

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5678689号  
(P5678689)

(45) 発行日 平成27年3月4日(2015.3.4)

(24) 登録日 平成27年1月16日(2015.1.16)

(51) Int. Cl.		F I			
<b>G06F 11/34</b>	<b>(2006.01)</b>	G06F 11/34		S	
<b>G06F 9/44</b>	<b>(2006.01)</b>	G06F 9/06		62OK	
<b>G06F 11/28</b>	<b>(2006.01)</b>	G06F 11/28		31OD	

請求項の数 8 (全 28 頁)

(21) 出願番号	特願2011-16842 (P2011-16842)	(73) 特許権者	000005223 富士通株式会社
(22) 出願日	平成23年1月28日 (2011.1.28)		神奈川県川崎市中原区上小田中4丁目1番1号
(65) 公開番号	特開2012-155680 (P2012-155680A)	(74) 代理人	100078868 弁理士 河野 登夫
(43) 公開日	平成24年8月16日 (2012.8.16)	(74) 代理人	100114557 弁理士 河野 英仁
審査請求日	平成25年10月7日 (2013.10.7)	(72) 発明者	上村 学 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72) 発明者	前田 芳晴 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

最終頁に続く

(54) 【発明の名称】 コンピュータプログラム、プログラム解析方法及びプログラム解析装置

(57) 【特許請求の範囲】

【請求項1】

対象プログラムを解析する処理をコンピュータに実行させるコンピュータプログラムにおいて、

第1入力データ及び該第1入力データと異なる第2入力データに夫々基づく前記対象プログラムの第1駆動履歴及び第2駆動履歴を記録する記録手段に記録された該第1駆動履歴から、該対象プログラムに含まれる関数の呼び出し情報に基づいて、該関数の第1呼び出し履歴を抽出し、

前記記録手段に記録された第2駆動履歴から、前記関数の呼び出し情報に基づいて、該関数の第2呼び出し履歴を抽出し、

前記関数の第1及び第2呼び出し履歴に夫々基づいて、前記第1及び第2駆動履歴各々における該関数の呼び出し回数又は呼び出し箇所を取得する

処理をコンピュータに実行させるコンピュータプログラム。

【請求項2】

前記関数の呼び出し情報は該関数の呼び出し開始又は呼び出し終了を示す情報を含み、前記関数の第1及び第2呼び出し履歴を抽出する処理は、該関数の呼び出し開始又は呼び出し終了に係る第1及び第2呼び出し履歴を夫々抽出し、

前記関数の呼び出し回数又は呼び出し箇所を取得する処理は、該関数の呼び出し開始又は呼び出し終了に係る第1及び第2呼び出し履歴に夫々基づいて、前記第1及び第2駆動履歴各々における該関数の呼び出し回数又は呼び出し箇所を取得する

請求項 1 に記載のコンピュータプログラム。

【請求項 3】

前記対象プログラムの第 1 及び第 2 駆動履歴は行単位の文字列であり、

前記関数の呼び出し箇所を取得する処理は、該関数の呼び出し開始又は呼び出し終了に対応する行番号を取得する

請求項 2 に記載のコンピュータプログラム。

【請求項 4】

前記関数の呼び出し開始又は呼び出し終了に対応する行番号に基づいて、該関数の処理に関するステップ数に応じた行の数量を算出する

請求項 3 に記載のコンピュータプログラム。

10

【請求項 5】

前記ステップ数に応じた行の数量を算出する処理は、前記対象プログラムの第 1 及び第 2 駆動履歴各々における関数の同一呼び出し回数目の呼び出し開始及び呼び出し終了に夫々対応する行番号に基づいて、前記ステップ数に応じた行の数量を算出する

請求項 4 に記載のコンピュータプログラム。

【請求項 6】

前記ステップ数に応じた行の数量を算出する処理は、前記対象プログラムの第 1 及び第 2 駆動履歴各々における関数の呼び出し開始に対応する最初の行番号及び呼び出し開始に対応する最後の行番号、又は該関数の呼び出し終了に対応する最初の行番号及び呼び出し終了に対応する最後の行番号に基づいて、前記ステップ数に応じた行の数量を算出する

請求項 4 に記載のコンピュータプログラム。

20

【請求項 7】

対象プログラムを解析するプログラム解析方法において、

コンピュータが、

第 1 入力データ及び該第 1 入力データと異なる第 2 入力データに夫々基づく前記対象プログラムの第 1 駆動履歴及び第 2 駆動履歴を記録する記録手段に記録された該第 1 駆動履歴から、該対象プログラムに含まれる関数の呼び出し情報に基づいて、該関数の第 1 呼び出し履歴を抽出し、

前記記録手段に記録された第 2 駆動履歴から、前記関数の呼び出し情報に基づいて、該関数の第 2 呼び出し履歴を抽出し、

前記関数の第 1 及び第 2 呼び出し履歴に夫々基づいて、前記第 1 及び第 2 駆動履歴各々における該関数の呼び出し回数又は呼び出し箇所を取得する

プログラム解析方法。

30

【請求項 8】

対象プログラムを解析するプログラム解析装置において、

第 1 入力データ及び該第 1 入力データと異なる第 2 入力データに夫々基づく前記対象プログラムの第 1 駆動履歴及び第 2 駆動履歴を記録する記録手段と、

該記録手段に記録された第 1 駆動履歴から、前記対象プログラムに含まれる関数の呼び出し情報に基づいて、該関数の第 1 呼び出し履歴を抽出する第 1 抽出部と、

前記記録手段に記録された第 2 駆動履歴から、前記関数の呼び出し情報に基づいて、該関数の第 2 呼び出し履歴を抽出する第 2 抽出部と、

前記第 1 及び第 2 抽出部が夫々抽出した関数の前記第 1 及び第 2 呼び出し履歴に基づいて、前記第 1 及び第 2 駆動履歴各々における該関数の呼び出し回数又は呼び出し箇所を取得する取得部と

を備えるプログラム解析装置。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、プログラムの動作を解析するコンピュータプログラム、プログラム解析方法及びプログラム解析装置に関する。

50

## 【背景技術】

## 【0002】

開発されたプログラムは、その運用及び保守に係る歳月の経過にともない、改良、機能追加等の修正が必要になる。しかし、修正対象のソースコード、設計仕様書、要求仕様書等が手に入らない場合、プログラムの動作がわからないため、プログラムの運用及び保守は困難を極める。そこで、プログラムの動作を解析するリバースエンジニアリングが行われる（例えば、特許文献1～5参照）。

## 【先行技術文献】

## 【特許文献】

## 【0003】

【特許文献1】特開平06-139224号公報

【特許文献2】特開2005-124112号公報

【特許文献3】特許第4288978号公報

【特許文献4】特開2004-302751号公報

【特許文献5】特表2001-527235号公報

## 【発明の概要】

## 【発明が解決しようとする課題】

## 【0004】

しかしながら、多くの場合、リバースエンジニアリングによりプログラムの分析を行うときは、ソースコードが必要とされる。リバースエンジニアリングはオブジェクトコードをソースコードに戻す逆コンパイルを含む。しかし、逆コンパイルによってコンパイル前のソースコードを復元することは難しい。また、逆コンパイルは著作権を侵害するおそれがある。

## 【0005】

本発明は斯かる事情に鑑みてなされたものである。その目的は、入力データの変化に基づいてプログラムに含まれる関数の動作変化を取得することが可能なコンピュータプログラム、プログラム解析方法及びプログラム解析装置を提供することである。

## 【課題を解決するための手段】

## 【0006】

本願による開示の一観点は、対象プログラムを解析する処理をコンピュータに実行させるコンピュータプログラムにおいて、第1入力データ及び該第1入力データと異なる第2入力データに夫々基づく前記対象プログラムの第1駆動履歴及び第2駆動履歴を記録する記録手段に記録された該第1駆動履歴から、該対象プログラムに含まれる関数の呼び出し情報に基づいて、該関数の第1呼び出し履歴を抽出し、前記記録手段に記録された第2駆動履歴から、前記関数の呼び出し情報に基づいて、該関数の第2呼び出し履歴を抽出し、前記関数の第1及び第2呼び出し履歴に夫々基づいて、前記第1及び第2駆動履歴各々における該関数の呼び出し回数又は呼び出し箇所を取得する処理をコンピュータに実行させる。

## 【発明の効果】

## 【0007】

本願による開示の一観点によれば、入力データの変化に基づいて、プログラムに含まれる関数の動作変化を取得することが可能である。

## 【図面の簡単な説明】

## 【0008】

【図1】プログラム解析装置のハードウェア構成例を示すブロック図である。

【図2】データ処理の流れの一例を示すデータフロー図である。

【図3】プログラム解析処理の手順の一例を示すフローチャートである。

【図4】関数の呼び出し履歴を計測する処理の手順の一例を示すフローチャートである。

【図5】関数の呼び出し履歴を計測する処理の手順の一例を示すフローチャートである。

【図6】関数の呼び出し履歴を分析する処理の手順の一例を示すフローチャートである。

10

20

30

40

50

【図 7】関数の呼び出し履歴を分析する処理の手順の一例を示すフローチャートである。

【図 8】関数の呼び出し履歴を分析する処理の手順の一例を示すフローチャートである。

【図 9】基準データである商品在庫テーブルのレコード例を示す説明図である。

【図 10】変更データである商品在庫テーブルのレコード例を示す説明図である。

【図 11】差分データの一例を示す説明図である。

【図 12】基準データ実行履歴ファイルの内容の一例を示す説明図である。

【図 13】変更データ実行履歴ファイルの内容の一例を示す説明図である。

【図 14】基準データ関数別計測ファイルの内容の一例を示す説明図である。

【図 15】変更データ関数別計測ファイルの内容の一例を示す説明図である。

【図 16】関数別呼び出し履歴分析ファイルの内容の一例を示す説明図である。

10

【図 17】差分リストファイルの内容の一例を示す説明図である。

【図 18】基準データに基づく関数の分析情報と変更データに基づく関数の分析情報との差分を示す説明図である。

【図 19】解析対象プログラムの実行履歴における関数間の構造の一例を示す説明図である。

【発明を実施するための形態】

【0009】

以下、実施の形態をその図面に基づいて説明する。本願に開示するプログラム解析装置を P C (パーソナルコンピュータ) に適用した実施の形態について説明する。

なお、本発明は、以下の実施の形態に限定されるものではない。

20

【0010】

図 1 は、プログラム解析装置 1 のハードウェア構成例を示すブロック図である。プログラム解析装置 1 は、C P U (Central Processing Unit) 1 1、R O M (Read Only Memory) 1 2、R A M (Random Access Memory) 1 3、ハードディスク 1 4、ディスクドライブ 1 5、通信部 1 6、表示部 1 7 及び操作部 1 8 を含む。プログラム解析装置 1 のハードウェア各部は、バス 1 b を介して相互に接続されている。

【0011】

C P U 1 1 は、プログラム解析装置 1 のハードウェア各部を制御する。C P U 1 1 は、ハードディスク 1 4 に記録されたプログラム 1 P を R A M 1 3 に読み込み、プログラム 1 P を実行する。

30

【0012】

R O M 1 2 は、プログラム解析装置 1 の起動時に C P U 1 1 が実行する B I O S (Basic Input/Output System) 等を記録している。

R A M 1 3 は、例えば S R A M (Static RAM)、D R A M (Dynamic RAM) であり、C P U 1 1 が実行する処理の過程で必要な作業変数、データ等を一時的に記録する。なお、R A M 1 3 は主記憶装置の一例であり、R A M 1 3 の代わりにフラッシュメモリ、メモリカード等が用いられてもよい。

