Author: Xie, X., Chen, T. Y., Kuo, F. C. & Xu, B.
Title: A theoretical analysis of the risk evaluation formulas for spectrum-based fault localization
Article number: 31
Year: 2013
Journal: ACM Transactions on Software Engineering and Methodology
Volume: 22
Issue: 4
URL: http://doi.org/10.1145/2522920.2522924

A

A Theoretical Analysis of the Risk Evaluation Formulas for Spectrum-Based Fault Localization

XIAOYUAN XIE, Swinburne University of Technology
TSONG YUEH CHEN, Swinburne University of Technology
FEI-CHING KUO, Swinburne University of Technology
BAOWEN XU, Nanjing University

An important research area of spectrum-based fault localization (SBFL) is the effectiveness of risk evaluation formulas. Most previous studies have adopted an empirical approach, which can hardly be considered as sufficiently comprehensive because of the huge number of combinations of various factors in SBFL. Though some studies aimed at overcoming the limitations of the empirical approach, none of them has provided a completely satisfactory solution. Therefore, we provide a theoretical investigation on the effectiveness of risk evaluation formulas. We define two types of relations between formulas, namely, equivalent and better. To identify the relations between 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 the risk value of the faulty statement. We group all program statements into three disjoint sets with risk values higher than, equal to and lower than the risk value 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 formulas to be used in SBFL.

Categories and Subject Descriptors: D.2.5 [**Software Engineering**]: Testing and Debugging

General Terms: Verification

Additional Key Words and Phrases: Debugging, risk evaluation formulas, spectrum-based fault localization, testing

1. INTRODUCTION

It is commonly recognized that testing and debugging are important but resource consuming 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. Due to a great amount of manual involvement, fault localization is a very resource consuming task in the whole software development life cycle. Therefore many researchers have proposed various automatic and effective techniques for fault localization, in order to decrease its cost, as well as to increase the software quality.

One promising approach towards fault localization is Spectrum-Based Fault Localization (referred to as SBFL in this paper). Generally speaking, this approach tries to locate the suspicious parts in program by utilizing various program spectra acquired

This work is partially supported by an ARC Discovery Project (DP120104773) and the National Natural Science Foundation of China (90818027, 61170071).
Author's addresses: Xiaoyuan Xie, Tsong Yueh Chen and Fei-Ching Kuo, Faculty of Information and Communication Technologies, Swinburne University of Technology; Baowen Xu, State Key Laboratory for Novel Software Technology & Department of Computer Science and Technology, Nanjing University.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

dynamically from software testing, and the associated testing result of *failed* or *passed*, for each test case. The program spectrum can be any granularity of program entities [Reps et al. 1997; Harrold et al. 1998]. For example, one of the most widely adopted spectra is the execution slice [Agrawal et al. 1995; Wong and Qi 2006]. After collecting all the necessary information, SBFL uses different formulas to evaluate the risk of having 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 [Abreu et al. 2007].

SBFL has received a lot of attention due to its simplicity and effectiveness. One popular research area is on the effectiveness of various risk evaluation formulas that are also known as the suspiciousness metrics. 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. 2009], in which various approaches have been applied to control the threats to validity (e.g. using the established experimental set-up and benchmark, Siemens Suite [SIR 2005]), in order to provide a fair evaluation and comparison. However, the performance results of risk evaluation formulas in SBFL are 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 number of possible combinations of various factors. In other words, the representative set-up choices are still not comprehensive enough to provide a fair evaluation of the investigated SBFL technique. In summary, despite of all the above attempts, the problems and limitations of the experimental study still remain.

Therefore, some researchers have conducted theoretical analyses 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 have been proved [Naish et al. 2011]. However, the type of equivalence used in both Lee et al. [2009a] and Naish et al. [2011] is the most strict type of equivalence, which requires identical ranking lists for equivalent formulas. In the study by Naish et al. [2011], a model program was used 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. They have proposed two optimal risk evaluation formulas, with respect to their model and performance measurement. However, their performance measurement is not the measurement commonly used by the SBFL community. Thus, it is worthwhile to investigate whether their proposed formulas still remain optimal with respect to the commonly used measurement.

The contributions of this paper can be summarized as follows, while their impact and significance would be discussed in Section 7.

1. 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 numbers of statements with risk values higher than the risk value of the faulty statements among different formulas using the notion of subset. Our framework has provided an innovative approach towards theoretical comparison of risk evaluation formulas.
2. Using this framework, we investigate 30 SBFL risk evaluation formulas, and are able to find five out of these 30 formulas as maximal formulas, under the single-fault scenario. Naish et al.'s optimal formulas are found to be two of these five maximal formulas.

The rest of this paper is organized as follows. Section 2 provides background information about SBFL. Section 3 introduces the intuition of our theoretical analysis and presents our innovative approach of grouping the statements into three mutually exclusive sets, based on which we develop all the definitions and theorems in our approach. Section 4 investigates 30 risk evaluation formulas, and identifies five maximal formulas. Section 5 provides a review of previous empirical studies, and compares these empirical results with our theoretical results. Section 6 discusses the assumptions used in this study, covering their justifications, impacts as well as limitations, and elaborates the validity of our results. And finally, Section 7 gives the conclusions for this paper.

2. BACKGROUND

2.1. Spectrum-based fault localization (SBFL)

SBFL is a dynamic approach and 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 *failed* or *passed*. 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 profile could be the binary coverage status, the number of times that the entity has been covered, and the program state before and after executing the program entity, etc. In practice, there are many kinds of combinations [Harrold et al. 1998; Harrold et al. 2000]. The most widely adopted combination involves statement and its binary coverage status in a test execution, which is effectively the execution slice [Agrawal et al. 1995; Wong and Qi 2006]. In this paper, we will also follow the common practice to use the execution slice.

Given 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 1 shows the essential information required by SBFL. Matrix ***MS*** represents the program spectrum and ***RE*** records the testing results of all test cases, in which p indicates *pass* and f indicates *fail*. The element in the i^{th} row and j^{th} column of matrix ***MS*** represents the coverage information of statement s_i, by test case t_j, with 1 indicating s_i is executed, and 0 otherwise.

For each statement s_i, these data can be represented as a vector of four elements, denoted as A_i=$<a^i_{ef}, a^i_{ep}, a^i_{nf}, a^i_{np}>$, where a^i_{ef} and a^i_{ep} represent the number of test cases in ***TS*** that execute statement s_i and return the testing result of *fail* or *pass*, respectively; a^i_{nf} and a^i_{np} denote the number of test cases that do not execute s_i, and return the testing result of *fail* 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.

In Figure 2, program ***PG*** has four statements $\{s_1, s_2, s_3, s_4\}$, and test suite ***TS*** has six test cases $\{t_1, t_2, t_3, t_4, t_5, t_6\}$. t_5 and t_6 give rise to *failed* runs and the remaining four test cases give rise to *passed* runs, as indicated in ***RE***. 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^1_{np}=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^4_{ef}=2 for s_4 represents that s_4 is executed by two test cases which can detect failure.

A risk evaluation formula R is applied on each statement s_i to assign a real value that indicates the risk of being faulty for s_i. For example, the risk evaluation formula

$$TS:(t_1 \quad t_2 \quad ... \quad t_m)$$

$$PG:\begin{pmatrix} s_1 \\ s_2 \\ \cdot \\ \cdot \\ \cdot \\ s_n \end{pmatrix} \quad MS:\begin{pmatrix} 1/0 & 1/0 & ... & 1/0 \\ 1/0 & 1/0 & ... & 1/0 \\ & & \cdot & \\ & & \cdot & \\ & & \cdot & \\ 1/0 & 1/0 & ... & 1/0 \end{pmatrix}$$

$$RE:(p/f \quad p/f \quad ... \quad p/f)$$

Fig. 1. Essential information for SBFL

$$TS:(t_1 \quad t_2 \quad t_3 \quad t_4 \quad t_5 \quad t_6) \quad A_i < a^i_{ef} \; a^i_{ep} \; a^i_{nf} \; a^i_{np} >$$

$$PG:\begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} \quad MS:\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \quad MA:\begin{pmatrix} 2 & 4 & 0 & 0 \\ 0 & 1 & 2 & 3 \\ 1 & 3 & 1 & 1 \\ 2 & 3 & 0 & 1 \end{pmatrix}$$

$$RE:(p \quad p \quad p \quad p \quad f \quad f)$$

Fig. 2. An example for SBFL

Tarantula is defined as follows [Jones et al. 2002].

$$R_T(s_i) = \frac{a^i_{ef}}{a^i_{ef} + a^i_{nf}} / (\frac{a^i_{ef}}{a^i_{ef} + a^i_{nf}} + \frac{a^i_{ep}}{a^i_{ep} + a^i_{np}})$$

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 assigning the risk values to all statements, the statements are sorted descendingly according to their risk values. Debugging starts from the top to the bottom of the list. 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 measurement or its equivalent, which is the percentage of the code that needs (or needs not) to be examined before the faulty statement is examined. Such a measurement 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 the study of 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 our analysis. Obviously, a lower *EXAM* score of formula R indicates a better performance.

2.2. Risk evaluation formulas

One of SBFL's popular research areas is the design of effective risk evaluation formulas, aiming at having the faulty statements as high in the risk list as possible. Many formulas have been proposed for SBFL, which included Tarantula [Jones et al. 2002], Jaccard [Chen et al. 2002], 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 paper), etc. [Reps et al. 1997; Zeller 2002; Liblit 2004; Liblit et al. 2005; Liu et al. 2006; Wong et al. 2008; Wong et al. 2010]. Generally speaking, different formulas were developed from different intuitions or designed to serve for different purposes. But no matter from what intuitions the formulas were derived, they should all comply with the expectation that statements associated with more failed and less passed testing results should have higher faulty risks.

With more and more formulas proposed, some people started to compare their performance [Jones and Harrold 2005; Abreu et al. 2006; 2007; Abreu et al. 2009]. 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 measurement or its equivalents, the standardized experimental set-up and the unified benchmarks. In addition, both the mutation analysis and real case studies were conducted. However, despite of all the above efforts, the problems and limitations of the experimental study are only alleviated rather than completely solved.

First, for 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 significantly 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. Therefore, the experimental results can only be considered as sampled observations, which are very likely to change when the set-up is varied. For example, in Abreu et al.'s comparison of four formulas, using *EXAM* score, the Ochiai formula showed improvements ranging from 2.4% to 10% on average over the Jaccard formula [Abreu et al. 2006]. Obviously, if we are given a scenario which is very different from their experimental set-up, there is no way to know whether the observed average range of improvements is still valid for the given scenario. In other words, we are not able to generalize our observed results from the experimental analysis.

Secondly, we observed that in many of the experimental results, the differences among some formulas are actually quite marginal or statistically insignificant. For instance, Abreu et al. [2006] found that the difference in *EXAM* scores between the Ochiai and Jaccard formula can be just 2.4%. Since some of the object programs used in the experiments are very small in scales, 2.4% sometimes refers to only one statement, which makes the result meaningless in this instance.

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 pilot study was followed by a more comprehensive investigation [Naish et al. 2011], where over 30 formulas were studied and more equivalence relations were identified, using the same definition of equivalence as Lee et al. [2009a]. 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. The multiset of execution paths was actually the abstraction of the path coverage information and the testing results of each 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 $1/k$, where k denoted the number of statements (including the faulty statement) having equal risk values as the faulty statement. The overall performance of a formula was measured by the total score, which was the sum (or average) of the scores over all possible distinct multisets of t 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 the 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.

They have proposed two optimal formulas which are equivalent with respect to their model program and the total score. They also compared the performance among other non-equivalent formulas using experimental analysis. Besides, empirical analysis was conducted on 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.

However, the studies of Lee et al. [2009a] and Naish et al. [2011] have their limitations. First, they both adopted the most strict type of equivalence that two equivalent formulas produce identical ranking lists. 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 rankings for the faulty statement. In other words, some formulas which are intuitively equivalent would be regarded as non-equivalent according to the type of equivalence used in the study 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 measurement, their optimal formulas may not be optimal with respect to the most popular performance measurement of *EXAM* score in SBFL community.

In summary, none of the previous studies has actually provided a definite answer to the question that which formula is optimal with respect to *EXAM* score, and hence should be used when SBFL is applied. This paper uses a theoretical analysis to provide the answer.

3. OUR FRAMEWORK

As mentioned above, 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 into a ranking list. Debugging is then conducted on statements from top to bottom of the ranking list. Therefore, the relative risk values rather than the absolute risk values of all statements, are the key factor determining the performance for a formula R.

In this study, we will use the *EXAM* score to measure the performance of a formula R. According to the definition of *EXAM* score, the higher the faulty statement s_f can be ranked according to R, the lower *EXAM* score R can be obtained; and a lower *EXAM* score indicates a better performance. Given a ranking list in descending order of the risk values evaluated by a formula R, we can divide the set of all statements (denoted as S) into three mutually exclusive subsets, S_B^R, S_F^R and S_A^R with respect to an arbitrary s_f, as follows.

Definition 3.1. 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 $A_i=<a_{ef}^i, a_{ep}^i, a_{nf}^i, a_{np}^i>$ 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:

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

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

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

In the practice of SBFL, a tie-breaking scheme is required to determine the order of the statements with identical risk values, and such a scheme is also required in our theoretical analysis. In the context of *EXAM* score, all $s_i \in S_B^R$ are ranked higher than s_f, and all $s_i \in S_A^R$ are ranked lower than s_f. Thus, the ordering of the statements within S_B^R or S_A^R does not affect the ranking of s_f. As a consequence, we need not

consider the application or impact of tie-breaking scheme on S_B^R or S_A^R and hence we are only interested in how a tie-breaking scheme distinguishes and ranks $s_i \in S_F^R$.

In previous studies, various tie-breaking schemes have been used, including *WORST*, *BEST*, *ORIGINAL ORDER*, etc. [Wong et al. 2008; Wong et al. 2010; Xie et al. 2011]. As a theoretical analysis, our framework cannot assume any arbitrary tie-breaking scheme, because some of them may be unreasonable or counter-intuitive. Actually, a tie-breaking scheme solves the ordering problem that a risk evaluation formula cannot solve. Thus, when comparing different formulas, it is reasonable to expect that a tie-breaking scheme returns consistent rankings for all formulas, which is independent of the risk evaluation formulas. Let us use an example to further illustrate what are meant by consistent rankings. Suppose that two risk evaluation formulas R_1 and R_2 return the same S_F^R. Then, an identical ordering for statements in $S_F^{R_1}$ and $S_F^{R_2}$ would be expected after the application of a tie-breaking scheme. Based on this intuition, an intuitive generalization is that a tie-breaking scheme should preserve the relative order of any pair of common statements in S_F^R returned by different formulas. We refer such schemes as consistent tie-breaking schemes, which are formally defined as follows.

Definition 3.2. 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 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 3.3 (*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 $E_1 \leq E_2$.

It should be noted that the relation "$\rightarrow$" is reflexive, that is, we have $R_1 \rightarrow R_1$. Besides, 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 3.4 (*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 $E_1 = E_2$.

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 $R_1 \leftrightarrow R_3$.

In the context of the *EXAM* score which is the most widely accepted measurement to compare different risk evaluation formulas, 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 3.4, two formulas are equivalent if and only if they have the same number of statements preceding the faulty statement in the ranking lists, that is, they produce the same *EXAM* score. With respect to the definition of equivalence used by Lee et al. [2009a] and Naish et al. [2011], two formulas are equivalent if they produce identical ranking lists for all the statements. As a consequence, their equivalent formulas always produce the same *EXAM* score, and hence their equivalent formulas are also equivalent with respect to our type of equivalence. In summary, if two formulas are equivalent according to their definition, these two formulas are also equivalent according to our equivalence in Definition 3.4; but not vice versa. Thus, their definition is a special case of ours.

Immediately from Definition 3.3 and Definition 3.4, we have the following property.

THEOREM 3.5. *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 $R_2 \rightarrow R_1$.*

Intuitively speaking, the most straightforward approach to compare the ***EXAM*** scores of different formulas is to compare the sizes of their S_B^R and the numbers of statements that are from S_F^R but ranked before s_f by the tie-breaking scheme. For instance, in the above example that two risk evaluation formulas R_1 and R_2 return the same S_F^R, suppose that the size of $S_B^{R_1}$ of R_1 is smaller than the size of $S_B^{R_2}$ of R_2. Then, R_1 would have a lower ***EXAM*** score. However, since the sizes of S_B^R and S_F^R depend on the program and test suite, which can be very varying, a size comparison appears to be mathematically intractable. One of the innovative contributions in our study is to make use of the subset relationships among S_B^R (or S_F^R) of different formulas, to facilitate the analysis. It turns out that the use of the notion of subset is sufficient to identify the maximal risk evaluation formulas. In fact, we have the following sufficient condition for $R_1 \rightarrow R_2$ involving the notion of subset, which plays an important role in identifying the maximal risk evaluation formulas.

THEOREM 3.6. *Given any two risk evaluation formulas R_1 and R_2, if for any program, faulty statement s_f and test suite, we have $S_B^{R_1} \subseteq S_B^{R_2}$ and $S_A^{R_2} \subseteq S_A^{R_1}$, then $R_1 \rightarrow R_2$.*

PROOF. Consider a formula R_3, such that for any program, s_f and test suite, $S_B^{R_3}=S_B^{R_1}$ and $S_A^{R_3}=S_A^{R_2}$. 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. For R_1 and R_3, we have $S_B^{R_3}=S_B^{R_1}$, $S_F^{R_1} \subseteq S_F^{R_3}$ and $S_A^{R_3} \subseteq S_A^{R_1}$. 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 $S_B^{R_3} \subseteq S_B^{R_2}$, $S_F^{R_2} \subseteq S_F^{R_3}$ and $S_A^{R_3}=S_A^{R_2}$. If the tie-breaking scheme is consistent, s_f always has the same relative order with any element of $S_F^{R_2}$, in both L_2 and L_3. However, all elements in $S_F^{R_3} \backslash S_F^{R_2}$ will definitely be ranked higher than s_f in L_2, but not necessarily be ranked higher than s_f in L_3. As a consequence, $E_3 \leq E_2$.

