2 < a^i_{ep} \leq 10 \text{ and } (a^i_{ef} - F) + (0.001a^i_{ep} - 0.1a^i_{ep}) + 0.99 < 0) \text{ or } \]
\[ (a^i_{ep} > 10 \text{ and } (a^i_{ef} - F) + (0.001a^i_{ep} - 0.001a^i_{ep}) < 0), 1 \leq i \leq n \} \]

(4.3.129)

28. Arithmetic Mean

As stated in Table 4.1, formula Arithmetic Mean is defined as follows.

\[ R_{AM}(s_i) = \frac{2a^i_{ef}a^i_{np} - 2a^i_{ef}a^i_{np}}{(a^i_{ef} + a^i_{np})(a^i_{np} + a^i_{ef}) + (a^i_{ef} + a^i_{nf})(a^i_{np} + a^i_{nf})} \]

From Definition 4.2.1 and Lemmas 4.3.1 and 4.3.2, after simplification, we have

\[ S^W_B = \{ s_i | \frac{a^i_{ef}P - a^i_{np}F}{(a^i_{ef} + a^i_{np})(P + F - a^i_{ef} - a^i_{np}) + PF} > \frac{PF - Fa^i_{ef}}{(F + a^i_{ef})(P - a^i_{ef}) + PF}, 1 \leq i \leq n \} \]

(4.3.130)

\[ S^W_F = \{ s_i | \frac{a^i_{ef}P - a^i_{np}F}{(a^i_{ef} + a^i_{np})(P + F - a^i_{ef} - a^i_{np}) + PF} = \frac{PF - Fa^i_{ef}}{(F + a^i_{ef})(P - a^i_{ef}) + PF}, 1 \leq i \leq n \} \]

(4.3.131)

\[ S^W_A = \{ s_i | \frac{a^i_{ef}P - a^i_{np}F}{(a^i_{ef} + a^i_{np})(P + F - a^i_{ef} - a^i_{np}) + PF} < \frac{PF - Fa^i_{ef}}{(F + a^i_{ef})(P - a^i_{ef}) + PF}, 1 \leq i \leq n \} \]

(4.3.132)
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

29. Cohen

As stated in Table 4.1, formula Cohen is defined as follows.

\[
R_{CO}(s_i) = \frac{2a_{ef}a_{np} - 2a_{nf}a_{ep}}{(a_{ef} + a_{ep})(a_{np} + a_{ep}) + (a_{ef} + a_{nf})(a_{nf} + a_{np})}
\]

From Definition 4.2.1 and Lemmas 4.3.1 and 4.3.2, after simplification, we have

\[
S_{B}^{CO} = \{ s_i | \frac{a_{ef}^i P - a_{ep}^i F}{P(a_{ef}^i + a_{ep}^i) + F(P + P - a_{ef}^i - a_{ep}^i)} > \frac{PF - F a_{ep}^i}{P(F + a_{ep}^i) + F(P - a_{ep}^i)}, 1 \leq i \leq n \} \quad (4.3.133)
\]

\[
S_{F}^{CO} = \{ s_i | \frac{a_{ef}^i P - a_{ep}^i F}{P(a_{ef}^i + a_{ep}^i) + F(P + P - a_{ef}^i - a_{ep}^i)} = \frac{PF - F a_{ep}^i}{P(F + a_{ep}^i) + F(P - a_{ep}^i)}, 1 \leq i \leq n \} \quad (4.3.134)
\]

\[
S_{A}^{CO} = \{ s_i | \frac{a_{ef}^i P - a_{ep}^i F}{P(a_{ef}^i + a_{ep}^i) + F(P + P - a_{ef}^i - a_{ep}^i)} < \frac{PF - F a_{ep}^i}{P(F + a_{ep}^i) + F(P - a_{ep}^i)}, 1 \leq i \leq n \} \quad (4.3.135)
\]

30. Fleiss

As stated in Table 4.1, formula Cohen is defined as follows.

\[
R_{F}(s_i) = \frac{4a_{ef}a_{np} - 4a_{nf}a_{ep} - (a_{nf} - a_{ep})^2}{(2a_{ef} + a_{np} + a_{ep}) + (2a_{np} + a_{nf} + a_{ep})}
\]

From Definition 4.2.1 and Lemmas 4.3.1 and 4.3.2, after simplification, we have

\[
S_{B}^{F} = \{ s_i | -F^2 + 4a_{ef}^i P + 2Fa_{ef}^i - 2Fa_{ep}^i - (a_{ep}^i + a_{ef}^i)^2 > \frac{4PF - 4Fa_{ep}^i - (a_{ep}^i)^2}{2P + 2F}, 1 \leq i \leq n \} \quad (4.3.136)
\]

\[
S_{F}^{F} = \{ s_i | -F^2 + 4a_{ef}^i P + 2Fa_{ef}^i - 2Fa_{ep}^i - (a_{ep}^i + a_{ef}^i)^2 = \frac{4PF - 4Fa_{ep}^i - (a_{ep}^i)^2}{2P + 2F}, 1 \leq i \leq n \} \quad (4.3.137)
\]

\[
S_{A}^{F} = \{ s_i | -F^2 + 4a_{ef}^i P + 2Fa_{ef}^i - 2Fa_{ep}^i - (a_{ep}^i + a_{ef}^i)^2 < \frac{4PF - 4Fa_{ep}^i - (a_{ep}^i)^2}{2P + 2F}, 1 \leq i \leq n \} \quad (4.3.138)
\]

Since \(2P + 2F > 0\), obviously, \(S_{B}^{F}\), \(S_{F}^{F}\) and \(S_{A}^{F}\) are equal to the following sets \(X^{F}\), \(Y^{F}\) and \(Z^{F}\), respectively.

\[
X^{F} = \{ s_i | -F^2 + 4a_{ef}^i P + 2Fa_{ef}^i - 2Fa_{ep}^i - (a_{ep}^i + a_{ef}^i)^2 > 4PF - 4Fa_{ep}^i - (a_{ep}^i)^2, 1 \leq i \leq n \} \quad (4.3.136)
\]

\[
Y^{F} = \{ s_i | -F^2 + 4a_{ef}^i P + 2Fa_{ef}^i - 2Fa_{ep}^i - (a_{ep}^i + a_{ef}^i)^2 = 4PF - 4Fa_{ep}^i - (a_{ep}^i)^2, 1 \leq i \leq n \} \quad (4.3.137)
\]

\[
Z^{F} = \{ s_i | -F^2 + 4a_{ef}^i P + 2Fa_{ef}^i - 2Fa_{ep}^i - (a_{ep}^i + a_{ef}^i)^2 < 4PF - 4Fa_{ep}^i - (a_{ep}^i)^2, 1 \leq i \leq n \} \quad (4.3.138)
\]
4.3 EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

4.3.4 Maximal risk evaluation formulas

In this section, we will apply our framework to identify the most effective formulas, namely the maximal formulas, among the 30 investigated formulas, under the single-fault scenario.

Normally speaking, an element \( a \) in a set \( S \) is said to be maximal if for any element \( b \in S \), whenever \( b > a \), \( b \) is \( a \) (where “>” is a partial ordering). But in our study, we use the following definition for maximal formulas: a risk evaluation formula \( R_1 \) is said to be a maximal formula of a set of formulas, if for any element \( R_2 \) of this set of formulas, \( R_2 \to R_1 \) implies \( R_2 \leftrightarrow R_1 \). We use “\( R_2 \leftrightarrow R_1 \)” instead of “\( R_2 \) is \( R_1 \)” because \( EXAM \) score is the key to distinguish the performance between different formulas and different formulas may have the same \( EXAM \) score. Hence, it is more intuitively appealing to have “\( R_2 \leftrightarrow R_1 \)” in the context of SBFL.

First, let us consider the equivalence relation. From the discussion in previous section, it is not difficult to find that among the 30 investigated formulas, there are six groups of equivalent formulas (which are referred to as “ER1” to “ER6”), as follows.

- ER1 consists of Op1 and Op2.
- ER2 consists of Jaccard, Anderberg, Sørensen-Dice, Dice and Goodman.
- ER3 consists of Tarantula, \( q_e \) and CBI Inc.
- ER4 consists of Wong2, Hamann, Simple Matching, Sokal, Rogers & Tanimoto, Hamming etc., and Euclid.
- ER5 consists of Wong1, Russell & Rao and Binary.
- ER6 consists of Scott and Rogot1.

The following six propositions formally present these six equivalent groups of formulas.

**Proposition 4.3.1.** For ER1, we have \( Op1 \leftrightarrow Op2 \).

**Proof.** As proved in Section 4.3.3, \( S^R_B \), \( S^R_F \) and \( S^R_A \) of both Op1 and Op2 are equal to the sets defined in (4.3.4), (4.3.5) and (4.3.6), respectively, as follows.

\[
S^R_B = \{ s_i | a_{ef}^i = F \text{ and } a_{cp}^f - a_{ep}^i > 0, 1 \leq i \leq n \}
\]

\[
S^R_F = \{ s_i | a_{ef}^i = F \text{ and } a_{cp}^f - a_{ep}^i = 0, 1 \leq i \leq n \}
\]

\[
S^R_A = \{ s_i | (a_{ef}^i < F) \text{ or } (a_{ef}^i = F \text{ and } a_{cp}^f - a_{ep}^i < 0), 1 \leq i \leq n \}
\]

Therefore, we have \( S^Op1_B = S^Op2_B \), \( S^Op1_F = S^Op2_F \) and \( S^Op1_A = S^Op2_A \). Immediately after Theorem 4.2.3, \( Op1 \leftrightarrow Op2 \).
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Proposition 4.3.2. For ER2, we have Jaccard ↔ Anderberg ↔ Sørensen-Dice ↔ Dice ↔ Goodman.

Proof. As proved in Section 4.3.3, \( S^R_B, S^R_F \) and \( S^R_A \) of each formula \( R \) in ER2 are the same as the sets defined in (4.3.14), (4.3.15) and (4.3.16), respectively, as follows.

\[
S^R_B = \{ s_i | a^i_{ef} > 0 \text{ and } 1 + \frac{a^f_{ep}}{F} \frac{a^i_{ep}}{a^i_{ef}} > 0, 1 \leq i \leq n \} \\
S^R_F = \{ s_i | a^i_{ef} > 0 \text{ and } 1 + \frac{a^f_{ep}}{F} \frac{a^i_{ep}}{a^i_{ef}} = 0, 1 \leq i \leq n \} \\
S^R_A = \{ s_i | (a^i_{ef} = 0) \text{ or } (a^i_{ef} > 0 \text{ and } 1 + \frac{a^f_{ep}}{F} \frac{a^i_{ep}}{a^i_{ef}} < 0), 1 \leq i \leq n \}
\]

Obviously, for any two formulas \( R_1 \) and \( R_2 \) of ER2, we have \( S^R_B = S^R_B, S^R_F = S^R_F \) and \( S^R_A = S^R_A \).
Immediately after Theorem 4.2.3, \( R_1 \leftrightarrow R_2 \), that is, Jaccard ↔ Anderberg ↔ Sørensen-Dice ↔ Dice ↔ Goodman.

Proposition 4.3.3. For ER3, we have Tarantula ↔ qe ↔ CBI Inc.

Proof. As proved in Section 4.3.3, \( S^R_B, S^R_F \) and \( S^R_A \) of each formula \( R \) in ER3 are the same as the sets defined in (4.3.37), (4.3.38) and (4.3.39), respectively, as follows.

\[
S^R_B = \{ s_i | a^i_{ef} > 0 \text{ and } 1 + \frac{a^f_{ep}}{F} \frac{a^i_{ep}}{a^i_{ef}} > 0, 1 \leq i \leq n \} \\
S^R_F = \{ s_i | a^i_{ef} > 0 \text{ and } 1 + \frac{a^f_{ep}}{F} \frac{a^i_{ep}}{a^i_{ef}} = 0, 1 \leq i \leq n \} \\
S^R_A = \{ s_i | (a^i_{ef} = 0) \text{ or } (a^i_{ef} > 0 \text{ and } 1 + \frac{a^f_{ep}}{F} \frac{a^i_{ep}}{a^i_{ef}} < 0), 1 \leq i \leq n \}
\]

Therefore, for any two formulas \( R_1 \) and \( R_2 \) of ER3, we have \( S^R_B = S^R_B, S^R_F = S^R_F \) and \( S^R_A = S^R_A \).
It follows immediately from Theorem 4.2.3 that Tarantula ↔ qe ↔ CBI Inc.

Proposition 4.3.4. For ER4, we have Wong2 ↔ Hamann ↔ Simple Matching ↔ Sokal ↔ Rogers & Tanimoto ↔ Hamming etc. ↔ Euclid.

Proof. As proved in Section 4.3.3, \( S^R_B, S^R_F \) and \( S^R_A \) of each formula \( R \) in ER4 are equal to the sets defined in (4.3.48), (4.3.49) and (4.3.50), respectively, as follows.

\[
S^R_B = \{ s_i | (a^i_{ef} - F) + (a^f - a^i_{ep}) > 0, 1 \leq i \leq n \} \\
S^R_F = \{ s_i | (a^i_{ef} - F) + (a^f - a^i_{ep}) = 0, 1 \leq i \leq n \} \\
S^R_A = \{ s_i | (a^i_{ef} - F) + (a^f - a^i_{ep}) < 0, 1 \leq i \leq n \}
\]
Therefore, for any two formulas $R_1$ and $R_2$ of ER4, we have $S_{B_1}^{R_1} = S_{B_2}^{R_2}$, $S_{F_1}^{R_1} = S_{F_2}^{R_2}$ and $S_{A_1}^{R_1} = S_{A_2}^{R_2}$.

Immediately after Theorem 4.2.3, we have Wong2 $\leftrightarrow$ Hamann $\leftrightarrow$ Simple Matching $\leftrightarrow$ Sokal $\leftrightarrow$ Rogers & Tanimoto $\leftrightarrow$ Hamming etc. $\leftrightarrow$ Euclid.

**Proposition 4.3.5.** For ER5, we have Wong1 $\leftrightarrow$ Russell & Rao $\leftrightarrow$ Binary.

**Proof.** As proved in Section 4.3.3, $S_{B_1}^{R_1}$, $S_{F_1}^{R_1}$ and $S_{A_1}^{R_1}$ of each formula $R$ in ER5 are equal to sets $\emptyset$, \{s_i|a_{ef}^i = F, 1 \leq i \leq n\} and \{s_i|a_{ef}^i < F, 1 \leq i \leq n\} defined in (4.3.69), (4.3.70) and (4.3.71), respectively.

Obviously, we have $S_{B_1}^{W_1} = S_{B_1}^{R_1}$, $S_{F_1}^{W_1} = S_{F_1}^{R_1}$ and $S_{A_1}^{W_1} = S_{A_1}^{R_1}$. After Theorem 4.2.3, Wong1 $\leftrightarrow$ Russell & Rao $\leftrightarrow$ Binary.

**Proposition 4.3.6.** For ER6, we have Scott $\leftrightarrow$ Rogot1.

**Proof.** As proved in Section 4.3.3, $S_{B_1}^{R_1}$, $S_{F_1}^{R_1}$ and $S_{A_1}^{R_1}$ of both Scott and Rogot1 are the same as the sets defined in (4.3.78), (4.3.79) and (4.3.80), respectively, as follows.

\[
S_{B_1}^{R_1} = \{s_i|\frac{-F^2 + 4a_{ef}^i P + 2F a_{ef}^i - 2F a_{ep}^i - (a_{ep}^i + a_{ef}^i)^2}{(F + 2P - a_{ep}^i - a_{ef}^i)(F + a_{ef}^i + a_{ep}^i)} > \frac{4PF - 4F a_{ep}^i - (a_{ep}^i)^2}{(2F + a_{ep}^i)(2P - a_{ep}^i)}, 1 \leq i \leq n\}
\]

\[
S_{F_1}^{R_1} = \{s_i|\frac{-F^2 + 4a_{ef}^i P + 2F a_{ef}^i - 2F a_{ep}^i - (a_{ep}^i + a_{ef}^i)^2}{(F + 2P - a_{ep}^i - a_{ef}^i)(F + a_{ef}^i + a_{ep}^i)} > \frac{4PF - 4F a_{ep}^i - (a_{ep}^i)^2}{(2F + a_{ep}^i)(2P - a_{ep}^i)}, 1 \leq i \leq n\}
\]

\[
S_{A_1}^{R_1} = \{s_i|\frac{-F^2 + 4a_{ef}^i P + 2F a_{ef}^i - 2F a_{ep}^i - (a_{ep}^i + a_{ef}^i)^2}{(F + 2P - a_{ep}^i - a_{ef}^i)(F + a_{ef}^i + a_{ep}^i)} < \frac{4PF - 4F a_{ep}^i - (a_{ep}^i)^2}{(2F + a_{ep}^i)(2P - a_{ep}^i)}, 1 \leq i \leq n\}
\]

Obviously, the sets $S_{B_1}^{R_1}$, $S_{F_1}^{R_1}$ and $S_{A_1}^{R_1}$ of Scott are equal to the corresponding sets of Rogot1, respectively. After Theorem 4.2.3, Scott $\leftrightarrow$ Rogot1.

By using the notion of subset, we have provided the proofs of the equivalence relations with respect to our definition of equivalence in Definition 4.2.4. The results are the same as the ones in [Naish et al., 2011], because their definition of equivalence is a special case of ours. In other words, if two risk evaluation formulas $R_1$ and $R_2$ are equivalent with respect to the type of equivalence proposed by Naish et al., they are also equivalent with respect to our type of equivalence (Definition 4.2.4). The reason is simple: with respect to Naish et al.’s definition of equivalence, $R_1$ is equivalent to $R_2$, if they always produce the same rankings for all the statements. Hence, $R_1$ and $R_2$ will have the same EXAM scores. It follows from Definition 4.2.4 that $R_1$ and $R_2$ are equivalent with respect to our type of equivalence.