【0013】

ハードディスク 1 4 は、C P U 1 1 が実行するプログラム 1 P を記録している。

なお、ハードディスク 1 4 は、プログラム解析装置 1 の内部に取り付けられるものであっても、プログラム解析装置 1 の外部に置かれるものであってもよい。また、ハードディスク 1 4 は補助記憶装置の一例であり、大容量の情報の記録が可能なフラッシュメモリ、C D (Compact Disc)、D V D (Digital Versatile Disk)、B D (Blu-ray Disc、登録商標) 等の光ディスク 1 a で代替してもよい。

40

【0014】

ディスクドライブ 1 5 は、外部の記録媒体である C D、D V D、B D 等の光ディスク 1 a から情報を読み込み、光ディスク 1 a に情報を記録する。C P U 1 1 は、操作部 1 8 からディスクドライブ 1 5 に対する光ディスク 1 a の排出命令を受け付けた場合、ディスクドライブ 1 5 の図示しないトレイを排出する。

【0015】

50

通信部 16 は、有線又は無線通信のモデム又は LAN カード等であり、インターネット N に接続されている。通信部 16 は、電話回線と接続することができてよい。

【0016】

表示部 17 は、例えば液晶ディスプレイ、有機 EL (Electro-Luminescence) ディスプレイ、CRT (Cathode Ray Tube) ディスプレイ等の画面を有し、CPU 11 からの指示に従って、プログラム 1P に係る各種情報を表示する。

【0017】

操作部 18 は、ユーザが各種の入力を行うキーボード、マウス、電源スイッチ、ディスクドライブ 15 から光ディスク 1a を排出するための操作ボタン等の入力デバイスを含む。操作部 18 は、ユーザによる操作に基づいて入力信号を生成する。生成された入力信号は、バス 1b を介して CPU 11 に送信される。

10

【0018】

次に、プログラム解析装置 1 の動作の概略について説明する。

図 2 は、データ処理の流れの一例を示すデータフロー図である。対象プログラム実行部 21、関数呼び出し履歴計側部 22 及び関数呼び出し履歴分析部 23 は、CPU 11 がハードディスク 14 に記録されたプログラム 1P に基づいて動作することにより実行される各機能部である。

【0019】

対象プログラム実行部 21 は、入力データを受け付ける。プログラム解析装置 1 が受け付けるデータには、この入力データの他に当該入力データの値又はデータ量を変更した変更データ (第 2 入力データ) CD がある。そこで以後、変更データ CD の元となる入力データを変更データ CD と区別するために基準データ SD (第 1 入力データ) と呼ぶ。対象プログラム実行部 21 は、受け付けた基準データ SD に基づいて、解析対象プログラムを当該プログラムの機能単位毎に実行する。対象プログラム実行部 21 は、基準データ SD に基づいて実行した解析対象プログラムの実行履歴を取得する。対象プログラム実行部 21 は、取得した解析対象プログラムの実行履歴を基準データ実行履歴ファイル (第 1 駆動履歴) 21F としてハードディスク 14 に記録する。

20

【0020】

対象プログラム実行部 21 は、基準データ SD の値又はデータ量を変更した変更データ CD を受け付ける。対象プログラム実行部 21 は、変更データ CD に対しても基準データ SD と同様に解析対象プログラムを機能単位毎に実行する。対象プログラム実行部 21 は、変更データ CD に基づいて解析対象プログラムを実行し、その実行履歴を取得する。対象プログラム実行部 21 は、取得した実行履歴を変更データ実行履歴ファイル (第 2 駆動履歴) 210F としてハードディスク 14 に記録する。

30

【0021】

基準データ実行履歴ファイル 21F 及び変更データ実行履歴ファイル 210F には、ともに解析対象プログラムの実行履歴が行単位の文字列として記述されている。基準データ実行履歴ファイル 21F 及び変更データ実行履歴ファイル 210F には、呼び出された関数に関する情報が記述された行と、呼び出された関数に関する情報が記述されていない行とを含む。なお、以下では関数の呼び出しと関数の実行は同じ意味である。呼び出された関数に関する情報が記述された行には、少なくとも呼び出された関数名と、関数の呼び出し開始を示す情報又は関数の呼び出し終了を示す情報とが記述されている。関数の呼び出し開始を示す情報又は関数の呼び出し終了を示す情報の有無により、呼び出された関数に関する情報が記述された行と、呼び出された関数に関する情報が記述されていない行とを判別することができる。なお、関数の呼び出し開始及び呼び出し終了を示す情報は、呼び出し開始と呼び出し終了とを判別することができるのであれば、記号、数式、語句、文等のいずれであってもよい。

40

【0022】

関数呼び出し履歴計側部 22 は、ハードディスク 14 から基準データ実行履歴ファイル 21F を先頭行から読み込み、関数の呼び出し回数、関数の呼び出し開始行、関数の呼び

50

出し終了行を行番号に基づいて計測する。関数呼び出し履歴計測部 2 2 は、計測結果を基準データ関数別計測ファイル（第 1 呼び出し履歴）2 2 F としてハードディスク 1 4 に記録する。

【 0 0 2 3 】

関数呼び出し履歴計測部 2 2 は、変更データ実行履歴ファイル 2 1 0 F についても同様に関数の呼び出し回数、関数の呼び出し開始行、関数の呼び出し終了行を行番号に基づいて計測し、その計測結果を変更データ関数別計測ファイル（第 2 呼び出し履歴）2 2 0 F としてハードディスク 1 4 に記録する。

【 0 0 2 4 】

なお、基準データ実行履歴ファイル 2 1 F 及び変更データ実行履歴ファイル 2 1 0 F には、行番号が記述されていてもよいし、記述されていなくてもよい。基準データ実行履歴ファイル 2 1 F 及び変更データ実行履歴ファイル 2 1 0 F に行番号が記述されている場合、関数呼び出し履歴計測部 2 2 は、当該行番号を利用して上述の計測を行う。一方、基準データ実行履歴ファイル 2 1 F 及び変更データ実行履歴ファイル 2 1 0 F に行番号が記述されていない場合、関数呼び出し履歴計測部 2 2 は、内部的に行番号をカウントしながら上述の計測を行う。

【 0 0 2 5 】

関数呼び出し履歴分析部 2 3 は、ハードディスク 1 4 から基準データ関数別計測ファイル 2 2 F 及び変更データ関数別計測ファイル 2 2 0 F を読み込む。関数呼び出し履歴分析部 2 3 は、基準データ関数別計測ファイル 2 2 F に記録された関数の呼び出し開始行及び呼び出し終了行の行番号に基づいて、関数毎に各種の行間距離を算出する。ここでの行間距離は、行番号の差である。なお、行間距離は、行番号の差に所定数を加算又は減算したものでもよいし、行番号の差に所定数を掛けたものでもよい。

関数呼び出し履歴分析部 2 3 は、変更データ関数別計測ファイル 2 2 0 F に記録された関数についても、呼び出し開始行及び呼び出し終了行の行番号に基づいて、関数毎に各種の行間距離を算出する。

【 0 0 2 6 】

関数呼び出し履歴分析部 2 3 は、基準データ S D 及び変更データ C D に夫々対応する関数の呼び出し回数及び上記の行間距離が一覧形式に整理された情報を、関数別呼び出し履歴分析ファイル 2 3 F としてハードディスク 1 4 に記録する。関数呼び出し履歴分析部 2 3 は、基準データ S D に対応する関数の分析情報と変更データ C D に対応する関数の分析情報とを比較し、この比較結果に基づいて各関数を数種類に分類する。関数呼び出し履歴分析部 2 3 は、基準データ S D と変更データ C D との差分である差分データ D D を受け付ける。関数呼び出し履歴分析部 2 3 は、分類した関数を受け付けた差分データ D D と対応付けた態様で差分リストファイル 2 3 0 F に出力する。

【 0 0 2 7 】

図 2 では、基準データ S D の受け付けから基準データ関数別計測ファイル 2 2 F を出力するまでの処理フローと、変更データ C D の受け付けから変更データ関数別計測ファイル 2 2 0 F を出力するまでの処理フローとが並列に示されている。しかし、これらの処理フローは直列に実行されてもよい。

【 0 0 2 8 】

図 2 の対象プログラム実行部 2 1 が実行する処理は、外部の情報処理装置が実行してもよい。かかる場合、プログラム解析装置 1 は、外部の情報処理装置が生成した基準データ実行履歴ファイル 2 1 F 及び変更データ実行履歴ファイル 2 1 0 F をディスクドライブ 1 5 又は通信部 1 6 を介して受け付ける。あるいは、プログラム解析装置 1 は、フラッシュメモリ等の半導体メモリ 1 c を介して、基準データ実行履歴ファイル 2 1 F 及び変更データ実行履歴ファイル 2 1 0 F を受け付けてもよい。関数呼び出し履歴計測部 2 2 は、外部から受け付けた基準データ実行履歴ファイル 2 1 F 及び変更データ実行履歴ファイル 2 1 0 F を用いて、関数の呼び出し履歴を計測する。

【 0 0 2 9 】

10

20

30

40

50

図3は、プログラム解析処理の手順の一例を示すフローチャートである。図3は、プログラム解析処理の全体的な概要を示している。

CPU11は、基準データSD及び変更データCDを受け付ける(ステップS101)。CPU11は、受け付けた基準データSD及び変更データCDに基づいて、解析対象プログラムを実行する(ステップS102)。CPU11は、基準データSD及び変更データCDの全データに基づいて、解析対象プログラムを実行したか否か判定する(ステップS103)。CPU11は、基準データSD及び変更データCDの全データに基づいて、解析対象プログラムを実行していないと判定した場合(ステップS103:NO)、ステップS102に処理を戻す。CPU11は、基準データSD及び変更データCDの全データに基づいて、解析対象プログラムを実行したと判定した場合(ステップS103:YES)、基準データ実行履歴ファイル21F及び変更データ実行履歴ファイル210Fを作成する(ステップS104)。

#### 【0030】

CPU11は、基準データ実行履歴ファイル21F及び変更データ実行履歴ファイル210Fに基づいて、解析対象プログラムに含まれる関数の呼び出し回数、関数の呼び出し開始行及び関数の呼び出し終了行を計測する(ステップS105)。CPU11は、基準データ実行履歴ファイル21F及び変更データ実行履歴ファイル210Fの全データに基づいて、解析対象プログラムに含まれる関数の呼び出し回数等、ステップS105で計測する項目を全て計測したか否か判定する(ステップS106)。CPU11は、基準データ実行履歴ファイル21F及び変更データ実行履歴ファイル210Fの全データに基づいて、当該関数の呼び出し回数等、ステップS105で計測する項目を一つでも計測していないと判定した場合(ステップS106:NO)、ステップS105に処理を戻す。CPU11は、基準データ実行履歴ファイル21F及び変更データ実行履歴ファイル210Fの全データに基づいて、当該関数の呼び出し回数等を計測したと判定した場合(ステップS106:YES)、ステップS107に処理を進める。CPU11は、計測結果に基づいて基準データ関数別計測ファイル22F及び変更データ関数別計測ファイル220Fを作成する(ステップS107)。

#### 【0031】

CPU11は、基準データ関数別計測ファイル22F及び変更データ関数別計測ファイル220Fに基づいて、関数毎に各種の行間距離を算出し、算出した行間距離が関数の呼び出し回数と対応付けられた関数別呼び出し履歴分析ファイル23Fを作成する(ステップS108)。CPU11は、差分データDDを受け付ける(ステップS109)。CPU11は、基準データSD及び変更データCDに夫々対応する関数の分析情報を比較することにより、関数を分類し、分類結果が差分データDDと対応付けられた差分リストファイル230Fを作成し(ステップS110)、処理を終える。

#### 【0032】

上述のステップS101からステップS104までの処理は、対象プログラム実行部21が行う処理機能に該当する。ステップS105からステップS107までの処理は、関数呼び出し履歴計測部22が行う処理機能に該当する。ステップS108からステップS110までの処理は、関数呼び出し履歴分析部23が行う処理機能に該当する。