Therefore, we have $E_1 \leq E_2$. Following immediately from Definition 3.3, we have $R_1 \rightarrow R_2$. □

With Theorems 3.5 and 3.6, we can now establish a sufficient condition for $R_1 \leftrightarrow R_2$.

THEOREM 3.7. *Given any two risk evaluation formulas R_1 and R_2, if for any program, faulty statement s_f and test suite, we have $S_B^{R_1}=S_B^{R_2}$, $S_F^{R_1}=S_F^{R_2}$ and $S_A^{R_1}=S_A^{R_2}$, then $R_1 \leftrightarrow R_2$.*

PROOF. Suppose that for any program, s_f and test suite, we have $S_B^{R_1}=S_B^{R_2}$ and $S_A^{R_1}=S_A^{R_2}$. In other words, we have $S_B^{R_1} \subseteq S_B^{R_2}$ and $S_A^{R_2} \subseteq S_A^{R_1}$, as well as $S_B^{R_2} \subseteq S_B^{R_1}$ and $S_A^{R_1} \subseteq S_A^{R_2}$. It follows immediately from Theorem 3.6 that $R_1 \rightarrow R_2$ and $R_2 \rightarrow R_1$. Therefore, we have $R_1 \leftrightarrow R_2$ after Theorem 3.5. □

4. EFFECTIVENESS OF RISK EVALUATION FORMULAS

In this section, we are going to compare the effectiveness of risk evaluation formulas, using the framework developed in previous section.

4.1. Investigated formulas

In this study, we investigate 30 risk evaluation formulas, which are selected from Naish et al. [2011], because their theoretical investigation is the most comprehensive one. These formulas are listed in Table I. 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 $\frac{a_{ef}+a_{np}}{a_{nf}+a_{ep}}$. In M1, all statements with $a^i_{nf}+a^i_{ep}=0$ would then have undefined risk values. However, there is no intuition to justify why we need to have the constraint of $a^i_{nf}+a^i_{ep}\neq 0$.

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 I is $|\frac{a_{ef}}{a_{ef}+a_{nf}}-\frac{a_{ep}}{a_{ep}+a_{np}}|$, which was originally proposed to identify faulty classes in object-oriented software, with the assumption that there is exactly one failing run [Dallmeier et al. 2005]. Since this original form always returns an absolute value, the magnitude order of the computed signed values may be changed. Therefore, the original form does not comply with 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 to SBFL, we follow Naish et al. [2011] to use its variant defined in Table I.

Table I: Investigated formulas

Name		Formula expression
ER1	Naish1 [Naish et al. 2011]	$\begin{cases} -1 & \text{if } a_{ef}<F \\ P-a_{ep} & \text{if } a_{ef}=F \end{cases}$
	Naish2 [Naish et al. 2011]	$a_{ef}-\frac{a_{ep}}{a_{ep}+a_{np}+1}$
ER2	Jaccard [Chen et al. 2002]	$\frac{a_{ef}}{a_{ef}+a_{nf}+a_{ep}}$
	Anderberg [Naish et al. 2011]	$\frac{a_{ef}}{a_{ef}+2(a_{nf}+a_{ep})}$
	Sørensen-Dice [Naish et al. 2011]	$\frac{2a_{ef}}{2a_{ef}+a_{nf}+a_{ep}}$
	Dice [Naish et al. 2011]	$\frac{2a_{ef}}{a_{ef}+a_{nf}+a_{ep}}$
	Goodman [Naish et al. 2011]	$\frac{2a_{ef}-a_{nf}-a_{ep}}{2a_{ef}+a_{nf}+a_{ep}}$

Table I: Investigated formulas (*cont.*)

Name		Formula expression
ER3	Tarantula [Jones et al. 2002]	$\frac{a_{ef}}{a_{ef}+a_{nf}} / (\frac{a_{ef}}{a_{ef}+a_{nf}} + \frac{a_{ep}}{a_{ep}+a_{np}})$
	q_e [Lee et al. 2009a]	$\frac{a_{ef}}{a_{ef}+a_{ep}}$
	CBI Inc. [Liblit et al. 2005]	$\frac{a_{ef}}{a_{ef}+a_{ep}} - \frac{a_{ef}+a_{nf}}{a_{ef}+a_{nf}+a_{ep}+a_{np}}$
ER4	Wong2 [Wong et al. 2007]	$a_{ef} - a_{ep}$
	Hamann [Naish et al. 2011]	$\frac{a_{ef}+a_{np}-a_{nf}-a_{ep}}{a_{ef}+a_{nf}+a_{ep}+a_{np}}$
	Simple Matching [Naish et al. 2011]	$\frac{a_{ef}+a_{np}}{a_{ef}+a_{nf}+a_{ep}+a_{np}}$
	Sokal [Naish et al. 2011]	$\frac{2(a_{ef}+a_{np})}{2(a_{ef}+a_{np})+a_{nf}+a_{ep}}$
	Rogers&Tanimoto [Naish et al. 2011]	$\frac{a_{ef}+a_{np}}{a_{ef}+a_{np}+2(a_{nf}+a_{ep})}$
	Hamming etc. [Naish et al. 2011]	$a_{ef} + a_{np}$
	Euclid [Naish et al. 2011]	$\sqrt{a_{ef} + a_{np}}$

Table I: Investigated formulas (*cont.*)

Name		Formula expression
ER5	Wong1 [Wong et al. 2007]	a_{ef}
	Russel & Rao [Naish et al. 2011]	$\frac{a_{ef}}{a_{ef}+a_{nf}+a_{ep}+a_{np}}$
	Binary [Naish et al. 2011]	$\begin{cases} 0 \text{ if } a_{ef}<F \\ 1 \text{ if } a_{ef}=F \end{cases}$
ER6	Scott [Naish et al. 2011]	$\frac{4a_{ef}a_{np}-4a_{nf}a_{ep}-(a_{nf}-a_{ep})^2}{(2a_{ef}+a_{nf}+a_{ep})(2a_{np}+a_{nf}+a_{ep})}$
	Rogot1 [Naish et al. 2011]	$\frac{1}{2}(\frac{a_{ef}}{2a_{ef}+a_{nf}+a_{ep}}+\frac{a_{np}}{2a_{np}+a_{nf}+a_{ep}})$
Kulczynski2 [Naish et al. 2011]		$\frac{1}{2}(\frac{a_{ef}}{a_{ef}+a_{nf}}+\frac{a_{ef}}{a_{ef}+a_{ep}})$
Ochiai [Abreu et al. 2006]		$\frac{a_{ef}}{\sqrt{(a_{ef}+a_{nf})(a_{ef}+a_{ep})}}$
M2 [Naish et al. 2011]		$\frac{a_{ef}}{a_{ef}+a_{np}+2(a_{nf}+a_{ep})}$
AMPLE2 [Naish et al. 2011]		$\frac{a_{ef}}{a_{ef}+a_{nf}}-\frac{a_{ep}}{a_{ep}+a_{np}}$
Wong3 [Wong et al. 2007]		$a_{ef}-h$, where $h=\begin{cases} a_{ep} & \text{if } a_{ep}\leq 2 \\ 2+0.1(a_{ep}-2) & \text{if } 2<a_{ep}\leq 10 \\ 2.8+0.001(a_{ep}-10) & \text{if } a_{ep}>10 \end{cases}$
Arithmetic Mean [Naish et al. 2011]		$\frac{2a_{ef}a_{np}-2a_{nf}a_{ep}}{(a_{ef}+a_{ep})(a_{np}+a_{nf})+(a_{ef}+a_{nf})(a_{ep}+a_{np})}$

Table I: Investigated formulas (*cont.*)

Name	Formula expression
Cohen [Naish et al. 2011]	$\frac{2a_{ef}a_{np}-2a_{nf}a_{ep}}{(a_{ef}+a_{ep})(a_{np}+a_{ep})+(a_{ef}+a_{nf})(a_{nf}+a_{np})}$
Fleiss [Naish et al. 2011]	$\frac{4a_{ef}a_{np}-4a_{nf}a_{ep}-(a_{nf}-a_{ep})^2}{(2a_{ef}+a_{nf}+a_{ep})+(2a_{np}+a_{nf}+a_{ep})}$

4.2. Assumptions

Before presenting our performance analysis of the selected formulas, we first list our assumptions. In Section 6, we will provide a detailed discussion of these 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. 2011].
2. We assume that debuggers examine the statements one by one from the top to the bottom of the ranking list returned by SBFL, and once the faulty statement is examined, the fault can always be identified. This is also known as "*perfect bug detection*", which is adopted by most of the previous SBFL studies [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 *fail* or *pass*. 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.
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 $a_{ef}^i + a_{nf}^i > 0$.

4.3. Maximal risk evaluation formulas

In this section, five out of the 30 investigated formulas would be identified as the most efficient formulas (known as the maximal formulas), under the single-fault scenario. This section consists of three subsections. Section 4.3.1 will first define some notations and give some lemmas, which would be used to identify the equivalent groups of formulas and the relations between non-equivalent formulas in Sections 4.3.2 and 4.3.3, respectively.

4.3.1. Preliminary. Generally speaking, in a partially ordered set $(S, >)$, an element a is said to be maximal if for any element $b \in S$, whenever $b > a$, b is a. In our context, 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, $R_2 \rightarrow R_1$ implies $R_2 \leftrightarrow R_1$. We use "$R_2 \leftrightarrow R_1$" instead of "R_2 is R_1" because *EXAM* score is our sole measurement to distinguish the performance between different formulas and different formulas may have the same *EXAM* score.

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 immediate after the definitions and the above assumptions.

LEMMA 4.1. *For any* $A_i = <a^i_{ef}, a^i_{ep}, a^i_{nf}, a^i_{np}>$, *we have* $a^i_{ef} + a^i_{ep} > 0$, $a^i_{ef} + a^i_{nf} = F$, $a^i_{ep} + a^i_{np} = P$, $a^i_{ef} \leq F$ *and* $a^i_{ep} \leq P$.

LEMMA 4.2. *For any faulty statement* s_f *with* $A_f = <a^f_{ef}, a^f_{ep}, a^f_{nf}, a^f_{np}>$, *if* s_f *is the only faulty statement in the program, we have* $a^f_{ef} = F$ *and* $a^f_{nf} = 0$.

4.3.2. Equivalent groups of formulas. We have identified six equivalent groups among all the 30 investigated formulas, as follows.

PROPOSITION 4.3. *Amongst all the investigated formulas as stated in Table I, there are six groups of equivalent formulas, which are defined and referred to as "ER1" to "ER6", as follows.*

— *ER1 consists of Naish1 and Naish2.*

— *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.*

PROOF. These six equivalent groups of formulas are identical to the six groups identified by Naish et al. [2011] with respect to their definition of equivalence. As explained after the presentation of Definition 3.4, the equivalence of Naish et al. [2011] is a special case of ours. Therefore, it follows immediately that ER1 to ER6 are equivalent groups. □

Intuitively speaking, $R_1 \leftrightarrow R_2$ does not necessarily imply that R_1 and R_2 are equivalent with respect to Naish et al.'s type of equivalence. However, amongst the 30 investigated formulas, there does not exist any pair of distinct R_1 and R_2 such that $R_1 \leftrightarrow R_2$ and they are not equivalent with respect to Naish et al.'s type of equivalence. Therefore, we provide the following example to demonstrate that our equivalence is more general than that of Naish et al. [2011].

Example 4.4. Consider a risk evaluation formula $\mathrm{r_e}$ defined as follows.

$$\mathrm{r_e} = \begin{cases} -\frac{a_{ep}}{a_{ef}} & \text{if } a_{ef} > 0 \\ -\frac{P}{F} - 1 & \text{if } a_{ef} = 0 \end{cases}$$

Though $\mathrm{r_e}$ is artificially constructed, it is an intuitively appealing risk evaluation formula because of the following reasons.

— For s_i with $a^i_{ef} > 0$, $\mathrm{r_e}$ complies with 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 $a^i_{ef} = 0$, $\mathrm{r_e}$ assigns risk value of $(-\frac{P}{F} - 1)$ to them. Since $F \geq 1$ and $a^f_{ep} \leq P$ after Lemma 4.1, we have $-\frac{a^f_{ep}}{F} \geq -\frac{P}{F} > (-\frac{P}{F} - 1)$. Therefore, the risk value of s_i with

a^i_{ef}=0, which is $(-\frac{P}{F}-1)$, is always lower than the risk value of s_f, which is $-\frac{a^f_{ep}}{F}$. This also complies with the general expectation that under the single-fault scenario, a statement s_i with a^i_{ef}=0 can never be the faulty statement [Xie et al. 2010].

We are going to show that our equivalence is more general than Naish et al.'s equivalence [Naish et al. 2011], through the proof that $r_e \leftrightarrow$ Tarantula and that r_e and Tarantula are not equivalent with respect to Naish et al.'s equivalence.

(A) To prove that $r_e \leftrightarrow$ Tarantula.
Our approach of proof is to show that S^R_B, S^B_F and S^R_A for both Tarantula and r_e are equal to the following sets X^R, Y^R and Z^R, respectively.

$$X^R=\{s_i|a^i_{ef}>0 \text{ and } \frac{a^f_{ep}}{F}-\frac{a^i_{ep}}{a^i_{ef}}>0, 1\leq i\leq n\} \tag{4.1}$$

$$Y^R=\{s_i|a^i_{ef}>0 \text{ and } \frac{a^f_{ep}}{F}-\frac{a^i_{ep}}{a^i_{ef}}=0, 1\leq i\leq n\} \tag{4.2}$$

$$Z^R=\{s_i|(a^i_{ef}=0) \text{ or } (a^i_{ef}>0 \text{ and } \frac{a^f_{ep}}{F}-\frac{a^i_{ep}}{a^i_{ef}}<0), 1\leq i\leq n\}=S\backslash(X^R\cup Y^R) \tag{4.3}$$

First, after Lemma 4.2 and Definition 3.1, for r_e, we have

$$S^{RE}_B=\{s_i|(a^i_{ef}>0 \text{ and } -\frac{a^i_{ep}}{a^i_{ef}}>-\frac{a^f_{ep}}{F}) \text{ or } (a^i_{ef}=0 \text{ and } -\frac{P}{F}-1>-\frac{a^f_{ep}}{F}), 1\leq i\leq n\}$$

S^{RE}_B can be re-written as

$$S^{RE}_B=\{s_i|a^i_{ef}>0 \text{ and } -\frac{a^i_{ep}}{a^i_{ef}}>-\frac{a^f_{ep}}{F}, 1\leq i\leq n\}\cup\{s_i|a^i_{ef}=0 \text{ and } -\frac{P}{F}-1>-\frac{a^f_{ep}}{F}, 1\leq i\leq n\}$$

Assume $(-\frac{P}{F}-1)>-\frac{a^f_{ep}}{F}$. Then, we have $\frac{a^f_{ep}}{F}>(\frac{P}{F}+1)$. Therefore, $a^f_{ep}>P$, which is a contradiction to Lemma 4.1. Thus, $\{s_i|a^i_{ef}=0 \text{ and } -\frac{P}{F}-1>-\frac{a^f_{ep}}{F}, 1\leq i\leq n\}=\emptyset$. Then, $S^{RE}_B=\{s_i|a^i_{ef}>0 \text{ and } -\frac{a^i_{ep}}{a^i_{ef}}>-\frac{a^f_{ep}}{F}, 1\leq i\leq n\}$, which can be re-written as $\{s_i|a^i_{ef}>0$ and $\frac{a^f_{ep}}{F}-\frac{a^i_{ep}}{a^i_{ef}}>0, 1\leq i\leq n\}=X^R$ in (4.1).
Next, after Lemma 4.2 and Definition 3.1, we have

$$S^{RE}_F=\{s_i|(a^i_{ef}>0 \text{ and } -\frac{a^i_{ep}}{a^i_{ef}}=-\frac{a^f_{ep}}{F}) \text{ or } (a^i_{ef}=0 \text{ and } -\frac{P}{F}-1=-\frac{a^f_{ep}}{F}), 1\leq i\leq n\}$$

Similarly, we can prove that S^{RE}_F is equal to Y^R defined in (4.2). Then, it follows immediately that $S^{RE}_A=S\backslash(X^R\cup Y^R)=Z^R$.

