

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5556655号
(P5556655)

(45) 発行日 平成26年7月23日(2014.7.23)

(24) 登録日 平成26年6月13日(2014.6.13)

(51) Int.Cl. F 1
G 0 6 F 11/36 (2006.01) G 0 6 F 9/06 6 2 0 M

請求項の数 6 (全 31 頁)

(21) 出願番号	特願2010-293472 (P2010-293472)	(73) 特許権者	000005223
(22) 出願日	平成22年12月28日(2010.12.28)		富士通株式会社
(65) 公開番号	特開2012-141741 (P2012-141741A)		神奈川県川崎市中原区上小田中4丁目1番1号
(43) 公開日	平成24年7月26日(2012.7.26)	(74) 代理人	100111822
審査請求日	平成25年10月7日(2013.10.7)		弁理士 渡部 章彦
		(74) 代理人	100119161
			弁理士 重久 啓子
		(72) 発明者	瀬川 和也
			神奈川県横浜市港北区新横浜二丁目4番地19 株式会社富士通ソフトウェアテクノロジーズ内
		審査官	大塚 俊範

最終頁に続く

(54) 【発明の名称】 解析支援プログラム、解析支援装置および解析支援方法

(57) 【特許請求の範囲】

【請求項1】

コンピュータに、

ソースコードに対して行われた静的解析において未確定要素を含むために欠陥であるか否か確定できないと判定された前記ソースコード中の箇所であって、共通の未確定要素を含む複数の箇所について、前記複数の箇所のそれぞれを示す情報と、前記共通の未確定要素を示す情報とを対応付けて表示装置に表示させる制御を行う

ことを実行させることを特徴とする解析支援プログラム。

【請求項2】

前記コンピュータに、

前記共通の未確定要素を制限する条件の入力を受け付け、

前記複数の箇所のそれぞれにおいて、前記共通の未確定要素が前記条件を満たす場合に欠陥であるか否かを再判定する

ことを実行させることを特徴とする請求項1に記載の解析支援プログラム。

【請求項3】

前記コンピュータに、さらに、

前記共通の未確定要素を含む複数の箇所のそれぞれについて、欠陥であると判定される未確定要素の範囲、または欠陥でないと判定される未確定要素の範囲を表示装置に表示させる制御を行う

ことを実行させることを特徴とする請求項1又は請求項2に記載の解析支援プログラム

【請求項 4】

前記コンピュータに、さらに、

前記未確定要素が前記条件を満たす場合に欠陥であると判定した前記ソースコード中の箇所を示す情報を、表示装置に表示させる制御を行う

ことを実行させることを特徴とする請求項 2 に記載の解析支援プログラム。

【請求項 5】

ソースコードに対して行われた静的解析において、未確定要素を含むために欠陥であるか否か確定できないと判定された前記ソースコード中の箇所の情報を記憶する欠陥情報記憶部と、

前記欠陥情報記憶部に記憶された共通の未確定要素を含む複数の箇所について、前記複数の箇所のそれぞれを示す情報と、前記共通の未確定要素を示す情報とを対応付けて表示装置に表示させる欠陥指摘画面出力部とを備える

ことを特徴とする解析支援装置。

【請求項 6】

コンピュータが、

ソースコードに対して行われた静的解析において未確定要素を含むために欠陥であるか否か確定できないと判定された前記ソースコード中の箇所であって、共通の未確定要素を含む複数の箇所について、前記複数の箇所のそれぞれを示す情報と、前記共通の未確定要素を示す情報とを対応付けて表示装置に表示させる制御を実行する

ことを特徴とする解析支援方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、プログラミング言語で記述されたソースプログラムの静的解析における解析支援プログラム、解析支援装置および解析支援方法に関するものである。

【背景技術】

【0002】

プログラミング言語で記述されたソースプログラムを解析し、ソースプログラムの欠陥を抽出して、それらの抽出された欠陥の指摘を、プログラム開発者等のユーザに対して提示する技術がある。実行形式のプログラムを実行しながら解析を行う手法が動的解析と呼ばれるのに対して、ソースプログラムの状態でプログラムを実行せずに解析を行う手法は、静的解析と呼ばれる。

【0003】

なお、第 1 のコンピュータの問題を解析する第 2 のコンピュータから可能原因類別を含む警告を受けた第 3 の遠隔コンピュータが、問題が第 1 のコンピュータ側で是正できる場合に、問題の原因の追加情報を取得して問題の解析を行う技術が知られている。

【先行技術文献】

【特許文献】

【0004】

【特許文献 1】特開平 6 - 5 2 0 0 3 号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

ソースプログラムの静的解析では、ソースプログラム上の欠陥の可能性のある箇所について、大きく分けて、次の 3 通りに分類する判定を行う。

【0006】

- ・欠陥である
- ・欠陥でない
- ・欠陥であるか否かを確定できない

10

20

30

40

50

これらの判定結果のうち、欠陥であるか否かが確定できないとの判定は、対象のソースプログラムの解析だけでは欠陥であるか否かの答えが得られない場合になされる。例えば、ソースプログラムの解析では関数の復帰値が確定できず、その復帰値の条件次第で欠陥となる可能性も欠陥とならない可能性もあるような場合には、欠陥であるか否かを確定できないと判定される。特に、関数の多くがライブラリになっており、ソースの実態が特定できない場合などには、欠陥であるか否かを確定できないとの判定結果が多くなる。

【0007】

ソースプログラムの静的解析結果の出力では、プログラム開発者などのユーザに対して、ソースプログラムから抽出された欠陥に関する指摘が提示される。

【0008】

このとき、ソースプログラム上の欠陥の可能性のある箇所のすべてについて、その判定結果に応じた欠陥に関する指摘を行うことも、一応は可能である。ただし、開発するプログラムの規模によっては、欠陥に関する指摘を行う箇所の数も膨大な数となるため、それらのすべてについて欠陥に関する指摘を行うことは、好ましくない。そのため、欠陥でないと判定された箇所については、無理に欠陥に関する指摘を出力する必要はない。

【0009】

また、ユーザに提示する指摘の数を削減するために、欠陥であるか否かを確定できないと判定された箇所についても、欠陥に関する指摘を出力しないようにできる。しかし、欠陥であるか否かを確定できないと判定された箇所については、欠陥である可能性が残っており、それを指摘しないと欠陥の見逃しが増えてしまうため、ソースプログラムの静的解析の成果が向上しない。

【0010】

これに対して、欠陥であるか否かを確定できないと判定された箇所について、欠陥に関する指摘を出力することもできる。しかし、この場合には、出力される欠陥に関する指摘が膨大な数になってしまう可能性がある。

【0011】

欠陥であるか否かを確定できないと判定された箇所については、欠陥でない可能性もある。そのため、欠陥であるか否かを確定できないと判定された箇所の1つ1つについて、ユーザは、それぞれの指摘を個別に確認しながら、欠陥であるか否かの調査を行う必要がある。膨大な数の指摘を1つ1つ確認する作業は、ユーザにとって非常な手間となる。

【0012】

一側面では、本発明は、ソースプログラムの静的解析において、欠陥であるか否かを確定できないと判定された箇所の情報をユーザに対して提示する際に、それらの指摘を確認するユーザの手間を軽減することが可能となる技術を提供することを目的とする。

【課題を解決するための手段】

【0013】

1態様では、プログラムは、コンピュータを、次のように機能させる。

【0014】

すなわち、プログラムは、プログラムがインストールされて実行されるコンピュータに、ソースコードに対して行われた静的解析において未確定要素を含むために欠陥であるか否か確定できないと判定された前記ソースコード中の箇所であって、共通の未確定要素を含む複数の箇所について、前記複数の箇所のそれぞれを示す情報と、前記共通の未確定要素を示す情報とを対応付けて表示装置に表示させる制御を行う。

【発明の効果】

【0015】

1態様では、ユーザの手間を軽減することが可能となる。

【図面の簡単な説明】

【0016】

【図1】本実施の形態による静的解析装置の構成例を示す図である。

【図2】本実施の形態による静的解析装置のハードウェア構成例を示す図である。

10

20

30

40

50

【図 3】本実施の形態の静的解析装置による静的解析処理フローチャートである。

【図 4】本実施の形態によるソースプログラムの例 (1) を示す図である。

【図 5】本実施の形態によるソース解析処理フローチャートである。

【図 6】本実施の形態による制御フローの例を示す図である。

【図 7】本実施の形態によるデータフローの例を示す図である。

【図 8】本実施の形態によるデータフローの例を示す図である。

【図 9】本実施の形態の欠陥判定抽出部による配列外参照欠陥箇所抽出処理フローチャートである。

【図 10】本実施の形態による欠陥情報の例 (1) を示す図である。

【図 11】本実施の形態による欠陥指摘画面の例 (1) を示す図である。

10

【図 12】本実施の形態による欠陥指摘画面の例 (2) を示す図である。

【図 13】本実施の形態による条件設定画面の例 (1) を示す図である。

【図 14】本実施の形態のヒント情報生成部および条件設定画面出力部による条件設定画面出力処理フローチャートである。

【図 15】本実施の形態による条件設定画面の例 (2) を示す図である。

【図 16】本実施の形態による条件情報の例を示す図である。

【図 17】本実施の形態による欠陥指摘画面の例 (3) を示す図である。

【図 18】本実施の形態の欠陥指摘画面出力部による欠陥指摘画面出力処理フローチャートである。

【図 19】本実施の形態の指摘変換部による指摘変換処理フローチャートである。

20

【図 20】本実施の形態によるソースプログラムの例 (2) を示す図である。

【図 21】本実施の形態による欠陥情報の例 (2) を示す図である。

【図 22】本実施の形態による欠陥指摘画面の例 (4) を示す図である。

【図 23】本実施の形態による条件設定画面の例 (3) を示す図である。

【図 24】本実施の形態による条件設定画面の例 (4) を示す図である。

【図 25】本実施の形態による欠陥指摘画面の例 (5) を示す図である。

【図 26】本実施の形態によるソースプログラムの例 (3) を示す図である。

【図 27】本実施の形態による欠陥情報の例 (3) を示す図である。

【図 28】本実施の形態による欠陥指摘画面の例 (6) を示す図である。

【図 29】本実施の形態による条件設定画面の例 (5) を示す図である

30

【図 30】本実施の形態による条件設定画面の例 (6) を示す図である。

【図 31】本実施の形態による欠陥指摘画面の例 (7) を示す図である。

【発明を実施するための形態】

【0017】

以下、本実施の形態について、図を用いて説明する。

【0018】

図 1 は、本実施の形態による静的解析装置の構成例を示す図である。

【0019】

静的解析装置 10 は、プログラミング言語で記述されたソースプログラムに対する静的解析を行い、ソースプログラムの欠陥を指摘する。ソースプログラムは、ソースコードや原始プログラムなどとも呼ばれる。静的解析は、ソースプログラムの状態で、プログラムを実行せずに、プログラムの欠陥を抽出する解析手法である。静的解析装置 10 には、ディスプレイ 20 等の表示装置や、キーボード / マウス 30 等の入力装置が接続されている。

40

【0020】

静的解析装置 10 は、ソースプログラム記憶部 11、欠陥情報生成部 12、欠陥情報記憶部 13、欠陥指摘画面出力部 14、ヒント情報生成部 15、条件設定画面出力部 16、条件入力受付部 17、条件情報記憶部 18 を備える。

【0021】

ソースプログラム記憶部 11 は、プログラミング言語で記述されたソースプログラムの

50

ファイルを記憶する，コンピュータがアクセス可能な記憶部である。

【 0 0 2 2 】

欠陥情報生成部 1 2 は，ソースプログラムから欠陥箇所を抽出し，抽出された欠陥箇所の情報を収集して欠陥情報を生成する。このとき，ソースプログラムから抽出される欠陥箇所には，欠陥であるか否かを確定できないと判定された箇所も含まれる。欠陥情報生成部 1 2 は，ソース解析部 1 2 1，ソース情報記憶部 1 2 2，欠陥判定抽出部 1 2 3 を備える。

【 0 0 2 3 】

ソース解析部 1 2 1 は，ソースプログラム記憶部 1 1 に記憶されたソースプログラムに対して字句解析，構文解析等の解析を行い，ソースプログラムの解析結果であるソース情報

10

【 0 0 2 4 】

ソース情報記憶部 1 2 2 は，ソース解析部 1 2 1 により生成されたソース情報を記憶する，コンピュータがアクセス可能な記憶部である。本実施の形態では，ソース情報として，制御フロー，データフロー等が生成され，ソース情報記憶部 1 2 2 に記憶される。