#### 【0033】

次に、プログラム解析装置1の動作について詳細に説明する。

本実施の形態では、対象プログラム実行部21が解析対象プログラムを実行する前に、入力データである基準データSDと変更データCDとを用意する必要がある。基準データSD及び変更データCDは、データベースのテーブルデータ、画面からの入力データ、ファイルのデータ等である。基準データSDは、既存の入力データでもよいし、手作業又は入力データ作成ソフト等により新規に作成されてもよい。

#### 【0034】

変更データCDは、基準データSDの値の一部若しくは全部を変更したデータ、又は基準データSDのデータ量を変更したデータである。基準データSDのデータ量の変更は、

10

20

30

40

50

データの削除又はデータの追加により行われる。データの追加は、基準データSDの複製又は基準データSDから変更されたデータの追加により行われる。

【0035】

基準データSDを変更する場合、基準データSDのデータ特性が考慮される。例えば、基準データSDがデータベースのテーブルレコードである場合、変更データCDは当該テーブルに関するテーブル制約及び列制約に整合するように基準データSDを変更することにより、作成される。

【0036】

基準データSD及び変更データCDが用意された段階で、基準データSDと変更データCDとの差分データDDが作成される。例えば、基準データSD及び変更データCDがデータベースのテーブルデータである場合、テーブル名、差異が生じている列名、差異が生じているレコードの主キーデータ値等が差分に該当する。あるいは、基準データSDへのデータ削除又はデータ追加により変更データCDが作成された場合、削除されたレコードセット又は追加されたレコードセットの記録を差分データDDとしてもよい。

なお、基準データSDに対して最初に差分データDDを作成し、その後に基準データSDと差分データDDとから変更データCDを作成してもよい。

【0037】

プログラム解析装置1のハードディスク14に解析対象プログラムをインストールする。この解析対象プログラムは、機械語で記述されている。

また、用意した基準データSD及び変更データCDをプログラム解析装置1のハードディスク14に記録する。

【0038】

CPU11は、基準データSD及び変更データCDを解析対象プログラムに入力する。CPU11は解析対象プログラムを機能単位で実行し、基準データSD及び変更データCDの夫々について解析対象プログラムの実行履歴を取得する。この実行履歴を取得するために、既存の技術が利用される。例えば、解析対象プログラムがJAV A(登録商標)アプリケーションである場合、スタックの中身を実行履歴とし、仮想マシンにスタックトレースを出力させる。

CPU11は、基準データSD及び変更データCDに基づいて実行された解析対象プログラムの実行履歴を夫々基準データ実行履歴ファイル21F及び変更データ実行履歴ファイル210Fとしてハードディスク14に記録する。

【0039】

図4及び図5は、関数の呼び出し履歴を計測する処理の手順の一例を示すフローチャートである。以下は、基準データSDに係る基準データ実行履歴ファイル21Fに基づく処理であるが、変更データCDに係る変更データ実行履歴ファイル210Fに基づく処理についても、CPU11は同様の処理を実行する。

以下のステップS202からステップS212までの処理は、基準データ実行履歴ファイル21Fの1行についての処理であり、基準データ実行履歴ファイル21Fの行数だけ繰り返し実行される。

【0040】

CPU11は、基準データ実行履歴ファイル21Fをハードディスク14からRAM13にロードする(ステップS201)。CPU11は、基準データ実行履歴ファイル21Fから次の1行を読み込む(ステップS202)。ステップS202において、CPU11は基準データ実行履歴ファイル21Fを開いた直後の場合、先頭の1行を読み込む。CPU11は、読み込んだ行から行番号と、関数名と、当該関数の呼び出し開始又は呼び出し終了の情報とを取得する(ステップS203)。なお、行番号は、基準データ実行履歴ファイル21F又は変更データ実行履歴ファイル210Fの作成時に記述されてもよいし、CPU11がこれらのファイルから1行ずつデータを読み込む際に、カウントしてもよい。

【0041】



CPU11は、ステップS203において関数名を取得したか否か判定する(ステップS204)。CPU11は、ステップS203において関数名を取得しなかったと判定した場合(ステップS204:NO)、ステップS202に処理を戻す。CPU11は、ステップS203において関数名を取得したと判定した場合(ステップS204:YES)、配列を読み込む(ステップS205)。この配列は基準データ実行履歴ファイル21Fから読み込んだ関数名を格納する作業用配列である。

#### 【0042】

CPU11は、配列の中にステップS203で取得した関数名があるか否か判定する(ステップS206)。CPU11は、配列の中にステップS203で取得した関数名がないと判定した場合(ステップS206:NO)、配列にステップS203で取得した関数名を格納し(ステップS207)、ステップS208へ処理を移す。CPU11は、配列の中にステップS203で取得した関数名があると判定した場合(ステップS206:YES)、ステップS203で取得した情報が関数の呼び出し開始情報か否か判定する(ステップS208)。

10

#### 【0043】

CPU11は、ステップS203で取得した情報が関数の呼び出し開始情報であると判定した場合(ステップS208:YES)、当該関数の呼び出し回数をインクリメントする(ステップS209)。CPU11は、ステップS203で取得した行番号を当該関数の呼び出し開始行番号として記録し(ステップS210)、ステップS212へ処理を移す。CPU11は、ステップS203で取得した情報が関数の呼び出し開始情報でないと判定した場合(ステップS208:NO)、ステップS203で取得した行番号を当該関数の呼び出し終了行番号として記録し(ステップS211)、ステップS212へ処理を移す。CPU11は、基準データ実行履歴ファイル21Fの全行を処理したか否か判定する(ステップS212)。CPU11は、基準データ実行履歴ファイル21Fの全行を処理していないと判定した場合(ステップS212:NO)、ステップS202に処理を戻す。CPU11は、基準データ実行履歴ファイル21Fの全行を処理したと判定した場合(ステップS212:YES)、処理を終了する。

20

#### 【0044】

図4及び図5の処理により、基準データSD及び変更データCDに基づいて夫々呼び出された関数毎に呼び出し回数、呼び出し開始行番号及び呼び出し終了行番号がRAM13に記録される。

30

呼び出し回数が複数回である関数の場合、複数の呼び出し開始行番号及び複数の呼び出し終了行番号がRAM13に記録される。かかる場合、呼び出し開始行番号の数と呼び出し終了行番号の数とは同数である。

#### 【0045】

CPU11は、上述において計測した関数の呼び出し履歴を整理し、基準データ関数別計測ファイル22F及び変更データ関数別計測ファイル220Fを作成する。

例えば、基準データ関数別計測ファイル22Fを作成するために、CPU11は1つの関数について、関数名及び呼び出し回数をRAM13に記録した関数の呼び出し履歴から抽出する。

40

CPU11は、当該関数の呼び出し開始が最初に取得された行番号をRAM13に記録した関数の呼び出し履歴から抽出する。以後、この行番号を初出開始行と呼ぶ。

CPU11は、当該関数の呼び出し終了が最初に取得された行番号をRAM13に記録した関数の呼び出し履歴から抽出する。以後、この行番号を初出終了行と呼ぶ。

CPU11は、当該関数の呼び出し開始が最後に取得された行番号をRAM13に記録した関数の呼び出し履歴から抽出する。以後、この行番号を最終開始行と呼ぶ。

CPU11は、当該関数の呼び出し終了が最後に取得された行番号をRAM13に記録した関数の呼び出し履歴から抽出する。以後、この行番号を最終終了行と呼ぶ。

#### 【0046】

CPU11は、1つの関数について、関数名、関数の呼び出し回数、初出開始行、初出

50

終了行、最終開始行及び最終終了行を1レコードとする基準データ関数別計測ファイル22Fを作成する。基準データ関数別計測ファイル22Fは、CPU11が取得した関数の数だけこれらのレコードを含む。

【0047】

同様に、CPU11は、変更データ実行履歴ファイル210Fに基づいて計測した関数の呼び出し履歴から、変更データ関数別計測ファイル220Fを作成する。変更データ関数別計測ファイル220Fのファイル形式は、基準データ関数別計測ファイル22Fと同一である。すなわち、変更データ関数別計測ファイル220Fは、1つの関数について、関数名、関数の呼び出し回数、初出開始行、初出終了行、最終開始行及び最終終了行を含む。そして、変更データ関数別計測ファイル220Fは、CPU11が取得した関数の数だけこれらのレコードを含む。

10

CPU11は、作成した基準データ関数別計測ファイル22F及び変更データ関数別計測ファイル220Fをハードディスク14に記録する。

【0048】

上述では、関数の呼び出し回数が複数回である関数の場合、複数の呼び出し開始行番号及び複数の呼び出し終了行番号がRAM13に記録された。しかし、RAM13に記録する呼び出し開始行番号及び呼び出し終了行番号は、夫々2つずつでもよい。すなわち、CPU11は、初出開始行、初出終了行、最終開始行及び最終終了行を記録する領域を夫々1つずつRAM13に確保する。CPU11は、関数の呼び出し開始の行を読み込んだ場合、初出開始行及び最終開始行に、読み込んだ行の行番号を記録する。CPU11は、次に再び関数の呼び出し開始の行を読み込んだ場合、初出開始行はそのままにし、最終開始行を新たに読み込んだ行の行番号に更新する。CPU11は、その後さらに関数の呼び出し開始の行を読み込んだ場合、次々と最終開始行だけを更新する。

20

CPU11は、関数の呼び出し終了の行を読み込んだ場合、初出終了行及び最終終了行に、読み込んだ行の行番号を記録する。CPU11は、次に再び関数の呼び出し終了の行を読み込んだ場合、初出終了行はそのままにし、最終終了行を新たに読み込んだ行の行番号に更新する。CPU11は、その後さらに関数の呼び出し終了の行を読み込んだ場合、次々と最終終了行だけを更新する。

【0049】

図6、図7及び図8は、関数の呼び出し履歴を分析する処理の手順の一例を示すフローチャートである。

30

CPU11は、基準データ関数別計測ファイル22Fをハードディスク14からRAM13にロードする(ステップS301)。CPU11は、基準データ関数別計測ファイル22Fから次の1行を読み込む(ステップS302)。ステップS302において、CPU11は基準データ関数別計測ファイル22Fを開いた直後の場合、先頭の1行を読み込む。CPU11は、関数名及び関数の呼び出し回数を取得する(ステップS303)。

【0050】

CPU11は、初出開始行及び初出終了行を取得する(ステップS304)。CPU11は、初出開始行及び初出終了行の差を算出する(ステップS305)。以後、初出開始行及び初出終了行の差を関数行間距離と呼ぶ。

40

【0051】

CPU11は、初出開始行及び最終開始行を取得する(ステップS306)。CPU11は、初出開始行及び最終開始行の差を算出する(ステップS307)。以後、初出開始行及び最終開始行の差を開始行間距離と呼ぶ。

【0052】

CPU11は、基準データ関数別計測ファイル22Fの全行、すなわち全関数についてステップS302からステップS307までの処理を実行したか否か判定する(ステップS308)。CPU11は、基準データ関数別計測ファイル22Fの全関数についてステップS302からステップS307までの処理を実行しなかったと判定した場合(ステップS308:NO)、ステップS302に処理を戻す。CPU11は、基準データ関数別

50

計測ファイル 2 2 F の全関数についてステップ S 3 0 2 からステップ S 3 0 7 までの処理を実行したと判定した場合（ステップ S 3 0 8 : Y E S ）、ステップ S 3 0 9 に処理を進める。

