

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2011-248887

(P2011-248887A)

(43) 公開日 平成23年12月8日(2011.12.8)

(51) Int.Cl. F I テーマコード (参考)
G06F 11/28 (2006.01) G06F 11/28 340A 5B042

審査請求 未請求 請求項の数 15 O L (全 37 頁)

<p>(21) 出願番号 特願2011-115090 (P2011-115090)</p> <p>(22) 出願日 平成23年5月23日 (2011. 5. 23)</p> <p>(31) 優先権主張番号 12/785, 756</p> <p>(32) 優先日 平成22年5月24日 (2010. 5. 24)</p> <p>(33) 優先権主張国 米国 (US)</p> <p>(特許庁注：以下のものは登録商標)</p> <p>1. J A V A</p> <p>2. イーサネット</p> <p>3. G S M</p>	<p>(71) 出願人 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</p> <p>(74) 代理人 100070150 弁理士 伊東 忠彦</p> <p>(74) 代理人 100146776 弁理士 山口 昭則</p> <p>(72) 発明者 ゴーシュ・インドラディープ アメリカ合衆国, カリフォルニア州 95129, サン・ノゼ, レデラー・サークル1672番</p> <p>Fターム(参考) 5B042 HH14 HH17 HH42 HH49</p>
--	--

(54) 【発明の名称】 ソフトウェアモジュールのテスト方法及びシステム

(57) 【要約】

【課題】

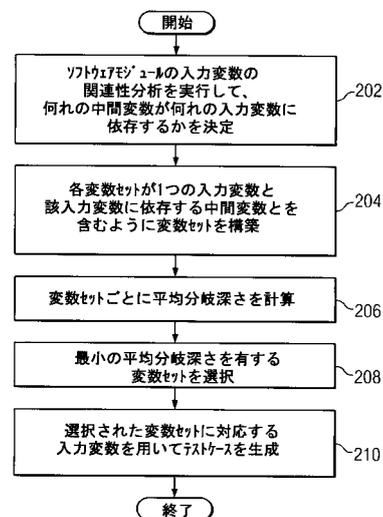
ソフトウェアモジュールのテスト又は検証を支援し得る方法及びシステムを提供する。

【解決手段】

一実施形態において、ソフトウェアモジュールの1つ以上の入力変数に対応する1つ以上の変数セットが構築される。各変数セットに関して、該変数セットによって影響される平均分岐深さが計算され、前記1つ以上の変数セットのうち最小の平均分岐深さを有する第1変数セットが選択される。第1変数セットの入力変数に第1記号値を割り当ててソフトウェアモジュールが記号的に実行され、第1変数セットに対する第1テストケースセットが生成される。必要に応じて、冗長テストケースを除去し、且つ第1テストケースセットのうち、全ての条件付き分岐点をカバーする最小のサブセットを選択することにより、第1テストケースセットが圧縮される。第1テストケースセットを用いてソフトウェアモジュールが検証される。

【選択図】 図2

ソフトウェアモジュールの入力変数を記号的に実行する順序を決定する方法例を示す図



【特許請求の範囲】

【請求項 1】

1つ以上の計算装置によって実行される方法であって、
ソフトウェアモジュールにアクセスするステップであり、
前記ソフトウェアモジュールは、

1つ以上の入力変数と、

1つ以上の中間変数であり、各中間変数の値が前記入力変数のうちの少なくとも1つの入力変数の値に直接的あるいは間接的に依存する、1つ以上の中間変数と、

1つ以上の条件付き分岐点であり、各条件付き分岐点が、前記入力変数及び前記中間変数のうちの少なくとも1つの入力変数及び/又は中間変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも1つの入力変数及び/又は中間変数の値に依存する、1つ以上の条件付き分岐点と、を有し、且つ

10

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進む、アクセスするステップと、

前記1つ以上の入力変数に対応する1つ以上の変数セットを構築するステップであり、各変数セットが、前記入力変数のうちの1つと、前記入力変数のうちの該1つの値に直接的あるいは間接的に依存する値を有する中間変数とを含む、構築するステップと、

各変数セットに関して、該変数セットの前記入力変数及び前記中間変数によって影響される平均分岐深さを計算するステップと、

20

前記1つ以上の変数セットのうち最小の平均分岐深さを有する第1変数セットを選択するステップと、

前記第1変数セットの前記入力変数に第1記号値を割り当てるステップと、

前記第1記号値を有する前記第1変数セットの前記入力変数を用いて、前記ソフトウェアモジュールを記号的に実行し、前記第1変数セットの前記入力変数及び前記中間変数に関する第1テストケースセットを生成するステップと、

前記第1テストケースセットを用いて前記ソフトウェアモジュールを検証するステップと、

を有する方法。

【請求項 2】

30

前記入力変数の各々に関して、前記中間変数のうちの何れが該入力変数の値に直接的あるいは間接的に依存する値を有するかを、関連性分析を用いて決定するステップ、を更に有する請求項1に記載の方法。

【請求項 3】

前記ソフトウェアモジュールは1以上の分岐階層を有し、

前記分岐階層の各々は、前記条件付き分岐点のうちの少なくとも1つを有し、

各変数セットに関して、該変数セットの前記入力変数及び前記中間変数によって影響される前記平均分岐深さを計算する前記ステップは、

該変数セットの前記入力変数に関して、分岐条件が該入力変数に依存する条件付き分岐点を有する分岐階層の数を計算するステップと、

40

該変数セットの前記中間変数の各々に関して、分岐条件が該中間変数に依存する条件付き分岐点を有する分岐階層の数を計算するステップと、

これらの分岐階層の数の平均を、該変数セットの前記平均分岐深さとして計算するステップと

を有する、

請求項1又は2に記載の方法。

【請求項 4】

最小の平均分岐深さを有する前記第1変数セットを選択するとき、最小の平均分岐深さを有する変数セットが複数存在する場合、前記第1変数セットとして、最小の平均分岐深さを有する変数セットのうちの1つをランダムに選択する、請求項1乃至3の何れか一項

50

に記載の方法。

【請求項 5】

前記 1 つ以上の変数セットのうち、次に最小の平均分岐深さを有する第 2 変数セットを選択するステップと、

前記第 2 変数セットの前記入力変数に第 2 記号値を割り当てるステップと、

前記第 2 記号値を有する前記第 2 変数セットの前記入力変数を用いて、前記ソフトウェアモジュールを記号的に実行し、前記第 2 変数セットの前記入力変数及び前記中間変数に関する第 2 テストケースセットを生成するステップと、

前記第 2 テストケースセットを用いて前記ソフトウェアモジュールを検証するステップと、

を更に有する請求項 1 乃至 4 の何れか一項に記載の方法。

【請求項 6】

1 つ以上の計算装置によって実行される方法であって、

ソフトウェアモジュール及び該ソフトウェアモジュール用の複数のテストケースにアクセスするステップであり、

前記ソフトウェアモジュールは、

1 つ以上の変数と、

1 つ以上の条件付き分岐点であり、各条件付き分岐点が、前記変数のうちの少なくとも 1 つの変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも 1 つの変数の値に依存する、1 つ以上の条件付き分岐点と、を有し、

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進み、

前記テストケースの各々は、前記条件付き分岐点のうちの少なくとも 1 つをカバーし、且つ前記条件付き分岐点のうちの前記少なくとも 1 つの分岐条件を規定する前記少なくとも 1 つの変数に関する少なくとも 1 つのテスト値を有する、アクセスするステップと、

前記条件付き分岐点の各々に関して、前記複数のテストケースのうちに、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースが複数存在する場合に、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースのうち 1 つを除く全てを、前記複数のテストケースから除去するステップと、

残りのテストケースから 1 つ以上の第 1 テストケースを選択するステップであり、各第 1 テストケースは、前記条件付き分岐点のうちの少なくとも 1 つが該第 1 テストケースによってのみカバーされるテストケースである、選択するステップと、

前記条件付き分岐点の全てが前記第 1 テストケース及び第 2 テストケースのうちの少なくとも 1 つのテストケースによってカバーされるまで、残りのテストケースから 1 つ以上の第 2 テストケースを繰り返し選択するステップであり、各繰り返しにおいて選択される第 2 テストケースは、該繰り返しにおいて最も多数の条件付き分岐点をカバーするテストケースである、選択するステップと、

選択された前記第 1 テストケース及び前記第 2 テストケースのみを用いて、前記ソフトウェアモジュールをテストするステップと、

を有する方法。

【請求項 7】

前記ソフトウェアモジュールの前記変数は、1 つ以上の入力変数と 1 つ以上の中間変数とを有し、各中間変数の値は、前記入力変数のうちの少なくとも 1 つの入力変数の値に直接的あるいは間接的に依存する、請求項 6 に記載の方法。

【請求項 8】

前記入力変数と前記入力変数に関する前記テストケースからの前記テスト値とを表すグラフを構築するステップを更に有し、

前記グラフは、

10

20

30

40

50

各階層が前記入力変数のうちの1つに関連付けられた、前記1つ以上の入力変数に対応する1つ以上の階層と、

前記階層の各々の1つ以上のノードと、

1つ以上のエッジであり、各エッジが、隣接する2つの階層からの2つのノードを結び付け、且つ前記隣接する2つの階層のうちの高い側の階層に関連付けられた入力変数の前記テスト値のうちの1つを表す、1つ以上のエッジと、

を有する、

請求項7に記載の方法。

【請求項9】

前記テストケースと前記条件付き分岐点を表す二部グラフを構築するステップを更に有し、

10

前記二部グラフは、

各第1ノードが前記テストケースのうちの1つを表す、前記複数のテストケースに対応する複数の第1ノードと、

各第2ノードが前記条件付き分岐点のうちの1つを表す、前記1つ以上の条件付き分岐点に対応する1つ以上の第2ノードと、

各エッジが前記第1ノードのうちの1つと前記第2ノードのうちの1つとを結び付ける複数のエッジであり、前記第1ノードのうちの該1つによって表されるテストケースが、前記第2ノードのうちの該1つによって表される条件付き分岐点をカバーする、複数のエッジと

20

を有する、

請求項6乃至8の何れか一項に記載の方法。

【請求項10】

1つ以上のプロセッサによって実行可能な命令を有するメモリと、

前記メモリに結合され、前記命令を実行するよう動作する1つ以上のプロセッサとを有するシステムであって、

前記1つ以上のプロセッサは、前記命令を実行するとき、

- ソフトウェアモジュールにアクセスし、

前記ソフトウェアモジュールは、

1つ以上の入力変数と、

30

1つ以上の中間変数であり、各中間変数の値が前記入力変数のうちの少なくとも1つの入力変数の値に直接的あるいは間接的に依存する、1つ以上の中間変数と、

1つ以上の条件付き分岐点であり、各条件付き分岐点が、前記入力変数及び前記中間変数のうちの少なくとも1つの入力変数及び/又は中間変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも1つの入力変数及び/又は中間変数の値に依存する、1つ以上の条件付き分岐点と、を有し、且つ

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進み、

- 前記1つ以上の入力変数に対応する1つ以上の変数セットを構築し、各変数セットは、前記入力変数のうちの1つと、前記入力変数のうちの該1つの値に直接的あるいは間接的に依存する値を有する中間変数とを含み、

40

- 各変数セットに関して、該変数セットの前記入力変数及び前記中間変数によって影響される平均分岐深さを計算し、

- 前記1つ以上の変数セットのうち最小の平均分岐深さを有する第1変数セットを選択し、

- 前記第1変数セットの前記入力変数に第1記号値を割り当て、

- 前記第1記号値を有する前記第1変数セットの前記入力変数を用いて、前記ソフトウェアモジュールを記号的に実行することで、前記第1変数セットの前記入力変数及び前記中間変数に関する第1テストケースセットを生成し、

50

- 前記第1テストケースセットを用いて前記ソフトウェアモジュールを検証する、よう動作する、システム。

【請求項11】

1つ以上のプロセッサによって実行可能な命令を有するメモリと、前記メモリに結合され、前記命令を実行するよう動作する1つ以上のプロセッサとを有するシステムであって、

前記1つ以上のプロセッサは、前記命令を実行するとき、

- ソフトウェアモジュール及び該ソフトウェアモジュール用の複数のテストケースにアクセスし、

前記ソフトウェアモジュールは、

1つ以上の変数と、

1つ以上の条件付き分岐点であり、各条件付き分岐点が、前記変数のうちの少なくとも1つの変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも1つの変数の値に依存する、1つ以上の条件付き分岐点と、を有し、

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進み、

前記テストケースの各々は、前記条件付き分岐点のうちの少なくとも1つをカバーし、且つ前記条件付き分岐点のうちの前記少なくとも1つの分岐条件を規定する前記少なくとも1つの変数に関する少なくとも1つのテスト値を有し、

- 前記条件付き分岐点の各々に関して、前記複数のテストケースのうちに、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースが複数存在する場合に、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースのうち1つを除く全てを、前記複数のテストケースから除去し、

残りのテストケースから1つ以上の第1テストケースを選択し、各第1テストケースは、前記条件付き分岐点のうちの少なくとも1つが該第1テストケースによってのみカバーされるテストケースであり、

- 前記条件付き分岐点の全てが前記第1テストケース及び第2テストケースのうちの少なくとも1つのテストケースによってカバーされるまで、残りのテストケースから1つ以上の第2テストケースを繰り返し選択し、各繰り返しにおいて選択される第2テストケースは、該繰り返しにおいて最も多数の条件付き分岐点をカバーするテストケースであり、

- 選択された前記第1テストケース及び前記第2テストケースのみを用いて、前記ソフトウェアモジュールをテストする、よう動作する、システム。

【請求項12】

プログラムを格納したコンピュータ読み取り可能記憶媒体であって、前記プログラムは、1つ以上の計算装置によって実行されるとき、該1つ以上の計算装置に、

- ソフトウェアモジュールにアクセスさせ、

前記ソフトウェアモジュールは、

1つ以上の入力変数と、

1つ以上の中間変数であり、各中間変数の値が前記入力変数のうちの少なくとも1つの入力変数の値に直接的あるいは間接的に依存する、1つ以上の中間変数と、

1つ以上の条件付き分岐点であり、各条件付き分岐点が、前記入力変数及び前記中間変数のうちの少なくとも1つの入力変数及び/又は中間変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも1つの入力変数及び/又は中間変数の値に依存する、1つ以上の条件付き分岐点と、を有し、且つ