【 0 0 2 5 】

欠陥判定抽出部 1 2 3 は，ソース情報記憶部 1 2 2 に記憶されたソース情報に基づいて，ソースプログラム上の欠陥の可能性のある箇所を欠陥に関する指摘を行う箇所の候補とし，その候補に対して欠陥の種類に応じた詳細な探索を行い，欠陥であるか否かの判定を行う。

20

【 0 0 2 6 】

欠陥判定抽出部 1 2 3 による判定結果は，大きく分けて，欠陥である，欠陥でない，欠陥であるか否かを確定できないの 3 通りに分類される。欠陥判定抽出部 1 2 3 は，欠陥の可能性のある箇所について行った探索で，未確定の要素がなく，欠陥であるか否かの答えが得られた場合に，それぞれ欠陥である，欠陥でないと判定する。欠陥判定抽出部 1 2 3 は，欠陥の可能性のある箇所について行った探索で，復帰値が未確定である関数などの未確定要素を含むために欠陥であるか否かの答えが得られない場合に，欠陥であるか否かが確定できないと判定する。なお，以下では，欠陥であることを \times ，欠陥でないことを \circ ，欠陥であるか否かを確定できないことを Δ で示す。

【 0 0 2 7 】

本実施の形態の静的解析装置 1 0 では，欠陥判定抽出部 1 2 3 は， \times と判定した箇所と Δ と判定した箇所とを指摘する欠陥箇所として抽出し，抽出された欠陥箇所の情報を収集して欠陥情報に記録する。 Δ と判定した箇所については，欠陥判定抽出部 1 2 3 は，欠陥を確定できない理由についても欠陥情報に記録する。

30

【 0 0 2 8 】

欠陥情報記憶部 1 3 は，ソースプログラムから抽出された欠陥箇所に関する情報が記録された欠陥情報を記憶する，コンピュータがアクセス可能な記憶部である。欠陥情報記憶部 1 3 には，欠陥箇所の情報として， Δ と判定された箇所，すなわち，ソースプログラムに対して行われた静的解析において，未確定要素を含むために欠陥であるか否か確定できないと判定されたソースプログラム中の箇所の情報も記憶されている。

40

【 0 0 2 9 】

欠陥指摘画面出力部 1 4 は，欠陥情報記憶部 1 3 に記憶された欠陥情報に基づいて，ユーザに対してソースプログラムの欠陥箇所を指摘する欠陥指摘画面を，ディスプレイ 2 0 等の表示装置に表示させる制御を行う。このとき，欠陥指摘画面出力部 1 4 は，共通の未確定要素を含む複数の Δ と判定された箇所について，それら複数の箇所のそれぞれを示す情報と，共通の未確定要素を示す情報とを対応付けて，欠陥指摘画面をディスプレイ 2 0 等の表示装置に表示させる。

【 0 0 3 0 】

以下，本実施の形態では， Δ と判定された箇所の指摘を，欠陥未確定の指摘と呼ぶ。また，欠陥を確定できない理由を，未確定理由と呼ぶ。未確定理由には，欠陥を確定できな

50

い理由の要因となる未確定要素が含まれる。例えば、「関数の復帰値が未確定である」という未確定理由には、「関数の復帰値」という未確定要素が含まれる。

【 0 0 3 1 】

ヒント情報生成部 1 5 は、ユーザから条件設定する旨の指示を受けたときに、ユーザによる未確定理由を解消する条件の入力を支援するヒント情報を生成する。ヒント情報は、欠陥未確定の指摘が欠陥である (×) ことを示す指摘に変換される条件の情報や、欠陥未確定の指摘が欠陥でない () ことを示す指摘に変換される条件の情報などである。欠陥未確定の指摘が欠陥である (×) ことを示す指摘に変換される条件は、欠陥であると判定される未確定要素の範囲である。また、欠陥未確定の指摘が欠陥でない () ことを示す指摘に変換される条件は、欠陥でないと判定される未確定要素の範囲である。

10

【 0 0 3 2 】

例えば、ヒント情報生成部 1 5 は、関数の復帰値が未確定要素である欠陥未確定の指摘について、欠陥情報記憶部 1 3 に記憶された欠陥情報を参照して、その欠陥未確定の指摘が欠陥でない () ことを示す指摘に変換される関数の復帰値の範囲を、ヒント情報として求める。また、ヒント情報生成部 1 5 は、関数の復帰値が未確定要素である欠陥未確定の指摘について、欠陥情報記憶部 1 3 に記憶された欠陥情報を参照して、その欠陥未確定の指摘が欠陥である (×) ことを示す指摘に変換される関数の復帰値の範囲を、ヒント情報として求める。

【 0 0 3 3 】

条件設定画面出力部 1 6 は、ユーザから条件設定する旨の指示を受けたときに、ユーザが未確定理由を解消する条件を入力する条件設定画面を、ディスプレイ 2 0 等の表示装置に表示させる制御を行う。このとき、条件設定画面出力部 1 6 は、条件設定画面に、生成されたヒント情報を掲載する。条件設定画面にヒント情報が掲載されることにより、ユーザは、設定する条件を判断し易くなる。本実施の形態による条件設定画面では、未確定理由を解消する条件が、未確定理由ごとにまとめて入力できるようになっている。

20

【 0 0 3 4 】

条件入力受付部 1 7 は、欠陥未確定の指摘について、未確定理由ごとにまとめて、その未確定理由を解消する条件の入力を受け付ける。欠陥未確定の指摘の未確定理由を解消する条件は、例えば、欠陥未確定とされた箇所に含まれる未確定要素を制限する条件となる。条件入力受付部 1 7 は、共通の未確定要素を含む欠陥未確定とされた箇所ごとに、その共通の未確定要素を制限する条件の入力を受け付ける。

30

【 0 0 3 5 】

ユーザは、例えば、キーボード/マウス 3 0 等の入力装置を用いて、条件設定画面上で未確定理由を解消する条件を入力する。条件入力受付部 1 7 は、ユーザによる未確定理由を解消する条件の入力を受け付け、受け付けた条件を未確定理由に対応付けて条件情報記憶部 1 8 に記憶された条件情報に記録する。

【 0 0 3 6 】

条件情報記憶部 1 8 は、未確定理由と、ユーザにより入力された、その未確定理由を解消する条件との対応が記録された条件情報を記憶する、コンピュータがアクセス可能な記憶部である。

40

【 0 0 3 7 】

上述の欠陥指摘画面出力部 1 4 は、指摘変換部 1 4 1 を備える。指摘変換部 1 4 1 は、条件情報記憶部 1 8 に記憶された条件情報を参照し、共通の未確定要素を含む複数の箇所のそれぞれにおいて、その共通の未確定要素がユーザにより入力された条件を満たす場合に欠陥であるか否かを再判定する。指摘変換部 1 4 1 は、再判定の結果に基づいて、欠陥未確定の指摘を、欠陥であることを示す指摘または欠陥でないことを示す指摘に変換する。

【 0 0 3 8 】

欠陥指摘画面出力部 1 4 は、欠陥未確定の指摘を変換した結果を含む欠陥指摘画面を、ディスプレイ 2 0 などの表示装置に表示させる制御を行う。例えば、欠陥指摘画面出力部

50

14は、欠陥未確定の指摘から変換された欠陥であることを示す指摘、すなわち再判定で未確定要素がユーザにより入力された条件を満たす場合に欠陥であると判定したソースプログラム中の箇所を示す情報を含む欠陥指摘画面を表示させる。

【0039】

欠陥判定抽出部123によるソースプログラムを解析した情報のみでの欠陥の判定では、対象のソースプログラムだけでは値が確定できない関数の復帰値などの未確定要素を含む箇所について、欠陥であるか否かを確定することができない。ただし、プログラム開発者であるユーザは、そのような未確定要素となる関数の復帰値などをあらかじめ想定してソースプログラムの設計を行っており、ユーザが未確定要素となる関数の復帰値の範囲などを特定できるケースもある。このような場合に、特定できる関数の復帰値の範囲などを未確定要素を制限する条件としてユーザに入力させ、入力された条件で欠陥であるか否かの判定を行うことで、確定できなかった欠陥を、欠陥であるかまたは欠陥でないかに確定できる。

【0040】

このように、本実施の形態による静的解析装置10は、ソースプログラムの静的解析結果として欠陥未確定の指摘を含めてユーザに提示する際に、それらの解析結果を確認するユーザを支援する解析支援機能を有する。

【0041】

例えば、静的解析装置10の解析支援機能によって、複数の欠陥未確定の指摘と、それらの欠陥未確定の指摘に共通の未確定要素とを対応付けて、ユーザに提示することが可能となる。また、例えば、静的解析装置10の解析支援機能によって、ユーザにより入力された共通の未確定要素を制限する条件で、共通の未確定要素を持つ複数の欠陥未確定の指摘を、それぞれ欠陥であることを示す指摘と欠陥でないことを示す指摘に変換して、ユーザに提示することが可能となる。

【0042】

本実施の形態の静的解析装置10が有する解析支援機能によって、ソースプログラムの静的解析において、欠陥であるか否かを確定できないと判定された箇所の指摘をユーザに対して提示する際に、それらの指摘を確認するユーザの手間を軽減することが可能となる。

【0043】

図2は、本実施の形態による静的解析装置のハードウェア構成例を示す図である。

【0044】

図1に示す本実施の形態の静的解析装置10は、CPU(Central Processing Unit)101、主記憶となるメモリ102、入出力インタフェース103、記憶装置104、入力装置105、出力装置106を備えるコンピュータ100によって実現される。記憶装置104は、例えばHDD(Hard Disk Drive)などである。出力装置は、例えば図1に示すディスプレイ20等の表示装置などである。入力装置105は、例えば図1に示すキーボード/マウス30などである。

【0045】

図1に示す静的解析装置10および静的解析装置10が備える各機能部は、コンピュータ100が備えるCPU101、メモリ102等のハードウェアと、ソフトウェアプログラムとによって、実現することが可能である。コンピュータ100が実行可能なプログラムは、記憶装置104に記憶され、その実行時にメモリ102に読み出され、CPU101により実行される。

【0046】

なお、コンピュータ100は、可搬型記録媒体から直接プログラムを読み取り、そのプログラムに従った処理を実行することもできる。また、コンピュータ100は、サーバコンピュータからプログラムが転送されるごとに、逐次、受け取ったプログラムに従った処理を実行することもできる。さらに、このプログラムは、コンピュータ100で読み取り可能な記録媒体に記録しておくことができる。

10

20

30

40

50

【 0 0 4 7 】

図 3 は、本実施の形態の静的解析装置による静的解析処理フローチャートである。

【 0 0 4 8 】

静的解析装置 1 0 において、ユーザは、キーボード/マウス 3 0 を用いて、ソースプログラム記憶部 1 1 に記憶されているソースプログラムを指定した、静的解析の実行を指示する。

【 0 0 4 9 】

欠陥情報生成部 1 2 は、ソースプログラムから、と判定された箇所を含む欠陥箇所を抽出する(ステップ S 1 0)。本実施の形態では、静的解析装置 1 0 は、解析対象のソースプログラムの欠陥箇所として、×と判定された箇所とと判定された箇所とを、ユーザ 10 に対して提示する。欠陥情報生成部 1 2 は、ソースプログラムを解析して×と判定された箇所とと判定された箇所とを欠陥箇所として抽出し、抽出された欠陥箇所の情報を収集して欠陥情報を生成する。欠陥情報は、欠陥情報記憶部 1 3 に記憶される。

【 0 0 5 0 】

欠陥指摘画面出力部 1 4 は、抽出した欠陥箇所の指摘を出力する。本実施の形態では、欠陥指摘画面出力部 1 4 は、欠陥情報記憶部 1 3 に記憶された欠陥情報に基づいて、ソースプログラムから抽出した欠陥箇所を指摘する欠陥指摘画面を、ディスプレイ 2 0 等の表示装置に出力する(ステップ S 1 1)。このとき、欠陥指摘画面出力部 1 4 は、欠陥未確定の指摘について未確定理由ごとにまとめた情報を含む欠陥指摘画面を出力する。

【 0 0 5 1 】

ユーザは、欠陥指摘画面において、未確定理由を指定して、その未確定理由を解消する条件の設定を指示する。 20

【 0 0 5 2 】

ヒント情報生成部 1 5 は、欠陥情報記憶部 1 3 に記憶された欠陥情報に基づいて、条件の設定が指示された未確定理由を解消する条件の入力を支援するヒント情報を生成する(ステップ S 1 2)。本実施の形態では、ヒント情報として、条件の設定が指示された未確定理由を持つ欠陥未確定の指摘ごとに、それぞれ欠陥であることを示す指摘に変換される条件の情報と、欠陥でないことを示す指摘に変換される条件の情報を生成する。