【 0 0 5 3 】

C P U 1 1 は、変更データ関数別計測ファイル 2 2 0 F の全関数についてステップ S 3 0 2 からステップ S 3 0 8 の処理を実行したか否か判定する（ステップ S 3 0 9 ）。C P U 1 1 は、変更データ関数別計測ファイル 2 2 0 F の全関数についてステップ S 3 0 2 からステップ S 3 0 8 の処理を実行していないと判定した場合（ステップ S 3 0 9 : N O ）、ステップ S 3 0 1 に処理を戻す。なお、C P U 1 1 は、ステップ S 3 0 1 に戻った後、ステップ S 3 0 1 からステップ S 3 0 8 までの処理について、基準データ関数別計測ファイル 2 2 F を変更データ関数別計測ファイル 2 2 0 F に置き換えて、実行する。

10

C P U 1 1 は、変更データ関数別計測ファイル 2 2 0 F の全関数についてステップ S 3 0 2 からステップ S 3 0 8 の処理を実行したと判定した場合（ステップ S 3 0 9 : Y E S ）、ステップ S 3 1 0 に処理を進める。

【 0 0 5 4 】

ここまでの段階で、基準データ S D 及び変更データ C D に基づいて実行された解析対象プログラムの関数に関する呼び出し履歴の分析情報が、関数別に R A M 1 3 に記録されている。この分析情報は、関数名、呼び出し回数、関数行間距離及び開始行間距離を含む。

なお、関数行間距離は、初出開始行及び初出終了行の差に限らず、最終開始行及び最終終了行の差であってもよい。あるいは、関数行間距離は、関数の呼び出し開始が n 回目（ただし、n は自然数）に取得された行番号と、関数の呼び出し終了が n 回目に取得された行番号との差であってもよい。すなわち、関数行間距離は、関数の呼び出し開始及び呼び出し終了が夫々同一の呼び出し回数目に取得された行番号同士の差であってもよい。

20

また、開始行間距離は、初出開始行及び最終開始行の差に限らず、初出終了行及び最終終了行の差であってもよい。

【 0 0 5 5 】

C P U 1 1 は、基準データ S D に基づく関数の分析情報と、変更データ C D に基づく関数の分析情報とを関数毎に対応付け、その結果を関数別呼び出し履歴分析ファイル 2 3 F としてハードディスク 1 4 に記録する（ステップ S 3 1 0 ）。関数別呼び出し履歴分析ファイル 2 3 F は、関数名、基準データ S D に基づく関数の分析情報及び変更データ C D に基づく関数の分析情報を 1 レコードとし、解析対象プログラムの実行履歴から取得された関数の数だけレコード数を有する。ただし、基準データ S D に基づく関数の分析情報又は変更データ C D に基づく関数の分析情報のうち、一方にのみ含まれる関数がある場合、関数別呼び出し履歴分析ファイル 2 3 F は、存在しない分析情報のデータ部分を空データとする。

30

この段階では、R A M 1 3 に関数別呼び出し履歴分析ファイル 2 3 F の内容が記録されている。

【 0 0 5 6 】

C P U 1 1 は、R A M 1 3 上の関数別呼び出し履歴分析ファイル 2 3 F から次の 1 行を読み込む（ステップ S 3 1 1 ）。ステップ S 3 1 1 において、C P U 1 1 は最初の処理時には先頭の 1 行を読み込む。C P U 1 1 は、基準データ S D に基づく関数の分析情報又は変更データ C D に基づく関数の分析情報のうち、一方の分析情報にのみ存在する関数があるか否か判定する（ステップ S 3 1 2 ）。C P U 1 1 は、基準データ S D に基づく関数の分析情報又は変更データ C D に基づく関数の分析情報のうち、一方の分析情報にのみ存在する関数があると判定した場合（ステップ S 3 1 2 : Y E S ）、当該関数を一方の分析情報にのみ存在する関数として抽出し（ステップ S 3 1 3 ）、ステップ S 3 1 1 に処理を戻す。C P U 1 1 は、基準データ S D に基づく関数の分析情報又は変更データ C D に基づく関数の分析情報のうち、一方の分析情報にのみ存在する関数はないと判定した場合（ステップ S 3 1 2 : N O ）、ステップ S 3 1 4 に処理を進める。

40

【 0 0 5 7 】

50

C P U 1 1 は、双方の分析情報に記録された呼び出し回数が同じか否か判定する（ステップ S 3 1 4）。C P U 1 1 は、双方の分析情報に記録された呼び出し回数と同じでないと判定した場合（ステップ S 3 1 4：N O）、当該関数を呼び出し回数が異なる関数として抽出し（ステップ S 3 1 5）、ステップ S 3 1 6 に処理を進める。C P U 1 1 は、双方の分析情報に記録された呼び出し回数と同じであると判定した場合（ステップ S 3 1 4：Y E S）、双方の分析情報に記録された関数行間距離が同じか否か判定する（ステップ S 3 1 6）。

**【 0 0 5 8 】**

C P U 1 1 は、双方の分析情報に記録された関数行間距離が同じでないと判定した場合（ステップ S 3 1 6：N O）、当該関数を関数行間距離が異なる関数として抽出し（ステップ S 3 1 7）、ステップ S 3 1 8 に処理を進める。C P U 1 1 は、双方の分析情報に記録された関数行間距離が同じであると判定した場合（ステップ S 3 1 6：Y E S）、双方の分析情報に記録された開始行間距離が同じか否か判定する（ステップ S 3 1 8）。C P U 1 1 は、双方の分析情報に記録された開始行間距離が同じでないと判定した場合（ステップ S 3 1 8：N O）、当該関数を開始行間距離が異なる関数として抽出し（ステップ S 3 1 9）、ステップ S 3 2 0 に処理を進める。C P U 1 1 は、双方の分析情報に記録された開始行間距離が同じであると判定した場合（ステップ S 3 1 8：Y E S）、全行について、すなわち全関数についてステップ S 3 1 1 からステップ S 3 1 9 までの処理を実行したか否か判定する（ステップ S 3 2 0）。

**【 0 0 5 9 】**

C P U 1 1 は、全関数についてステップ S 3 1 1 からステップ S 3 1 9 までの処理を実行していないと判定した場合（ステップ S 3 2 0：N O）、ステップ S 3 1 1 に処理を戻す。C P U 1 1 は、全関数についてステップ S 3 1 1 からステップ S 3 1 9 までの処理を実行したと判定した場合（ステップ S 3 2 0：Y E S）、差分データ D D を受け付ける（ステップ S 3 2 1）。C P U 1 1 は、上述で抽出した関数のリストを差分リストファイル 2 3 0 F としてハードディスク 1 4 に記録し（ステップ S 3 2 2）、処理を終える。差分リストファイル 2 3 0 F は、一方の分析情報にのみ存在する関数、呼び出し回数が異なる関数、関数行間距離が異なる関数及び開始行間距離が異なる関数を含み、これらの関数毎に対応する差分データ D D が特定可能な態様で記録されている。

**【 0 0 6 0 】**

差分リストファイル 2 3 0 F から関数の動作を推定することにより、解析対象プログラムの動作を把握することができる。

一方の分析情報にのみ存在する関数は、入力データの変化によって実行されなかった関数が実行されるように変化した関数又は実行された関数が実行されないように変化した関数である。例えば、一方の分析情報にのみ存在する関数は、例外処理、エラー処理等を実行する関数を含む。入力データの値又は量に変化した場合、関数の設計仕様によっては例外処理又はエラー処理の関数が呼び出されることがある。

**【 0 0 6 1 】**

なお、一方の分析情報にのみ存在する関数は、一方の分析情報における呼び出し回数が 1 回以上であり、他方の分析情報における呼び出し回数が 0 回である関数である。従って、一方の分析情報にのみ存在する関数であるか否かは、呼び出し回数に基づいて判定されてもよい。一方の分析情報にのみ存在する関数は、呼び出し回数が異なる関数に含めて分類されてもよい。

**【 0 0 6 2 】**

呼び出し回数が異なる関数は、入力データの変化によって呼び出し回数が増えた関数である。例えば、入力データ 1 件に対して 1 回呼び出される関数は、入力データの量変化に相関して呼び出し回数が増えると考えられる。あるいは、入力データの値が増えることにより、関数の呼び出し回数が増える場合がある。例えば、入力データが日付型であり、関数が 1 日分の処理ごとに 1 回ずつ呼び出される場合、1 週間後の日付を入力したときと 1 カ月後の日付を入力したときとで、関数の呼び出し回数は増えると考えられる

10

20

30

40

50

## 【 0 0 6 3 】

関数行間距離が異なる関数は、入力データの変化によって関数の初出の呼び出し開始行と関数の初出の呼び出し終了行との間の行数が変化した関数である。関数行間距離は、関数が複数回呼び出されたとしても、2回目以降の呼び出し履歴は除外している。関数行間距離は、1回目の呼び出しにおける関数のステップ数変化を反映した情報を記録していると考えられる。

例えば、関数が解析対象プログラムの中で上位の機能を有し、当該関数から別の関数が呼び出されて処理が実行される場合がある。かかる場合、呼び出し先の別の関数の呼び出し回数が増えたとき、呼び出し元の関数の関数行間距離は変化する。従って、関数行間距離が異なる関数は、解析対象プログラムにおいて上位の機能を有する可能性がある。

なお、ここでのステップ数とは、解析対象プログラムの実行履歴（駆動履歴）の各行から取得される関数のステップ数であり、解析対象プログラムのソースコードにおけるステップ数とは異なる場合が含まれる。

## 【 0 0 6 4 】

開始行間距離が異なる関数は、入力データの変化によって関数の初出の呼び出し開始行と関数の最終の呼び出し開始行との間の行数が変化した関数である。開始行間距離は、ステップ数に応じた情報処理量が反映される情報である。入力データの変化によって自身の呼び出し回数が増減した場合、開始行間距離も増減する。

一方、入力データが変化しても自身の呼び出し回数は変わらない関数がある。かかる場合、開始行間距離が異なる関数は、自身から呼び出す別の関数を有し、この別の関数を呼び出す回数を変化させた可能性がある。この場合、開始行間距離が異なる関数は、解析対象プログラムにおいて上位の機能を有する可能性がある。

入力データが変化しても開始行間距離が異なる関数の呼び出し回数が変わらない場合、他の解釈も考え得る。例えば、入力データの変化によって、関数の初出の呼び出し開始行と関数の最終の呼び出し開始行との間に、割り込み処理を実行する関数が呼び出されることが考えられる。あるいは、入力データの変化によって、ステップ数が異なる複数の関数の実行順序が変化した可能性が考えられる。

## 【 0 0 6 5 】

差分リストファイル 2 3 0 F の解釈には多様性があり、一義的に関数の動作が特定されない場合がある。上述は差分リストファイル 2 3 0 F の解釈の一例である。差分リストファイル 2 3 0 F の解釈は、具体的な差分データ D D との対応に基づいて行われる。また、差分リストファイル 2 3 0 F の解釈には、他の情報が利用されてよい。例えば、基準データ実行履歴ファイル 2 1 F 又は変更データ実行履歴ファイル 2 1 0 F が差分リストファイル 2 3 0 F の解釈に利用されてよい。また、解析対象プログラムがオブジェクト指向言語で作成されている場合、クラスの継承、クラスとそれに属するメソッド等の情報が差分リストファイル 2 3 0 F の解釈に利用されてよい。すなわち、言語特性が差分リストファイル 2 3 0 F の解釈に利用されてよい。設計仕様書、要求仕様書等が入手可能な場合、これらのドキュメントが差分リストファイル 2 3 0 F の解釈に利用されてよいことは勿論である。

## 【 0 0 6 6 】

次に、上述したプログラム解析装置 1 の動作を、具体的な例を用いて説明する。

まず、入力データである基準データ S D 及び変更データ C D について説明する。基準データ S D 及び変更データ C D として、データベースのテーブルレコードを扱う。