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる

10

20

30

40

50

経路に進み、

- 前記 1 つ以上の入力変数に対応する 1 つ以上の変数セットを構築させ、各変数セットは、前記入力変数のうちの 1 つと、前記入力変数のうちの該 1 つの値に直接的あるいは間接的に依存する値を有する中間変数とを含み、
- 各変数セットに関して、該変数セットの前記入力変数及び前記中間変数によって影響される平均分岐深さを計算させ、
- 前記 1 つ以上の変数セットのうち最小の平均分岐深さを有する第 1 変数セットを選択させ、
- 前記第 1 変数セットの前記入力変数に第 1 記号値を割り当てさせ、
- 前記第 1 記号値を有する前記第 1 変数セットの前記入力変数を用いて、前記ソフトウェアモジュールを記号的に実行させて、前記第 1 変数セットの前記入力変数及び前記中間変数に関する第 1 テストケースセットを生成させ、
- 前記第 1 テストケースセットを用いて前記ソフトウェアモジュールを検証させる、コンピュータ読み取り可能記憶媒体。

10

【請求項 13】

プログラムを格納したコンピュータ読み取り可能記憶媒体であって、前記プログラムは、1 つ以上の計算装置によって実行されるとき、該 1 つ以上の計算装置に、

- ソフトウェアモジュール及び該ソフトウェアモジュール用の複数のテストケースにアクセスさせ、

20

前記ソフトウェアモジュールは、

1 つ以上の変数と、

1 つ以上の条件付き分岐点であり、各条件付き分岐点が、前記変数のうちの少なくとも 1 つの変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも 1 つの変数の値に依存する、1 つ以上の条件付き分岐点と、を有し、

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進み、

前記テストケースの各々は、前記条件付き分岐点のうちの少なくとも 1 つをカバーし、且つ前記条件付き分岐点のうちの前記少なくとも 1 つの分岐条件を規定する前記少なくとも 1 つの変数に関する少なくとも 1 つのテスト値を有し、

30

- 前記条件付き分岐点の各々に関して、前記複数のテストケースのうちに、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースが複数存在する場合に、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースのうち 1 つを除く全てを、前記複数のテストケースから除去させ、

残りのテストケースから 1 つ以上の第 1 テストケースを選択させ、各第 1 テストケースは、前記条件付き分岐点のうちの少なくとも 1 つが該第 1 テストケースによってのみカバーされるテストケースであり、

- 前記条件付き分岐点の全てが前記第 1 テストケース及び第 2 テストケースのうちの少なくとも 1 つのテストケースによってカバーされるまで、残りのテストケースから 1 つ以上の第 2 テストケースを繰り返し選択させ、各繰り返しにおいて選択される第 2 テストケースは、該繰り返しにおいて最も多数の条件付き分岐点をカバーするテストケースであり、

40

選択された前記第 1 テストケース及び前記第 2 テストケースのみを用いて、前記ソフトウェアモジュールをテストさせる、コンピュータ読み取り可能記憶媒体。

【請求項 14】

ソフトウェアモジュールにアクセスする手段であり、

前記ソフトウェアモジュールは、

1 つ以上の入力変数と、

50

1つ以上の中間変数であり、各中間変数の値が前記入力変数のうちの少なくとも1つの入力変数の値に直接的あるいは間接的に依存する、1つ以上の中間変数と、

1つ以上の条件付き分岐点であり、各条件付き分岐点が、前記入力変数及び前記中間変数のうちの少なくとも1つの入力変数及び/又は中間変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも1つの入力変数及び/又は中間変数の値に依存する、1つ以上の条件付き分岐点と、を有し、且つ

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進む、アクセスする手段と、

前記1つ以上の入力変数に対応する1つ以上の変数セットを構築する手段であり、各変数セットが、前記入力変数のうちの1つと、前記入力変数のうちの該1つの値に直接的あるいは間接的に依存する値を有する中間変数とを含む、構築する手段と、

各変数セットに関して、該変数セットの前記入力変数及び前記中間変数によって影響される平均分岐深さを計算する手段と、

前記1つ以上の変数セットのうち最小の平均分岐深さを有する第1変数セットを選択する手段と、

前記第1変数セットの前記入力変数に第1記号値を割り当てる手段と、

前記第1記号値を有する前記第1変数セットの前記入力変数を用いて、前記ソフトウェアモジュールを記号的に実行し、前記第1変数セットの前記入力変数及び前記中間変数に関する第1テストケースセットを生成する手段と、

前記第1テストケースセットを用いて前記ソフトウェアモジュールを検証する手段と、を有するシステム。

【請求項15】

ソフトウェアモジュール及び該ソフトウェアモジュール用の複数のテストケースにアクセスする手段であり、

前記ソフトウェアモジュールは、

1つ以上の変数と、

1つ以上の条件付き分岐点であり、各条件付き分岐点が、前記変数のうちの少なくとも1つの変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも1つの変数の値に依存する、1つ以上の条件付き分岐点と、を有し、

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進み、

前記テストケースの各々は、前記条件付き分岐点のうちの少なくとも1つをカバーし、且つ前記条件付き分岐点のうちの前記少なくとも1つの分岐条件を規定する前記少なくとも1つの変数に関する少なくとも1つのテスト値を有する、アクセスする手段と、

前記条件付き分岐点の各々に関して、前記複数のテストケースのうちに、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースが複数存在する場合に、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースのうち1つを除く全てを、前記複数のテストケースから除去する手段と、

残りのテストケースから1つ以上の第1テストケースを選択する手段であり、各第1テストケースは、前記条件付き分岐点のうちの少なくとも1つが該第1テストケースによってのみカバーされるテストケースである、選択する手段と、

前記条件付き分岐点の全てが前記第1テストケース及び第2テストケースのうちの少なくとも1つのテストケースによってカバーされるまで、残りのテストケースから1つ以上の第2テストケースを繰り返し選択する手段であり、各繰り返しにおいて選択される第2テストケースは、該繰り返しにおいて最も多数の条件付き分岐点をカバーするテストケースである、選択する手段と、

選択された前記第1テストケース及び前記第2テストケースのみを用いて、前記ソフト

10

20

30

40

50

ウェアモジュールをテストする手段と、
を有するシステム。

【発明の詳細な説明】

【技術分野】

【0001】

この開示は概して、ソフトウェアモジュールのテスト及び検証を行うことに関する。

【背景技術】

【0002】

ソフトウェアアプリケーションは幾つもの数のモジュールを含むことができ、各モジュールが個々にテストあるいは検証されている。ソフトウェアモジュールは、手動あるいは自動的にテストあるいは検証され得る。手動の場合、人（例えば、ソフトウェアテストエンジニア）が、ソフトウェアモジュールのテストケースを該モジュールの設計仕様に基づいて手作業で設計し、該テストケースの下で該モジュールを実行し、テストケースに合致しないモジュール挙動又は出力を調べる。自動の場合、コンピュータソフトウェア又はハードウェアとして実装されたソフトウェアテストツールが、テスト対象のソフトウェアモジュールのテストケースを自動生成し、テストケースをシミュレートしながらその被テストモジュールを実行し、テストケースに合致しないモジュール挙動又は出力を調べ得る。

10

【発明の概要】

【発明が解決しようとする課題】

【0003】

ソフトウェアモジュールのテスト又は検証を支援し得る方法及びシステムが開示される。

20

【課題を解決するための手段】

【0004】

特定の一態様によれば、1つ以上の計算装置によって実行される方法は、ソフトウェアモジュールにアクセスすることを含む。前記ソフトウェアモジュールは、1つ以上の入力変数と、1つ以上の中間変数と、1つ以上の条件付き分岐点とを有する。各中間変数の値は、前記入力変数のうちの少なくとも1つの入力変数の値に直接的あるいは間接的に依存する。各条件付き分岐点は、前記入力変数及び前記中間変数のうちの少なくとも1つの入力変数及び/又は中間変数によって規定される分岐条件を有し、該分岐条件が満たされるかは、前記少なくとも1つの入力変数及び/又は中間変数の値に依存する。前記ソフトウェアモジュールの実行中、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進む。当該方法はまた、前記1つ以上の入力変数に対応する1つ以上の変数セットを構築することを含む。各変数セットは、前記入力変数のうちの1つと、前記入力変数のうちの該1つの値に直接的あるいは間接的に依存する値を有する中間変数とを含む。当該方法はまた、各変数セットに関して、該変数セットの前記入力変数及び前記中間変数によって影響される平均分岐深さを計算し、前記1つ以上の変数セットのうち最小の平均分岐深さを有する第1変数セットを選択することを含む。当該方法は更に、前記第1変数セットの前記入力変数に第1記号値を割り当て、前記第1記号値を有する前記第1変数セットの前記入力変数を用いて、前記ソフトウェアモジュールを記号的に実行し、前記第1変数セットの前記入力変数及び前記中間変数に関する第1テストケースセットを生成することを含み、前記第1テストケースセットを用いて前記ソフトウェアモジュールが検証される。

30

40

【0005】

特定のおの態様によれば、1つ以上の計算装置によって実行される方法は、ソフトウェアモジュール及び該ソフトウェアモジュール用の複数のテストケースにアクセスすることを含む。前記ソフトウェアモジュールは、1つ以上の変数と1つ以上の条件付き分岐点とを有する。各条件付き分岐点は、前記変数のうちの少なくとも1つの変数によって規定される分岐条件を有し、該分岐条件が満たされるかは、前記少なくとも1つの変数の値に依存する。前記ソフトウェアモジュールの実行中、前記条件付き分岐点の各々において、

50

前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進む。前記テストケースの各々は、前記条件付き分岐点のうち少なくとも1つをカバーし、且つ前記条件付き分岐点のうちの前記少なくとも1つの分岐条件を規定する前記少なくとも1つの変数に関する少なくとも1つのテスト値を有する。当該方法はまた、前記条件付き分岐点の各々に関して、前記複数のテストケースのうちに、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースが複数存在する場合に、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースのうち1つを除く全てを、前記複数のテストケースから除去することを含む。当該方法はまた、残りのテストケースから1つ以上の第1テストケースを選択することを含み、各第1テストケースは、前記条件付き分岐点のうち少なくとも1つが該第1テストケースによってのみカバーされるテストケースである。当該方法は更に、前記条件付き分岐点の全てが前記第1テストケース及び第2テストケースのうち少なくとも1つのテストケースによってカバーされるまで、残りのテストケースから1つ以上の第2テストケースを繰り返し選択することを含み、各繰り返しにおいて選択される第2テストケースは、該繰り返しにおいて最も多数の条件付き分岐点をカバーするテストケースである。選択された前記第1テストケース及び前記第2テストケースのみを用いて、前記ソフトウェアモジュールがテストされる。

10

【図面の簡単な説明】

【0006】

【図1】一例に係るソフトウェアモジュール上で記号的実行を行うステップを表す実行フロー例を示す図である。

20

【図2】ソフトウェアモジュールの入力変数を記号的に実行する順序を決定する方法例を示す図である。

【図3】ソフトウェアモジュールの入力変数を記号的に実行する順序を決定するシステム例を示す図である。

【図4】複数レベルのネスト化された条件付き分岐を有する一例に係るソフトウェアモジュールの実行経路を示す図である。

【図5】テストケースセットのサイズを縮小する方法例を示す図である。

【図6】テストケースセットのサイズを縮小するシステム例を示す図である。

【図7】一例に係るソフトウェアモジュールの入力変数の例のテスト値の例を表すグラフ例を示す図である。

30

【図8】一例に係るソフトウェアモジュールの実行経路上でのテスト網羅率を示す二部グラフ例を示す図である。

【図9】一例に係るネットワーク環境を示す図である。

【図10】一例に係るコンピュータシステムを示す図である。

【発明を実施するための形態】

【0007】

以下、図面に示す幾つかの実施形態を参照して詳細に説明する。以下の説明においては、この開示の十分な理解を提供するため、数多くの具体的詳細事項を説明する。しかしながら、ここで開示される事項は、それら具体的詳細事項の一部又は全てを用いずして実施されることもある。また、この開示をいたずらに不明瞭にしないよう、周知の処理ステップ及び/又は構造は詳細には説明しない。さらに、ここでは特定の実施形態との関連で説明するが、理解されるように、ここでの説明は、本開示をそのような実施形態に限定することを意図するものではない。逆に、ここでの説明は、添付の請求項により定められる本開示の精神及び範囲に含まれ得る代替例、変形例及び均等例に及ぶものである。

40

【0008】

ソフトウェアモジュールをテストあるいは検証するため、典型的に、例えば該モジュールの設計仕様又はその他の種類の要求（例えば、機能上、挙動上若しくは性能上の要求）に基づいて、該モジュールのためのテストケースが生成される。その後、テストケースの下でソフトウェアモジュールが実行され、該モジュールの挙動又は出力がテストケースに対して

50

検証される。特定の実施形態において、各テストケースは、一組の入力値と、該入力値に関連してソフトウェアモジュール上で実行される具体的なテストを記述する情報とを含み得る。特定の実施形態において、入力値は、ソフトモジュールがテストされているときに該モジュールの入力変数に割り当てられ得る。各入力値は、モジュールに特定の応答（例えば、特定の出力値を提供する）をさせるように選択され得る。テスト情報は、ソフトウェアモジュールのどのような特定の機能がテストされるか、どのような状況下でどのように入力値がソフトウェアモジュールの入力変数に適用されるか、ソフトウェアモジュールが入力値に応答して提供すべき正しい挙動又は出力値は何か、等々を記述し得る。

【0009】

ソフトウェアモジュールに関するテストケースは、手動（例えば、人によって）あるいは自動（例えば、コンピュータハードウェア又はソフトウェアとして実装されたテストケース生成ツールによって）の何れかで生成され得る。しばしば、手動生成されるテストケースは、テスト対象のソフトウェアモジュールのソースコードの様々な部分で比較的乏しい網羅率（カバレッジ）を有し、また、テストケースを手動生成する実処理は、時間を消費し且つコストのかかるものとなる。一方、自動生成されるテストケースは、より良好なテスト網羅率を提供するとともに、より少ない時間及び人的労力のみを必要とする。

【0010】

特定の実施形態において、テストケースの入力値は、テスト対象ソフトウェアモジュール上で記号的実行（symbolic execution）を行うことによって取得され得る。しかしながら、この手法は幾つかの欠点を有することがある。記号的実行は、CPU時間及びその他の種類のコンピュータリソースの点で非常に高くつき、ソフトウェアモジュール上で、典型的に、一度に該ソフトウェアモジュールの1つの入力変数のみを記号的にすることによって実行される。ソフトウェアモジュールが2つ以上の入力変数を有する場合、記号的実行が該ソフトウェアモジュール上で何度も実行される必要がある。また、複数の入力変数のうち、何れの1つが最初に記号的に実行され、何れの1つが2番目に記号的に実行されるかに依存して、記号的実行から得られる入力値は、異なり得るとともに異なったテスト網羅率をもたらし得る。第2に、ソフトウェアモジュールが比較的多数の入力変数を有する場合、記号的実行から得られる入力変数の数、ひいては、生成されるテストケースの数は、テストケース・イクスプロージョン（爆発）といえるほどに非常に大きいものとなり得る。テストケース爆発は、テストケース群が例えばテスト集合（テストスイート）に統合されるときに多大な回帰テスト時間を生じさせ得る。