We use the following example formula $re$ to show that even if $R_1 \leftrightarrow R_2$, $R_1$ and $R_2$ are not...
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

necessarily equivalent with respect to Naish et al.'s type of equivalence.

\[
\text{re} = \begin{cases} 
\frac{a_{ep}}{a_{ef}} & \text{if } a_{ef} > 0 \\
-P & \text{if } a_{ef} = 0 
\end{cases} 
\] (4.3.139)

Though \( \text{re} \) is artificially constructed, it is an intuitively appealing risk evaluation formula because of the following reasons.

- For \( s_i \) with \( a_{ef}^i > 0 \), \( \text{re} \) follows the general expectation as other widely adopted formulas, that statements associated with more failed and less passed testing results should have higher risk values.

- For \( s_i \) with \( a_{ef}^i = 0 \), \( \text{re} \) assigns risk value of \( (-\frac{P}{F} - 1) \) to them. Since \( F \geq 1 \) and \( a_{ep}^f \leq P \) after Lemma 4.3.1, we have \( -\frac{a_{ep}^f}{F} \geq -\frac{P}{F} > (-\frac{P}{F} - 1) \). Therefore, the risk value of \( s_i \) with \( a_{ef}^i = 0 \), which is \( (-\frac{P}{F} - 1) \), is always lower than the risk value of \( s_f \), which is \( -\frac{a_{ef}^f}{F} \). This is intuitively expected because under the single-fault scenario, a statement \( s_i \) with \( a_{ef}^i = 0 \) can never be the faulty statement [Xie et al., 2010].

We will prove that \( \text{re} \leftrightarrow \text{Tarantula} \). After Lemma 4.3.2 and Definition 4.2.1, we have

\[
S_{RE}^B = \{s_i | (a_{ef}^i > 0 \text{ and } \frac{a_{ep}^i}{a_{ef}^i} > \frac{a_{ef}^f}{F}) \text{ or } (a_{ef}^i = 0 \text{ and } \frac{a_{ep}^f}{F} - 1 > -\frac{a_{ep}^f}{F}), 1 \leq i \leq n \} 
\]

\[
S_{RE}^E = \{s_i | (a_{ef}^i > 0 \text{ and } \frac{a_{ep}^i}{a_{ef}^i} = -\frac{a_{ef}^f}{F}) \text{ or } (a_{ef}^i = 0 \text{ and } \frac{a_{ep}^f}{F} - 1 = -\frac{a_{ep}^f}{F}), 1 \leq i \leq n \} 
\]

\[
S_{RE}^A = \{s_i | (a_{ef}^i > 0 \text{ and } \frac{a_{ep}^i}{a_{ef}^i} < -\frac{a_{ef}^f}{F}) \text{ or } (a_{ef}^i = 0 \text{ and } \frac{a_{ep}^f}{F} - 1 < -\frac{a_{ep}^f}{F}), 1 \leq i \leq n \} 
\]

\( S_{RE}^B \) can be re-written as

\[
S_{RE}^B = \{s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ep}^i}{a_{ef}^i} > \frac{a_{ef}^f}{F}, 1 \leq i \leq n \} \cup \{s_i | a_{ef}^i = 0 \text{ and } \frac{a_{ep}^f}{F} - 1 > -\frac{a_{ep}^f}{F}, 1 \leq i \leq n \}
\]

Assume \( (-\frac{P}{F} - 1) > -\frac{a_{ep}^f}{F} \). Then, we have \( \frac{a_{ep}^f}{F} > (\frac{P}{F} + 1) \). Therefore, \( a_{ep}^f > P \), which is a contradiction to Lemma 4.3.1. Thus, \( \{s_i | a_{ef}^i = 0 \text{ and } -\frac{P}{F} - 1 > -\frac{a_{ep}^f}{F}, 1 \leq i \leq n \} = \emptyset \). Then, \( S_{RE}^B = \{s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ep}^i}{a_{ef}^i} > -\frac{a_{ef}^f}{F}, 1 \leq i \leq n \} \), which can be re-written as \( \{s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ep}^f}{F} - a_{ef}^f > 0, 1 \leq i \leq n \} = S_{RE}^B \) in (4.3.37).
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Similarly, we can prove that $S_R^E$ is equal to $S_T^E$ defined in (4.3.38).

Next, consider $S_R^A$. Since we have $(-P^{\frac{F}{F}} - 1) < -\frac{a_{ep}}{a_{ef}^{i}}$ after Lemma 4.3.1, $(a_{ef}^{i} = 0$ and $-P^{\frac{F}{F}} - 1 < -\frac{a_{ep}}{a_{ef}^{i}}$) is logically equivalent to $(a_{ef}^{i} = 0)$. Thus,

$$S_R^A = \{ s_i | (a_{ef}^{i} = 0) \text{ or } (a_{ef}^{i} > 0 \text{ and } -\frac{a_{ep}}{a_{ef}^{i}} < -\frac{a_{ep}}{a_{ef}^{i}}, 1 \leq i \leq n \}$$

which can be re-written as $\{ s_i | (a_{ef}^{i} = 0) \text{ or } (a_{ef}^{i} > 0 \text{ and } -\frac{a_{ep}}{a_{ef}^{i}} < 0), 1 \leq i \leq n \} = S_T^A$ in (4.3.39).

In summary, $S_R^B$, $S_R^E$ and $S_R^A$ are proved to be equal to the sets in (4.3.37), (4.3.38) and (4.3.39) of Tarantula, respectively. After Theorem 4.2.3, we have $r_e \leftrightarrow$ Tarantula.

However, $r_e$ cannot guarantee to produce identical ranking list as formula Tarantula. For Tarantula, $s_i$ with $a_{ef}^{i} = 0$ will be assigned with the lowest risk value of 0, and hence is ranked lower than any $s_j$ with $a_{ef}^{j} > 0$. Consider a test suite with $P=6$ and $F=2$. For $r_e$, if there is $s_i$ with $a_{ef}^{i} = 0$, we have $R_{RE}(s_i) = -\frac{P^{\frac{F}{F}} - 1}{-4} = -4$. Suppose there exists $s_j$ such that $a_{ef}^{j} = 1$ and $a_{ep}^{j} = 5$. Then, we have $R_{RE}(s_j) = -\frac{a_{ep}^{j}}{a_{ef}^{j}} - 5 < -4 = R_{RE}(s_i)$. Therefore, $r_e$ does not produce the same ranking list as Tarantula. If Naish et al.’s type of equivalence is used, $r_e$ is not equivalent to Tarantula. As compared with their type of equivalence, ours is not only more general, but also more intuitively appealing in the context of SBFL. As a reminder, Naish et al. also observed that “Wong1 and Russell & Rao result in the same ranking for the single bug as Binary”. However, they did not use this observation to further extend their definition of equivalence.

With the above six groups of equivalent formulas, we only need to search the maximal formulas from the following 14 individual formulas or groups of equivalent formulas, namely, ER1, ER2, ER3, ER4, ER5, ER6, Kulczynski2, M2, Ochiai, AMPLE2, Wong3, Arithmetic Mean, Cohen and Fleiss. We adopt an approach of elimination to identify the maximal formulas in the following analysis.

**Proposition 4.3.7.** ER2 $\rightarrow$ ER3.

**Proof.** In order to prove ER2 $\rightarrow$ ER3, it is sufficient to prove Jaccard $\rightarrow$ Tarantula. As proved in Section 4.3.3, $S_B^J$ and $S_A^J$ are equal to the sets defined in (4.3.14) and (4.3.16), respectively, as follows.

$$S_B^J = \{ s_i | a_{ef}^{i} > 0 \text{ and } 1 + \frac{a_{ep}^{i}}{a_{ef}^{i}} - \frac{F}{a_{ef}^{i}} - \frac{a_{ep}^{i}}{a_{ef}^{i}} > 0, 1 \leq i \leq n \}$$

$$S_A^J = \{ s_i | (a_{ef}^{i} = 0) \text{ or } (a_{ef}^{i} > 0 \text{ and } 1 + \frac{a_{ep}^{i}}{a_{ef}^{i}} - \frac{F}{a_{ef}^{i}} - \frac{a_{ep}^{i}}{a_{ef}^{i}} < 0), 1 \leq i \leq n \}$$
And $S^T_B$ and $S^T_A$ are equal to the sets defined in (4.3.37) and (4.3.39), respectively, as follows.

$$S^T_B = \{ s_i | a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i}{F} - \frac{a_{ep}^i}{a_{ef}^i} > 0, 1 \leq i \leq n \}$$

$$S^T_A = \{ s_i | (a_{ef}^i = 0) \text{ or } (a_{ef}^i > 0 \text{ and } \frac{a_{ef}^i}{F} - \frac{a_{ep}^i}{a_{ef}^i} < 0), 1 \leq i \leq n \}$$

After re-arranging the terms in $1+\frac{a_{ep}^i}{a_{ef}^i} - \frac{F}{a_{ef}^i} - \frac{a_{ep}^i}{a_{ef}^i}$ from (4.3.14) and (4.3.16), we have

$$1+\frac{a_{ep}^i}{a_{ef}^i} - \frac{F}{a_{ef}^i} - \frac{a_{ep}^i}{a_{ef}^i} = \left( \frac{a_{ep}^i}{a_{ef}^i} - \frac{a_{ep}^i}{a_{ef}^i} \right) + \left( 1 - \frac{F}{a_{ef}^i} \right)$$

Since $1 - \frac{F}{a_{ef}^i} \leq 0$ after Lemma 4.3.1, we have

$$1+\frac{a_{ep}^i}{a_{ef}^i} - \frac{F}{a_{ef}^i} - \frac{a_{ep}^i}{a_{ef}^i} \leq \frac{a_{ep}^i}{a_{ef}^i} - \frac{a_{ep}^i}{a_{ef}^i} \quad (4.3.140)$$

Now, we are going to prove $S^I_B \subseteq S^T_B$ and $S^T_A \subseteq S^I_A$.

First, we will prove $S^I_B \subseteq S^T_B$. Assume $s_i \in S^I_B$. Then, we have $(a_{ef}^i > 0$ and $1+\frac{a_{ep}^i}{a_{ef}^i} - \frac{F}{a_{ef}^i} - \frac{a_{ep}^i}{a_{ef}^i} > 0)$ after (4.3.14). As a consequence, we have $\frac{a_{ep}^i}{a_{ef}^i} > 0$ from (4.3.140). Thus, $s_i \in S^T_B$ after (4.3.37). Therefore, $S^I_B \subseteq S^T_B$.

Secondly, we will prove $S^T_A \subseteq S^I_A$. Assume $s_i \in S^T_A$. Then, we have either $(a_{ef}^i = 0)$ or $(a_{ef}^i > 0$ and $1+\frac{a_{ep}^i}{a_{ef}^i} - \frac{F}{a_{ef}^i} - \frac{a_{ep}^i}{a_{ef}^i} < 0)$ after (4.3.39).

- Consider the case that $(a_{ef}^i = 0)$. Immediately after (4.3.16), $s_i \in S^I_A$.
- Consider the case that $(a_{ef}^i > 0$ and $1+\frac{a_{ep}^i}{a_{ef}^i} - \frac{F}{a_{ef}^i} - \frac{a_{ep}^i}{a_{ef}^i} < 0)$. Then, we have $1+\frac{a_{ep}^i}{a_{ef}^i} - \frac{F}{a_{ef}^i} - \frac{a_{ep}^i}{a_{ef}^i} < 0$ after (4.3.140). Thus, $s_i \in S^I_A$ after (4.3.16).

In summary, we have proved that $S^T_A \subseteq S^I_A$.

In conclusion, we have $S^I_B \subseteq S^T_B$ and $S^T_A \subseteq S^I_A$. Immediately after Theorem 4.2.2, Jaccard $\rightarrow$ Tarantula. And after Propositions 4.3.2 and 4.3.3, ER2 $\rightarrow$ ER3.

**Proposition 4.3.8.** ER2 $\rightarrow$ ER4.

**Proof.** In order to prove ER2 $\rightarrow$ ER4, it is sufficient to prove Jaccard $\rightarrow$ Wong2. As proved in Section 4.3.3, $S^I_B$ and $S^I_A$ are equal to the sets defined in (4.3.14) and (4.3.16), respectively; and $S^W_B$
and \( S_W^2 \) are equal to the sets defined in (4.3.48) and (4.3.50), respectively, as follows.

\[
S_B^W = \{ s_i | (a_{ef}^i - F) + (a_{ep}^i - a_{ep}) > 0, 1 \leq i \leq n \}
\]

\[
S_A^W = \{ s_i | (a_{ef}^i - F) + (a_{ep}^i - a_{ep}) < 0, 1 \leq i \leq n \}
\]

After re-arranging the terms in \( 1 + \frac{a_{ep}^i}{a_{ef}^i} - \frac{F - a_{ep}^i}{a_{ef}^i} \) from (4.3.14) and (4.3.16), we have

\[
1 + \frac{a_{ep}^i}{a_{ef}^i} - \frac{F - a_{ep}^i}{a_{ef}^i} = (1 + \frac{a_{ep}^i}{a_{ef}^i} - \frac{F - a_{ep}^i}{a_{ef}^i}) + a_{ep}^i (\frac{1}{F} - \frac{1}{a_{ef}^i})
\]

Since \( a_{ep}^i (\frac{1}{F} - \frac{1}{a_{ef}^i}) \leq 0 \) after Lemma 4.3.1, we have

\[
1 + \frac{a_{ep}^i}{a_{ef}^i} - \frac{F - a_{ep}^i}{a_{ef}^i} \leq 1 + \frac{a_{ep}^i}{a_{ef}^i} - \frac{F - a_{ep}^i}{a_{ef}^i}
\]

(4.3.141)

Now, we are going to prove \( S_B^I \subseteq S_B^W \) and \( S_A^W \subseteq S_A^I \).

First, we will prove \( S_B^I \subseteq S_B^W \). Assume \( s_i \in S_B^I \). Then, we have \( (a_{ef}^i - F) + (a_{ep}^i - a_{ep}) > 0 \) after (4.3.14). As a consequence, we have \( 1 + \frac{a_{ep}^i}{a_{ef}^i} - \frac{F - a_{ep}^i}{a_{ef}^i} > 0 \) from (4.3.141). Furthermore, since \( \frac{a_{ep}^i}{a_{ef}^i} - \frac{F - a_{ep}^i}{a_{ef}^i} = (a_{ef}^i - F) + (a_{ep}^i - a_{ep}) \) and \( a_{ef}^i > 0 \), we have \( (a_{ef}^i - F) + (a_{ep}^i - a_{ep}) > 0 \). After (4.3.48), \( s_i \in S_B^W \). Therefore, \( S_B^I \subseteq S_B^W \).

Secondly, we will prove \( S_A^W \subseteq S_A^I \). Assume \( s_i \in S_A^W \). Then, we have \( (a_{ef}^i - F) + (a_{ep}^i - a_{ep}) < 0 \) after (4.3.50). Let us consider the following situations:

- Suppose \( a_{ef}^i = 0 \). Immediately after (4.3.16), \( s_i \in S_A^I \).

- Suppose \( a_{ef}^i > 0 \). Since \( \frac{(a_{ef}^i - F) + (a_{ep}^i - a_{ep})}{a_{ef}^i} = 1 + \frac{a_{ep}^i}{a_{ef}^i} - \frac{F - a_{ep}^i}{a_{ef}^i} \), we have \( 1 + \frac{a_{ep}^i}{a_{ef}^i} - \frac{F - a_{ep}^i}{a_{ef}^i} < 0 \).

As a consequence, we have \( 1 + \frac{a_{ep}^i}{a_{ef}^i} - \frac{F - a_{ep}^i}{a_{ef}^i} < 0 \) after (4.3.141). Thus, \( s_i \in S_A^I \) after (4.3.16).

In summary, we have proved that \( S_A^W \subseteq S_A^I \).

In conclusion, we have \( S_B^I \subseteq S_B^W \) and \( S_A^W \subseteq S_A^I \). Immediately after Theorem 4.2.2, Jaccard \( \Rightarrow \) Wong2. And after Proposition 4.3.2 and 4.3.4, ER2 \( \Rightarrow \) ER4.

**Proposition 4.3.9.** Ochiai \( \Rightarrow \) ER2.

**Proof.** In order to prove Ochiai \( \Rightarrow \) ER2, it is sufficient to prove Ochiai \( \Rightarrow \) Jaccard. As proved in Section 4.3.3, \( S_B^I \) and \( S_A^I \) are equal to the sets defined in (4.3.14) and (4.3.16), respectively; and \( S_B^O \)
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

and \( S_A^O \) are equal to the sets defined in (4.3.101) and (4.3.103), respectively, as follows.

\[
S_B^O = \{ s_i | a_{ef}^i > 0 \text{ and } (1 + \frac{a_{ef}^i}{F}) \frac{a_{ef}^i}{F} - 1 - \frac{a_{ep}^i}{a_{ef}^i} > 0, 1 \leq i \leq n \} \\
S_A^O = \{ s_i | (a_{ef}^i = 0) \text{ or } (a_{ef}^i > 0 \text{ and } (1 + \frac{a_{ef}^i}{F}) \frac{a_{ef}^i}{F} - 1 - \frac{a_{ep}^i}{a_{ef}^i} < 0), 1 \leq i \leq n \}
\]

Let \( f_J \) and \( f_O \) denote the following expressions.

\[
f_J(s_i) = 1 + \frac{a_{ep}^i}{a_{ef}^i} - \frac{F}{a_{ef}^i} = \frac{a_{ef}^i F + a_{ef}^i a_{ep}^i - F^2}{F a_{ef}^i} \quad \text{(4.3.142)}
\]

\[
f_O(s_i) = (1 + \frac{a_{ep}^i}{a_{ef}^i}) \frac{a_{ef}^i}{F} - 1 = \frac{a_{ef}^i F + a_{ef}^i a_{ep}^i - F^2}{F^2} \quad \text{(4.3.143)}
\]

Now, we are going to prove \( S_B^O \subseteq S_B^J \) and \( S_A^J \subseteq S_A^O \).