【 0 0 5 3 】

条件設定画面出力部 1 6 は、ユーザに指定された未確定理由を解消する条件を設定する条件設定画面を、ディスプレイ 2 0 に出力する(ステップ S 1 3)。本実施の形態では、条件設定画面に、欠陥未確定の指摘ごとに生成されたヒント情報が提示される。 30

【 0 0 5 4 】

ユーザは、キーボード/マウス 3 0 を用いて、条件設定画面において、未確定理由を解消する条件を入力する。

【 0 0 5 5 】

条件入力受付部 1 7 は、ユーザによる未確定理由を解消する条件の入力を受け付ける(ステップ S 1 4)。すなわち、条件入力受付部 1 7 は、欠陥未確定の指摘について、未確定理由ごとにまとめて、その理由を解消する条件の入力を受け付ける。入力された条件は、未確定理由に対応付けられて、条件情報記憶部 1 8 に記憶された条件情報に記録される 40

【 0 0 5 6 】

欠陥指摘画面出力部 1 4 の指摘変換部 1 4 1 は、欠陥未確定の指摘を、ユーザにより入力された未確定理由を解消する条件で、欠陥であることを示す指摘または欠陥でないことを示す指摘に変換する(ステップ S 1 5)。

【 0 0 5 7 】

欠陥指摘画面出力部 1 4 は、欠陥未確定の指摘を変換した結果を含む、抽出した欠陥箇所の指摘を出力する。ここでは、欠陥指摘画面出力部 1 4 は、ソースプログラムから抽出した欠陥箇所を指摘する欠陥指摘画面をディスプレイ 2 0 に出力するとき(ステップ S 1 6)、変換された欠陥未確定の指摘については変換した結果に基づいた欠陥指摘画面へ 50

の提示を行う。

【0058】

静的解析装置10では、ユーザが条件の設定を指示するごとに、ステップS12～ステップS16の処理が繰り返される。ユーザが未確定理由を解消する条件を設定するごとに、欠陥未確定の指摘が、欠陥であることを示す指摘または欠陥でないことを示す指摘に変換され、欠陥指摘画面でユーザに提示される欠陥未確定の指摘の数が減る。欠陥でないことを示す指摘を欠陥指摘画面に表示しないことにすれば、その分だけ欠陥指摘画面でユーザに提示される欠陥箇所の指摘の数が減ることになる。

【0059】

このように、本実施の形態の静的解析装置10では、×と判定された箇所の指摘だけではなく、欠陥未確定の指摘もユーザに対して提示することで、欠陥未確定の指摘に含まれる欠陥が指摘され、ソースプログラムの静的解析の成果が上がる。このとき、ユーザによる未確定理由ごとの条件の設定によって、欠陥未確定の指摘を未確定理由ごとにまとめて減らすことができるので、欠陥未確定の指摘を確認するユーザの負担も少ない。

【0060】

以下では、本実施の形態の静的解析装置10による処理について、より具体的な例を挙げて説明する。

【0061】

図4は、本実施の形態によるソースプログラムの例(1)を示す図である。

【0062】

図4に示すソースプログラム110aは、C言語で記述されたソースプログラムの一例である。ここでは、図4に示すソースプログラム110aが、静的解析の対象プログラムとして、ソースプログラム記憶部11に記憶されている。以下、図4に示すソースプログラム110aの静的解析を行う例を説明する。

【0063】

静的解析装置10において、欠陥情報生成部12は、図4に示すソースプログラム110a上の欠陥を抽出する。プログラムの欠陥は、一般にバグや誤りなどとも呼ばれる。

【0064】

欠陥情報生成部12において、ソース解析部121は、図4に示すソースプログラム110aを解析し、ソース情報を生成する。生成されたソース情報は、ソース情報記憶部122に記憶される。

【0065】

図5は、本実施の形態によるソース解析処理フローチャートである。

【0066】

ソース解析部121は、静的解析の対象として指定されたソースプログラムの字句解析を実施する(ステップS100)。字句解析は、ソースプログラムの記述を単語に分解する解析である。

【0067】

ソース解析部121は、ソースプログラムの字句解析結果に基づいて、ソースプログラムの構文解析を実施する(ステップS101)。構文解析は、字句解析によって単語に分解されたものの意味を解析する。

【0068】

ソース解析部121は、制御フローを生成する(ステップS102)。制御フローは、ソースプログラムの各制御の構造を示す情報である。

【0069】

ソース解析部121は、データフローを生成する(ステップS103)。データフローは、ソースプログラムにおける変数の更新や参照等を示す情報である。

【0070】

生成された制御フローやデータフローは、ソース情報として、ソース情報記憶部122に記憶される。

10

20

30

40

50

【 0 0 7 1 】

図 6 は、本実施の形態による制御フローの例を示す図である。

【 0 0 7 2 】

図 6 に示す制御フローは、図 4 に示すソースプログラム 1 1 0 a を解析して得られる制御フローの例を示している。図 6 に示す制御フローでは、ソースプログラム 1 1 0 a の制御の流れが、〔 1 〕～〔 1 5 〕の制御で示されている。以下の説明で、〔 〕付きの数字は、図 6 に示す制御フローにおける各制御を示す。

【 0 0 7 3 】

図 7、図 8 は、本実施の形態によるデータフローの例を示す図である。

【 0 0 7 4 】

図 7、図 8 には、図 6 に示す制御フローを簡略化したものが示されている。図 7、図 8 において、矢印が、ソースプログラム 1 1 0 a のデータフローを示している。図 7 に示す実線の矢印が、変数更新の流れを示しており、図 8 に示す破線の矢印が、定義と参照の関係を示している。

【 0 0 7 5 】

図 7 において、実線矢印で示された変数更新の流れは、次の通りである。

〔 a 〕：〔 1 0 〕の変数 (x) は、〔 8 〕の代入結果である。

〔 b 〕：〔 1 2 〕の変数 (x) は、〔 8 〕の代入結果である。

〔 c 〕：〔 1 3 〕の変数 (i n d e x 2) は、〔 1 2 〕の f o r ループの結果である。

〔 d 〕：〔 1 4 〕の変数 (y) は、〔 1 0 〕の代入結果である。

〔 e 〕：〔 1 5 〕の変数 (i n d e x 3) は、〔 1 4 〕の f o r ループの結果である。

【 0 0 7 6 】

図 8 において、破線矢印で示された定義と参照の関係は、次の通りである。

〔 f 〕：〔 8 〕の関数 (g e t n u m) は、未定義である。

〔 g 〕：〔 1 1 〕の変数 (a r r a y 1) は、〔 1 〕で定義されている。

〔 h 〕：〔 1 3 〕の変数 (a r r a y 2) は、〔 2 〕で定義されている。

〔 i 〕：〔 1 5 〕の変数 (a r r a y 3) は、〔 3 〕で定義されている。

【 0 0 7 7 】

なお、図 7、図 8 には、以下で説明する欠陥箇所の抽出に用いるデータフローの一部のみが示されている。

【 0 0 7 8 】

欠陥情報生成部 1 2 において、欠陥判定抽出部 1 2 3 は、ソース情報記憶部 1 2 2 に記憶されたソース情報に基づいて、ソースプログラム上の欠陥の可能性のある箇所の詳細な探索を行い、欠陥であるか否かを判定する。ここでは、欠陥判定抽出部 1 2 3 は、x と判定された箇所と と判定された箇所とを、指摘を行う欠陥箇所として抽出する。

【 0 0 7 9 】

欠陥判定抽出部 1 2 3 は、様々な種類の欠陥を抽出する。例えば、ポインタ変数に N U L L を設定しただけでポインタの指す領域をアクセスすると、0 番地をアクセスしてしまうため、実行環境によってはエラーが発生する可能性がある。このような欠陥は、0 番地参照と呼ばれる。また、変数の参照前にその変数の値が設定されていない場合に、初期化漏れと呼ばれる欠陥となる。また、取得した資源が解放されない場合に、メモリリークと呼ばれる欠陥となる。また、配列の添え字が、宣言された配列の範囲を超えている場合に、配列外参照と呼ばれる欠陥となる。0 番地参照、初期化漏れ、メモリリーク、配列外参照以外にも、様々な種類の欠陥がある。

【 0 0 8 0 】

以下では、ソースプログラムから x と判定された箇所と と判定された箇所とを欠陥箇所として抽出する例として、図 6 に示す制御フロー、図 7、図 8 に示すデータフローを用いて、配列外参照に関する欠陥箇所を抽出する例を説明する。

【 0 0 8 1 】

配列外参照に関する欠陥箇所の抽出では、配列の参照箇所を起点に配列範囲と添字範囲

10

20

30

40

50

とを求め、求められた範囲の比較により、欠陥であるか否かの判定が行われる。ソースプログラム 110a における配列の参照箇所としては、図 6 の制御フローから、〔11〕の「array 1」、〔13〕の「array 2」、〔15〕の「array 3」が挙げられる。

【0082】

・〔11〕の「array 1」について

〔11〕から図 8 に示す〔g〕の関係を追った先の〔1〕の定義で、配列範囲：0～9 が求まる。また、〔11〕において添字が定数（10）であるから、添字範囲：10 が求まる。

【0083】

配列範囲：0～9 よりも添字範囲：10の方が大きいので、〔11〕の「array 1」については、配列外参照の欠陥である（×）と判定される。

【0084】

・〔13〕の「array 2」について

〔13〕から図 8 に示す〔h〕の関係を追った先の〔2〕の定義で、配列範囲：0～14 が求まる。

【0085】

また、〔13〕の添字：index 2 は、図 7 に示す〔c〕の関係から、〔12〕の for ループの条件に左右されることがわかる。〔12〕の for ループの条件から、添字範囲は、0～変数（x）となる。図 7 に示す〔b〕から、変数（x）が〔8〕の代入結果すなわち getnum 復帰値であることがわかるので、添字範囲は、0～getnum 復帰値となる。〔8〕から図 8 に示す〔f〕の関係を追った先で、関数 getnum が未定義であることがわかる。関数 getnum が未定義であるため、getnum 復帰値も確定せず、それにより、添字範囲：0～getnum 復帰値は求まらない。

【0086】

配列範囲は求まったが添字範囲が求まらないため、〔13〕の「array 2」については、欠陥であるか否かが確定できない（ ）と判定される。このときの未確定理由は、getnum 復帰値が確定できないことである。

【0087】

・〔15〕の「array 3」について

〔15〕から図 8 に示す〔i〕の関係を追った先の〔3〕の定義で、配列範囲：0～19 が求まる。

【0088】

また、〔15〕の添字：index 3 は、図 7 に示す〔e〕の関係から、〔14〕の for ループの条件に左右されることがわかる。〔14〕の for ループの条件から、添字範囲は、0～変数（y）となる。図 7 に示す〔d〕から、変数（y）が〔10〕の代入結果すなわち変数（x）+10 であることがわかるので、添字範囲は、0～変数（x）+10 となる。また、図 7 に示す〔a〕から、変数（x）は、〔8〕の代入結果すなわち getnum 復帰値であることがわかるので、添字範囲は、0～getnum 復帰値+10 となる。〔8〕から図 8 に示す〔f〕の関係を追った先で、関数 getnum が未定義であることがわかる。関数 getnum が未定義であるため、getnum 復帰値も確定せず、それにより、添字範囲：0～getnum 復帰値+10 は求まらない。

【0089】

配列範囲は求まったが添字範囲が求まらないため、〔15〕の「array 3」については、欠陥であるか否かが確定できない（ ）と判定される。このときの未確定理由は、getnum 復帰値が確定できないことである。

【0090】

欠陥判定抽出部 123 は、指摘を行う欠陥箇所として抽出された、×と判定された箇所と と判定された箇所とについて情報を収集し、欠陥情報に記録する。

【0091】

10

20

30

40

50

図 9 は、本実施の形態の欠陥判定抽出部による配列外参照欠陥箇所抽出処理フローチャートである。

【 0 0 9 2 】

欠陥判定抽出部 1 2 3 は、配列外参照に関する欠陥箇所の抽出において、ソース情報記憶部 1 2 2 に記憶されたソース情報に基づいて、ソースプログラム上の配列の参照箇所を抽出する（ステップ S 1 1 0）。

【 0 0 9 3 】

欠陥判定抽出部 1 2 3 は、抽出された配列の参照箇所を 1 つ選択する（ステップ S 1 1 1）。欠陥判定抽出部 1 2 3 は、抽出された配列の参照箇所について、配列範囲を求める（ステップ S 1 1 2）。また、欠陥判定抽出部 1 2 3 は、抽出された配列の参照箇所について、添字範囲を求める（ステップ S 1 1 3）。欠陥判定抽出部 1 2 3 は、求めた配列範囲および添字範囲が確定しているか否かを判定する（ステップ S 1 1 4）。