【0011】

これらの欠点を解消するため、複数の入力変数を有するソフトウェアモジュール上で記号的実行を行うとき、特定の実施形態は、それら入力変数を記号的に実行する順序を、入力変数が影響する分岐深さに基づいて知的に決定し得る。また、記号的実行の結果としてテストケースが生成された後、特定の実施形態は、ソフトウェアモジュールをテストするために最終的に使用されるテストケースの数を削減するため、重複あるいは過大なテスト網羅率をもたらすテストケースを排除し得る。

【0012】

コンピュータサイエンスの分野において、記号的実行は、抽象解釈として実際の値ではなく記号的な値を追跡することによるソフトウェアプログラムの分析を意味する。これは、ソフトウェアモジュールへの入力を記号変数として扱うノンエクスプリシット状態モデル検査技術である。記号的実行は、記号変数を用いてソフトウェアモジュール内の限定された経路を実行することによって複雑な数学的方程式を作成し、ソルバーを用いて該複雑な方程式を解く（典型的に、決定手順として知られる）ことで、もしあれば、誤ったシナリオを得る。エクスプリシット状態モデル検査と異なり、記号的実行は、分析対象のソフトウェアモジュールの全ての入力変数の、全ての取り得る入力値及び全ての取り得るユースケースを分析することができる。

【0013】

記号的実行の更なる説明のため、以下の“foo”という名のソフトウェアモジュール

例を考える。

(外1)

1	foo (a, b) {
2	string a, b, c, d;
3	c = a.concat (b);
4	if !(c.equals ("qrs")) {
5	d = c.concat ("t");
6	return d;
7	} else {
8	return c;
9	}
10	}

10

サンプルコード1

【0014】

ソフトウェアモジュール“foo”は、2つの入力変数“a”及び“b”と、2つの中間変数“c”及び“d”とを有している。特定の実施形態において、ソフトウェアモジュールの中間変数は、そのソフトウェアモジュールにローカルな変数であり、そのソフトウェアモジュールのコンテキスト内のみ存在する。中間変数はまた、局所的な範囲を有するのみであり、且つその中間変数が宣言されたソフトウェアモジュール内からのみアクセス可能であるので、ローカル変数とも呼ばれる。中間変数の値は、直接的あるいは間接的に、ソフトウェアモジュールの1つ以上の入力変数の値に依存し得る。例えば、モジュール“foo”では、中間変数“c”の値は、上記コードの第3行にて指し示されるように、入力変数“a”及び“b”の値に直接的に依存し、中間変数“d”の値は、上記コードの第5行にて指し示されるように、中間変数“c”を介して間接的に入力変数“a”及び“b”の値に依存する。また、モジュール“foo”は、上記コードの第4行に、“if - else”命令文によって引き起こされる条件付き分岐点を含んでいる。第4行の条件付き分岐点は、分岐条件“!(c.equals (“qrs”))”に関連付けられている。この分岐条件が満足される、あるいは成立するかに応じて、すなわち、中間変数“c”が“qrs”に等しいかに応じて、モジュール“foo”はそれ以降、異なる実行経路へと進み、モジュール“foo”のコードの異なる部分が実際に実行される。より具体的には、中間変数“c”が“qrs”に等しくない場合には、上記コードの第5及び6行にて指し示されるように、中間変数“d”の値が計算されて返される。一方、中間変数“c”が“qrs”に等しい場合には、上記コードの第8行にて指し示されるように、中間変数“c”の値が返される。

20

30

【0015】

モジュール“foo”上で記号的実行が行われるとき、その入力変数及び中間変数の各々に、実際値に代えて記号値が割り当てられる。図1は、モジュール“foo”上で記号的実行を行うステップを表す一例に係る実行フロー100を示している。この例において、入力変数“a”は記号値“x”を割り当てられ、入力変数“b”は記号値“y”を割り当てられ、中間変数“c”は記号値“z”を割り当てられ、中間変数“d”は記号値“w”を割り当てられている。変数“a”、“b”、“c”及び“d”は“string”型であるので、記号値“x”、“y”、“z”及び“w”は各々、任意の文字列を表す。

40

【0016】

また、“ ”は、実行経路に沿った様々な点での記号的実行の結果を表す記号表現である。より具体的には、上記コードの第2行に対応する102において、変数“a”、“b

50

”、“c”及び“d”にそれぞれの記号値“x”、“y”、“z”及び“w”が割り当てられ、“ ”は当初、エンपティすなわちヌル表現を有する。この実行が更に進むとき、どのコードが実行されたかに応じて“ ”に表現が追加される。上記コードの第3行に対応する104において、“ ”は表現“z = concat(x, y)”を有している。これは、上記コードの第3行が“c = a . concat(b)”であり、“x”、“y”及び“z”がそれぞれ変数“a”、“b”及び“c”に割り当てられた記号値であるからである。次いで、上記コードの第4行は条件付き分岐点であり、それ以降、この実行が進み得る2つの取り得る実行経路が存在する。故に、記号的実行も104から下に、上記コードの第5及び6行に対応する106及び108を含む第1経路“PATH1”と、上記コードの第8行に対応する110を含む第2経路“PATH2”という2つの異なる経路に進み得る。

10

【0017】

PATH1を進めるには、変数“c”は“qrs”に等しくなく、これは記号値“z”が“qrs”に等しくないことを意味する。故に、106にて、表現“z != “qrs””が“ ”に追加される。逆に、PATH2を進めるには、変数“c”は“qrs”に等しく、これは記号値“z”が“qrs”に等しいことを意味する。故に、110にて、表現“z = “qrs””が“ ”に追加される。PATH1に沿うとき、変数“d”の値は上記コードの第5行にて決定され、これが108に対応する。故に、108にて、表現“w = concat(z, “t”)”が“ ”に追加される。なお、“z = concat(x, y)”であるので、“w”の表現は“w = concat(concat(x, y), “t”)”に書き直され得る。108はPATH1の終点であるので、108における“ ”の表現は、PATH1の終点に到達するために満足される必要がある条件を記号の形態で表している。同様に、110はPATH2の終点であるので、110における“ ”の表現は、PATH2の終点に到達するために満足される必要がある条件を記号の形態で表している。

20

【0018】

特定の実施形態において、108での“ ”の表現の値を求めることは、モジュール“foo”をPATH1の終点まで到達させた入力変数“a”及び“b”の実際値を提供し、また、110での“ ”の表現の値を求めることは、モジュール“foo”をPATH2の終点まで到達させた入力変数“a”及び“b”の実際値を提供し得る。特定の実施形態において、これらの実際値は、その後、モジュール“foo”をテストするときに入力変数“a”及び“b”に割り当てられ得るテスト値として使用され得る。これらのテスト値は、PATH1の終点及びPATH2の終点の双方にモジュール“foo”を到達させる入力変数“a”及び“b”の実際値を含むので、これらのテスト値が適用される場合、モジュール“foo”の全ての取り得る実行経路がテストされることが保証され得る。

30

【0019】

上述のように、記号的実行は高くつくので、通常、一度に1つの入力変数のみが記号的にされて記号的に実行される。ソフトウェアモジュールが複数の入力変数を有する場合、記号的実行は、一度に1つずつ全ての入力変数にわたって繰り返し実行される必要がある。記号的にされるように複数の入力変数が選択される順序は、最終的な網羅率の点で、自動生成されるテストセットの品質に多大な影響を有し得る。例えば、以下のコードセグメントを考える。

40

(外2)

1	if (a > 5) {
2	...
3	if (b > 2) {
4	...
5	} else {
6	...
7	}
8	} else {
9	...
10	}

10

サンプルコード2

【0020】

“a”及び“b”は入力変数であるとする。入力変数“a”が最初に記号的にされる場合、記号的実行を行うことは、例えば、第1行の条項“(a > 5)”に対応する変数“a”の2つの具体値(すなわち、実際値)6及び4を生じさせ得る。6及び4である“a”の具体値を用いて“b”が2番目に記号的にされる場合、記号的実行を行うことによって2つの異なる“b”の具体値1及び3がもたらされ得る。入力値の対(a, b) = {(6, 1), (6, 3), (4, 1), (4, 3)}は、上記コードセグメント中の全ての取り得る実行経路をカバーすることが保証される。

20

【0021】

一方、入力変数“a”が最初に記号的にされることに代えて、入力変数“b”が最初に記号的にされるように選択されるとする。記号的実行において、入力変数“a”について何らかの具体値が仮定される必要があるため、“a”にランダム値が割り当てられ得る。“a”の値が5未満であるように仮定される場合、記号的実行は、第3及び4行にある“if”ブロックを完全に見落とすことになる。結果として、記号的実行は変数“b”について有意義な値を生成することができない。最終的に得られるテストセットは、入力変数“b”について生成された何らかのランダム値を用いてその後計算された、“a”の2つの値のみを有することになる。得られるテストセットは、(a, b) = {(6, 7), (4, 7)}のようなものに見えることになる。ただし、7は、入力変数“b”に割り当てられたランダム値である。この第2のテストセットは、第1のテストセットから得られるコード網羅率より遙かに劣悪なコード網羅率をもたらすテスト入力値を生じさせることになる。従って、記号的実行の目的で入力変数が記号的にされる順序は、自動生成されるテストセットによって得られる最終的なコード網羅率のレベルに多大なる影響を及ぼし得る。

30

【0022】

この可能性ある問題を解決するため、特定の実施形態は、ソフトウェアモジュールの複数の入力変数を記号的に実行する順序を知的に決定し得る。図2は、ソフトウェアモジュールの入力変数を記号的に実行する順序を決定する一例に係る方法を示し、図3は、ソフトウェアモジュールの入力変数を記号的に実行する順序を決定する一例に係るシステム300を示している。図2及び3を互いに関連付けて説明する。

40

【0023】

特定の実施形態において、ソフトウェアモジュールは任意の数の入力変数と任意の数の中間変数とを有し得る。特定の実施形態において、各中間変数の値は、直接的あるいは間接的に、少なくとも1つの入力変数の値に依存し得る。すなわち、入力変数の値に対する変化は、その入力変数に直接的あるいは間接的に依存する中間変数の値に対する変化を生じさせる。当然ながら、1つの中間変数が複数の入力変数に依存することがあり、また、複数の中間変数が同一の入力変数に依存することがある。特定の実施形態は、図2のステ

50

ップ202に示すように、入力変数の関連性分析を実行することによって、何れの間変数が直接的あるいは間接的にソフトウェアモジュールの何れの入力変数に依存するかを決定し得る。特定の実施形態において、ステップ202は、図3に示すシステム300の構成要素302によって実行され得る。構成要素302は、ソフトウェアモジュール312を入力として用いて、ソフトウェアモジュール312の入力変数について関連性分析を実行するものである。

【0024】

特定の実施形態において、関連性分析は、一組の記号変数に関連するソフトウェアモジュールの変数の組を計算し得る。特定の実施形態は、一組の記号入力を与えられたときに変数が記号値を格納する場合に、その変数を関連あるものと見なし得る。これは、ソフトウェア解析は抽象解釈のモデル検査と見なされ得るという洞察に基づく。結果として、ソフトウェア解析は、抽象解釈による健全性保証と、モデル検査による自動化の利益とを享受する。特定の実施形態において、ソフトウェアモジュール312上で実行される関連性分析は、重み付けプッシュダウンモデル検査技術に基づき得る。関連性分析は、2009年2月27日に出願された“Efficient Symbolic Execution of Software Using Static Analysis”なるタイトルの米国特許出願第12/395515号、及び2009年3月19日に出願された“Environmental Data Refinement Based on Static Analysis and Symbolic Execution”なるタイトルの米国特許出願第12/407695号に更に詳細に説明されている。

10

【0025】

何れの変数がソフトウェアモジュールの何れの入力変数に依存するかが決定されると、特定の実施形態は、図2のステップ204に示すように、各入力変数に対する変数セットを構築し得る。変数セットは、入力変数自体と、関連性分析の結果に従った、該入力変数に直接的あるいは間接的に依存する全ての間変数とを含む。斯くして構築される変数セットの数は、ソフトウェアモジュールが有する入力変数の数に等しい。なお、中間変数は複数の入力変数に依存し得るので、1つの中間変数が複数の変数セットに属することがあり得る。

20

【0026】

各変数セットに対し、特定の実施形態は、図2のステップ206に示すように、該セットからの入力変数及び中間変数によって到達される平均分岐深さを計算し得る。特定の実施形態において、ステップ206は、図3に示すシステム300の構成要素304によって実行され得る。特定の実施形態において、ソフトウェアモジュールは如何なる数の条件付き分岐点（例えば、“if”条項）を有していてもよく、各条件付き分岐点は、少なくとも1つの入力変数又は中間変数によって規定される分岐条件を有する。分岐条件が満たされるかに応じて、ソフトウェアモジュールはそれ以降、異なる実行経路を進み得る。例えば、以下のコードセグメントを考える。

30

(外3)

```

1  if (m > n && k <= 10) {
2      ...
3      if (k > 2) {
4          ...
5      } else {
6          if (n == k || n == m) {
7              ...
8          } else {
9              ...
10         }
11     }
12 } else {
13     ...
14 }

```

10

【0027】

第1行に条件付き分岐点が存在し、その分岐条件は“ $m > n \ \&\& \ k \leq 10$ ”である。なお、1つの分岐条件が複数の副(サブ)条件(例えば、“ $m > n$ ”及び“ $k \leq 10$ ”)を含んでいてもよい。第1行の分岐条件が満足される(すなわち、成立する)には、変数“ m ”の値が変数“ n ”の値より大きい必要があり、且つ変数“ k ”の値が10以下である必要がある。第1行の分岐条件が満たされる場合、第2 - 11行の間のコードが実行される。それ以外の場合(すなわち、第1行の分岐条件が満たされない場合)には、第13行にあるコードが実行される。第3行に別の条件付き分岐点が存在し、その分岐条件は“ $k > 2$ ”である。第3行の分岐条件が満足されるには、変数“ k ”の値が2より大きい必要がある。同様に、第3行の分岐条件が満たされる場合、第4行にあるコードが実行される。それ以外の場合には、第6 - 11行の間のコードが実行される。第6行に第3の条件付き分岐点が存在し、その分岐条件は、複数の副条件(例えば、“ $n = k$ ”及び“ $n = m$ ”)を含む“ $n = k \ || \ n = m$ ”である。第6行の分岐条件が満足されるには、変数“ n ”の値が変数“ k ”の値又は変数“ m ”の値の何れかに等しい必要がある。第6行の分岐条件が満たされる場合、第7行にあるコードが実行される。それ以外の場合には、第9行にあるコードが実行される。

