

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6627658号
(P6627658)

(45) 発行日 令和2年1月8日(2020.1.8)

(24) 登録日 令和1年12月13日(2019.12.13)

(51) Int.Cl.

F 1

G06F 11/36 (2006.01)
G06F 11/07 (2006.01)G06F 11/36 124
G06F 11/07 193

請求項の数 12 (全 21 頁)

(21) 出願番号 特願2016-125499 (P2016-125499)
 (22) 出願日 平成28年6月24日 (2016.6.24)
 (65) 公開番号 特開2017-45446 (P2017-45446A)
 (43) 公開日 平成29年3月2日 (2017.3.2)
 審査請求日 平成31年1月28日 (2019.1.28)
 (31) 優先権主張番号 14/835,561
 (32) 優先日 平成27年8月25日 (2015.8.25)
 (33) 優先権主張国・地域又は機関
米国(US)

(73) 特許権者 000005223
富士通株式会社
神奈川県川崎市中原区上小田中4丁目1番
1号
 (74) 代理人 100107766
弁理士 伊東 忠重
 (74) 代理人 100070150
弁理士 伊東 忠彦
 (74) 代理人 100192636
弁理士 加藤 隆夫
 (72) 発明者 吉田 浩章
アメリカ合衆国、カリフォルニア州 94
085, サニーヴェイル, イースト アー
クス アヴェニュー 1240番 フジツウ
ラボラトリーズ アメリカ内
最終頁に続く

(54) 【発明の名称】ソフトウェアプログラムを修復する方法、記憶媒体及びシステム

(57) 【特許請求の範囲】

【請求項1】

コンピュータが、

ソフトウェアプログラムに関する第1プログラム表現を生成し、

テストツールを利用して前記ソフトウェアプログラムにおける誤りを特定し、

前記誤りの特定に応じて、前記ソフトウェアプログラムを、修復テンプレートを利用して修正し、

前記ソフトウェアプログラムを修正した後に、前記ソフトウェアプログラムに関する第2プログラム表現を生成し、

前記第1プログラム表現と前記第2プログラム表現との比較に基づいて、前記第1プログラム表現に対する前記第2プログラム表現の相違を検出し、 10

修正が不適切であるか否かを示すアンチパターン条件を前記相違が充足することを判定し、

前記アンチパターン条件を前記相違が充足することに応じて、前記ソフトウェアプログラムの修正を不許可にすること、

を特徴とする方法。

【請求項2】

前記第1プログラム表現は第1制御フローグラフを含み、前記第2プログラム表現は第2制御フローグラフを含むことを特徴とする、請求項1に記載の方法。

【請求項3】

前記アンチパターン条件は、制御フローグラフのエグジットパスの削除を含むことを特徴とする、請求項2に記載の方法。

【請求項4】

前記アンチパターン条件は、制御フローグラフのノードの削除を含むことを特徴とする、請求項2に記載の方法。

【請求項5】

前記アンチパターン条件は、先行する制御フローグラフのリターンノードの前に、制御フローグラフのリターンパスを挿入することを含むことを特徴とする、請求項2に記載の方法。

【請求項6】

前記第1プログラム表現は第1データフローグラフを含み、前記第2プログラム表現は第2データフローグラフを含むことを特徴とする、請求項1に記載の方法。

【請求項7】

前記アンチパターン条件は、データフローノードの到来エッジの削除を含むことを特徴とする、請求項6に記載の方法。

【請求項8】

前記アンチパターン条件は、第1データフローノード及び第2データフローノードを含むデータフローノードの中斷を含み、前記第1データフローノードは前記ソフトウェアプログラムのループのループ条件ステートメントに対応し、前記第2データフローノードは、前記ソフトウェアプログラムのループ内に存在し且つ前記ループ条件ステートメントの変数に関連する指定ステートメントに対応することを特徴とする、請求項6に記載の方法。

【請求項9】

前記アンチパターン条件は：「リターン」ステートメントの削除；「エグジット」ステートメントの削除；正規の表現に合致する名称を有するメソッドコールを含むステートメントの削除；「アサート」ステートメントの削除；制御フローグラフノードに関する全てのステートメントの削除；経路の条件の述語の否定の追加；条件の検査をディセーブルにする「真」の述語の追加；条件をディセーブルにする「偽」の述語の追加；制御フローブロックにおいて最後のステートメントでないリターンステートメントの制御フローブロックへの挿入；変数の定義であって対応する変数が始めて使用される際に不定となってしまうものの削除；及び、ループのループ条件がループ変数に基づく場合における、ループ内のループ変数についての指定ステートメントの削除；のうちの1つ以上に基づくことを特徴とする、請求項1に記載の方法。

【請求項10】

前記第1プログラム表現は、修正前の前記ソフトウェアプログラムの少なくとも一部分のコピー；及び、修正前の前記ソフトウェアプログラムの少なくとも一部分の抽象構文ツリー；のうちの1つ以上を含み、

前記第2プログラム表現は、修正後の前記ソフトウェアプログラムの少なくとも一部分のコピー；及び、修正後の前記ソフトウェアプログラムの少なくとも一部分の抽象構文ツリー；のうちの1つ以上を含むことを特徴とする、請求項1に記載の方法。

【請求項11】

テストスーストを利用してソフトウェアプログラムにおける誤りを特定し、

前記誤りの特定に応じて、前記ソフトウェアプログラムを、修復テンプレートを利用して修正し、

修正が不適切であるか否かを示すアンチパターン条件を前記修正が充足するか否かを判定し、

前記アンチパターン条件を前記修正が充足することに応じて、前記修正を不許可にし、

前記アンチパターン条件を前記修正が充足しないことに応じて、修正された前記ソフトウェアプログラムについて検査を実行する

処理をコンピュータに実行させることを特徴とするプログラム。

【請求項12】

10

20

30

40

50

—ソフトウェアプログラムに関する第1プログラム表現を生成する第1生成部と、
テストツールを利用して前記ソフトウェアプログラムにおける誤りを特定する誤り特定部と、

前記誤りの特定に応じて、前記ソフトウェアプログラムを、修復テンプレートを利用して修正する修正部と、

前記ソフトウェアプログラムを修正した後に、前記ソフトウェアプログラムに関する第2プログラム表現を生成する第2生成部と、

前記第1プログラム表現と前記第2プログラム表現との比較に基づいて、前記第1プログラム表現に対する前記第2プログラム表現の相違を検出する相違検出部と、

修正が不適切であるか否かを示すアンチパターン条件を前記相違が充足するか否かを判定する判定部と、10

前記アンチパターン条件を前記相違が充足することに応じて、前記修正を不許可にする不許可部と、

前記アンチパターン条件を前記相違が充足しないことに応じて、修正された前記ソフトウェアプログラムについて検査を実行する検査部と

を備えることを特徴とする修復装置。

【発明の詳細な説明】

【技術分野】

【0001】

20

本開示で説明される実施形態はソフトウェアプログラムの修復に関連する。

【背景技術】

【0002】

ソフトウェアプログラムはその中にしばしば誤り(一般に、「バグ」と言及される)を含み、それらは意図されるように動作しないおそれがある。ソフトウェアプログラム中の誤りを識別及び修正する試みとしてしばしば自動修復システムが使用される。

【0003】

本開示で対象とされる特定事項は、何れの欠点をも解決する実施形態や上述したような環境でしか動作しない実施形態には限定されない。むしろこの「背景技術」は、本開示で説明される何らかの実施形態が実施されてよい技術分野の一例を示すために提供されているに過ぎない。30

【発明の概要】

【0004】

実施形態の一側面によれば、1つ以上のシステム又は方法は、テストツール(test suit e)を利用してソフトウェアプログラム中の誤りを特定するように構成される。システム又は方法は、誤りの特定に応じて、ソフトウェアプログラムを修復テンプレートで修正するように更に構成されてよい。更に、システム又は方法は、その修正がアンチパターン条件(anti-pattern condition)を充足するか否かを判定するように構成されてよい。アンチパターン条件は、修正が不適切であるか否かを示してもよい。システム又は方法は、修正がアンチパターン条件を充足することに応じて、その修正を許可しないように構成され、あるいは、修正がアンチパターン条件を充足しないことに応じて、修正後のソフトウェアプログラムについて更なるテストを実行するように構成されてもよい。40

【0005】

実施形態の目的及び効果は、特許請求の範囲で特に規定される要素、特徴及び組み合わせにより少なくとも実現及び達成される。

【0006】

以下の一般的な説明及び以下の具体的な説明は何れも例示として与えられる具体例であり、請求項に係る本発明の限定ではない。

【図面の簡単な説明】

【0007】

40

50

添付図面を利用することにより、追加的な特殊性及び詳細とともに例示的な実施形態が記述及び説明される。

【0008】

【図1A】ソフトウェアプログラムを修復することに関する環境例を示す図。

【0009】

【図1B】検査対象コードの例示的な制御フローグラフ、及び、制御フローグラフに対応する検査対象コードの例示的な一部分を示す図。

【0010】

【図1C】修正された検査対象コードについての修正された例示的な制御フローグラフ、及び、修正された制御フローグラフに対応する修正された検査対象コードについての修正された例示的な一部分を示す図。 10