Now let us consider Tarantula. As stated in Table I, formula Tarantula is defined as follows.

$$R_T(s_i)=\frac{a^i_{ef}}{a^i_{ef}+a^i_{nf}}/(\frac{a^i_{ef}}{a^i_{ef}+a^i_{nf}}+\frac{a^i_{ep}}{a^i_{ep}+a^i_{np}})$$

It follows from Lemma 4.1 and 4.2 that $R_T(s_i)=\frac{a^i_{ef}}{F}/(\frac{a^i_{ef}}{F}+\frac{a^i_{ep}}{P})$ and $R_T(s_f)=1/(1+\frac{a^f_{ep}}{P})$. Then, after Definition 3.1, we have

$$S^T_B = \{s_i|\frac{a^i_{ef}}{F}/(\frac{a^i_{ef}}{F}+\frac{a^i_{ep}}{P})>1/(1+\frac{a^f_{ep}}{P}), 1\leq i\leq n\}$$

Now, we are going to prove $S^T_B=X^R$. For any s_i, we have either ($a^i_{ef}=0$) or ($a^i_{ef}>0$). Therefore, S^T_B can be re-written as

$$\begin{aligned} S^T_B=\{s_i|a^i_{ef}=0 \text{ and } \frac{a^i_{ef}}{F}/(\frac{a^i_{ef}}{F}+\frac{a^i_{ep}}{P})>1/(1+\frac{a^f_{ep}}{P}), 1\leq i\leq n\} \\ \cup\{s_i|a^i_{ef}>0 \text{ and } \frac{a^i_{ef}}{F}/(\frac{a^i_{ef}}{F}+\frac{a^i_{ep}}{P})>1/(1+\frac{a^f_{ep}}{P}), 1\leq i\leq n\} \end{aligned}$$

Consider the case that ($a^i_{ef}=0$). Since $(1+\frac{a^f_{ep}}{P})>0$ after Lemma 4.1, we have $\frac{a^i_{ef}}{F}/(\frac{a^i_{ef}}{F}+\frac{a^i_{ep}}{P})=0<1/(1+\frac{a^f_{ep}}{P})$, which is contradictory to $\frac{a^i_{ef}}{F}/(\frac{a^i_{ef}}{F}+\frac{a^i_{ep}}{P})>1/(1+\frac{a^f_{ep}}{P})$. Thus, $\{s_i|a^i_{ef}=0 \text{ and } \frac{a^i_{ef}}{F}/(\frac{a^i_{ef}}{F}+\frac{a^i_{ep}}{P})>1/(1+\frac{a^f_{ep}}{P}), 1\leq i\leq n\}=\emptyset$, and hence we have

$$S^T_B=\{s_i|a^i_{ef}>0 \text{ and } \frac{a^i_{ef}}{F}/(\frac{a^i_{ef}}{F}+\frac{a^i_{ep}}{P})>1/(1+\frac{a^f_{ep}}{P}), 1\leq i\leq n\} \tag{4.4}$$

— Assume that $s_i\in S^T_B$. After (4.4), we have ($a^i_{ef}>0$ and $\frac{a^i_{ef}}{F}/(\frac{a^i_{ef}}{F}+\frac{a^i_{ep}}{P})>1/(1+\frac{a^f_{ep}}{P})$). Since $a^i_{ef}>0$, we have $\frac{F}{a^i_{ef}}>0$ because $F>0$. Then, $\frac{a^i_{ef}}{F}/(\frac{a^i_{ef}}{F}+\frac{a^i_{ep}}{P})>1/(1+\frac{a^f_{ep}}{P})$ implies $1/(1+\frac{a^i_{ep}}{P}\frac{F}{a^i_{ef}})>1/(1+\frac{a^f_{ep}}{P})$. Furthermore, it follows from $\frac{F}{a^i_{ef}}>0$ and Lemma 4.1 that $(1+\frac{a^i_{ep}}{P}\frac{F}{a^i_{ef}})>0$ and $(1+\frac{a^f_{ep}}{P})>0$, then we have $\frac{a^i_{ep}}{P}\frac{F}{a^i_{ef}}<\frac{a^f_{ep}}{P}$. Since $\frac{P}{F}>0$, after multiplying each side by $\frac{P}{F}$ and re-arranging the terms, we have $\frac{a^f_{ep}}{F}-\frac{a^i_{ep}}{a^i_{ef}}>0$. Then, we have $s_i\in X^R$ after (4.1). Therefore, $S^T_B\subseteq X^R$.

— Assume that $s_i\in X^R$. After (4.1), we have ($a^i_{ef}>0$ and $\frac{a^f_{ep}}{F}-\frac{a^i_{ep}}{a^i_{ef}}>0$). Since $\frac{F}{P}>0$, after re-arranging the terms and multiplying each side by $\frac{F}{P}$, $\frac{a^f_{ep}}{F}-\frac{a^i_{ep}}{a^i_{ef}}>0$ becomes $\frac{a^i_{ep}}{P}\frac{F}{a^i_{ef}}<\frac{a^f_{ep}}{P}$ which implies $\frac{F}{a^i_{ef}}(\frac{a^i_{ef}}{F}+\frac{a^i_{ep}}{P})<(1+\frac{a^f_{ep}}{P})$. Therefore, we have $\frac{a^i_{ef}}{F}/(\frac{a^i_{ef}}{F}+\frac{a^i_{ep}}{P})>1/(1+\frac{a^f_{ep}}{P})$ because $a^i_{ef}>0$, $F>0$, $\frac{a^i_{ep}}{P}>0$ and $\frac{a^f_{ep}}{P}>0$. Then, we have $s_i\in S^T_B$ after (4.4). Therefore, $X^R\subseteq S^T_B$.

In summary, we have proved $S^T_B=X^R$.

Similarly, we have $S^T_F=\{s_i|\frac{a^i_{ef}}{F}/(\frac{a^i_{ef}}{F}+\frac{a^i_{ep}}{P})=1/(1+\frac{a^f_{ep}}{P}), 1\leq i\leq n\}$, which can be proved to be equal to Y^R. Then, it follows immediately that $S^T_A=S\backslash(X^R\cup Y^R)=Z^R$.

In conclusion, S^R_B, S^B_F and S^R_A for both Tarantula and r_e are equal to the sets in (4.1), (4.2) and (4.3), respectively. It follows after Theorem 3.7 that $r_e \leftrightarrow$ Tarantula.

(B) To prove that r_e and Tarantula are not equivalent with respect to Naish et al.'s equivalence.

For Tarantula, s_i with $a^i_{ef}=0$ will be assigned with the lowest risk value of 0, and hence is ranked lower than any s_j with $a^j_{ef}>0$. Consider a test suite with $P=6$ and $F=2$. For

r_e, if there exists s_i such that a^i_{ef}=0, we have $R_{RE}(s_i)$=$-\frac{P}{F}-1$=-4. Suppose there exists s_j such that a^j_{ef}=1 and a^j_{ep}=5. Then, we have $R_{RE}(s_j)$=$-\frac{a^j_{ep}}{a^j_{ef}}$=$-5$<$-4$=$R_{RE}(s_i)$. While for Tarantula, we have $R_T(s_j)$=$\frac{3}{8}$>0=$R_T(s_i)$. Therefore, r_e does not produce the same ranking list as Tarantula, and hence r_e and Tarantula are not equivalent with respect to Naish et al.'s equivalence.

4.3.3. Relations between non-equivalent formulas. 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, Ochiai, M2, AMPLE2, Wong3, Arithmetic Mean, Cohen and Fleiss. Among them, some constitute certain performance hierarchy chains, which are presented in Figure 3.

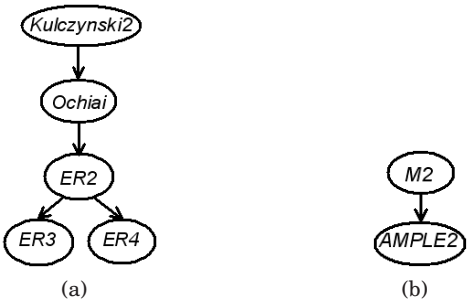

Fig. 3. Performance hierarchy chains of risk evaluation formulas

In Figure 3, each node represents a formula, or an equivalent group of formulas. The arrow from node N_1 to node N_2 means that for any formulas R_1 and R_2 in N_1 and N_2, respectively, $R_1 \rightarrow R_2$. Obviously, since the relation "$\rightarrow$" is transitive, for any formulas R_i and R_j in N_i and N_j, respectively, we have $R_i \rightarrow R_j$, as long as N_j is a direct or indirect descendant node of N_j in the corresponding chain.

First, let us consider Figure 3(a), which involves equivalent groups ER3, ER4 and ER2, as well as formulas Ochiai and Kulczynski2. Proposition 4.6, Proposition 4.7, Proposition 4.10 and Proposition 4.11 establish this figure. Before presenting the proof for ER2 $\rightarrow$ ER3, we need the following lemma for Jaccard of ER2.

LEMMA 4.5. *For Jaccard, we have S^J_B=X^J and S^J_A=Z^J, where*

$$X^J=\{s_i|a^i_{ef}>0 \textit{ and } 1+\frac{a^f_{ep}}{F}-\frac{F}{a^i_{ef}}-\frac{a^i_{ep}}{a^i_{ef}}>0, 1\leq i\leq n\} \tag{4.5}$$

$$Z^J=\{s_i|(a^i_{ef}=0) \textit{ or } (a^i_{ef}>0 \textit{ and } 1+\frac{a^f_{ep}}{F}-\frac{F}{a^i_{ef}}-\frac{a^i_{ep}}{a^i_{ef}}<0), 1\leq i\leq n\} \tag{4.6}$$

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

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

It follows from Lemma 4.1 and Lemma 4.2 that $R_J(s_i)=\frac{a^i_{ef}}{F+a^i_{ep}}$ and $R_J(s_f)=\frac{F}{F+a^f_{ep}}$. Then, after Definition 3.1, we have

$$S^J_B=\{s_i|\frac{a^i_{ef}}{F+a^i_{ep}}>\frac{F}{F+a^f_{ep}}, 1\leq i\leq n\} \tag{4.7}$$

$$S^J_A=\{s_i|\frac{a^i_{ef}}{F+a^i_{ep}}<\frac{F}{F+a^f_{ep}}, 1\leq i\leq n\} \tag{4.8}$$

(A) To prove that $S^J_B=X^J$.
For any s_i, we have either ($a^i_{ef}=0$) or ($a^i_{ef}>0$). Therefore, S^J_B in (4.7) can be re-written as

$$S^J_B=\{s_i|a^i_{ef}=0 \text{ and } \frac{a^i_{ef}}{F+a^i_{ep}}>\frac{F}{F+a^f_{ep}}, 1\leq i\leq n\}\cup\{s_i|a^i_{ef}>0 \text{ and } \frac{a^i_{ef}}{F+a^i_{ep}}>\frac{F}{F+a^f_{ep}}, 1\leq i\leq n\}$$

Consider the case that ($a^i_{ef}=0$). Since $F>0$ and $(F+a^f_{ep})>0$ after Lemma 4.1, then we have $\frac{a^i_{ef}}{F+a^i_{ep}}=\frac{0}{F+a^i_{ep}}=0<\frac{F}{F+a^f_{ep}}$, which is contradictory to $\frac{a^i_{ef}}{F+a^i_{ep}}>\frac{F}{F+a^f_{ep}}$. Thus, $\{s_i|a^i_{ef}=0 \text{ and } \frac{a^i_{ef}}{F+a^i_{ep}}>\frac{F}{F+a^f_{ep}}, 1\leq i\leq n\}=\emptyset$, and hence we have

$$S^J_B=\{s_i|a^i_{ef}>0 \text{ and } \frac{a^i_{ef}}{F+a^i_{ep}}>\frac{F}{F+a^f_{ep}}, 1\leq i\leq n\} \tag{4.9}$$

— Assume that $s_i\in S^J_B$. Refer to (4.9), we have ($a^i_{ef}>0$ and $\frac{a^i_{ef}}{F+a^i_{ep}}>\frac{F}{F+a^f_{ep}}$). Since $a^i_{ef}>0$, $F>0$, $(F+a^i_{ep})>0$ and $(F+a^f_{ep})>0$, $\frac{a^i_{ef}}{F+a^i_{ep}}>\frac{F}{F+a^f_{ep}}$ implies $\frac{F+a^i_{ep}}{a^i_{ef}}<\frac{F+a^f_{ep}}{F}$. After re-arranging the terms, we have $1+\frac{a^f_{ep}}{F}-\frac{F}{a^i_{ef}}-\frac{a^i_{ep}}{a^i_{ef}}>0$. Thus, $s_i\in X^J$ after (4.5). Therefore, $S^J_B\subseteq X^J$.

— Assume that $s_i\in X^J$. Refer to (4.5), we have ($a^i_{ef}>0$ and $1+\frac{a^f_{ep}}{F}-\frac{F}{a^i_{ef}}-\frac{a^i_{ep}}{a^i_{ef}}>0$). After re-arranging the terms, $1+\frac{a^f_{ep}}{F}-\frac{F}{a^i_{ef}}-\frac{a^i_{ep}}{a^i_{ef}}>0$ becomes $\frac{F+a^i_{ep}}{a^i_{ef}}<\frac{F+a^f_{ep}}{F}$, which implies $\frac{a^i_{ef}}{F+a^i_{ep}}>\frac{F}{F+a^f_{ep}}$ because $a^i_{ef}>0$, $F>0$, $(F+a^i_{ep})>0$ and $(F+a^f_{ep})>0$. It follows from (4.9) that $s_i\in S^J_B$. Therefore, $X^J\subseteq S^J_B$.

In summary, we have proved $S^J_B=X^J$.

(B) To prove that $S^J_A=Z^J$.
S^J_A in (4.8) can be re-written as

$$S^J_A=\{s_i|(a^i_{ef}=0 \text{ and } \frac{a^i_{ef}}{F+a^i_{ep}}<\frac{F}{F+a^f_{ep}}) \text{ or } (a^i_{ef}>0 \text{ and } \frac{a^i_{ef}}{F+a^i_{ep}}<\frac{F}{F+a^f_{ep}}), 1\leq i\leq n\}$$

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

$$S^J_A=\{s_i|(a^i_{ef}=0) \text{ or } (a^i_{ef}>0 \text{ and } \frac{a^i_{ef}}{F+a^i_{ep}}<\frac{F}{F+a^f_{ep}}, 1\leq i\leq n\} \tag{4.10}$$

— Assume that $s_i \in S_A^J$. Refer to (4.10), we have $(a_{ef}^i=0)$ or $(a_{ef}^i>0$ and $\frac{a_{ef}^i}{F+a_{ep}^i}<\frac{F}{F+a_{ep}^f})$. Consider the sub-case that $(a_{ef}^i=0)$. Immediately after (4.6), we have $s_i \in Z^J$. Now, consider the sub-case that $(a_{ef}^i>0$ and $\frac{a_{ef}^i}{F+a_{ep}^i}<\frac{F}{F+a_{ep}^f})$. Since $a_{ef}^i>0$, $F>0$, $(F+a_{ep}^i)>0$ and $(F+a_{ep}^f)>0$, $\frac{a_{ef}^i}{F+a_{ep}^i}<\frac{F}{F+a_{ep}^f}$ implies $\frac{F+a_{ep}^i}{a_{ef}^i}>\frac{F+a_{ep}^f}{F}$. After re-arranging the terms, we have $1+\frac{a_{ep}^f}{F}-\frac{F}{a_{ef}^i}-\frac{a_{ep}^i}{a_{ef}^i}<0$. Thus, $s_i \in Z^J$ after (4.6). Therefore, $S_A^J \subseteq Z^J$.

— Assume that $s_i \in Z^J$. Refer to (4.6), we have $(a_{ef}^i=0)$ or $(a_{ef}^i>0$ and $1+\frac{a_{ep}^f}{F}-\frac{F}{a_{ef}^i}-\frac{a_{ep}^i}{a_{ef}^i}<0)$. Consider the sub-case that $(a_{ef}^i=0)$. Immediately after (4.10), we have $s_i \in S_A^J$. Now, consider the sub-case that $(a_{ef}^i>0$ and $1+\frac{a_{ep}^f}{F}-\frac{F}{a_{ef}^i}-\frac{a_{ep}^i}{a_{ef}^i}<0)$. After re-arranging the terms, $1+\frac{a_{ep}^f}{F}-\frac{F}{a_{ef}^i}-\frac{a_{ep}^i}{a_{ef}^i}<0$ becomes $\frac{F+a_{ep}^i}{a_{ef}^i}>\frac{F+a_{ep}^f}{F}$, which implies $\frac{a_{ef}^i}{F+a_{ep}^i}<\frac{F}{F+a_{ep}^f}$ because $a_{ef}^i>0$, $F>0$, $(F+a_{ep}^i)>0$ and $(F+a_{ep}^f)>0$. It follows from (4.10) that $s_i \in S_A^J$. Therefore, $Z^J \subseteq S_A^J$.

In summary, we have proved $S_A^J=Z^J$.

In conclusion, we have proved that $S_B^J=X^J$ and $S_A^J=Z^J$. □

With Lemma 4.5, we are going to prove the following proposition.