First, we will prove that \( S_B^O \subseteq S_B^J \). Assume \( s_i \in S_B^O \). Then, we have \( (a_{ef}^i > 0 \text{ and } f_O(s_i) - \frac{a_{ep}^i}{a_{ef}^i} > 0) \) after (4.3.101). Since \( a_{ep}^i \geq 0 \), we have \( \frac{a_{ep}^i}{a_{ef}^i} \geq 0 \). Therefore, \( f_O(s_i) > 0 \). Then from Equation (4.3.143), we have \( (a_{ef}^i F + a_{ef}^i a_{ep}^i - F^2) > 0 \) because \( F^2 > 0 \). It follows from Lemma 4.3.1 that \( F^2 \geq F a_{ef}^i \).

Then, from Equations (4.3.143) and (4.3.142), we have \( f_J(s_i) \geq f_O(s_i) \). As a consequence, we have \( f_J(s_i) - \frac{a_{ef}^i}{a_{ef}^i} \geq f_O(s_i) - \frac{a_{ep}^i}{a_{ef}^i} > 0 \). It follows from (4.3.14) that \( s_i \in S_B^J \). Thus, \( S_B^O \subseteq S_B^J \).

Next, we will prove \( S_A^J \subseteq S_A^O \). Assume \( s_i \in S_A^J \). Then, we have either \( (a_{ef}^i = 0) \) or \( (a_{ef}^i > 0 \text{ and } f_J(s_i) - \frac{a_{ef}^i}{a_{ef}^i} < 0) \) after (4.3.16).

- Consider the case that \( (a_{ef}^i = 0) \). Immediately, we have \( s_i \in S_A^O \) after (4.3.103).

- Consider the case that \( (a_{ef}^i > 0 \text{ and } f_J(s_i) - \frac{a_{ef}^i}{a_{ef}^i} < 0) \). Assume further that \( f_J(s_i) < 0 \). Since \( F a_{ef}^i > 0 \), we have \( (a_{ef}^i F + a_{ef}^i a_{ep}^i - F^2) < 0 \) from Equation (4.3.142). Then, \( f_O(s_i) < 0 \) from Equation (4.3.143) because \( F^2 > 0 \). As a consequence, \( f_O(s_i) - \frac{a_{ep}^i}{a_{ef}^i} < 0 \). Hence, \( s_i \in S_A^O \) after (4.3.103). Next consider the sub-case that \( f_J(s_i) = 0 \). Then, \( (a_{ef}^i F + a_{ef}^i a_{ep}^i - F^2) = 0 \). Thus we have \( f_O(s_i) = f_J(s_i) = 0 \) from Equation (4.3.143). Furthermore, since \( f_J(s_i) - \frac{a_{ep}^i}{a_{ef}^i} < 0 \) and \( f_J(s_i) = 0 \), we have \( \frac{a_{ep}^i}{a_{ef}^i} > 0 \). As a consequence, \( f_O(s_i) - \frac{a_{ep}^i}{a_{ef}^i} < 0 \). Hence, \( s_i \in S_A^O \) after (4.3.103). Finally, consider the sub-case that \( f_J(s_i) > 0 \). Since \( F a_{ef}^i > 0 \), we have \( (a_{ef}^i F + a_{ef}^i a_{ep}^i - F^2) > 0 \). It follows from Lemma 4.3.1 that \( F^2 \geq F a_{ef}^i \). Then, from Equations (4.3.143) and (4.3.142), we have \( f_J(s_i) \geq f_O(s_i) \). As a consequence, we have \( f_O(s_i) - \frac{a_{ep}^i}{a_{ef}^i} \leq f_J(s_i) - \frac{a_{ep}^i}{a_{ef}^i} < 0 \). Thus, \( s_i \in S_A^O \).
after (4.3.103).

In summary, we have proved that $S^I_A \subseteq S^O_A$.

In conclusion, we have $S^O_B \subseteq S^I_B$ and $S^I_A \subseteq S^O_A$. Immediately after Theorem 4.2.2, Ochiai $\rightarrow$ Jaccard. And after Proposition 4.3.2, Ochiai $\rightarrow$ ER2.

**Proposition 4.3.10.** Kulczynski2 $\rightarrow$ Ochiai.

**Proof.** As proved in Section 4.3.3, $S^O_B$ and $S^O_A$ are equal to the sets defined in (4.3.101) and (4.3.103), respectively; and $S^K_B$ and $S^K_A$ are equal to the sets defined in (4.3.87) and (4.3.89), respectively, as follows.

$S^K_B = \{ s_i | a^i_{ef} > 0 \}$ and $\frac{a^i_{ef}F + a^i_{ep}a^f_{ep} - F^2}{F^2 + (F + a^i_{ep})(F - a^i_{ef}) - a^i_{ep}} > 0, 1 \leq i \leq n \}

S^K_A = \{ s_i | (a^i_{ef} = 0) \text{ or } (a^i_{ef} > 0 \text{ and } \frac{a^i_{ef}F + a^i_{ep}a^f_{ep} - F^2}{F^2 + (F + a^i_{ep})(F - a^i_{ef}) - a^i_{ep}} < 0), 1 \leq i \leq n \}

Let $f_{K2}$ denotes the following expression.

$$f_{K2}(s_i) = \frac{a^i_{ef}F + a^i_{ep}a^f_{ep} - F^2}{F^2 + (F + a^i_{ep})(F - a^i_{ef})} \quad (4.3.144)$$

Now, we are going to prove $S^K_B \subseteq S^O_B$ and $S^O_A \subseteq S^K_A$.

First, we will prove $S^K_B \subseteq S^O_B$. Assume $s_i \in S^K_B$. Then, we have $(a^i_{ef} > 0 \text{ and } f_{K2}(s_i) - \frac{a^i_{ep}}{a^i_{ef}} > 0)$ after (4.3.87). Since $a^i_{ep} > 0$, we have $\frac{a^i_{ep}}{a^i_{ef}} > 0$. Therefore, $f_{K2}(s_i) > 0$. Then, from Equation (4.3.144), we have $(a^i_{ef}F + a^i_{ep}a^f_{ep} - F^2) > 0$ because $(F^2 + (F + a^i_{ep})(F - a^i_{ef})) > 0$ after Lemma 4.3.1. It also follows from Lemma 4.3.1 that $F^2 + (F + a^i_{ep})(F - a^i_{ef}) \geq F^2 > 0$, thus from Equations (4.3.144) and (4.3.143), we have $f_O(s_i) \geq f_{K2}(s_i)$. As a consequence, we have $f_O(s_i) - \frac{a^i_{ep}}{a^i_{ef}} \geq f_{K2}(s_i) - \frac{a^i_{ep}}{a^i_{ef}} > 0$. It follows from (4.3.101) that $s_i \in S^O_B$. Thus, $S^K_B \subseteq S^O_B$.

Next, we will prove $S^O_A \subseteq S^K_A$. Assume $s_i \in S^O_A$. Then, we have either $(a^i_{ef} = 0)$ or $(a^i_{ef} > 0 \text{ and } f_{O}(s_i) - \frac{a^i_{ep}}{a^i_{ef}} < 0)$ after (4.3.103).

- Consider the case that $(a^i_{ef} = 0)$. It follows immediately from (4.3.89) that $s_i \in S^K_A$.
- Consider the case that $(a^i_{ef} > 0 \text{ and } f_{O}(s_i) - \frac{a^i_{ep}}{a^i_{ef}} < 0)$. Assume further that $f_{O}(s_i) < 0$. Since $F^2 > 0$, we have $(a^i_{ef}F + a^i_{ep}a^f_{ep} - F^2) < 0$ from Equation (4.3.143). Then, $f_{K2}(s_i) < 0$ from Equation (4.3.144) because $(F^2 + (F + a^i_{ep})(F - a^i_{ef})) > 0$. As a consequence, $f_{K2}(s_i) - \frac{a^i_{ep}}{a^i_{ef}} < 0$. 

Hence, \( s_i \in S^K_2 \) after (4.3.89). Next consider the sub-case that \( f_O(s_i) = 0 \). Then, 
\((a^i_{ef}F + a^i_{ep}a^p_{ep} - F^2) = 0 \). Thus we have \( f_{K2}(s_i) = f_O(s_i) = 0 \) from Equation (4.3.144).

Furthermore, since \( f_O(s_i) - \frac{a^i_{ep}}{a^i_{ef}} < 0 \) and \( f_O(s_i) = 0 \), we have \( \frac{a^i_{ep}}{a^i_{ef}} > 0 \). As a consequence, 
\( f_{K2}(s_i) - \frac{a^i_{ep}}{a^i_{ef}} < 0 \). Thus, \( s_i \in S^K_2 \) after (4.3.89). Finally, consider the sub-case that 
\( f_O(s_i) > 0 \). Since \( F^2 > 0 \), we have \((a^i_{ef}F + a^i_{ep}a^p_{ep} - F^2) > 0 \). It follows from Lemma 4.3.1 that 
\( F^2 + (F + a^i_{ep})(F - a^i_{ef}) \geq F^2 > 0 \). Thus, from Equations (4.3.144) and (4.3.143), we have 
\( f_O(s_i) \geq f_{K2}(s_i) \). As a consequence, we have \( f_{K2}(s_i) - \frac{a^i_{ep}}{a^i_{ef}} < f_O(s_i) - \frac{a^i_{ep}}{a^i_{ef}} < 0 \). Thus, we have 
\( s_i \in S^K_2 \) after (4.3.89).

In summary, we have proved that \( S^Q_A \subseteq S^K_2 \).

In conclusion, we have \( S^K_2 \subseteq S^Q_B \) and \( S^Q_A \subseteq S^K_2 \). Immediately after Theorem 4.2.2, Kulczynski2 \( \rightarrow \) Ochiai.

Following from Propositions 4.3.7 to 4.3.10, we have Kulczynski2 \( \rightarrow \) Ochiai \( \rightarrow \) ER2 \( \rightarrow \) ER3 and 
ER2 \( \rightarrow \) ER4. Obviously, if Kulczynski2 is not a maximal formula, Ochiai, ER2, ER3 and ER4 can never be maximal. Thus, we need not to consider Ochiai, ER2, ER3 and ER4 as potential candidates for the maximal formulas for the time being, unless Kulczynski2 is proved to be maximal.

**Proposition 4.3.11.** M2 \( \rightarrow \) AMPL2.

**Proof.** As proved in Section 4.3.3, \( S^M_2 \) and \( S^M_2 \) are equal to the sets defined in (4.3.94) and (4.3.96), respectively, as follows.

\[
S^M_2 = \{ s_i | a^i_{ef} > 0 \text{ and } \frac{P + a^i_{ep}}{F} - \frac{2F + P}{a^i_{ef}} + 2 - \frac{a^i_{ep}}{a^i_{ef}} > 0, 1 \leq i \leq n \} \\
S^M_A = \{ s_i | (a^i_{ef} = 0) \text{ or } (a^i_{ef} > 0 \text{ and } \frac{P + a^i_{ep}}{F} - \frac{2F + P}{a^i_{ef}} + 2 - \frac{a^i_{ep}}{a^i_{ef}} < 0), 1 \leq i \leq n \}
\]

And \( S^A_B \) and \( S^A_A \) are equal to the sets defined in (4.3.108) and (4.3.110), respectively, as follows.

\[
S^A_B = \{ s_i | a^i_{ef} > 0 \text{ and } \frac{P a^i_{ef} - PF + Fa^i_{ep}}{Fa^i_{ef}} - \frac{a^i_{ep}}{a^i_{ef}} > 0, 1 \leq i \leq n \} \\
S^A_A = \{ s_i | (a^i_{ef} = 0) \text{ or } (a^i_{ef} > 0 \text{ and } \frac{P a^i_{ef} - PF + Fa^i_{ep}}{Fa^i_{ef}} - \frac{a^i_{ep}}{a^i_{ef}} < 0), 1 \leq i \leq n \}
\]

If \( a^i_{ef} > 0 \), the expression \( \frac{P + a^i_{ep}}{F} - \frac{2F + P}{a^i_{ef}} + 2 - \frac{a^i_{ep}}{a^i_{ef}} \) from (4.3.94) and (4.3.96) can be re-written as
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Immediately after Proposition 4.3.11, we know that if $M_2$ is not a maximal formula, AMPLE2 cannot be a maximal formula. Thus, we can exclude AMPLE2 as a potential candidate for maximal formula for the time being, unless $M_2$ is proved to be maximal. With the discussion immediately after the proof of Proposition 4.3.10, for the time being, we only need to search the maximal formulas from ER1, Kulczynski2, M2, ER6, Wong3, Arithmetic Mean, Cohen, Fleiss and ER5.

In the following seven propositions, we find that ER1 is better than Kulczynski2, M2, ER6, Wong3, Arithmetic Mean, Cohen and Fleiss.

\[
\frac{P+a_{ef}^f}{F} - 2\frac{F}{a_{ef}^i} + 2 - \frac{a_{ef}^i}{a_{ef}^i} = \left(\frac{P}{F} - a_{ef}^i + \frac{a_{ef}^f}{a_{ef}^f} - \frac{a_{ef}^i}{a_{ef}^i} + 2\right) + \frac{a_{ef}^f, 2F - a_{ef}^f}{a_{ef}^i}
\]

Then, we have

\[
\frac{P+a_{ef}^f}{F} - 2\frac{F}{a_{ef}^i} + 2 - \frac{a_{ef}^i}{a_{ef}^i} \leq \frac{P a_{ef}^i - P F + F a_{ef}^f}{F a_{ef}^f} - \frac{a_{ef}^i}{a_{ef}^i}
\]

because \(\frac{(a_{ef}^i + 2F)(a_{ef}^i - F)}{F a_{ef}^f}\) \(\leq 0\) after Lemma 4.3.1.

Now, we are going to prove \(S_B^M \subseteq S_B^A\) and \(S_A^M \subseteq S_A^M\).

First, we will prove \(S_B^M \subseteq S_B^A\). Assume \(s_i \in S_B^M\). Then, we have \(a_{ef, i} > 0\) and \(P + a_{ef}^f - 2\frac{F}{a_{ef}^i} + 2 - \frac{a_{ef}^i}{a_{ef}^i} > 0\) after (4.3.94). As a consequence, we have \(P a_{ef}^i - P F + F a_{ef}^f - \frac{a_{ef}^i}{a_{ef}^i} > 0\) from (4.3.145). Thus, \(s_i \in S_B^A\) after (4.3.108). Therefore, \(S_B^M \subseteq S_B^A\).

Secondly, we will prove \(S_A^M \subseteq S_A^A\). Assume \(s_i \in S_A^M\). Then, we have either \(a_{ef, i} = 0\) or \(a_{ef, i} > 0\) and \(P a_{ef}^i - P F + F a_{ef}^f - \frac{a_{ef}^i}{a_{ef}^i} < 0\) after (4.3.110).

- Consider the case that \(a_{ef, i} = 0\). Immediately after (4.3.96), \(s_i \in S_A^M\).
- Consider the case that \(a_{ef, i} > 0\) and \(P a_{ef}^i - P F + F a_{ef}^f - \frac{a_{ef}^i}{a_{ef}^i} < 0\). Then, we have \(P + a_{ef}^f - 2\frac{F}{a_{ef}^i} + 2 - \frac{a_{ef}^i}{a_{ef}^i} < 0\) after (4.3.145). Thus, \(s_i \in S_A^M\) after (4.3.96).

In summary, we have proved that \(S_A^M \subseteq S_A^M\).

In conclusion, we have \(S_B^M \subseteq S_B^M\) and \(S_A^M \subseteq S_A^M\). Immediately after Theorem 4.2.2, \(M_2 \rightarrow \) AMPLE2.

Immediately after Proposition 4.3.11, we know that if M2 is not a maximal formula, AMPLE2 can never be a maximal formula. Thus, we can exclude AMPLE2 as a potential candidate for maximal formula for the time being, unless M2 is proved to be maximal. With the discussion immediately after the proof of Proposition 4.3.10, for the time being, we only need to search the maximal formulas from ER1, Kulczynski2, M2, ER6, Wong3, Arithmetic Mean, Cohen, Fleiss and ER5.

In the following seven propositions, we find that ER1 is better than Kulczynski2, M2, ER6, Wong3, Arithmetic Mean, Cohen and Fleiss.
Proposition 4.3.12. ER1 → Kulczynski2.

Proof. In order to prove ER1 → Kulczynski2, it is sufficient to prove Op1 → Kulczynski2. As proved in Section 4.3.3, $S_{B}^{K2}$ and $S_{A}^{K2}$ are equal to the sets defined in (4.3.87) and (4.3.89), respectively; and $S_{B}^{Op1}$ and $S_{A}^{Op1}$ are equal to the sets defined in (4.3.4) and (4.3.6), respectively, as follows.

\[
S_{B}^{Op1} = \{ s_{i} | a_{ep}^{i} = F \text{ and } a_{ep}^{f} - a_{ep}^{i} > 0, 1 \leq i \leq n \} \\
S_{A}^{Op1} = \{ s_{i} | (a_{ep}^{i} < F) \text{ or } (a_{ep}^{f} = F \text{ and } a_{ep}^{f} - a_{ep}^{i} < 0), 1 \leq i \leq n \} 
\]

We are going to prove $S_{B}^{Op1} \subseteq S_{B}^{K2}$ and $S_{A}^{K2} \subseteq S_{A}^{Op1}$.

First, we will prove $S_{B}^{Op1} \subseteq S_{B}^{K2}$. Assume $s_{i} \in S_{B}^{Op1}$. Then, $a_{ep}^{i} = F > 0$ and $(a_{ep}^{f} - a_{ep}^{i}) > 0$ after (4.3.4). As a consequence, we have $a_{ep}^{i} F + a_{ep}^{f} a_{ep}^{i} - F^2 - F a_{ep}^{i} = a_{ep}^{i} - a_{ep}^{f} > 0$. Therefore, $s_{i} \in S_{B}^{K2}$ after (4.3.87). Thus, $S_{B}^{Op1} \subseteq S_{B}^{K2}$.