【0011】

【図1D】修正された検査対象コードについての修正された別の例示的な制御フローグラフ、及び、修正された制御フローグラフに対応する修正された検査対象コードについての修正された別の例示的な一部分を示す図。

【0012】

【図1E】修正された検査対象コードについての修正された別の例示的な制御フローグラフ、及び、修正された制御フローグラフに対応する修正された検査対象コードについての修正された別の例示的な一部分を示す図。

【0013】

【図1F】実行されるデータフローの修正を有するコード例を示す図。 20

【0014】

【図2】ソフトウェアプログラムを修復するように構成されるコンピューティングシステム例を示す図。

【0015】

【図3】ソフトウェアプログラムを修復する方法例のフローチャートを示す図。

【発明を実施するための形態】

【0016】

本開示で説明される実施形態は、ソフトウェアプログラムを修復する方法及びシステムに関連する。ソフトウェアプログラムは誤り(一般に、「バグ」として言及される)をしばしば含み、誤り(fault)は、ソフトウェアプログラムを、意図されていない仕方で動作させる。誤りを検出して修正し、ソフトウェアプログラムを修復するために、自動修復システム及び技術がしばしば使用される。しかしながら、一般的に使用されている自動修復システム及び技術は、誤りを解決する試みでソフトウェアプログラムに、ソフトウェアプログラム中の誤りを適切には修復しない修正をしばしば施してしまう。 30

【0017】

本開示で説明される1つ以上の実施形態によれば、ソフトウェアプログラムの修復候補として機能する、ソフトウェアプログラムの修正が、分析される。分析は、修正が目の前の誤りに対する適切な修正に対応しないかもしれない点で、修正が不適切であるか否かを判定するために使用される。この判定は、修正が、ソフトウェアプログラムの動作についての不適切な(例えば、非現実的な又は過剰な)修正に対応し、従って、誤りに対する適切な修正に対応しそうにないものであるか否かを検査することによりなされてよい。修正がそのような基準を満たすことに応じて、修正は不許可にされてもよい。 40

【0018】

具体的には、一実施形態では、修正がアンチパターン条件(本開示では、「安定パターン」又は「アンチパターン群」のように言及されてもよい)を充足するか否かが判定され、その判定は、修正が不適切であるか否かを示してもよいし、あるいは、修正が誤りについての適切な修復に対応するか否かの指示を提供してもよい。一実施形態では、修正がアンチパターン条件を充足することは、修正が不適切であることを示してもよい。これに対して、アンチパターン条件を充足しないことは、更なる分析のためにその修正を提示する 50

ことに対して確実性を提供する。あるいは、修正がアンチパターンを充足しないことに応じて、修正されることに關し、ソフトウェアプログラムに更なる検査が実行されてもよい。上述したような修正についての許可又は不許可は、修正に關し、ソフトウェアプログラムに対してテストスツーツが動作する前に実行されてよい静的な分析に基づいて実行されてもよい。静的な分析は、不要な処理を減らし、ソフトウェアプログラムを修復する効率を改善する。少なくとも上記の理由により、本開示で説明されるシステム及び方法は、ソフトウェアプログラムの自動検査及び修復の改善を促す。

【0019】

以下、添付図面を参照しながら本開示による実施形態を説明する。

【0020】

図1Aは、ソフトウェアプログラムを修復することに関する環境例100を示す図であり、その環境例は本開示で説明される少なくとも1つの実施形態に従って構成される。環境100は、誤りに關し、検査対象コード104を分析するように構成される修復モジュール106を含む。修復モジュール106は、その誤りを修復し、検査対象コード104と比較した場合に1つ以上の修正を含む修正された検査対象コード108を出力するように構成され、その修正は、検査対象コード104内の1つ以上の誤りを修復するものである。

【0021】

検査対象コード104は電子データを含み、電子データは、例えば、ソフトウェアプログラム、ソフトウェアプログラムのコード、ライブラリ、アプリケーション、スクリプト、あるいは、処理装置による実行のための他のロジック又は命令などであってもよい。一実施形態において、検査対象コード104は、ソフトウェアプログラムの完全なインスタンスを含んでもよい。追加的又は代替的に、検査対象コード104はソフトウェアプログラムの一部分を含んでもよい。検査対象コード104は、ソフトウェアプログラムに使用されてよい適切な任意の種類のコンピュータ言語で書かれてよい。

【0022】

修復モジュール106は、検査対象コード104の1つ以上の修正を実行し、修正された検査対象コード108を生成することを、コンピューティングデバイスに行わせるように構成されるコード及びルーチンを含んでよい。追加的又は代替的に、修復モジュール106はハードウェアを用いて実現されてもよく、ハードウェアは、例えば、プロセッサ、マイクロプロセッサ(例えば、1つ以上の動作を実行する又はそのパフォーマンスを制御するもの)、フィールドプログラマブルゲートアレイ(FPGA)、又は、特定用途向け集積回路(ASIC)等を含んでよい。なお、修復モジュール106は、ハードウェア及びソフトウェアの組み合わせを用いて実現されてもよい。本開示において、修復モジュール106により実行されるように説明される動作は、修復モジュール106が、対応するシステムに実行を指図する動作を含んでもよい。

【0023】

修復モジュール106は、検査対象コード104中の1つ以上の誤りを修復(又は修正)するために使用されてよい検査対象コード104に関する一連の動作を実行するように構成される。一実施形態において、修復モジュール106は、修復テンプレート112、1つ以上のテストスツーツ111、及び、1つ以上のアンチパターン109に基づいて、1つ以上の修復処理を実行するように構成されてよい。

【0024】

修復テンプレート112は適切な任意のタイプの命令又はルーチンを含んでよく、その命令又はルーチンは、実行される場合に、検査対象コード104中に誤りが存在することに応じて、検査対象コード104に対する1つ以上の修正を実行するように構成される。修正は、誤りを修復する又は誤りの修復を試みる検査対象コード104における変更を含む。

【0025】

上述したように、アンチパターン109は、修正が不適切であるか否かに関する指示を提供してもよい(不適切であるとは、例えば、修正が1つ以上のアンチパターン109を充足することであってもよい)。追加的又は代替的に、アンチパターン109は、修正に関する更な

10

20

30

40

50

る検査が実行されるように、修正が正しい修復に対応している確実性を提供することを促す(正しい修復は、例えば、修正が何れのアンチパターン109も充足しないことであってもよい)。以下に説明されるように、その充足は対応する修正が不適切であることを示すので、1つ以上のアンチパターン109を満たすタイプの修正は許可されなくてもよい。

【0026】

一実施形態において、アンチパターン109は、不適切な修正に起因するソフトウェアプログラム制御フロー変化に基づいてもよいし、あるいは、それを含んでもよい。例えば、一実施形態において、1つ以上のアンチパターン109は、ソフトウェアプログラムの制御フロー・ブロックのエグジットパス(exit path)の削除に基づいていてもよいし、あるいは、それを含んでいてもよい。特に、一実施形態において、1つ以上のアンチパターン109は、10 ソフトウェアプログラムの「リターン(return)」ステートメントの削除、ソフトウェアプログラムの「エグジット(exit)」ステートメントの削除、正規の表現に合致する名称のメソッド呼び出し(method call)を含むソフトウェアプログラムのステートメントの削除、あるいは、ソフトウェアプログラムの「アサート(assert)」の削除に基づいていてもよいし、あるいは、それらを含んでいてもよい。これら又は他の実施形態において、1つ以上のアンチパターン109は、修正によって削除が引き起こされる制御フローグラフのエグジットパス(「制御フローグラフエグジットパス」)の削除に基づいていてもよいし、あるいは、それを含んでいてもよい。

【0027】

具体例として、図1Bは、検査対象コード104の修正前の検査対象コード104に対して生成された例示的な制御フローグラフ120aを示す。制御フローグラフ120aは、検査対象コード104の例示的な一部分118aに対応してもよい。更に、図1Cは、修正が当該一部分118aのリターンステートメント122を削除した後に、検査対象コード104に対して生成された例示的な制御フローグラフ120bを示す。リターンステートメント122の削除は、図1Cの修正部分118bに示されている。特定の例では、制御フローグラフ120aのエグジットパス124が制御フローグラフ120bに含まれないように、修正がなされてよい。従って、この特定の例における修正は、エグジットパス124を削除する結果となる。図示の例では、制御フローグラフ120bの破線は、対応するノード及びエッジが制御フローグラフ120a内には存在するが、修正に起因して、制御フローグラフ120bには最早存在しないことを示す。以下に説明されるように、制御フローグラフ120aから制御フローグラフ120bへの変化をもたらすことになる修正は許可されず、その理由は、制御フローグラフのエグジットパスを削除するという特定のアンチパターンを充足するからである。

