

令和 6 年 5 月 23 日現在

機関番号：14401

研究種目：基盤研究(C)（一般）

研究期間：2020～2023

課題番号：20K11747

研究課題名（和文）グラフデータベースをバックエンドとするソフトウェアに対するテスト手法の確立

研究課題名（英文）Development of testing methodology for software with a graph database backend

研究代表者

土屋 達弘（Tsuchiya, Tatsuhiro）

大阪大学・大学院情報科学研究科・教授

研究者番号：30283740

交付決定額（研究期間全体）：（直接経費） 3,300,000円

研究成果の概要（和文）：グラフデータベースを用いるソフトウェアのテストについて研究を行った。まず、グラフデータベースを理解するために、グラフデータベースを用いてテストを支援する手法を開発した。情報システムの設計仕様書とテスト仕様書との関係性をグラフデータベース上で管理する。データベースに対してクエリを用いて情報を取得することで、どの機能がテスト不足かといった情報の可視化を実現した。この知見を元に、クエリへ人為的にバグを埋め込む操作とツールを開発し、多数のバグを含むクエリを自動的に生成することを実現した。これらのクエリを用いて、テストスイートのバグ検出能力の計測するといった、ミューテーションテストを可能とした。

研究成果の学術的意義や社会的意義

ネットワークの形でデータを管理するグラフデータベースは、近年普及が進んでいる新しいタイプのデータベースである。グラフデータベースを用いるソフトウェアのテストについて、本研究では取り組んだ。グラフデータベースとソフトウェアとのやり取りを行うための、ある種のプログラムであるクエリについて、人為的にバグを埋め込む方法を開発した。このような予めバグを含むことが分かっているプログラムを正しく検出できるかを調べることで、既存のテストが十分かといった評価が可能になり、情報システムの信頼性に貢献することが可能となる。

研究成果の概要（英文）：The research aimed to address the testing of software using graph databases. First, in order to understand graph databases, we developed a method to support software testing using a graph database. In this method, the relationship between the design and test specifications of an information system is managed on a graph database. By retrieving information using queries to the database, users can visualize useful information, such as which functions are under-tested.

Based on this knowledge, we developed mutation operators for queries and a tool implementing these operators to embed bugs into queries artificially. This tool can automatically generate buggy queries, which can be used for mutation testing, such as measuring the bug detection capability of an existing test suite.

研究分野：ディペンダブルシステム

キーワード：グラフデータベース ミューテーションテスト グラフクエリ テストカバレッジ

1. 研究開始当初の背景

本研究では、グラフデータベースを用いるソフトウェアを対象として、そのテスト手法の確立を目的として研究を計画した。グラフとは、ネットワーク構造を表す基本的な数学的オブジェクトである。研究開始当初、ソーシャルネットワークの管理や解析や、リコメンダー・システムといった応用において、巨大なグラフを管理、処理する必要性が高まっていた。

また、このような背景から、グラフデータベースと呼ばれる新しいデータベースシステムが注目を集めて始めていた。従来の関係データベースではテーブルを基本的なデータの表現形式としているが、ネットワーク構造をテーブルで表現し管理しようとする、煩雑で非効率になる場合が知られている。グラフデータベースは、ネットワーク構造そのものを管理することができるため、上記のような応用に極めて適している。そこで、研究開始当初、今後はグラフデータベースをバックエンドに用いた応用システムが、更に普及していくことと予想した。現在、この予想通り、グラフデータベースはよりポピュラーなものとなっている。

2. 研究の目的

本研究では、グラフクエリに注目することで、グラフデータベースを用いるソフトウェアのテスト手法を確立することを目的とした。グラフクエリは、グラフデータベースへの操作を記述しており、ソフトウェアがグラフデータベースにアクセスするためには、グラフクエリを用いて指示を行う必要がある。適切なグラフクエリを実行することで、グラフとして表されている知識やデータから、必要な情報を抽出したり、グラフ自体を変更したりすることができる。一方で、グラフクエリが正しく設計されていない場合、ソフトウェアは不具合を有することとなる。