20

30

【0028】

また、第3行の条件付き分岐点は第1行の条件付き分岐点内に入れ子にされ(ネスト化され)ており、さらに、第6行の条件付き分岐点は第3行の条件付き分岐点内にネスト化されている。第1行の条件付き分岐点に関する分岐条件が満たされない場合、第3行の条件付き分岐点は、そのコードの実行中に到達されることはない。しかしながら、第3行の条件付き分岐点に関する分岐条件が満たされない場合、第6行の条件付き分岐点は、そのコードの実行中に到達される。実際には、第6行の条件付き分岐点に到達するためには、第1行の分岐条件は満たされる必要があるが、第3行の分岐条件は満たされることができない。

40

【0029】

図4は、複数レベルのネスト化された条件付き分岐を有する一例に係るソフトウェアモジュールの実行経路を示している。条件付き分岐点402において、“条件1”が満たされる場合、実行経路は処理ステップ412に進み、“条件1”が満たされない場合、実行経路は処理ステップ414に進む。条件付き分岐点402内にネスト化された条件付き分岐点404において、“条件2”が満たされる場合、実行経路は処理ステップ416に進み、“条件2”が満たされない場合、実行経路は終了する。やはり条件付き分岐点402内にネスト化された条件付き分岐点406において、“条件3”が満たされる場合、実行

50

経路は終了し、“条件3”が満たされない場合、実行経路は条件付き分岐点408に進む。そして、条件付き分岐点406内に更にネスト化された条件付き分岐点408において、“条件4”が満たされる場合、実行経路は処理ステップ418に進み、“条件4”が満たされない場合、実行経路は処理ステップ422に進む。図4においては、全部で3階層のネスト化された条件付き分岐が存在している。

【0030】

複数の変数セットのうちの一つからの入力変数及び中間変数によって到達される平均分岐深さを計算するため、該セットからの入力変数及び各中間変数に対し、特定の実施形態は、その入力変数又はその中間変数によって到達される分岐深さを計算し得る。その後、特定の実施形態は、全ての個々の入力変数及び中間変数について計算された分岐深さの平均を取って、該変数セットの平均分岐深さを決定し得る。該変数セットからの複数の変数のうちの一つによって到達される分岐深さを計算するため、特定の実施形態は、ソフトウェアモジュール内の条件付き分岐点の各々を、最も外側の条件付き分岐点から始めて、徐々に内部のネスト化された条件付き分岐点へと進みながら調べ得る。その変数が、或る条件付き分岐点に関連付けられた分岐条件に影響を及ぼす場合、特定の実施形態は、その変数の分岐深さを1だけ増大させ得る。上述のコードセグメント例においては、第1行の条件付き分岐点に関する分岐条件は変数“m”、“n”及び“k”に影響される。一緒になって該分岐条件を形成する副条件に、3つ全ての変数が関与しているからである。一方、第3行の条件付き分岐点に関する分岐条件は、変数“k”のみが該分岐条件に関与しているため、変数“k”によって影響されるのみである。

10

20

【0031】

全ての変数セットに関して、平均分岐深さが斯くして計算されると、特定の実施形態は、図2のステップ208に示すように、最小の平均分岐深さを有する変数セットを選択し得る。同一の最小平均分岐深さを有する変数セットが複数存在する場合、特定の実施形態は、最小平均分岐深さを有する複数の変数セットのうちの一つをランダムに選択してもよい。

【0032】

特定の実施形態は、図2のステップ210に示すように、選択された変数セットからの入力変数について記号的実行を行って、ソフトウェアモジュールのテストケースを生成し得る。特定の実施形態において、ステップ210は、図3に示したシステム300の構成要素306によって実行され得る。構成要素306は、出力として一組のテストケース314を提供するものである。図1に関連して上述したように、特定の実施形態は、選択された変数セットからの入力変数に記号値を割り当て、ソフトウェアモジュール上で記号的実行を行い、異なる実行経路の終点で得られる記号表現（例えば、“ ”）を解くことで、選択された変数セットからの入力変数に関する実際値を取得し得る。これら実際値は、後にソフトウェアモジュールがテストされるときに入力変数に適用されるべきテスト値として使用され得るものである。入力変数に関するこれらの実際値は、ソフトウェアモジュールのテストケースを生成するために使用され得る。特定の実施形態は、毎回別の変数セットを選択して、この処理を複数回繰り返し得る。例えば、第2の繰り返しにおいて、2番目に小さい平均分岐深さを有する変数セットが選択され、記号値を有するこの第2の変数セットからの入力変数を用いて、ソフトウェアモジュール上で記号的実行が行われ得る。この記号的実行の結果に基づいて、この入力変数に関する実際値が決定され得る。この場合も、複数の変数セットが、同一の、この繰り返しにおける平均分岐深さを有する場合、それらの変数セットのうちの一つがランダムに選択され得る。第3の繰り返しにおいて、3番目に小さい平均分岐深さを有する変数セットが選択され、記号値を有するこの第3の変数セットからの入力変数を用いて、ソフトウェアモジュール上で記号的実行が行われ得る。そして、ソフトウェアモジュールに対して十分なテストケースが得られるまで、同様に繰り返される。換言すれば、各繰り返しにおいて最小の平均分岐深さを有する変数セットが該繰り返しのために選択され、選択された変数セットからの入力変数に関して実際値が決定される。

30

40

50

【 0 0 3 3 】

図 2 に示した処理又はその他の好適な手段の何れかを介してソフトウェアモジュールに対して一組のテストケース（テストケースセット）が生成されるとき、時にして、該セットからの全てのテストケースがソフトウェアモジュールについて価値あるテストカバレッジを提供するわけでないことがあり、また、ソフトウェアモジュールをテストするためにこれら全てのテストケースを適用すると、多大なリソース（例えば、コンピュータリソースの点で）を必要とすることがある。特定の実施形態は、上記テストケースセットからの個々のテストケースのカバレッジ値を調べ、ソフトウェアモジュールをテストするために、それらテストケースのうちの部分集合（サブセット）のみを選択してもよい。図 5 は、テストケースセットのサイズを縮小するための一例に係る方法を示し、図 6 は、テストケースセットのサイズを縮小する一例に係るシステム 6 0 0 を示している。図 5 及び 6 を互いに関連付けて説明する。

10

【 0 0 3 4 】

上述のように、特定の実施形態において、ソフトウェアモジュールは、任意の数の入力変数と、それら入力変数のうちの少なくとも 1 つに各々が直接的あるいは間接的に依存する任意の数の中間変数と、任意の数の条件付き分岐点とを有し得る。各条件付き分岐点は、入力変数又は中間変数のうちの少なくとも 1 つに影響される分岐条件に関連付けられる。特定の実施形態において、所与の 1 つの入力変数に関し、該入力変数自体によって、あるいは該入力変数に直接的あるいは間接的に依存する何れかの中間変数によって、の何れかで影響を受ける分岐条件を有する全ての条件付き分岐点が、該入力変数に影響されるものと見なされる。例えば、中間変数 “ j ” が入力変数 “ i ” に依存すると仮定する。さらに、1 つの条件付き分岐点が、入力変数 “ i ” に影響を受ける分岐条件を有し、且つ別の 1 つの条件付き分岐点が、中間変数 “ j ” に影響を受ける分岐条件を有すると仮定する。この場合、中間変数 “ j ” は入力変数 “ i ” に依存しているので、双方の条件付き分岐点が、入力変数 “ i ” に影響されるものと見なされる。

20

【 0 0 3 5 】

特定の実施形態において、上述のテストケースセットにおいて、該セットからの各テストケースは、ソフトウェアモジュールのテスト中にテスト入力値としてソフトウェアモジュールの入力変数に適用され得る実際値の組み合わせを含み得る。例えば、ソフトウェアモジュールが、“ i n t e g e r ” 型の 3 つの入力変数 “ a ”、“ b ” 及び “ c ” を有すると仮定する。さらに、入力変数 “ a ” は実際値 2 及び 7 を有し、入力変数 “ b ” は実際値 5 及び 1 1 を有し、入力変数 “ c ” は実際値 5 2、9 及び 2 7 を有することが決定されていると仮定する。3 つの入力変数のこれらの実際値から、ソフトウェアモジュールに対して、1 2 個の取り得るテストケースを含むセットが生成され得る。何故なら、これら 3 組の実際値には 1 2 個の固有の組み合わせ：

30

【表 1】

テストケース	“a” の値	“b” の値	“c” の値
1	2	5	52
2	2	5	9
3	2	5	7
4	2	11	52
5	2	11	9
6	2	11	7
7	7	5	52
8	7	5	9
9	7	5	7
10	7	11	52
11	7	11	9
12	7	11	7

10

が存在するからである。

【0036】

特定の実施形態は、図5のステップ502に示すように、ソフトウェアモジュールに対して生成されたテストケースセットを分析し得る。特定の実施形態において、ステップ502は、図6に示したシステム600の構成要素602によって実行され得る。構成要素602は、分析のためにテストケースセット612を入力として取り、テストケースセット612から冗長なテストケースを除去することによって、テストケースセット612のサイズを縮小することを試みる。

20

【0037】

ソフトウェアモジュールの個々の条件付き分岐点は、入力変数に影響される。すなわち、或る入力変数にテスト値として割り当てられる相異なる実際値は、ソフトウェアモジュールを、該入力変数に影響される或る条件付き分岐点から通じる相異なる実行経路に進ませ得る。上述の例において、入力変数“a”は条件付き分岐点“cbp1”に影響を及ぼすと仮定する。条件付き分岐点“cbp1”から、ソフトウェアモジュールは下方に、2つの異なる実行経路に進み得る。より具体的には、入力変数“a”にテスト値2が割り当てられる場合、ソフトウェアモジュールは一方の実行経路に進み、入力変数“a”にテスト値7が割り当てられる場合、ソフトウェアモジュールは他方の実行経路に進み得る。この場合、入力変数“a”の値が2である全てのテストケースは第1の実行経路を対象とし、上述の例では、そのようなテストケースが6個存在する。同様に、入力変数“a”の値が7である全てのテストケースは第2の実行経路を対象とし、上述の例では、そのようなテストケースがやはり6個存在する。

30

【0038】

特定の実施形態は、個々の入力変数に割り当てられ得るテストケースセットからのテスト値を表すグラフを構築し得る。図7は、入力変数“a”、“b”及び“c”を有する上述の例で構築される一例に係るグラフ700を示している。グラフ700内の各エッジは1つの入力変数を表し、グラフ700内の各ノードは、例示した12個のテストケースからの或る入力変数に関して取り得る1つのテスト値を表している。グラフ700の異なる実行経路に沿って進むことは、最終的に、異なるテストケースにつながる。特定の実施形態において、このようなグラフは、何れのテストケースがソフトウェアモジュールの何れの実行経路をカバーするかを特定する助けとなり得る。特定の実施形態において、同一の実行経路を対象とする複数のテストケースが存在する場合、図5のステップ504に示すように、冗長なテストケースのうち1つを残して全てがテストケースセットから除去され得る。そのような追加的なテストケースは、実際には、ソフトウェアモジュールに対して

40

50

如何なる追加のテストカバレッジをも提供しないからである。これにより、テストケースセットのサイズが縮小され得る。例えば、図7の場合、ノード702及び704において、ノード702及び704からのグラフ700のその後の分岐はソフトウェアモジュールの実行経路に、ノード706、708及び712の位置のテストケースによってカバーされていないような新たなテストカバレッジを提供するものでないことが決定され得る。故に、ノード702及び704より下の全てのテストケースが、テストケースセットから除去され得る。

【0039】

図7に示したグラフ700のようなグラフを用いることでは、テストケースセットから全ての冗長テストケースを除去することができないことがある。特定の実施形態は、図8に一例を示すような二部(bipartite)グラフに基づいて、テストケースセットのサイズを更に縮小し得る(例えば、テストセット圧縮を更に実行する)。特定の実施形態において、例えば図8に示したものなどの二部グラフが、図6に示したシステム600の構成要素604によって、ソフトウェアモジュールのテストケース及び条件付き分岐に対して構築され得る。特定の実施形態において、二部グラフ(例えば、二部グラフ800)は2つの部分を含み得る。1つの部分(例えば810)はテストケースセットからの残存テストケースを表し、各ノードが残存テストケースのうちの1つを表す。もう1つの部分(例えば820)はソフトウェアモジュールの全ての取り得る条件付き分岐を表し、各ノードが条件付き分岐のうちの1つを表す。或るテストケースが或る条件付き分岐をカバーする場合、1つの有向エッジが、そのテストケースを表す部分810内のノードを、その条件付き分岐を表す部分820内のノードに結び付ける。或る1つの条件付き分岐が、如何なる利用可能なテストケースによってもカバーされない(例えば、ノード822)、唯一のテストケースによってカバーされる(例えば、ノード824)、あるいは複数のテストケースによってカバーされる(例えば、ノード826)の何れもが可能である。

【0040】

斯くして二部グラフ(例えば、二部グラフ800)が構築されると、特定の実施形態は、図5のステップ506及び508に示すように、欲張り(グリーディ)アルゴリズムを用いて反復的にテストケースを選択し得る。特定の実施形態において、ステップ506及び508は、図6に示したシステム600の構成要素606によって実行され得る。構成要素606は、テストケースセット612から選択されたテストケースのサブセット614を出力として提供する。特定の実施形態は、先ず、図5のステップ506に示すように、条件付き分岐のうちの少なくとも1つを単独でカバーする二部グラフ内の不可欠なテストケースの全てを選択し得る。すなわち、選択される不可欠なテストケースの各々は、或る特定の条件付き分岐をカバーする唯一のテストケースである。例えば、図8において、ノード812は、ノード824及び828によって表される条件付き分岐をカバーする唯一のテストケースであるので、不可欠なテストケースを表している。同様に、ノード814も不可欠なテストケースを表している。これらのテストケースは最初に選択される。しかしながら、ノード816は、如何なる条件付き分岐も単独ではカバーしていないので、不可欠なテストケースを表しておらず、故に、このテストケースはこの時点では選択されない。