【0028】

別の例による一実施形態では、1つ以上のアンチパターン109は、制御フローグラフのノード(「制御フローグラフノード」)に対応する制御フロー・ブロックの全てのステートメントを削除することに基づいてもよいし、あるいは、それを含んでもよい。追加的又は代替的に、1つ以上のアンチパターン109は、制御フローグラフのノードに対応する全てのステートメントの削除に意味的に等価である修正に基づいてもよいし、あるいは、それを含んでもよい。例えば、一実施形態において、1つ以上のアンチパターン109は、経路条件の述語の否定の追加(これは、例えば、“C & & !C”又は“C||!C”等のようなパスする条件となる)、条件の検査をディセーブルする「真」の述語の追加、あるいは、条件をディセーブルにする「偽」の述語の追加に基づいていてもよく、なぜなら、これら各々は、制御フローグラフのノードに対応するコード部分を入力しない結果をもたらすからである。従って、これら又は他の実施形態において、アンチパターン109は、修正によって削除が引き起こされる制御フローグラフノードの削除に基づく又はそれを含んでもよい。

【0029】

一例として、図1Dは、コード修正後の検査対象コード104に対して生成された例示的な制御フローグラフ120cを示す。特定の例において、制御フローグラフ120aの128, 130, 132が制御フローグラフ120cに含まれないように、修正がなされる。一実施形態では、図1Bの一部分118aの制御フロー・ブロック126の全てのステートメントが削除されるように、修

10

20

30

40

50

正がなされてもよい。制御フローブロック126のステートメントの削除は、図1Dの修正された部分118cに示されている。

【0030】

追加的又は代替的に、修正は、制御フローブロック126への進入を許可しないことにより、制御フローブロック126のステートメントの削除と意味的に同様に機能してもよい。そのような修正の具体例は、図1Dの修正部分118dには含まれ、図1Bの一部分118aには含まれていない条件127により示される。

【0031】

この特定の具体例における修正は、制御フローグラフノード128, 130, 132の削除の結果となる。図示の例において、制御フローグラフ120cにおける破線は、対応するノード及びエッジが制御フローグラフ120aには存在していたが、修正に起因して、制御フローグラフ120cには最早存在しないことを示す。以下に説明されるように、1つ以上の制御フローグラフノードを削除するという特定のアンチパターンを満たすことに起因して、制御フローグラフ120aから制御フローグラフ120cへの変更をもたらす修正は許容されなくてもよい。

10

【0032】

別の例として、一実施形態において、1つ以上のアンチパターン109は、ソフトウェアプログラムの制御フローブロックからの時期尚早の復帰(premature return)を引き起こす修正に基づいていてもよいし、あるいは、それを含んでもよい。例えば、一実施形態において、1つ以上のアンチパターン109は、挿入される「リターン」ステートメントが制御フローブロックの最終ステートメントではない、制御フローブロックへの「リターン」ステートメント(例えば、"return (void)" 又は "return(const)")の挿入に基づいていてもよいし、あるいは、それを含んでもよい。これら又は他の実施形態において、アンチパターン109は、挿入が修正によって引き起こされる、先行する制御フローグラフのリターンノードの前に、制御フローグラフリターンパスを挿入することに基づいていてもよいし、あるいは、それを含んでもよい。

20

【0033】

一例として、図1Eは、検査対象コード104の修正後の検査対象コード104に対して生成された例示的な制御フローグラフ120dを示す。特定の例では、制御フローグラフリターンパス140が、制御フローグラフ120dに含まれるが、それは制御フローグラフ120aには含まれていなかつたものであるように、修正がなされる。制御フローグラフ120aと制御フローグラフ120dとの間の比較により示されるように、制御フローグラフリターンパス140は、制御フローグラフ120aのリターンノード132及び138の前に挿入される。そのような修正の具体例は、図1Eの修正部分118eには含まれ、図1Bの一部分118aには含まれていないステートメント129により示される。

30

【0034】

この特定の例における修正は、制御フローグラフ120aの制御フローグラフノード128, 130, 134, 136及び140に対応するコード部分のスキップをもたらす。説明される例では、制御フローグラフ120dにおける破線は、対応するノード及びエッジが制御フローグラフ120aには存在していたが、修正に起因して、制御フローグラフ120dには最早存在しないことを示す。以下に説明されるように、先行する制御フローグラフのリターンノードの前に制御フローグラフリターンパスを挿入するという特定のアンチパターンを充足することに起因して、制御フローグラフ120aから制御フローグラフ120dへの変更をもたらす修正は許容されなくてもよい。

40

【0035】

許容されない制御フロー変化に関する上記のアンチパターン109は、具体例として与えられているに過ぎず、制御フローの変化に関連する他のアンチパターン109も、本開示の範囲内に存在する。更に、アンチパターン109は相互に排他的ではなく、あるアンチパターンを満たす修正が、何らかの例における1つ以上の他のアンチパターンを充足してもよい。

50

【0036】

追加的又は代替的に、一実施形態において、1つ以上のアンチパターン109は、不適切な修正に起因するソフトウェアプログラムのデータフロー変更に基づいてもよいし、あるいは、それを含んでもよい。例えば、一実施形態において、1つ以上のアンチパターン109は、修正によって引き起こされるデータフロー分岐のディセーブルに基づいてもよいし、あるいは、それを含んでもよい。特に、一実施形態において、1つ以上のアンチパターン109は、対応する変数が後の使用時に未定であるような変数定義を削除することに基づいてもよいし、あるいは、それを含んでもよい。

【0037】

データフローグラフに関し、上述した変数定義の削除は、データフローノードの到来エッジ(incoming edge)が削除されるようなものであってもよい。従って、一実施形態において、1つ以上のアンチパターン109は、データフローノードの到来エッジの削除に基づいてもよいし、あるいは、それを含んでもよい。以下において更に説明されるように、データフローノードの到来エッジの削除をもたらす修正は、許容されなくてよい。

10

【0038】

具体例として、図1Fは、ステートメント152及びステートメント154を含む例示的なコード150を示す。ステートメント152は、ステートメント154で使用される変数「tmp」を定義する。対応するデータフローグラフは、ステートメント152における「tmp」の定義に対応する第1ノードと、ステートメント154における「tmp」の利用に対応する第2ノードとを含み、エッジは第1ノードから第2ノードを指す。第1ノードから第2ノードを指示するエッジは、第2ノードに対して「到来エッジ」と考えられる。ステートメント152の削除は、アンチパターンを満たすことになり、その理由は、「tmp」の定義を削除し、ステートメント154で使用される場合に「tmp」が定義されていないようになるからである。対応するデータフローグラフの観点からは、ステートメント152の削除は、第1ノードと、第1ノード及び第2ノードを接続するエッジとの削除となり、対応するデータフローグラフの到来エッジが削除される。従って、ステートメント152の削除は、データフローノードの到来エッジの削除という特定のアンチパターンを充足することになるので、許容されない。

20

【0039】

追加的又は代替的に、一実施形態において、1つ以上のアンチパターン109は、無限ループをもたらす修正に基づいてもよいし、あるいは、それを含んでもよい。特に、一実施形態において、1つ以上のアンチパターン109は、ループ内のループ変数の指定(assignment)ステートメントの削除に基づいてもよいし、あるいは、それを含んでもよく、ループのループ条件は、指定ステートメントに対応するループ変数に基づく。

30

【0040】

データフローグラフに関し、上記の指定ステートメントの削除は、対応するデータフローグラフのうちの特定のタイプのデータフローループが中断されるようになされる。特定のタイプのデータフローループは、第1データフローノード及び第2データフローノードを含むデータフローループであってもよい。第1データフローノードは、ソフトウェアプログラムのループのうちのループ条件ステートメントに対応し、第2データフローノードは、ソフトウェアプログラムのループ内にある指定ステートメントに対応してもよい。更に、第2データフローノードは、第1データフローノードに対応するループ条件ステートメントのループ変数に関連してもよい。以下において更に説明されるように、そのようなデータフローループの中止をもたらす修正は、特定のアンチパターンを充足することに起因して、許可されなくてもよい。

40

【0041】

具体例として、図1Fは、ステートメント162及びステートメント164を含む例示的なコード160を示す。コード160は「while」ループに対応する。ステートメント162は、ループ条件がループ変数「x」に基づいているループ条件ステートメントを含む。特に、「x」が「5」未満である限り、コード160の実行は、「while」ループに残る。ステートメント164は、ループ変数「x」をインクリメントする指定ステートメントを含み、ステートメント164

50

はコード160の「while」ループの中にあってよい。従って、ステートメント162に関連するループ条件の充足性は、対応する「while」ループの中にあるステートメント164により指定される「x」の値に基づく。

