(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2014 年 10 月 2 日(02.10.2014)

WIPO | PCT

(10) 国際公開番号
WO 2014/157056 A1

(51) 国際特許分類:
G06F 9/44 (2006.01)

(21) 国際出願番号: PCT/JP2014/058021

(22) 国際出願日: 2014 年 3 月 24 日(24.03.2014)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願 2013-074623 2013 年 3 月 29 日(29.03.2013) JP

(71) 出願人: ＮＥＣソリューションイノベータ株式会社(NEC SOLUTION INNOVATORS, LTD.) [JP/JP]; 〒1368627 東京都江東区新木場一丁目１８番７号 Tokyo (JP).

(72) 発明者: 坂本　久(SAKAMOTO Hisashi); 〒1368627 東京都江東区新木場一丁目１８番７号　ＮＥＣソリューションイノベータ株式会社内 Tokyo (JP).

(74) 代理人: 木村　満(KIMURA Mitsuru); 〒1010054 東京都千代田区神田錦町二丁目７番地　協販ビル２階 Tokyo (JP).

(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

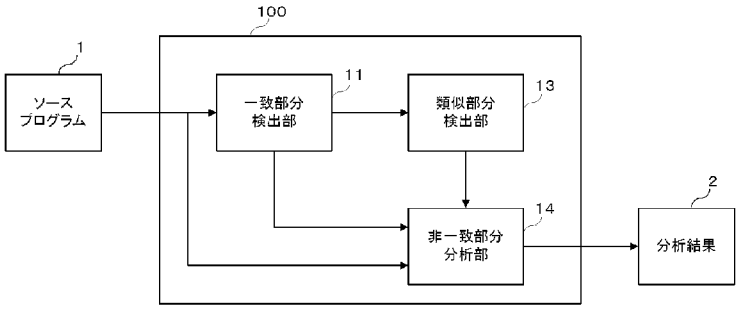
(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

添付公開書類:
— 国際調査報告（条約第 21 条(3)）

WO 2014/157056 A1

(54) **Title**: SOURCE PROGRAM ANALYSIS SYSTEM, SOURCE PROGRAM ANALYSIS METHOD, AND RECORDING MEDIUM ON WHICH PROGRAM IS RECORDED

(54) 発明の名称 : ソースプログラム解析システム、ソースプログラム解析方法およびプログラムを記録した記録媒体



1 Source program
2 Analysis result
11 Matching portion detection unit
13 Similar portion detection unit
14 Non-matching portion analysis unit

(57) **Abstract**: A matching portion detection unit (11) detects matching portions for which the content matches in a source program (1). On the basis of the matching portions detected by the matching portion detection unit (11) a similar portion detection unit (13) detects similar portions in the source program, which conform to a standard and are configured with portions for which the content matches and portions for which the content does not match. A non-matching portion analysis unit analyzes the similar portions detected by the similar portion detection unit (13), and outputs an index indicating the degree of similarity between a given similar portion and another similar portion having a portion matching the given similar portion.

(57) 要約:

[続葉有]

一致部分検出部（１１）は、ソースプログラム（１）の中の内容が互いに一致する一致部分を検出する。類似部分検出部（１３）は、一致部分検出部（１１）が検出した一致部分に基づいて、ソースプログラム（１）の中の内容が互いに一致する部分と一致しない部分とで構成され、かつ、基準に合致する類似部分を検出する。非一致部分解析部（１４）は、類似部分検出部（１３）が検出した類似部分を分析し、当該類似部分と互いに一致する部分を有する他の類似部分とどれだけ類似しているかを示す指標を出力する。

明　細　書

発明の名称 ：

ソースプログラム解析システム、ソースプログラム解析方法およびプログラムを記録した記録媒体

技術分野

[0001] 本発明は、ソースプログラム解析システム、ソースプログラム解析方法およびプログラムを記録した記録媒体に関する。

背景技術

[0002] 現在、多くの企業や組織で大規模なコンピュータシステムが様々な用途で稼働している。大規模なコンピュータシステムを運営する上では、稼働しているソフトウェアの不具合の修正、あるいは必要な機能の追加などの保守活動が必要である。コンピュータシステムを利用する企業や組織は、こういった保守活動のためのコストを的確に把握したいという要求を持っている。