PROPOSITION 4.6. *ER2 → ER3.*

PROOF. In order to prove ER2 → ER3, it is sufficient to prove Jaccard → Tarantula. As proved in Example 4.4, for Tarantula, S_B^T and S_A^T are equal to sets defined in (4.1) and (4.3), respectively, as follows.

$$(4.1): S_B^T=\{s_i | a_{ef}^i>0 \text{ and } \frac{a_{ep}^f}{F}-\frac{a_{ep}^i}{a_{ef}^i}>0, 1\leq i\leq n\}$$

$$(4.3): S_A^T=\{s_i | (a_{ef}^i=0) \text{ or } (a_{ef}^i>0 \text{ and } \frac{a_{ep}^f}{F}-\frac{a_{ep}^i}{a_{ef}^i}<0), 1\leq i\leq n\}$$

It follows from Lemma 4.5 that S_B^J and S_A^J are equal to the sets defined in (4.5) and (4.6), respectively, as follows.

$$(4.5): S_B^J=\{s_i | a_{ef}^i>0 \text{ and } 1+\frac{a_{ep}^f}{F}-\frac{F}{a_{ef}^i}-\frac{a_{ep}^i}{a_{ef}^i}>0, 1\leq i\leq n\}$$

$$(4.6): S_A^J=\{s_i | (a_{ef}^i=0) \text{ or } (a_{ef}^i>0 \text{ and } 1+\frac{a_{ep}^f}{F}-\frac{F}{a_{ef}^i}-\frac{a_{ep}^i}{a_{ef}^i}<0), 1\leq i\leq n\}$$

Refer to (4.5) and (4.6), after re-arranging the terms in $1+\frac{a_{ep}^f}{F}-\frac{F}{a_{ef}^i}-\frac{a_{ep}^i}{a_{ef}^i}$, we have

$$1+\frac{a_{ep}^f}{F}-\frac{F}{a_{ef}^i}-\frac{a_{ep}^i}{a_{ef}^i}=(\frac{a_{ep}^f}{F}-\frac{a_{ep}^i}{a_{ef}^i})+(1-\frac{F}{a_{ef}^i})$$

Since $1-\frac{F}{a^i_{ef}} \leq 0$ after Lemma 4.1, we have

$$1+\frac{a^f_{ep}}{F}-\frac{F}{a^i_{ef}}-\frac{a^i_{ep}}{a^i_{ef}} \leq \frac{a^f_{ep}}{F}-\frac{a^i_{ep}}{a^i_{ef}} \tag{4.11}$$

(A) To prove that $S^J_B \subseteq S^T_B$.

Assume $s_i \in S^J_B$. Then, we have $(a^i_{ef}>0$ and $1+\frac{a^f_{ep}}{F}-\frac{F}{a^i_{ef}}-\frac{a^i_{ep}}{a^i_{ef}}>0)$ after (4.5). It follows from (4.11) that $\frac{a^f_{ep}}{F}-\frac{a^i_{ep}}{a^i_{ef}}>0$. Thus, $s_i \in S^T_B$ after (4.1). Therefore, $S^J_B \subseteq S^T_B$.

(B) To prove that $S^T_A \subseteq S^J_A$.

Assume $s_i \in S^T_A$. Then, we have either $(a^i_{ef}=0)$ or $(a^i_{ef}>0$ and $\frac{a^f_{ep}}{F}-\frac{a^i_{ep}}{a^i_{ef}}<0)$ after (4.3).

— Consider the case that $(a^i_{ef}=0)$. Immediately after (4.6), $s_i \in S^J_A$.

— Consider the case that $(a^i_{ef}>0$ and $\frac{a^f_{ep}}{F}-\frac{a^i_{ep}}{a^i_{ef}}<0)$. Then, it follows from (4.11) that $1+\frac{a^f_{ep}}{F}-\frac{F}{a^i_{ef}}-\frac{a^i_{ep}}{a^i_{ef}}<0$. Thus, $s_i \in S^J_A$ after (4.6).

In summary, we have proved $S^T_A \subseteq S^J_A$.

In conclusion, we have $S^J_B \subseteq S^T_B$ and $S^T_A \subseteq S^J_A$. Immediately after Theorem 3.6, Jaccard $\rightarrow$ Tarantula. Thus, it follows after Proposition 4.3 that ER2 $\rightarrow$ ER3. □

We now present the following proposition about the relation between ER2 and ER4.

PROPOSITION 4.7. *ER2 $\rightarrow$ ER4.*

PROOF. In order to prove ER2 $\rightarrow$ ER4, it is sufficient to prove Jaccard $\rightarrow$ Wong2. As stated in Table I, Wong2 is defined as follows.

$$R_{W2}(s_i) = a^i_{ef} - a^i_{ep}$$

After Lemma 4.2 and Definition 3.1, we have

$$S^{W2}_B = \{s_i | a^i_{ef} - a^i_{ep} > F - a^f_{ep}, 1 \leq i \leq n\}$$
$$S^{W2}_A = \{s_i | a^i_{ef} - a^i_{ep} < F - a^f_{ep}, 1 \leq i \leq n\}$$

Use the set descriptions for S^J_B and S^J_A in Lemma 4.5. Similar to the proof of Proposition 4.6, we can prove that $S^J_B \subseteq S^{W2}_B$ and $S^{W2}_A \subseteq S^J_A$, and hence Jaccard $\rightarrow$ Wong2. After Proposition 4.3, ER2 $\rightarrow$ ER4. □

Now, let us consider the relations among ER2, Ochiai and Kulczynski2. To prove their relations, we need the following two lemmas for Ochiai and Kulczynski2, of which the proofs are omitted as they are similar to the proof of Lemma 4.5[1].

LEMMA 4.8. *For Ochiai, we have*

$$S^O_B = \{s_i | a^i_{ef} > 0 \textit{ and } (1 + \frac{a^f_{ep}}{F})\frac{a^i_{ef}}{F} - 1 - \frac{a^i_{ep}}{a^i_{ef}} > 0, 1 \leq i \leq n\} \tag{4.12}$$

$$S^O_A = \{s_i | (a^i_{ef} = 0) \textit{ or } (a^i_{ef} > 0 \textit{ and } (1 + \frac{a^f_{ep}}{F})\frac{a^i_{ef}}{F} - 1 - \frac{a^i_{ep}}{a^i_{ef}} < 0), 1 \leq i \leq n\} \tag{4.13}$$

[1]Readers who are interested in the proofs that are omitted in this paper because of their similarity with other proofs, may consult [Xie 2012].

LEMMA 4.9. *For Kulczynski2, we have*

$$S_B^{K2}=\{s_i|a_{ef}^i>0 \textit{ and } \frac{a_{ef}^i F+a_{ef}^i a_{ep}^f-F^2}{F^2+(F+a_{ep}^f)(F-a_{ef}^i)}-\frac{a_{ep}^i}{a_{ef}^i}>0, 1\leq i\leq n\} \tag{4.14}$$

$$S_A^{K2}=\{s_i|(a_{ef}^i=0) \textit{ or } (a_{ef}^i>0 \textit{ and } \frac{a_{ef}^i F+a_{ef}^i a_{ep}^f-F^2}{F^2+(F+a_{ep}^f)(F-a_{ef}^i)}-\frac{a_{ep}^i}{a_{ef}^i}<0), 1\leq i\leq n\} \tag{4.15}$$

With Lemma 4.5, Lemma 4.8 and Lemma 4.9, we are going to prove the following two propositions.

PROPOSITION 4.10. *Ochiai → ER2.*

PROOF. In order to prove Ochiai → ER2, it is sufficient to prove Ochiai → Jaccard. It follows from Lemma 4.5 that S_B^J and S_A^J are equal to the sets defined in (4.5) and (4.6), respectively, as follows.

$$(4.5): S_B^J=\{s_i|a_{ef}^i>0 \text{ and } 1+\frac{a_{ep}^f}{F}-\frac{F}{a_{ef}^i}-\frac{a_{ep}^i}{a_{ef}^i}>0, 1\leq i\leq n\}$$

$$(4.6): S_A^J=\{s_i|(a_{ef}^i=0) \text{ or } (a_{ef}^i>0 \text{ and } 1+\frac{a_{ep}^f}{F}-\frac{F}{a_{ef}^i}-\frac{a_{ep}^i}{a_{ef}^i}<0), 1\leq i\leq n\}$$

And it follows from Lemma 4.8 that S_B^O and S_A^O are equal to the sets defined in (4.12) and (4.13), respectively, as follows.

$$(4.12): S_B^O=\{s_i|a_{ef}^i>0 \text{ and } (1+\frac{a_{ep}^f}{F})\frac{a_{ef}^i}{F}-1-\frac{a_{ep}^i}{a_{ef}^i}>0, 1\leq i\leq n\}$$

$$(4.13): S_A^O=\{s_i|(a_{ef}^i=0) \text{ or } (a_{ef}^i>0 \text{ and } (1+\frac{a_{ep}^f}{F})\frac{a_{ef}^i}{F}-1-\frac{a_{ep}^i}{a_{ef}^i}<0), 1\leq i\leq n\}$$

Let f_J and f_O denote the following expressions.

$$f_J(s_i)=1+\frac{a_{ep}^f}{F}-\frac{F}{a_{ef}^i}=\frac{a_{ef}^i F+a_{ef}^i a_{ep}^f-F^2}{F a_{ef}^i} \tag{4.16}$$

$$f_O(s_i)=(1+\frac{a_{ep}^f}{F})\frac{a_{ef}^i}{F}-1=\frac{a_{ef}^i F+a_{ef}^i a_{ep}^f-F^2}{F^2} \tag{4.17}$$

(A) To prove that $S_B^O\subseteq S_B^J$.

Assume $s_i\in S_B^O$. Then, we have $(a_{ef}^i>0$ and $f_O(s_i)-\frac{a_{ep}^i}{a_{ef}^i}>0)$ after (4.12) and (4.17). Since $a_{ep}^i\geq 0$ and $a_{ef}^i>0$, then $\frac{a_{ep}^i}{a_{ef}^i}\geq 0$ and thus $f_O(s_i)>0$. Then from (4.17), we have $(a_{ef}^i F+a_{ef}^i a_{ep}^f-F^2)>0$ because $F^2>0$. It follows from Lemma 4.1 and $a_{ef}^i>0$ that $F^2\geq F a_{ef}^i>0$. Then, from (4.16) and (4.17), we have $f_J(s_i)\geq f_O(s_i)$. As a consequence, $f_J(s_i)-\frac{a_{ep}^i}{a_{ef}^i}\geq f_O(s_i)-\frac{a_{ep}^i}{a_{ef}^i}>0$. It follows from (4.5) that $s_i\in S_B^J$. Thus, $S_B^O\subseteq S_B^J$.

(B) To prove that $S_A^J\subseteq S_A^O$.

Assume $s_i\in S_A^J$. Then, we have either $(a_{ef}^i=0)$ or $(a_{ef}^i>0$ and $f_J(s_i)-\frac{a_{ep}^i}{a_{ef}^i}<0)$ after (4.6).

— Consider the case that $(a_{ef}^i=0)$. Immediately, we have $s_i\in S_A^O$ after (4.13).

— Consider the case that $(a^i_{ef}>0$ and $f_J(s_i)-\frac{a^i_{ep}}{a^i_{ef}}<0)$. Consider the sub-case that $f_J(s_i)<0$. Since $Fa^i_{ef}>0$, we have $(a^i_{ef}F+a^i_{ef}a^f_{ep}-F^2)<0$ from (4.16). Then, $f_O(s_i)<0$ from (4.17) because $F^2>0$. As a consequence, $f_O(s_i)-\frac{a^i_{ep}}{a^i_{ef}}<0$. Hence, $s_i\in S^O_A$ after (4.13). Next consider the sub-case that $f_J(s_i)=0$. Then, $(a^i_{ef}F+a^i_{ef}a^f_{ep}-F^2)=0$. Thus we have $f_O(s_i)=f_J(s_i)=0$ from (4.17). Furthermore, since $f_J(s_i)-\frac{a^i_{ep}}{a^i_{ef}}<0$ and $f_J(s_i)=0$, we have $\frac{a^i_{ep}}{a^i_{ef}}>0$. As a consequence, $f_O(s_i)-\frac{a^i_{ep}}{a^i_{ef}}<0$. Hence, $s_i\in S^O_A$ after (4.13). Finally, consider the sub-case that $f_J(s_i)>0$. Since $Fa^i_{ef}>0$, we have $(a^i_{ef}F+a^i_{ef}a^f_{ep}-F^2)>0$. It follows from Lemma 4.1 that $F^2\geq Fa^i_{ef}$. Then, from (4.16) and (4.17), we have $f_J(s_i)\geq f_O(s_i)$. As a consequence, $f_O(s_i)-\frac{a^i_{ep}}{a^i_{ef}}\leq f_J(s_i)-\frac{a^i_{ep}}{a^i_{ef}}<0$. Thus, $s_i\in S^O_A$ after (4.13).

In summary, we have proved $S^J_A\subseteq S^O_A$.

In conclusion, we have $S^O_B\subseteq S^J_B$ and $S^J_A\subseteq S^O_A$. Immediately after Theorem 3.6, Ochiai $\rightarrow$ Jaccard. It follows after Proposition 4.3 that Ochiai $\rightarrow$ ER2. □

PROPOSITION 4.11. *Kulczynski2 $\rightarrow$ Ochiai.*

PROOF. Use the set descriptions for S^O_B and S^O_A in Lemma 4.8 and the set descriptions for S^{K2}_B and S^{K2}_A in Lemma 4.9. Similar to the proof of Proposition 4.10, we can prove that $S^{K2}_B\subseteq S^O_B$ and $S^O_A\subseteq S^{K2}_A$, and hence Kulczynski2 $\rightarrow$ Ochiai. □

Figure 3(a) follows immediately from Proposition 4.6, Proposition 4.7, Proposition 4.10 and Proposition 4.11. Now, let us consider Figure 3(b) that states the relation between M2 and AMPLE2, of which the proof needs the following two lemmas for M2 and AMPLE2. The proofs of these two lemmas are omitted since they are similar to the proof of Lemma 4.5.

LEMMA 4.12. *For M2, we have*

$$S^{M2}_B=\{s_i|a^i_{ef}>0 \textit{ and } \frac{P+a^f_{ep}}{F}-\frac{2F+P}{a^i_{ef}}+2-\frac{a^i_{ep}}{a^i_{ef}}>0, 1\leq i\leq n\} \tag{4.18}$$

$$S^{M2}_A=\{s_i|(a^i_{ef}=0) \textit{ or } (a^i_{ef}>0 \textit{ and } \frac{P+a^f_{ep}}{F}-\frac{2F+P}{a^i_{ef}}+2-\frac{a^i_{ep}}{a^i_{ef}}<0), 1\leq i\leq n\} \tag{4.19}$$

LEMMA 4.13. *For AMPLE2, we have*

$$S^A_B=\{s_i|a^i_{ef}>0 \textit{ and } \frac{Pa^i_{ef}-PF+Fa^f_{ep}}{Fa^i_{ef}}-\frac{a^i_{ep}}{a^i_{ef}}>0, 1\leq i\leq n\} \tag{4.20}$$

$$S^A_A=\{s_i|(a^i_{ef}=0) \textit{ or } (a^i_{ef}>0 \textit{ and } \frac{Pa^i_{ef}-PF+Fa^f_{ep}}{Fa^i_{ef}}-\frac{a^i_{ep}}{a^i_{ef}}<0), 1\leq i\leq n\} \tag{4.21}$$

With the above two lemmas, we can prove the following proposition.

PROPOSITION 4.14. *M2 $\rightarrow$ AMPLE2.*

PROOF. Use the set descriptions for S^{M2}_B and S^{M2}_A in Lemma 4.12 and the set descriptions for S^A_B and S^A_A in Lemma 4.13. Similar to the proof of Proposition 4.6, we can prove that $S^{M2}_B\subseteq S^A_B$ and $S^A_A\subseteq S^{M2}_A$, and hence M2 $\rightarrow$ AMPLE2. □

Figure 3 implies that if Kulczynski2 is not a maximal formula, Ochiai, ER2, ER3 and ER4 can never be maximal; if M2 is not a maximal formula, AMPLE2 can never be a maximal formula. As a matter of fact, all formulas in Figure 3 are not maximal formulas, because apart from the performance hierarchy chains shown in Figure 3, there exists another hierarchy chain shown in Figure 4, where all relations "$\rightarrow$" are strictly "*better*" relations. Therefore, Kulczynski2 and M2, as well as ER6, Arithmetic Mean, Cohen, Fleiss and Wong3, are not maximal formulas.

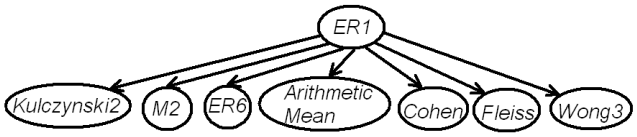

Fig. 4. Another performance hierarchy chain of risk evaluation formulas

In order to prove the relations shown in Figure 4, we need the following lemma for Naish1 of ER1.

LEMMA 4.15. *For Naish1, we have* $S_B^{N1}=X^{N1}$ *and* $S_A^{N1}=Z^{N1}$*, where*

$$X^{N1}=\{s_i|a_{ef}^i=F \textit{ and } a_{ep}^f-a_{ep}^i>0, 1\leq i\leq n\} \tag{4.22}$$

$$Z^{N1}=\{s_i|(a_{ef}^i < F) \textit{ or } (a_{ef}^i=F \textit{ and } a_{ep}^f-a_{ep}^i<0), 1\leq i\leq n\} \tag{4.23}$$

PROOF. As stated in Table I, Naish1 is defined as follows.

$$R_{N1}(s_i) = \begin{cases} -1 & \text{if } a_{ef}^i<F \\ P-a_{ep}^i & \text{if } a_{ef}^i=F \end{cases}$$

After Definition 3.1, we have

$$S_B^{N1}=\{s_i|(a_{ef}^i<F \text{ and } -1>P-a_{ep}^f) \text{ or } (a_{ef}^i=F \text{ and } P-a_{ep}^i>P-a_{ep}^f), 1\leq i\leq n\}$$

which can be re-written as

$$S_B^{N1}=\{s_i|a_{ef}^i<F \text{ and } -1>P-a_{ep}^f, 1\leq i\leq n\}\cup\{s_i|a_{ef}^i=F \text{ and } a_{ep}^f-a_{ep}^i>0, 1\leq i\leq n\}$$

Since $(-1<P-a_{ep}^f)$ after Lemma 4.1, we have $\{s_i|a_{ef}^i<F \text{ and } -1>P-a_{ep}^f, 1\leq i\leq n\}=\emptyset$. Therefore, $S_B^{N1}=\{s_i|a_{ef}^i=F \text{ and } a_{ep}^f-a_{ep}^i>0, 1\leq i\leq n\}=X^{N1}$.