【0042】

対応するデータフローグラフは、ステートメント162において「x」を利用することに対応する第1ノードと、ステートメント164の「x」の指定に対応する第2ノードとを含むかもしれない。更に、対応するデータフローグラフは、第1ノードから第2ノードへの第1エッジと、第2ノードから第1ノードに戻る第2エッジとを含み、その理由は、ステートメント162及び164における「x」の関連する利用及び指定に起因する。

【0043】

ステートメント164の削除は、コード160の「while」ループを立ち去らない結果をもたらすので、アンチパターンを満たす。対応するデータフローグラフの観点からは、そのステートメントの削除は、対応するデータフローグラフの第2エッジの削除をもたらす。第2エッジの削除は、第1ノードと第2ノードとの間のデータフローを中断させる。従って、上記の特定のタイプのデータフローループの中止という特定のアンチパターンを充足することになるので、ステートメント164の削除は許容されない。

【0044】

テストスース111は、検査対象コード104に対するテストケースとして機能する1つ以上のルーチンを含んでよい。テストスースは、検査対象コード104が、指定された仕方で振る舞うか否かを決定するように構成される。テストスース111は、適切な任意の技術に従って構成されてよい。

【0045】

修復モジュール106は、検査対象コード104に対して1つ以上のテストスース111を適用し、検査対象コード104内の1つ以上の誤りを検出又は判定するように構成される。修復モジュール106は、修復テンプレート112に基づいて、並びに、検出された誤りに応じて及びそれに基づいて、検査対象コード104に対して1つ以上の修正を実行するように構成される。誤りの検出及び修正は、適切な任意の技術に基づいて実行されてよい。

【0046】

修復モジュール106は、1つ以上の修正が1つ以上のアンチパターン109を充足するか否かを判定するように構成されてよい。修復モジュール106は、1つ以上のアンチパターンを充足する修正を不許可にするように構成されてもよい。追加的又は代替的に、修復モジュール106は、修正が何れかのアンチパターン109を充足しない場合に、修正された検査対象コード104について更なる検査を実行するように構成されてもよい。一実施形態において、更なる検査は、1つ以上のテストスース111を利用して実行されてもよい。これら又は他の実施形態において、修復モジュール106は、検査対象コード104に対して1つ以上の修正を実現した後に、修正された検査対象コード104を、修正済み検査対象コード108として出力するように構成されてもよい。追加的又は代替的に、修正済み検査対象コード108は1つ以上の修正を含み、各々の修正は上記のアンチパターン109を充足しない。更に、修正は、修正済み検査対象コード108が出力される前に、テストスース111の1つ以上の他の検査に合格しているかもしれない。一実施形態において、修復モジュール106は、図3に関連して後述される方法300の1つ以上の動作に基づいて、検査対象コード104の修正を実行し、修正済み検査対象コード108を出力するように構成されてよい。

【0047】

従って、修復モジュール106は、検査対象コード104内の1つ以上の誤りを修復する1つ以上の修正とともに修正済み検査対象コード108を生成するように構成されてよい。修復モジュール106は、(例えば、修正が1つ以上のアンチパターン109を充足するか否かを判断すること等のような)静的な分析を利用して修正済み検査対象コード108を生成するように構成されてもよく、静的な分析は、所定のタイプの修正を除くために、テストスース111とともに実行時の分析(動的な分析と言及されてもよい)に先行して、修正に対して実行されてもよい。動的な分析の前のそのような静的な分析は、自動修復技術と比較した場合に修

10

20

30

40

50

復モジュール106の動作効率を改善することに加えて、修復システムが、より多くの場合に、誤りに対して適切な又は優れた修正を実行できるようにする。

【0048】

本開示の範囲から逸脱することなく、図1A-1Fに対して修正、追加又は省略がなされてもよい。例えば、環境100は、本開示で説明及び記述されるものより多い又は少ない要素を含んでもよい。更に、図1B-1Fに関して与えられるコード及びフローグラフの例は、単なる具体例に過ぎず、本開示の限定ではない。更に、一実施形態において、1つ以上のルーチン、1つ以上の命令、修復モジュール106のコードの少なくとも一部分、アンチパターン109、テストスース111、及び、修復テンプレート112は組み合わせられてよく、その組み合わせ方は、それらが同等であると考えられるように、あるいは、修復モジュール106、アンチパターン109、テストスース111、及び、修復テンプレート112のうちの2つ以上のパートと考えられてよい共通するセクションを有するようになされてよい。

10

【0049】

図2は、本開示の少なくとも1つの実施形態による例示的なコンピューティングシステム202のプロック図を示す。コンピューティングシステム202は、修復モジュール(例えば、修復モジュール106)に関連する1つ以上の動作を実現するように構成される。コンピューティングシステム202は、プロセッサ250、メモリ252、及び、データストレージ254を含んでよい。プロセッサ250、メモリ252、及び、データストレージ254は、通信可能に結合される。

20

【0050】

一般に、プロセッサ250は、適切な任意の専用又は汎用のコンピュータ、コンピューティングエンティティ、又は、様々なコンピュータハードウェア又はソフトウェアモジュールを含む処理装置を含んでよく、適用可能な任意のコンピュータ読み取り可能な記憶媒体に保存される命令を実行するように構成される。例えば、プロセッサ250は、マイクロプロセッサ、マイクロコントローラ、ディジタル信号プロセッサ(DSP)、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)、あるいは、プログラム命令を解釈及び/又は実行する及び/又はデータを処理するように構成される他の任意のディジタル又はアナログ回路を含んでよい。図2には単独のプロセッサとして示されているが、プロセッサ250は、本開示で説明される任意の数の処理を個別的に又はまとめて実行する又はそのパフォーマンスを管理するように構成される任意の数のプロセッサを含んでよい。更に、1つ以上のプロセッサは、例えば様々なサーバのような1つ以上の異なる電子装置に存在してもよい。

30

【0051】

一実施形態において、プロセッサ250は、メモリ252、データストレージ254、又は、メモリ252及びデータストレージ254に保存されるプログラム命令を解釈及び/又は実行する及び/又はそこに保存されるデータを処理する。一実施形態において、プロセッサ250は、データストレージ254からプログラム命令をフェッチし、プログラム命令をメモリ252にロードする。プログラム命令がメモリ252にロードされた後に、プロセッサ250はプログラム命令を実行してよい。

40

【0052】

例えば、一実施形態において、修復モジュールは、プログラム命令としてデータストレージ254に含まれてもよい。プロセッサ250は、データストレージ254から修復モジュールのプログラム命令をフェッチし、修復モジュールのプログラム命令をメモリ252にロードしてもよい。修復モジュールのプログラム命令がメモリ252にロードされた後に、プロセッサ250は、命令により指示されるように修復モジュールに関連する動作をコンピューティングシステムがするように、プログラム命令を実行する。

【0053】

メモリ252及びデータストレージ254は、コンピュータ実行可能な命令又はデータ構造を担う又はそこに保存するコンピュータ読み取り可能な記憶媒体を含んでよい。そのようなコンピュータ読み取り可能な記憶媒体は、プロセッサ250のような汎用又は専用のコンピ

50

ユータによりアクセスされることが可能な任意の利用可能な媒体を含んでよい。限定ではない一例として、そのようなコンピュータ読み取り可能な記憶媒体は、有形の又は非一時的なコンピュータ読み取り可能な記憶媒体を含んでよく、記憶媒体は、RAM、ROM、EEPROM、CD-ROM又は他の光ディスクストレージ、磁気ディスク又は他の磁気ストレージデバイス、フラッシュメモリデバイス(例えば、ソリッドステートメモリデバイス)、あるいは、他の任意のストレージデバイスを含んでよく、他の任意のストレージデバイスは、コンピュータ実行可能な命令又はデータ構造の形式で特定のプログラムコードを搬送又は保存するために使用され、かつ、汎用又は専用のコンピュータによりアクセスされてよい。上記のものの組み合わせも、コンピュータ読み取り可能な記憶媒体の範囲内に含まれてよい。コンピュータ実行可能な命令は、例えば、所定の動作又は一群の動作をプロセッサ250に実行させるように構成される命令及びデータを含む。

【0054】

本開示の範囲から逸脱することなく、コンピューティングシステム202に対して修正、付加又は省略がなされてよい。例えば、一実施形態において、コンピューティングシステム202は、明示的には図示も記述もされていない他のコンポーネントを幾つでも含んでよい。

【0055】

図3は、本開示で説明される少なくとも1つの実施形態によりソフトウェアプログラムを修復する方法例300のフローチャートを示す。方法300は、任意の適切なシステム、装置又はデバイスにより実行されてよい。例えば、図1の修復モジュール106又は(修復モジュールにより指図される)図2のコンピューティングシステム202は、方法300に関する1つ以上の動作を実行してよい。個別的なブロックで示されているが、方法300の1つ以上のブロックに関するステップ及び動作は、特定の実現手段に依存して、追加的なブロックに分割されてもよいし、より少数のブロックに統合されてもよいし、あるいは、削除されてもよい。