10

【 0 0 9 4 】

求めた範囲が確定していなければ（ステップ 1 1 4 の N O ）、欠陥判定抽出部 1 2 3 は、選択された配列の参照箇所について、欠陥であるか否かが確定できない（ ）と判定する（ステップ S 1 1 5）。欠陥判定抽出部 1 2 3 は、選択された配列の参照箇所を指摘する欠陥箇所として情報を収集し、欠陥情報に記録する（ステップ S 1 1 6）。ここで欠陥情報に記録される情報には、未確定理由が含まれる。

【 0 0 9 5 】

求めた範囲が確定していれば（ステップ S 1 1 4 の Y E S ）、欠陥判定抽出部 1 2 3 は、添字範囲が配列範囲内であるか否かを判定する（ステップ S 1 1 7）。

20

【 0 0 9 6 】

添字範囲が配列範囲内でなければ（ステップ S 1 1 7 の N O ）、欠陥判定抽出部 1 2 3 は、選択された配列の参照箇所について、欠陥である（ × ）と判定する（ステップ S 1 1 8）。欠陥判定抽出部 1 2 3 は、選択された配列の参照箇所を指摘する欠陥箇所として情報を収集し、欠陥情報に記録する（ステップ S 1 1 6）。

【 0 0 9 7 】

添字範囲が配列範囲内であれば（ステップ S 1 1 7 の Y E S ）、欠陥判定抽出部 1 2 3 は、選択された配列の参照箇所について、欠陥でない（ ）と判定する（ステップ S 1 1 9）。この場合には、欠陥判定抽出部 1 2 3 は、選択された配列の参照箇所の情報を欠陥情報に記録しない。

30

【 0 0 9 8 】

欠陥判定抽出部 1 2 3 は、抽出されたすべての配列の参照箇所について処理が終了したかを判定する（ステップ S 1 2 0）。すべての配列の参照箇所について処理が終了していなければ（ステップ S 1 2 0 の N O ）、欠陥判定抽出部 1 2 3 は、ステップ S 1 1 1 の処理に戻り、次の配列の参照箇所についての処理に移る。すべての配列の参照箇所について処理が終了していれば（ステップ S 1 2 0 の Y E S ）、欠陥判定抽出部 1 2 3 は、配列外参照に関する欠陥箇所の抽出処理を終了する。

【 0 0 9 9 】

ここでは、配列外参照に関する欠陥箇所の抽出の例のみを説明したが、欠陥判定抽出部 1 2 3 は、0 番地参照、初期化漏れ、メモリリークなど、配列外参照以外の様々な欠陥についても、該当する欠陥箇所の抽出を行う。

40

【 0 1 0 0 】

欠陥情報生成部 1 2 により生成された、ソースプログラムから抽出された欠陥箇所についての情報が記録された欠陥情報は、欠陥情報記憶部 1 3 に記憶される。

【 0 1 0 1 】

図 1 0 は、本実施の形態による欠陥情報の例（ 1 ）を示す図である。

【 0 1 0 2 】

図 1 0 に示す欠陥情報 1 3 0 a は、図 4 に示すソースプログラム 1 1 0 a から抽出された配列外参照に関する欠陥箇所が記録された欠陥情報の例である。図 1 0 に示す欠陥情報

50

130 aは、ファイル、行番号、指摘番号、メッセージ、判定理由、未確定理由等の情報を持つ。欠陥情報生成部12において、欠陥判定抽出部123は、xと判定された箇所と判定された箇所とについて、これらの各情報を収集し、欠陥情報に記録する。

【0103】

欠陥情報において、ファイルは、静的解析の対象となったソースプログラムのファイル名を示す。図10に示す欠陥情報130 aにおいて、ファイルa.cは、図4に示すソースプログラム110 aのファイルを示している。静的解析装置10は、複数のソースプログラムをまとめて解析することが可能である。複数のソースプログラムをまとめて解析する場合に、生成される欠陥情報は、複数のソースプログラムから抽出された欠陥の情報となる。

10

【0104】

欠陥情報において、行番号は、ソースプログラムにおける抽出された欠陥箇所の行番号を示す。

【0105】

欠陥情報において、指摘番号は、欠陥の種類と欠陥であるか否かの判定結果とに応じた番号である。指摘番号は、欠陥の種類と欠陥であるか否かの判定結果との組合せに応じてあらかじめ決められている。

【0106】

例えば、図10に示す欠陥情報130 aの例では、欠陥の種類が配列外参照であり、欠陥であるか否かの判定結果が欠陥である(x)とされた欠陥箇所については、指摘番号として“5011”を記録することが決められている。また、例えば、図10に示す欠陥情報130 aの例では、欠陥の種類が配列外参照であり、欠陥であるか否かの判定結果が欠陥であるか否かが確定できない()とされた欠陥箇所については、指摘番号として“5012”を記録することが決められている。

20

【0107】

欠陥判定抽出部123は、欠陥情報に抽出された欠陥箇所の情報を記録する際に、該欠陥箇所の欠陥の種類と、該欠陥箇所の欠陥であるか否かの判定結果から、それらの組合せに応じてあらかじめ決められた指摘番号を欠陥情報に記録する。

【0108】

欠陥情報において、メッセージは、ディスプレイ20に出力される欠陥指摘画面で提示されるメッセージである。

30

【0109】

欠陥情報において、判定理由は、欠陥であるか否かの判定において、xと判定された理由または と判定された理由を示す情報である。例えば、図10に示す欠陥情報130 aにおいて、ファイルa.cの行番号13のレコードでは、添字範囲：10が配列範囲：0～9外であることを理由に、配列外参照についてxと判定されたことが示されている。

【0110】

欠陥情報において、未確定理由は、 と判定された箇所について、欠陥を確定できない理由を示す。例えば、図10に示す欠陥情報130 aにおいて、ファイルa.cの行番号15のレコードでは、getnum復帰値が未確定であることを理由に、欠陥を確定できないことが示されている。

40

【0111】

判定理由や未確定理由の情報は、欠陥判定抽出部123による欠陥箇所抽出の過程で生成される。ここで、図4に示すソースプログラム110 aの行番号15における、配列array2の参照箇所を例として、判定理由と未確定理由とを生成する例を説明する。

【0112】

図9に示す配列外参照欠陥箇所抽出処理のフローチャートにおいて、欠陥判定抽出部123は、ステップS114、ステップS115の処理で、配列範囲または添字範囲のいずれかが確定していない場合に、欠陥であるか否かが確定できないと判定している。配列array2の参照箇所の場合には、添字範囲が0～getnum復帰値であり、その範囲

50

が確定していないので、欠陥であるか否かが確定できないと判定される。

【0113】

このとき、欠陥判定抽出部123は、範囲が確定していない添字範囲：0～getnum復帰値を抽出し、判定理由を生成する。なお、本実施の形態では、欠陥判定抽出部123は、後の条件設定時にユーザが容易に比較判断できるように、判定理由に範囲が確定している配列範囲：0～14も含める。また、欠陥判定抽出部123は、範囲が確定していない添字範囲：0～getnum復帰値から、その範囲が確定できない要因である未確定要素としてgetnum復帰値を抽出し、getnum復帰値未確定という未確定理由を生成する。

【0114】

図9に示す配列外参照欠陥箇所抽出処理のフローチャートにおいて、欠陥判定抽出部123は、ステップS116の処理で、生成された判定理由と未確定理由とを、欠陥情報に記録する。図10の欠陥情報130aに示すように、行番号15のレコードに、判定理由“配列範囲：0～14，添字範囲：0～getnum復帰値”，未確定理由“getnum復帰値未確定”が記録される。

【0115】

なお、図10に示す欠陥情報130aにおいて、ファイルa.cの行番号13のレコードは、上述の〔11〕の「array1」について抽出された欠陥箇所の情報である。また、ファイルa.cの行番号15のレコードは、上述の〔13〕の「array2」について抽出された欠陥箇所の情報である。また、ファイルa.cの行番号18のレコードは、上述の〔15〕の「array3」について抽出された欠陥箇所の情報である。

【0116】

欠陥指摘画面出力部14は、欠陥情報記憶部13に記憶された欠陥情報に基づいて欠陥指摘画面を生成し、生成された欠陥指摘画面を、ディスプレイ20に出力する。

【0117】

図11は、本実施の形態による欠陥指摘画面の例(1)を示す図である。

【0118】

図11に示す欠陥指摘画面145aは、欠陥指摘画面出力部14が、欠陥情報記憶部13に記憶された欠陥情報130aに基づいて生成し、ディスプレイ20に出力する欠陥指摘画面の一例である。図11に示す欠陥指摘画面145aは、指摘番号一覧、未確定理由一覧、指摘一覧の3つの一覧を有している。

【0119】

欠陥指摘画面において、指摘番号一覧は、欠陥情報記憶部13に記憶された欠陥情報に記録された各欠陥箇所の指摘を、指摘番号ごとに集計した情報の一覧である。指摘番号一覧において、件数は、同じ指摘番号を持つ欠陥箇所の指摘の集計結果である。

【0120】

欠陥指摘画面において、未確定理由一覧は、欠陥情報記憶部13に記憶された欠陥情報に記録された各欠陥箇所の指摘を、未確定理由ごとに集計した情報の一覧である。すなわち、未確定理由一覧は、欠陥未確定の指摘を未確定理由ごとにまとめた情報の一覧となる。未確定理由一覧において、件数は、同じ未確定理由を持つ欠陥未確定の指摘の集計結果である。条件設定は、その未確定理由に対して、未確定理由を解消する条件が設定されているか否かを示している。条件設定が“未”であれば、未確定理由を解消する条件がまだ設定されていない。条件設定が“済”であれば、未確定理由を解消する条件が設定されている。

【0121】

欠陥指摘画面において、指摘一覧は、指摘番号一覧または未確定理由一覧において、選択された項目に属する欠陥箇所の指摘の一例である。

【0122】

図11に示す欠陥指摘画面145aは、指摘番号一覧において、指摘番号5011の項目が選択された状態の欠陥指摘画面の例を示している。図11に示す欠陥指摘画面145

10

20

30

40

50

aでは、指摘番号5011に該当する1件のxと判定された箇所の指摘が提示されている。図11の欠陥指摘画面145aに示すように、指摘一覧では、欠陥箇所の指摘ごとに、欠陥情報130aに記録された各情報が提示される。

【0123】

図12は、本実施の形態による欠陥指摘画面の例(2)を示す図である。

【0124】

図11に示す欠陥指摘画面145aにおいて、ユーザが、未確定理由一覧のgetnum復帰値未確定の項目を選択すると、欠陥指摘画面出力部14は、図12に示す欠陥指摘画面145bをディスプレイ20に表示する。図12に示す欠陥指摘画面145bでは、指摘一覧に、未確定理由がgetnum復帰値未確定である3件の欠陥未確定の指摘が提示されている。

10

【0125】

図12に示すように、本実施の形態では、共通の未確定要素を含む複数の箇所について、それらの複数の箇所のそれぞれを示す情報と、共通の未確定要素を示す情報とを対応付けた欠陥指摘画面145bが表示される。共通の未確定要素を含む複数の箇所の指摘をまとめて提示することにより、これらの指摘を確認するユーザの手間を軽減することができる。

【0126】

図12に示す欠陥指摘画面145bの未確定理由一覧において、getnum復帰値未確定について、未確定理由を解消する条件は未設定である。図12に示す欠陥指摘画面145bの指摘一覧では、未確定理由を解消する条件の設定が未設定であるため、判定がすべて のままである。

20

【0127】

未確定理由を解消する条件を設定する場合には、ユーザは、未確定理由が選択された状態で、メニュー等により条件の設定を実行する。条件設定画面出力部16は、欠陥情報記憶部13に記憶された欠陥情報から、選択された未確定理由を持つ欠陥箇所の情報を抽出し、未確定理由を解消する条件を入力する条件設定画面を生成する。条件設定画面出力部16は、生成された条件設定画面をディスプレイ20に出力する。

【0128】

このとき、ヒント情報生成部15が、ユーザによる未確定理由を解消する条件の入力を支援するヒント情報を生成するようにしてもよい。条件設定画面出力部16は、生成されたヒント情報を含む条件設定画面を、ディスプレイ20に出力する。

30

【0129】

図13は、本実施の形態による条件設定画面の例(1)を示す図である。

【0130】

図13に示す条件設定画面160aは、未確定理由がgetnum復帰値未確定である欠陥未確定の指摘について、条件設定画面出力部16が生成し、ディスプレイ20に出力する条件設定画面の一例である。図13に示す条件設定画面160aにおいて、ユーザは、未確定理由がgetnum復帰値未確定である欠陥未確定の指摘について、まとめて、未確定理由を解消する条件を入力することができる。図13に示す条件設定画面160aは、条件設定の欄と、指摘一覧とを有している。