[0003] ソフトウェアの保守コストを見積もる上で、いわゆるコードクローンの位置や数から保守コストを見積もる技術が存在する。コードクローンとは、プログラムのソースコード上に存在する部分的な複製を指す。一般的に、コードクローンが多数存在すると、ソフトウェアの保守性が低下することが知られている。これは例えば、コードクローンの中に不具合が発見された場合、不具合を解消するために、開発者は同一の内容を持つコードクローンについて、個別に修正を要するか検討しなければならないためである。特に大規模なシステムでは、ソフトウェアにある全てのコードクローンについて検討することは、膨大な労力を要する。

[0004] コードクローンの検出に関して、現在までいろいろな研究がなされている。非特許文献１は、コードクローンを以下の３種類に分類した。タイプ１は、空白、括弧等を除いて完全に一致するコードクローンである。タイプ２は、変数名、ラベル名、手続き名等のユーザ定義名称、または変数の型など一部の予約語のみが異なるコードクローンである。タイプ３は、タイプ２のコ

ードクローンに、さらに文の挿入や削除、変更が施されているコードクローンである。非特許文献１は、タイプ１および２のコードクローンを検出する技術を開示している。

[0005] 非特許文献２は、コードクローンを検出するシステムおよびソフトウェアを開示している。非特許文献２が開示するソフトウェア（ＣＣＦｉｎｄｅｒＸ）は、処理対象のソースプログラムを字句解析し、ソースプログラムに含まれるコードクローンを検出する。このソフトウェアは、上記のタイプ１およびタイプ２のコードクローンを検出することができる。

[0006] 特許文献１は、関数単位でコードクローンを検出し、そのコードクローンの類似性を評価するシステムを開示している。特許文献１が開示するシステムは、ソースプログラムを関数単位で解析し、利用者が指定した類似度検出条件に合致するものを検出する。利用者は、内容が一致するもの（タイプ１のコードクローン）、あるいは一部が異なるもの（タイプ２または３のコードクローン）の検出を指定できる。また、特許文献１が開示するシステムは、検出したコードクローンにおいて、ソースコードが互いに一致するか、関数インターフェースが同一かの２点から評価し、その結果からコードクローンを４カテゴリに分類する。操作者は、このカテゴリを参照することで、例えば関数インターフェースのみ異なるコードクローンが他にあるか、ソースコードが一致するコードクローンが他にあるかを知ることができ、コードクローンを再利用する際に参考にすることができる。

先行技術文献

特許文献

[0007] 特許文献1：特開２０１２－１６４２１１号公報

非特許文献

[0008] 非特許文献1：肥後芳樹、楠本信二、井上克郎、「コードクローン検出とその関連技術」電子情報通信学会論文誌Ｄ、ＶＯＬ．Ｊ９１－Ｄ、Ｎｏ６（２００８）、pp．１４６５－１４８１．

非特許文献2：ＣＣＦｉｎｄｅｒホームページ＜URL: http://www.ccfinder.n

et ＞

発明の概要

発明が解決しようとする課題

[0009] 上述の文献で開示された技術では、文の挿入や削除、変更が施されているコードクローンの保守コストを算出するための有効な情報を出力できないという課題がある。

[0010] より具体的に説明すると、非特許文献１および２は、文の挿入や削除、変更が施されているコードクローンを検出する技術を開示していない。ソフトウェア開発において開発者がソースプログラムの一部をコピーして再利用する場合、そのまま利用することよりも、改変を加えて利用することのほうが多い。従って、ソフトウェアを修正や改良する際にはタイプ３のコードクローンも対象とする必要がある。この点で、非特許文献１および２の技術によるコードクローンの検出だけでは不十分である。また、非特許文献１および２は、検出したコードクローンの類似性や保守性を評価する技術を開示していない。従って、これらの文献が開示する技術では、検出したコードクローンの保守作業のコストを見積もる上で指標となる情報を提供することができない。