【0056】

方法300はブロック302から始まり、ソフトウェアプログラムの検査対象コードが受信される。検査対象コードは、図1Aに関連して説明された検査対象コード104に類似するものであってよい。追加的又は代替的に、一実施形態において、修復テンプレート(例えば、図1Aにおける修復テンプレート112)、1つ以上のアンチパターン(例えば、図1Aにおけるアンチパターン109)、及び、1つ以上のテストスース(例えば、図1Aにおけるテストスース)が、ブロック302において読み込まれてよい。

【0057】

ブロック304において、適切な任意の技術を利用して、検査対象コードに関して誤りの特定が実行される。一実施形態において、誤りの特定は、検査対象コードの中の1つ以上の誤りを特定し、及び、検査対象コードに対する1つ以上のテストスースの適用に基づいて実行されてもよい。

【0058】

ブロック306において、誤りの特定に応じて及び誤りに基づいて、1つ以上の修正が検査対象コードに対してなされてよい。修正は、修復テンプレートに基づいてよい。一実施形態において、修正を行う前に、検査対象コードに関し、1つ以上のプログラム表現が生成されてよい。プログラム表現は、フローグラフ(例えば、制御フローグラフ、データフローグラフ、又は、それら双方)、抽象構文ツリー(Abstract Syntax Tree)、あるいは、検査対象コード自体のコピーを含んでもよい。

【0059】

ブロック308において、修正が不適切であるか否かを示すアンチパターンのうちの1つ以上を修正が充足するか否かが判断されてよい。例えば、アンチパターンは、図1B-1Fに関して上述したようなソフトウェアプログラム制御フロー又はデータフローの変更に基づいてもよいし、あるいは、それを含んでもよい。

【0060】

10

20

30

40

50

一実施形態において、修正の前及び後に、検査対象コードに関して生成されるプログラム表現同士の比較に基づいて、修正が、1つ以上のアンチパターンを充足するか否かが、判定されてよい。

【0061】

例えば、上述したように、検査対象コードのコピー等のような、あるいは、検査対象コードに関する抽象構文ツリー等のような第1プログラム表現が、検査対象コードの修正の前に生成されてよい。更に、修正された検査対象コードのコピー等のような、あるいは、修正された検査対象コードに関する抽象構文ツリー等のような第2プログラム表現が、検査対象コードの修正の後に生成されてもよい。第1プログラム表現の第2プログラム表現に対する比較は、修正に対応し且つ1つ以上のアンチパターンを充足する構文上の変化を示す。

10

【0062】

例えば、上述したように、1つ以上のアンチパターンは構文変化に基づく又はそれを含んでよく、構文変化は：「リターン」ステートメントの削除；「エグジット」ステートメントの削除；正規の表現に合致する名称を有するメソッドコールを含むステートメントの削除；「アサート」ステートメントの削除；ある制御フローグラフノードに関する全てのステートメントの削除；経路条件の述語の否定の追加；条件の検査をディセーブルにする「真」の述語の追加；条件をディセーブルにする「偽」の述語の追加；制御フローブロックにおいて最後のステートメントでないリターンステートメントの制御フローブロックへの挿入；変数の定義であって、対応する変数が始めて使用される際に不定となってしまうものの削除；及び；ループのループ条件がループ変数に基づく場合における、ループ内のループ変数についての指定ステートメントの削除；等を含んでよい。一実施形態において、第1プログラム表現及び第2プログラム表現の間の静的な構文比較は、何れか任意の上記の構文上のアンチパターンを対応する修正が充足するか否かを判定するために、実行されてよい。

20

【0063】

これら又は他の実施形態において、プログラム表現としてフローグラフの生成を含む静的な分析に基づいて、修正が1つ以上のアンチパターンを充足するか否かを判定してもよい。上述したように、一実施形態において、アンチパターンは、図1B-1Fに関して上述したようにフローグラフ中の変化に基づいてもよいし、あるいは、それを含んでもよく、例えば：制御フローグラフのエグジットパスの削除；制御フローグラフのノードの削除；先行する制御フローグラフのリターンノードの前に制御フローグラフのリターンノードを挿入すること；データフローノードの到来エッジの削除；及び；データフローループの中斷；等を含んでもよく、データフローループは、第1データフローノード及び第2データフローノードを含み、第1データフローノードはソフトウェアプログラムのループのループ条件ステートメントに対応し、第2データフローノードは指定ステートメントに対応し、指定ステートメントは、ソフトウェアプログラムのループ内にあり、かつ、ループ条件ステートメントの変数に関連付けられる。

30

【0064】

一例として、一実施形態では、制御フローグラフ、データフローグラフ又はそれら双方は、上述した1つ以上の修正を実行する前に、制御対象コードに関連して生成されてよい。更に、制御フローグラフ、データフローグラフ又はそれら双方は、修正を実行した後に、検査対象コードに関して生成されてもよい。前後のフローグラフが互いに比較され、修正によって引き起こされるフローグラフ中の1つ以上の相違を特定してもよい。相違は、フローグラフに対応するアンチパターンに対して比較され、その相違が1つ以上のアンチパターンを充足するか否かを判定してよい。相違が修正に関連付けられているので、相違がアンチパターンを充足するか否かの判断は、修正がアンチパターンを充足するか否かを示す。

40

【0065】

フローグラフにおける変化に基づく又はそれを含むアンチパターンは、図1B-1Fに関し

50

て上述した対応する構文上の変化に基づいてもよい。従って、一実施形態において、構文上の分析(例えば、検査対象コードのコピー又は抽象構文ツリーを含むプログラム表現の生成)又はフローグラフ分析(例えば、制御フロー又はデータフローのグラフを含むプログラム表現の生成)は、不適切であるかもしれない同様な修正を検出するために使用されてよい。

【0066】

方法300は、1つ以上のアンチパターンを充足する修正に関し、ブロック308からブロック310に進む。方法300は、何れのアンチパターンも充足しない修正に関し、ブロック308からブロック312に進む。

【0067】

上述したように、アンチパターンを充足することは、対応する修正が不適切であるかもしれないことを示す。従って、ブロック310では、1つ以上のアンチパターンを充足する修正は不許可にされてもよい(ブロック310)。修正を不許可にすることは、修正を削除する、あるいは、修正を実行しないこととし、検査対象コードが修正前のかつての姿になる。ブロック310に続いて、一実施形態では、方法300はブロック304に戻る。追加的又は代替的に、図3では明示的には示されていないが、ブロック310に続いて、方法300はブロック306に戻ってもよく、不許可にされた修正に対応する誤りに関して異なる修正が施されてもよい。

【0068】

上述したように、特定の修正が何れのアンチパターンも充足しないことは、その特定の修正が、正しい修復に対応する確実性を提供する。従って、何れのアンチパターンも充足しない修正に関し、更にダイナミックな検査が検査対象コードに実行されてよい。例えば、ブロック312において、ネガティブ検査(negative test)(例えば、誤りの存在に起因して、検査対象コード104において誤りとなっていたテストケース)が実行され、実行された修正が、対応する誤りを修復したか否かを判定してもよい。

【0069】

ブロック314において、ネガティブ検査に合格したか否かが判定される。例えば、修正は当該修正の対応する誤りを修復したか否かが判定される。ネガティブ検査に合格した旨の判断に応じて、方法300はブロック314からブロック316に進む。ネガティブ検査に合格しなかった旨の判断に応じて、方法300はブロック314からブロック310に進む。

【0070】

これら又は他の実施形態において、更なる検査の別の例として、ブロック316においてポジティブ検査(positive test)が実行され、その修正が、ブロック308に関連するアンチパターンには対応しない別の誤りを導入したか否かを判定してもよい。ブロック318において、ポジティブ検査(例えば、誤りの存在にもかかわらず、検査対象コード104に合格していたテストケース)が、適用される修正の下で合格し続けるか否かが判定されてもよい。

【0071】

ポジティブ検査は合格した旨の判定に応じて、方法300はブロック318からブロック320に進む。ポジティブ検査は合格しなかった旨の判定に応じて、方法300はブロック318からブロック310に進む。

【0072】

ブロック320において、修正済み検査対象コードが出力される。修正済み検査対象コードは、アンチパターンを充足せず且つポジティブ及びネガティブ検査の双方に合格する修正を含み、修正済み検査対象コードは、ブロック302で受信された検査対象コードの修復されたバージョンを含む。

【0073】

方法300は、ソフトウェアプログラムの検査及び修復の効率を改善する。例えば、ネガティブ又はポジティブ検査を実行する前に、修正がアンチパターンを充足するか否かを判断することは、ポジティブ又はネガティブ検査が実行される修正の量を削減する。本方法

