

**Ministry of Higher Education and
Scientific Research
University of Mosul
College of Computer Science and
Mathematics
Department of Software**



Design and Implementation of a Testing Tool (DFT-Tool) based on Data Flow Specifications

**A Thesis Submitted to the Council of the
College of
Computer Science and Mathematics
University of Mosul
as a Partial Fulfillment of Requirements
for the Degree of Master of Science
in software**

**By
Abdullah Hazim Ali**

**Supervised by
Asst.Prof. Dr.Nada Nemat Salem**

Abstract

Data flow testing is a significant software testing technique utilized to uncover potential errors and vulnerabilities in code. It focuses on the movement and processing of data within the code, enabling the identification of issues like uninitialized variables, unexecuted code, and unused variables. By ensuring the software's proper behavior, correct operation, and adherence to user requirements, data flow testing plays a crucial role. It aids in the detection and resolution of uninitialized variables, data dependencies. Ultimately, employing data flow testing enhances software quality and reduces end-users' encountered issues.

The proposed data flow testing tool, known as the DFT tool, has been developed to achieve its intended objectives in multiple stages. In the initial stage, the tool translates and analyzes the input code into an abstract syntax tree by utilizing a custom-built parser. This abstract syntax tree serves as a suitable model for drawing the flow graph. Subsequently, the DFT tool breaks down the input code into blocks, each containing a set of statements.

Moving to the second stage, the DFT tool generates a table that identifies transformations for each variable across each line of the block. These transformations are categorized as 'D' for definition, 'U' for use (which may involve prediction or computation), 'DU' for a variable defined and then used, and 'UD' for a variable used after being defined.

In the third stage, the DFT tool creates another table to detect anomalies, such as variables that are defined but not used. It also categorizes variables as 'D' for defined, 'U' for used, and 'K' for killed, considering all

lines of the tested code. Additionally, the tool identifies the paths of each variable, distinguishing between clear and unclear paths.

Finally, in the fourth stage, the DFT tool generates a table for merging paths that contain duplicated nodes. It then runs test cases on the resulting merged paths.

Overall, the proposed DFT tool successfully accomplishes its objectives through a systematic process involving stages for drawing the flow graph, analyzing variable transformations, detecting anomalies, identifying variable paths, and merging paths to facilitate effective data flow testing.

The proposed DFT tool exhibits exceptional levels of coverage and accuracy when applied to tested code. Depending on the behavior of the code being tested, the tool can achieve an accuracy rate surpassing 95%. Moreover, the testing coverage extends to over 90%, ensuring a comprehensive assessment of the code's functionality and potential issues.



وزارة التعليم العالي والبحث العلمي
جامعة الموصل
كلية علوم الحاسوب والرياضيات
قسم البرمجيات

تصميم وتنفيذ أداة الاختبار (DFT-Tool) بناءاً على مواصفات تدفق البيانات

رسالة مقدمة

إلى مجلس كلية علوم الحاسوب والرياضيات في جامعة الموصل
كجزء من متطلبات نيل شهادة ماجستير علوم في
البرمجيات

من قبل
عبدالله حازم علي

بإشراف
ا.م.د. ندى نعمت سليم

الخلاصة

إن اختبار تدفق البيانات هو أسلوب مهم يستخدم في اختبار البرامج لاكتشاف الأخطاء المحتملة ونقاط الضعف في الكود. حيث يركز على كيفية معالجة البيانات وانتقالها عبر الكود ، ويمكن أن يساعد في تحديد مشكلات مثل المتغيرات غير المهيأة ، والتعليمات البرمجية التي لم يتم تنفيذها مطلقًا ، والمتغيرات التي لا يتم استخدامها مطلقًا. يساعد هذا في ضمان أن يتصرف البرنامج كما هو متوقع ، ويعمل بشكل صحيح ، ويلبي متطلبات المستخدم. يمكن أن يساعد اختبار تدفق البيانات في تحديد المتغيرات غير المهيأة ، والتعليمات البرمجية الميتة ، وتبعيات البيانات ، والتحقق من صحة الإدخال ، والبيانات التي أدخلها المستخدم وحلها. يمكن أن يساعد أيضًا في تحديد المشكلات المحتملة ، مثل البيانات غير الصالحة أو المتغيرات غير المهيأة ، والتأكد من التحقق من صحة البيانات التي أدخلها المستخدم ومعالجتها بشكل صحيح. يمكن أن يساعد ذلك في تحسين جودة البرنامج وتقليل عدد المشكلات التي يواجهها المستخدمون النهائيون.

وقد تم تطوير أداة DFT المقترحة لنجاح اختبار تدفق البيانات و خلال ذلك تم تحقيق جميع الأهداف المخطط لها بمراحل متعددة ، حيث في المرحلة الأولى تم تحويل البرنامج النصي الى رسم تدفق البيانات بعد عملية الاعراب وتحليل البرنامج الذي يراد اختباره الى شجرة الاعراب باستخدام المحلل اللغوي الذي تم إنشاؤه ، إن شجرة الاعراب المجردة هي النموذج المناسب للرسم البياني ، حيث تقوم أداة DFT المقترحة بتقسيم كود الإدخال إلى كتل تحتوي كل كتلة على عدد من الشفرات البرمجية ، وفي المرحلة الثانية تقوم الاداة المقترحة بعمل جدول لاكتشاف انواع التحويلات لكل متغير عبر كل سطر من الكتل البرمجية وتصنيفها إلى D :التعريف ، U(قد يكون الاستخدام: التنبؤ أو الحساب) ، زوج (DU) المتغير المحدد ثم زوج المستخدم UD(متغير يستخدم بعد التعريف). وفي المرحلة الثالثة من عمل أداة DFT يتم عمل جدول آخر لاكتشاف الحالات الشاذة مثل المتغير الذي تم تعريفه ولم يتم استخدامه وتحديد المتغيرات إلى: معرف (D) ، مستخدم (U) ، مهمل بدون استخدام (K) على طول جميع أسطر الكود الذي تم اختباره ، سيحدد أيضًا مسارات كل متغير ونوعه إلى: واضح وغير واضح ، في المرحلة الرابعة ، تقوم الاداة المقترحة بعمل جدول لدمج المسارات التي تحتوي على تكرار للعقدة وتنفيذ حالات الاختبار على المسارات التي تم إنتاجها بعد عملية الدمج .

DFT المقترحة تتمتع الأداة بتغطية ودقة عالية جدًا للبرامج التي تم اختبارها ، وقد تصل هذه الدقة إلى أكثر من 95% . وأيضًا تصل نسبة تغطية المسارات الى أكثر من 90% اعتمادًا على سلوك الكود الذي تم اختباره.