ソフトウェアのテスト手法として、ミューテーションテストが広く知られている。これは、プログラムに不具合(バグ)を埋め込んだミュータントを人為的に生成し、これらのミュータントが正しく不具合を持つものとして検出されるかを評価する手法である。ミューテーションテストによって、利用しているソフトウェアのテスト方法やテストデータが十分かといった評価が可能となる。

通常のプログラミング言語を前提としたミューテーションテストについては、数多くの研究がなされてきた。しかし、グラフクエリ上の不具合については、これまで研究が行われていなかった。そこで本研究では、グラフクエリのバグを人工的に生成し、グラフクエリを用いるソフトウェアのミューテーションテストを実現することを目指した。

3. 研究の方法

まず、グラフデータベース、グラフクエリ、また、それらを用いるソフトウェアを理解するために、グラフデータベースを用いたアプリケーションを開発することとした。開発するアプリケーションとしては、グラフデータベースの理解に加えて、得られた知見が本研究の過程で有用に活用できるように、ソフトウェアのテストに応用できるようなものとした。具体的には、ソフトウェアシステムに対する機能を記述した機能仕様記述と、テストケースについて記述された文章であるテスト仕様記述が与えられたときに、それらの関係性をグラフデータベースで管理し、必要な情報をグラフクエリを通じて抽出できるようなものとした。

このアプリケーションの開発で得られた知見をもとに、グラフクエリに対して、自動的に不具合を埋め込む操作を設計するとともに、この操作を実装したツールを開発した。

4. 研究成果

(1) まず、先述の通り、グラフデータベースを用いたテスト支援アプリケーションを開発した。このアプリケーションでは、情報システムの開発における多数の機能仕様記述と、同じく多数のテスト仕様記述との関連性をネットワークとして表し、それをグラフデータベースで管理する。仕様間の関連度を使われている語彙の類似性から推定して、各仕様をノードとして表し、仕様間を関連度を属性とするリンクで接続することで、グラフの形式で管理することが可能となる。グラフデータベースには Neo4j を用いた。その上で、このデータベースとグラフクエリを用いて、システムの各機能が十分にテストされているか、あるいは、テストが不十分な機能はどれか、といった、テストカバレッジに関する問いに答えることのできるアプリケーションを作成した。

実際の商用のシステム開発で得られたデータに対して、作成したアプリケーションを実行し、その処理性能の評価を行った。また、同様の処理を関係データベースと SQL を用いて実現した場

合との比較も行った。この結果、グラフクエリは SQL クエリよりも各段に簡潔に取り出すべき情報を記述できること、および、処理性能では関係データベース (PostgreSQL を使用) の方が優れているが、グラフデータベースでも十分実用的な実行時間で処理が可能なが示せた (右表)。この成果は国内の研究会と国際会議 [1] で発表した。

Query type	Neo4j	PostgreSQL
Query 1	0.004	0.056
Query 2	0.054	0.075
Query 3	0.002	0.054
Query 4	0.047	0.125
Query 5	0.359	0.048
Query 6	5.086	0.098
Query 7	13.440	0.098

表. 実行時間の比較 [1] (単位: 秒)。グラフデータベースでは比較的時間がかかる処理もあるが、実用上問題ないことが示せた。

(2) 次に、グラフクエリに対するミューテーションテストを開発した。ここでは、Neo4j でも用いられている代表的なグラフクエリ言語である Cypher で記述されたグラフクエリを対象とした。Cypher で利用可能な様々な構文について、多数のミューテーション操作、つまり、不具合を埋め込む操作を定義した。これらの操作には、関係データベースのクエリ言語である SQL に対する既存のミューテーション操作を参考にしたものもあるが、Cypher 独自の構文に注目して、まったく新しく定義したものも多い。