40

【0131】

条件設定画面において、条件設定の欄は、ユーザが未確定理由を解消する条件を入力し、条件設定の実行を指示する欄である。図13に示す条件設定画面160aにおける条件設定の欄では、未確定理由を解消する条件として、getnum復帰値の条件が入力可能である。

【0132】

条件設定画面において、指摘一覧は、条件を設定する未確定理由を持つ欠陥未確定の指摘の一例である。指摘一覧において、ヒントの欄には、ヒント情報生成部15により生成されたヒント情報が提示される。図13に示す条件設定画面160aにおいて、ヒントの

50

欄には、`getnum`復帰値未確定の未確定理由を持つ欠陥未確定の指摘ごとに、判定がとなる`getnum`復帰値の範囲と、判定が`x`となる`getnum`復帰値の範囲とが提示されている。ユーザは、これらのヒントを参照して、入力する未確定理由を解消する条件を決定する。

【0133】

ここで、図13に示す条件設定画面160aの指摘一覧における、ファイル`a.c`の行番号15に関する欠陥未確定の指摘を例として、ヒント情報の生成を説明する。

【0134】

ヒント情報生成部15は、図10に示す欠陥情報130aにおいて、`getnum`復帰値未確定の未確定理由を持つファイル`a.c`の行番号15のレコードから、判定理由の情報
10
を取得する。図10に示す欠陥情報130aから、判定理由の情報として、配列範囲：`0~14`，添字範囲：`0~getnum`復帰値が得られる。

【0135】

ヒント情報生成部15は、得られた判定理由の情報から、判定がとなる`getnum`復帰値の範囲を求める。配列範囲が`0~14`であるので、添字範囲が配列範囲内となる`getnum`復帰値の範囲は、`0~14`となる。同様に、ヒント情報生成部15は、得られた判定理由の情報から、判定が`x`となる`getnum`復帰値の範囲を求める。配列範囲が
20
`0~14`であるので、添字範囲が配列範囲外となる`getnum`復帰値の範囲は、`0~14`以外となる。

【0136】

このように、ヒント情報生成部15は、欠陥情報の判定理由の情報から、欠陥未確定()の指摘を欠陥でない()ことを示す指摘に変換する条件や、欠陥未確定()の指摘を欠陥である(`x`)ことを示す指摘に変換する条件を生成することができる。

【0137】

図14は、本実施の形態のヒント情報生成部および条件設定画面出力部による条件設定画面出力処理フローチャートである。

【0138】

ユーザにより、未確定理由が指定されて、その指定された未確定理由を解消する条件の設定が指示されると、ヒント情報生成部15は、選択された未確定理由を持つ欠陥未確定の指摘を1つ選択する(ステップS130)。
30

【0139】

ヒント情報生成部15は、欠陥情報記憶部13に記憶された欠陥情報から、選択された欠陥未確定の指摘における判定理由の情報を取得する(ステップS131)。

【0140】

ヒント情報生成部15は、取得された判定理由の情報から、選択された欠陥未確定の指摘を欠陥である(`x`)ことを示す指摘に変換する条件の情報を生成する(ステップS132)。また、ヒント情報生成部15は、取得された判定理由の情報から、選択された欠陥未確定の指摘を欠陥でない()ことを示す指摘に変換する条件の情報を生成する(ステップS133)。これらの生成された情報が、ヒント情報となる。
40

【0141】

ヒント情報生成部15は、指定された未確定理由を持つすべての欠陥未確定の指摘について、処理が終了したかを判定する(ステップS134)。

【0142】

すべての欠陥未確定の指摘について処理が終了していなければ(ステップS134のNO)，ヒント情報生成部15は、ステップS130に戻って、次の欠陥未確定の指摘についての処理に移る。

【0143】

すべての欠陥未確定の指摘について処理が終了していれば(ステップS134のYES)，条件設定画面出力部16は、生成されたヒント情報を含む条件設定画面を、ディスプレイ20に出力する(ステップS135)。
50

【0144】

ディスプレイ20に表示された条件設定画面において、ユーザは、キーボード/マウス30を用いて、未確定理由を解消する条件を入力する。条件入力受付部17は、ユーザによる未確定理由を解消する条件の入力を受け付ける。

【0145】

図15は、本実施の形態による条件設定画面の例(2)を示す図である。

【0146】

図13に示す条件設定画面160aにおいて、ユーザが入力条件の欄に条件を入力し、設定ボタンを押下すると、条件設定画面出力部16は、図15に示す条件設定画面160bをディスプレイ20に表示する。図15に示す条件設定画面160bでは、入力された条件に応じて、指摘一覧に提示された各欠陥未確定の指摘における○の判定が、△の判定や×の判定に変換されている。

10

【0147】

例えば、ファイルa.cの行番号15に関する欠陥未確定の指摘の判定理由において、getnum復帰値に入力された条件0,5~14を設定すると、添字範囲:0~getnum復帰値は、0~0,5~14となる。このとき添字範囲は、配列範囲:0~14の範囲内であるので、図15の条件設定画面160bに示すように、ファイルa.cの行番号15に関する欠陥未確定の指摘の判定は、入力されたgetnum復帰値の条件によって、欠陥でない(○)ことを示す指摘の判定に変換される。

【0148】

また、ファイルa.cの行番号18に関する欠陥未確定の指摘の判定理由において、getnum復帰値に入力された条件0,5~14を設定すると、添字範囲:0~getnum復帰値+10は、0~10,15~24となる。このとき添字範囲は、配列範囲:0~19の範囲を超えているので、図15の条件設定画面160bに示すように、ファイルa.cの行番号18に関する欠陥未確定の指摘の判定は、入力されたgetnum復帰値の条件によって、欠陥である(×)ことを示す指摘の判定に変換される。

20

【0149】

また、ファイルb.cの行番号30に関する欠陥未確定の指摘の判定理由において、getnum復帰値に入力された条件0,5~14を設定すると、除数は、0~10,15~24となる。このとき、除数が0になってしまう可能性があるので、図15の条件設定画面160bに示すように、ファイルb.cの行番号30に関する欠陥未確定の指摘の判定は、入力されたgetnum復帰値の条件によって、欠陥である(×)ことを示す指摘の判定に変換される。

30

【0150】

図15に示す条件設定画面160bの指摘一覧において、ファイルb.cの行番号30に関する欠陥未確定の指摘は、0除算についての欠陥箇所の指摘である。図15の条件設定画面160bに示すように、配列外参照と0除算のように異なる欠陥の欠陥未確定の指摘についても、同じ未確定理由を持つ欠陥未確定の指摘であれば、ユーザによる1回の未確定理由を解消する条件の入力で、まとめて変換することができる。個々の欠陥未確定の指摘ごとや欠陥の種類ごとに、未確定理由を解消する条件を設定する必要がないので、ユーザの負担は少ない。

40

【0151】

条件入力受付部17により受け付けられた未確定理由を解消する条件は、未確定理由に対応付けられて、条件情報記憶部18に記憶された条件情報に記録される。

【0152】

図16は、本実施の形態による条件情報の例を示す図である。

【0153】

図16に示す条件情報180は、getnum復帰値未確定の未確定理由を解消する条件が記録された条件情報の例を示している。図16に示す条件情報180において、入力条件は、ユーザにより入力された未確定理由を解消する条件である。図16に示す条件情

50

報180では、getnum復帰値未確定の未確定理由に対応付けて、その未確定理由を解消する条件としてgetnum復帰値の条件0, 5~14が記録されている。

【0154】

指摘変換部141は、欠陥指摘画面出力部14が欠陥情報に基づいた欠陥指摘画面を出力する際に、条件情報で条件が対応付けられている未確定理由を持つ欠陥未確定の指摘を、その条件で変換する。欠陥指摘画面出力部14は、条件で変換された欠陥未確定の指摘については、その変換結果を用いて欠陥指摘画面を生成し、出力する。

【0155】

図17は、本実施の形態による欠陥指摘画面の例(3)を示す図である。

【0156】

図17に示す欠陥指摘画面145cは、図11に示す欠陥指摘画面145aに対してgetnum復帰値未確定の未確定理由を解消する条件が設定された後の、欠陥指摘画面の例である。図17の欠陥指摘画面145cに示すように、未確定理由一覧のgetnum復帰値未確定の項目において、条件設定が、未確定理由を解消する条件が設定されていることを示す“済”となっている。

【0157】

図15の条件設定画面160bに示すように、getnum復帰値未確定の未確定理由を持つ指摘番号5012の2件の欠陥未確定()の指摘は、設定された条件によって、1件の欠陥でない()ことを示す指摘と、1件の欠陥である(x)ことを示す指摘とに変換される。この変換により、欠陥指摘画面でユーザに提示される指摘番号5012の欠陥未確定()の指摘が2件減る。図17に示すように、欠陥指摘画面145cの指摘番号一覧の指摘番号5012の項目の件数が、図11に示す欠陥指摘画面145aの20件から2件減って18件となっている。減った欠陥未確定()の指摘のうち1件は、指摘番号5011の欠陥である(x)ことを示す欠陥に変換されている。そのため、図17に示すように、欠陥指摘画面145cの指摘番号一覧の指摘番号5011の項目の件数が、図11に示す欠陥指摘画面145aの1件から増えて2件となっている。

【0158】

また、図15の条件設定画面160bに示すように、getnum復帰値未確定の未確定理由を持つ指摘番号5042の欠陥未確定()の指摘は、設定された条件によって、欠陥である(x)ことを示す指摘に変換される。この変換により、欠陥指摘画面でユーザに提示される指摘番号5042の欠陥未確定()の指摘が1件減る。図17に示すように、欠陥指摘画面145cの指摘番号一覧の指摘番号5042の項目の件数が、図11に示す欠陥指摘画面145aの1件から減って、なくなっている。そのかわりに、図17に示す欠陥指摘画面145cの指摘番号一覧では、指摘番号5041の項目が増えている。図17に示す欠陥指摘画面145cにおいて、指摘番号5041は、指摘番号5042が0除算についてと判定された箇所の指摘番号であるのに対して、0除算についてxと判定された箇所の指摘番号である。

【0159】

なお、本実施の形態では、欠陥でない()ことを示す欠陥に変換されたものについては、欠陥指摘画面上での指摘を行わない。図17に示す欠陥指摘画面145cでは、欠陥未確定()の指摘が欠陥でない()ことを示す欠陥に変換された分だけ、図11に示す欠陥指摘画面145aに比べて欠陥箇所の指摘数が減っている。

【0160】

図17に示す欠陥指摘画面145cは、指摘番号一覧において、指摘番号5011の項目が選択された状態の欠陥指摘画面の例を示している。図17に示す欠陥指摘画面145cでは、指摘番号5011に該当する2件のxと判定された箇所の指摘が提示されている。そのうちの1件の指摘は、欠陥未確定の指摘から変換された欠陥であることを示す指摘である。

【0161】

図18は、本実施の形態の欠陥指摘画面出力部による欠陥指摘画面出力処理フローチャ

10

20

30

40

50

ートである。

【0162】

欠陥指摘画面出力部14は、欠陥情報記憶部13に記憶された欠陥情報に記録された欠陥箇所から、指摘する欠陥箇所を1つ選択する(ステップS140)。

【0163】

欠陥指摘画面出力部14において、指摘変換部141は、選択された欠陥箇所の指摘が、欠陥未確定の指摘であるかを判定する(ステップS141)。

【0164】

欠陥未確定の指摘であれば(ステップS141のYES)、指摘変換部141は、条件情報記憶部18に記憶された条件情報で、選択された欠陥未確定の指摘の未確定理由に応じた未確定理由を解消する条件が設定されているかを判定する(ステップS142)。

10

【0165】

未確定理由を解消する条件が設定されている場合には(ステップS142のYES)、指摘変換部141は、条件情報から、選択された欠陥未確定の指摘の未確定理由に対応付けられた、未確定理由を解消する条件を取得する(ステップS143)。指摘変換部141は、取得された未確定理由を解消する条件で、選択された欠陥未確定の指摘を変換する(ステップS144)。

【0166】

欠陥未確定の指摘でない場合(ステップS141のNO)、または未確定理由を解消する条件が設定されていない場合には(ステップS142のNO)、指摘変換部141は、選択された欠陥箇所の指摘の変換を行わない。

20

【0167】

欠陥指摘画面出力部14は、指摘するすべての欠陥箇所について処理が終了したかを判定する(ステップS145)。

【0168】

指摘するすべての欠陥箇所について処理が終了していなければ(ステップS145のNO)、欠陥指摘画面出力部14は、ステップS140に戻って、次の欠陥箇所の処理に移る。

