

Finding the Efficiency of Query by Using Predictive Energy Saving Online Scheduling Algorithm

Chhinabala Vinod Kumar¹, P. Prasad Babu²

¹Student, Department of MCA, RCR Institutes of Management & Technology, Tirupathi, Andhra Pradesh, India

²Assistant Professor, Department of MCA, RCR Institutes of Management & Technology, Tirupathi, Andhra Pradesh, India

ABSTRACT

Web search engines are composed by thousands of query processing nodes, i.e., servers dedicated to process user queries. Metamorphic testing may be a testing technique which will be used to verify the useful correctness of software system within the absence of an ideal oracle. This paper extends metamorphic testing into a user-oriented approach to software system verification, validation, and quality assessment, and conducts large scale empirical studies with four major net search engines: Google, Bing, Chinese Bing, and Baidu. These search engines are very tough to check and assess using conventional approaches owing to the lack of an objective and generally recognized oracle. The results are useful for each search engine developers and users, and demonstrate that our approach will effectively alleviate the oracle drawback and challenges close a lack of specifications when verifying, validating, and evaluating giant and complex software systems.

Keywords: - Metamorphic testing, Google, Bing, Chinese Bing.

Introduction:-

The objective of programming designing practices is to grow top notch programming. It is in this manner urgent to create assessment strategies for different sorts of programming characteristics. Testing is a broadly utilized approach for assessing programming characteristics and helping engineers to discover and expel programming deficiencies. The greater parts of programming testing strategies accept the accessibility of a prophet, an instrument against which analyzers can confirm the rightness of the results of experiment executions. In a few circumstances, be that as it may, a prophet isn't accessible or is accessible yet is too costly

to ever be utilized – a circumstance known as the prophet issue, a central test for programming testing.

A transformative testing (MT) strategy has been produced to reduce the prophet issue. Not at all like regular testing strategies, MT does not center on the check of every individual yield, yet checks the connections among the sources of info and yields of numerous executions of the program under test. Such connections are known as transformative relations (MRs), furthermore, are vital properties of the planned program's usefulness: If a MR infringement is identified, at that point blame is said to be uncovered. MT has been utilized to check the practical

accuracy of different applications and has additionally been connected to program demonstrating and investigating Its adequacy has likewise been precisely examined The present research expands transformative testing into a quantifiable approach for programming quality assessment, which incorporates, however isn't constrained to, the check furthermore, approval of programming rightness. We connected our way to deal with ease the prophet issue for the testing furthermore, quality appraisal of (Web) web indexes. Pursuit motors are programming frameworks intended to scan for data on the World Wide Web, and are the fundamental interface through which individuals find data on the Internet; Web looking is a standout amongst the most well known functionalities of the Internet, second just to email As an ever increasing number of administrations and information are being made accessible on the Internet, web indexes are getting to be progressively critical. In the present profoundly focused seek advertise, it is basic that web crawlers give the coveted outcome as indicated by the inquiries entered. It is, in any case, greatly hard to evaluate some key characteristics of these web crawlers. For example, attributable to the sheer volume of information on the Internet, it is extremely hard to check or then again approve the accuracy of the product frameworks or to assess the exactness and fulfillment of the inquiry comes about. Likewise, given the conspicuous subjectivity of various judges, target evaluation of output importance furthermore, positioning quality is extremely troublesome.

Proposed system:-

METAMORPHIC TESTING:-

Metamorphic Testing (MT) is property bases testing which aims to find some useful relations (called Metamorphic Relations) to alleviate the oracle problems (Chen et al, 2003). As explained by Asrfai et al. (Asrafi et al, 2011), a metamorphic relation (MR) is an expected relation of the program under test which should be valid over a set of distinct input data and their corresponding output for multiple executions. Figure-1 summarizes the relations in MT which involve source and follow-up inputs and outputs.MT checks the validity of MRs by multiply executing of the target program. The steps of MT are as folllowings: (i) determining specific properties of the SUT to construct MRs, (ii) generating source test case by some traditional testing techniques (such as random testing), (iii) generating follow-up test cases based on source test cases according to the MRs, (iv) executing the test cases, and (v) verifying the outputs of the test cases against MRs. If the outputs of the source and follow-up test cases do not match their relations in corresponding MR, then the test fails.