こうして定義したミューテーション操作を実装したツールを作成した。このツールに Cypher で記述されたグラフクエリを入力すると、適用可能な操作を実行して、不具合を含む多数のグラフクエリを得ることができる。このツールは Node.js と JavaScript を用いて開発した。図に、入力されたグラフクエリから、不具合が埋め込まれた多数のクエリが出力されている様子を示す。この成果についても、国内研究会と国際会議 [2] にて発表を行っている。

```

KEY : MATCH Clause
  OPTIONAL MATCH (r:Requirement)-[s:Similarity]-(t:Testcase) WHERE (s.value >= 0.5) R

KEY : Null in Results
  MATCH (r:Requirement)-[s:Similarity]-(t:Testcase) WHERE (s.value >= 0.5) RETURN DIS
ame ORDER BY req_name ASC

KEY : ORDER BY clause
  MATCH (r:Requirement)-[s:Similarity]-(t:Testcase) WHERE (s.value >= 0.5) RETURN DIS
  MATCH (r:Requirement)-[s:Similarity]-(t:Testcase) WHERE (s.value >= 0.5) RETURN DIS

KEY : RETURN Clause
  MATCH (r:Requirement)-[s:Similarity]-(t:Testcase) WHERE (s.value >= 0.5) RETURN r.n

32 mutants were generated.

```

図. スクリーンショット [2]。入力されたグラフクエリから、32 個のバグを含むクエリ (ミュータント) が生成されている。

引用文献

[1] Shingo Ariwaka, Hiroyuki Nakagawa, Tatsuhiro Tsuchiya, “Graph queries for analyzing the coverage of requirements by test cases,” International Conference on Software Engineering and Knowledge Engineering 2021 (SEKE 2021). pp.544-549, doi: 10.18293/SEKE2021-122.

[2] S. Ariwaka and T. Tsuchiya, “On Mutation Testing of Graph Database Queries in the Cypher Language,” 2023 30th Asia-Pacific Software Engineering Conference (APSEC), pp. 579-583, doi: 10.1109/APSEC60848.2023.00074.

5. 主な発表論文等

〔雑誌論文〕 計4件（うち査読付論文 2件 / うち国際共著 0件 / うちオープンアクセス 0件）

1. 著者名 Shingo Ariwaka, Hiroyuki Nakagawa, Tatsuhiro Tsuchiya	4. 巻 ISBN: 978-1-7138-3081-8
2. 論文標題 Graph queries for analyzing the coverage of requirements by test cases	5. 発行年 2021年
3. 雑誌名 Proceedings of 33rd International Conference on Software Engineering & Knowledge Engineering	6. 最初と最後の頁 544-549
掲載論文のDOI (デジタルオブジェクト識別子) 10.18293/SEKE2021-122	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -
1. 著者名 有若 新悟, 土屋達弘	4. 巻 2021-SE-209
2. 論文標題 グラフクエリに対するミュートーションテストの提案	5. 発行年 2021年
3. 雑誌名 情報処理学会 研究報告ソフトウェア工学 (SE)	6. 最初と最後の頁 1-4
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -
1. 著者名 有若新悟, 中川博之, 土屋達弘	4. 巻 IEICE-KBSE2020-25
2. 論文標題 要求-テストケース間のカバレッジ分析におけるグラフクエリの応用可能性の検討	5. 発行年 2020年
3. 雑誌名 信学技報	6. 最初と最後の頁 53-58
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -
1. 著者名 Ariwaka Shingo, Tsuchiya Tatsuhiro	4. 巻 ISBN: 979-8-3503-4417-2
2. 論文標題 On Mutation Testing of Graph Database Queries in the Cypher Language	5. 発行年 2023年
3. 雑誌名 2023 30th Asia-Pacific Software Engineering Conference (APSEC)	6. 最初と最後の頁 579-583
掲載論文のDOI (デジタルオブジェクト識別子) 10.1109/APSEC60848.2023.00074	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

〔学会発表〕 計1件（うち招待講演 0件 / うち国際学会 0件）

1. 発表者名 有若新悟, 中川博之, 土屋達弘
2. 発表標題 要求仕様に対するテストカバレッジ分析におけるグラフクエリの適用について
3. 学会等名 ソフトウェア・シンポジウム 2021
4. 発表年 2021年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織	氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
---------	---------------------------	-----------------------	----

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関
---------	---------