Now, consider S_A^{N1}. After the definition of Naish1 and Definition 3.1, we have

$$S_A^{N1}=\{s_i|(a_{ef}^i<F \text{ and } -1<P-a_{ep}^f) \text{ or } (a_{ef}^i=F \text{ and } P-a_{ep}^i<P-a_{ep}^f), 1\leq i\leq n\}$$

Since $(-1<P-a_{ep}^f)$ after Lemma 4.1, $(a_{ef}^i<F \text{ and } -1<P-a_{ep}^f)$ is logically equivalent to $(a_{ef}^i<F)$. Therefore, S_A^{N1} becomes $\{s_i|(a_{ef}^i<F) \text{ or } (a_{ef}^i=F \text{ and } a_{ep}^f-a_{ep}^i<0), 1\leq i\leq n\}$. That is, $S_A^{N1}=Z^{N1}$. □

With Lemma 4.9 and Lemma 4.15, we can prove that ER1 $\rightarrow$ Kulczynski2.

PROPOSITION 4.16. *ER1* $\rightarrow$ *Kulczynski2.*

PROOF. In order to prove ER1 $\rightarrow$ Kulczynski2, it is sufficient to prove Naish1 $\rightarrow$ Kulczynski2. It follows from Lemma 4.9 that S_B^{K2} and S_A^{K2} are equal to the sets defined in (4.14) and (4.15), respectively, as follows.

$$(4.14): S_B^{K2}=\{s_i|a_{ef}^i>0 \text{ and } \frac{a_{ef}^i F+a_{ef}^i a_{ep}^f-F^2}{F^2+(F+a_{ep}^f)(F-a_{ef}^i)}-\frac{a_{ep}^i}{a_{ef}^i}>0, 1\leq i\leq n\}$$

$$(4.15): S_A^{K2}=\{s_i|(a_{ef}^i=0) \text{ or } (a_{ef}^i>0 \text{ and } \frac{a_{ef}^i F+a_{ef}^i a_{ep}^f-F^2}{F^2+(F+a_{ep}^f)(F-a_{ef}^i)}-\frac{a_{ep}^i}{a_{ef}^i}<0), 1\leq i\leq n\}$$

It follows from Lemma 4.15 that S_B^{N1} and S_A^{N1} are equal to the sets defined in (4.22) and (4.23), respectively, as follows.

$$(4.22): S_B^{N1}=\{s_i|a_{ef}^i=F \text{ and } a_{ep}^f-a_{ep}^i>0, 1\leq i\leq n\}$$

$$(4.23): S_A^{N1}=\{s_i|(a_{ef}^i<F) \text{ or } (a_{ef}^i=F \text{ and } a_{ep}^f-a_{ep}^i<0), 1\leq i\leq n\}$$

(A) To prove that $S_B^{N1}\subseteq S_B^{K2}$.
Assume $s_i\in S_B^{N1}$. Then, $a_{ef}^i=F>0$ and $(a_{ep}^f-a_{ep}^i)>0$ after (4.22). As a consequence, we have $\frac{a_{ef}^i F+a_{ef}^i a_{ep}^f-F^2}{F^2+(F+a_{ep}^f)(F-a_{ef}^i)}-\frac{a_{ep}^i}{a_{ef}^i}=\frac{F^2+Fa_{ep}^f-F^2-Fa_{ep}^i}{F^2}=\frac{a_{ep}^f-a_{ep}^i}{F}>0$. Therefore, $s_i\in S_B^{K2}$ after (4.14). Thus, $S_B^{N1}\subseteq S_B^{K2}$.

(B) To prove that $S_A^{K2}\subseteq S_A^{N1}$.
Suppose $s_i\in S_A^{K2}$. Then we have either $(a_{ef}^i=0)$ or $(a_{ef}^i>0$ and $\frac{a_{ef}^i F+a_{ef}^i a_{ep}^f-F^2}{F^2+(F+a_{ep}^f)(F-a_{ef}^i)}-\frac{a_{ep}^i}{a_{ef}^i}<0)$ after (4.15).

— Consider the case that $(a_{ef}^i=0)$. Obviously, $a_{ef}^i<F$. Immediately after (4.23), $s_i\in S_A^{N1}$.

— Consider the case that $(a_{ef}^i>0$ and $\frac{a_{ef}^i F+a_{ef}^i a_{ep}^f-F^2}{F^2+(F+a_{ep}^f)(F-a_{ef}^i)}-\frac{a_{ep}^i}{a_{ef}^i}<0)$. Consider the sub-case that $0<a_{ef}^i<F$. After (4.23), we have $s_i\in S_A^{N1}$. Next, consider the sub-case that $a_{ef}^i=F$. Then we have $\frac{a_{ef}^i F+a_{ef}^i a_{ep}^f-F^2}{F^2+(F+a_{ep}^f)(F-a_{ef}^i)}-\frac{a_{ep}^i}{a_{ef}^i}=\frac{a_{ep}^f-a_{ep}^i}{F}$. Since $\frac{a_{ef}^i F+a_{ef}^i a_{ep}^f-F^2}{F^2+(F+a_{ep}^f)(F-a_{ef}^i)}-\frac{a_{ep}^i}{a_{ef}^i}<0$ and $F>0$, we have $(a_{ep}^f-a_{ep}^i)<0$. Thus, $s_i\in S_A^{N1}$ after (4.23).

In summary, we have proved $S_A^{K2}\subseteq S_A^{N1}$.

In conclusion, we have $S_B^{N1}\subseteq S_B^{K2}$ and $S_A^{K2}\subseteq S_A^{N1}$. Immediately after Theorem 3.6, Naish1 $\rightarrow$ Kulczynski2. It follows after Proposition 4.3 that ER1 $\rightarrow$ Kulczynski2. □

The following Proposition 4.17, Proposition 4.18, Proposition 4.19, Proposition 4.20 and Proposition 4.21 can be proved in a similar way as Proposition 4.16, and hence their proofs are omitted.

PROPOSITION 4.17. *ER1 $\rightarrow$ M2.*

PROPOSITION 4.18. *ER1 $\rightarrow$ ER6.*

PROPOSITION 4.19. *ER1 $\rightarrow$ Arithmetic Mean.*

PROPOSITION 4.20. *ER1 $\rightarrow$ Cohen.*

PROPOSITION 4.21. *ER1 $\rightarrow$ Fleiss.*

Prior to the presentation of the relation between ER1 and Wong3, we need the following lemma for Wong3.

LEMMA 4.22. *For Wong3, we have its* S_B^{W3}, S_F^{W3} *and* S_A^{W3} *as follows.*

1. *If* $a_{ep}^f \leq 2$, *then we have* $S_B^{W3}=X_1^{W3}$ *and* $S_A^{W3}=Z_1^{W3}$, *where*

$$X_1^{W3}=\{s_i|a_{ep}^i \leq 2 \textit{ and } (a_{ef}^i-F)+(a_{ep}^f-a_{ep}^i)>0, 1\leq i\leq n\} \tag{4.24}$$

$$Z_1^{W3}=\{s_i|(a_{ep}^i>2) \textit{ or } (a_{ep}^i \leq 2 \textit{ and } (a_{ef}^i-F)+(a_{ep}^f-a_{ep}^i)<0), 1\leq i\leq n\} \tag{4.25}$$

2. *If* $2<a_{ep}^f \leq 10$, *then we have* $S_B^{W3}=X_2^{W3}$ *and* $S_A^{W3}=Z_2^{W3}$, *where*

$$\begin{aligned} X_2^{W3}=\{s_i|&(a_{ep}^i \leq 2 \textit{ and } (a_{ef}^i-F)+(0.1a_{ep}^f-a_{ep}^i)+1.8>0) \textit{ or} \\ &(2<a_{ep}^i \leq 10 \textit{ and } (a_{ef}^i-F)+(0.1a_{ep}^f-0.1a_{ep}^i)>0), 1\leq i\leq n\} \end{aligned} \tag{4.26}$$

$$\begin{aligned} Z_2^{W3}=\{s_i|&(a_{ep}^i \leq 2 \textit{ and } (a_{ef}^i-F)+(0.1a_{ep}^f-a_{ep}^i)+1.8<0) \textit{ or} \\ &(2<a_{ep}^i \leq 10 \textit{ and } (a_{ef}^i-F)+(0.1a_{ep}^f-0.1a_{ep}^i)<0) \textit{ or} \\ &(a_{ep}^i>10), 1\leq i\leq n\} \end{aligned} \tag{4.27}$$

3. *If* $a_{ep}^f>10$, *then we have* $S_B^{W3}=X_3^{W3}$ *and* $S_A^{W3}=Z_3^{W3}$, *where*

$$\begin{aligned} X_3^{W3}=\{s_i|&(a_{ep}^i \leq 2 \textit{ and } (a_{ef}^i-F)+(0.001a_{ep}^f-a_{ep}^i)+2.79>0) \textit{ or} \\ &(2<a_{ep}^i \leq 10 \textit{ and } (a_{ef}^i-F)+(0.001a_{ep}^f-0.1a_{ep}^i)+0.99>0) \textit{ or} \\ &(a_{ep}^i>10 \textit{ and } (a_{ef}^i-F)+(0.001a_{ep}^f-0.001a_{ep}^i)>0), 1\leq i\leq n\} \end{aligned} \tag{4.28}$$

$$\begin{aligned} Z_3^{W3}=\{s_i|&(a_{ep}^i \leq 2 \textit{ and } (a_{ef}^i-F)+(0.001a_{ep}^f-a_{ep}^i)+2.79<0) \textit{ or} \\ &(2<a_{ep}^i \leq 10 \textit{ and } (a_{ef}^i-F)+(0.001a_{ep}^f-0.1a_{ep}^i)+0.99<0) \textit{ or} \\ &(a_{ep}^i>10 \textit{ and } (a_{ef}^i-F)+(0.001a_{ep}^f-0.001a_{ep}^i)<0), 1\leq i\leq n\} \end{aligned} \tag{4.29}$$

PROOF. As stated in Table I, 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. **Case 1:** Assume $a_{ep}^f \leq 2$.
Then, $R_{W3}(s_f) = F-a_{ep}^f$. After Definition 3.1, we have

$$\begin{aligned} S_B^{W3}=\{&s_i|a_{ep}^i \leq 2 \text{ and } a_{ef}^i-a_{ep}^i>F-a_{ep}^f, 1\leq i\leq n\} \\ \cup\{&s_i|2<a_{ep}^i \leq 10 \text{ and } a_{ef}^i-2-0.1(a_{ep}^i-2)>F-a_{ep}^f, 1\leq i\leq n\} \\ \cup\{&s_i|a_{ep}^i>10 \text{ and } a_{ef}^i-2.8-0.001(a_{ep}^i-10)>F-a_{ep}^f, 1\leq i\leq n\} \end{aligned}$$

By re-arranging the terms, we have

$$\begin{aligned} S_B^{W3}=\{&s_i|a_{ep}^i \leq 2 \text{ and } (a_{ef}^i-F)+(a_{ep}^f-a_{ep}^i)>0, 1\leq i\leq n\} \\ \cup\{&s_i|2<a_{ep}^i \leq 10 \text{ and } (a_{ef}^i-F)+(a_{ep}^f-0.1a_{ep}^i)-1.8>0, 1\leq i\leq n\} \\ \cup\{&s_i|a_{ep}^i>10 \text{ and } (a_{ef}^i-F)+(a_{ep}^f-0.001a_{ep}^i)-2.79>0, 1\leq i\leq n\} \end{aligned} \tag{4.30}$$

Consider the case that $(a_{ep}^i>10)$. We have $(a_{ep}^f-0.001a_{ep}^i)-2.79<0$ because $a_{ep}^f \leq 2$ and $a_{ep}^i>10$. Since $a_{ef}^i-F \leq 0$, we have $(a_{ef}^i-F)+(a_{ep}^f-0.001a_{ep}^i)-2.79<0$,

which is contradictory to $(a^i_{ef}-F)+(a^f_{ep}-0.001a^i_{ep})-2.79>0$. Thus, $\{s_i|a^i_{ep}>10$ and $(a^i_{ef}-F)+(a^f_{ep}-0.001a^i_{ep})-2.79>0, 1\leq i\leq n\}=\emptyset$. Hence, S^{W3}_B in (4.30) becomes

$$\begin{aligned}S^{W3}_B=&\{s_i|a^i_{ep}\leq 2 \text{ and } (a^i_{ef}-F)+(a^f_{ep}-a^i_{ep})>0, 1\leq i\leq n\}\\ &\cup\{s_i|2<a^i_{ep}\leq 10 \text{ and } (a^i_{ef}-F)+(a^f_{ep}-0.1a^i_{ep})-1.8>0, 1\leq i\leq n\}\end{aligned} \tag{4.31}$$

Next, consider the case that $(2<a^i_{ep}\leq 10)$. Since $a^f_{ep}\leq 2<a^i_{ep}\leq 10$, we have $(a^f_{ep}-0.1a^i_{ep})-1.8<0$. It follows after Lemma 4.1 that $a^i_{ef}-F\leq 0$. Therefore, we have $(a^i_{ef}-F)+(a^f_{ep}-0.1a^i_{ep})-1.8<0$, which is contradictory to $(a^i_{ef}-F)+(a^f_{ep}-0.1a^i_{ep})-1.8>0$. Thus, $\{s_i|2<a^i_{ep}\leq 10$ and $(a^i_{ef}-F)+(a^f_{ep}-0.1a^i_{ep})-1.8>0, 1\leq i\leq n\}=\emptyset$. Therefore, S^{W3}_B in (4.31) becomes

$$S^{W3}_B=\{s_i|a^i_{ep}\leq 2 \text{ and } (a^i_{ef}-F)+(a^f_{ep}-a^i_{ep})>0, 1\leq i\leq n\}=X^{W3}_1$$

After the definition of Wong3 and Definition 3.1, and re-arranging the terms, we have

$$\begin{aligned}S^{W3}_A=&\{s_i|a^i_{ep}\leq 2 \text{ and } (a^i_{ef}-F)+(a^f_{ep}-a^i_{ep})<0, 1\leq i\leq n\}\\ &\cup\{s_i|2<a^i_{ep}\leq 10 \text{ and } (a^i_{ef}-F)+(a^f_{ep}-0.1a^i_{ep})-1.8<0, 1\leq i\leq n\}\\ &\cup\{s_i|a^i_{ep}>10 \text{ and } (a^i_{ef}-F)+(a^f_{ep}-0.001a^i_{ep})-2.79<0, 1\leq i\leq n\}\end{aligned} \tag{4.32}$$

As explained in the above proof of $S^{W3}_B=X^{W3}_1$, since $a^f_{ep}\leq 2$, $(2<a^i_{ep}\leq 10)$ implies $(a^i_{ef}-F)+(a^f_{ep}-0.1a^i_{ep})-1.8<0$, and $(a^i_{ep}>10)$ implies $(a^i_{ef}-F)+(a^f_{ep}-0.001a^i_{ep})-2.79<0$. Thus, $(2<a^i_{ep}\leq 10$ and $(a^i_{ef}-F)+(a^f_{ep}-0.1a^i_{ep})-1.8<0)$ is logically equivalent to $(2<a^i_{ep}\leq 10)$, and $(a^i_{ep}>10$ and $(a^i_{ef}-F)+(a^f_{ep}-0.001a^i_{ep})-2.79<0)$ is logically equivalent to $(a^i_{ep}>10)$. Therefore, S^{W3}_A in (4.32) becomes

$$\begin{aligned}S^{W3}_A=&\{s_i|a^i_{ep}\leq 2 \text{ and } (a^i_{ef}-F)+(a^f_{ep}-a^i_{ep})<0, 1\leq i\leq n\}\\ &\cup\{s_i|2<a^i_{ep}\leq 10, 1\leq i\leq n\}\cup\{s_i|a^i_{ep}>10, 1\leq i\leq n\}\\ =&\{s_i|a^i_{ep}\leq 2 \text{ and } (a^i_{ef}-F)+(a^f_{ep}-a^i_{ep})<0, 1\leq i\leq n\}\cup\{s_i|a^i_{ep}>2, 1\leq i\leq n\}\\ =&\{s_i|(a^i_{ep}>2) \text{ or } (a^i_{ep}\leq 2 \text{ and } (a^i_{ef}-F)+(a^f_{ep}-a^i_{ep})<0), 1\leq i\leq n\}=Z^{W3}_1\end{aligned}$$

2. **Case 2:** Assume $2<a^f_{ep}\leq 10$.
Then, $R_{W3}(s_f)=F-2-0.1(a^f_{ep}-2)=F-0.1a^f_{ep}-1.8$. After Definition 3.1 and re-arranging the terms, we have

$$\begin{aligned}S^{W3}_B=&\{s_i|a^i_{ep}\leq 2 \text{ and } (a^i_{ef}-F)+(0.1a^f_{ep}-a^i_{ep})+1.8>0, 1\leq i\leq n\}\\ &\cup\{s_i|2<a^i_{ep}\leq 10 \text{ and } (a^i_{ef}-F)+(0.1a^f_{ep}-0.1a^i_{ep})>0, 1\leq i\leq n\}\\ &\cup\{s_i|a^i_{ep}>10 \text{ and } (a^i_{ef}-F)+(0.1a^f_{ep}-0.001a^i_{ep})-0.99>0, 1\leq i\leq n\}\end{aligned} \tag{4.33}$$