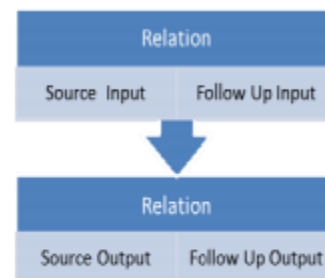


Figure-1. Relation in MT.

Asrafi et al (Asrafi et al, 2011) presented a simple example of MT in a sorting program as follows. The program sorts a set of integers in the ascending order. Suppose S is a set of numbers to be sorted. If the set S is rearranged in reverse order the output of the sorting program will still remain same. This MR can be denoted by $\text{Sort}(S) = \text{Sort}(\text{reverse}(S))$. Suppose $S = \{35, 15, 32, 25\}$, $\text{Sort}(S)$ will yield $\{15, 25, 32, 35\}$. We reverse the set S to generate the follow-up test case $\text{reverse}(S) = \{25, 32, 15, 35\}$. If $\text{Sort}(\text{reverse}(S)) \{15, 25, 32, 35\}$, we can say a fault is detected. MT has been widely used in solving many oracle problems (Barus et al, 2009; Chen et al, 1998; Chen et al, 2009; Chen et al, 2004).

Proposed system:-

To apply MT to the automatic quality assessment of search engines, without the need for an oracle or human assessor, two groups of MRs were used: The “No Missing Web Page” group assesses the search engines’ capability in retrieving appropriate Web pages to meet the users’ needs; and the “Consistent Ranking” group assesses the ranking quality of the search results. This section provides a brief description of these MRs.

Metamorphic Relation: MPSite

MPSite belongs to the “No Missing Web Page” group of MRs, which assess the search engine’s Web page retrieval capability. MPSite is focused on the search engine’s reliability when retrieving Web pages that contain an exact word or phrase. It therefore assesses the keyword based search feature. MPSite is described as follows: Let A be a source query

for which the search engine returns a non-empty list of results (called the source response), namely, (p_1, p_2, \dots, p_n) , where $0 < n$ and p_i is a Web page from domain d_i , $1 \leq i \leq n$. To enhance accuracy and validity of our approach, in MPSite we only consider situations where $0 < n \leq 20$ so that we can avoid the inaccuracy associated with large result sets (such as a large list being truncated by the search engine to improve response time).

For the source response (p_1, p_2, \dots, p_n) , n follow-up queries are constructed as follows: The i th follow-up query B_i ($1 \leq i \leq n$) is constructed in such a way that B_i is identical to A except that B_i includes an additional criterion which requires that all results be retrieved from domain d_i . Let FR_i (a follow-up response) be the list of Web pages returned by the search engine for query B_i . The metamorphic relation MPSite requires that $p_i \in FR_i$ (note that there is no requirement on the ranking of p_i in FR_i). For example, let us test Google by issuing the following source query:



Fig. 2: Excerpts from Google help page.

"side effect of antibiotics in babies" where the quotation marks are part of the query. Google returned a total of 7 Web pages. Without loss

of generality, let us consider the top result, which is: This Web page is from the .uk domain. 1 The metamorphic relation MPsite enables the construction of the following follow-up query: ["side effect of antibiotics in babies" site:uk], 2 where "site:" is a Google search operator that specifies domains (see Figure 2 (lower)). Obviously, the previously returned top result (<http://www.dailymail.co.uk/>. . .) meets this search criterion, is indexed in Google database, and therefore should still be returned by Google for this follow-up query. In this example, Google returned a total of 7 Web pages for the source query. Therefore, 7 followup queries are constructed by referring to MPsite. 3 Using MPsite, even if the assessor is unable to verify or evaluate each individual response, he/she can still verify the logical consistency relationship among multiple responses. Here, the basic approach is to use the search engine's source response to check its follow-up response. Figure 3 shows a failure detected using MPsite, All MRs identified in this paper were implemented into a testing tool and, hence, the testing and assessment process is automated.