【0041】

不可欠なテストケースが全て選択されると、それらは二部グラフから除去され得る。また、不可欠なテストケースがカバーする条件付き分岐も二部グラフから除去され得る。次に、二部グラフ内に残ったテストケースから、特定の実施形態は、図5のステップ508に示すように、利用可能なテストケースによってカバーされる全ての条件付き分岐が、選択されたテストケースによってカバーされるまで、反復的に、一度に最も多数の条件付き分岐をカバーするテストケースを選択し、選択したテストケースを二部グラフから除去し得る。利用可能なテストケースの何れにもカバーされていない条件付き分岐(例えば、図8のノード822によって表される条件付き分岐)が存在する場合、特定の実施形態は、この欲張りアルゴリズムの目的上それを無視し得る。その後、特定の実施形態は、図5の

10

20

30

40

50

ステップ510に示すように、選択されたテストケースのみを用いてソフトウェアモジュールをテストし得る。

【0042】

特定の実施形態において、図3及び6に示した構成要素は、コンピュータハードウェア又はソフトウェアとして実装され得る。特定の実施形態は、ネットワーク環境内に実装され得る。図9は、サービスとしてソフトウェアの検証を提供するのに好適な、一例に係るネットワーク環境900を示している。ネットワーク環境900は、1つ以上のサーバ920と1つ以上のクライアント930とを相互に結合するネットワーク910を含んでいる。特定の実施形態において、ネットワーク910は、インターネット、エクストラネット、仮想プライベートネットワーク(VPN)、ローカルエリアネットワーク(LAN)、無線LAN(WLAN)、ワイドエリアネットワーク(WAN)、メトロポリタンエリアネットワーク(MAN)、インターネットの一部、若しくはその他のネットワーク、又は2つ以上のこれらネットワークの組み合わせである。この開示は、如何なる好適なネットワーク910をも意図している。

10

【0043】

1つ以上のリンク950が、サーバ920又はクライアント930をネットワーク910に結合する。特定の実施形態において、1つ以上のリンク950は各々、1つ以上の有線リンク、無線リンク、又は光リンクを含む。特定の実施形態において、1つ以上のリンク950は各々、イントラネット、エクストラネット、VPN、LAN、WLAN、WAN、MAN、インターネットの一部、若しくはその他のリンク、又は2つ以上のこれらリンクの組み合わせを含む。この開示は、サーバ920及びクライアント930をネットワーク910に結合する如何なる好適なリンク950をも意図している。

20

【0044】

特定の実施形態において、各サーバ920は、単一のサーバであってもよいし、複数のコンピュータ又は複数のデータセンターに跨る分散サーバであってもよい。サーバ920は、例えば、ウェブサーバ、ニュースサーバ、メールサーバ、メッセージサーバ、広告サーバ、ファイルサーバ、アプリケーションサーバ、交換サーバ、データベースサーバ、又はプロキシサーバなど、様々な種類のサーバとし得る。特定の実施形態において、各サーバ920は、そのサーバ920によって実現あるいは支援される適切な機能を実行するための、ハードウェア、ソフトウェア、若しくは埋込ロジック要素、又は2つ以上のこれら要素の組み合わせを含み得る。例えば、ウェブサーバは一般的に、ウェブページ、又はウェブページの特定の要素、を含むウェブサイトのホストを務めることが可能である。より具体的には、ウェブサーバは、HTMLファイル又はその他のファイルタイプのホストとなることができ、あるいは要求を受けてファイルを動的に作成あるいは構成して、クライアント930からのHTTP又はその他の要求に回答して該ファイルをクライアント930に通信することができる。メールサーバは一般的に、様々なクライアント930に電子メールサービスを提供することができる。データベースサーバは一般的に、1つ以上のデータ庫に格納されたデータを管理するためのインタフェースを提供することができる。

30

【0045】

特定の実施形態において、1つ以上のサーバ920に、1つ以上のリンク950を介して、1つ以上のデータストレージ940が通信可能に結合され得る。特定の実施形態において、データストレージ940は、様々な種類の情報を保管するために使用され得る。特定の実施形態において、データストレージ940に格納される情報は、特定のデータ構造に従って編成され得る。特定の実施形態において、各データストレージ940はリレーショナルデータベースとし得る。特定の実施形態は、サーバ920又はクライアント930が、データストレージ940に格納された情報を管理する(例えば、検索、修正、追加あるいは消去する)ことを可能にするインタフェースを提供し得る。

40

【0046】

特定の実施形態において、各クライアント930は、ハードウェア、ソフトウェア、若しくは埋込ロジック要素、又は2つ以上のこれら要素の組み合わせを含み、クライアント

50

930によって実現あるいは支援される適切な機能を実行することが可能である。例えば、クライアント930は、以下に限られないが、デスクトップ型コンピュータシステム、ノート型コンピュータシステム、ネットブック型コンピュータシステム、手持ち式電子機器、又は携帯電話とし得る。この開示は、如何なる好適なクライアント930をも意図している。クライアント930は、クライアント930の位置のネットワークユーザがネットワーク910にアクセスすることを可能にし得る。クライアント930は、そのユーザが、その他のクライアント930の他ユーザと通信することを可能にしてもよい。

【0047】

クライアント930は、例えば、マイクロソフト・インターネットエクスプローラ、グーグル・クローム又はモジラ・ファイアフォックスなどのウェブブラウザ932を有することができ、また、1つ以上のアドオン、プラグイン、又は例えばツールバー若しくはヤフー・ツールバーなどのその他のエクステンションを有していてもよい。クライアント930のユーザは、URL又はウェブブラウザ932をサーバ920に導くその他のアドレスを入力することができ、ウェブブラウザ932は、ハイパーテキスト転送プロトコル(H T T P)要求を生成して、該H T T P要求をサーバ920に通信し得る。サーバ920は、H T T P要求を受理し、該H T T P要求に応じた1つ以上のハイパーテキスト・マークアップ言語(H T M L)ファイルをクライアント930に通信し得る。クライアント930は、サーバ920からのH T M Lファイルに基づいて、ユーザへの提示のためにウェブページを表示し得る。この開示は、如何なる好適なウェブページファイルをも意図している。非限定的な一例として、ウェブページは、具体的なニーズに従って、H T M Lファイル、拡張可能ハイパーテキスト・マークアップ言語(X H T M L)ファイル、又は拡張マークアップ言語(X M L)ファイルから表示を行い得る。このようなページはまた、例えば、以下に限られないが、J A V Aスクリプト、J A V A、マイクロソフト・シルバライトで記述されたものなどのスクリプト、例えばA J A X (Asynchronous JavaScript and XML)などのマークアップ言語とスクリプトとの組み合わせ、及びこれらに類するものを実行し得る。ここでは、適宜、ウェブページへの言及は、(ブラウザがウェブページを表示するために使用する)1つ以上の対応するウェブページファイルを含むし、その逆も然りである。

【0048】

特定の実施形態は、1つ以上のコンピュータシステム上で実現され得る。図10は、一例に係るコンピュータシステム1000を示している。特定の実施形態において、1つ以上のコンピュータシステム1000が、ここに記載あるいは図示した1つ以上の方法の1つ以上のステップを実行する。特定の実施形態において、1つ以上のコンピュータシステム1000が、ここに記載あるいは図示した機能を提供する。特定の実施形態において、1つ以上のコンピュータシステム1000上で起動されるソフトウェアが、ここに記載あるいは図示した1つ以上の方法の1つ以上のステップを実行し、あるいは、ここに記載あるいは図示した機能を提供する。特定の実施形態は、1つ以上のコンピュータシステム1000の1つ以上の部分を含む。

【0049】

この開示は、如何なる好適な数のコンピュータシステム1000をも意図している。この開示は、如何なる物理形態を採るコンピュータシステム1000をも意図している。非限定的な一例として、コンピュータシステム1000は、組込型コンピュータシステム、システム・オン・チップ(S O C)、シングルボード・コンピュータシステム(S B C)(例えば、コンピュータ・オン・モジュール(C O M)若しくはシステム・オン・モジュール(S O M)など)、デスクトップ型コンピュータシステム、ラップトップ型又はノート型のコンピュータシステム、対話式キオスク端末、メインフレーム、メッシュ状コンピュータシステム、携帯電話、携帯情報端末(P D A)、サーバ、又はこれらの2つ以上の組み合わせとし得る。必要に応じて、コンピュータシステム1000は、単一あるいは分散型の；複数の位置にまたがる；複数の機械にまたがる；あるいはクラウド(1つ以上のネットワーク内に1つ以上のクラウド要素を含み得る)内の；1つ以上のコンピュータシ

10

20

30

40

50

システム 1000 を含んでいてもよい。必要に応じて、1つ以上のコンピュータシステム 1000 は、ここに記載あるいは図示した1つ以上の方法の1つ以上のステップを、空間的あるいは時間的な実質的な制約なく実行し得る。非限定的な一例として、1つ以上のコンピュータシステム 1000 は、ここに記載あるいは図示した1つ以上の方法の1つ以上のステップを、リアルタイムあるいはバッチモードで実行し得る。必要に応じて、1つ以上のコンピュータシステム 1000 は、相異なる時点で、あるいは相異なる位置で、ここに記載あるいは図示した1つ以上の方法の1つ以上のステップを実行してもよい。

【0050】

特定の実施形態において、コンピュータシステム 1000 は、プロセッサ 1002、メモリ 1004、ストレージ 1006、入力/出力 (I/O) インタフェース 1008、通信インタフェース 1010、及びバス 1012 を含んでいる。この開示は、特定の構成にされた特定数の特定の構成要素を有する特定のコンピュータシステムを図示して説明しているが、この開示は、如何なる好適配置にされた如何なる好適数の如何なる好適構成要素を有する如何なる好適なコンピュータシステムをも意図している。

10

【0051】

特定の実施形態において、プロセッサ 1002 は、例えばコンピュータプログラムを構築する命令などの命令を実行するハードウェアを含む。非限定的な一例として、命令を実行するために、プロセッサ 1002 は、内部レジスタ、内部キャッシュ、メモリ 1004、又はストレージ 1006 から命令を取り出し (すなわち、フェッチし) ; それを復号化して実行し ; その後、1つ以上の結果を内部レジスタ、内部キャッシュ、メモリ 1004、又はストレージ 1006 に書き込み得る。特定の実施形態において、プロセッサ 1002 は、データ、命令又はアドレス用の1つ以上の内部キャッシュを含み得る。この開示は、プロセッサ 1002 が必要に応じて、如何なる好適数の如何なる好適な内部キャッシュを含むことも意図している。非限定的な一例として、プロセッサ 1002 は、1つ以上の命令キャッシュと、1つ以上のデータキャッシュと、1つ以上のトランスレーション・ルックアサイド・バッファ (TLB) とを含み得る。命令キャッシュ内の命令は、メモリ 1004 又はストレージ 1006 内の命令の複製としてもよく、命令キャッシュは、プロセッサ 1002 によるこれら命令の取り出しを高速化し得る。データキャッシュ内のデータは、メモリ 1004 又はストレージ 1006 内の、プロセッサ 1002 にて実行されている命令が処理すべきデータの複製 ; プロセッサ 1002 にて実行される後続の命令によるアクセスのための、あるいはメモリ 1004 又はストレージ 1006 に書き込むための、プロセッサ 1002 にて実行された以前の命令の結果 ; 又はその他の好適データとし得る。データキャッシュは、プロセッサ 1002 による読み出し処理又は書き込み処理を高速化し得る。TLB は、プロセッサ 1002 のために仮想アドレス変換を高速化し得る。特定の実施形態において、プロセッサ 1002 は、データ、命令又はアドレス用の1つ以上の内部レジスタを含み得る。この開示は、プロセッサ 1002 が適宜、如何なる好適数の如何なる好適な内部レジスタを含むことも意図している。必要に応じて、プロセッサ 1002 は、1つ以上の演算論理ユニット (ALU) を含んでいてもよく ; マルチコアプロセッサであってもよく ; あるいは1つ以上のプロセッサ 1002 を含んでいてもよい。この開示は特定のプロセッサを図示して説明しているが、この開示は如何なる好適なプロセッサをも意図している。

20

30

40

【0052】

特定の実施形態において、メモリ 1004 は、プロセッサ 1002 が実行する命令、又はプロセッサ 1002 が処理するデータを格納するメインメモリを含む。非限定的な一例として、コンピュータシステム 1000 はストレージ 1006 又は別のソース (例えば、別のコンピュータシステム 1000 など) からメモリ 1004 に命令をロードし得る。その後、プロセッサ 1002 は、メモリ 1004 から内部レジスタ又は内部キャッシュに命令をロードし得る。命令を実行するため、プロセッサ 1002 は、内部レジスタ又は内部キャッシュから命令を取り出し、取り出した命令を復号化する。命令の実行中又は実行後、プロセッサ 1002 は、1つ以上の結果 (中間結果又は最終結果とし得る) を内部レジ

50

スタ又は内部キャッシュに書き込み得る。そして、プロセッサ1002は、それらの結果のうちの一つ以上をメモリ1004に書き込み得る。特定の実施形態において、プロセッサ1002は、一つ以上の内部レジスタ若しくは内部キャッシュ又はメモリ1004内の命令のみを実行してもよく（ストレージ1006又はその他の場所の命令を実行するのとは異なる）、また、一つ以上の内部レジスタ若しくは内部キャッシュ又はメモリ1004内のデータのみを処理してもよい（ストレージ1006又はその他の場所のデータを処理するのとは異なる）。一つ以上のメモリバス212（各々がアドレスバス及びデータバスを含み得る）が、プロセッサ1002をメモリ1004に結合してもよい。バス1012は、後述のように、一つ以上のメモリバスを含み得る。特定の実施形態において、一つ以上のメモリ管理装置（MMU）が、プロセッサ1002とメモリ1004との間に存在し、プロセッサ1002により要求されるメモリ1004へのアクセスを容易にし得る。特定の実施形態において、メモリ1004は、ランダムアクセスメモリ（RAM）を含む。このRAMは、適当であれば、揮発性メモリであってもよい。必要に応じて、このRAMはダイナミックRAM（DRAM）又はスタティックRAM（SRAM）にされ得る。また、必要に応じて、このRAMは、シングルポート又はマルチポートのRAMにされ得る。この開示は、如何なる好適なRAMをも意図している。メモリ1004は、必要に応じて、一つ以上のメモリ1004を含んでいてもよい。この開示は特定のメモリを図示して説明しているが、この開示は如何なる好適なメモリをも意図している。