Consider the case that $a^i_{ep}>10$. Thus, we have $2<a^f_{ep}\leq 10<a^i_{ep}$, which implies $(0.1a^f_{ep}-0.001a^i_{ep})-0.99<0$. It follows after Lemma 4.1 that $a^i_{ef}-F\leq 0$. Therefore, we have $(a^i_{ef}-F)+(0.1a^f_{ep}-0.001a^i_{ep})-0.99<0$, which is contradictory to $(a^i_{ef}-F)+(0.1a^f_{ep}-0.001a^i_{ep})-0.99>0$. Thus, $\{s_i|a^i_{ep}>10$ and $(a^i_{ef}-F)+(0.1a^f_{ep}-0.001a^i_{ep})-0.99>0, 1\leq i\leq n\}=\emptyset$. Then, S^{W3}_B in (4.33) becomes

$$S^{W3}_B=\{s_i|a^i_{ep}\leq 2 \text{ and } (a^i_{ef}-F)+(0.1a^f_{ep}-a^i_{ep})+1.8>0, 1\leq i\leq n\}$$

$$\begin{aligned}
&\cup\{s_i|2<a^i_{ep}\leq 10 \text{ and } (a^i_{ef}-F)+(0.1a^f_{ep}-0.1a^i_{ep})>0, 1\leq i\leq n\}\\
=&\{s_i|(a^i_{ep}\leq 2 \text{ and } (a^i_{ef}-F)+(0.1a^f_{ep}-a^i_{ep})+1.8>0) \text{ or}\\
&(2<a^i_{ep}\leq 10 \text{ and } (a^i_{ef}-F)+(0.1a^f_{ep}-0.1a^i_{ep})>0), 1\leq i\leq n\}=X^{W3}_2
\end{aligned}$$

After the definition of Wong3 and Definition 3.1, and re-arranging the terms, we have

$$\begin{aligned}
S^{W3}_A=&\{s_i|a^i_{ep}\leq 2 \text{ and } (a^i_{ef}-F)+(0.1a^f_{ep}-a^i_{ep})+1.8<0, 1\leq i\leq n\}\\
&\cup\{s_i|2<a^i_{ep}\leq 10 \text{ and } (a^i_{ef}-F)+(0.1a^f_{ep}-0.1a^i_{ep})<0, 1\leq i\leq n\}\\
&\cup\{s_i|a^i_{ep}>10 \text{ and } (a^i_{ef}-F)+(0.1a^f_{ep}-0.001a^i_{ep})-0.99<0, 1\leq i\leq n\} \qquad (4.34)
\end{aligned}$$

As explained in the above proof of $S^{W3}_B=X^{W3}_2$, since $2<a^f_{ep}\leq 10$, $(a^i_{ep}>10)$ implies $(a^i_{ef}-F)+(0.1a^f_{ep}-0.001a^i_{ep})-0.99<0$. Thus, $(a^i_{ep}>10$ and $(a^i_{ef}-F)+(0.1a^f_{ep}-0.001a^i_{ep})-0.99<0)$ is logically equivalent to $(a^i_{ep}>10)$. Therefore, S^{W3}_A in (4.34) becomes

$$\begin{aligned}
S^{W3}_A=&\{s_i|a^i_{ep}\leq 2 \text{ and } (a^i_{ef}-F)+(0.1a^f_{ep}-a^i_{ep})+1.8<0, 1\leq i\leq n\}\\
&\cup\{s_i|2<a^i_{ep}\leq 10 \text{ and } (a^i_{ef}-F)+(0.1a^f_{ep}-0.1a^i_{ep})<0, 1\leq i\leq n\}\cup\{s_i|a^i_{ep}>10, 1\leq i\leq n\}\\
=&\{s_i|(a^i_{ep}\leq 2 \text{ and } (a^i_{ef}-F)+(0.1a^f_{ep}-a^i_{ep})+1.8<0) \text{ or}\\
&(2<a^i_{ep}\leq 10 \text{ and } (a^i_{ef}-F)+(0.1a^f_{ep}-0.1a^i_{ep})<0) \text{ or } (a^i_{ep}>10), 1\leq i\leq n\}\\
=&Z^{W3}_2
\end{aligned}$$

3. **Case 3:** Assume $a^f_{ep}>10$.
Then, $R_{W3}(s_f)=F-2.8-0.001(a^f_{ep}-10)=F-0.001a^f_{ep}-2.79$. After Definition 3.1, we have

$$\begin{aligned}
S^{W3}_B=&\{s_i|a^i_{ep}\leq 2 \text{ and } a^i_{ef}-a^i_{ep}>F-0.001a^f_{ep}-2.79, 1\leq i\leq n\}\\
&\cup\{s_i|2<a^i_{ep}\leq 10 \text{ and } a^i_{ef}-2-0.1(a^i_{ep}-2)>F-0.001a^f_{ep}-2.79, 1\leq i\leq n\}\\
&\cup\{s_i|a^i_{ep}>10 \text{ and } a^i_{ef}-2.8-0.001(a^i_{ep}-10)>F-0.001a^f_{ep}-2.79, 1\leq i\leq n\} \qquad (4.35)
\end{aligned}$$

$$\begin{aligned}
S^{W3}_A=&\{s_i|a^i_{ep}\leq 2 \text{ and } a^i_{ef}-a^i_{ep}<F-0.001a^f_{ep}-2.79, 1\leq i\leq n\}\\
&\cup\{s_i|2<a^i_{ep}\leq 10 \text{ and } a^i_{ef}-2-0.1(a^i_{ep}-2)<F-0.001a^f_{ep}-2.79, 1\leq i\leq n\}\\
&\cup\{s_i|a^i_{ep}>10 \text{ and } a^i_{ef}-2.8-0.001(a^i_{ep}-10)<F-0.001a^f_{ep}-2.79, 1\leq i\leq n\} \qquad (4.36)
\end{aligned}$$

It is obvious that through re-arranging the terms and merging the subsets, we have $S^{W3}_B=X^{W3}_3$ and $S^{W3}_A=Z^{W3}_3$.

□

With Lemma 4.15 and Lemma 4.22, we are now ready to prove the following relation between ER1 and Wong3.

PROPOSITION 4.23. *ER1 → Wong3.*

PROOF. In order to prove ER1 → Wong3, it is sufficient to prove Naish1 → Wong3. It follows from Lemma 4.15 that S^{N1}_B and S^{N1}_A are equal to the sets defined in (4.22) and (4.23), respectively, as follows.

$$\begin{aligned}
&(4.22): S^{N1}_B=\{s_i|a^i_{ef}=F \text{ and } a^f_{ep}-a^i_{ep}>0, 1\leq i\leq n\}\\
&(4.23): S^{N1}_A=\{s_i|(a^i_{ef}<F) \text{ or } (a^i_{ef}=F \text{ and } a^f_{ep}-a^i_{ep}<0), 1\leq i\leq n\}
\end{aligned}$$

For Wong3, as shown in Lemma 4.22, S_B^{W3} and S_A^{W3} are different in three situations. Under each situation, we are going to prove that $S_B^{N1} \subseteq S_B^{W3}$ and $S_A^{W3} \subseteq S_A^{N1}$.

1. **Case 1:** Assume $a_{ep}^f \leq 2$.
It follows from Lemma 4.22 that S_B^{W3} and S_A^{W3} are equal to the sets defined in (4.24) and (4.25), respectively.

(A) To prove that $S_B^{N1} \subseteq S_B^{W3}$.
Assume $s_i \in S_B^{N1}$. Refer to (4.22), we have $(a_{ef}^i = F)$ and $(a_{ep}^f - a_{ep}^i) > 0$. As a consequence, $(a_{ef}^i - F) + (a_{ep}^f - a_{ep}^i) = (a_{ep}^f - a_{ep}^i) > 0$. Furthermore, since $a_{ep}^f \leq 2$, we have $a_{ep}^i < a_{ep}^f \leq 2$. Then, $s_i \in S_B^{W3}$ after (4.24) that defines S_B^{W3}. Thus, $S_B^{N1} \subseteq S_B^{W3}$.

(B) To prove that $S_A^{W3} \subseteq S_A^{N1}$.
Assume $s_i \in S_A^{W3}$. Refer to (4.25) that defines S_A^{W3}, we have either $(a_{ep}^i > 2)$, or $(a_{ep}^i \leq 2$ and $(a_{ef}^i - F) + (a_{ep}^f - a_{ep}^i) < 0)$.
— Consider the case that $(a_{ep}^i > 2)$. Consider the sub-case that $a_{ef}^i < F$. Immediately, we have $s_i \in S_A^{N1}$ after (4.23). Then consider the sub-case that $a_{ef}^i = F$. Since $a_{ep}^i > 2$ and $a_{ep}^f \leq 2$, we have $(a_{ep}^f - a_{ep}^i) < 0$. Thus after (4.23), $s_i \in S_A^{N1}$.
— Consider the case that $(a_{ep}^i \leq 2$ and $(a_{ef}^i - F) + (a_{ep}^f - a_{ep}^i) < 0)$. Consider the sub-case that $a_{ef}^i < F$. Then, $s_i \in S_A^{N1}$ after (4.23). Now consider the sub-case that $a_{ef}^i = F$. We have $(a_{ef}^i - F) + (a_{ep}^f - a_{ep}^i) = (a_{ep}^f - a_{ep}^i)$. Since $(a_{ef}^i - F) + (a_{ep}^f - a_{ep}^i) < 0$, then $(a_{ep}^f - a_{ep}^i) < 0$. After (4.23), $s_i \in S_A^{N1}$.
In summary, we have proved $S_A^{W3} \subseteq S_A^{N1}$.

2. **Case 2:** Assume $2 < a_{ep}^f \leq 10$.
It follows from Lemma 4.22 that S_B^{W3} and S_A^{W3} are equal to the sets descriptions in (4.26) and (4.27), respectively.

(A) To prove that $S_B^{N1} \subseteq S_B^{W3}$.
Assume $s_i \in S_B^{N1}$. Refer to (4.22), we have $(a_{ef}^i = F)$ and $(a_{ep}^f - a_{ep}^i) > 0$. Since $2 < a_{ep}^f \leq 10$, we have $a_{ep}^i < 10$. Consider the following two cases:
— Suppose $a_{ep}^i \leq 2$. Since $a_{ef}^i = F$, we have

$$(a_{ef}^i - F) + (0.1a_{ep}^f - a_{ep}^i) + 1.8 = 0.1(a_{ep}^f - a_{ep}^i) + (1.8 - 0.9a_{ep}^i) > 0$$

because $(a_{ep}^f - a_{ep}^i) > 0$ and $(1.8 - 0.9a_{ep}^i) \geq 0$ after $a_{ep}^i \leq 2$. After (4.26) that defines S_B^{W3}, we have $s_i \in S_B^{W3}$.
— Suppose $2 < a_{ep}^i < 10$. Since $a_{ef}^i = F$, we have

$$(a_{ef}^i - F) + 0.1(a_{ep}^f - a_{ep}^i) = 0.1(a_{ep}^f - a_{ep}^i) > 0$$

because $(a_{ep}^f - a_{ep}^i) > 0$. Thus, $s_i \in S_B^{W3}$ after (4.26).
In summary, we have proved $S_B^{N1} \subseteq S_B^{W3}$.

(B) To prove that $S_A^{W3} \subseteq S_A^{N1}$.
Assume $s_i \in S_A^{W3}$. Refer to (4.27) that defines S_A^{W3}, we have either $(a_{ep}^i > 10)$, $(2 < a_{ep}^i \leq 10$ and $(a_{ef}^i - F) + 0.1(a_{ep}^f - a_{ep}^i) < 0)$, or $(a_{ep}^i \leq 2$ and $(a_{ef}^i - F) + (0.1a_{ep}^f - a_{ep}^i) + 1.8 < 0)$.
— Consider the case that $a_{ep}^i > 10$. Consider the sub-case that $a_{ef}^i < F$. Immediately after (4.23), we have $s_i \in S_A^{N1}$. Then consider the sub-case that $a_{ef}^i = F$. Since $2 < a_{ep}^f \leq 10$ and $a_{ep}^i > 10$, we have $(a_{ep}^f - a_{ep}^i) < 0$. After (4.23), $s_i \in S_A^{N1}$.

— Consider the case that $(2<a^i_{ep}\leq 10$ and $(a^i_{ef}-F)+0.1(a^f_{ep}-a^i_{ep})<0)$. Consider the sub-case that $a^i_{ef}<F$. Then, we have $s_i\in S^{N1}_A$ after (4.23). Now consider the sub-case that $a^i_{ef}=F$. Then, we have $(a^i_{ef}-F)+0.1(a^f_{ep}-a^i_{ep})=0.1(a^f_{ep}-a^i_{ep})$. Since $(a^i_{ef}-F)+0.1(a^f_{ep}-a^i_{ep})<0$, then $(a^f_{ep}-a^i_{ep})<0$. After (4.23), $s_i\in S^{N1}_A$.
— Consider the case that $(a^i_{ep}\leq 2$ and $(a^i_{ef}-F)+(0.1a^f_{ep}-a^i_{ep})+1.8<0)$. Assume further that $a^i_{ef}=F$. Then, we have $(a^i_{ef}-F)+(0.1a^f_{ep}-a^i_{ep})+1.8=0.1a^f_{ep}-a^i_{ep}+1.8<0$. However, it follows from $2<a^f_{ep}\leq 10$ and $a^i_{ep}\leq 2$ that $0.1a^f_{ep}-a^i_{ep}+1.8>0$, which is contradictory to $0.1a^f_{ep}-a^i_{ep}+1.8<0$. Therefore, it is impossible to have $a^i_{ef}=F$ and all statements in this case have $a^i_{ef}<F$. Then, we have $s_i\in S^{N1}_A$ after (4.23).

In summary, we have proved $S^{W3}_A\subseteq S^{N1}_A$.

3. **Case 3:** Assume $a^f_{ep}>10$.
It follows from Lemma 4.22 that S^{W3}_B and S^{W3}_A are equal to the sets defined in (4.28) and (4.29), respectively.

(A) To prove that $S^{N1}_B\subseteq S^{W3}_B$.
Assume $s_i\in S^{N1}_B$. Refer to (4.22), we have ($a^i_{ef}=F$) and $(a^f_{ep}-a^i_{ep})>0$. Since $a^f_{ep}>10$, a^i_{ep} can be any value within $[0, P]$. Then, let us consider the following cases:
— Suppose $a^i_{ep}\leq 2$. Since $a^i_{ef}=F$, we have

$$(a^i_{ef}-F)+(0.001a^f_{ep}-a^i_{ep})+2.79=0.001(a^f_{ep}-a^i_{ep})+(2.79-0.999a^i_{ep})>0$$

because $(a^f_{ep}-a^i_{ep})>0$ and $(2.79-0.999a^i_{ep})>0$ after $a^i_{ep}\leq 2$. After (4.28) that defines S^{W3}_B, we have $s_i\in S^{W3}_B$.
— Suppose $2<a^i_{ep}\leq 10$. Since $a^i_{ef}=F$, we have

$$(a^i_{ef}-F)+(0.001a^f_{ep}-0.1a^i_{ep})+0.99=0.001(a^f_{ep}-a^i_{ep})+(0.99-0.099a^i_{ep})>0$$

because $(a^f_{ep}-a^i_{ep})>0$ and $(0.99-0.099a^i_{ep})\geq 0$ after $2<a^i_{ep}\leq 10$. After (4.28), $s_i\in S^{W3}_B$.
— Suppose $a^i_{ep}>10$. Since $a^i_{ef}=F$, we have

$$(a^i_{ef}-F)+0.001(a^f_{ep}-a^i_{ep})=0.001(a^f_{ep}-a^i_{ep})>0$$

because $(a^f_{ep}-a^i_{ep})>0$. Thus, $s_i\in S^{W3}_B$ after (4.28).

In summary, we have proved $S^{N1}_B\subseteq S^{W3}_B$.

(B) To prove that $S^{W3}_A\subseteq S^{N1}_A$.
Assume $s_i\in S^{W3}_A$. Refer to (4.29) that defines S^{W3}_A, we have either $(a^i_{ep}\leq 2$ and $(a^i_{ef}-F)+(0.001a^f_{ep}-a^i_{ep})+2.79<0)$, $(2<a^i_{ep}\leq 10$ and $(a^i_{ef}-F)+(0.001a^f_{ep}-0.1a^i_{ep})+0.99<0)$, or $(a^i_{ep}>10$ and $(a^i_{ef}-F)+0.001(a^f_{ep}-a^i_{ep})<0)$.
— Consider the case that ($a^i_{ep}\leq 2$ and $(a^i_{ef}-F)+(0.001a^f_{ep}-a^i_{ep})+2.79<0$). Assume further $a^i_{ef}=F$. Then, we have

$$(a^i_{ef}-F)+(0.001a^f_{ep}-a^i_{ep})+2.79 = 0.001a^f_{ep}-a^i_{ep}+2.79 < 0$$

However, it follows from $a^f_{ep}>10$ and $a^i_{ep}\leq 2$ that $0.001a^f_{ep}-a^i_{ep}+2.79>0.8$, which is contradictory to $0.001a^f_{ep}-a^i_{ep}+2.79<0$. Therefore, it is impossible to have $a^i_{ef}=F$ and all statements in this case have $a^i_{ef}<F$. Then, we have $s_i\in S^{N1}_A$ after (4.23).