[0011] また、特許文献１が開示する技術は、関数単位でのコードクローンを検出することができるが、関数内に部分的に存在するコードクローンを検出できない。ソフトウェアを保守する上では関数内に部分的に存在するコードクローンへの修正を要するケースもあり、そのような場合は、特許文献１が開示する技術は十分な情報を出力することができない。また、関数のないプログラム言語で記述されたソースプログラムを解析することもできない。また、特許文献１が開示するシステムは、ソースコードが互いに一致するか、関数インターフェースが同一かの２点から検出したコードクローンを評価する。しかし、この２つの観点による指標は、コードクローン間の類似の度合いを示すのみであり、ソフトウェアの修正や改良を行う際に、どのコードクローンをどの程度変更する必要があるかを示すことができない。そのため、特許

文献１が開示する技術は、ソフトウェアの保守コストを算出する上で有効な情報を提供することができなかった。

[0012] 本発明は、以上の問題点に着目してなされたものであり、文の挿入や削除、変更が施されているコードクローンを検出し、検出したコードクローンに基づいてそのプログラムの修正コストを算出するために有効な情報を出力するソースプログラム解析システム、ソースプログラム解析方法およびプログラムを記録した記録媒体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] 上記の目的を達成するため、本発明の第１の観点にかかるソースプログラム解析システムは、

ソースプログラムの中の、内容が互いに一致する一致部分を検出する一致部分検出手段と、

前記一致部分検出手段が検出した一致部分に基づいて、ソースプログラムの中の、内容が互いに一致する部分と一致しない部分とで構成され、かつ、基準に合致する類似部分を検出する類似部分検出手段と、

前記類似部分検出手段が検出した類似部分を分析し、当該類似部分と互いに一致する部分を有する他の類似部分とどれだけ類似しているかを示す指標を出力する非一致部分分析手段と、を備える、

ことを特徴とする。

[0014] 上記の目的を達成するため、本発明の第２の観点にかかるソースプログラム解析方法は、

ソースプログラムの中の、内容が互いに一致する一致部分を検出する一致部分検出ステップと、

前記一致部分検出ステップで検出した一致部分に基づいて、ソースプログラムの中の内容が互いに一致する部分と一致しない部分とで構成され、かつ、基準に合致する類似部分を検出する類似部分検出ステップと、

前記類似部分検出ステップで検出した類似部分を分析し、当該類似部分と互いに一致する部分を有する他の類似部分とどれだけ類似しているかを示す

指標を出力する非一致部分分析ステップと、を備える、

ことを特徴とする。

[0015] 上記の目的を達成するため、本発明の第３の観点にかかる記録媒体は、コンピュータに、

ソースプログラムの中の、内容が互いに一致する一致部分を検出する一致部分検出機能、

前記一致部分検出機能が検出した一致部分に基づいて、ソースプログラムの中の内容が互いに一致する部分と一致しない部分とで構成される所定の基準を満たす類似部分を検出する類似部分検出機能、

前記類似部分検出機能が検出した類似部分を分析し、当該類似部分と互いに一致する部分を有する他の類似部分とどれだけ類似しているかを示す指標を出力する非一致部分分析機能、

を実現させるプログラムを記録している。

発明の効果

[0016] 本発明によれば、文の挿入や削除、変更が施されているコードクローンを検出し、検出したコードクローンに基づいてプログラムの修正コストを算出するために有効な情報を出力するソースプログラム解析システム、ソースプログラム解析方法およびプログラムを記録した記録媒体を実現できる。