Secondly, we are going to prove $S_{A}^{K2} \subseteq S_{A}^{Op1}$. Suppose $s_{i} \in S_{A}^{K2}$. Then we have either $(a_{ep}^{i} = 0)$ or $(a_{ep}^{f} > 0$ and $a_{ep}^{i} F + a_{ep}^{f} a_{ep}^{i} - F^2 - F a_{ep}^{i} = a_{ep}^{i} - a_{ep}^{f} < 0$) after (4.3.89).

- Consider the case that $(a_{ep}^{i} = 0)$. Obviously, $a_{ep}^{i} < F$. Immediately after (4.3.6), $s_{i} \in S_{A}^{Op1}$.

- Consider the case that $(a_{ep}^{i} > 0$ and $a_{ep}^{i} F + a_{ep}^{f} a_{ep}^{i} - F^2 - F a_{ep}^{i} = a_{ep}^{i} - a_{ep}^{f} < 0)$. Assume further that $0 < a_{ep}^{i} < F$. After (4.3.6), we have $s_{i} \in S_{A}^{Op1}$. Next, consider the sub-case that $a_{ep}^{i} = F$. Then we have $a_{ep}^{i} F + a_{ep}^{f} a_{ep}^{i} - F^2 - F a_{ep}^{i} = a_{ep}^{i} - a_{ep}^{f} < 0$. Since $a_{ep}^{i} F + a_{ep}^{f} a_{ep}^{i} - F^2 - F a_{ep}^{i} = a_{ep}^{i} a_{ep}^{f} < 0$ and $F > 0$, we have $(a_{ep}^{f} - a_{ep}^{i}) < 0$. Thus, $s_{i} \in S_{A}^{Op1}$ after (4.3.6).

In summary, we have proved that $S_{A}^{K2} \subseteq S_{A}^{Op1}$.

In conclusion, we have $S_{B}^{Op1} \subseteq S_{B}^{K2}$ and $S_{A}^{K2} \subseteq S_{A}^{Op1}$. Immediately after Theorem 4.2.2, Op1 → Kulczynski2. And after Proposition 4.3.1, ER1 → Kulczynski2. \hfill \Box

Proposition 4.3.13. ER1 → M2.

Proof. In order to prove ER1 → M2, it is sufficient to prove Op1 → M2. As proved in Section 4.3.3, $S_{B}^{Op1}$ and $S_{A}^{Op1}$ are equal to the sets defined in (4.3.4) and (4.3.6), respectively; and $S_{B}^{M2}$ and $S_{A}^{M2}$ are equal to the sets defined in (4.3.94) and (4.3.96), respectively.

We are going to prove $S_{B}^{Op1} \subseteq S_{B}^{M2}$ and $S_{A}^{M2} \subseteq S_{A}^{Op1}$.

First, we will prove $S_{B}^{Op1} \subseteq S_{B}^{M2}$. Assume $s_{i} \in S_{B}^{Op1}$. Then we have $a_{ep}^{i} F > 0$ and $(a_{ep}^{f} - a_{ep}^{i}) > 0$.
after (4.3.4). As a consequence, we have

\[ \frac{P+a^f_{ep} - 2F+P}{F} - a^i_{ef} + 2 - \frac{a^i_{ef}}{a^i_{ef}} = \frac{P+a^f_{ep} - 2F-P+2F - a^i_{ef}}{F} - a^i_{ep} - a^i_{ep} > 0 \]

Therefore, \( s_i \in S_B^{M2} \) after (4.894). Thus, \( S_B^{Op1} \subseteq S_B^{M2} \).

Secondly, we are going to prove \( S_A^{M2} \subseteq S_A^{Op1} \). Suppose \( s_i \in S_A^{M2} \). Then we have either \( a^i_{ef} = 0 \) or \( a^i_{ef} > 0 \) and \( \frac{P+a^f_{ep}}{a^i_{ef}} - 2 \frac{F+P}{a^i_{ef}} + 2 - \frac{a^i_{ef}}{a^i_{ef}} < 0 \) after (4.896).

- Consider the case that \( a^i_{ef} = 0 \). Obviously, \( a^i_{ef} < F \). Immediately after (4.3.6), \( s_i \in S_A^{Op1} \).

- Consider the case that \( a^i_{ef} > 0 \) and \( \frac{P+a^f_{ep}}{a^i_{ef}} - 2 \frac{F+P}{a^i_{ef}} + 2 - \frac{a^i_{ef}}{a^i_{ef}} < 0 \). Assume further that \( 0 < a^i_{ef} < F \). After (4.3.6), we have \( s_i \in S_A^{Op1} \). Next, consider the sub-case that \( a^i_{ef} = F \). Then, we have \( \frac{P+a^f_{ep}}{a^i_{ef}} - 2 \frac{F+P}{a^i_{ef}} + 2 - \frac{a^i_{ef}}{a^i_{ef}} = a^f_{ef} - a^i_{ef} \). Since \( \frac{P+a^f_{ep}}{a^i_{ef}} - 2 \frac{F+P}{a^i_{ef}} + 2 - \frac{a^i_{ef}}{a^i_{ef}} < 0 \) and \( F > 0 \), we have \( a^f_{ef} - a^i_{ef} < 0 \). Thus, \( s_i \in S_A^{Op1} \) after (4.3.6).

In summary, we have proved that \( S_A^{M2} \subseteq S_A^{Op1} \).

In conclusion, we have \( S_B^{Op1} \subseteq S_B^{M2} \) and \( S_A^{M2} \subseteq S_A^{Op1} \). Immediately after Theorem 4.2.2, Op1 \( \rightarrow \) M2. And after Proposition 4.3.1, ER1 \( \rightarrow \) M2.

\[ \square \]

**Proposition 4.3.14.** ER1 \( \rightarrow \) ER6.

**Proof.** In order to prove ER1 \( \rightarrow \) ER6, it is sufficient to prove Op1 \( \rightarrow \) Scott. As proved in Section 4.3.3, \( S_B^{Op1} \) and \( S_A^{Op1} \) are equal to the sets defined in (4.3.4) and (4.3.6), respectively; and \( S_B^{SC} \) and \( S_A^{SC} \) are equal to the sets defined in (4.3.78) and (4.3.80), respectively, as follows.

\[
S_B^{SC} = \{ s_i \} \left[ \frac{-F^2+4a^i_{ef}P+2Fa^i_{ef} - 2Fa^i_{ep} - (a^i_{ep} + a^i_{ef})^2}{(F+2P-a^i_{ep} - a^i_{ef})(F+a^i_{ef} + a^i_{ep})} + 4PF - 4Fa^i_{ep} - (a^i_{ef})^2}{(2F+a^i_{ef})(2P-a^i_{ep})} \right] \leq i \leq n \}
\]

\[
S_A^{SC} = \{ s_i \} \left( \frac{-F^2+4a^i_{ef}P+2Fa^i_{ef} - 2Fa^i_{ep} - (a^i_{ep} + a^i_{ef})^2}{(F+2P-a^i_{ep} - a^i_{ef})(F+a^i_{ef} + a^i_{ep})} + 4PF - 4Fa^i_{ep} - (a^i_{ef})^2}{(2F+a^i_{ef})(2P-a^i_{ep})} \right) \leq i \leq n \}
\]

If \( a^i_{ef} = F \), we have

\[
\frac{-F^2+4a^i_{ef}P+2Fa^i_{ef} - 2Fa^i_{ep} - (a^i_{ep} + a^i_{ef})^2}{(F+2P-a^i_{ep} - a^i_{ef})(F+a^i_{ef} + a^i_{ep})} + 4PF - 4Fa^i_{ep} - (a^i_{ef})^2}{(2F+a^i_{ef})(2P-a^i_{ep})}
\]

\[
= \frac{4PF - 4Fa^i_{ep} - (a^i_{ep})^2}{(2F+a^i_{ef})(2P-a^i_{ep})} + \frac{4PF - 4Fa^i_{ep} - (a^i_{ef})^2}{(2F+a^i_{ef})(2P-a^i_{ep})}
\]

\[
= \frac{8PF^2+8P^2F+2a^i_{ep}F+2Fa^i_{ep}a^i_{ef} + 2Fa^i_{ep}a^i_{ef}}{(2F+a^i_{ep})(2P-a^i_{ep})(2P-a^i_{ep})}
\]

\[
(4.3.146)
\]
Now, we are going to prove $S_{B}^{Op1} \subseteq S_{B}^{SC}$ and $S_{A}^{SC} \subseteq S_{A}^{Op1}$.

First, we will prove $S_{B}^{Op1} \subseteq S_{B}^{SC}$. Assume $s_{i} \in S_{B}^{Op1}$. Then, $a_{i}^{f} = F$ and $(a_{i}^{e} - a_{i}^{p})>0$ after (4.3.4). It follows from Lemma 4.3.1 that $(8PF^{2}+8P^{2}F+2Pa_{i}^{f}a_{i}^{e}p+2Fa_{i}^{f}a_{i}^{e}p)>0$ and $(2F+a_{i}^{e})(2P-a_{i}^{p})(2F+a_{i}^{f})(2P-a_{i}^{p})>0$, then we have

$$\frac{(8PF^{2}+8P^{2}F+2Pa_{i}^{f}a_{i}^{e}p+2Fa_{i}^{f}a_{i}^{e}p)}{(2F+a_{i}^{e})(2P-a_{i}^{p})(2F+a_{i}^{f})(2P-a_{i}^{p})}>0$$

because $(a_{i}^{e} - a_{i}^{p})>0$. From Equation (4.3.146), we have

$$\frac{-F^{2}+4a_{i}^{f}P+2Fa_{i}^{f}a_{i}^{e}p-2Fa_{i}^{e}p-(a_{i}^{e}+a_{i}^{f})^{2}}{(F+2P-a_{i}^{e}-a_{i}^{f})(F+a_{i}^{e}+a_{i}^{p})} \leq \frac{4PF-4Fa_{i}^{e}p-(a_{i}^{f})^{2}}{(2F+a_{i}^{f})(2P-a_{i}^{f})} \text{ after (4.3.78). Thus, } S_{B}^{Op1} \subseteq S_{B}^{SC}$.

Secondly, we are going to prove $S_{A}^{SC} \subseteq S_{A}^{Op1}$. Suppose $s_{i} \in S_{A}^{SC}$. Then we have

$$\frac{-F^{2}+4a_{i}^{f}P+2Fa_{i}^{f}a_{i}^{e}p-2Fa_{i}^{e}p-(a_{i}^{e}+a_{i}^{f})^{2}}{(F+2P-a_{i}^{e}-a_{i}^{f})(F+a_{i}^{e}+a_{i}^{p})} \leq \frac{4PF-4Fa_{i}^{e}p-(a_{i}^{f})^{2}}{(2F+a_{i}^{f})(2P-a_{i}^{f})} \text{ after (4.3.80).}$$

- Suppose $(a_{i}^{f} < F)$. Immediately after (4.3.6), $s_{i} \in S_{A}^{Op1}$.

- Suppose $(a_{i}^{f} = F)$. It follows from

$$\frac{-F^{2}+4a_{i}^{f}P+2Fa_{i}^{f}a_{i}^{e}p-2Fa_{i}^{e}p-(a_{i}^{e}+a_{i}^{f})^{2}}{(F+2P-a_{i}^{e}-a_{i}^{f})(F+a_{i}^{e}+a_{i}^{p})} \leq \frac{4PF-4Fa_{i}^{e}p-(a_{i}^{f})^{2}}{(2F+a_{i}^{f})(2P-a_{i}^{f})}$$

and Equation (4.3.146) that

$$\frac{(8PF^{2}+8P^{2}F+2Pa_{i}^{f}a_{i}^{e}p+2Fa_{i}^{f}a_{i}^{e}p)}{(2F+a_{i}^{f})(2P-a_{i}^{f})}<0,$$

which implies $(a_{i}^{e} - a_{i}^{p})<0$, because $(8PF^{2}+8P^{2}F+2Pa_{i}^{f}a_{i}^{e}p+2Fa_{i}^{f}a_{i}^{e}p)>0$ and $(2F+a_{i}^{e})(2P-a_{i}^{p})(2F+a_{i}^{f})(2P-a_{i}^{p})>0$ after Lemma 4.3.1. Thus, $s_{i} \in S_{A}^{Op1}$ after (4.3.6).

In summary, we have proved that $S_{A}^{SC} \subseteq S_{A}^{Op1}$.

In conclusion, we have $S_{B}^{Op1} \subseteq S_{B}^{SC}$ and $S_{A}^{SC} \subseteq S_{A}^{Op1}$. Immediately after Theorem 4.2.2, Op1 $\rightarrow$ Scott. And after Propositions 4.3.1 and 4.3.6, ER1 $\rightarrow$ ER6.

**Proposition 4.3.15.** ER1 $\rightarrow$ Wong3.

**Proof.** In order to prove ER1 $\rightarrow$ Wong3, it is sufficient to prove Op1 $\rightarrow$ Wong3. As proved in Section 4.3.3, $S_{B}^{Op1}$ and $S_{A}^{Op1}$ are equal to the sets defined in (4.3.4) and (4.3.6), respectively; and for Wong3, the definitions of $S_{B}^{W3}$ and $S_{A}^{W3}$ vary in three situations: $a_{i}^{f} < 2 < a_{i}^{p}$, $2 < a_{i}^{p} \leq 10$, and $a_{i}^{p} > 10$. Under each of these situations, we are going to prove that $S_{B}^{Op1} \subseteq S_{B}^{W3}$ and $S_{A}^{W3} \subseteq S_{A}^{Op1}$.
1. **Case 1:** Assume \( a_{ep}^i \leq 2 \).

As proved in Section 4.3.3, \( S_B^{W_3} \) and \( S_A^{W_3} \) are equal to the sets defined in (4.3.115) and (4.3.117), respectively, as follows.

\[
\begin{align*}
S_B^{W_3} &= \{ s_i | a_{ep}^i \leq 2 \text{ and } (a_{ef}^i - F) + (a_{ep}^i - a_{ep}^i) > 0, 1 \leq i \leq n \} \\
S_A^{W_3} &= \{ s_i | (a_{ep}^i > 2) \text{ or } (a_{ep}^i \leq 2 \text{ and } (a_{ef}^i - F) + (a_{ep}^i - a_{ep}^i) < 0), 1 \leq i \leq n \}
\end{align*}
\]

First, we will prove \( S_B^{Op1} \subseteq S_B^{W_3} \). Assume \( s_i \in S_B^{Op1} \). Then, \( (a_{ef}^i = F) \) and \( (a_{ep}^i - a_{ep}^i) > 0 \) after (4.3.4). As a consequence, \( (a_{ef}^i - F) + (a_{ep}^i - a_{ep}^i) = (a_{ep}^i - a_{ep}^i) > 0 \). Furthermore, since \( a_{ep}^i \leq 2 \), we have \( a_{ep}^i < a_{ep}^i \leq 2 \). Then, \( s_i \in S_B^{W_3} \) after (4.3.115). Thus, \( S_B^{Op1} \subseteq S_B^{W_3} \).

Secondly, we are going to prove \( S_A^{W_3} \subseteq S_A^{Op1} \). Assume \( s_i \in S_B^{W_3} \). Then we have either \( a_{ep}^i > 2 \), or \( (a_{ep}^i \leq 2 \text{ and } (a_{ef}^i - F) + (a_{ep}^i - a_{ep}^i) < 0) \) after (4.3.117).

- Consider the case that \( a_{ep}^i > 2 \). Assume further that \( a_{ep}^i < F \). Immediately, we have \( s_i \in S_B^{Op1} \) after (4.3.6). Then consider the sub-case that \( a_{ep}^i = F \). Since \( a_{ep}^i > 2 \) and \( a_{ep}^i \leq 2 \), we have \( (a_{ep}^i - a_{ep}^i) < 0 \). Thus after (4.3.6), \( s_i \in S_B^{Op1} \).

- Consider the case that \( a_{ep}^i \leq 2 \) and \( (a_{ef}^i - F) + (a_{ep}^i - a_{ep}^i) < 0 \). Assume further that \( a_{ep}^i < F \). Then, \( s_i \in S_B^{Op1} \) after (4.3.6). Now consider the sub-case that \( a_{ep}^i = F \). We have \( (a_{ef}^i - F) + (a_{ep}^i - a_{ep}^i) = (a_{ep}^i - a_{ep}^i) \). Since \( (a_{ef}^i - F) + (a_{ep}^i - a_{ep}^i) < 0 \), then \( (a_{ep}^i - a_{ep}^i) < 0 \). After (4.3.6), \( s_i \in S_B^{Op1} \).

In summary, we have proved that \( S_B^{W_3} \subseteq S_B^{Op1} \).

2. **Case 2:** Assume \( 2 < a_{ep}^i \leq 10 \).

As proved in Section 4.3.3, \( S_B^{W_3} \) and \( S_A^{W_3} \) are equal to the sets defined in (4.3.121) and (4.3.123), respectively, as follows.

\[
\begin{align*}
S_B^{W_3} &= \{ s_i | (a_{ep}^i \leq 2 \text{ and } (a_{ef}^i - F) + (0.1 a_{ep}^i - a_{ep}^i) + 1.8 > 0) \text{ or } \} \\
&\quad \quad (2 < a_{ep}^i \leq 10 \text{ and } (a_{ef}^i - F) + (0.1 a_{ep}^i - 0.1 a_{ep}^i) > 0), 1 \leq i \leq n \} \\
S_A^{W_3} &= \{ s_i | (a_{ep}^i \leq 2 \text{ and } (a_{ef}^i - F) + (0.1 a_{ep}^i - a_{ep}^i) + 1.8 > 0) \text{ or } \} \\
&\quad \quad (2 < a_{ep}^i \leq 10 \text{ and } (a_{ef}^i - F) + (0.1 a_{ep}^i - 0.1 a_{ep}^i) < 0) \text{ or } \} \\
&\quad \quad (a_{ep}^i > 10), 1 \leq i \leq n \}
\end{align*}
\]

First, we will prove \( S_B^{Op1} \subseteq S_B^{W_3} \). Assume \( s_i \in S_B^{Op1} \). Then, \( a_{ef}^i = F \) and \( (a_{ep}^i - a_{ep}^i) > 0 \) after (4.3.4). Since \( 2 < a_{ep}^i \leq 10 \), we have \( a_{ep}^i < 10 \). Consider the following two cases:
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

In summary, we have proved that $S_A^{O^1} \subseteq S_B^{W^3}$. Secondly, we are going to prove $S_B^{W^3} \subseteq S_A^{O^1}$. Assume $s_i \in S_B^{W^3}$. Then, we have either ($a_{i ep}^i > 10$), ($2 < a_{i ep}^i \leq 10$ and ($a_{i ep}^i - F) + 0.1(a_{i ep}^f - a_{i ep}^i) < 0$), or ($a_{i ep}^i < 2$ and ($a_{i ep}^i - F) + (0.1a_{i ep}^f - a_{i ep}^i) + 1.8 < 0$).