図 9 は、基準データ S D である商品在庫テーブルのレコード例を示す説明図である。商品在庫テーブルは、商品の在庫を管理するテーブルである。商品在庫テーブルは、商品コード、商品名、登録日時及び在庫数の各列を含む。商品コードは、商品の識別コードを格納し、商品在庫テーブルの主キーである。商品名は、商品の名称を格納する。登録日時は、在庫数が増えたときの日時を格納する。在庫数は、商品の在庫数を格納する。基準データ S D の総数は 2 件である。

## 【 0 0 6 7 】

図 1 0 は、変更データ C D である商品在庫テーブルのレコード例を示す説明図である。変更データ C D は、基準データ S D に対して 4 レコードが追加されたデータであり、その総数は 6 件である。追加データの登録日時及び在庫数は、基準データ S D の先頭レコードの登録日時及び在庫数を複製している。追加データの商品コード及び商品名には、基準データ S D の商品コード及び商品名と重複しない値が格納されている。

## 【 0 0 6 8 】

図 1 1 は、差分データ D D の一例を示す説明図である。差分データ D D は、基準データ S D と変更データ C D との差分であるレコードが格納されたテーブル名と、差分レコードとを含む。差分レコードは、主キーである商品コードの値を列挙することにより示されて

10

いる。本実施例では、基準データ S D 及び変更データ C D の組は 1 組であるため、差分データ D D は 1 種類である。しかし、基準データ S D 及び変更データ C D の組を複数用意した場合、各組ごとに差分データ D D を作成する。

## 【 0 0 6 9 】

プログラム解析装置 1 に基準データ S D を入力し、解析対象プログラムを実行する。プログラム解析装置 1 は、解析対象プログラムの実行履歴を取得し、取得した実行履歴を基準データ実行履歴ファイル 2 1 F としてハードディスク 1 4 に出力する。

同様に、プログラム解析装置 1 に変更データ C D を入力し、解析対象プログラムの実行履歴を変更データ実行履歴ファイル 2 1 0 F としてハードディスク 1 4 に出力させる。

20

## 【 0 0 7 0 】

図 1 2 は、基準データ実行履歴ファイル 2 1 F の内容の一例を示す説明図である。図 1 3 は、変更データ実行履歴ファイル 2 1 0 F の内容の一例を示す説明図である。

図 1 2 及び図 1 3 は、解析対象プログラムの同一機能を実行した場合の実行履歴を示している。各行の左端の数字は行番号である。なお、基準データ実行履歴ファイル 2 1 F 及び変更データ実行履歴ファイル 2 1 0 F は、行番号が記録されていてもよいし、記録されていなくてもよい。

## 【 0 0 7 1 】

基準データ実行履歴ファイル 2 1 F の 1 行目左端の > は、関数の呼び出し開始を示す記号である。その右隣りの `toplist1` は関数名である。従って、> `toplist1` は、関数 `toplist1` の呼び出し開始を示している。基準データ実行履歴ファイル 2 1 F の 2 行目から 5 行目までは、関数内部に記述された命令文乃至制御文等である。6 行目の > `init1` は、関数 `init1` の呼び出し開始を示している。7 行目左端の < は、関数の呼び出し終了を示す記号である。従って、< `init1` は、関数 `init1` の呼び出し終了を示している。

30

## 【 0 0 7 2 】

変更データ実行履歴ファイル 2 1 0 F における記号の意味は、基準データ実行履歴ファイル 2 1 F における記号の意味と同じである。基準データ実行履歴ファイル 2 1 F の先頭から最終行まで読むことにより、関数の呼び出し開始及び呼び出し終了を関数毎に時系列で追跡することができる。同様に変更データ実行履歴ファイル 2 1 0 F の先頭から最終行

40

## 【 0 0 7 3 】

プログラム解析装置 1 は、基準データ実行履歴ファイル 2 1 F 及び変更データ実行履歴ファイル 2 1 0 F を入力データとして、夫々基準データ関数別計測ファイル 2 2 F 及び変更データ関数別計測ファイル 2 2 0 F をハードディスク 1 4 に出力する。

## 【 0 0 7 4 】

図 1 4 は、基準データ関数別計測ファイル 2 2 F の内容の一例を示す説明図である。基準データ関数別計測ファイル 2 2 F は、関数名、初出開始行、初出終了行、最終開始行、最終終了行及び呼び出し回数の各列を含む。関数名は、基準データ実行履歴ファイル 2 1

50

F から抽出された関数名である。初出開始行は、関数の呼び出し開始が最初に読み込まれた行番号である。初出終了行は、関数の呼び出し終了が最初に読み込まれた行番号である。最終開始行は、関数の呼び出し開始が最後に読み込まれた行番号である。最終終了行は、関数の呼び出し終了が最後に読み込まれた行番号である。呼び出し回数は、関数が呼び出された回数である。

**【 0 0 7 5 】**

ここで、CPU 11 が基準データ実行履歴ファイル 2 1 F を読み込み、基準データ関数別計測ファイル 2 2 F を出力するまでの流れについて詳細に説明する。

CPU 11 は、基準データ実行履歴ファイル 2 1 F の 1 行目を読み込み、toplist1 の関数名と、toplist1 の呼び出し開始が 1 行目にあることを取得する。CPU 11 は、作業用の配列を参照し、toplist1 がこの配列に格納されていないことから、初めて登場した関数として toplist1 を配列に格納する。CPU 11 は、> の記号から 1 行目は呼び出し開始情報であると判定し、toplist1 の呼び出し回数を 0 から 1 にインクリメントする。1 行目は toplist1 の呼び出し開始の最初に相当するので、CPU 11 は、toplist1 の初出開始行に 1 を記録する。また、現時点で 1 行目は toplist1 の呼び出し開始の最後に相当するので、CPU 11 は toplist1 の最終開始行に 1 を記録する。

10

**【 0 0 7 6 】**

CPU 11 は、基準データ実行履歴ファイル 2 1 F の 2 行目を読み込む。2 行目は「a = 1」とあり、関数の呼び出し情報ではないので、CPU 11 は 2 行目を無視し、3 行目を読み込む。以降、2 行目と同じ理由から 3 行目、4 行目及び 5 行目を CPU 11 は無視する。

20

**【 0 0 7 7 】**

CPU 11 は、基準データ実行履歴ファイル 2 1 F の 6 行目を読み込む。CPU 11 は、init1 の関数名と、init1 の呼び出し開始が 6 行目にあることを取得する。CPU 11 は、配列を参照し、init1 が配列に格納されていないことから、初めて登場した関数として init1 を配列に格納する。CPU 11 は、init1 の呼び出し回数を 0 から 1 にインクリメントする。CPU 11 は、init1 の初出開始行及び最終開始行に 6 を記録する。

30

**【 0 0 7 8 】**

CPU 11 は、基準データ実行履歴ファイル 2 1 F の 7 行目を読み込む。CPU 11 は、init1 の関数名と、init1 の呼び出し終了が 7 行目にあることを取得する。CPU 11 は、配列を参照し、init1 が配列に既に格納されていることから、配列に init1 を格納しない。また、CPU 11 は、init1 の呼び出し回数をインクリメントしない。7 行目は init1 の呼び出し終了の最初に相当するので、CPU 11 は init1 の初出終了行に 7 を更新する。現時点で 7 行目は init1 の呼び出し終了の最後に相当するので、CPU 11 は init1 の最終終了行に 7 を記録する。

**【 0 0 7 9 】**

CPU 11 は、基準データ実行履歴ファイル 2 1 F の 8 行目を読み込む。CPU 11 は、item1 の関数名と、item1 の呼び出し開始が 8 行目にあることを取得する。CPU 11 は、初めて登場した関数として item1 を配列に格納する。CPU 11 は、item1 の呼び出し回数を 0 から 1 にインクリメントする。CPU 11 は、item1 の初出開始行及び最終開始行に 8 を記録する。

40

**【 0 0 8 0 】**

CPU 11 は、基準データ実行履歴ファイル 2 1 F の 9 行目を読み込む。CPU 11 は、item1 の関数名と、item1 の呼び出し終了が 9 行目にあることを取得する。CPU 11 は、配列を参照し、item1 が配列に既に格納されていることから、配列に item1 を格納しない。また、CPU 11 は、item1 の呼び出し回数をインクリメントしない。9 行目は item1 の呼び出し終了の最初に相当するので、CPU 11 は item1 の初出終了行に 9 を記録する。現時点で 9 行目は item1 の呼び出し終了の最後

50

に相当するので、CPU11はitem1の最終終了行に9を記録する。

【0081】

CPU11は、基準データ実行履歴ファイル21Fの10行目を読み込む。CPU11は、item1の関数名と、item1の呼び出し開始が10行目にあることを取得する。CPU11は、配列を参照し、item1が配列に既に格納されていることから、配列にitem1を格納しない。また、CPU11は、item1の呼び出し回数を1から2にインクリメントする。item1は初出ではないため、CPU11は初出開始行を更新しない。CPU11は、item1の最終開始行を8から10に上書き更新する。

【0082】

CPU11は、基準データ実行履歴ファイル21Fの11行目を読み込む。CPU11は、item1の関数名と、item1の呼び出し終了が11行目にあることを取得する。CPU11は、配列を参照し、item1が配列に既に格納されていることから、配列にitem1を格納しない。また、CPU11は、item1の呼び出し回数をインクリメントしない。11行目はitem1の呼び出し終了の最初ではないので、CPU11はitem1の初出終了行を更新しない。CPU11は、item1の最終終了行を9から11に上書き更新する。

10

【0083】

CPU11は、基準データ実行履歴ファイル21Fの12行目を読み込む。CPU11は、init1の関数名と、init1の呼び出し開始が12行目にあることを取得する。CPU11は、配列を参照し、init1が配列に既に格納されていることから、配列にinit1を格納しない。また、CPU11は、init1の呼び出し回数を1から2にインクリメントする。init1は初出ではないため、CPU11は初出開始行を更新しない。CPU11は、init1の最終開始行を6から12に上書き更新する。

20

【0084】

CPU11は、基準データ実行履歴ファイル21Fの13行目を読み込む。CPU11は、init1の関数名と、init1の呼び出し終了が13行目にあることを取得する。CPU11は、配列を参照し、init1が配列に既に格納されていることから、配列にinit1を格納しない。また、CPU11は、init1の呼び出し回数をインクリメントしない。init1の呼び出し終了は初出ではないため、CPU11は初出終了行を更新しない。CPU11は、init1の最終終了行を7から13に上書き更新する。

30

【0085】

CPU11は、基準データ実行履歴ファイル21Fの14行目を読み込む。CPU11は、toplist1の関数名と、toplist1の呼び出し終了が14行目にあることを取得する。CPU11は、配列を参照し、toplist1が配列に既に格納されていることから、配列にtoplist1を格納しない。また、CPU11は、toplist1の呼び出し回数をインクリメントしない。toplist1の呼び出し終了は初出であるため、CPU11はtoplist1の初出終了行に14を記録する。CPU11は、toplist1の最終終了行に14を記録する。

【0086】

CPU11は、基準データ実行履歴ファイル21FのEOF (End Of File) を取得する。CPU11は、上述の関数の呼び出し履歴について計測した結果を基準データ関数別計測ファイル22Fとしてハードディスク14に記録する。図14の基準データ関数別計測ファイル22Fは、CPU11が基準データ実行履歴ファイル21Fについて計測処理を実行した最終状態を示している。

40

【0087】

図15は、変更データ関数別計測ファイル220Fの内容の一例を示す説明図である。

CPU11は、基準データ実行履歴ファイル21Fについて実行した処理と同じ処理を変更データ実行履歴ファイル210Fについても実行する。図13に示す変更データ実行履歴ファイル210Fの場合、12行目以降の部分に基準データ実行履歴ファイル21Fとの差がある。CPU11は、基準データ実行履歴ファイル21Fよりもitem1が4