— Consider the case that $(2<a^i_{ep}\leq 10$ and $(a^i_{ef}-F)+(0.001a^f_{ep}-0.1a^i_{ep})+0.99<0)$. Assume further $a^i_{ef}=F$. Then, we have

$$(a^i_{ef}-F)+(0.001a^f_{ep}-0.1a^i_{ep})+0.99 = 0.001a^f_{ep}-0.1a^i_{ep}+0.99 < 0$$

However, it follows from $a^f_{ep}>10$ and $2<a^i_{ep}\leq 10$ that $0.001a^f_{ep}-0.1a^i_{ep}+0.99>0$, which is contradictory to $0.001a^f_{ep}-0.1a^i_{ep}+0.99<0$. Therefore, it is impossible to have $a^i_{ef}=F$ and all statements in this case have $a^i_{ef}<F$. Then, we have $s_i\in S^{N1}_A$ after (4.23).

— Consider the case that $(a^i_{ep}>10$ and $(a^i_{ef}-F)+0.001(a^f_{ep}-a^i_{ep})<0)$. Consider the sub-case that $a^i_{ef}<F$. Then, $s_i\in S^{N1}_A$ after (4.23). Now consider the sub-case that $a^i_{ef}=F$. Then, we have $(a^i_{ef}-F)+0.001(a^f_{ep}-a^i_{ep})=0.001(a^f_{ep}-a^i_{ep})$. Since $(a^i_{ef}-F)+0.001(a^f_{ep}-a^i_{ep})<0$, then $(a^f_{ep}-a^i_{ep})<0$. Therefore, $s_i\in S^{N1}_A$ after (4.23).

In summary, we have proved $S^{W3}_A\subseteq S^{N1}_A$.

In conclusion, for any value of a^f_{ep}, we have $S^{N1}_B\subseteq S^{W3}_B$ and $S^{W3}_A\subseteq S^{N1}_A$. It follows from Theorem 3.6 that Naish1 $\rightarrow$ Wong3. Therefore, we have ER1 $\rightarrow$ Wong3 after Proposition 4.3. □

Following from Proposition 4.16, Proposition 4.17, Proposition 4.18, Proposition 4.19, Proposition 4.20, Proposition 4.21 and Proposition 4.23, Figure 4 is formally established. Now, we are going to prove that all the "*better*" ("$\rightarrow$") relations in Figure 4 are strictly "*better*", by showing that there are scenarios where Kulczynski2, M2, ER6, Arithmetic Mean, Cohen, Fleiss or Wong3 are worse (that is, not "*better*") than ER1.

In the proofs of the following propositions, we will construct the relevant scenarios by referring to the two sample programs PG_1 and PG_2 shown in Figure 5 and Figure 6, respectively. In these two programs, s_5 is the faulty statement. Table II lists the A_i for PG_1 with respect to three test suites (TS_1, TS_2 and TS_3) and Table III gives the A_i for PG_2 with respect to test suite TS_3.

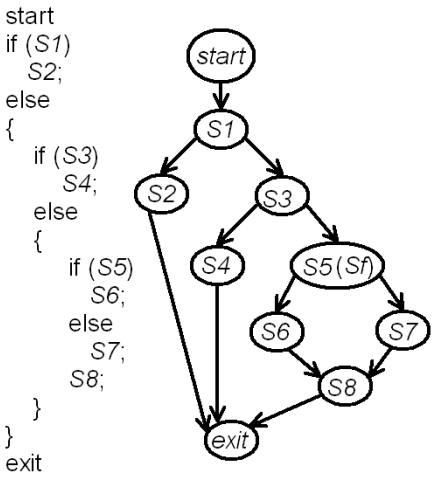

Fig. 5. Sample program PG_1

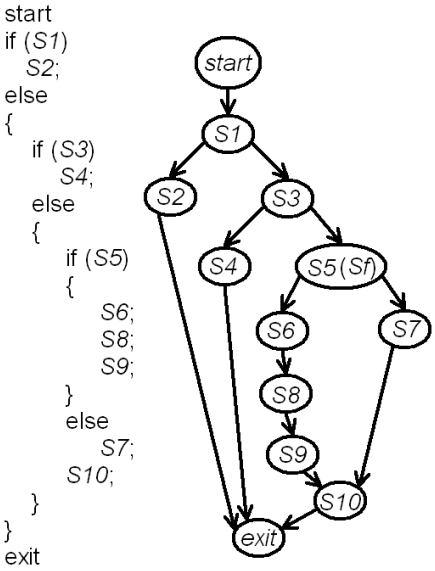

Fig. 6. Sample program PG_2

Table II. A_i for PG_1 with different test suites

Statement	A_i=<$a^i_{ef}, a^i_{ep}, a^i_{nf}, a^i_{np}$>		
	TS_1	TS_2	TS_3
s_1	<40, 160, 0, 0>	<1, 10, 0, 0>	<40, 160, 0, 0>
s_2	<0, 40, 40, 120>	<0, 1, 1, 9>	<0, 70, 40, 90>
s_3	<40, 120, 0, 40>	<1, 9, 0, 1>	<40, 90, 0, 70>
s_4	<0, 40, 40, 120>	<0, 7, 1, 3>	<0, 30, 40, 130>
s_5	<40, 80, 0, 80>	<1, 2, 0, 8>	<40, 60, 0, 100>
s_6	<35, 20, 5, 140>	<1, 1, 0, 9>	<40, 30, 0, 130>
s_7	<5, 60, 35, 100>	<0, 1, 1, 9>	<0, 30, 40, 130>
s_8	<40, 80, 0, 80>	<1, 2, 0, 8>	<40, 60, 0, 100>

Table III. A_i for PG_2 with TS_3

Statement	A_i=<$a^i_{ef}, a^i_{ep}, a^i_{nf}, a^i_{np}$>
s_1	<40, 160, 0, 0>
s_2	<0, 70, 40, 90>
s_3	<40, 90, 0, 70>
s_4	<0, 30, 40, 130>
s_5	<40, 60, 0, 100>
s_6	<40, 30, 0, 130>
s_7	<0, 30, 40, 130>
s_8	<40, 30, 0, 130>
s_9	<40, 30, 0, 130>
s_{10}	<40, 60, 0, 100>

As a reminder, all these test suites are feasible. First, they comply with Lemma 4.1 and Lemma 4.2. Secondly, the entry statement s_1 has (a^1_{nf}=0) and (a^1_{np}=0). Thirdly, for any s_i in PG_1 and PG_2, the value of element a^i_{ef} or a^i_{ep} 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.

Table IV lists the statement divisions for different formulas with respect to different test suites and programs. Row headings from the second row to the seventh row list six scenarios that are referred to in the following proofs; column headings from the second column to the fifth column list four combinations of program and test suite used in the following proofs. For any formula appearing in a specific entry of Table IV, if it is applied on the relevant combination of program and test suite (column), its sets

Table IV. Sets for different combinations of formula and test suite

Scenarios			TS_1 on PG_1	TS_2 on PG_1	TS_3 on PG_1	TS_3 on PG_2
A	S_B^R	$\emptyset$	ER1			
	S_F^R	$\{s_5, s_8\}$				
	S_A^R	$\{s_1, s_2, s_3, s_4, s_6, s_7\}$				
B	S_B^R	$\{s_6\}$	Kulczynski2, M2,	ER1	ER1	
	S_F^R	$\{s_5, s_8\}$	ER6, Cohen, Fleiss,			
	S_A^R	$\{s_1, s_2, s_3, s_4, s_7\}$	Arithmetic Mean			
C	S_B^R	$\{s_6\}$		Wong3		
	S_F^R	$\{s_2, s_5, s_7, s_8\}$				
	S_A^R	$\{s_1, s_3, s_4\}$				
D	S_B^R	$\emptyset$			ER5	
	S_F^R	$\{s_1, s_3, s_5, s_6, s_8\}$				
	S_A^R	$\{s_2, s_4, s_7\}$				
E	S_B^R	$\{s_6, s_8, s_9\}$				ER1
	S_F^R	$\{s_5, s_{10}\}$				
	S_A^R	$\{s_1, s_2, s_3, s_4, s_7\}$				
F	S_B^R	$\emptyset$				ER5
	S_F^R	$\{s_1, s_3, s_5, s_6, s_8, s_9, s_{10}\}$				
	S_A^R	$\{s_2, s_4, s_7\}$				

Note: A_i for PG_1 with TS_1 to TS_3 are shown in Table II; A_i for PG_2 with TS_3 are shown in Table III.

would be the same as the relevant scenario (row). For example, ER1 in the column of "TS_1 on PG_1" and the row of "A" means that by using test suite TS_1 on program PG_1, the S_B^R, S_F^R and S_A^R for ER1 are as scenario A which has S_B^R=$\emptyset$, S_F^R=$\{s_5, s_8\}$ and S_A^R=$\{s_1, s_2, s_3, s_4, s_6, s_7\}$.

PROPOSITION 4.24. *ER1 is strictly "*better*" than formulas Kulczynski2, M2, ER6, Arithmetic Mean, Cohen, Fleiss and Wong3.*

PROOF. As listed in Table IV, for the column of "TS_1 on PG_1", the statement divisions for ER1 are as scenario A; while for formulas Kulczynski2, M2, ER6, Arithmetic Mean, Cohen and Fleiss, the corresponding S_B^R, S_F^R and S_A^R are as scenario B. As seen in Table IV, the S_F^R for Kulczynski2, M2, ER6, Arithmetic Mean, Cohen and Fleiss are the same as the S_F^R for ER1; but the sizes of S_B^R (=1) for these six formulas are larger than the corresponding size (=0) for ER1. If we adopt a consistent tie-breaking scheme, such as the *ORIGINAL ORDER* scheme that ranks all statements in S_F^R according to

their original order in program, the faulty statement which is s_5 would be ranked the highest in ER1 and the second highest in the other six formulas. Thus, the *EXAM* score of ER1 is less than the *EXAM* scores of the six formulas. It follows immediately from Proposition 4.16, Proposition 4.17, Proposition 4.18, Proposition 4.19, Proposition 4.20 and Proposition 4.21 that ER1 is strictly "*better*" than these six formulas.

Next, for the column of "TS_2 on PG_1" in Table IV, the relevant sets for ER1 are as scenario B; while for Wong3, the relevant sets are as scenario C. For ER1 and Wong3, the sizes of their S_B^R ($=1$) are the same, but their S_F^R are different. If we adopt the *ORIGINAL ORDER* tie-breaking scheme, the faulty statement (s_5) would be ranked the second highest in ER1 and the third highest in Wong3. Therefore, the *EXAM* score of ER1 is less than the *EXAM* score of Wong3. It follows from Proposition 4.23 that ER1 is strictly "*better*" than Wong3. □

Immediately after all the propositions between Proposition 4.6 and Proposition 4.24, Ochiai, ER2, ER3, ER4, AMPLE2, Kulczynski2, M2, ER6, Arithmetic Mean, Cohen, Fleiss and Wong3 cannot be maximal. As a consequence, the only remaining candidates for maximal formulas are ER1 and ER5, which are in fact the maximal formulas as stated in the following proposition.

PROPOSITION 4.25. *ER1 and ER5 are the maximal formulas.*

PROOF. First, we will prove that ER5 $\rightarrow$ ER1 does not hold. Refer to the column of "TS_3 on PG_1" in Table IV, the relevant sets for ER1 are as scenario B; while for ER5, the relevant sets are as scenario D. Though the size of S_B^R ($=1$) for ER1 is larger than the corresponding size ($=0$) for ER5, the S_F^R for these two equivalent groups are different. If we adopt the *ORIGINAL ORDER* tie-breaking scheme, the faulty statement (s_5) would be ranked higher in ER1 (the second highest) than in ER5 (the third highest), that is, the *EXAM* score of ER1 is less than the *EXAM* score of ER5. Thus, ER5 is not "*better*" ("$\rightarrow$") than ER1.

Secondly, we will prove that ER1 $\rightarrow$ ER5 does not hold. Refer to the column of "TS_3 on PG_2" in Table IV, the relevant sets for ER1 are as scenario E; while for ER5, they are as scenario F. As seen from the table, the size of S_B^R ($=3$) for ER1 is larger than the corresponding size ($=0$) for ER5. If we adopt the *ORIGINAL ORDER* tie-breaking scheme, the faulty statement (s_5) would be ranked the fourth highest in ER1 and the third highest in ER5. In other words, the *EXAM* score of ER1 is greater than the *EXAM* score of ER5. Thus, ER1 is not "*better*" ("$\rightarrow$") than ER5.

The above examples demonstrate that neither ER1 $\rightarrow$ ER5 nor ER5 $\rightarrow$ ER1 holds. Following after all the propositions between Proposition 4.3 and Proposition 4.24, we can conclude that ER1 and ER5 are the only maximal formulas among all the 30 investigated formulas. □

5. RELATED WORK

The performance of various risk evaluation formulas in SBFL has been compared through empirical studies and theoretical analyses.

Abreu et al. conducted empirical performance comparison among different risk evaluation formulas [Abreu et al. 2006; 2007; Abreu et al. 2009]. They first introduced formula Ochiai from the discipline of molecular biology into SBFL. Their experimental results showed that Ochiai outperformed Jaccard (ER2), and Jaccard (ER2) outperformed Tarantula (ER3). Their observations are consistent with our theoretical results as reported in Figure 3(a). In order to find out the reason why such observations occur, they analyzed how different elements in vector A_i would affect the returned values of these formulas. However, their analysis was unable to reveal the underlying rationale. They also discovered that the original version of AMPLE2 performed the worst in most

cases. This observation is understandable because the original version is against the intuition of SBFL formulas, as discussed in Section 4.1. Actually, their results were also validated by other empirical studies. For example, Santelices et al. [2009] have reported that Ochiai outperformed Tarantula in their earlier experiments. But they neither described the performance relations between other formulas as Abreu et al. did, nor gave an analysis for their observation.

In [Lee et al. 2009b] and [Naish et al. 2011], apart from the theoretical analyses, comprehensive experimental studies were also included to compare the average *EXAM* scores of some formulas, using the widely adopted Siemens Suite and Space as benchmarks. Their experimental results for executable codes and single-fault scenario are consistent with our conclusions. First, formulas in each of the six equivalent groups (ER1 to ER6) gave identical average *EXAM* scores in their experiments. Secondly, their experimental results are consistent with our results depicted in Figures 3 and 4.

Despite such a comprehensive experimental study, Naish et. al were still unable to reveal the underlying rationale for all of their observations. Moreover, their analysis by its nature can never deliver a full picture. For example, in their experiments, ER1 was observed to outperform all the other investigated formulas, including ER5. However, their experimental results did not demonstrate that there is no "*better*" ("$\rightarrow$") relation between these two groups of formulas, as stated in Proposition 4.25. A similar problem existed in Wong et al.'s study, in which they have reported that by using the same tie-breaking scheme of either *BEST* or *WORST*, Wong3 outperformed Tarantula [Wong et al. 2007]. Actually, our framework can show that neither Wong3 is "*better*" ("$\rightarrow$") than Tarantula, nor Tarantula is "*better*" ("$\rightarrow$") than Wong3. However, their experimental results are still consistent with our theoretical results.

Nevertheless, there exist some empirical results showing difference between Tarantula and CBI [Yu et al. 2008; Jiang et al. 2009] of the same equivalent group (ER3) in our analysis. Such discrepancy is due to the use of different tie-breaking schemes on Tarantula and CBI. In the study by Yu et al. [2008], there were two tie-breaking schemes applied together on Tarantula. The first one was an additional metric, namely *confidence*, computed as follows:

$$confidence{=}max(\frac{a_{ef}}{a_{ef}+a_{nf}}, \frac{a_{ep}}{a_{ep}+a_{np}})$$

The second one was effectively the "*WORST*" tie-breaking scheme in terms of the *EXAM* score. For statements with the same risk value and the same *confidence* value, the sum of the number of these tied statement and the number of statements ranked before them were assigned as the ranking of these statements. While for CBI, only the "*WORST*" tie-breaking scheme was applied. Similarly, in the study by Jiang et al. [2009], Tarantula adopted both the *confidence* and the "*BEST*" tie-breaking schemes; while CBI only adopted the "*BEST*" tie-breaking scheme. In other words, their results and our results are not comparable because of the different context.

Due to the limitation of empirical study, some theoretical studies were conducted for the performance comparison. Lee et al. [2009a] have proved that formulas Tarantula and q_e yield identical ranking lists. In a follow-up study, Naish et al. [2011] conducted a more comprehensive investigation, where more equivalence relations were identified, using the same definition of equivalence as Lee et al. [2009a]. However, their 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, but using a hybrid approach, with a model program and a group of multisets of execution paths. However, their performance measurement was not commonly adopted by the SBFL community, and their analysis still involved sampling and simulation.

It should be noted that both the studies mentioned above and our theoretical analysis are focused on risk evaluation formulas that use the same information, namely A_i. Recently, some extended SBFL techniques have emerged, which integrate the basic procedure with other models or employ additional information, 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 the basic version of SBFL should also 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, because of its empirically good performance. Naish et al. [2009] have applied their weighting factors in different formulas, but interestingly, the performance comparison results of these formulas with weighting factors are consistent with the results without weighting factors. Therefore, our framework that provides a definite solution to the choice of the risk evaluation formula for SBFL, can help in both the basic version of SBFL and its extended versions.

6. DISCUSSION

As stated in Section 4.2, our framework is based on several assumptions, which are discussed in details in this section.

We have assumed that the programs under debugging have testing oracle and the faults are deterministic ones. This is reasonable because only with these assumptions, can deterministic testing results of *pass* or *fail* be obtained, which are required for risk evaluation.