Metamorphic Relation: MPTitle:-

For many search engines including those investigated in the present paper, if the words are not enclosed by double quotation marks, synonyms will be employed automatically. For instance, Google specifies that "Google employs synonyms automatically, so that it finds pages that mention, for example, childcare for the query [child care] (with a space), or California history for the query [ca

history]." Synonyms are employed because the search engines attempt to return Web pages that best meet users' information needs. In other words, the search engines attempt to imitate the behavior of a human operator, to which end, correct understanding of the Web pages and of the user intent are key. To test a search engine's information retrieval capability in situations where synonyms may be used for semantic search, a good strategy is to construct a test query q that best characterizes a target Web page p (the words in q may or may not directly appear in p). Furthermore, p must have been indexed in the search engine's database. The search engine can be tested on q . If p is not retrieved, then the user's perception of the search quality will be poor. A research question is: How can such q 's and p 's be identified in a fully automatic fashion for search engine assessment? The metamorphic relation MPTitle is designed to meet this challenge.

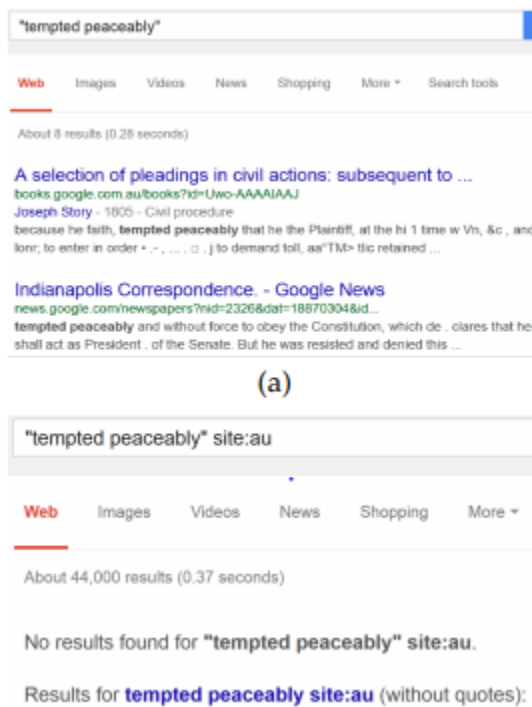


Fig. 3: A Google failure detected using MPSite.

The top result in (a) cannot be retrieved in (b).

Metamorphic Relation: MPReverseJD:-

The third MR of the “No Missing Web Page” group is MPReverseJD. Its design was inspired by a search engine assessment technique informally used in industry, which is based on the rationale that a good search engine should return similar results for similar queries. For instance, although a search for [today’s movies in Redmond] and a search for [Redmond movies today] (without double quotes) may return different results, the two result sets should share a large intersection if the search engine is robust to the nonessential differences between these two queries. 5 This idea was also employed by Imielinski and Signorini to test semantic search engines using

semantically equivalent queries. The MR MPReverseJD is designed as follows: The source query A is defined to be a query for which the search engine returns a non-empty list of up to 20 results. A is further defined to be the conjunction of up to 4 terms, namely:

$$A = A_1 \text{ AND } A_2 \underbrace{\text{AND } A_3}_{\text{optional}} \underbrace{\text{AND } A_4}_{\text{optional}}$$

where Ai (i=1, 2, 3, 4) is a name enclosed by double quotation marks. Terms A3 and A4 are optional: A3 is applied only when the conjunction of A1 and A2 has more than 20 results, and A4 is applied only when the conjunction of A1, A2, and A3 has more than 20 results. If the conjunction of all 4 terms still has more than 20 results, all these terms will be discarded and a new query will be formed. The following is an example of the source query A: ["Vincent Van Gogh" AND "Elvis Presley" AND "Albert Einstein" AND "Plato"]. In this example, A1="Vincent Van Gogh," A2="Elvis Presley," A3="Albert Einstein," and A4="Plato." The follow-up query B is constructed by reversing the order of A’s terms: ["Plato" AND "Albert Einstein" AND "Elvis Presley" AND "Vincent Van Gogh."] MPReverseJD states that a stable search engine should return similar results for the source query A and followup query B.