10

20

30

40

50

は、多くの場合に、ソフトウェアプログラムの開発者にとって受け入れられる適切な又は優れた修復をもたらす。

【0074】

本開示の範囲から逸脱することなく、方法300に対して修正、付加又は省略がなされてよい。例えば、方法300の動作は異なる順序で実行されてもよい。追加的又は代替的に、2つ以上の動作が同時に実行されてもよい。特に、ポジティブ及びネガティブ検査を実行する順序は入れ替わってもよい。追加的又は代替的に、ブロック314及び318に関連する動作は、入れ替えられてもよいし、あるいは、組み合わせられてもよい。更に、説明される処理及び動作は具体例として提供されているに過ぎず、開示される実施形態の本質から逸脱することなく、処理及び動作のうちの幾つかは、選択的であってもよいし、より少ない処理及び動作に統合されてもよいし、あるいは、追加的な処理及び動作に拡張されてもよい。10

【0075】

更に、一実施形態において、方法300は反復的に実行されてもよく、反復の中で一度に一つの修正が分析されてもよい。追加的又は代替的に、1つ以上のブロックに関連する1つ以上の動作は、一度に複数の修正に関して実行されてもよい。

【0076】

上述したように、本開示で説明される実施形態は、本願で言及されるような様々なコンピュータハードウェア又はソフトウェアモジュールを含む専用又は汎用のコンピュータ(例えば、図2のプロセッサ250)を利用するすることを含んでよい。更に、上述したように、本開示で説明される実施形態は、コンピュータ読み取り可能な媒体(例えば、図2のメモリ252)を利用して実現されてもよく、媒体は、そこに保存されるコンピュータ実行可能な命令又はデータ構造を搬送する又は有する。20

【0077】

本開示で使用されるように、「モジュール」又は「コンポーネント」という用語は、モジュール又はコンポーネント及び/又はソフトウェアオブジェクト又はソフトウェアルーチンの動作を実行するように構成される特定のハードウェア実現手段を指し、そのモジュール等はコンピューティングシステムの汎用ハードウェア(例えば、コンピュータ読み取り可能な媒体、処理デバイス等)により保存され及び/又は実行されてよい。一実施形態において、本開示で説明される様々なコンポーネント、モジュール、エンジン及びサービスは、コンピューティングシステム上で実行されるオブジェクト又はプロセスとして(例えば、個々のスレッドとして)実現されてもよい。本開示で説明される或る種のシステム及び方法は、(汎用ハードウェアにより保存及び/又は実行される)ソフトウェアで実現されるように一般的に説明されているが、特定のハードウェア実現手段、あるいは、ソフトウェア及びハードウェア実現手段の組み合わせも可能であり、本願で想定されている。本説明において、「コンピューティングエンティティ」は、本開示で上記に規定されるような任意のコンピューティングシステム、任意のモジュール、あるいは、コンピューティングシステムで動作するモジュールの任意の組み合わせであってもよい。30

【0078】

本開示において、特に添付の特許請求の範囲において使用される用語(例えば、添付の特許請求の範囲におけるボディ)は、一般に「非限定的(open)」用語として意図されている(例えば、「含む」という用語は、「含んでいるが、それに限定されない」ように解釈されるべきであり、「有する」という用語は、「少なくとも有する」ように解釈されるべきであり、「包含する」という用語は、「包含しているが、それに限定されない」ように解釈されるべきである、等々)。40

【0079】

更に、対象の請求項の記述のうちで具体的な数が意図される場合、そのような意図は請求項の中で明示的に記述され、そのような記述が無い場合、そのような意図は存在しない。例えば、理解を促す一例として、添付の特許請求の範囲は、請求項の記述を行うために「少なくとも1つ」及び「1つ以上」という導入語句の利用を含むかもしれない。しかしながら

50

がら、そのような語句の利用は、「或る」(「a」又は「an」)のような不定冠詞的な請求項の記述の使用が、そのように言及される請求項の記述を含む何らかの特定の請求項を、そのような記述のみを含む実施形態に限定することを意図するように解釈されるべきでなく、たとえ同じ請求項が「1つ以上」又は「少なくとも1つ」及び「或る」のような不定冠詞的な用語を含む場合でさえ、限定を意図するように解釈されるべきでない(例えば、「或る」は「少なくとも1つ」又は「1つ以上」を意味するように解釈されるべきである)。同様なことは、請求項の記述を導入するために使用される定冠詞的な用語の利用についても成り立つ。

【0080】

更に、導入される請求項の記述のうち特定の数が明示的に言及されたとしても、そのような記述は、少なくとも、言及された数を意味するように解釈されるべきであることを当業者は認めるであろう(例えば、他の修飾語句を伴わない「2つの記述事項」の記述事項自体は、少なくとも2つの記述事項、或いは、2つ又はそれより多い記述事項を意味する)。更に、「A, B, 及びC等のうちの少なくとも1つ」あるいは「A, B, 及びC等のうちの1つ以上」に類する言い回しが使用される場合、一般に、その構文は、Aのみ、Bのみ、Cのみ、A及びB双方、A及びC双方、B及びC双方、あるいは、A, B, C全て等を含むように意図されている。

【0081】

更に、2つ以上の代替的な用語を提示する何らかの離接語又は語句は、明細書、特許請求の範囲又は図面によらず、用語のうちの1つ、何れかの用語、又は、双方の用語を含む可能性を想定しているように理解されるべきである。例えば、「A」又は「B」という語句は、「A」又は「B」又は「A及びB」の可能性を含むように理解されるべきである。

【0082】

本開示で記述される全ての具体例及び条件的な言葉は、技術進歩に発明者等が貢献した本開示及び概念の読者による理解を促す教育的な目的のために意図されており、そのように具体的に記述された具体例や条件に限定することなく本願を解釈すべきである。以上、本開示の実施形態が詳細に説明されてきたが、本開示の精神及び範囲から逸脱することなく、様々な変形、置換及び代替がなされてよい。

以上の実施の形態に関し、更に以下の付記を開示する。

(付記1)

ソフトウェアプログラムに関する第1プログラム表現を生成する工程；
テストツールを利用して前記ソフトウェアプログラムにおける誤りを特定する工程；
前記誤りの特定に応じて、前記ソフトウェアプログラムを、修復テンプレートを利用して修正する工程；

前記ソフトウェアプログラムを修正した後に、前記ソフトウェアプログラムに関する第2プログラム表現を生成する工程；

前記第1プログラム表現と前記第2プログラム表現との比較に基づいて、前記第1プログラム表現に対する前記第2プログラム表現の相違を検出する工程；

修正が不適切であるか否かを示すアンチパターン条件を、前記相違が充足することを判定する工程；及び

前記相違が前記アンチパターン条件を充足することに応じて、前記ソフトウェアプログラムの修正を不許可にする工程；

を有する方法。

(付記2)

前記第1プログラム表現は第1制御フローグラフを含み、前記第2プログラム表現は第2制御フローグラフを含む、付記1に記載の方法。

(付記3)

前記アンチパターン条件は、制御フローグラフのエグジットパスの削除を含む、付記2に記載の方法。

(付記4)

10

20

30

40

50

前記アンチパターン条件は、制御フローグラフのノードの削除を含む、付記2に記載の方法。

(付記5)

前記アンチパターン条件は、先行する制御フローグラフのリターンノードの前に、制御フローグラフのリターンパスを挿入することを含む、付記2に記載の方法。

(付記6)

前記第1プログラム表現は第1データフローグラフを含み、前記第2プログラム表現は第2データフローグラフを含む、付記1に記載の方法。

(付記7)

前記アンチパターン条件は、データフローノードの到来エッジの削除を含む、付記6に記載の方法。

10

(付記8)

前記アンチパターン条件は、第1データフローノード及び第2データフローノードを含むデータフローノードの中断を含み、前記第1データフローノードは前記ソフトウェアプログラムのループのループ条件ステートメントに対応し、前記第2データフローノードは、前記ソフトウェアプログラムのループ内に存在し且つ前記ループ条件ステートメントの変数に関連する指定ステートメントに対応する、付記6に記載の方法。

(付記9)

前記アンチパターン条件は：「リターン」ステートメントの削除；「エグジット」ステートメントの削除；正規の表現に合致する名称を有するメソッドコールを含むステートメントの削除；「アサート」ステートメントの削除；制御フローグラフノードに関する全てのステートメントの削除；経路の条件の述語の否定の追加；条件の検査をディセーブルにする「真」の述語の追加；条件をディセーブルにする「偽」の述語の追加；制御フローブロックにおいて最後のステートメントでないリターンステートメントの制御フローブロックへの挿入；変数の定義であって対応する変数が始めて使用される際に不定となってしまうものの削除；及び；ループのループ条件がループ変数に基づく場合における、ループ内のループ変数についての指定ステートメントの削除；のうちの1つ以上に基づく、付記1に記載の方法。

20

(付記10)