10

【0053】

特定の実施形態において、ストレージ1006は、データ又は命令用の大容量記憶装置を含む。非限定的な一例として、ストレージ1006は、HDD、フロッピーディスク（登録商標）ドライブ、フラッシュメモリ、光ディスク、磁気光ディスク、磁気テープ、若しくはユニバーサル・シリアル・バス（USB）ドライブ、又はこれらの2つ以上の組み合わせを含み得る。ストレージ1006は、必要に応じて、リムーバブル（取り外し可能）あるいは非リムーバブル（すなわち、固定）のメディアを含み得る。ストレージ1006は、必要に応じて、コンピュータシステム1000の内部にあってもよいし、外部にあってもよい。特定の実施形態において、ストレージ1006は、不揮発性のソリッドステートメモリである。特定の実施形態において、ストレージ1006は、読み出し専用メモリ（ROM）を含む。必要に応じて、このROMは、マスクプログラムされたROM、プログラマブルROM（PROM）、消去可能PROM（EPROM）、電気的消去可能PROM（EEPROM）、電気的消去再書込可能ROM（EAROM）、若しくはフラッシュメモリ、又はこれらの2つ以上の組み合わせとし得る。この開示は、如何なる好適な物理形態を有する大容量記憶装置1006をも意図している。ストレージ1006は、必要に応じて、プロセッサ1002とストレージ1006との間での通信を容易にする一つ以上のストレージ制御装置を含み得る。この開示は特定のストレージを図示して説明しているが、この開示は如何なる好適なストレージをも意図している。

20

30

【0054】

特定の実施形態において、I/Oインタフェース1008は、コンピュータシステム1000と一つ以上のI/O装置との間での通信のための一つ以上のインタフェースを提供するハードウェア、ソフトウェア、又はこれら双方を含む。コンピュータシステム1000は、必要に応じて、これらI/O装置のうちの一つ以上を含んでいてもよい。これらI/O装置のうちの一つ以上は、個人とコンピュータシステム1000との間での通信を可能にし得る。非限定的な一例として、I/O装置は、キーボード、キーパッド、マイク、モニタ、マウス、プリンタ、スキャナ、スピーカ、スチルカメラ、スタイラスペン、タブレット、タッチスクリーン、トラックボール、ビデオカメラ、その他の好適なI/O装置、又はこれらの2つ以上の組み合わせを含み得る。I/O装置は、一つ以上のセンサを含んでいてもよい。この開示は、如何なる好適なI/O装置、及びそれ用の如何なる好適なI/Oインタフェース1008をも意図している。必要に応じて、I/Oインタフェース1008は、プロセッサ1002がこれらのI/O装置のうちの一つ以上を駆動することを可能にする一つ以上の装置又はソフトウェアドライバを含んでいてもよい。I/Oイン

40

50

タフェース1008は、必要に応じて、1つ以上のインタフェース1008を含んでもよい。この開示は特定のI/Oインタフェースを図示して説明しているが、この開示は如何なる好適なI/Oインタフェースをも意図している。

【0055】

特定の実施形態において、通信インタフェース1010は、コンピュータシステム1000と、1つ以上のその他のコンピュータシステム若しくは1つ以上のネットワークとの間での通信（例えば、パケットベースの通信など）のための1つ以上のインタフェースを提供するハードウェア、ソフトウェア、又はこれら双方を含む。非限定的な一例として、通信インタフェース1010は、イーサネット若しくはその他のワイヤベースのネットワークと通信するためのネットワークインタフェースコントローラ（NIC）若しくはネットワークアダプタ、又は例えばWi-Fiネットワークなどの無線ネットワークと通信するための無線NIC（WNIC）若しくは無線アダプタを含み得る。この開示は、如何なる好適なネットワーク及びそれ用の如何なる好適な通信インタフェース1010をも意図している。非限定的な一例として、コンピュータシステム1000は、アドホックネットワーク、パーソナルエリアネットワーク（PAN）、ローカルエリアネットワーク（LAN）、ワイドエリアネットワーク（WAN）、メトロポリタンエリアネットワーク（MAN）、若しくはインターネットの1つ以上の部分、又はこれらの2つ以上の組み合わせと通信し得る。これらのネットワークのうち1つ以上の1つ以上の部分は、有線であってもよいし、無線であってもよい。一例として、コンピュータシステム1000は、無線PAN（WPAN）（例えば、ブルートゥースWPANなど）、Wi-Fiネットワーク、Wi-MAXネットワーク、携帯電話ネットワーク（例えば、グローバル・システム・フォー・モバイル・コミュニケーションズ（GSM）ネットワークなど）、若しくはその他の好適な無線ネットワーク、又はこれらの2つ以上の組み合わせと通信し得る。コンピュータシステム1000は、必要に応じて、これらのネットワークの何れかのための如何なる好適なインタフェース1010を含んでもよい。この開示は特定の通信インタフェースを図示して説明しているが、この開示は如何なる好適な通信インタフェースをも意図している。

10

20

【0056】

特定の実施形態において、バス1012は、コンピュータシステム1000の構成要素を相互に結合するための、ハードウェア、ソフトウェア、又はこれら双方を含む。非限定的な一例として、バス1012は、アクセラレーテッド・グラフィクス・ポート（AGP）若しくはその他のグラフィック専用バス、EISA（Extended Industry Standard Architecture）バス、フロントサイドバス（FSB）、ハイパートランスポート（HT）インターコネクト、業界標準アーキテクチャ（ISA）バス、インフィニバンド・インターコネクト、LPC（low-pin-count）バス、メモリバス、マイクロ・チャンネル・アーキテクチャ（MCA）バス、コンピュータ用拡張バスアーキテクチャ（PCI）バス、PCIエクスプレス（PCI-X）バス、シリアルATA（SATA）バス、VLB（Video Electronics Standards Association local）バス、若しくはその他の好適なバス、又はこれらの2つ以上の組み合わせを含み得る。バス1012は、必要に応じて、1つ以上のバス1012を含んでもよい。この開示は特定のバスを図示して説明しているが、この開示は如何なる好適なバス又はインターコネクトをも意図している。

30

40

【0057】

ここで、コンピュータ読み取り可能記憶媒体への言及は、1つ以上の一過性でない有形のコンピュータ読み取り可能記憶媒体保有構造をも包含するものである。非限定的な一例として、コンピュータ読み取り可能記憶媒体は、必要に応じて、半導体ベースあるいはその他の集積回路（IC）（例えば、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IC（ASIC）など）、ハードディスク、HDD、ハイブリッド式ハードドライブ（HHD）、光ディスク、光ディスクドライブ（ODD）、磁気光ディスク、磁気光ドライブ、フロッピーディスク（登録商標）、フロッピーディスクドライブ（FDD）、磁気テープ、ホログラフィック記憶媒体、ソリッドステートドライブ（

50

SSD)、RAMドライブ、セキュアデジタル(SD)カード、セキュアデジタルドライブ、若しくはその他の好適なコンピュータ読み取り可能記憶媒体、又はこれらの2つ以上の組み合わせを含み得る。

【0058】

この開示は、何らかの好適なストレージを実現する1つ以上のコンピュータ読み取り可能記憶媒体を意図している。特定の実施形態において、コンピュータ読み取り可能記憶媒体は、必要に応じて、プロセッサ1002の1つ以上の部分(例えば、1つ以上の内部レジスタ若しくはキャッシュなど)、メモリ1004の1つ以上の部分、ストレージ1006の1つ以上の部分、又はこれらの組み合わせを実現してもよい。特定の実施形態において、コンピュータ読み取り可能記憶媒体はRAM又はROMを実現する。特定の実施形態において、コンピュータ読み取り可能記憶媒体は揮発性あるいは永続的なメモリを実現する。特定の実施形態において、1つ以上のコンピュータ読み取り可能記憶媒体がソフトウェアを具現化する。ここで、ソフトウェアへの言及は適宜、1つ以上のアプリケーション、バイトコード、1つ以上のコンピュータプログラム、1つ以上の実行ファイル、1つ以上の命令、ロジック、機械語、1つ以上のスクリプト、又はソースコードを包含するものであり、この逆もまた然りである。特定の実施形態において、ソフトウェアは、1つ以上のアプリケーション・プログラミング・インタフェース(API)を含む。この開示は、如何なる好適なプログラミング言語又は複数のプログラミング言語の組み合わせで書き込まれ、あるいはその他の方法で表現された、如何なる好適なソフトウェアをも意図している。特定の実施形態において、ソフトウェアはソースコード又はオブジェクトコードとして表現され得る。特定の実施形態において、ソフトウェアは、例えばC、Perl又はこれらの好適な拡張版など、より高級(高水準)なプログラミング言語で表現される。特定の実施形態において、ソフトウェアは、例えばアセンブリ言語(又は機械語)など、より低水準のプログラミング言語で表現される。特定の実施形態において、ソフトウェアはJAVAVで表現される。特定の実施形態において、ソフトウェアはハイパーテキスト・マークアップ言語(HTML)、拡張マークアップ言語(XML)、又はその他の好適なマークアップ言語で表現される。

10

20

【0059】

この開示は、ここで例示した実施形態への、当業者が理解されるであろう全ての変更、代用、変形、改変及び改良を包含するものである。同様に、添付の請求項は適宜、ここで例示した実施形態への当業者が理解されるであろう全ての変更、代用、変形、改変及び改良を包含するものである。

30

【0060】

以上の説明に関し、更に以下の付記を開示する。

(付記1)

1つ以上の計算装置によって実行される方法であって、
ソフトウェアモジュールにアクセスするステップであり、

前記ソフトウェアモジュールは、

1つ以上の入力変数と、

1つ以上の中間変数であり、各中間変数の値が前記入力変数のうちの少なくとも1つの入力変数の値に直接的あるいは間接的に依存する、1つ以上の中間変数と、

40

1つ以上の条件付き分岐点であり、各条件付き分岐点が、前記入力変数及び前記中間変数のうちの少なくとも1つの入力変数及び/又は中間変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも1つの入力変数及び/又は中間変数の値に依存する、1つ以上の条件付き分岐点と、を有し、且つ

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進む、アクセスするステップと、

前記1つ以上の入力変数に対応する1つ以上の変数セットを構築するステップであり、各変数セットが、前記入力変数のうちの1つと、前記入力変数のうちの該1つの値に直接

50

的あるいは間接的に依存する値を有する中間変数とを含む、構築するステップと、

各変数セットに関して、該変数セットの前記入力変数及び前記中間変数によって影響される平均分岐深さを計算するステップと、

前記1つ以上の変数セットのうち最小の平均分岐深さを有する第1変数セットを選択するステップと、

前記第1変数セットの前記入力変数に第1記号値を割り当てるステップと、

前記第1記号値を有する前記第1変数セットの前記入力変数を用いて、前記ソフトウェアモジュールを記号的に実行し、前記第1変数セットの前記入力変数及び前記中間変数に関する第1テストケースセットを生成するステップと、

前記第1テストケースセットを用いて前記ソフトウェアモジュールを検証するステップと、

を有する方法。

(付記2)

前記入力変数の各々に関して、前記中間変数のうちの何れが該入力変数の値に直接的あるいは間接的に依存する値を有するかを、関連性分析を用いて決定するステップ、を更に有する付記1に記載の方法。

(付記3)

前記ソフトウェアモジュールは1以上の分岐階層を有し、

前記分岐階層の各々は、前記条件付き分岐点のうち少なくとも1つを有し、

各変数セットに関して、該変数セットの前記入力変数及び前記中間変数によって影響される前記平均分岐深さを計算する前記ステップは、

該変数セットの前記入力変数に関して、分岐条件が該入力変数に依存する条件付き分岐点を有する分岐階層の数を計算するステップと、

該変数セットの前記中間変数の各々に関して、分岐条件が該中間変数に依存する条件付き分岐点を有する分岐階層の数を計算するステップと、

これらの分岐階層の数の平均を、該変数セットの前記平均分岐深さとして計算するステップと

を有する、

付記1又は2に記載の方法。

(付記4)

最小の平均分岐深さを有する前記第1変数セットを選択するとき、最小の平均分岐深さを有する変数セットが複数存在する場合、前記第1変数セットとして、最小の平均分岐深さを有する変数セットのうち1つをランダムに選択する、付記1乃至3の何れかーに記載の方法。

(付記5)

前記1つ以上の変数セットのうち、次に最小の平均分岐深さを有する第2変数セットを選択するステップと、

前記第2変数セットの前記入力変数に第2記号値を割り当てるステップと、

前記第2記号値を有する前記第2変数セットの前記入力変数を用いて、前記ソフトウェアモジュールを記号的に実行し、前記第2変数セットの前記入力変数及び前記中間変数に関する第2テストケースセットを生成するステップと、

前記第2テストケースセットを用いて前記ソフトウェアモジュールを検証するステップと、

を更に有する付記1乃至4の何れかーに記載の方法。

(付記6)

1つ以上の計算装置によって実行される方法であって、

ソフトウェアモジュール及び該ソフトウェアモジュール用の複数のテストケースにアクセスするステップであり、

前記ソフトウェアモジュールは、

1つ以上の変数と、

10

20

30

40

50

1つ以上の条件付き分岐点であり、各条件付き分岐点が、前記変数のうちの少なくとも1つの変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも1つの変数の値に依存する、1つ以上の条件付き分岐点と、を有し、

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進み、

前記テストケースの各々は、前記条件付き分岐点のうちの少なくとも1つをカバーし、且つ前記条件付き分岐点のうちの前記少なくとも1つの分岐条件を規定する前記少なくとも1つの変数に関する少なくとも1つのテスト値を有する、アクセスするステップと、

前記条件付き分岐点の各々に関して、前記複数のテストケースのうちに、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースが複数存在する場合に、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースのうち1つを除く全てを、前記複数のテストケースから除去するステップと、

残りのテストケースから1つ以上の第1テストケースを選択するステップであり、各第1テストケースは、前記条件付き分岐点のうちの少なくとも1つが該第1テストケースによってのみカバーされるテストケースである、選択するステップと、

前記条件付き分岐点の全てが前記第1テストケース及び第2テストケースのうちの少なくとも1つのテストケースによってカバーされるまで、残りのテストケースから1つ以上の第2テストケースを繰り返し選択するステップであり、各繰り返しにおいて選択される第2テストケースは、該繰り返しにおいて最も多数の条件付き分岐点をカバーするテストケースである、選択するステップと、

選択された前記第1テストケース及び前記第2テストケースのみを用いて、前記ソフトウェアモジュールをテストするステップと、

を有する方法。

(付記7)

前記ソフトウェアモジュールの前記変数は、1つ以上の入力変数と1つ以上の中間変数とを有し、各中間変数の値は、前記入力変数のうちの少なくとも1つの入力変数の値に直接的あるいは間接的に依存する、付記6に記載の方法。

(付記8)