• Consider the case that $a_{i ep}^i > 10$. Assume further $a_{i ep}^i F$. Immediately after (4.3.6), we have $s_i \in S_A^{O^1}$. Then consider the sub-case that $a_{i ep}^i = F$. Since $2 < a_{i ep}^i \leq 10$ and $a_{i ep}^i = 10$, we have ($a_{i ep}^f - a_{i ep}^i < 0$). After (4.3.6), $s_i \in S_A^{O^1}$. Then, we have ($a_{i ep}^i F$). Then, we have ($a_{i ep}^i - F) + 0.1(a_{i ep}^f - a_{i ep}^i) = 0.1(a_{i ep}^f - a_{i ep}^i)$. Since ($a_{i ep}^i - F) + 0.1(a_{i ep}^f - a_{i ep}^i) < 0$, then ($a_{i ep}^f - a_{i ep}^i < 0$). After (4.3.6), $s_i \in S_A^{O^1}$.

• Consider the case that $a_{i ep}^i < 2$ and ($a_{i ep}^i - F) + (0.1a_{i ep}^f - a_{i ep}^i) + 1.8 < 0$). Assume further that $a_{i ep}^i = F$. Then, we have ($a_{i ep}^i - F) + (0.1a_{i ep}^f - a_{i ep}^i) + 1.8 = 0.1a_{i ep}^f - a_{i ep}^i + 1.8 < 0$. However, it follows from $2 < a_{i ep}^i \leq 10$ and $a_{i ep}^i < 2$ that $0.1a_{i ep}^f - a_{i ep}^i + 1.8 > 0$, which is contradictory to $0.1a_{i ep}^f - a_{i ep}^i + 1.8 < 0$. Therefore, it is impossible to have $a_{i ep}^i = F$ and all statements in this case have $a_{i ep}^i < F$. Then, we have $s_i \in S_A^{O^1}$ after (4.3.6).

In summary, we have proved that $S_A^{W^3} \subseteq S_A^{O^1}$.

3. Case 3: Assume $a_{i ep}^i > 10$.

As proved in Section 4.3.3, $S_B^{W^3}$ and $S_A^{W^3}$ are equal to the sets defined in (4.3.127) and (4.3.129), respectively, as follows.

$$S_B^{W^3} = \{s_i | (a_{i ep}^i \leq 2 \text{ and } (a_{i ep}^i - F) + (0.001a_{i ep}^f - a_{i ep}^i) + 2.79 > 0) \text{ or}$$
$$2 < a_{i ep}^i \leq 10 \text{ and } (a_{i ep}^i - F) + (0.001a_{i ep}^f - 0.1a_{i ep}^i) + 0.99 > 0) \text{ or}$$
$$a_{i ep}^i > 10 \text{ and } (a_{i ep}^i - F) + (0.001a_{i ep}^f - 0.001a_{i ep}^i) > 0), 1 \leq i \leq n\}$$

$$S_A^{W^3} = \{s_i | (a_{i ep}^i \leq 2 \text{ and } (a_{i ep}^i - F) + (0.001a_{i ep}^f - a_{i ep}^i) + 2.79 < 0) \text{ or}$$
$$2 < a_{i ep}^i \leq 10 \text{ and } (a_{i ep}^i - F) + (0.001a_{i ep}^f - 0.1a_{i ep}^i) + 0.99 < 0) \text{ or}$$

First, we will prove $S_B^{Op1} \subseteq S_B^{W^3}$. Assume $s_i \in S_B^{Op1}$. Then, we have $a_{ef}^i = F$ and $(a_{ep}^i - a_{cp}^i) > 0$ after (4.3.4). Since $a_{ep}^i > 10$, $a_{ep}^i$ can be any value within $[0, P]$. Now, let us consider the following cases:

- Suppose $a_{ep}^i \leq 2$. Since $a_{ef}^i = F$, we have
  \[ (a_{ef}^i - F) + (0.001a_{ep}^i - a_{ep}^i) + 2.79 = 0.001(a_{ep}^i - a_{ep}^i) + (2.79 - 0.999a_{ep}^i) > 0 \]
  because $(a_{ep}^i - a_{ep}^i) > 0$ and $(2.79 - 0.999a_{ep}^i) > 0$ after $a_{ep}^i \leq 2$. After (4.3.127), $s_i \in S_B^{W^3}$.

- Suppose $2 < a_{ep}^i \leq 10$. Since $a_{ef}^i = F$, we have
  \[ (a_{ef}^i - F) + (0.001a_{ep}^i - 0.1a_{ep}^i) + 0.99 = 0.001(a_{ep}^i - a_{ep}^i) + (0.99 - 0.099a_{ep}^i) > 0 \]
  because $(a_{ep}^i - a_{ep}^i) > 0$ and $(0.99 - 0.099a_{ep}^i) \geq 0$ after $2 < a_{ep}^i \leq 10$. After (4.3.127), $s_i \in S_B^{W^3}$.

- Suppose $a_{ep}^i > 10$. Since $a_{ef}^i = F$, we have
  \[ (a_{ef}^i - F) + 0.001(a_{ep}^i - a_{ep}^i) = 0.001(a_{ep}^i - a_{ep}^i) > 0 \]
  because $(a_{ep}^i - a_{ep}^i) > 0$. Thus, $s_i \in S_B^{W^3}$ after (4.3.127).

In summary, we have proved that $S_B^{Op1} \subseteq S_B^{W^3}$.

Secondly, we will prove $S_A^{W^3} \subseteq S_A^{Op1}$. Assume $s_i \in S_A^{W^3}$. Then we have either $(a_{ep}^i \leq 2$ and $(a_{ef}^i - F) + (0.001a_{ep}^i - a_{ep}^i) + 2.79 < 0$, $(2 < a_{ep}^i \leq 10$ and $(a_{ef}^i - F) + (0.001a_{ep}^i - 0.1a_{ep}^i) + 0.99 < 0)$, or $(a_{ep}^i > 10$ and $(a_{ef}^i - F) + 0.001(a_{ep}^i - a_{ep}^i) < 0)$.

- Consider the case that $a_{ep}^i \leq 2$ and $(a_{ef}^i - F) + (0.001a_{ep}^i - a_{ep}^i) + 2.79 < 0$. Assume further $a_{ef}^i = F$. Then, we have
  \[ (a_{ef}^i - F) + (0.001a_{ep}^i - a_{ep}^i) + 2.79 = 0.001a_{ep}^i - a_{ep}^i + 2.79 < 0 \]
  However, it follows from $a_{ep}^i > 10$ and $a_{ep}^i < 2$ that $0.001a_{ep}^i - a_{ep}^i + 2.79 > 0.8$, which is contradictory to $0.001a_{ep}^i - a_{ep}^i + 2.79 < 0$. Therefore, it is impossible to have $a_{ef}^i = F$ and all statements in this case have $a_{ef}^i = F$. Then, we have $s_i \in S_A^{Op1}$ after (4.3.6).

- Consider the case that $(2 < a_{ep}^i \leq 10$ and $(a_{ef}^i - F) + (0.001a_{ep}^i - 0.1a_{ep}^i) + 0.99 < 0)$. Assume further $a_{ef}^i = F$. Then, we have
  \[ (a_{ef}^i - F) + (0.001a_{ep}^i - 0.1a_{ep}^i) + 0.99 = 0.001a_{ep}^i - 0.1a_{ep}^i + 0.99 < 0 \]
  However, it follows from $a_{ep}^i > 10$ and $2 < a_{ep}^i \leq 10$ that $0.001a_{ep}^i - 0.1a_{ep}^i + 0.99 > 0$, which is
In order to prove ER1 respectively; and

In conclusion, for any value of $\alpha^f$, we have $S^{Op1}_B \subseteq S^{W3}_B$ and $S^{W3}_A \subseteq S^{Op1}_A$. It follows from Theorem 4.2.2 that Op1 → Wong3. Therefore, ER1 → Wong3, after Proposition 4.3.1.

**Proposition 4.3.16.** ER1 → Arithmetic Mean.

**Proof.** In order to prove ER1 → Arithmetic Mean, it is sufficient to prove Op1 → Arithmetic Mean. As proved in Section 4.3.3, $S^{Op1}_B$ and $S^{Op1}_A$ are equal to the sets defined in (4.3.4) and (4.3.6), respectively; and $S^{AM}_B$ and $S^{AM}_A$ are equal to the sets defined in (4.3.130) and (4.3.132), respectively, as follows.

$$
S^{AM}_B = \{ s_i | \frac{\alpha^f - \alpha^i_{ep}}{\alpha^i_{ep} + (F + P - \alpha^i_{ep})} + PF > \frac{PF - Fa^f_{ep}}{(F + \alpha^f_{ep})(P - \alpha^i_{ep} + PF)}, 1 \leq i \leq n \} \\
S^{AM}_A = \{ s_i | \frac{\alpha^f - \alpha^i_{ep}}{\alpha^i_{ep} + (F + P - \alpha^i_{ep})} + PF < \frac{PF - Fa^f_{ep}}{(F + \alpha^f_{ep})(P - \alpha^i_{ep} + PF)}, 1 \leq i \leq n \}
$$

If $\alpha^i_{ef} = F$, we have

$$
\frac{\alpha^f - \alpha^i_{ep}}{\alpha^i_{ep} + (F + P - \alpha^i_{ep})} + PF = \frac{PF - Fa^f_{ep}}{(F + \alpha^f_{ep})(P - \alpha^i_{ep} + PF)} \\
= \frac{(F + \alpha^f_{ep})(P - \alpha^i_{ep}) + PF}{(F + \alpha^f_{ep})(P - \alpha^i_{ep} + PF)}\frac{PF - Fa^f_{ep}}{(F + \alpha^f_{ep})(P - \alpha^i_{ep} + PF)} \\
= \frac{F[(P - \alpha^f_{ep})(P - \alpha^i_{ep}) + PF] - (a^f_{ep} - a^i_{ep})}{(F + \alpha^f_{ep})(P - \alpha^i_{ep} + PF)[(F + \alpha^f_{ep})(P - \alpha^i_{ep} + PF)]}
$$

(4.3.147)

Now, we are going to prove $S^{Op1}_B \subseteq S^{AM}_B$ and $S^{AM}_A \subseteq S^{Op1}_A$.

First, we will prove $S^{Op1}_B \subseteq S^{AM}_B$. Assume $s_i \in S^{Op1}_B$. Then, $\alpha^i_{ef} = F$ and $(\alpha^f_{ep} - \alpha^i_{ep}) > 0$ after (4.3.4). It follows from Lemma 4.3.1 that $F[(P - \alpha^i_{ep})(P - \alpha^i_{ep}) + PF] > 0$, $(F + \alpha^f_{ep})(P - \alpha^i_{ep} + PF) > 0$ and $(F + \alpha^f_{ep})(P - \alpha^i_{ep}) + PF > 0$, then we have $a^i_{ef} F - \alpha^i_{ep}$$ \frac{a^f_{ep} - \alpha^i_{ep}}{a^i_{ep} + (P + F - a^i_{ep})} + PF > \frac{PF - Fa^f_{ep}}{(F + \alpha^f_{ep})(P - \alpha^i_{ep} + PF)} > 0$ because $(\alpha^f_{ep} - \alpha^i_{ep}) > 0$. From Equation (4.3.147), we have $\frac{a^f_{ep} - \alpha^i_{ep}}{a^i_{ep} + (P + F - a^i_{ep}) + PF} > \frac{PF - Fa^f_{ep}}{(F + \alpha^f_{ep})(P - \alpha^i_{ep} + PF)}$. Therefore, $s_i \in S^{AM}_B$ after (4.3.130). Thus, $S^{Op1}_B \subseteq S^{AM}_B$. 

contradictory to $0.001a^f_{ep} - 0.1a^i_{ep} + 0.99 < 0$. Therefore, it is impossible to have $\alpha^i_{ef} = F$ and all statements in this case have $\alpha^i_{ef} < F$. Then, we have $s_i \in S^{Op1}_A$ after (4.3.6).

- Consider the case that $(\alpha^i_{ep} > 10$ and $(\alpha^i_{ef} = F) + 0.001(\alpha^f_{ep} - \alpha^i_{ep}) < 0$. Assume further $\alpha^i_{ef} < F$. Then, $s_i \in S^{Op1}_A$ after (4.3.6). Now consider the sub-case that $\alpha^i_{ef} = F$. Then, we have $(\alpha^i_{ef} = F) + 0.001(\alpha^f_{ep} - \alpha^i_{ep}) = 0.001(\alpha^f_{ep} - \alpha^i_{ep})$. Since $(\alpha^i_{ef} = F) + 0.001(\alpha^f_{ep} - \alpha^i_{ep}) < 0$, then $(\alpha^f_{ep} - \alpha^i_{ep}) < 0$. Therefore, $s_i \in S^{Op1}_A$ after (4.3.6).

In summary, we have proved that $S^{W3}_A \subseteq S^{Op1}_A$.
Secondly, we are going to prove \( S_{AM}^A \subseteq S_{Op}^A \). Suppose \( s_i \in S_{AM}^A \). Then we have 
\[
\frac{a_{ij}^F - a_{ij}^P}{(a_{ij}^F + a_{ij}^P)(P + F - a_{ij}^F - a_{ij}^P) + PF} < \frac{PF - Fa_{ij}^F}{(F + a_{ij}^F)(P - a_{ij}^F) + PF}
\]
after (4.3.132).

- Suppose \((a_{ij}^F) < F\). Immediately after (4.3.6), \( s_i \in S_{Op}^A \).

- Suppose \((a_{ij}^F) = F\). It follows from 
\[
\frac{a_{ij}^F - a_{ij}^P}{(a_{ij}^F + a_{ij}^P)(P + F - a_{ij}^F - a_{ij}^P) + PF} < \frac{PF - Fa_{ij}^F}{(F + a_{ij}^F)(P - a_{ij}^F) + PF}
\]
and Equation (4.3.147) that 
\[
\frac{F(P - a_{ij}^F)(P - a_{ij}^P) + PF}{(F + a_{ij}^F)(P - a_{ij}^F) + PF} < 0,
\]
which implies 
\[
(a_{ij}^F - a_{ij}^P) < 0
\]
and 
\[
(F + a_{ij}^F)(P - a_{ij}^F) + PF > 0
\]
after Lemma 4.3.1. Thus, \( s_i \in S_{Op}^A \) after (4.3.6).

In summary, we have proved that \( S_{AM}^A \subseteq S_{Op}^A \).

In conclusion, we have \( S_{Op}^B \subseteq S_{AM}^B \) and \( S_{AM}^A \subseteq S_{Op}^A \). Immediately after Theorem 4.2.2, \( Op1 \rightarrow \text{Arithmetic Mean} \). And after Proposition 4.3.1, \( ER1 \rightarrow \text{Arithmetic Mean} \).

**Proposition 4.3.17.** \( ER1 \rightarrow Cohen \).

*Proof.* In order to prove \( ER1 \rightarrow Cohen \), it is sufficient to prove \( Op1 \rightarrow Cohen \). As proved in Section 4.3.3, \( S_{Op}^B \) and \( S_{Op}^A \) are equal to the sets defined in (4.3.4) and (4.3.6), respectively; and \( S_{B}^C \) and \( S_{A}^C \) are equal to the sets defined in (4.133) and (4.135), respectively, as follows.

\[
S_B^C = \{ s_i | \frac{a_{ij}^F - a_{ij}^P}{P(a_{ij}^F + a_{ij}^P) + F(P + F - a_{ij}^F - a_{ij}^P)} > \frac{PF - Fa_{ij}^F}{P(F + a_{ij}^F) + F(P - a_{ij}^F)}, 1 \leq i \leq n \}
\]

\[
S_A^C = \{ s_i | \frac{a_{ij}^F - a_{ij}^P}{P(a_{ij}^F + a_{ij}^P) + F(P + F - a_{ij}^F - a_{ij}^P)} < \frac{PF - Fa_{ij}^F}{P(F + a_{ij}^F) + F(P - a_{ij}^F)}, 1 \leq i \leq n \}
\]

If \( a_{ij}^F = F \), we have

\[
\frac{a_{ij}^F - a_{ij}^P}{P(a_{ij}^F + a_{ij}^P) + F(P + F - a_{ij}^F - a_{ij}^P)} < \frac{PF - Fa_{ij}^F}{P(F + a_{ij}^F) + F(P - a_{ij}^F)}
\]

\[
= \frac{PF - F(a_{ij}^F - a_{ij}^P)}{P(F + a_{ij}^F) + F(P - a_{ij}^F)}
\]

\[
= \frac{F(P + P^2)(a_{ij}^F - a_{ij}^P)}{[P(F + a_{ij}^F) + F(P - a_{ij}^F)][P(F + a_{ij}^F) + F(P - a_{ij}^F)]}
\]

(4.3.148)

Now, we are going to prove \( S_{Op}^B \subseteq S_{B}^C \) and \( S_{A}^C \subseteq S_{Op}^A \).