【0169】

指摘するすべての欠陥箇所について処理が終了していれば(ステップS145のYES)、欠陥指摘画面出力部14は、出力する欠陥箇所の指摘を集計する(ステップS146)。ここでは、例えば、図17に示す欠陥指摘画面145cの例において、指摘番号一覧の各指摘番号ごとの欠陥箇所の指摘の件数が集計される。このとき、欠陥指摘画面出力部14は、欠陥未確定の指摘が未確定理由を解消する条件で変換された結果に基づいて、集計を行う。

30

【0170】

欠陥指摘画面出力部14は、欠陥指摘画面を生成する(ステップS147)。このとき、欠陥指摘画面出力部14は、未確定理由を解消する条件が設定された欠陥未確定の指摘については、未確定理由を解消する条件で変換された結果を含む欠陥指摘画面を生成する。

40

【0171】

欠陥指摘画面出力部14は、生成された欠陥指摘画面を、ディスプレイ20に出力する(ステップS148)。出力される欠陥指摘画面は、未確定理由を解消する条件で欠陥未確定の指摘が変換された結果を含むものとなる。

【0172】

図19は、本実施の形態の指摘変換部による指摘変換処理フローチャートである。

【0173】

ここでは、図18のステップS144の処理において、配列外参照についての欠陥未確定の指摘を変換する例を説明する。

【0174】

50

指摘変換部 141 は、変換する対象の欠陥未確定の指摘の判定理由において、未確定理由となる要素に対して、設定された未確定理由を解消する条件を適用する（ステップ S150）。例えば、未確定理由となる要素が判定理由の添字範囲における `getnum` 復帰値である場合に、その `getnum` 復帰値に、設定された `getnum` 復帰値の条件を適用する。

【0175】

指摘変換部 141 は、未確定理由を解消する条件適用後に、添字範囲が配列範囲内であるかを判定する（ステップ S151）。

【0176】

添字範囲が配列範囲内であれば（ステップ S151 の YES）、指摘変換部 141 は、対象の欠陥未確定（ ）の指摘を、欠陥でない（ ）ことを示す指摘に変換する（ステップ S152）。

10

【0177】

添字範囲が配列範囲内でなければ（ステップ S151 の NO）、指摘変換部 141 は、対象の欠陥未確定（ ）の指摘を、欠陥である（×）ことを示す指摘に変換する（ステップ S153）。

【0178】

ここまで説明したように、本実施の形態の静的解析装置 10 は、欠陥であることを示す指摘だけではなく、欠陥未確定の指摘の出力も行う。これにより、より多くの欠陥の可能性のある箇所の指摘が可能となる。

20

【0179】

このとき、ユーザにより入力された未確定理由を解消する条件で、欠陥未確定の指摘が欠陥であることを示す指摘や欠陥でないことを示す指摘に変換される。そのため、欠陥未確定の指摘を出力しつつも、出力される欠陥未確定の指摘の数を減らすことが可能である。

【0180】

また、欠陥未確定の指摘については未確定理由ごとにまとめた情報がユーザに提示され、ユーザは、未確定理由ごとにまとめて、その未確定理由を解消する条件を入力する条件を入力できる。これにより、欠陥未確定の指摘について、1つ1つ調べて欠陥を判断するといったユーザの負担を軽減することができる。

30

【0181】

ここまで、主に配列外参照の欠陥を指摘する例について説明したが、本実施の形態の静的解析により抽出された欠陥箇所の指摘は、当然他の様々な種類の欠陥でも実施可能である。以下では、さらに 0 番地参照の欠陥の例と、型の範囲を超える代入の欠陥の例を紹介する。

【0182】

まず、0 番地参照に関する欠陥を指摘する例を説明する。0 番地参照は、上述したように、ポインタ変数に `NULL` を設定しただけでポインタの指す領域をアクセスすることで、0 番地をアクセスしてしまう欠陥である。

【0183】

図 20 は、本実施の形態によるソースプログラムの例（2）を示す図である。

40

【0184】

図 20 に示すソースプログラム 110 b は、0 番地参照について欠陥であるか否かを確定できない箇所を含むソースプログラムの一例である。

【0185】

ここでは、欠陥情報生成部 12 は、図 20 に示すソースプログラム 110 b の静的解析を行い、欠陥情報を生成する。

【0186】

図 21 は、本実施の形態による欠陥情報の例（2）を示す図である。

【0187】

50

図 2 1 に示す欠陥情報 1 3 0 b は、図 2 0 に示すソースプログラム 1 1 0 b から抽出された 0 番地参照に関する欠陥箇所が記録された欠陥情報の例である。

【 0 1 8 8 】

図 2 1 に示す欠陥情報 1 3 0 b において、ファイル c . c は、図 2 0 に示すソースプログラム 1 1 0 b のファイルを示している。図 2 1 に示す欠陥情報 1 3 0 b において、ファイル c . c の行番号 2 のレコードで示される欠陥箇所の情報は、0 番地参照について欠陥であるか否かを確定できないとされた箇所の情報である。その未確定理由は、get pointer の復帰値が、図 2 0 に示すソースプログラム 1 1 0 b の解析だけでは特定できないということである。

【 0 1 8 9 】

欠陥指摘画面出力部 1 4 は、図 2 1 に示す欠陥情報 1 3 0 b に基づいて、欠陥指摘画面をディスプレイ 2 0 に出力する。

【 0 1 9 0 】

図 2 2 は、本実施の形態による欠陥指摘画面の例 (4) を示す図である。

【 0 1 9 1 】

図 2 2 に示す欠陥指摘画面 1 4 5 d は、図 2 1 に示す欠陥情報 1 3 0 b に基づいて生成され、ディスプレイ 2 0 に出力された欠陥指摘画面の一例である。図 2 2 に示す欠陥指摘画面 1 4 5 d では、未確定理由一覧において get pointer 復帰値未確定の項目が選択され、指摘一覧に、get pointer 復帰値未確定の未確定理由を持つ、0 番地参照に関する欠陥未確定の指摘が提示されている。

【 0 1 9 2 】

図 2 3 は、本実施の形態による条件設定画面の例 (3) を示す図である。

【 0 1 9 3 】

図 2 3 に示す条件設定画面 1 6 0 c は、図 2 2 に示す欠陥指摘画面 1 4 5 d において、get pointer 復帰値未確定が選択された状態で、ユーザに条件設定が指示されたときに、ディスプレイ 2 0 に出力される条件設定画面の例である。

【 0 1 9 4 】

ユーザは、図 2 3 に示す条件設定画面 1 6 0 c において、get pointer 復帰値未確定の未確定理由を解消する条件として、get pointer 復帰値の条件を入力する。条件入力受付部 1 7 は、ユーザによる get pointer 復帰値の条件の入力を受け付ける。

【 0 1 9 5 】

図 2 4 は、本実施の形態による条件設定画面の例 (4) を示す図である。

【 0 1 9 6 】

図 2 4 に示す条件設定画面 1 6 0 d は、図 2 3 に示す条件設定画面 1 6 0 c において get pointer 復帰値の条件が入力された後の条件設定画面の例を示している。図 2 4 に示す条件設定画面 1 6 0 d では、ユーザにより get pointer 復帰値の条件 not NULL が入力され、get pointer 復帰値未確定の未確定理由を解消する条件として設定されている。

【 0 1 9 7 】

図 2 4 の条件設定画面 1 6 0 d において、指摘一覧では、ファイル c . c の行番号 2 に関する欠陥未確定の指摘の判定理由において、get pointer 復帰値に入力された条件 not NULL が設定される。このとき、変数 p が参照する get pointer 復帰値は NULL 以外となるので、図 2 4 の条件設定画面 1 6 0 d に示すように、ファイル c . c の行番号 2 に関する欠陥未確定の指摘の判定は、欠陥でない () ことを示す指摘の判定に変換される。

【 0 1 9 8 】

ユーザにより入力され、設定された get pointer 復帰値の条件は、get pointer 復帰値未確定の未確定理由と対応付けられて、条件情報記憶部 1 8 に記憶された条件情報に記録される。欠陥指摘画面出力部 1 4 において、指摘変換部 1 4 1 は、条件

10

20

30

40

50

情報に基づいて、`getpointer` 復帰値未確定の未確定理由を持つ欠陥未確定の指摘を、欠陥であることを示す指摘や欠陥でないことを示す指摘に変換する。欠陥指摘画面出力部 14 は、`getpointer` 復帰値未確定の未確定理由を持つ欠陥未確定の指摘については、設定された `getpointer` 復帰値の条件で変換された結果を用いて欠陥指摘画面を生成し、ディスプレイ 20 に出力する。

【0199】

図 25 は、本実施の形態による欠陥指摘画面の例 (5) を示す図である。

【0200】

図 25 に示す欠陥指摘画面 145 e は、図 22 に示す欠陥指摘画面 145 d に対して `getpointer` 復帰値未確定の未確定理由を解消する条件が設定された後の、欠陥指摘画面の例である。図 25 の欠陥指摘画面 145 e に示すように、未確定理由一覧の `getpointer` 復帰値未確定の項目において、条件設定が、未確定理由を解消する条件が設定されていることを示す“済”となっている。

10

【0201】

図 24 の条件設定画面 160 d に示すように、`getpointer` 復帰値未確定の未確定理由を持つ指摘番号 5022 の欠陥未確定 () の指摘は、設定された条件によって、欠陥でない () ことを示す指摘に変換される。この変換により、欠陥指摘画面でユーザに提示される指摘番号 5022 の欠陥未確定 () の指摘が 1 件減る。図 25 に示す欠陥指摘画面 145 e では、指摘番号一覧の指摘番号 5022 の項目の件数が図 22 に示す欠陥指摘画面 145 d の 1 件から減って 0 件となってしまったため、指摘番号 5022 の項目がなくなっている。

20

【0202】

なお、本実施の形態の説明を簡単にするために、図 22 に示す欠陥指摘画面 145 d は 0 番地参照についての欠陥未確定の指摘 1 件のみの欠陥指摘画面としていたため、図 25 に示す欠陥指摘画面 145 e は、欠陥箇所の指摘がない欠陥指摘画面となっている。実際の欠陥指摘画面には、関連する複数のソースプログラムに対する、複数の欠陥の種類についての静的解析の結果として抽出された多数の欠陥箇所の指摘が提示されるので、欠陥箇所の指摘がない欠陥指摘画面となることはまずない。

【0203】

次に、型の範囲を超える代入に関する欠陥を指摘する例を説明する。型の範囲を超える代入は、代入式において左式の型の幅よりも右式の型の幅が大きい場合に、正しい値が代入されない欠陥である。

30

【0204】

図 26 は、本実施の形態によるソースプログラムの例 (3) を示す図である。

【0205】

図 26 に示すソースプログラム 110 c は、型の範囲を超える代入について欠陥であるか否かを確定できない箇所を含むソースプログラムの一例である。

【0206】

ここでは、欠陥情報生成部 12 は、図 26 に示すソースプログラム 110 c の静的解析を行い、欠陥情報を生成する。

40

【0207】

図 27 は、本実施の形態による欠陥情報の例 (3) を示す図である。

【0208】

図 27 に示す欠陥情報 130 c は、図 26 に示すソースプログラム 110 c から抽出された型の範囲を超える代入に関する欠陥箇所が記録された欠陥情報の例である。

【0209】

図 27 に示す欠陥情報 130 c において、ファイル d.c は、図 26 に示すソースプログラム 110 c のファイルを示している。図 27 に示す欠陥情報 130 c において、ファイル d.c の行番号 5 のレコードで示される欠陥箇所の情報は、型の範囲を超える代入について欠陥であるか否かを確定できないとされた箇所の情報である。その未確定理由は、

50

d a t aの画面入力値が，図 2 6 に示すソースプログラム 1 1 0 c の解析だけでは特定できないということである。

【 0 2 1 0 】

欠陥指摘画面出力部 1 4 は，図 2 7 に示す欠陥情報 1 3 0 c に基づいて，欠陥指摘画面をディスプレイ 2 0 に出力する。

【 0 2 1 1 】

図 2 8 は，本実施の形態による欠陥指摘画面の例（ 6 ）を示す図である。

【 0 2 1 2 】

図 2 8 に示す欠陥指摘画面 1 4 5 f は，図 2 7 に示す欠陥情報 1 3 0 c に基づいて生成され，ディスプレイ 2 0 に出力された欠陥指摘画面の一例である。図 2 8 に示す欠陥指摘画面 1 4 5 f では，未確定理由一覧において d a t a の画面入力値未確定の項目が選択され，指摘一覧に，d a t a の画面入力値未確定の未確定理由を持つ，型の範囲を超える代入に関する欠陥未確定の指摘が提示されている。