前記第1プログラム表現は、修正前の前記ソフトウェアプログラムの少なくとも一部分のコピー；及び、修正前の前記ソフトウェアプログラムの少なくとも一部分の抽象構文ツリー；のうちの1つ以上を含み、並びに、

30

前記第2プログラム表現は、修正後の前記ソフトウェアプログラムの少なくとも一部分のコピー；及び、修正後の前記ソフトウェアプログラムの少なくとも一部分の抽象構文ツリー；のうちの1つ以上を含む、付記1に記載の方法。

(付記11)

コンピュータ実行可能な命令を含むコンピュータ読み取り可能な記憶媒体であって、前記命令は、1つ以上のプロセッサによる実行に応答して、システムに動作を実行させるよう構成され、前記動作は：

テストスツールを利用してソフトウェアプログラムにおける誤りを特定する工程；

40

前記誤りの特定に応じて、前記ソフトウェアプログラムを、修復テンプレートを利用して修正する工程；

修正が不適切であるか否かを示すアンチパターン条件を、前記修正が充足するか否かを判定する工程；及び

前記修正が前記アンチパターン条件を充足することに応じて、前記修正を不許可にする、或いは、前記修正が前記アンチパターン条件を充足しないことに応じて、修正された前記ソフトウェアプログラムについて更なる検査を実行する工程；

を含む、コンピュータ読み取り可能な記憶媒体。

(付記12)

前記アンチパターン条件は：「リターン」ステートメントの削除；「エグジット」ステ

50

ートメントの削除；正規の表現に合致する名称を有するメソッドコールを含むステートメントの削除；「アサート」ステートメントの削除；制御フローグラフノードに関する全てのステートメントの削除；経路の条件の述語の否定の追加；条件の検査をディセーブルにする「真」の述語の追加；条件をディセーブルにする「偽」の述語の追加；制御フローブロックにおいて最後のステートメントでないリターンステートメントの制御フローブロックへの挿入；変数の定義であって対応する変数が始めて使用される際に不定となってしまうものの削除；及び；ループのループ条件がループ変数に基づく場合における、ループ内のループ変数についての指定ステートメントの削除；のうちの1つ以上に基づく、付記11に記載のコンピュータ読み取り可能な記憶媒体。

(付記13)

10

前記ソフトウェアプログラムの修正がアンチパターン条件を充足するか否かの判定は：

前記ソフトウェアプログラムを修正する前に、前記ソフトウェアに関する第1フローグラフを生成すること；

前記ソフトウェアプログラムを修正した後に、前記ソフトウェアに関する第2フローグラフを生成すること；

前記第1フローグラフと前記第2フローグラフとの比較に基づいて、前記第1フローグラフに対する前記第2フローグラフの相違を検出すること；及び

前記相違が前記アンチパターン条件を充足することを確認すること；

を含む、付記11に記載のコンピュータ読み取り可能な記憶媒体。

(付記14)

20

前記第1フローグラフは第1制御フローグラフを含み、前記第2フローグラフは第2制御フローグラフを含む、付記13に記載のコンピュータ読み取り可能な記憶媒体。

(付記15)

前記アンチパターン条件は：制御フローグラフのエグジットパスの削除；制御フローグラフのノードの削除；及び；先行する制御フローグラフのリターンノードの前に、制御フローグラフのリターンパスを挿入すること；のうちの1つ以上を含む、付記14に記載のコンピュータ読み取り可能な記憶媒体。

(付記16)

前記第1フローグラフは第1データフローグラフを含み、前記第2フローグラフは第2データフローグラフを含む、付記13に記載のコンピュータ読み取り可能な記憶媒体。

30

(付記17)

前記アンチパターン条件は：データフローノードの到来エッジの削除；及び；第1データフローノード及び第2データフローノードを含むデータフローノードの中斷；のうちの1つ以上を含み、前記第1データフローノードは前記ソフトウェアプログラムのループのループ条件ステートメントに対応し、前記第2データフローノードは、前記ソフトウェアプログラムのループ内に存在し且つ前記ループ条件ステートメントの変数に関連する指定ステートメントに対応する、付記16に記載のコンピュータ読み取り可能な記憶媒体。

(付記18)

1つ以上のプロセッサ；及び

前記1つ以上のプロセッサに通信可能に結合される1つ以上のコンピュータ読み取り可能な記憶媒体；

40

を有するシステムであって、前記記憶媒体はコンピュータ実行可能な命令を含み、前記命令は、前記1つ以上のプロセッサによる実行に応答して、前記システムに動作を実行させ、前記動作は：

ソフトウェアプログラムに関する第1プログラム表現を生成する工程；

テストスーストを利用して前記ソフトウェアプログラムにおける誤りを特定する工程；

前記誤りの特定に応じて、前記ソフトウェアプログラムを、修復テンプレートを利用して修正する工程；

前記ソフトウェアプログラムを修正した後に、前記ソフトウェアプログラムに関する第2プログラム表現を生成する工程；

50

前記第1プログラム表現と前記第2プログラム表現との比較に基づいて、前記第1プログラム表現に対する前記第2プログラム表現の相違を検出する工程；

前記修正が不適切であるか否かを示すアンチパターン条件を、前記相違が充足するか否かを判定する工程であって、前記アンチパターン条件は、「リターン」ステートメントの削除；「エグジット」ステートメントの削除；正規の表現に合致する名称を有するメソッドコールを含むステートメントの削除；「アサート」ステートメントの削除；制御フローグラフノードに関する全てのステートメントの削除；経路の条件の述語の否定の追加；条件の検査をディセーブルにする「真」の述語の追加；条件をディセーブルにする「偽」の述語の追加；制御フローブロックにおいて最後のステートメントでないリターンステートメントの制御フローブロックへの挿入；変数の定義であって対応する変数が始めて使用される際に不定となってしまうものの削除；及び；ループのループ条件がループ変数に基づく場合における、ループ内のループ変数についての指定ステートメントの削除；のうちの1つ以上に基づく、工程；及び

前記修正が前記アンチパターン条件を充足することに応じて、前記修正を不許可にする、或いは、前記修正が前記アンチパターン条件を充足しないことに応じて、修正された前記ソフトウェアプログラムについて更なる検査を実行する工程；

を含む、システム。

(付記19)

前記第1プログラム表現は、前記ソフトウェアプログラムの少なくとも一部分の第1コピー；前記ソフトウェアプログラムの少なくとも一部分の第1抽象構文ツリー；第1制御フローグラフ；及び；第1データフローグラフのうちの1つ以上を含み、並びに、

前記第2プログラム表現は、前記ソフトウェアプログラムの少なくとも一部分の第2コピー；前記ソフトウェアプログラムの少なくとも一部分の第2抽象構文ツリー；第2制御フローグラフ；及び；第2データフローグラフのうちの1つ以上を含む、付記18に記載のシステム。

(付記20)

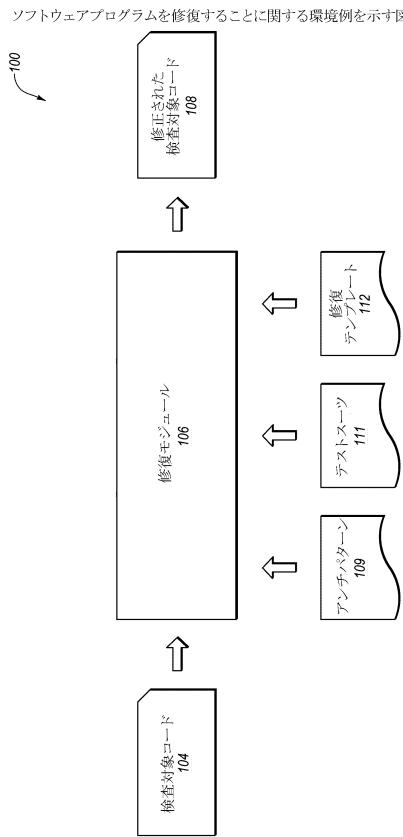
前記アンチパターン条件は：制御フローグラフのエグジットパスの削除；制御フローグラフのノードの削除；先行する制御フローグラフのリターンノードの前に、制御フローグラフのリターンパスを挿入すること；データフローノードの到来エッジの削除；第1データフローノード及び第2データフローノードを含むデータフローノードの中斷；のうちの1つ以上を含み、前記第1データフローノードは、前記ソフトウェアプログラムのループのループ条件ステートメントに対応し、前記第2データフローノードは、前記ソフトウェアプログラムのループ内に存在し且つ前記ループ条件ステートメントの変数に関連する指定ステートメントに対応する、付記18に記載のシステム。