50



回多く登場することを変更データ関数別計測ファイル 2 2 0 F に記録する。図 1 5 の変更データ関数別計測ファイル 2 2 0 F は、CPU 1 1 が変更データ実行履歴ファイル 2 1 0 F について計測処理を実行した最終状態を示している。

【 0 0 8 8 】

図 1 6 は、関数別呼び出し履歴分析ファイル 2 3 F の内容の一例を示す説明図である。関数別呼び出し履歴分析ファイル 2 3 F は、基準データ関数別計測ファイル 2 2 F 及び変更データ関数別計測ファイル 2 2 0 F のデータに基づいて作成される。

【 0 0 8 9 】

関数別呼び出し履歴分析ファイル 2 3 F は、関数名、基準データ S D に基づく関数の分析情報及び変更データ C D に基づく関数の分析情報を含む。基準データ S D 及び変更データ C D に基づく関数の分析情報は、ともに呼び出し回数、開始行間距離及び関数行間距離を含む。関数名は、基準データ関数別計測ファイル 2 2 F 又は変更データ関数別計測ファイル 2 2 0 F に含まれる関数名である。呼び出し回数は、基準データ関数別計測ファイル 2 2 F 及び変更データ関数別計測ファイル 2 2 0 F に含まれる関数の呼び出し回数である。開始行間距離は、基準データ関数別計測ファイル 2 2 F 及び変更データ関数別計測ファイル 2 2 0 F に夫々含まれる初出開始行及び最終開始行の差である。関数行間距離は、基準データ関数別計測ファイル 2 2 F 及び変更データ関数別計測ファイル 2 2 0 F に夫々含まれる初出開始行及び初出終了行の差である。

【 0 0 9 0 】

ここで、CPU 1 1 が基準データ関数別計測ファイル 2 2 F を読み込み、関数別呼び出し履歴分析ファイル 2 3 F を出力するまでの流れについて説明する。

CPU 1 1 は、基準データ関数別計測ファイル 2 2 F の 1 行目を読み込む。CPU 1 1 は、関数名 `toplist1` 及び呼び出し回数 1 を取得する。CPU 1 1 は、最終開始行 1 と初出開始行 1 との差から開始行間距離 0 を取得する。CPU 1 1 は、初出終了行 1 4 と初出開始行 1 との差から関数行間距離 1 3 を取得する。

【 0 0 9 1 】

CPU 1 1 は、基準データ関数別計測ファイル 2 2 F の 2 行目及び 3 行目を読み込み、上述と同様に夫々関数名、呼び出し回数、開始行間距離及び関数行間距離を取得する。

CPU 1 1 は、変更データ関数別計測ファイル 2 2 0 F の各行からも同様に、夫々関数名、呼び出し回数、開始行間距離及び関数行間距離を取得する。

【 0 0 9 2 】

CPU 1 1 は、取得した基準データ S D に基づく関数の分析情報及び変更データ C D に基づく関数の分析情報に対して、関数名が一致するもの同士を対応付ける。CPU 1 1 は、関数毎に基準データ S D に基づく関数の分析情報及び変更データ C D に基づく関数の分析情報を 1 レコードとする関数別呼び出し履歴分析ファイル 2 3 F を作成する。図 1 6 の関数別呼び出し履歴分析ファイル 2 3 F の例では、`toplist1`、`item1` 及び `init1` の 3 関数についての 3 レコードが示されている。

【 0 0 9 3 】

ただし、基準データ S D に基づく関数の分析情報又は変更データ C D に基づく関数の分析情報のうち、一方にのみ含まれる関数がある場合、CPU 1 1 は関数の対応付けをすることができない。かかる場合、CPU 1 1 は、存在しない分析情報のデータを空データとする。例えば、関数 `item1` の分析情報が基準データ S D に基づく関数の分析情報に存在し、変更データ C D に基づく関数の分析情報に存在しない場合、CPU 1 1 は、図 1 6 において変更データ C D に基づく `item1` の分析情報の欄を空欄とする。

【 0 0 9 4 】

図 1 7 は、差分リストファイル 2 3 0 F の内容の一例を示す説明図である。差分リストファイル 2 3 0 F は、差分データ D D 及び関数別呼び出し履歴分析ファイル 2 3 F のデータに基づいて作成される。すなわち、差分リストファイル 2 3 0 F は、基準データ S D に基づく関数の分析情報又は変更データ C D に基づく関数の分析情報を比較することにより作成される。

10

20

30

40

50

## 【 0 0 9 5 】

差分リストファイル 2 3 0 F は、差分データ D D、一方の分析情報にのみ存在する関数、呼び出し回数が異なる関数、開始行間距離が異なる関数及び関数行間距離が異なる関数の各リストを含む。差分データ D D は、基準データ S D と変更データ C D との差分に相当するデータである。一方の分析情報にのみ存在する関数は、入力データの変化によって実行されなかった関数が実行されるように変化した関数又は実行された関数が実行されないように変化した関数である。呼び出し回数が異なる関数は、入力データの変化によって呼び出し回数が増えた関数である。開始行間距離が異なる関数は、入力データの変化によって関数の初出の呼び出し開始行と関数の最終の呼び出し開始行との間の行数が増えた関数である。関数行間距離が異なる関数は、入力データの変化によって関数の初出の呼び出し開始行と関数の初出の呼び出し終了行との間の行数が増えた関数である。

10

## 【 0 0 9 6 】

ここで、C P U 1 1 が関数別呼び出し履歴分析ファイル 2 3 F を読み込み、差分リストファイル 2 3 0 F を出力するまでの流れについて説明する。

C P U 1 1 は、関数別呼び出し履歴分析ファイル 2 3 F の 1 行目を読み込む。

C P U 1 1 は、基準データ S D に基づく関数の分析情報又は変更データ C D に基づく関数の分析情報のうち、一方の分析情報にのみ存在する関数があるか否かを判定する。t o p l i s t 1 は双方の関数の分析情報に含まれているため、C P U 1 1 は t o p l i s t 1 の有無に差分はないと判定する。

C P U 1 1 は、双方の分析情報に記録された t o p l i s t 1 の呼び出し回数が増えたり減ったりするかを判定する。呼び出し回数は夫々 1、1 であり同一なので、C P U 1 1 は、t o p l i s t 1 は呼び出し回数が増えたり減ったりする関数ではないと判定する。

20

C P U 1 1 は、双方の分析情報に記録された t o p l i s t 1 の開始行間距離が増えたり減ったりするかを判定する。開始行間距離は夫々 0、0 なので、C P U 1 1 は、t o p l i s t 1 は開始行間距離が増えたり減ったりする関数ではないと判定する。

C P U 1 1 は、双方の分析情報に記録された t o p l i s t 1 の関数行間距離が増えたり減ったりするかを判定する。関数行間距離は、夫々 1 3、2 1 であり異なるので、C P U 1 1 は、t o p l i s t 1 は関数行間距離が増えたり減ったりする関数であると判定する。

## 【 0 0 9 7 】

C P U 1 1 は、関数別呼び出し履歴分析ファイル 2 3 F の 2 行目及び 3 行目を読み込み、上述と同様の処理を実行する。

30

その結果、C P U 1 1 は、i t e m 1 は一方の分析情報にのみ存在する関数ではないと判定する。C P U 1 1 は、i t e m 1 は呼び出し回数が増えたり減ったりする関数であると判定する。C P U 1 1 は、i t e m 1 は開始行間距離が増えたり減ったりする関数であると判定する。C P U 1 1 は、i t e m 1 は関数行間距離が増えたり減ったりする関数でないと判定する。

C P U 1 1 は、i n i t 1 は一方の分析情報にのみ存在する関数ではないと判定する。C P U 1 1 は、i n i t 1 は呼び出し回数が増えたり減ったりする関数でないと判定する。C P U 1 1 は、i n i t 1 は開始行間距離が増えたり減ったりする関数であると判定する。C P U 1 1 は、i n i t 1 は関数行間距離が増えたり減ったりする関数でないと判定する。

## 【 0 0 9 8 】

40

C P U 1 1 は、関数別呼び出し履歴分析ファイル 2 3 F の E O F (End Of File) を取得する。C P U 1 1 は、上述の判定に基づいて、一方の分析情報にのみ存在する関数、呼び出し回数が増えたり減ったりする関数、開始行間距離が増えたり減ったりする関数及び関数行間距離が増えたり減ったりする関数を抽出する。C P U 1 1 は、抽出した関数を差分リストファイル 2 3 0 F としてハードディスク 1 4 に記録する。図 1 7 は、C P U 1 1 により抽出された関数を示している。

図 1 7 の例では、一方の分析情報にのみ存在する関数は存在せず、呼び出し回数が増えたり減ったりする関数は i t e m 1 である。開始行間距離が増えたり減ったりする関数は、i t e m 1 及び i n i t 1、関数行間距離が増えたり減ったりする関数は t o p l i s t 1 である。

なお、一方の分析情報にのみ存在する関数は、一方の分析情報の呼び出し回数が 1 回以上、他方の分析情報の呼び出し回数が 0 回の関数として表示されてもよい。

50

## 【 0 0 9 9 】

一方の分析情報にのみ存在する関数は存在しないので、本実施例において入力データの変化によって例外、エラー処理等を実行する関数は、実行されなかったと考えられる。

`item1`は、呼び出し回数が異なる関数であり、開始行間距離が異なる関数でもある。基準データSDと変更データCDとの違いはデータ数であることから、`item1`は入力データ数の違いにより呼び出し回数が増加する関数であるといえる。そのため、`item1`は、商品在庫の情報を商品の種類ごとに処理する関数である可能性がある。

## 【 0 1 0 0 】

`init1`は、呼び出し回数が異なる関数ではなく、開始行間距離が異なる関数である。`init1`は、解析対象プログラムの中で上位の機能を有し、複数回実行された関数を呼び出した可能性がある。また、基準データ実行履歴ファイル21F又は変更データ実行履歴ファイル210Fを参照した場合、`init1`と`item1`とは入れ子構造であることが予想される。そうすると、`init1`から`item1`が呼び出された可能性が示唆される。

10

## 【 0 1 0 1 】

`toplist1`は、呼び出し回数が異なる関数ではなく、関数行間距離が異なる関数である。`toplist1`は、入力データの量が増加することにより少なくとも1回目の呼び出しにおける実行されたステップ数が増大している。しかし、自身の呼び出し回数に差は生じなかったため、`toplist1`は解析対象プログラムの中で上位の機能を有するものと考えられる。また、基準データ実行履歴ファイル21F又は変更データ実行履歴ファイル210Fを参照した場合、`toplist1`と`init1`とは入れ子構造であることが予想される。そうすると、`toplist1`から`init1`が呼び出された可能性が示唆される。

20

## 【 0 1 0 2 】

プログラム解析装置1によれば、入力データの変化に基づいて、プログラムに含まれる関数の動作変化を取得することが可能である。

解析対象プログラムの機能毎に実行履歴から抽出された関数の属性は、入力データの変化に対する動作変化に基づいて4種類に分類される。差分リストファイル230Fにおいて、関数は基準データSD及び変更データCDの差分である差分データDDと、対応づけられており、解析対象プログラムにおける各関数の動作の理解を支援することができる。

30

## 【 0 1 0 3 】

プログラム解析装置1によれば、解析対象プログラムの実行履歴に基づいて、関数の呼び出し回数、開始行間距離及び関数行間距離を求めることができる。これにより、プログラム解析装置1は、関数が上位の機能を有するか否かについて分析を行う上で、有益な情報を提供することができる。