10

【 0 2 1 3 】

図 2 9 は，本実施の形態による条件設定画面の例（ 5 ）を示す図である。

【 0 2 1 4 】

図 2 9 に示す条件設定画面 1 6 0 e は，図 2 8 に示す欠陥指摘画面 1 4 5 f において，d a t a の画面入力値未確定が選択された状態で，ユーザに条件設定が指示されたときに，ディスプレイ 2 0 に出力される条件設定画面の例である。

【 0 2 1 5 】

ユーザは，図 2 9 に示す条件設定画面 1 6 0 e において，d a t a の画面入力値未確定の未確定理由を解消する条件として，d a t a の画面入力値の条件を入力する。条件入力受付部 1 7 は，ユーザによる d a t a の画面入力値の条件の入力を受け付ける。

20

【 0 2 1 6 】

図 3 0 は，本実施の形態による条件設定画面の例（ 6 ）を示す図である。

【 0 2 1 7 】

図 3 0 に示す条件設定画面 1 6 0 f は，図 2 9 に示す条件設定画面 1 6 0 e において d a t a の画面入力値の条件が入力された後の条件設定画面の例を示している。図 3 0 に示す条件設定画面 1 6 0 f では，ユーザにより d a t a の画面入力値の条件 0 ~ 6 5 5 3 5 が入力され，d a t a の画面入力値未確定の未確定理由を解消する条件として設定されている。

30

【 0 2 1 8 】

図 3 0 の条件設定画面 1 6 0 f において，指摘一覧では，ファイル d . c の行番号 5 に関する欠陥未確定の指摘の判定理由において，d a t a の画面入力値に入力された条件 0 ~ 6 5 5 3 5 が設定される。このとき，代入式の右辺の範囲は左辺の範囲を超えているので，図 3 0 の条件設定画面 1 6 0 f に示すように，ファイル d . c の行番号 5 に関する欠陥未確定の指摘の判定は，欠陥である（ x ）ことを示す指摘の判定に変換される。

【 0 2 1 9 】

ユーザにより入力され，設定された d a t a の画面入力値の条件は，d a t a の画面入力値未確定の未確定理由と対応付けられて，条件情報記憶部 1 8 に記憶された条件情報に記録される。欠陥指摘画面出力部 1 4 において，指摘変換部 1 4 1 は，条件情報に基づいて，d a t a の画面入力値未確定の未確定理由を持つ欠陥未確定の指摘を，欠陥であることを示す指摘や欠陥でないことを示す指摘に変換する。欠陥指摘画面出力部 1 4 は，d a t a の画面入力値未確定の未確定理由を持つ欠陥未確定の指摘については，設定された d a t a の画面入力値の条件で変換された結果を用いて欠陥指摘画面を生成し，ディスプレイ 2 0 に出力する。

40

【 0 2 2 0 】

図 3 1 は，本実施の形態による欠陥指摘画面の例（ 7 ）を示す図である。

【 0 2 2 1 】

図 3 1 に示す欠陥指摘画面 1 4 5 g は，図 2 8 に示す欠陥指摘画面 1 4 5 f に対して d

50

a t aの画面入力値未確定の未確定理由を解消する条件が設定された後の、欠陥指摘画面の例である。図31の欠陥指摘画面145gに示すように、未確定理由一覧のd a t aの画面入力値未確定の項目において、条件設定が、未確定理由を解消する条件が設定されていることを示す“済”となっている。

【0222】

図30の条件設定画面160fに示すように、d a t aの画面入力値未確定の未確定理由を持つ指摘番号5052の欠陥未確定()の指摘は、設定された条件によって、欠陥である(x)ことを示す指摘に変換される。この変換により、欠陥指摘画面でユーザに提示される指摘番号5052の欠陥未確定()の指摘が1件減り、変換結果として指摘番号5051の欠陥である(x)ことを示す指摘が増える。図31に示す欠陥指摘画面145gでは、指摘番号一覧の指摘番号5052の項目の件数が図28に示す欠陥指摘画面145fの1件から減って0件となってしまったため、指摘番号5052の項目がなくなり、指摘番号5051の項目が増えている。また、図31に欠陥指摘画面145gの指摘一覧には、欠陥未確定()の指摘から変換された、欠陥である(x)ことを示す指摘が提示されている。

10

【0223】

なお、型の範囲を超える代入の例でも、0番地参照の例と同様に、本実施の形態の説明を簡単にするために、型の範囲を超える代入の欠陥のみの指摘が提示された欠陥指摘画面の例を挙げている。

【0224】

以上、本実施の形態について説明したが、本発明はその主旨の範囲において種々の変形が可能であることは当然である。

20

【0225】

例えば、本実施の形態では、C言語で記述されたソースプログラムの静的解析の例が示されているが、静的解析の対象となるソースプログラムを記述するプログラミング言語が、JavaやFortranなどの他のプログラミング言語であってもよい。なお、Javaは、米国サンマイクロシステムズ社の商標である。

【符号の説明】

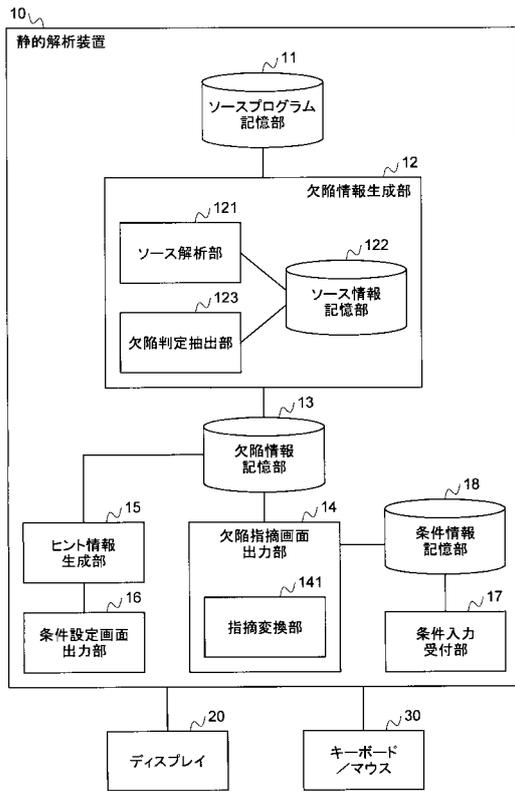
【0226】

- 10 静的解析装置
- 11 ソースプログラム記憶部
- 12 欠陥情報生成部
- 121 ソース解析部
- 122 ソース情報記憶部
- 123 欠陥判定抽出部
- 13 欠陥情報記憶部
- 14 欠陥指摘画面出力部
- 141 指摘変換部
- 15 ヒント情報生成部
- 16 条件設定画面出力部
- 17 条件入力受付部
- 18 条件情報記憶部
- 20 ディスプレイ
- 30 キーボード/マウス

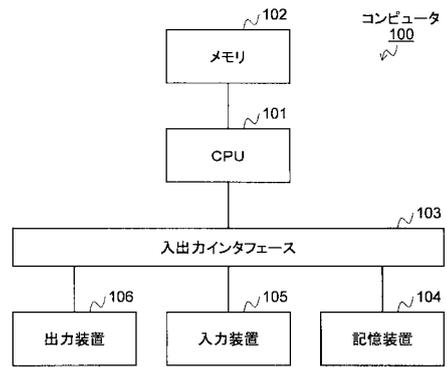
30

40

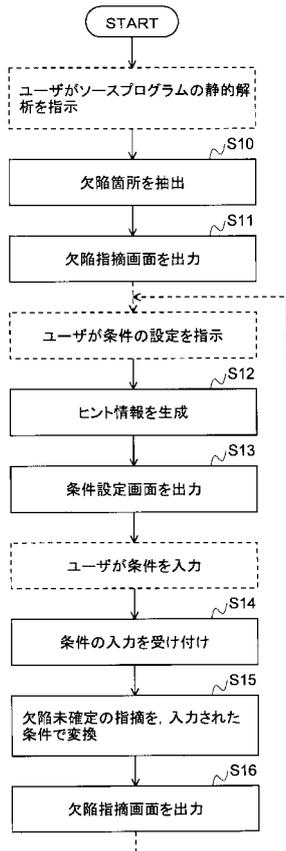
【図1】



【図2】



【図3】



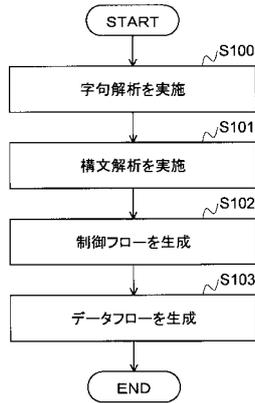
【図4】

```

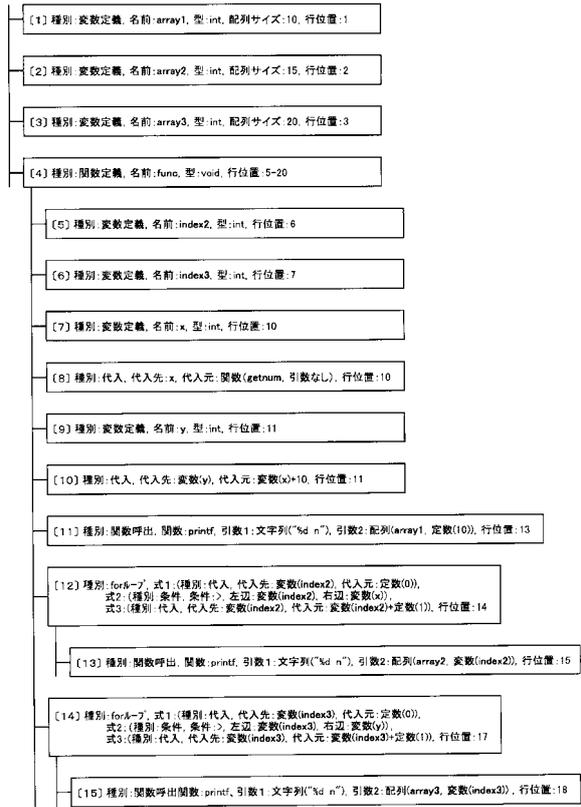
ソースプログラム 110a
0001 int array1[10];
0002 int array2[15];
0003 int array3[20];
0004
0005 void func() {
0006     int index2;
0007     int index3;
0008
0009     int x = getnum();
0010     int y = x + 10;
0011
0012     printf(" %d n", array1[ 10 ] );
0013     for( index2=0; index2<x; index2++){
0014         printf(" %d n", array2[ index2 ] );
0015     }
0016     for( index3=0; index3<y; index3++){
0017         printf(" %d n", array3[ index3 ] );
0018     }
0019 }
0020 }