In other words, the two result sets should have a large intersection – we refer to this kind of quality as stability. This requirement is reasonable especially given that the result set of A is very small (containing no more than 20 results) and that the source and follow-up queries have similar semantic meanings – this is because the queries only consist of names

whose orders do not change the meaning of the queries. To measure the similarity of the two result sets, we use the metric Jaccard similarity coefficient (or Jaccard coefficient for short), defined as $|X|/|Y|$, where $X = \text{source response} \cap \text{follow-up response}$ and $Y = \text{source response} \cup \text{follow-up response}$. The source and follow-up responses refer to the source and follow-up queries' result sets, respectively. Obviously, $0 \leq \text{Jaccard coefficient} \leq 1$. A larger Jaccard coefficient indicates higher similarity and, hence, better stability. Given that the vast majority of users would prefer stable search results, poor stability may result in a poor user experience. (In this paper, "user experience" refers to users' perceived quality of the search results.)

Metamorphic Relation: SwapJD:-

The second group of MRs is named "Consistent Ranking." Its first MR is SwapJD, which assesses the search engines' ranking stability based on the concept that a stable search engine should return similar results for similar queries. SwapJD is described as follows: The source query A contains only two words (without quotation marks) and the follow-up query B is constructed by swapping the two words. A stable search engine should return similar results for A and B if these two queries have similar meanings, regardless of their word orders. The similarity can be measured by calculating the Jaccard coefficient of the top x results in the two result lists, where x can be given by the assessor. In this research, we set x to 50, as our experience suggests that most users are unlikely to browse search results beyond the top 50.

Metamorphic Relation: Top1Absent:-

The Top1Absent MR focuses on the ranking quality of the very first result presented in the search results screen. This top result can be considered as the most important one among all search results. According to Imielinski and Signorini, more than 65% of search clicks are done on the first result. Top1Absent is designed by extending the idea of MPSTest, as described below: The source query A is a word randomly selected from an English dictionary (excluding common words such as "is" and "of") and is surrounded by double quotes. Let p_1 be the top result, that is, p_1 is the first listed Web page returned by the search engine for query A. The follow-up query B still uses A as the query term, but restricts the search to p_1 's domain only. The expected relationship is that p_1 should still appear in the search results for B.

Conclusion

Changeable testing (MT) was at first proposed as a confirmation strategy, where transformative relations (MRs) were recognized by alluding to the objective calculation to be actualized. In this paper, we have shown the plausibility of MT being a bound together system for programming confirmation, approval, and quality appraisal. We led an examination on web crawlers, where we recognized MRs from the clients' point of view without alluding to the objective calculations or framework details. All the more for the most part, this approach permits clients to perceive regardless of whether a framework is proper for their particular needs without

finish programming documentation, which is frequently the case with Web administrations, inadequately developed programming, and open source programming. We have connected our way to deal with evaluate a few key programming characteristics of web indexes under various operational profiles without a target and for the most part perceived prophet. All ANOVA investigations returned factually huge outcomes with expansive impact measure esteems. Most numerous examination comes about too had a factual and handy criticalness (with expansive Cohen's d esteems by and large), showing that our approach is compelling. We have additionally examined the researched programming characteristics in the system of the programming quality model standard ISO/IEC 25010.