First, we will prove \( S_{Op}^B \subseteq S_{B}^C \). Assume \( s_i \in S_{Op}^B \). Then, \( a_{ij}^F = F \) and \( (a_{ij}^F - a_{ij}^P) > 0 \) after (4.3.4). It follows from Lemma 4.3.1 that 
\( F(P + P^2) > 0 \),
\( P(F + a_{ij}^F) + F(P - a_{ij}^F) > 0 \) and
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

\[ P(F + a_{fp}^f) + F(P - a_{fp}^f) > 0, \text{ then we have} \]
\[ \frac{F(PP + P^2)(a_{fp}^e - a_{fp}^e)}{(P + a_{fp}^e + F)(P + a_{fp}^e + F)} > 0 \text{ because} \]
\[ (a_{fp}^f - a_{fp}^e) > 0. \text{ From Equation (4.3.148), we have} \]
\[ \frac{a_{fp}^f - a_{fp}^e}{P(a_{fp}^f + a_{fp}^e) + F(P + F - a_{fp}^e - a_{fp}^e)} > \frac{P - F}{P(F + a_{fp}^e) + F(P - a_{fp}^e)}. \]

Therefore, \( s_i \in S_B^C \) after (4.3.133). Thus, \( S_B^{Op1} \subseteq S_B^C \).

Secondly, we are going to prove \( S_A^C \subseteq S_A^{Op1} \). Suppose \( s_i \in S_A^C \). Then we have
\[ \frac{a_{fp}^f - a_{fp}^e}{P(a_{fp}^f + a_{fp}^e) + F(P + F - a_{fp}^e - a_{fp}^e)} < \frac{P - F}{P(F + a_{fp}^e) + F(P - a_{fp}^e)} \text{ after (4.3.155).} \]

- Suppose \( (a_{fp}^f < F) \). Immediately after (4.3.6), \( s_i \in S_A^{Op1} \).
- Suppose \( (a_{fp}^f = F) \). It follows from
\[ \frac{a_{fp}^f - a_{fp}^e}{P(a_{fp}^f + a_{fp}^e) + F(P + F - a_{fp}^e - a_{fp}^e)} < \frac{P - F}{P(F + a_{fp}^e) + F(P - a_{fp}^e)} \]
and Equation (4.3.148) that
\[ \frac{P(F + F^2)(a_{fp}^e - a_{fp}^e)}{(P + a_{fp}^e + F)(P + a_{fp}^e + F)} < 0, \text{ which implies} \]
\( (a_{fp}^f - a_{fp}^e) < 0 \) because \( F(F + P^2) > 0, \ P(a_{fp}^e) + F(P - a_{fp}^e) > 0 \)
and \( P(F + a_{fp}^e) + F(P - a_{fp}^e) > 0 \) after Lemma 4.3.1. Thus, \( s_i \in S_A^{Op1} \) after (4.3.6).

In summary, we have proved that \( S_A^C \subseteq S_A^{Op1} \).

In conclusion, we have \( S_B^{Op1} \subseteq S_B^C \) and \( S_A^C \subseteq S_A^{Op1} \). Immediately after Theorem 4.2.2, Op1 \( \rightarrow \) Cohen. And after Proposition 4.3.1, ER1 \( \rightarrow \) Cohen.

**Proposition 4.3.18.** ER1 \( \rightarrow \) Fleiss.

**Proof.** In order to prove ER1 \( \rightarrow \) Fleiss, it is sufficient to prove Op1 \( \rightarrow \) Fleiss. As proved in Section 4.3.3, \( S_B^{Op1} \) and \( S_A^{Op1} \) are equal to the sets defined in (4.3.4) and (4.3.6), respectively; and \( S_B^F \) and \( S_A^F \) are equal to the sets defined in (4.3.136) and (4.3.138), respectively, as follows.

\[ S_B^F = \{s_i \mid -F^2 + 4a_{fp}^f P + 2F a_{fp}^f - 2F a_{ep}^f - (a_{ep}^e + a_{fp}^e)^2 > 4PF - 4Fa_{ep}^f - (a_{fp}^e)^2, 1 \leq i \leq n\} \]
\[ S_A^F = \{s_i \mid -F^2 + 4a_{fp}^f P + 2F a_{fp}^f - 2F a_{ep}^f - (a_{ep}^e + a_{fp}^e)^2 < 4PF - 4Fa_{ep}^f - (a_{fp}^e)^2, 1 \leq i \leq n\} \]

If \( a_{fp}^f = F \), we have
\[ [-F^2 + 4a_{fp}^f P + 2F a_{fp}^f - 2F a_{ep}^f - (a_{ep}^e + a_{fp}^e)^2] - [4PF - 4Fa_{ep}^f - (a_{fp}^e)^2] \]
\[ = [4PF - 4Fa_{ep}^f - (a_{fp}^e)^2] - [4PF - 4Fa_{ep}^f - (a_{fp}^e)^2] \]
\[ = (4F + a_{fp}^f + a_{ep}^e)(a_{fp}^f - a_{ep}^e) \] (4.3.149)

Now, we are going to prove \( S_B^{Op1} \subseteq S_B^F \) and \( S_A^F \subseteq S_A^{Op1} \).

First, we will prove \( S_B^{Op1} \subseteq S_B^F \). Assume \( s_i \in S_B^{Op1} \). Then, \( a_{fp}^f = F \) and \( (a_{fp}^f - a_{fp}^e) > 0 \)
after (4.3.4). It follows from Lemma 4.3.1 that \( 4F + a_{fp}^f + a_{ep}^e > 0 \), then we have
\[ (4F + a_{fp}^f + a_{ep}^e)(a_{fp}^f - a_{fp}^e) > 0 \text{ because} \ (a_{fp}^f - a_{fp}^e) > 0. \] From Equation (4.3.149), we have
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

\[-F^2+4a_{ef}^jP+2Fa_{ef}^j-2Fa_{ep}^i-(a_{ep}^i+a_{ef}^i)^2 > 4PF-4Fa_{ep}^i-(a_{ep}^i)^2.\] Therefore, \(s_i \in S_B^F\) after (4.3.136). Thus, \(S_{Op1}^F \subseteq S_B^F.\)

Secondly, we are going to prove \(S_A^F \subseteq S_A^{Op1}.\) Suppose \(s_i \in S_A^F.\) Then we have

\[-F^2+4a_{ef}^jP+2Fa_{ef}^j-2Fa_{ep}^i-(a_{ep}^i+a_{ef}^i)^2 < 4PF-4Fa_{ep}^i-(a_{ep}^i)^2\] after (4.3.138).

- Suppose \((a_{ef}^i < F).\) Immediately after (4.3.6), \(s_i \in S_A^{Op1}.\)
- Suppose \((a_{ef}^i = F).\) It follows from \(-F^2+4a_{ef}^jP+2Fa_{ef}^j-2Fa_{ep}^i-(a_{ep}^i+a_{ef}^i)^2 < 4PF-4Fa_{ep}^i-(a_{ep}^i)^2\) and Equation (4.3.149) that \((4F+a_{ef}^j+a_{ep}^i)(a_{ep}^i-a_{ep}^j) < 0,\) which implies \((a_{ep}^i-a_{ep}^j) < 0\) after Lemma 4.3.1. Thus, \(s_i \in S_A^{Op1}\) after (4.3.6).

In summary, we have proved that \(S_A^F \subseteq S_A^{Op1}.\)

In conclusion, we have \(S_B^{Op1} \subseteq S_B^F\) and \(S_A^{Op1} \subseteq S_A^F.\) Immediately after Theorem 4.2.2, \(Op1 \rightarrow Fleiss.\) And after Proposition 4.3.1, \(ER1 \rightarrow Fleiss.\) □

Following Propositions 4.3.12 to 4.3.18, which state \(ER1 \rightarrow R,\) where \(R\) stands for formula Kulczynski2, M2, ER6, Wong3, Arithmetic Mean, Cohen or Fleiss, we can conclude that if \(R \rightarrow ER1\) does not hold, \(R\) cannot be maximal irrespective whether \(ER1\) is maximal or not. Thus, in the following proposition, we are going to show that \(R\) is not maximal by proving that \(R \rightarrow ER1\) does not hold.

**Proposition 4.3.19.** Formulas Kulczynski2, M2, ER6, Wong3, Arithmetic Mean, Cohen and Fleiss are not maximal formulas.

*Proof.* Consider a sample program \(PG1\) in Figure 4.1, where \(s_5\) is the faulty statement. Table 4.2 lists the \(A_i\) for \(PG1\) with respect to a test suite \(TS_1.\) As a reminder, data in Table 4.2 are feasible. First, they comply with Lemmas 4.3.1 and 4.3.2. Secondly, the entry statement \(s_1\) has \((a_{ef}^1 = 0)\) and \((a_{ep}^1 = 0).\) Thirdly, for any \(s_i\) in Figure 4.1, the value of element \(a_{ef}^i\) or \(a_{ep}^i\) is equal to the sum of the corresponding element contributed by all of its directly preceding statements, and also equal to the sum of its contribution to all of its directly succeeding statements.

Then, for \(PG1\) with \(TS_1, S_B^{R1}, S_F^{R1}\) and \(S_A^{R1}\) for \(ER1\) are shown as the scenario A in Table 4.4; while the corresponding sets for formulas Kulczynski2, M2, ER6, Wong3, Arithmetic Mean, Cohen and Fleiss are shown as the scenario B in Table 4.4. Then, using any consistent tie-breaking scheme, the \(EXAM\) score of \(ER1\) is less than the \(EXAM\) scores of the other formulas. As a consequence, we have demonstrated that \(R \rightarrow ER1\) does not hold, where \(R\) is Kulczynski2, M2, ER6, Wong3, Arithmetic Mean, Cohen or Fleiss. Thus, the proposition is proved. □

With Proposition 4.3.19, the only remaining candidates for maximal formulas are \(ER1\) and \(ER5.\)
### Table 4.2: $A_i$ for $PG_1$ with $TS_1$

<table>
<thead>
<tr>
<th>Statement</th>
<th>$A_i = \langle a_{ef}, a_{ep}, a_{nf}, a_{np} \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>$&lt;40, 160, 0, 0&gt;$</td>
</tr>
<tr>
<td>$s_2$</td>
<td>$&lt;0, 40, 40, 120&gt;$</td>
</tr>
<tr>
<td>$s_3$</td>
<td>$&lt;40, 120, 0, 40&gt;$</td>
</tr>
<tr>
<td>$s_4$</td>
<td>$&lt;0, 40, 40, 120&gt;$</td>
</tr>
<tr>
<td>$s_5$</td>
<td>$&lt;40, 80, 0, 80&gt;$</td>
</tr>
<tr>
<td>$s_6$</td>
<td>$&lt;39, 1, 1, 159&gt;$</td>
</tr>
<tr>
<td>$s_7$</td>
<td>$&lt;1, 79, 39, 91&gt;$</td>
</tr>
<tr>
<td>$s_8$</td>
<td>$&lt;40, 80, 0, 80&gt;$</td>
</tr>
</tbody>
</table>

### Table 4.3: $A_i$ for $PG_2$ with $TS_2$

<table>
<thead>
<tr>
<th>Statement</th>
<th>$A_i = \langle a_{ef}, a_{ep}, a_{nf}, a_{np} \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>$&lt;40, 160, 0, 0&gt;$</td>
</tr>
<tr>
<td>$s_2$</td>
<td>$&lt;0, 70, 40, 90&gt;$</td>
</tr>
<tr>
<td>$s_3$</td>
<td>$&lt;40, 90, 0, 70&gt;$</td>
</tr>
<tr>
<td>$s_4$</td>
<td>$&lt;0, 30, 40, 130&gt;$</td>
</tr>
<tr>
<td>$s_5$</td>
<td>$&lt;40, 60, 0, 100&gt;$</td>
</tr>
<tr>
<td>$s_6$</td>
<td>$&lt;40, 30, 0, 130&gt;$</td>
</tr>
<tr>
<td>$s_7$</td>
<td>$&lt;0, 30, 40, 130&gt;$</td>
</tr>
<tr>
<td>$s_8$</td>
<td>$&lt;40, 30, 0, 130&gt;$</td>
</tr>
<tr>
<td>$s_9$</td>
<td>$&lt;40, 30, 0, 130&gt;$</td>
</tr>
<tr>
<td>$s_{10}$</td>
<td>$&lt;40, 60, 0, 100&gt;$</td>
</tr>
</tbody>
</table>

### Table 4.4: Sets for different combinations of formula and test suite

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>$S_R^B$</th>
<th>$S_R^F$</th>
<th>$S_R^A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$\emptyset$</td>
<td>{s_5, s_8}</td>
<td>{s_1, s_2, s_3, s_4, s_6, s_7}</td>
</tr>
<tr>
<td>B</td>
<td>{s_6}</td>
<td>{s_5, s_8}</td>
<td>{s_1, s_2, s_3, s_4, s_7}</td>
</tr>
<tr>
<td>C</td>
<td>$\emptyset$</td>
<td>{s_1, s_3, s_5, s_8}</td>
<td>{s_2, s_4, s_6, s_7}</td>
</tr>
<tr>
<td>D</td>
<td>{s_6, s_8, s_9}</td>
<td>{s_5, s_{10}}</td>
<td>{s_1, s_2, s_3, s_4, s_7}</td>
</tr>
<tr>
<td>E</td>
<td>$\emptyset$</td>
<td>{s_1, s_3, s_5, s_6, s_8, s_9, s_{10}}</td>
<td>{s_2, s_4, s_7}</td>
</tr>
</tbody>
</table>
4.3. EFFECTIVENESS COMPARISON OF RISK EVALUATION FORMULAS

Proposition 4.3.20. ER1 and ER5 are maximal formulas.

Proof. First, we will prove that ER5 → ER1 does not hold. Consider PG₁ with TS₁ in Table 4.2. As shown in the above Proposition 4.3.19, the relevant sets for ER1 are as the scenario A in Table 4.4; while for ER5, they are as the scenario C in Table 4.4. If we adopt the ORIGINAL ORDER tie-breaking scheme, which is a consistent tie-breaking scheme and ranks all statements in Sᵣ according to their original order in program, the EXAM score of ER5 is greater than the EXAM score of ER1. After Definition 4.2.3, ER5 → ER1 does not hold.

Secondly, we will prove that ER1 → ER5 does not hold either. Consider another sample program PG₂ shown in Figure 4.2, where s₅ is the faulty statement. Table 4.3 gives the Aᵢ for PG₂ with respect to another test suite TS₂. Similar to the discussion in Proposition 4.3.19 for the data in Table 4.2, data in Table 4.3 are also feasible. For PG₂ with TS₂, the relevant sets for ER1 are as the scenario D in Table 4.4; while for ER5, they are as the scenario E in Table 4.4. If we adopt the ORIGINAL ORDER tie-breaking scheme, the EXAM score of ER1 is greater than the EXAM score of ER5. After Definition 4.2.3, ER1 → ER5 does not hold.

The above examples demonstrate that neither ER1 → ER5 nor ER5 → ER1 holds. With Propositions 4.3.7 to 4.3.19, we can conclude that ER1 and ER5 are the only maximal formulas among all the 30 investigated formulas.
4.4 Performance refinement on risk evaluation formulas

As discussed in previous sections, a good risk evaluation formula shall rank the faulty statements high in the risk list. Generally speaking, a reasonable formula should comply with two major intuitions, as follows.

1. Statements associated with more failed and less passed testing results should have higher risk values.

2. Statements with $a_i^{ef}=0$ should be assigned with risk values lower than those of the statements $a_i^{ef}>0$.

The first intuition is obvious, and is adopted by almost all the existing formulas. But the second intuition is not so obvious. Actually, it is also important to comply with this intuition, because $a_i^{ef}=0$ means that the corresponding $s_i$ has never been covered by any failed test case. Since these statements never trigger any failure executions with current test suite, they should be effectively regarded as “inactivated faulty statements”, even if they are actually faulty. Therefore, only the statements with $a_i^{ef}>0$ have potential to be the “activated faulty statements”. Obviously, a reasonable debugging procedure will focus on the “activated faulty statements”, due to the supportive information revealed by the testing results. While for the “inactivated faulty statements”, though they cannot be considered as fault-free, they should only be investigated when they have actually revealed some failures and
become activated. Therefore, an effective risk evaluation formula should assign lower risk values to statements with \( a_{e f}^i = 0 \) as compared to statements with \( a_{e f}^i > 0 \). Then, all \( s_i \) with \( a_{e f}^i = 0 \) will be ranked at the bottom of the final ranking list. Under the assumption of “perfect bug detection”, these statements will never be diagnosed [Xie et al., 2010].

However, this second intuition is not adopted by every risk evaluation formula. For example, formula Wong2 does not comply with this intuition. For \( s_j \) with \( a_{e f}^i > 0 \), we have \( R_{W 2}(s_j) = a_{e f}^i - a_{e p}^i \), and for \( s_i \) with \( a_{e f}^i = 0 \), we have \( R_{W 2}(s_i) = a_{e f}^i - a_{e p}^i = -a_{e p}^i \). Obviously, it is possible to have \(-a_{e p}^i > a_{e f}^i - a_{e p}^i\).

In order to make such formulas comply with the second intuition, we propose a refinement solution [Xie et al., 2010]. In our solution, we first categorize all statements into two groups: the suspicious group and unsuspicious group. The suspicious group consists of all \( s_i \) with \( a_{e f}^i > 0 \), which obviously contains the “activated faulty statements”. And the unsuspicious group contains the remaining statements with \( a_{e f}^i = 0 \) that can never be the “activated faulty statements”. Given a formula \( R \) that does not comply with the second intuition, we can define its refinement \( R' \) in the following way:

\[
R'(s_i) = \begin{cases} 
R(s_i) & \text{if } a_{e f}^i > 0 \\
\text{MIN}_RISK - 1 & \text{if } a_{e f}^i = 0
\end{cases}
\] (4.4.1)

where \( \text{MIN}_RISK = \min\{R(s_i)|a_{e f}^i > 0, 1 \leq i \leq n\} \).