We have excluded the omission faults, because SBFL is designed to assign risk values to the existent statements. 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 controversial. 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". And there is no consensus on the interpretation. Thus in order to avoid unnecessary noises, we leave the omission faults out of this study.

Furthermore, we have imposed a constraint on the test suite, namely, 100% statement coverage. It is true that this constraint is not easily satisfied in practice. However, whether a test suite actually achieves 100% statement coverage does not really affect the applicability of our framework. The rationale is that if a statement is never covered by any test case in the given test suite, it cannot be the faulty statement that triggers the observed failures. Therefore, in practice, if a test suite does not achieve 100% statement coverage, we should first exclude those uncovered statements and only focus on the covered portion of program for the debugging purpose, in which the 100% statement coverage is then satisfied, and thus our framework is applicable. For those uncovered statements, it is reasonable to rank them at the bottom of the ranking list [Xie et al. 2010]. As a matter of fact, in many previous empirical studies, the Siemens Suite and Space were adopted as benchmarks. These programs have well developed test suites, in which at least 30 test cases can exercise each executable statement or branch in the program [SIR 2005]. Therefore, these empirical studies did not encounter such a 100% coverage problem.

Besides, we have the assumption that the test suite contains at least one *failed* test case and one *passed* test case. Such an assumption is reasonable and practically

feasible. First, it is widely accepted that at least one *failed* test case is required for debugging. Secondly, except ER5, all the other investigated formulas have the prerequisite that there exists at least one *passed* test case, though this prerequisite was never explicitly stated. Without any *passed* test cases, such formulas either become totally undefined (such as Tarantula), or partially undefined (such as Scott), or become equivalent to ER5 (such as Naish1), or even become unable to function properly (such as q_e where all statements are assigned with the same risk value). As compared with the *failed* test cases that may be difficult to get, the *passed* test cases are much easier to find and normally exist. Even if the current test suite does not contain any *passed* test case, after a few rounds of regression testing, some *failed* test cases would have become *passed* ones. In other words, this assumption is realistic and is relatively easy to be achieved in practice.

In this study, we assume that debuggers examine the ranking list from the top to the bottom, because we use the ranking of faulty statement as the measurement to determine the performance of the risk evaluation formulas in SBFL. Without any assumption about the order that the debuggers would inspect the ranking list, comparing the ranking of the faulty statement between different formulas will become meaningless. Furthermore, we have the assumption of "*perfect bug detection*". This study focuses on the performance of the risk evaluation formulas, rather than the performance of an entire debugging process. And the effectiveness of bug detection usually involves many complicated issues, such as the complexity of the fault, the debugger's experience, etc. Thus, without any further information, the only available and also reasonable assumption is "*perfect bug detection*". Therefore, though this assumption is not normally satisfied in practice, it has actually been accepted by the SBFL community, because it provides a fair comparison among different SBFL techniques [Parnin and Orso 2011]. However, it should be noted that as an automatic debugging technique, SBFL only provides assistant information to the debuggers and its ranking list is not the sole determinant of the effectiveness in the entire debugging process. To what extent can debuggers benefit from the ranking list depends on many factors, including the effectiveness of the adopted SBFL technique, the complexity of the fault, the debugger's experience, the pattern that the debugger navigates the given ranking list, etc. as observed by Parnin and Orso [2011]. For example, they have found that expert debuggers received a much higher benefit from using Tarantula, and that debuggers do not necessarily examine each statement one by one, according to the given ranking list. However, these factors are out of the scope of this study.

In addition to the assumptions discussed above, our study has assumed the single-fault scenario. Even though we have only provided a theoretical analysis for single-fault scenario, we believe that our conclusions are still meaningful and useful in multiple-fault cases. Generally speaking, in practice where the quantity of the faults is unknown, there are two debugging approaches, namely "*sequential debugging*" and "*parallel debugging*". The sequential debugging approach iteratively fixes a localized fault and then conducts regression testing to localize the next fault [Jones et al. 2007]; while the parallel debugging approach tries to locate faults simultaneously, using the technique of "fault-focusing clustering" [Dickinson et al. 2001; Podgurski et al. 2003; Liu and Han 2006; Zheng et al. 2006; Jones et al. 2007].

DiGiuseppe and Jones [2011] have recently studied the impact of multiple-fault on SBFL and found that in terms of localizing at least one single fault, SBFL techniques remain to be effective as the quantity of the faults increases, even in the presence of fault-localization interference (that is how different faults interfere with each others' localizability). Thus, when adopting the sequential debugging approach, the performance of the SBFL techniques is not affected significantly by the quantity of faults.

Therefore, regardless of the number of faults, it is still intuitively attractive to choose risk evaluation formulas with better performance under the single-fault scenario.

On the other hand, when adopting the parallel debugging approach, test cases are first clustered into several specialized test suites based on various execution information, and each of the test suites targets an individual fault. As such, the fault-localization interference among multiple faults can be reduced [Liu and Han 2006; Zheng et al. 2006; Jones et al. 2007]. In practice, each specialized test suite is dispatched to a particular debugger, who is supposed to focus on the corresponding single fault. Therefore, before applying our theoretical framework to the parallel debugging approach, we just need to conduct an initial clustering process on the test suite. After such an initial pre-processing, a debugging task for multiple-fault is divided into several parallel sub-tasks, each for a single fault and a debugger. As a consequence, for each debugger, his/her sub-task can be considered as a single-fault scenario where the results of this paper can be applied. Since most clustering techniques use heuristic methods, they cannot guarantee to eliminate all fault-localization interference. Thus, it is worthwhile to further study how the noises brought by the fault-focusing clustering techniques would affect our theoretical framework.

7. CONCLUSIONS

With the emergence of more and more 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 many factors, such as object programs, test suites, types of fault, etc. 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. On the other hand, a theoretical approach has no such limitations, and hence more reliable and robust conclusions can be obtained.

In this paper, we have developed an innovative framework for theoretical analysis, using the notion of subset. Different from previous theoretical studies, we propose two types of relation, namely, the "*equivalent*" ("$\leftrightarrow$") relation and the "*better*" ("$\rightarrow$") relation. Our definition of "*equivalent*" relation that requires same ranking of s_f, is more intuitively appealing and general than that of Naish et al. requiring identical ranking list. Though identical ranking lists guarantee the same rankings for faulty statements regardless of the number of faulty statements, it may treat some formulas as non-equivalent even if they always yield the same rankings for the faulty statements. Thus, their definition of equivalence does not properly reflect a more realistic scenario.

Our framework identifies relations between different formulas based on a simple intuition that the number of statements with risk values higher than the risk value 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 the risk value of the faulty statement, and compares the sizes of these sets for different formulas using the notion of subset.

We apply our framework to the formulas investigated by Naish et al. [2011], but exclude some formulas with the justifications given in Section 4.1. Among the 30 investigated formulas in our study, we have proved that for single-fault scenario, there are five maximal formulas, namely, Naish1, Naish2, Wong1, Russell & Rao and Binary, which are grouped into two equivalent groups, ER1 and ER5. In other words, when we apply SBFL, we only need to consider risk evaluation formulas from these two maximal groups. By inspecting these two maximal groups, the intuition that a statement executed by more *failed* test cases has higher possibility to be faulty, is observed to have the most significant impact on the effectiveness of a risk evaluation

formula. These two groups of formulas use a_{ef} as either their sole (ER5) or primary (ER1) determinant in ranking the risk values. Apart from a_{ef}, ER1 also utilizes a_{ep} as its secondary determinant, which actually follows another intuition that a statement executed by less *passed* test cases has higher possibility to be faulty. Though such an additional information does not help to guarantee that ER1 always outperforms ER5, Naish et al. [2011] have observed from the experimental data that ER1 performed better than ER5 on average.

Overall, our experience of this study shows that the theoretical analysis and the empirical analysis are both essential and complementary to each other in software analysis and testing. The previously extensive empirical investigations have provided great amount of experimental data to show that some risk evaluation formulas appear to perform better than others. These experimental data have convinced us that it is not a coincidence but the existence of a definite relation between some pairs of risk evaluation formulas, and hence have motivated us to adopt a theoretical approach to search for the underlying rationale. This has led to the discovery of the maximal formulas in this study. Generally speaking, an empirical study is useful to expose or highlight some interesting phenomena or trends, which may conjecture a generalization that needs to be verified by a theoretical analysis. On the other hand, a theoretical study normally aims at finding a definite answer to a question. Even if a problem cannot be completely solved by a theoretical approach, problems may be highlighted during the analysis, which are worthwhile and critical to be attempted by an empirical approach. Actually, one interesting problem that has been highlighted by this study is to compare the performance of the two maximal groups of equivalent formulas using an empirical approach.

ACKNOWLEDGMENTS

This project is partially supported by an Australian Research Council Discovery Project (DP120104773) and the National Natural Science Foundation of China (90818027, 61170071).

REFERENCES

ABREU, R., ZOETEWEIJ, P., GOLSTEIJN, R., AND VAN GEMUND, A. J. C. 2009. A practical evaluation of spectrum-based fault localization. *Journal of Systems and Software 82,* 11, 1780–1792.

ABREU, R., ZOETEWEIJ, P., AND VAN GEMUND, A. J. C. 2006. An evaluation of similarity coefficients for software fault localization. In *Proceedings of the 12th Pacific Rim International Symposium on Dependable Computing*. Riverside, USA, 39–46.

ABREU, R., ZOETEWEIJ, P., AND VAN GEMUND, A. J. C. 2007. On the accuracy of spectrum-based fault localization. In *Proceedings of Testing: Academic and Industrial Conference Practice and Research Techniques-MUTATION*. Windsor, UK, 89–98.

AGRAWAL, H., HORGAN, J. R., LONDON, S., AND WONG, W. E. 1995. Fault localization using execution slices and dataflow tests. In *Proceedings of the 6th International Symposium on Software Reliability Engineering*. Toulouse, France, 143–151.

BAAH, G. K., PODGURSKI, A., AND HARROLD, M. J. 2010. Causal inference for statistical fault localization. In *Proceedings of the International Symposium on Software Testing and Analysis*. Trento, Italy, 73–84.

BANDYOPADHYAY, A. AND GHOSH, S. 2011. Proximity based weighting of test cases to improve spectrum based fault localization. In *Proceedings of the 26th IEEE/ACM International Conference on Automated Software Engineering*. Lawrence, USA, 420–423.

CHEN, M., KICIMAN, E., FRATKIN, E., FOX, A., AND BREWER, E. 2002. Pinpoint: problem determination in large, dynamic internet services. In *Proceedings of the 32th IEEE/IFIP International Conference on Dependable Systems and Networks*. Washington DC, USA, 595–604.

COLLOFELLO, J. S. AND WOODFIELD, S. N. 1989. Evaluating the effectiveness of reliability-assurance techniques. *Journal of Systems and Software 9,* 3, 191–195.

DALLMEIER, V., LINDIG, C., AND ZELLER, A. 2005. Lightweight defect localization for java. In *Proceedings of the 19th European Conference on Object-Oriented Programming*. Scotland, UK, 528–550.

DICKINSON, W., LEON, D., AND PODGURSKI, A. 2001. Finding failures by cluster analysis of execution profiles. In *Proceedings of the 23rd International Conference on Software Engineering*. Toronto, Ontario, Canada, 339–348.

DIGIUSEPPE, N. AND JONES, J. A. 2011. On the influence of multiple faults on coverage-based fault localization. In *Proceedings of the International Symposium on Software Testing and Analysis*. Toronto, Canada, 199–209.

HARROLD, M. J., ROTHERMEL, G., SAYRE, K., WU, R., AND YI, L. 2000. An empirical investigation of the relationship between spectra differences and regression faults. *Software Testing Verification and Reliability 10,* 3, 171–194.

HARROLD, M. J., ROTHERMEL, G., WU, R., AND YI, L. 1998. An empirical investigation of program spectra. In *Proceedings of the 1st ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering*. Montreal, Canada, 83–90.

JIANG, B., ZHANG, Z., TSE, T. H., AND CHEN, T. Y. 2009. How well do test case prioritization techniques support statistical fault localization. In *Proceedings of the 33rd Annual International Conference on Computer Software and Applications*. Vol. 1. Seattle, USA, 99–106.

JONES, J. A., BOWRING, J. F., AND HARROLD, M. J. 2007. Debugging in parallel. In *Proceedings of the International Symposium on Software Testing and Analysis*. New York, USA, 16–26.

JONES, J. A. AND HARROLD, M. J. 2005. Empirical evaluation of the tarantula automatic fault-localization technique. In *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering*. Long Beach, USA, 273–282.

JONES, J. A., HARROLD, M. J., AND STASKO, J. 2002. Visualization of test information to assist fault localization. In *Proceedings of the 24th International Conference on Software Engineering*. Florida, USA, 467–477.

LEE, H. J., NAISH, L., AND RAMAMOHANARAO, K. 2009a. Study of the relationship of bug consistency with respect to performance of spectra metrics. In *Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology*. Beijing, China, 501–508.

LEE, H. J., NAISH, L., AND RAMAMOHANARAO, K. 2009b. The effectiveness of using non redundant test cases with program spectra for bug localization. In *Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology*. Beijing, China, 127–134.

LIBLIT, B. R. 2004. Cooperative bug isolation. Ph.D. thesis, University of California, USA.

LIBLIT, B. R., NAIK, M., ZHENG, A. X., AIKEN, A., AND JORDAN, M. I. 2005. Scalable statistical bug isolation. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*. Chicago, USA, 15–26.

LIU, C., FEI, L., YAN, X., HAN, J., AND MIDKIFF, S. 2006. Statistical debugging: a hypothesis testing-based approach. *IEEE Transactions on Software Engineering 32,* 10, 831–848.

LIU, C. AND HAN, J. 2006. Failure proximity: a fault localization-based approach. In *Proceedings of the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. New York, USA, 46–56.

NAISH, L., LEE, H. J., AND RAMAMOHANARAO, K. 2009. Spectral debugging with weights and incremental ranking. In *Proceedings of the 16th Asia-Pacific Software Engineering Conference*. Penang, Malaysia, 168–175.

NAISH, L., LEE, H. J., AND RAMAMOHANARAO, K. 2011. A model for spectra-based software diagnosis. *ACM Transactions on Software Engineering and Methodology 20,* 3, 11:1–11:32.

PARNIN, C. AND ORSO, A. 2011. Are automated debugging techniques actually helping programmers? In *Proceedings of the International Symposium on Software Testing and Analysis*. Toronto, Canada, 199–209.

PODGURSKI, A., LEON, D., FRANCIS, P., MASRI, W., MINCH, M., SUN, J., AND WANG, B. 2003. Automated support for classifying software failure reports. In *Proceedings of the 25th International Conference on Software Engineering*. Portland, Oregon, USA, 465–475.

REPS, T., BALL, T., DAS, M., AND LARUS, J. 1997. The use of program profiling for software maintenance with applications to the year 2000 problem. In *Proceedings of the 6th European Software Engineering Conference held jointly with the 5th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. Number 6. Zurich, Switzerland, 432–449.

SANTELICES, R., JONES, J. A., YU, Y., AND HARROLD, M. J. 2009. Lightweight fault-localization using multiple coverage types. In *Proceedings of the 31st International Conference on Software Engineering*. Vancouver, Canada, 56–66.

SIR. 2005. http://sir.unl.edu/php/index.php.

WONG, W. E., DEBROY, V., AND CHOI, B. 2010. A family of code coverage-based heuristics for effective fault localization. *Journal of Systems and Software 83,* 2, 188–208.

WONG, W. E. AND QI, Y. 2006. Effective program debugging based on execution slices and inter-block data dependency. *Journal of Systems and Software 79,* 7, 891–903.

WONG, W. E., QI, Y., ZHAO, L., AND CAI, K. Y. 2007. Effective fault localization using code coverage. In *Proceedings of the 31st Annual International Conference on Computer Software and Applications*. Beijing, China, 449–456.

WONG, W. E., WEI, T., QI, Y., AND ZHAO, L. 2008. A crosstab-based statistical method for effective fault localization. In *Proceedings of the 1st International Conference on Software Testing, Verification and Validation*. Lillehammer, Norway, 42–51.

XIE, X. Y. 2012. On the analysis of spectrum-based fault localization. Ph.D. thesis, Swinburne University of Technology, Australia.

XIE, X. Y., CHEN, T. Y., AND XU, B. W. 2010. Isolating suspiciousness from spectrum-based fault localization techniques. In *Proceedings of the 10th International Conference on Quality Software*. Zhangjiajie, China, 385–392.

XIE, X. Y., WONG, W. E., CHEN, T. Y., AND XU, B. W. 2011. Spectrum-based fault localization: testing oracles are no longer mandatory. In *Proceedings of the 11th International Conference on Quality Software*. Madrid, Spain, 1–10.

YU, Y., JONES, J. A., AND HARROLD, M. J. 2008. An empirical study of the effects of test-suite reduction on fault localization. In *Proceedings of the 30th International Conference on Software Engineering*. Leipzig, Germany, 201–210.

ZELLER, A. 2002. Isolating cause-effect chains from computer programs. In Proceedings of the 10th ACM SIGSOFT Symposium on Foundations of Software Engineering. *ACM SIGSOFT Software Engineering Notes*, 1–10.

ZHENG, A. X., JORDAN, M. I., LIBLIT, B., NAIK, M., AND AIKEN, A. 2006. Statistical debugging: simultaneous identification of multiple bugs. In *Proceedings of the 23rd International Conference on Machine Learning*. New York, USA, 1105–1112.