```

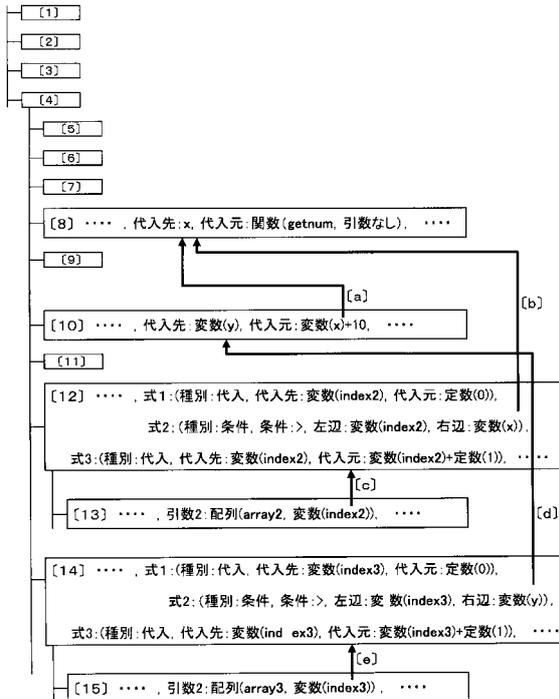
【図5】



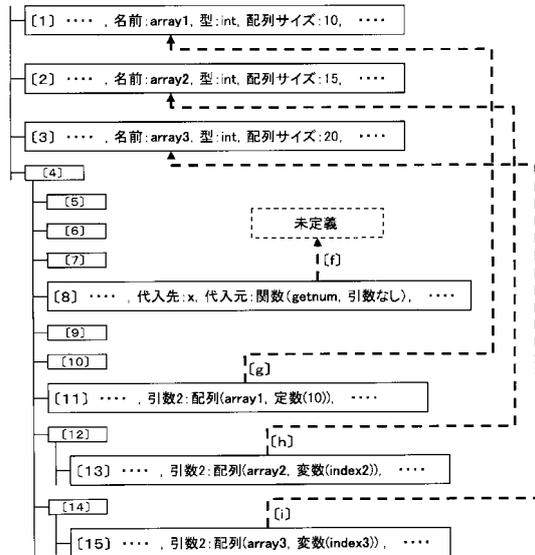
【図6】



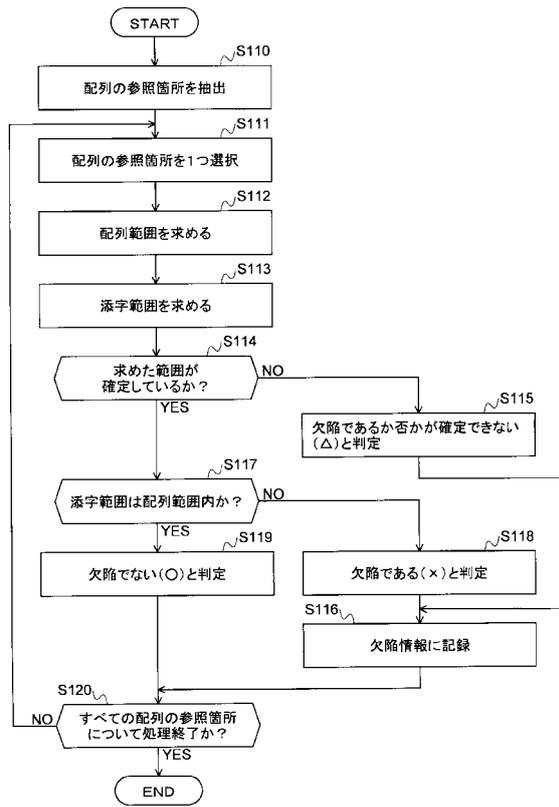
【図7】



【図8】



【図9】



【図10】

欠陥情報 130a

ファイル	行番号	指摘番号	メッセージ	判定理由	未確定理由
...
a.c	13	5011	array1[10]で配列外参照が発生します。	配列範囲:0~9 添字範囲:10	-
a.c	15	5012	array2[index2]で配列外参照が発生する可能性があります。	配列範囲:0~14 添字範囲:0~getnum復帰値	getnum復帰値未確定
a.c	18	5012	array3[index3]で配列外参照が発生する可能性があります。	配列範囲:0~19 添字範囲:0~getnum復帰値+10	getnum復帰値未確定
...

【図11】

欠陥指摘画面 145a

指摘番号一覧			未確定理由一覧		
指摘番号	判定	件数	未確定理由	件数	条件設定
5011	X	1件	getnum復帰値未確定	3件	未
5012	△	20件	画面入力値未確定	20件	未
5042	△	1件	ファイル読込内容未確定	30件	未

指摘一覧				
ファイル	行番号	指摘番号	メッセージ	判定理由
a.c	13	5011	array1[10]で配列外参照が発生します。	配列範囲:0~9 添字範囲:10

【図12】

欠陥指摘画面 145b

指摘番号一覧			未確定理由一覧		
指摘番号	判定	件数	未確定理由	件数	条件設定
5011	X	1件	getnum復帰値未確定	3件	未
5012	△	20件	画面入力値未確定	20件	未
5042	△	1件	ファイル読込内容未確定	30件	未

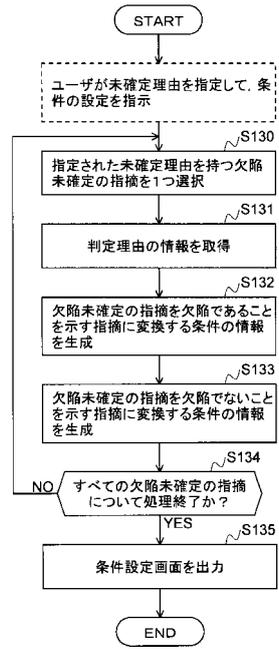
指摘一覧					
ファイル	行番号	指摘番号	メッセージ	判定理由	判定
a.c	15	5012	array2[index2]で配列外参照が発生する可能性があります。	配列範囲:0~14 添字範囲:0~getnum復帰値	△
a.c	18	5012	array3[index3]で配列外参照が発生する可能性があります。	配列範囲:0~19 添字範囲:0~getnum復帰値+10	△
b.c	30	5042	y/getnum()で0除算が発生する可能性があります。	除数:getnum復帰値	△

【図13】

条件設定画面 160a

条件設定							
未確定理由			入力条件		設定		
getnum復帰値未確定					設定		
指摘一覧							
ファイル	行番号	指摘番号	メッセージ	判定理由	ヒント		判定
					判定	条件	
a.c	15	5012	array2[index2]で配列外参照が発生する可能性があります。	配列範囲: 0~14 添字範囲: 0~getnum復帰値	○	0~14	△
a.c	18	5012	array3[index3]で配列外参照が発生する可能性があります。	配列範囲: 0~19 添字範囲: 0~getnum復帰値+10	○	0~9	△
b.c	30	5042	y/getnum()で0除算が発生する可能性があります。	除数: getnum復帰値	○	0以外	△
					x	0	△

【図14】



【図15】

条件設定画面 160b

条件設定							
未確定理由			入力条件		設定		
getnum復帰値未確定			0, 5~14		設定		
指摘一覧							
ファイル	行番号	指摘番号	メッセージ	判定理由	ヒント		判定
					判定	条件	
a.c	15	5012	array2[index2]で配列外参照が発生する可能性があります。	配列範囲: 0~14 添字範囲: 0~getnum復帰値	○	0~14	○
a.c	18	5012	array3[index3]で配列外参照が発生する可能性があります。	配列範囲: 0~19 添字範囲: 0~getnum復帰値+10	○	0~9	x
b.c	30	5042	y/getnum()で0除算が発生する可能性があります。	除数: getnum復帰値	○	0以外	x
					x	0	x

【図17】

欠陥指摘画面 145c

指摘番号一覧			未確定理由一覧		
指摘番号	判定	件数	未確定理由	件数	条件設定
5011	x	2件	getnum復帰値未確定	3件	済
5012	△	18件	画面入力値未確定	20件	未
5041	x	1件	ファイル読込内容未確定	30件	未

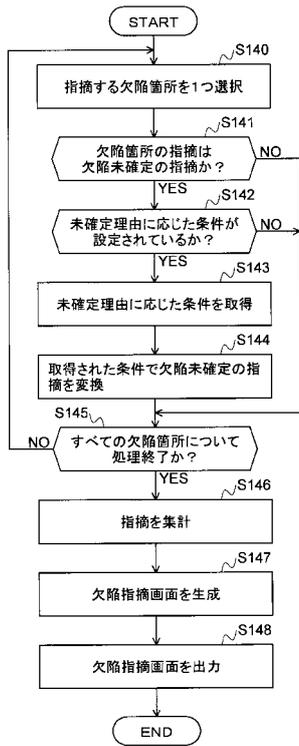
指摘一覧				
ファイル	行番号	指摘番号	メッセージ	判定理由
a.c	13	5011	array1[10]で配列外参照が発生します。	配列範囲: 0~9 添字範囲: 10
a.c	18	5011	array3[index3]で配列外参照が発生します。	配列範囲: 0~19 添字範囲: 0~getnum復帰値(0, 5~14)+10

【図16】

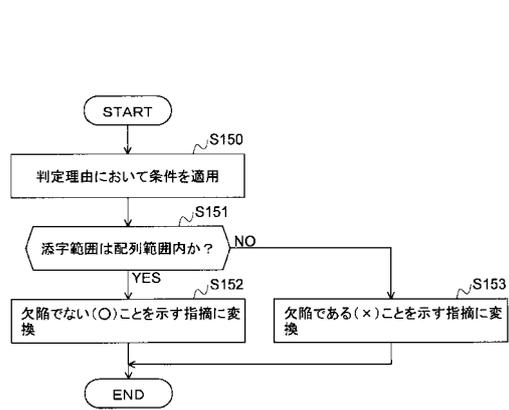
条件情報 180

未確定理由	入力条件
...	...
getnum復帰値未確定	0, 5~14
...	...

【図18】



【図19】



【図20】

ソースプログラム 110b

```

1: void f1(int *p){
2:   int x = *p;
3: }
4:
5: void f2(){
6:   char *x = getpointer();
7:   f1(x);
8: }
    
```

【図22】

欠陥指摘画面 145d

指摘番号一覧			未確定理由一覧		
指摘番号	判定	件数	未確定理由	件数	条件設定
5022	△	1件	getpointer復帰値未確定	1件	未

指摘一覧					
ファイル	行番号	指摘番号	メッセージ	判定理由	判定
c.c	2	5022	変数"p"は、0番地を参照する可能性があります。	参照:getpointer復帰値	△

【図21】

欠陥情報 130b

ファイル	行番号	指摘番号	メッセージ	判定理由	未確定理由
...
c.c	2	5022	変数"p"は、0番地を参照する可能性があります。	参照:getpointer復帰値	getpointer復帰値未確定
...

【図23】

条件設定画面 160c

条件設定			
未確定理由	入力条件	設定	
getpointer復帰値未確定		設定	

指摘一覧							
ファイル	行番号	指摘番号	メッセージ	判定理由	ヒント		判定
					判定	条件	
c.c	2	5022	変数"p"は、0番地を参照する可能性があります。	参照:getpointer復帰値	○	NULL以外	△
					x	NULL	

【図24】

条件設定画面 160d

条件設定					
未確定理由			入力条件		設定
getpointer復帰値未確定			not NULL		

指摘一覧							
ファイル	行番号	指摘番号	メッセージ	判定理由	ヒント	判定	
					判定		条件
c.c	2	5022	変数"s"は、0番地を参照する可能性があります。	参照: getpointer復帰値	○	NULL以外	○
					×	NULL	

【図26】

ソースプログラム 110c

```

1: void f1() {
2:   long data;
3:   scanf("%d", &data);
4:   short s;
5:   s = data;

```

【図25】

欠陥指摘画面 145e

指摘番号一覧			未確定理由一覧		
指摘番号	判定	件数	未確定理由	件数	条件設定
-	-	-	getpointer復帰値未確定	1件	済

指摘一覧					
ファイル	行番号	指摘番号	メッセージ	判定理由	判定
-	-	-	-	-	-

【図27】

欠陥情報 130c

ファイル	行番号	指摘番号	メッセージ	判定理由	未確定理由
...
d.c	5	5052	代入式 s = data の右辺の方がサイズが大きいため、正しい値が代入されない可能性があります。(左: short, 右: long)	左辺の範囲: -32768 ~ 32767 右辺の範囲: data の画面入力値	data の画面入力値未確定
...

【図28】

欠陥指摘画面 145f

指摘番号一覧			未確定理由一覧		
指摘番号	判定	件数	未確定理由	件数	条件設定
5052	△	1件	dataの画面入力値未確定	1件	未

指摘一覧					
ファイル	行番号	指摘番号	メッセージ	判定理由	判定
d.c	5	5052	代入式 s = data の右辺の方がサイズが大きいため、正しい値が代入されない可能性があります。(左: short, 右: long)	左辺の範囲: -32768 ~ 32767 右辺の範囲: data の画面入力値	△

【図30】

条件設定画面 160f

条件設定					
未確定理由			入力条件		設定
dataの画面入力値未確定			0 ~ 65535		

指摘一覧							
ファイル	行番号	指摘番号	メッセージ	判定理由	ヒント	判定	
					判定		条件
d.c	5	5052	代入式 s = data の右辺の方がサイズが大きいため、正しい値が代入されない可能性があります。(左: short, 右: long)	左辺の範囲: -32768 ~ 32767 右辺の範囲: data の画面入力値	○	-32768 ~ 32767	×
					×	-32768 ~ 32767 以外	

【図29】

条件設定画面 160e

条件設定					
未確定理由			入力条件		設定
dataの画面入力値未確定					

指摘一覧							
ファイル	行番号	指摘番号	メッセージ	判定理由	ヒント	判定	
					判定		条件
d.c	5	5052	代入式 s = data の右辺の方がサイズが大きいため、正しい値が代入されない可能性があります。(左: short, 右: long)	左辺の範囲: -32768 ~ 32767 右辺の範囲: data の画面入力値	○	-32768 ~ 32767	△
					×	-32768 ~ 32767 以外	

【図31】

欠陥指摘画面 145g

指摘番号一覧			未確定理由一覧		
指摘番号	判定	件数	未確定理由	件数	条件設定
5051	×	1件	dataの画面入力値未確定	1件	済

指摘一覧					
ファイル	行番号	指摘番号	メッセージ	判定理由	判定
d.c	5	5051	代入式 s = data の右辺の方がサイズが大きいため、正しい値が代入されません。(左: short, 右: long)	左辺の範囲: -32768 ~ 32767 右辺の範囲: data の画面入力値 (0 ~ 65535)	

フロントページの続き

(56)参考文献 特開平06-236295(JP,A)
特開2000-347894(JP,A)
特開2004-206501(JP,A)

(58)調査した分野(Int.Cl., DB名)
G06F 9/45
G06F 11/28
G06F 11/36