## 【 0 1 0 4 】

なお、CPU11は、基準データ実行履歴ファイル21F、変更データ実行履歴ファイル210F、基準データ関数別計測ファイル22F、変更データ関数別計測ファイル220F及び関数別呼び出し履歴分析ファイル23Fをハードディスク14に記録する。しかし、CPU11は、これらのファイルをRAM13に記録してもよい。

40

## 【 0 1 0 5 】

図16の関数別呼び出し履歴分析ファイル23Fには、基準データSDに基づく関数の分析情報と変更データCDに基づく関数の分析情報とが含まれていた。双方を比較することにより、入力データの変化に基づく関数の動作変化を知ることができる。しかし、基準データSDに基づく関数の分析情報と変更データCDに基づく関数の分析情報との差が定量的に求められてもよい。

図18は、基準データSDに基づく関数の分析情報と変更データCDに基づく関数の分析情報との差分を示す説明図である。図18の関数名は、図16の関数名と同じである。図16の呼び出し回数差、開始行間距離差及び関数行間距離差は、夫々図16の基準データSDに基づく関数の分析情報と変更データCDに基づく関数の分析情報との各差に該当

50

する。つまり、図18には、入力データの変化による関数の動作変化が定量的に示されている。

item1は、呼び出し回数に4回、開始行間距離に8行の変化がある。変更データCDの件数が基準データSDの件数より4件多いことを勘案した場合、item1は、商品在庫テーブルのレコードを1件ずつ処理する関数であることが想像される。

init1は、関数行間距離に8行の変化がある。この変化量はitem1の開始行間距離の変化に等しい。従って、init1は、item1を呼び出していることが想像される。

toplist1は、呼び出し回数及び開始行間距離に変化がなく、関数行間距離に8行の変化がある。この変化量はitem1及びinit1の開始行間距離の変化に等しい。従って、toplist1は、item1又はinit1を呼び出していることが想像される。

10

なお、基準データに対応する関数の分析情報と変更データに対する関数の分析情報との間に差がないことも、リバースエンジニアリングの情報として有益である。従って、図18において、0と記された情報からも、関数の解釈に有益な情報が得られる。例えば、toplist1及びinit1は、入力データのデータ量が変化しても呼び出し回数は変化していない。そのため、toplist1及びinit1は、データ量の変化とは無関係な処理をしていることが想像される。

#### 【0106】

本実施例では、基準データSDと変更データCDとの間で、商品在庫テーブルのレコード数を変化させた。しかし、基準データSDと変更データCDとの間で、入力データの値を変化させてもよいことは勿論である。例えば、変更データCDの作成に際し、商品在庫テーブルのレコード数は基準データSDのレコード数と同じ2件とする。ただし、変更データCDの登録日時又は在庫数を基準データSDの登録日時又は在庫数と異なる値に変更する。

20

#### 【0107】

本実施例では、解析対象プログラムの機能毎に、関数別の分析を行った。しかし、関数間の構造が解析されてもよい。例えば、解析対象プログラムの実行履歴において、関数Aの呼び出し開始と呼び出し終了との間に、関数Bの呼び出し開始と呼び出し終了とがある場合、関数Bは関数Aの入れ子関数である可能性がある。かかる場合、関数Aから関数Bが呼び出される関係にあり、関数Aは上位の関数、関数Bは下位の関数であるといえる。

30

図19は、解析対象プログラムの実行履歴における関数間の構造の一例を示す説明図である。図19では、関数の呼び出し順序に基づいた入れ子構造が示されている。

図12又は図13の実行履歴において、toplist1の呼び出し開始と呼び出し終了との間に、init1の呼び出し開始と呼び出し終了とがある。また、init1の呼び出し開始と呼び出し終了との間に、item1の呼び出し開始と呼び出し終了とがある。従って、ソースコードにおいてもitem1はinit1の内部で入れ子になり、init1はtoplist1の内部で入れ子になっている可能性がある。

関数の呼び出し回数、開始行間距離及び関数行間距離から関数の動作を決定することが困難な場合であっても、関数の呼び出し順序に基づいた関数間の構造を勘案することにより、関数動作推定の支援が可能となる。

40

#### 【0108】

本実施の形態で扱った関数は、手続き型言語、命令型言語、構造化言語等における関数に限らない。本実施の形態で扱った関数は、オブジェクト指向言語におけるクラスのメソッドに置き換えてもよい。

#### 【0109】

プログラム解析装置1を動作させるためのプログラム1Pは、ディスクドライブ15に光ディスク1aを読み取らせてハードディスク14に記録してもよい。また、プログラム1Pは、インターネットNを介して接続される他の情報処理装置又は記録装置(図示せず)からダウンロードすることも可能である。あるいは、また、プログラム1Pを記録した

50

フラッシュメモリ等の半導体メモリ 1 c がプログラム解析装置 1 内に実装されてもよい。

【 0 1 1 0 】

以上の実施の形態に関し、さらに以下の付記を開示する。

【 0 1 1 1 】

( 付記 1 )

対象プログラムを解析する処理をコンピュータに実行させるコンピュータプログラムにおいて、

第 1 入力データ及び該第 1 入力データと異なる第 2 入力データに夫々基づく前記対象プログラムの第 1 駆動履歴及び第 2 駆動履歴を記録する記録手段に記録された該第 1 駆動履歴から、該対象プログラムに含まれる関数の呼び出し情報に基づいて、該関数の第 1 呼び出し履歴を抽出し、

10

前記記録手段に記録された第 2 駆動履歴から、前記関数の呼び出し情報に基づいて、該関数の第 2 呼び出し履歴を抽出し、

前記関数の第 1 及び第 2 呼び出し履歴に夫々基づいて、前記第 1 及び第 2 駆動履歴各々における該関数の呼び出し回数又は呼び出し箇所を取得する

処理をコンピュータに実行させるコンピュータプログラム。

【 0 1 1 2 】

( 付記 2 )

前記関数の呼び出し情報は該関数の呼び出し開始又は呼び出し終了を示す情報を含み、前記関数の第 1 及び第 2 呼び出し履歴を抽出する処理は、該関数の呼び出し開始又は呼び出し終了に係る第 1 及び第 2 呼び出し履歴を夫々抽出し、

20

前記関数の呼び出し回数又は呼び出し箇所を取得する処理は、該関数の呼び出し開始又は呼び出し終了に係る第 1 及び第 2 呼び出し履歴に夫々基づいて、前記第 1 及び第 2 駆動履歴各々における該関数の呼び出し回数又は呼び出し箇所を取得する

付記 1 に記載のコンピュータプログラム。

【 0 1 1 3 】

( 付記 3 )

前記対象プログラムの第 1 及び第 2 駆動履歴は行単位の文字列であり、前記関数の呼び出し箇所を取得する処理は、該関数の呼び出し開始又は呼び出し終了に対応する行番号を取得する

30

付記 2 に記載のコンピュータプログラム。

【 0 1 1 4 】

( 付記 4 )

前記関数の呼び出し開始又は呼び出し終了に対応する行番号に基づいて、該関数の処理に関するステップ数に応じた行の数量を算出する

付記 3 に記載のコンピュータプログラム。

【 0 1 1 5 】

( 付記 5 )

前記ステップ数に応じた行の数量を算出する処理は、前記対象プログラムの第 1 及び第 2 駆動履歴各々における関数の同一呼び出し回数目の呼び出し開始及び呼び出し終了に夫々対応する行番号に基づいて、前記ステップ数に応じた行の数量を算出する

40

付記 4 に記載のコンピュータプログラム。

【 0 1 1 6 】

( 付記 6 )

前記ステップ数に応じた行の数量を算出する処理は、前記対象プログラムの第 1 及び第 2 駆動履歴各々における関数の呼び出し開始に対応する最初の行番号及び呼び出し開始に対応する最後の行番号、又は該関数の呼び出し終了に対応する最初の行番号及び呼び出し終了に対応する最後の行番号に基づいて、前記ステップ数に応じた行の数量を算出する

付記 4 に記載のコンピュータプログラム。

【 0 1 1 7 】

50

(付記7)

前記関数の呼び出し回数又はステップ数に応じた行の数量に基づいて、前記第1及び第2入力データに夫々対応する該関数の呼び出し回数又はステップ数に応じた行の数量の間の同異又は変化量を求める

付記4から付記6までのいずれか一つに記載のコンピュータプログラム。

【0118】

(付記8)

前記第1入力データは値を有し、

前記第2入力データは前記第1入力データの値又はデータ量を変更したデータである

付記1から付記7までのいずれか一つに記載のコンピュータプログラム。

10

【0119】

(付記9)

前記対象プログラムは機械語により記述されている

付記1から付記8までのいずれか一つに記載のコンピュータプログラム。

【0120】

(付記10)

対象プログラムを解析するプログラム解析方法において、

コンピュータが、

第1入力データ及び該第1入力データと異なる第2入力データに夫々基づく前記対象プログラムの第1駆動履歴及び第2駆動履歴を記録する記録手段に記録された該第1駆動履歴から、該対象プログラムに含まれる関数の呼び出し情報に基づいて、該関数の第1呼び出し履歴を抽出し、

20

前記記録手段に記録された第2駆動履歴から、前記関数の呼び出し情報に基づいて、該関数の第2呼び出し履歴を抽出し、

前記関数の第1及び第2呼び出し履歴に夫々基づいて、前記第1及び第2駆動履歴各々における該関数の呼び出し回数又は呼び出し箇所を取得するプログラム解析方法。

【0121】

(付記11)

対象プログラムを解析するプログラム解析装置において、

第1入力データ及び該第1入力データと異なる第2入力データに夫々基づく前記対象プログラムの第1駆動履歴及び第2駆動履歴を記録する記録手段と、

30

該記録手段に記録された第1駆動履歴から、前記対象プログラムに含まれる関数の呼び出し情報に基づいて、該関数の第1呼び出し履歴を抽出する第1抽出部と、

前記記録手段に記録された第2駆動履歴から、前記関数の呼び出し情報に基づいて、該関数の第2呼び出し履歴を抽出する第2抽出部と、

前記第1及び第2抽出部が夫々抽出した関数の前記第1及び第2呼び出し履歴に基づいて、前記第1及び第2駆動履歴各々における該関数の呼び出し回数又は呼び出し箇所を取得する取得部と

を備えるプログラム解析装置。

40

【符号の説明】

【0122】

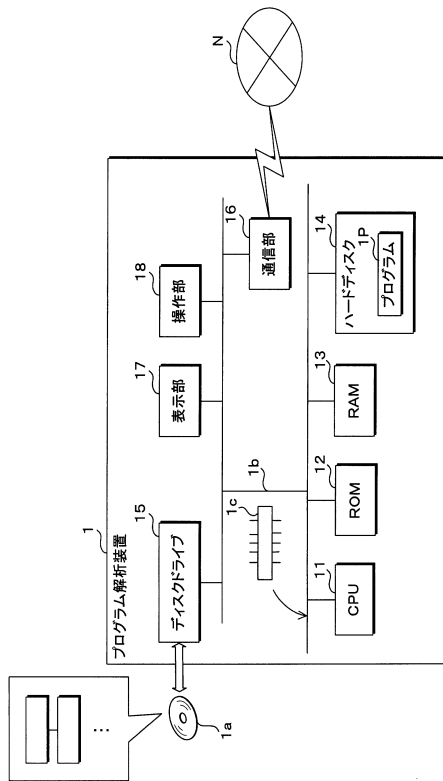
- |    |             |
|----|-------------|
| 1  | プログラム解析装置   |
| 11 | CPU         |
| 13 | RAM         |
| 14 | ハードディスク     |
| 22 | 関数呼び出し履歴計測部 |
| 23 | 関数呼び出し履歴分析部 |
| SD | 基準データ       |
| CD | 変更データ       |