前記入力変数と前記入力変数に関する前記テストケースからの前記テスト値とを表すグラフを構築するステップを更に有し、

前記グラフは、

各階層が前記入力変数のうちの1つに関連付けられた、前記1つ以上の入力変数に対応する1つ以上の階層と、

前記階層の各々の1つ以上のノードと、

1つ以上のエッジであり、各エッジが、隣接する2つの階層からの2つのノードを結び付け、且つ前記隣接する2つの階層のうちの高い側の階層に関連付けられた入力変数の前記テスト値のうちの1つを表す、1つ以上のエッジと、

を有する、

付記7に記載の方法。

(付記9)

前記テストケースと前記条件付き分岐点とを表す二部グラフを構築するステップを更に有し、

前記二部グラフは、

各第1ノードが前記テストケースのうちの1つを表す、前記複数のテストケースに対応する複数の第1ノードと、

各第2ノードが前記条件付き分岐点のうちの1つを表す、前記1つ以上の条件付き分岐点に対応する1つ以上の第2ノードと、

各エッジが前記第1ノードのうちの1つと前記第2ノードのうちの1つとを結び付け

10

20

30

40

50

る複数のエッジであり、前記第 1 ノードのうちの該 1 つによって表されるテストケースが、前記第 2 ノードのうちの該 1 つによって表される条件付き分岐点をカバーする、複数のエッジと

を有する、

付記 6 乃至 8 の何れか一に記載の方法。

(付記 10)

1 つ以上のプロセッサによって実行可能な命令を有するメモリと、

前記メモリに結合され、前記命令を実行するよう動作する 1 つ以上のプロセッサとを有するシステムであって、

前記 1 つ以上のプロセッサは、前記命令を実行するとき、

- ソフトウェアモジュールにアクセスし、

前記ソフトウェアモジュールは、

- 1 つ以上の入力変数と、

1 つ以上の中間変数であり、各中間変数の値が前記入力変数のうちの少なくとも 1 つの入力変数の値に直接的あるいは間接的に依存する、1 つ以上の中間変数と、

1 つ以上の条件付き分岐点であり、各条件付き分岐点が、前記入力変数及び前記中間変数のうちの少なくとも 1 つの入力変数及び / 又は中間変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも 1 つの入力変数及び / 又は中間変数の値に依存する、1 つ以上の条件付き分岐点と、を有し、且つ

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進み、

- 前記 1 つ以上の入力変数に対応する 1 つ以上の変数セットを構築し、各変数セットは、前記入力変数のうちの 1 つと、前記入力変数のうちの該 1 つの値に直接的あるいは間接的に依存する値を有する中間変数とを含み、

- 各変数セットに関して、該変数セットの前記入力変数及び前記中間変数によって影響される平均分岐深さを計算し、

- 前記 1 つ以上の変数セットのうち最小の平均分岐深さを有する第 1 変数セットを選択し、

- 前記第 1 変数セットの前記入力変数に第 1 記号値を割り当て、

- 前記第 1 記号値を有する前記第 1 変数セットの前記入力変数を用いて、前記ソフトウェアモジュールを記号的に実行することで、前記第 1 変数セットの前記入力変数及び前記中間変数に関する第 1 テストケースセットを生成し、

- 前記第 1 テストケースセットを用いて前記ソフトウェアモジュールを検証する、よう動作する、システム。

(付記 11)

前記 1 つ以上のプロセッサは更に、前記命令を実行するとき、前記入力変数の各々に関して、前記中間変数のうちの何れが該入力変数の値に直接的あるいは間接的に依存する値を有するかを、関連性分析を用いて決定するよう動作する、付記 10 に記載のシステム。

(付記 12)

前記ソフトウェアモジュールは 1 以上の分岐階層を有し、

前記分岐階層の各々は、前記条件付き分岐点のうちの少なくとも 1 つを有し、

各変数セットに関して、該変数セットの前記入力変数及び前記中間変数によって影響される前記平均分岐深さを計算することは、

該変数セットの前記入力変数に関して、分岐条件が該入力変数に依存する条件付き分岐点を有する分岐階層の数を計算し、

該変数セットの前記中間変数の各々に関して、分岐条件が該中間変数に依存する条件付き分岐点を有する分岐階層の数を計算し、

これらの分岐階層の数の平均を、該変数セットの前記平均分岐深さとして計算することを有する、

10

20

30

40

50

付記 10 又は 11 に記載のシステム。

(付記 13)

最小の平均分岐深さを有する前記第 1 変数セットを選択するとき、最小の平均分岐深さを有する変数セットが複数存在する場合、前記第 1 変数セットとして、最小の平均分岐深さを有する変数セットのうちの一つをランダムに選択する、付記 10 乃至 12 の何れかーに記載のシステム。

(付記 14)

前記 1 つ以上のプロセッサは更に、前記命令を実行するとき、

前記 1 つ以上の変数セットのうち、次に最小の平均分岐深さを有する第 2 変数セットを選択し、

前記第 2 変数セットの前記入力変数に第 2 記号値を割り当て、

前記第 2 記号値を有する前記第 2 変数セットの前記入力変数を用いて、前記ソフトウェアモジュールを記号的に実行することで、前記第 2 変数セットの前記入力変数及び前記中間変数に関する第 2 テストケースセットを生成し、

前記第 2 テストケースセットを用いて前記ソフトウェアモジュールを検証する、

よう動作する、付記 10 乃至 13 の何れかーに記載のシステム。

(付記 15)

1 つ以上のプロセッサによって実行可能な命令を有するメモリと、

前記メモリに結合され、前記命令を実行するよう動作する 1 つ以上のプロセッサとを有するシステムであって、

前記 1 つ以上のプロセッサは、前記命令を実行するとき、

- ソフトウェアモジュール及び該ソフトウェアモジュール用の複数のテストケースにアクセスし、

前記ソフトウェアモジュールは、

1 つ以上の変数と、

1 つ以上の条件付き分岐点であり、各条件付き分岐点が、前記変数のうちの少なくとも 1 つの変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも 1 つの変数の値に依存する、1 つ以上の条件付き分岐点と、を有し、

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進み、

前記テストケースの各々は、前記条件付き分岐点のうち少なくとも 1 つをカバーし、且つ前記条件付き分岐点のうちの前記少なくとも 1 つの分岐条件を規定する前記少なくとも 1 つの変数に関する少なくとも 1 つのテスト値を有し、

- 前記条件付き分岐点の各々に関して、前記複数のテストケースのうちに、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースが複数存在する場合に、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースのうち一つを除く全てを、前記複数のテストケースから除去し、

- 残りのテストケースから 1 つ以上の第 1 テストケースを選択し、各第 1 テストケースは、前記条件付き分岐点のうち少なくとも 1 つが該第 1 テストケースによってのみカバーされるテストケースであり、

- 前記条件付き分岐点の全てが前記第 1 テストケース及び第 2 テストケースのうち少なくとも 1 つのテストケースによってカバーされるまで、残りのテストケースから 1 つ以上の第 2 テストケースを繰り返し選択し、各繰り返しにおいて選択される第 2 テストケースは、該繰り返しにおいて最も多数の条件付き分岐点をカバーするテストケースであり、

- 選択された前記第 1 テストケース及び前記第 2 テストケースのみを用いて、前記ソフトウェアモジュールをテストする、

よう動作する、システム。

10

20

30

40

50

(付記 16)

前記ソフトウェアモジュールの前記変数は、1つ以上の入力変数と1つ以上の中間変数とを有し、各中間変数の値は、前記入力変数のうちの少なくとも1つの入力変数の値に直接的あるいは間接的に依存する、付記 15 に記載のシステム。

(付記 17)

前記1つ以上のプロセッサは更に、前記命令を実行するとき、前記入力変数と前記入力変数に関する前記テストケースからの前記テスト値とを表すグラフを構築するよう動作し

、
前記グラフは、

各階層が前記入力変数のうちの1つに関連付けられた、前記1つ以上の入力変数に対応する1つ以上の階層と、

前記階層の各々の1つ以上のノードと、

1つ以上のエッジであり、各エッジが、隣接する2つの階層からの2つのノードを結び付け、且つ前記隣接する2つの階層のうちの高い側の階層に関連付けられた入力変数の前記テスト値のうちの1つを表す、1つ以上のエッジと、

を有する、

付記 16 に記載のシステム。

(付記 18)

前記1つ以上のプロセッサは更に、前記命令を実行するとき、前記テストケースと前記条件付き分岐点とを表す二部グラフを構築するよう動作し、

前記二部グラフは、

各第1ノードが前記テストケースのうちの1つを表す、前記複数のテストケースに対応する複数の第1ノードと、

各第2ノードが前記条件付き分岐点のうちの1つを表す、前記1つ以上の条件付き分岐点に対応する1つ以上の第2ノードと、

各エッジが前記第1ノードのうちの1つと前記第2ノードのうちの1つとを結び付ける複数のエッジであり、前記第1ノードのうちの該1つによって表されるテストケースが、前記第2ノードのうちの該1つによって表される条件付き分岐点をカバーする、複数のエッジと

を有する、

付記 15 乃至 17 の何れか一に記載のシステム。

(付記 19)

プログラムを格納したコンピュータ読み取り可能記憶媒体であって、前記プログラムは、1つ以上の計算装置によって実行されるとき、該1つ以上の計算装置に、

- ソフトウェアモジュールにアクセスさせ、

前記ソフトウェアモジュールは、

1つ以上の入力変数と、

1つ以上の中間変数であり、各中間変数の値が前記入力変数のうちの少なくとも1つの入力変数の値に直接的あるいは間接的に依存する、1つ以上の中間変数と、

1つ以上の条件付き分岐点であり、各条件付き分岐点が、前記入力変数及び前記中間変数のうちの少なくとも1つの入力変数及び/又は中間変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも1つの入力変数及び/又は中間変数の値に依存する、1つ以上の条件付き分岐点と、を有し、且つ

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進み、

- 前記1つ以上の入力変数に対応する1つ以上の変数セットを構築させ、各変数セットは、前記入力変数のうちの1つと、前記入力変数のうちの該1つの値に直接的あるいは間接的に依存する値を有する中間変数とを含み、

- 各変数セットに関して、該変数セットの前記入力変数及び前記中間変数によって影

10

20

30

40

50

響される平均分岐深さを計算させ、

- 前記 1 つ以上の変数セットのうち最小の平均分岐深さを有する第 1 変数セットを選択させ、

- 前記第 1 変数セットの前記入力変数に第 1 記号値を割り当てさせ、

- 前記第 1 記号値を有する前記第 1 変数セットの前記入力変数を用いて、前記ソフトウェアモジュールを記号的に実行させて、前記第 1 変数セットの前記入力変数及び前記中間変数に関する第 1 テストケースセットを生成させ、

- 前記第 1 テストケースセットを用いて前記ソフトウェアモジュールを検証させる、コンピュータ読み取り可能記憶媒体。

(付記 20)

前記プログラムは更に、前記 1 つ以上の計算装置によって実行されるとき、前記 1 つ以上の計算装置に、前記入力変数の各々に関して、前記中間変数のうちの何れが該入力変数の値に直接的あるいは間接的に依存する値を有するかを、関連性分析を用いて決定させる、付記 19 に記載のコンピュータ読み取り可能記憶媒体。

(付記 21)

前記ソフトウェアモジュールは 1 以上の分岐階層を有し、

前記分岐階層の各々は、前記条件付き分岐点のうち少なくとも 1 つを有し、

各変数セットに関して、該変数セットの前記入力変数及び前記中間変数によって影響される前記平均分岐深さを計算させることは、

該変数セットの前記入力変数に関して、分岐条件が該入力変数に依存する条件付き分岐点を有する分岐階層の数を計算させ、

該変数セットの前記中間変数の各々に関して、分岐条件が該中間変数に依存する条件付き分岐点を有する分岐階層の数を計算させ、

これらの分岐階層の数の平均を、該変数セットの前記平均分岐深さとして計算させることを有する、

付記 19 又は 20 に記載のコンピュータ読み取り可能記憶媒体。

(付記 22)

最小の平均分岐深さを有する前記第 1 変数セットを選択させるとき、最小の平均分岐深さを有する変数セットが複数存在する場合、前記第 1 変数セットとして、最小の平均分岐深さを有する変数セットのうちの一つをランダムに選択させる、付記 19 乃至 21 の何れか一に記載のコンピュータ読み取り可能記憶媒体。

(付記 23)

前記プログラムは更に、前記 1 つ以上の計算装置によって実行されるとき、前記 1 つ以上の計算装置に、

前記 1 つ以上の変数セットのうち、次に最小の平均分岐深さを有する第 2 変数セットを選択させ、

前記第 2 変数セットの前記入力変数に第 2 記号値を割り当てさせ、

前記第 2 記号値を有する前記第 2 変数セットの前記入力変数を用いて、前記ソフトウェアモジュールを記号的に実行させて、前記第 2 変数セットの前記入力変数及び前記中間変数に関する第 2 テストケースセットを生成させ、

前記第 2 テストケースセットを用いて前記ソフトウェアモジュールを検証させる、

付記 19 乃至 22 の何れか一に記載のコンピュータ読み取り可能記憶媒体。

(付記 24)

プログラムを格納したコンピュータ読み取り可能記憶媒体であって、前記プログラムは、1 つ以上の計算装置によって実行されるとき、該 1 つ以上の計算装置に、

- ソフトウェアモジュール及び該ソフトウェアモジュール用の複数のテストケースにアクセスさせ、

前記ソフトウェアモジュールは、

1 つ以上の変数と、

1 つ以上の条件付き分岐点であり、各条件付き分岐点が、前記変数のうちの少なく

10

20

30

40

50

とも1つの変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも1つの変数の値に依存する、1つ以上の条件付き分岐点と、を有し、

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進み、

前記テストケースの各々は、前記条件付き分岐点のうちの少なくとも1つをカバーし、且つ前記条件付き分岐点のうちの前記少なくとも1つの分岐条件を規定する前記少なくとも1つの変数に関する少なくとも1つのテスト値を有し、

- 前記条件付き分岐点の各々に関して、前記複数のテストケースのうちに、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースが複数存在する場合に、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースのうち1つを除く全てを、前記複数のテストケースから除去させ、

- 残りのテストケースから1つ以上の第1テストケースを選択させ、各第1テストケースは、前記条件付き分岐点のうちの少なくとも1つが該第1テストケースによってのみカバーされるテストケースであり、

- 前記条件付き分岐点の全てが前記第1テストケース及び第2テストケースのうちの少なくとも1つのテストケースによってカバーされるまで、残りのテストケースから1つ以上の第2テストケースを繰り返し選択させ、各繰り返しにおいて選択される第2テストケースは、該繰り返しにおいて最も多数の条件付き分岐点をカバーするテストケースであり、