According to the definition, \( R' \) only evaluates the risk values with its original formula \( R \) for the statements in the suspicious group; meanwhile assigns all statements in the unsuspicious group with the lowest risk value. Therefore, after being refined by this method, any formula \( R \) that does not comply with the second intuition becomes to comply with this intuition. And one benefit of using the refined formula \( R' \) is that we may have a performance improvement. It is not difficult to prove that \( R \) and \( R' \) have the following relation.

**Proposition 4.4.1.** For any risk formula \( R \) and its refinement \( R' \) defined according to Equation (4.4.1), we have \( R' \rightarrow R \).

**Proof.** After the definition of \( R' \) and Definition 4.2.1, we have

\[
\begin{align*}
S_B^{R'} &= S_B^R \setminus \{s_i|a_{e f}^i = 0 \text{ and } R(s_i) > R(s_f), 1 \leq i \leq n\} \\
S_F^{R'} &= S_F^R \setminus \{s_i|a_{e f}^i = 0 \text{ and } R(s_i) = R(s_f), 1 \leq i \leq n\} \\
S_A^{R'} &= S_A^R \cup \{s_i|a_{e f}^i = 0 \text{ and } R(s_i) > R(s_f), 1 \leq i \leq n\} \cup \{s_i|a_{e f}^i = 0 \text{ and } R(s_i) = R(s_f), 1 \leq i \leq n\}
\end{align*}
\]

Therefore, we have \( S_B^{R'} \subseteq S_B^R \) and \( S_A^{R'} \subseteq S_A^R \). Immediately after Theorem 4.2.2, \( R' \rightarrow R \). \(\square\)
Among all of our investigated formulas, re, Binary, ER4 and Wong3 do not comply with the second intuition.

First, for formula re, as stated in the discussion after Proposition 4.3.6, it is possible that \( s_i \) with \( a_{ef}^i=0 \) has risk value higher than that of the \( s_i \) with \( a_{ef}^i>0 \). Thus, re does not comply with the second intuition. Similarly, for Binary defined in Table 4.1, the risk values of \( s_i \) with \( a_{ef}^i=0 \) and \( s_i \) with \( 0<a_{ef}^i<F \) are both 0. Thus, Binary does not comply with the second intuition either. However, for re, we have proved that the risk values of all \( s_i \) with \( a_{ef}^i=0 \) are always lower than the risk value of \( s_f \). And for Binary, obviously, the risk values of all \( s_i \) with \( a_{ef}^i=0 \), which are 0, are always lower than the risk value of \( s_f \), which is 1. Therefore, from Definition 4.2.1, though these two formulas do not comply with the second intuition, every \( s_i \) with \( a_{ef}^i=0 \) still belongs to their \( S_A^i \). As a consequence, the refinement only adjusts the risk values for statements in \( S_A^i \), without changing the membership of \( S_B^R \), \( S_F^R \) or \( S_A^R \). Thus, we have \( S_B^R=S_B^{R'} \), \( S_F^R=S_F^{R'} \) and \( S_A^R=S_A^{R'} \). Therefore, we have \( R' \leftrightarrow R \) after Theorem 4.2.3, that is, the refinement does not result in a performance improvement for Binary or re. As a consequence, applying the refinement method on these two formulas will not affect our conclusion about the maximal formulas.

Next, for formulas in ER4, as proved in Proposition 4.3.4, \( S_B^{R'} \), \( S_F^{R'} \) and \( S_A^{R'} \) of each formula \( R \) in ER4 are equal to the sets defined in (4.3.48), (4.3.49) and (4.3.50), respectively, as follows.

\[
S_B^{R'} = \{ s_i | (a_{ef}^i-F) + (a_{ep}^i-a_{ep}^i) > 0, 1 \leq i \leq n \} \\
S_F^{R'} = \{ s_i | (a_{ef}^i-F) + (a_{ep}^i-a_{ep}^i) = 0, 1 \leq i \leq n \} \\
S_A^{R'} = \{ s_i | (a_{ef}^i-F) + (a_{ep}^i-a_{ep}^i) < 0, 1 \leq i \leq n \}
\]

Consider an \( s_i \) with \( a_{ef}^i=0 \). Then, we have \((a_{ef}^i-F) + (a_{ep}^i-a_{ep}^i) = -F+a_{ep}^i-a_{ep}^i\). Consider a different \( s_j \) with \( a_{ef}^j>0 \). Obviously, it is possible to have both \((-F+a_{ep}^j-a_{ep}^i>0)\) and \((a_{ef}^j-F) + (a_{ep}^j-a_{ep}^j)<0\). Therefore, it is possible that \( s_i \) with \( a_{ef}^i=0 \) has risk value higher than that of the \( s_j \) with \( a_{ef}^j>0 \), which is against the second intuition. Moreover, as shown in this example, it is possible to have \( s_i \) with \( a_{ef}^i=0 \) as an element of \( S_B^R \) for these formulas. In other words, we can find examples to demonstrate \( \{ s_i | a_{ef}^i=0 \text{ and } R(s_i)>R(s_j), 1 \leq i \leq n \} \neq \emptyset \). From the proof of Proposition 4.4.1, after the refinement, we may have \( S_B^{R'} \subseteq S_B^{R} \). Since \( S_F^{R'} \subseteq S_F^{R} \), we have \( E_R > E_R' \) if the tie-breaking scheme is consistent. It follows from Definition 4.2.3 that for formulas in ER4, \( R \rightarrow R' \) does not hold. That is \( R' \rightarrow R \) is a strictly better relation for these formulas.

Similar conclusions can be obtained for Wong3. Let us consider the case that \( a_{ep}^i>10 \) as an example. Then, as discussed in Section 4.3.3, \( S_B^{W3} \), \( S_F^{W3} \) and \( S_A^{W3} \) of Wong3 are equal to sets defined in (4.3.127), (4.3.128) and (4.3.129), respectively, as follows.
4.4. PERFORMANCE REFINEMENT ON RISK EVALUATION FORMULAS

\[ S_B^{W_3} = \{ s_i | (a_{e,f}^i \leq 2 \text{ and } (a_{e,f}^i - F) + (0.001a_{e,p}^i - a_{e,p}^i) + 2.79 > 0) \text{ or } \]
\[ (2 < a_{e,p}^i \leq 10 \text{ and } (a_{e,f}^i - F) + (0.001a_{e,p}^i - 0.1a_{e,p}^i) + 0.99 > 0) \text{ or } \]
\[ (a_{e,p}^i > 10 \text{ and } (a_{e,f}^i - F) + (0.001a_{e,p}^i - 0.001a_{e,p}^i) > 0), 1 \leq i \leq n \} \]

\[ S_F^{W_3} = \{ s_i | (a_{e,f}^i \leq 2 \text{ and } (a_{e,f}^i - F) + (0.001a_{e,p}^i - a_{e,p}^i) + 2.79 = 0) \text{ or } \]
\[ (2 < a_{e,p}^i \leq 10 \text{ and } (a_{e,f}^i - F) + (0.001a_{e,p}^i - 0.1a_{e,p}^i) + 0.99 = 0) \text{ or } \]
\[ (a_{e,p}^i > 10 \text{ and } (a_{e,f}^i - F) + (0.001a_{e,p}^i - 0.001a_{e,p}^i) = 0), 1 \leq i \leq n \} \]

\[ S_A^{W_3} = \{ s_i | (a_{e,f}^i \leq 2 \text{ and } (a_{e,f}^i - F) + (0.001a_{e,p}^i - a_{e,p}^i) + 2.79 < 0) \text{ or } \]
\[ (2 < a_{e,p}^i \leq 10 \text{ and } (a_{e,f}^i - F) + (0.001a_{e,p}^i - 0.1a_{e,p}^i) + 0.99 < 0) \text{ or } \]
\[ (a_{e,p}^i > 10 \text{ and } (a_{e,f}^i - F) + (0.001a_{e,p}^i - 0.001a_{e,p}^i) < 0), 1 \leq i \leq n \} \]

Consider an \( s_i \) with \( a_{e,f}^i = 0 \) and \( a_{e,p}^i > 10 \). Then, we have \( (a_{e,f}^i - F) + (0.001a_{e,p}^i - 0.001a_{e,p}^i) = -F + 0.001a_{e,p}^i - 0.001a_{e,p}^i \). Consider a different \( s_j \) with \( a_{e,f}^j > 0 \) and \( a_{e,p}^j > 10 \). Obviously, it is possible to have \( (-F + 0.001a_{e,p}^j - 0.001a_{e,p}^j > 0) \) and \( (a_{e,f}^j - F) + (0.001a_{e,p}^j - 0.001a_{e,p}^j < 0) \). Therefore, it is possible that \( s_i \) with \( a_{e,f}^i = 0 \) has risk value higher than that of the \( s_j \) with \( a_{e,f}^j > 0 \), which is against the second intuition. Moreover, this example has shown that for \( Wong_3 \), it is also possible to have \( s_i \) with \( a_{e,f}^i = 0 \) as an element of \( S_B^{W_3} \). Similar to the above discussion for \( ER_4 \), we can prove that \( Wong_3 \rightarrow Wong_3' \) does not hold.

Now, an interesting question is: are the refined formulas of \( ER_4 \) and \( Wong_3 \) the maximal formulas? The answer is given by the following propositions.

**Proposition 4.4.2.** The refined formulas of \( ER_4 \) are not maximal formulas.

**Proof.** Let us denote \( ER_4' \) as the formulas after applying our refinement method on \( ER_4 \). It is obvious that formulas in \( ER_4' \) are still equivalent. Hence, it is sufficient to consider formula \( Wong_2' \) refined from \( Wong_2 \). As proved in Proposition 4.4.1, \( S_B^{W_2'} \) and \( S_A^{W_2'} \) for \( Wong_2' \) are as follows.

\[ S_B^{W_2'} = S_B^{W_2} \setminus \{ s_i | a_{e,f}^i = 0 \text{ and } R_{W_2}(s_i) > R_{W_2}(s_j), 1 \leq i \leq n \} \]
\[ S_A^{W_2'} = S_A^{W_2} \cup \{ s_i | a_{e,f}^i = 0 \text{ and } R_{W_2}(s_i) > R_{W_2}(s_j), 1 \leq i \leq n \} \]

We are going to prove that \( ER_2 \rightarrow Wong_2' \). Similarly, it is sufficient to prove \( Jaccard \rightarrow Wong_2' \).

First, we will prove \( S_B^{J} \supseteq S_B^{W_2'} \). As proved in Proposition 4.3.8, we have \( S_B^{J} \subseteq S_B^{W_2} \). Furthermore, since \( S_B^{J} = \{ s_i | a_{e,f}^i > 0 \text{ and } 1 + \frac{a_{e,f}^i}{a_{e,f}^i - F} - \frac{a_{e,f}^i}{a_{e,f}^i} > 0, 1 \leq i \leq n \} \) after (4.3.14), we have \( S_B^{J} \cap \{ s_i | a_{e,f}^i = 0 \text{ and } R_{W_2}(s_i) > R_{W_2}(s_j), 1 \leq i \leq n \} = \emptyset \). As a consequence, we have \( S_B^{J} \subseteq S_B^{W_2'} \).
Secondly, we will prove \( S_{A}^{W2'} \subseteq S_{A}^{J} \). As proved in Proposition 4.3.8, we have \( S_{A}^{W2} \subseteq S_{A}^{J} \). Besides, since \( S_{A}^{J} = \{ s_{i} | (a_{ef}^{i} = 0) \text{ or } (a_{ef}^{i} > 0 \text{ and } 1 + \frac{a_{ef}^{i}}{P} - \frac{P}{a_{ef}^{i}} < 0), 1 \leq i \leq n \} \) after (4.3.16), we have

\[
\{ s_{i} | a_{ef}^{i} = 0 \text{ and } R_{W2}(s_{i}) > R_{W2}(s_{f}), 1 \leq i \leq n \} \subseteq S_{A}^{J} \\
\{ s_{i} | a_{ef}^{i} = 0 \text{ and } R_{W2}(s_{i}) = R_{W2}(s_{f}), 1 \leq i \leq n \} \subseteq S_{A}^{J}
\]

Therefore, \( S_{A}^{W2'} \subseteq S_{A}^{J} \).

In conclusion, we have proved \( S_{B}^{W} \subseteq S_{B}^{W2'} \) and \( S_{A}^{W2'} \subseteq S_{A}^{J} \). Immediately after Theorem 4.2.2, Jaccard \( \rightarrow \) Wong2'. And after Proposition 4.3.2, ER2 \( \rightarrow \) ER4'. As discussed in Section 4.3.4, since formulas in ER2 are not maximal formulas, ER4' cannot be maximal.

**Proposition 4.4.3.** The refined formula of Wong3 is not a maximal formula.

**Proof.** Let us denote Wong3' as the formula after applying our refinement method on Wong3. As proved in Proposition 4.4.1, \( S_{B}^{W3'} \) and \( S_{A}^{W3'} \) for Wong3' are as follows.

\[
S_{B}^{W3'} = S_{B}^{W3} \setminus \{ s_{i} | a_{ef}^{i} = 0 \text{ and } R_{W3}(s_{i}) > R_{W3}(s_{f}), 1 \leq i \leq n \} \\
S_{A}^{W3'} = S_{A}^{W3} \cup \{ s_{i} | a_{ef}^{i} = 0 \text{ and } R_{W3}(s_{i}) > R_{W3}(s_{f}), 1 \leq i \leq n \} \cup \{ s_{i} | a_{ef}^{i} = 0 \text{ and } R_{W3}(s_{i}) = R_{W3}(s_{f}), 1 \leq i \leq n \}
\]

We are going to prove that ER1 \( \rightarrow \) Wong3'. Similarly, it is sufficient to prove Op1 \( \rightarrow \) Wong3'.

First, we will prove \( S_{B}^{Op1} \subseteq S_{B}^{W3'} \). As proved in Proposition 4.3.15, we have \( S_{B}^{Op1} \subseteq S_{B}^{W3} \). Furthermore, since \( S_{B}^{Op1} = \{ s_{i} | a_{ef}^{i} = F \text{ and } a_{ep}^{i} - a_{ep}^{i} > 0, 1 \leq i \leq n \} \) after (4.3.4), we have \( S_{B}^{Op1} \cap \{ s_{i} | a_{ef}^{i} = 0 \text{ and } R_{W3}(s_{i}) > R_{W3}(s_{f}), 1 \leq i \leq n \} = \emptyset \). As a consequence, we have \( S_{B}^{Op1} \subseteq S_{B}^{W3'} \).

Secondly, we will prove \( S_{A}^{W3'} \subseteq S_{A}^{J} \). As proved in Proposition 4.3.8, we have \( S_{A}^{W3} \subseteq S_{A}^{J} \). Besides, since \( F > 0 \) and \( S_{A}^{Op1} = \{ s_{i} | (a_{ef}^{i} < F) \text{ or } (a_{ef}^{i} = F \text{ and } a_{ep}^{i} - a_{ep}^{i} < 0), 1 \leq i \leq n \} \) after (4.3.6), we have

\[
\{ s_{i} | a_{ef}^{i} = 0 \text{ and } R_{W3}(s_{i}) > R_{W3}(s_{f}), 1 \leq i \leq n \} \subseteq S_{A}^{Op1} \\
\{ s_{i} | a_{ef}^{i} = 0 \text{ and } R_{W3}(s_{i}) = R_{W3}(s_{f}), 1 \leq i \leq n \} \subseteq S_{A}^{Op1}
\]

Therefore, \( S_{A}^{W3'} \subseteq S_{A}^{Op1} \).

In conclusion, we have proved \( S_{B}^{Op1} \subseteq S_{B}^{W3'} \) and \( S_{A}^{W3'} \subseteq S_{A}^{Op1} \). Immediately after Theorem 4.2.2, Op1 \( \rightarrow \) Wong3'. And after Proposition 4.3.1, ER1 \( \rightarrow \) Wong3'.

On the other hand, we will prove that Wong3' \( \rightarrow \) ER1 does not hold. Consider \( PG_{1} \) with \( TS_{1} \) in Table 4.2. Then, the \( S_{B}^{R}, S_{F}^{R} \) and \( S_{A}^{R} \) for ER1 are shown as the scenario A in Table 4.4; while
the corresponding sets for Wong3' are as the scenario B in Table 4.4. Then, using any consistent tie-
breaking scheme, the EXAM score of ER1 is less than the EXAM scores of Wong3'. As a consequence,
Wong3' → ER1 does not hold.

In summary, Wong3' is not a maximal formula.

Therefore, the above Propositions 4.4.2 and 4.4.3 mean that the refinements of the investigated
formulas do not affect our conclusions about the maximal formulas.

4.5 Discussion

As discussed in Section 4.3.1, our framework assumes that the test suite contains at least one failed
test case and one passed test case. Such an assumption is reasonable. First, it is widely accepted that
at least one failed test case is a necessary starting point for almost all types of debugging. Secondly,
in terms of the requirement of at least one passed test case, except ER5, all the other investigated
formulas actually have this latent assumption, although no one has explicitly stated this condition.
According to Table 4.1, ER5 uses the failed information, thus the number of passed test cases does
not affect its performance. However, for other formulas, without any passed test cases, they either
become undefined (such as Tarantula) or become ER5, which means that these formulas actually
conceive this assumption in their definition. Actually, as compared with the failed test cases that may
be difficult to get, the passed test cases normally exist. Even if the current test suite does not contain
any passed test case, after applying regression testing, some failed test cases could be changed into
passed ones. In other words, this assumption is realistic and can be easily achieved in practice.