10

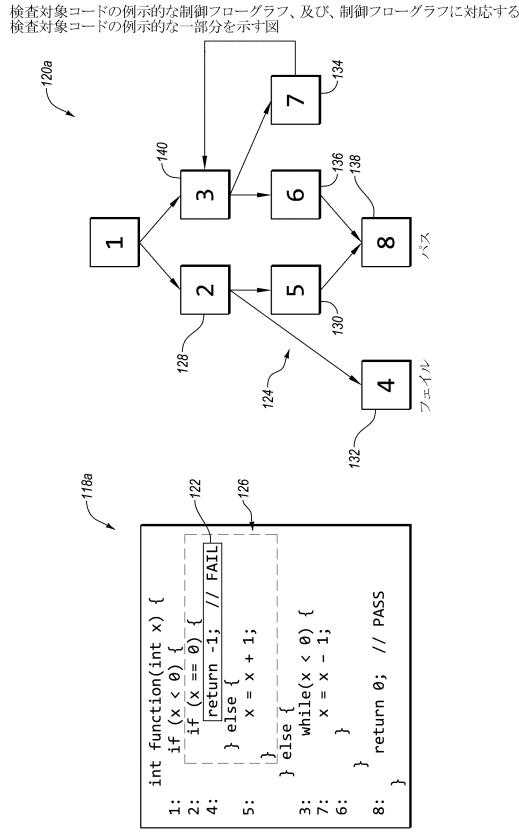
20

30

【図 1 A】

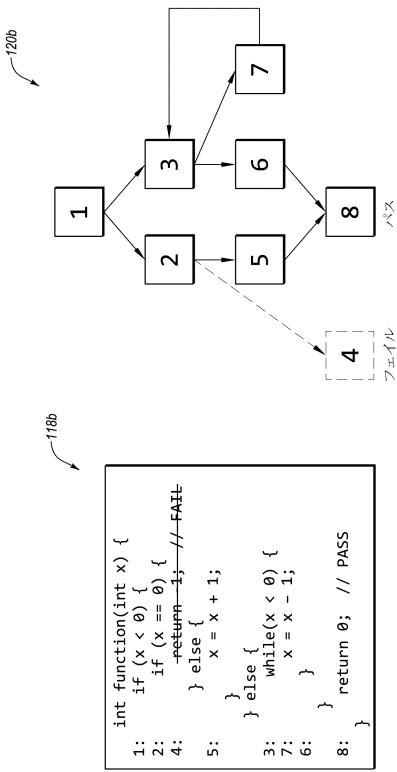


【図 1 B】



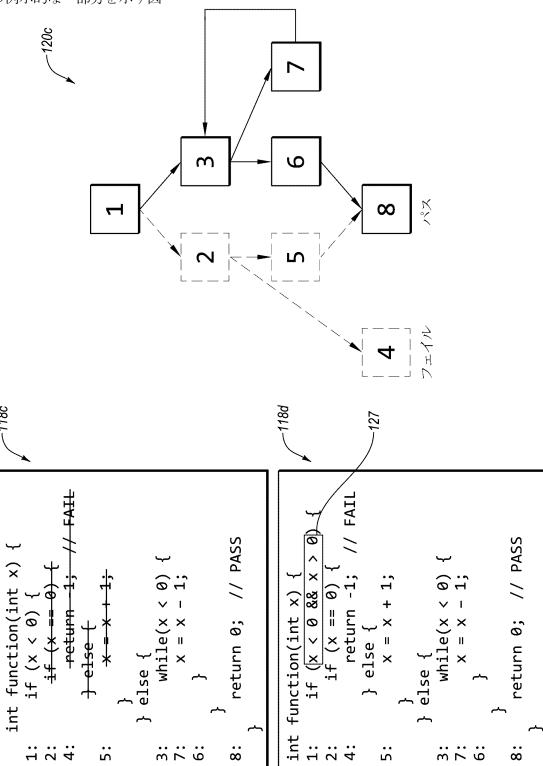
【図 1 C】

修正された検査対象コードについての修正された例示的な制御フローラフ、及び、修正された制御フローラフに対応する修正された検査対象コードについての修正された例示的な一部分を示す図



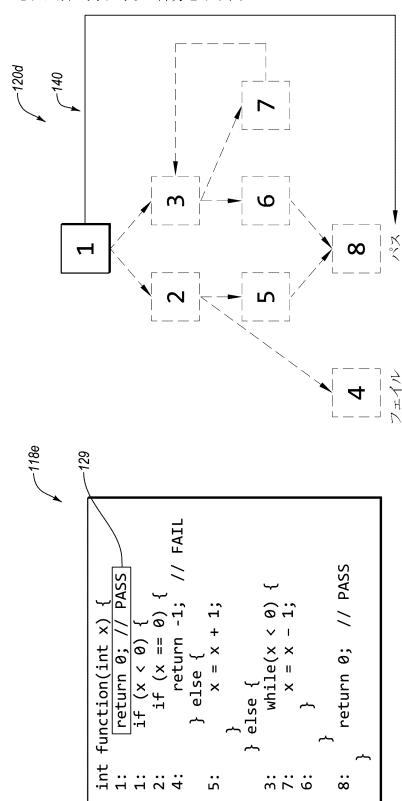
【図 1 D】

修正された検査対象コードについての修正された別の例示的な制御フローラフ、及び、修正された制御フローラフに対応する修正された検査対象コードについての修正された別の例示的な一部分を示す図



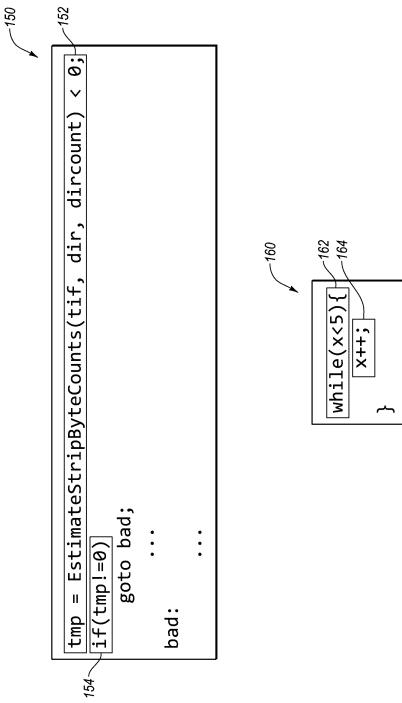
【図1E】

修正された検査対象コードについての修正された別の例示的な制御フローラフ、及び、修正された制御フローラフに対応する修正された検査対象コードについての修正された別の例示的な一部分を示す図



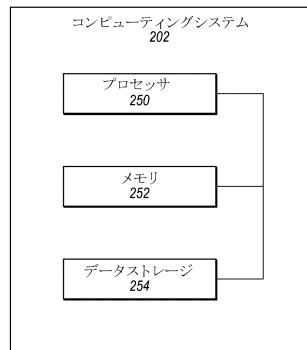
【図1F】

実行されるデータフローの修正を有するコード例を示す図



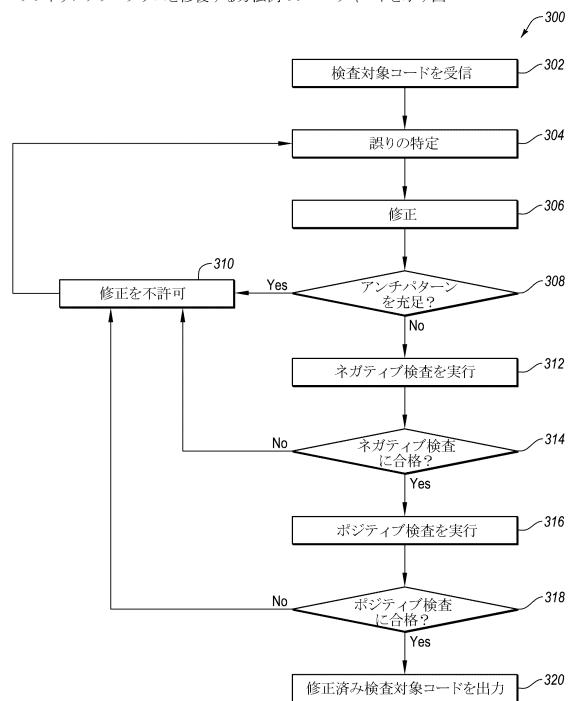
【図2】

ソフトウェアプログラムを修復するように構成されるコンピューティングシステム例を示す図



【図3】

ソフトウェアプログラムを修復する方法例のフローチャートを示す図



フロントページの続き

(72)発明者 プラサド・ムクル アール

アメリカ合衆国, カリフォルニア州 94085, サニーヴェイル, イースト アークス アヴェ
ニュ 1240番 フジツウ ラボラトリーズ アメリカ内

(72)発明者 タン・シン フェイ

アメリカ合衆国, カリフォルニア州 94085, サニーヴェイル, イースト アークス アヴェ
ニュ 1240番 フジツウ ラボラトリーズ アメリカ内

審査官 福西 章人

(56)参考文献 特開2008-250914 (JP, A)

特開2014-089700 (JP, A)

中国特許出願公開第101866317 (CN, A)

特開平04-270418 (JP, A)

特開平11-296366 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G 06 F 11/07

G 06 F 11/28 - 11/36