- 選択された前記第1テストケース及び前記第2テストケースのみを用いて、前記ソフトウェアモジュールをテストさせる、

コンピュータ読み取り可能記憶媒体。

(付記25)

前記ソフトウェアモジュールの前記変数は、1つ以上の入力変数と1つ以上の中間変数とを有し、各中間変数の値は、前記入力変数のうちの少なくとも1つの入力変数の値に直接的あるいは間接的に依存する、付記24に記載のコンピュータ読み取り可能記憶媒体。

(付記26)

前記プログラムは更に、前記1つ以上の計算装置によって実行されるとき、前記1つ以上の計算装置に、前記入力変数と前記入力変数に関する前記テストケースからの前記テスト値とを表すグラフを構築させ、

前記グラフは、

各階層が前記入力変数のうちの1つに関連付けられた、前記1つ以上の入力変数に対応する1つ以上の階層と、

前記階層の各々の1つ以上のノードと、

1つ以上のエッジであり、各エッジが、隣接する2つの階層からの2つのノードを結び付け、且つ前記隣接する2つの階層のうちの高い側の階層に関連付けられた入力変数の前記テスト値のうちの1つを表す、1つ以上のエッジと、

を有する、

付記25に記載のコンピュータ読み取り可能記憶媒体。

(付記27)

前記プログラムは更に、前記1つ以上の計算装置によって実行されるとき、前記1つ以上の計算装置に、前記テストケースと前記条件付き分岐点とを表す二部グラフを構築させ、

前記二部グラフは、

各第1ノードが前記テストケースのうちの1つを表す、前記複数のテストケースに対応する複数の第1ノードと、

各第2ノードが前記条件付き分岐点のうちの1つを表す、前記1つ以上の条件付き分岐点に対応する1つ以上の第2ノードと、

10

20

30

40

50

各エッジが前記第 1 ノードのうちの 1 つと前記第 2 ノードのうちの 1 つとを結び付ける複数のエッジであり、前記第 1 ノードのうちの該 1 つによって表されるテストケースが、前記第 2 ノードのうちの該 1 つによって表される条件付き分岐点をカバーする、複数のエッジと

を有する、

付記 2 4 乃至 2 6 の何れか一に記載のシステム。

(付記 2 8)

ソフトウェアモジュールにアクセスする手段であり、

前記ソフトウェアモジュールは、

1 つ以上の入力変数と、

1 つ以上の中間変数であり、各中間変数の値が前記入力変数のうちの少なくとも 1 つの入力変数の値に直接的あるいは間接的に依存する、1 つ以上の中間変数と、

1 つ以上の条件付き分岐点であり、各条件付き分岐点が、前記入力変数及び前記中間変数のうちの少なくとも 1 つの入力変数及び / 又は中間変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも 1 つの入力変数及び / 又は中間変数の値に依存する、1 つ以上の条件付き分岐点と、を有し、且つ

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進む、アクセスする手段と、

前記 1 つ以上の入力変数に対応する 1 つ以上の変数セットを構築する手段であり、各変数セットが、前記入力変数のうちの 1 つと、前記入力変数のうちの該 1 つの値に直接的あるいは間接的に依存する値を有する中間変数とを含む、構築する手段と、

各変数セットに関して、該変数セットの前記入力変数及び前記中間変数によって影響される平均分岐深さを計算する手段と、

前記 1 つ以上の変数セットのうち最小の平均分岐深さを有する第 1 変数セットを選択する手段と、

前記第 1 変数セットの前記入力変数に第 1 記号値を割り当てる手段と、

前記第 1 記号値を有する前記第 1 変数セットの前記入力変数を用いて、前記ソフトウェアモジュールを記号的に実行し、前記第 1 変数セットの前記入力変数及び前記中間変数に関する第 1 テストケースセットを生成する手段と、

前記第 1 テストケースセットを用いて前記ソフトウェアモジュールを検証する手段と、を有するシステム。

(付記 2 9)

ソフトウェアモジュール及び該ソフトウェアモジュール用の複数のテストケースにアクセスする手段であり、

前記ソフトウェアモジュールは、

1 つ以上の変数と、

1 つ以上の条件付き分岐点であり、各条件付き分岐点が、前記変数のうちの少なくとも 1 つの変数によって規定される分岐条件を有し、該分岐条件が満たされるかが、前記少なくとも 1 つの変数の値に依存する、1 つ以上の条件付き分岐点と、を有し、

前記ソフトウェアモジュールの実行中に、前記条件付き分岐点の各々において、前記ソフトウェアモジュールは、該条件付き分岐点の分岐条件が満たされるかに応じて異なる経路に進み、

前記テストケースの各々は、前記条件付き分岐点のうちの少なくとも 1 つをカバーし、且つ前記条件付き分岐点のうちの前記少なくとも 1 つの分岐条件を規定する前記少なくとも 1 つの変数に関する少なくとも 1 つのテスト値を有する、アクセスする手段と、

前記条件付き分岐点の各々に関して、前記複数のテストケースのうちに、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースが複数存在する場合に、該条件付き分岐点をカバーし且つ前記ソフトウェアモジュールを実行中に同一の経路に進ませるテストケースのうち 1 つを除く全てを、前記複数の

10

20

30

40

50

テストケースから除去する手段と、

残りのテストケースから1つ以上の第1テストケースを選択する手段であり、各第1テストケースは、前記条件付き分岐点のうちの少なくとも1つが該第1テストケースによってのみカバーされるテストケースである、選択する手段と、

前記条件付き分岐点の全てが前記第1テストケース及び第2テストケースのうちの少なくとも1つのテストケースによってカバーされるまで、残りのテストケースから1つ以上の第2テストケースを繰り返し選択する手段であり、各繰り返しにおいて選択される第2テストケースは、該繰り返しにおいて最も多数の条件付き分岐点をカバーするテストケースである、選択する手段と、

選択された前記第1テストケース及び前記第2テストケースのみを用いて、前記ソフトウェアモジュールをテストする手段と、

を有するシステム。

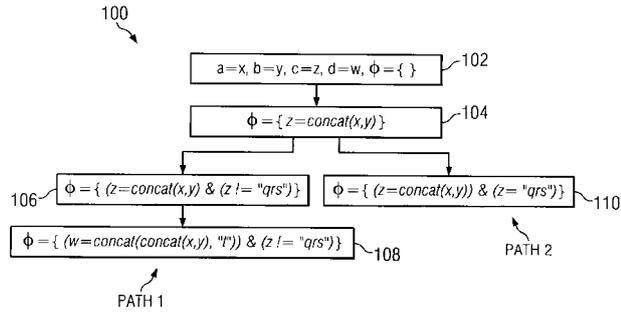
【符号の説明】

【0061】

300	システム	
302	関連性分析部	
304	分岐深さ計算部	
306	記号的実行部	
312	ソフトウェアモジュール	
314	テストケースセット	20
600	システム	
602	動的テストケース生成部	
604	二部グラフ構築部	
606	テストケース選択部	
612	テストケースセット	
614	選択されたテストケース	
700	グラフ	
702、704、706、708、710	ノード	
800	二部グラフ	
812、814、816	ノード(テストケース)	30
822、824、826、828	ノード(条件付き分岐点)	
900	ネットワーク環境	
910	ネットワーク	
920	サーバ	
930	クライアント	
932	ブラウザ	
940	データストレージ	
950	リンク	
1000	コンピュータシステム	
1002	プロセッサ	40
1004	メモリ	
1006	ストレージ	
1008	I/Oインタフェース	
1010	通信インタフェース	
1012	バス	

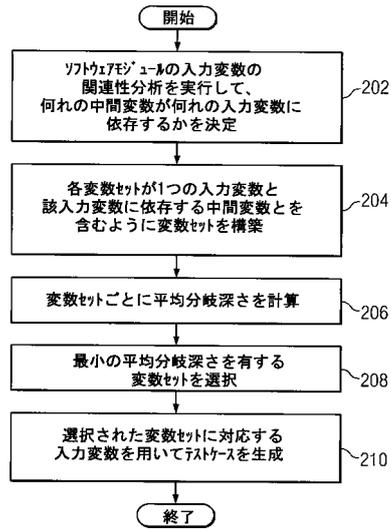
【 図 1 】

一例に係るソフトウェアモジュール上で記号的実行を行うステップを表す実行フロー例を示す図



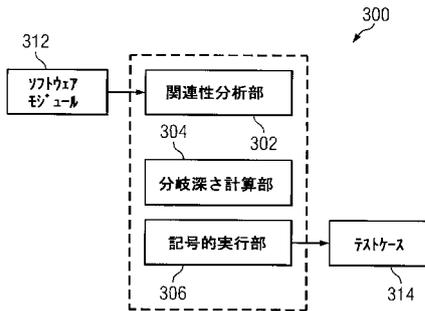
【 図 2 】

ソフトウェアモジュールの入力変数を記号的に実行する順序を決定する方法例を示す図



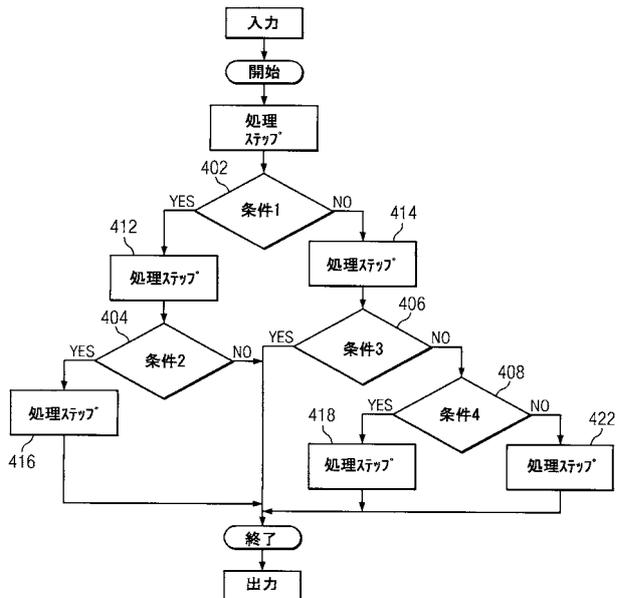
【 図 3 】

ソフトウェアモジュールの入力変数を記号的に実行する順序を決定するシステム例を示す図



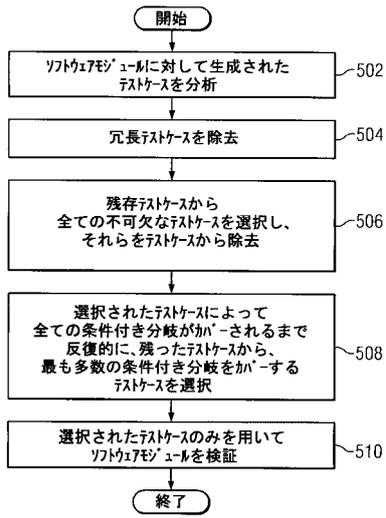
【 図 4 】

複数レベルのネスト化された条件付き分岐を有する一例に係るソフトウェアモジュールの実行経路を示す図



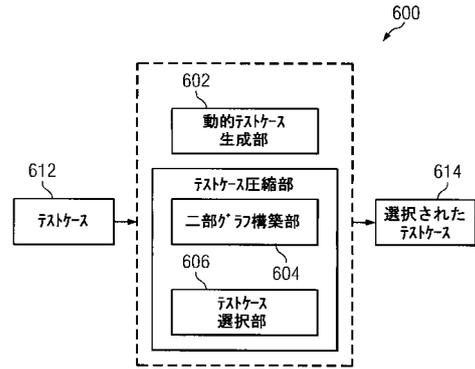
【 図 5 】

テストケースセットのサイズを縮小する方法例を示す図



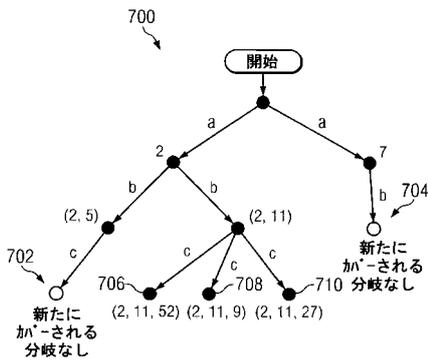
【 図 6 】

テストケースセットのサイズを縮小するシステム例を示す図



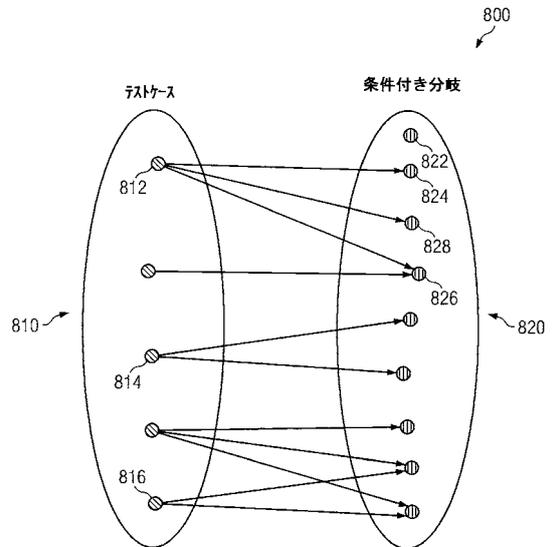
【 図 7 】

一例に係るソフトウェアモジュールの入力変数の例のテスト値の例を表すグラフ例を示す図



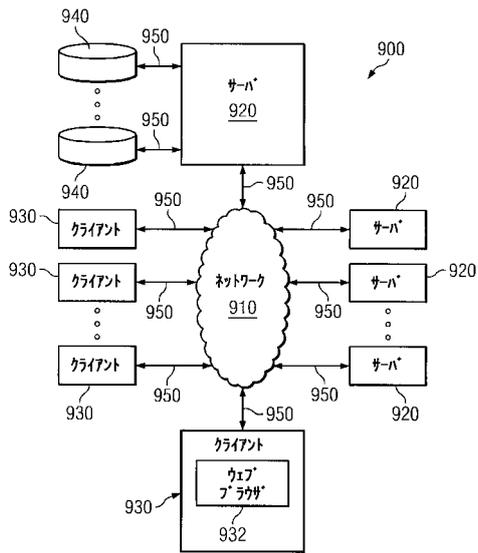
【 図 8 】

一例に係るソフトウェアモジュールの実行経路上でのテスト網羅率を示す二部グラフ例を示す図



【 図 9 】

一例に係るネットワーク環境を示す図



【 図 10 】

一例に係るコンピュータシステムを示す図