Besides, we also have the assumption that the test suite has 100% statement coverage. It is true
that such assumption is not always held in reality. However, whether a test suite actually achieves
100% statement coverage does not really affect the applicability of our framework. Because SBFL
uses coverage and testing results as its basic information to search for the faulty statements. Given
a test suite that reveals some failures, only those covered statements could be the potential faulty
statements resulting in the revealed failures. Specifically, if a statement is covered by a failed test
case, it has possibility to be the root for this failure; while if a statement is covered by a passed test
case, though there is no definite conclusion that can be obtained, intuitively, such information still
increases debugger’s confidence about the correctness of this statement. However, if a statement is
never covered by any test case in the given test suite, it can never be the faulty statement that triggers
the observed failures. Therefore, in practice, if a test suite does not achieve 100% statement coverage,
we should exclude those uncovered statements and only focus on the covered portion of program, on
which 100% statement coverage is always held and thus our framework is applicable.

In addition to the above, we have assumed that given a ranking list, debuggers examine the
statements from the top to the bottom and once the faulty statement is examined, the fault can always be identified. Such an assumption is widely accepted by the community of SBFL. However, it is also undeniable that debugging is a complex activity that involves many human factors, such as the working pattern and ability of the debugger, which can affect the final effectiveness of the debugging. For example, Parmin and Orso [2011] have found that debuggers do not necessarily examine each statement one by one, following the given ranking list. They also found that “perfect debugging” does not generally hold in reality. However, since this study focuses on the performance of the SBFL technique itself, in particular, on the performance of different risk evaluation formulas, we still adopt this assumption, and accordingly our conclusions are with respect to the performance of the techniques, rather than the effectiveness of the entire debugging process.

Therefore, the only constraint of our study may be the single-fault assumption. However, even though we did not provide a theoretical analysis on whether the maximal formulas under single-fault scenario still perform best in multiple-fault cases, we believe that our conclusions are still meaningful. DiGiuseppe and Jones [2011] have recently studied the influence of multiple-fault on SBFL and found that the performance does not actually vary significantly with respect to the quantity of faults. Therefore, in general cases where the quantity of the faults are unknown, it is still intuitive to choose risk evaluation formulas with better performance under single-fault scenario. Recently, some promising techniques for multiple-fault have been developed to locate faults simultaneously, using the approach of “fault-focusing clustering” [Dickinson et al., 2001; Podgurski et al., 2003; Liu and Han, 2006; Zheng et al., 2006; Jones et al., 2007]. When adopting this parallel debugging pattern, test cases are first clustered based on various execution information into several specialized test suites, each of which targets an individual fault. In practice, each specialized test suite is dispatched to a particular debugger, who is supposed to focus on the corresponding single fault. In other words, a debugging task for multiple-fault is divided into several parallel sub-tasks for single fault, and each sub-task is handled by one debugger. As a consequence, for each debugger, his/her sub-task can be considered as single-fault scenario where the results of this thesis can be applied. Obviously, since most clustering techniques use heuristic methods, they cannot guarantee to eliminate all fault-localization interference (that is how different faults interfere with each others’ localizability) [DiGiuseppe and Jones, 2011]. Thus, how the noises brought by the “fault-focusing clustering” techniques affect our theoretical framework need to be studied in future.

4.6 Related works

As one of the most critical components in SBFL, the risk evaluation formula has been extensively investigated. Some studies have focused on designing effective risk evaluation formulas, such as Tarantula [Jones et al., 2002], Jaccard [Chen et al., 2002a], Ochiai [Abreu et al., 2006], three formulas proposed by Wong et al. [2007] (which are referred to as Wong1, Wong2 and Wong3, respectively, in
this thesis), etc. [Reps et al., 1997; Zeller, 2002; Liblit, 2004; Liblit et al., 2005; Liu et al., 2006; Wong et al., 2008, 2010]. Generally speaking, different formulas were developed from different intuitions or designed to serve for different purposes. For instance, Wong3 [Wong et al., 2007] was aimed at distinguishing the effects of different passed test cases in risk evaluation. It was based on the heuristic that the impact of the first passed test case in evaluating the risk of its executed statement should not be less significant than that of the second passed test case that executes the relevant statement.

With more and more formulas being proposed, some people started to compare the performance among various formulas. Abreu et al. [2006, 2007, 2009a] conducted empirical performance comparison among different risk evaluation formulas. In these studies, the risk evaluation process was interpreted into the computation of the similarity between different vectors in the program spectrum and a vector that contains information about the detected errors. Accordingly, risk evaluation formulas became the similarity coefficients in such computation. Abreu et al.’s experimental results consistently presented that Ochiai outperformed Jaccard (ER2), and Jaccard (ER2) outperformed Tarantula (ER3). They have tried to explain this phenomenon by analyzing how different elements in vector $A_i$ affected the returned values of these formulas. However, this analysis was not able to reveal the core reason behind such a performance hierarchy. They also discovered that the original format of AMPLEx performed the worst in most cases, which is understandable as mentioned in Section 4.3.2 that this original format is against the intuition of SBFL formulas. Actually, the observations of Abreu et al. was also validated by other empirical studies. For example, Santelices et al. [2009] have confirmed that their early experimental results showed that Ochiai outperformed Tarantula. Recently, Lee et al. [2009b] and Naish et al. [2011] conducted more comprehensive studies on different risk evaluation formulas, which showed consistency with our conclusions.

Though these empirical studies gave us hints and confidence that there should be some kind of performance hierarchy, they did not show a clear hierarchy. For example, Wong et al. [2007] have reported that by using the same tie-breaking scheme of either BEST or WORST, Wong3 outperformed Tarantula. Though these results are possible, they actually revealed only one side of the fact. In our analysis, we have found that not only may Wong3 outperform Tarantula, it may also perform worse than or equal to Tarantula.

Due to the limitation of empirical study, later, some researchers have attempted to compare the performance of different formulas from a theoretical perspective. Lee et al. [2009a] have proved that formula Tarantula is equivalent to formula $q_e$, in terms of having identical ranking list. In a follow-up study, Naish et al. [2011] conducted a more comprehensive investigation, where over 30 formulas were studied and more equivalence relations were identified, using the same definition of equivalence as Lee et al. [2009a]. However, such an equivalence relation is the most strict type of equivalence that should be relaxed to cater for more realistic scenarios.

Naish et al. [2011] also investigated the non-equivalence relations, using a hybrid approach, with
a model program and a group of multisets of execution paths. Their model program was proposed to simulate a single-fault program, with the following two features.

- Having noise (that is, non-faulty statements may have risk values not less than that of the faulty statement)
- Having attenuated signal (that is, the faulty statement is not always executed, or may not trigger failure when it is executed)

And the average performance over all possible multisets of execution paths was used to measure the performance of a formula. A multiset of execution paths was effectively the abstraction of the path coverage information and the testing results of a concrete test suite, with respect to the model program. In their study, for a risk evaluation formula, the performance score with respect to a multiset of execution paths was 0 if the risk of the faulty statement was less than any other statement. Otherwise, the score was \( \frac{1}{k} \), where \( k \) was the number of statements having equal risk values as the faulty statement. And the overall performance of this formula was measured by the total score, which was the sum (or average) of the scores over all possible distinct multisets of execution paths that contain at least one failed test case. Technically speaking, given the number of test cases \( t \), the total score of a formula can be determined by summing up the scores over all possible multisets of \( t \) execution paths. However, even for their simple model program, the number of all possible distinct multisets increases dramatically with the increase of \( t \). Thus, in their study, when the number of possible multisets was not too large, all the possible multisets were used to evaluate the performance; while for large numbers of multisets, a random sample of them was used, which was selected according to a uniform distribution of the combinations of path coverage and testing results. Hence, their analysis still involved sampling and simulation.

Based on their model program and performance metric, two optimal risk evaluation formulas were proposed and proved. However, their performance metric was not commonly adopted by the SBFL community, and in the comparison among all the formulas, they still adopted an empirical approach.

Apart from the performance comparison, Naish et al. [2011] also empirically investigated the impacts of various factors, including test suite size, error detection accuracy, the number of failed test cases and the execution frequency of the buggy code. And their experimental results were more comprehensive than the results in [Abreu et al., 2009a].

It should be noted that our theoretical analysis is focused on risk evaluation formulas that use the same information, namely \( A_i \). Recently, some extended SBFL techniques, which integrate the basic procedure with other models or employ additional information have emerged, such as the SBFL with causal inference using program dependence graphs by Baah et al. [2010], some weighted SBFL techniques using additional information from either passed test case or failed test case as weighting factors by Bandyopadhyay and Ghosh [2011], and by Naish et al. [2009] etc. However, no matter how
SBFL is extended, selecting a well-performed risk evaluation formula is always the most fundamental and essential task. Intuitively speaking, a formula with better performance in basic SBFL should be preferred in the extended versions. Such an intuition is conceived by some extended SBFL techniques. For example, both Baah et al. [2010] and Bandyopadhyay and Ghosh [2011] chose formula Ochiai in their studies, due to its empirically good performance. Naish et al. [2009] have applied their weighting factors in different formulas, but interestingly, the performance comparison among these formulas after applying the weighting factor is consistent with the one without using this factor. As a consequence, our framework that provides a definite solution to the most fundamental issue of SBFL, can help in both the basic SBFL and its extended versions.
5

Conclusion

5.1 Summary

SBFL has been extensively investigated, due to its simplicity and effectiveness. However, there are still some challenging problems. In this thesis, we have focused on two major challenges.

The first challenging problem is to apply SBFL in the application domains without test oracle. Currently, all the SBFL techniques have assumed the existence of a test oracle for the program under debugging. If the oracle is not available, the program spectrum cannot be associated with any testing results, and hence there is insufficient information for risk evaluation. However, in practice, many application domains do not have test oracles. Thus, such an assumption has severely restricted the scope of applicability of the existing SBFL techniques. Recently, metamorphic testing has been proposed to alleviate the oracle problem. Thus, it is natural to investigate how metamorphic testing could be used to alleviate the oracle problem in SBFL.

We propose a novel concept, metamorphic slice, based on the integration of metamorphic testing and program slicing. In our proposal, for any existing SBFL techniques, the role of an individual test case is replaced by that of a metamorphic test group, the role of slice is replaced by that of metamorphic slice, and the testing result of failure or pass associated with each individual slice is replaced by the testing result of violation or non-violation associated with each metamorphic slice. With these one-to-one replacements, we could then construct the metamorphic versions for the existing SBFL techniques.

An experimental study on 9 programs of varying program sizes has been conducted to investigate the performance of our proposed approach and see whether it is practically effective or not. From the results, we can conclude that our approach has delivered a performance level very similar to that achieved by the conventional SBFL techniques for the situation with test oracle. Since the performance of the conventional SBFL has been generally regarded as effective and satisfactory by the testing community, our approach can be regarded as practically satisfactory. A major significant
contribution of this thesis is: our study has successfully demonstrated that test oracle is no longer mandatory in SBFL.

It should be noted that applying metamorphic slice requires efforts on generation of MRs, which is not a trivial task. However, recently studies have provided empirical evidence that after a brief general training, testers can properly define MRs and effectively apply MT on the target programs [Hu et al., 2006; Zhang et al., 2009]. Also, we have already started some projects of automatically generating MRs, and have obtained very encouraging results.

The second problem studied in this thesis is to identify the most effective risk evaluation formulas. With the emerging of many risk evaluation formulas, it is important to know which formulas should be used when SBFL is applied. Most of the related studies have adopted an empirical approach, and hence the reported results are strongly dependent on the experimental set-up. Though researchers used various approaches to control the threats to validity in order to provide a more fair comparison of various formulas, the empirical investigations can hardly be considered as sufficiently comprehensive due to the huge number of possible combinations of various factors in SBFL. Recently, some researchers have sought for a theoretical comparison of different risk evaluation formulas, including the first study by Lee et al. [2009a], followed by a comprehensive study by Naish et al. [2011]. In their studies, the equivalence of some risk evaluation formulas was proved. However, their type of equivalence is the most strict type of equivalence, which requires two formulas to produce identical ranking list. It is well-known that the determinant for the performance comparison of various formulas is the ranking of the faulty statement. Though identical ranking lists can guarantee the same rankings for faulty statements regardless of the number of faulty statements, it may treat some intuitively equivalent formulas as non-equivalent because having identical ranking lists is a sufficient condition but not a necessary condition to have the same rankings for the faulty statements. Thus, their definition of equivalence does not properly reflect a more realistic scenario. Naish et al. [2011] also identified two optimal formulas with respect to their model and performance metric. However, their performance metric was not the most widely adopted metric by the SBFL community.

In this thesis, we provide a theoretical solution to the effectiveness comparison among various risk evaluation formulas. We propose a new type of equivalent relation and a better relation. Our definition of equivalence is more general and intuitively appealing than that of Naish et al. [2011]. In order to reveal the actual relations between different formulas, we develop an innovative theoretical framework, based on the intuition that the number of statements with risk values higher than that of the faulty statement, predominantly determines the ranking of the faulty statement. Our framework divides all program statements into three disjoint sets with risk values higher than, equal to and lower than that of the faulty statement, and compares the sizes of these sets for different formulas using the notion of subset. We apply our framework on the formulas investigated by Naish et al. [2011], but exclude some formulas with the justifications given in Section 4.3.2. Among the 30 investigated formulas, our result suggests that for single-fault scenario, there are five maximal formulas, namely,
Op1, Op2, Wong1, Russell & Rao and Binary, which are grouped into two equivalent groups, ER1 and ER5. In other words, when we apply SBFL with the single-fault assumption, we only need to consider risk evaluation formulas from the maximal formulas.

The only constraint of our study may be the single-fault assumption. Currently, some promising techniques towards multiple-fault are based on the approach of “fault-focusing clustering”, which first clusters test cases based on various execution information into several specialized test suites, each of which targets a single fault [Dickinson et al., 2001; Podgurski et al., 2003; Liu and Han, 2006; Zheng et al., 2006; Jones et al., 2007]. Then, each specialized test suite is considered to provide supportive information for the debugging of each individual fault. Therefore, dealing with the multiple-fault scenario can be transformed into dealing with several single-fault scenarios. Hence, the analysis in this thesis for single-fault scenario can serve as the basis of the analysis for multiple-fault scenario.

5.2 Future works

In our future works, we will continue the studies in the following directions.

1. Further studies on the application of execution metamorphic slice in SBFL

In this thesis, we have demonstrated using the novel concept of execution metamorphic slice to alleviate the oracle problem in the conventional SBFL techniques. In our further study on this topic, more extensive and comprehensive experimental analysis on larger-scaled programs, more types of faults, as well as the programs with multiple-faults, will be conducted. And the effectiveness of different MRs, and how to select better MR to provide a better fault localization performance, will also be studied.

Currently, our approach only involves substituting the conventional matrix MS and the vector RE in Figure 2.1, with their “metamorphic” counterparts. Reformulating this information into $A_i = \langle a_i^f, a_i^{\text{ep}}, a_i^{nf}, a_i^{np} \rangle$ for each $s_i$, as well as mapping the $A_i$ to a risk value, have remained unchanged. Actually, since the metamorphic slice and their testing results have more complex structures than the traditional slices and their testing results, they should be able to provide greater variety of information to design more effective reformulating and mapping methods.

One potential direction is based on the indefiniteness of the metamorphic testing results. Different from the traditional slice that associated with only two possible testing results, namely, failure or pass, though a metamorphic slice is also associated with two possible results, violation or non-violation, there is other information. Consider an MR that involves one source test case and one follow-up test case. For any metamorphic test group with a violated testing result, there are three possibilities: (1) the traditional slices of both the source and follow-up test cases contain faults; (2) only the traditional slice of source test case contains faults while the slice of follow-up test case
5.2. FUTURE WORKS

(3) only the traditional slice of follow-up test case contains faults while the slice of the source test case does not. Obviously, the scenario would be even more complicated when more inputs are involved in the MR. Thus, more advanced techniques would be required to analyze the indefinite information associated with the metamorphic testing result. Based on this information, different contributions to the violation made by different $s_i$ in the execution metamorphic slice can be identified. Accordingly, we can design new methods to reformulate the information of matrix $MS$ and the vector $RE$ in Figure 3.1, as well as design new mapping method to evaluate the risk values. Some potential heuristic criteria include “maxi-min criterion” (aiming at getting the best out of the worst), “Laplace criterion” (treating all scenarios as equally likely in the absence of information), “Savage criterion” (aiming to minimize the maximum amount of regret when a wrong decision is made), “Hurwicz criterion” (allowing the debuggers to define their degree of optimism), etc.

2. Other applications of metamorphic slice

This thesis only addresses one application of one type of metamorphic slice. Hence, another future research direction is to revisit the applications of traditional slice beyond SBFL, such as, testing, debugging, program analysis, etc., and to investigate how to apply different types of metamorphic slice as counterparts of the traditional slice in these areas. By using metamorphic slice instead, we can extend the original techniques into the programs without test oracle.

More importantly, the application of metamorphic slice is far beyond simply alleviating the oracle problem. We believe that with richer information conceived in the metamorphic slice, it should be more versatile than the traditional slice. For example, one potential application of metamorphic slice is the debugging for multiple faults. As mentioned in the previous section, the “fault-focusing clustering” technique is one of the promising techniques towards multiple-fault scenario. Since metamorphic slice is a property-based (MR) concept and normally one property of a program is linked with one program behaviour and can be mapped into certain part of the source code, the MR of the metamorphic slice can provide a new perspective for the clustering, which is more natural and straightforward.

3. Further theoretical analysis of SBFL

Based on our current theoretical performance comparison for the single-fault scenario, we will continue the analysis for multiple-fault scenario in our future study, by using the “fault-focusing clustering” technique. As mentioned above, metamorphic slice can be applied to provide a new clustering method.

Another future project is to theoretically analyze the impacts of different parameters, such as, $A_i$, $A_f$, $P$, $F$, etc., on the performance of an individual formula. Such information can be used to identify the most critical factor to locate a fault, and to identify the features of a test suite that favours a particular formula and leads to a better performance.
Moreover, our theoretical framework has used the notion of subset to compare the sizes of different sets. Actually, the subset relationship between two sets is only a sufficient condition but not a necessary condition for one set to have an equal or larger size than the other. Thus, we are interested in investigating whether there is a tighter relation to facilitate such comparison for different formulas.
References


REFERENCES


REFERENCES


Publications of the Author