References

1. Sheeva Afshan, Phil McMinn, and Mark Stevenson. Evolving readable string test inputs using a natural language model to reduce human oracle cost. In International Conference on Software Testing, Verification and Validation (ICST 2013). IEEE, March 2013.
2. Wasif Afzal, Richard Torkar, and Robert Feldt. A systematic review of search-based testing for nonfunctional system properties. *Information and Software Technology*, 51(6):957–976, 2009.
3. Bernhard K. Aichernig. Automated black-box testing with abstract VDM oracles. In *SAFECOMP*, pages 250–259. Springer-Verlag, 1999.
4. Shaukat Ali, Lionel C. Briand, Hadi Hemmati, and Rajwinder Kaur Panesar-Walawege. A systematic review of the application and empirical investigation of search-based test-case generation. *IEEE Transactions on Software Engineering*, pages 742–762, 2010.
5. Nadia Alshahwan and Mark Harman. Automated session data repair for web application regression testing. In *Proceedings of 2008 International Conference on Software Testing, Verification, and Validation*, pages 298–307. IEEE Computer Society, 2008.
6. Nadia Alshahwan and Mark Harman. Automated web application testing using search based software engineering. In *ASE*, pages 3–12, 2011.
7. Dana Angluin. Learning regular sets from queries and counterexamples. *Inf. Comput.*, 75(2):87–106, 1987.
8. W Araujo, L.C. Briand, and Y. Labiche. Enabling the runtime assertion checking of concurrent contracts for the java modeling language. In *Software Engineering (ICSE), 2011 33rd International Conference on*, pages 786–795, 2011.
9. W. Araujo, L.C. Briand, and Y. Labiche. On the effectiveness of contracts as test oracles in the detection and diagnosis of race conditions and deadlocks in concurrent object-oriented software. In *Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on*, pages 10–19, 2011.
10. Andrea Arcuri and Xin Yao. Search based software testing of object-oriented containers. *Information Sciences*, 178(15):3075–3095, 2008.
11. Shay Artzi, Michael D. Ernst, Adam Kiezun, Carlos ? Pacheco, and Jeff H. Perkins. Finding the needles in the haystack: Generating legal test inputs for object-oriented programs. In *1st Workshop on ModelBased Testing and Object-Oriented Systems (M-TOOS), Portland, OR, October 23, 2006*.

12. Egidio Astesiano, Michel Bidoit, Hel' ene Kirchner, ` Bernd Krieg-Bruckner, Peter D. Mosses, Donald San- `` nella, and Andrzej Tarlecki. CASL: the common algebraic specification language. *Theory. Compute. Sci.*, 286(2):153–196, 2002.
13. Todd M. Austin, Scott E. Breach, and Gurindar S. Sohi. Efficient detection of all pointer and array access errors. In *PLDI*, pages 290–301. ACM, 1994.
14. A. Avizienis. The N-version approach to faulttolerant software. *IEEE Transactions on Software Engineering*, 11:1491–1501, 1985.
15. A. Avizienis and L. Chen. On the implementation of N-version programming for software fault-tolerance during execution. In *Proceedings of the First International Computer Software and Application Conference (COMPSAC '77)*, pages 149–155, 1977.
16. Arthur Baars, Mark Harman, Youssef Hassoun, Kiran Lakhota, Phil McMinn, Paolo Tonella, and Tanja Vos. Symbolic search-based testing. In *ASE*, pages 53–62, 2011.
17. A. F. Babich. Proving total correctness of parallel programs. *IEEE Trans. Softw. Eng.*, 5(6):558–574, November 1979.
18. Luciano Baresi and Michal Young. Test oracles. Technical Report CIS-TR-01-02, University of Oregon, Dept. of Computer and Information Science, August 2001. <http://www.cs.uoregon.edu/~michal/pubs/oracles.html>.
19. Sofia Bekrar, Chaouki Bekrar, Roland Groz, and Laurent Mounier. Finding software vulnerabilities by smart fuzzing. In *ICST*, pages 427–430, 2011.
20. Gilles Bernot. Testing against formal specifications: a theoretical view. In *Proceedings of the International Joint Conference on Theory and Practice of Software Development on Advances in Distributed Computing (ADC) and Colloquium on Combining Paradigms for Software Development (CCPSD): Vol. 2, TAPSOFT '91*, pages 99–119, New York, NY, USA, 1991. SpringerVerlag New York, Inc.