50

- DD 差分データ
- 21F 基準データ実行履歴ファイル
- 210F 変更データ実行履歴ファイル
- 22F 基準データ関数別計測ファイル
- 220F 変更データ関数別計測ファイル
- 23F 関数別呼び出し履歴分析ファイル
- 230F 差分リストファイル
- 1P プログラム
- 1a 光ディスク

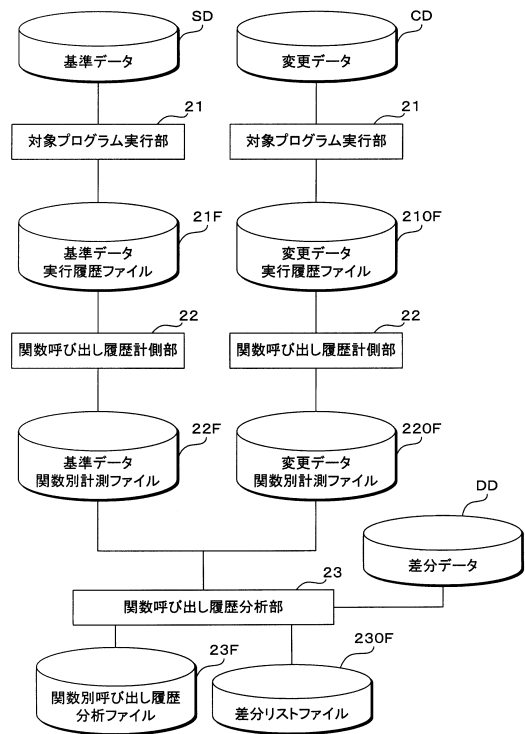
【図1】

プログラム解析装置のハードウェア構成例を示すブロック図



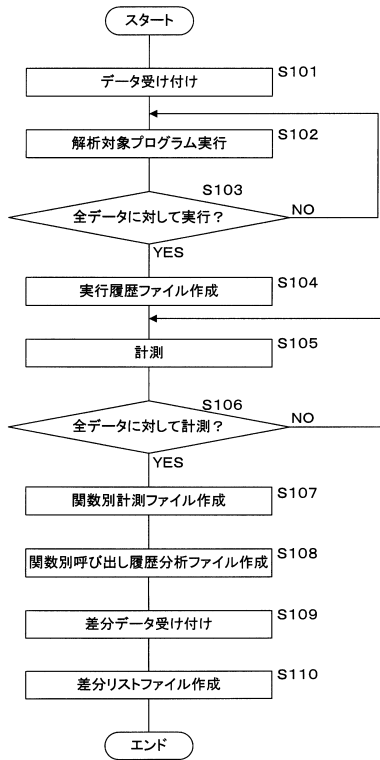
【図2】

データ処理の流れの一例を示すデータフロー図



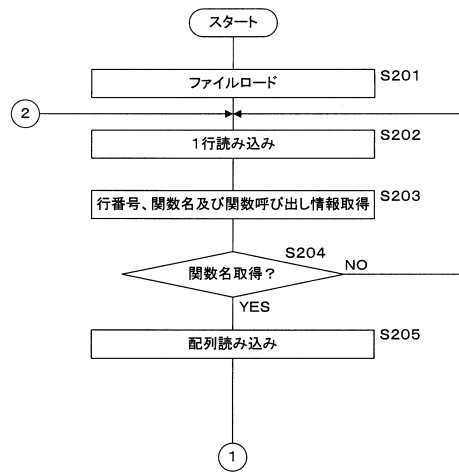
【図3】

プログラム解析処理の手順の一例を示すフローチャート



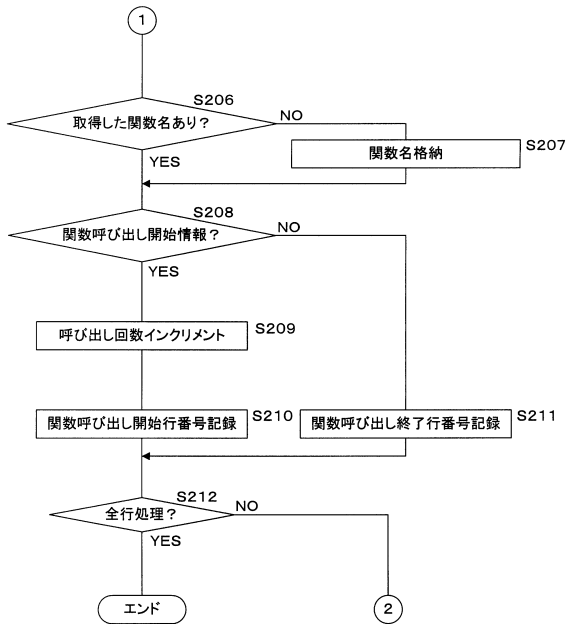
【図4】

関数の呼び出し履歴を計測する処理の手順の一例を示すフローチャート



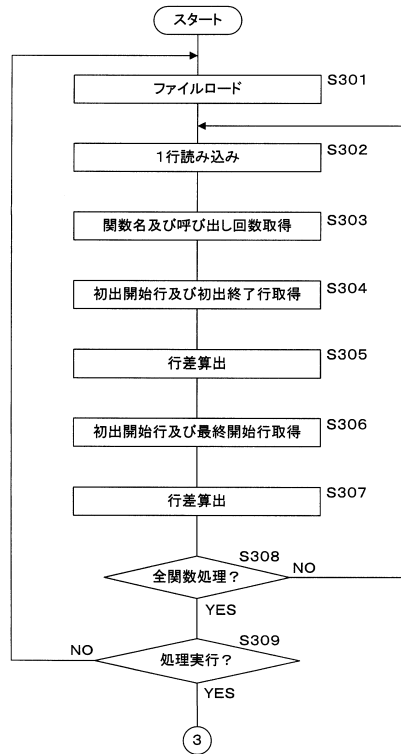
【図5】

関数の呼び出し履歴を計測する処理の手順の一例を示すフローチャート



【図6】

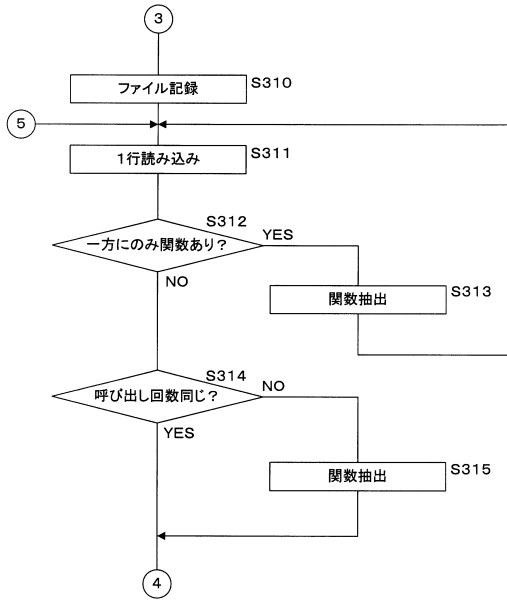
関数の呼び出し履歴を分析する処理の手順の一例を示すフローチャート





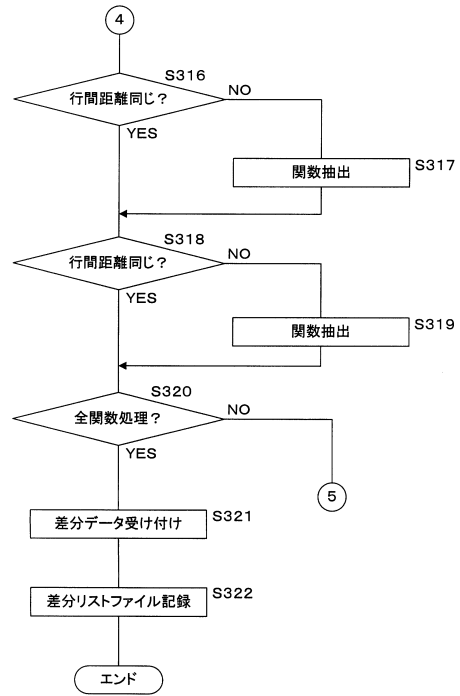
【図7】

関数の呼び出し履歴を分析する処理の手順の一例を示すフローチャート



【図8】

関数の呼び出し履歴を分析する処理の手順の一例を示すフローチャート



【図9】

基準データである商品在庫テーブルのレコード例を示す説明図

商品コード	商品名	登録日時	在庫数
01	商品A1	2010/1/1	10
02	商品A2	2010/1/1	10

【図11】

差分データの一例を示す説明図

差分テーブル名	商品在庫テーブル
差分項目名	商品コード 03
	商品コード 04
	商品コード 05
	商品コード 06

【図10】

変更データである商品在庫テーブルのレコード例を示す説明図

商品コード	商品名	登録日時	在庫数
01	商品A1	2010/1/1	10
02	商品A2	2010/1/1	10
03	商品A3	2010/1/1	10
04	商品A4	2010/1/1	10
05	商品A5	2010/1/1	10
06	商品A6	2010/1/1	10

【図12】

基準データ実行履歴ファイルの内容の一例を示す説明図

```

1>toplist
2a = 1
3print a
4if(a == 1)
5print a
6>init1
7<init1
8>item1
9<item1
10>item1
11<item1
12>init1
13<init1
14<toplist1
  
```

【 図 1 3 】

変更データ実行履歴ファイルの内容の一例を示す説明図

```

1>toplist1
2a = 1
3print a
4if (a == 1)
5print a
6>init1
7<init1
8>item1
9<item1
10>item1
11<item1
12>item1
13<item1
14>item1
15<item1
16>item1
17<item1
18>item1
19<item1
20>init1
21<init1
22<toplist1

```

【 図 1 5 】

変更データ関数別計測ファイルの内容の一例を示す説明図

関数名	初出開始行	初出終了行	最終開始行	最終終了行	呼び出し回数
toplist1	1	22	1	22	1
item1	8	9	18	19	6
init1	6	7	20	21	2

【 図 1 4 】

基準データ関数別計測ファイルの内容の一例を示す説明図

関数名	初出開始行	初出終了行	最終開始行	最終終了行	呼び出し回数
toplist1	1	14	1	14	1
item1	8	9	10	11	2
init1	6	7	12	13	2

【 図 1 6 】

関数別呼び出し履歴分析ファイルの内容の一例を示す説明図

関数名	基準データ		変更データ	
	呼び出し回数	開始行間距離	開始行間距離	関数行間距離
toplist1	1	0	13	0
item1	2	2	1	10
init1	2	6	1	14

【 図 1 7 】

差分リストファイルの内容の一例を示す説明図

差分データ	商品在庫グループ、商品コード=03、04、05、06
一方の分析情報にのみ存在する関数	
呼び出し回数異なる関数	item1
開始行間距離異なる関数	item1、init1
関数行間距離異なる関数	toplist

## 【図 18】

基準データに基づく関数の分析情報と変更データに基づく関数の分析情報との差分を示す説明図

関数名	呼び出し回数差	開始行間距離差	関数行間距離差
toplist1	0	0	8
item1	4	8	0
init1	0	8	0

## 【図 19】

解析対象プログラムの実行履歴における関数間の構造の一例を示す説明図

入れ子関数呼び出し元	入れ子関数呼び出し先
toplist1	init1
init1	item1

---

フロントページの続き

- (72)発明者 松尾 昭彦  
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
- (72)発明者 小林 健一  
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

審査官 大塚 俊範

- (56)参考文献 特開2010-102663(JP,A)  
特開2009-211424(JP,A)  
特開2010-049489(JP,A)  
米国特許出願公開第2005/0183067(US,A1)

- (58)調査した分野(Int.Cl., DB名)
- |      |               |
|------|---------------|
| G06F | 11/28 - 11/34 |
| G06F | 9/44          |