図面の簡単な説明

[0017] [図1]本発明の実施形態にかかるソースプログラム解析システムのハードウェア構成を示した図である。

[図2]ソースプログラム解析システムの機能を示したブロック図である。

[図3]ソースプログラム解析システムのデータの流れを示した図である。

[図4]一致部分情報ＤＢの内容を示した図である。

[図5]文法定義情報ＤＢの内容を示した図である。

[図6]実行単位標識情報ＤＢの内容を示した図である。

[図7]類似部分情報ＤＢの内容を示した図である。

[図8]非一致構造情報ＤＢの内容を示した図である。

[図9]実施形態の説明におけるソースプログラムを示した図である。

[図10]ソースプログラム解析処理を示したフローチャートである。

[図11]一致部分検出処理を示したフローチャートである。

[図12]一致部分検出処理後の一致部分情報ＤＢの内容を示した図である。

[図13]実行単位標識検出処理を示したフローチャートである。

[図14]実行単位標識検出処理で参照される文法定義情報ＤＢの内容を示した図である。

[図15]実行単位標識検出処理後の実行単位標識情報ＤＢの内容を示した図である。

[図16]類似部分検出処理を示したフローチャートである。

[図17]一致部分の出現順による類似部分検出処理を示したフローチャートである。

[図18]一致部分の前後部分を含む類似部分検出処理を示したフローチャートである。

[図19]類似部分検出処理後の類似部分情報ＤＢの内容を示した図である。

[図20]非一致部分分析処理を示したフローチャートである。

[図21A]非一致部分分析処理により得られる構文構造例を示した図である。

[図21B]非一致部分分析処理により得られる構文構造例を示した図である。

[図21C]非一致部分分析処理により得られる構文構造例を示した図である。

[図22]非一致部分分析処理後の非一致構造情報ＤＢの内容を示した図である。

[図23]本発明の実施形態にかかるソースプログラム解析システムの機能概要を示した図である。

発明を実施するための形態

[0018] 以下、本発明の実施形態について、図面を参照しながら説明する。

[0019] （実施形態）

本発明の実施形態にかかるソースプログラム解析システムは、ソースプログラムを解析し、文の挿入や削除、変更が施されている（すなわち非一致部

分を有する）コードクローンを検出し、検出したコードクローンの非一致部分のプログラム構造を解析し、類似性に関する指標を出力する。操作者は、この類似性に関する指標を参照し、そのプログラムの修正コストを算出することができる。

[0020] 本実施形態にかかるソースプログラム解析システム１００は、図１に示すように、制御部１１０、記憶部１２０、操作部１３０、画面表示部１４０を備える。ソースプログラム解析システム１００の構成要素は、バス１０１を介して相互に接続されている。

[0021] 制御部１１０は、ＣＰＵ（Central Processing Unit）１１１、ＲＯＭ（Read Only Memory）１１２、ＲＡＭ（Random Access Memory）１１３を備える。ＣＰＵ１１１は、ＲＯＭ１１２などに記憶されている制御用のプログラムに従って、ソースプログラム解析システム１００全体の動作を制御する。ＲＯＭ１１２は、制御用のプログラムを記憶する。ＲＡＭ１１３は、ＣＰＵ１１１が使用するデータを一時的に保存する。制御部１１０は、ソースプログラム解析システム１００が備える他の構成要素との間で、バス１０１を介して信号を送受信することにより、これらの構成要素を制御する。

[0022] 記憶部１２０は、ハードディスク、ＳＳＤ（Solid State Drive）、あるいはＣＤ－ＲＯＭ（Compact Disc - Read Only Memory）等を備え、ソースプログラム解析システム１００が稼動する上で必要な情報を記憶する。

[0023] 操作部１３０は、操作者の操作を入力として受けとり、制御部１１０に情報として伝達する。操作部１３０は、例えばマウスやキーボード、あるいはタッチパネルなどを備える。

[0024] 画面表示部１４０は、制御部１１０の制御に基づき、操作画面や結果表示画面などのソースプログラム解析システム１００が操作者に対して表示すべき情報を表示する。画面表示部１４０は、液晶ディスプレイや有機エレクトロルミネセンスディスプレイ等で構成される。

[0025] ソースプログラム解析システム１００が動作する際、図２に示すように、制御部１１０は、一致部分検出部１１、実行単位標識検出部１２、類似部分

検出部１３、非一致部分分析部１４として機能する。また、記憶部１２０は、一致部分情報ＤＢ（データベース）２１、文法定義情報ＤＢ２２、実行単位標識情報ＤＢ２３、類似部分情報ＤＢ２４、非一致構造情報ＤＢ２５として機能する。

[0026] 次に、図２に示す各部の機能および動作を、図３を参照して説明する。ソースプログラム解析システム１００は、全体としては、ソースプログラム１からコードクローンを検出し、検出したコードクローンの類似性に関する指標を非一致構造情報ＤＢ２５に記録する。

[0027] 一致部分検出部１１は、ソースプログラム１から、内容が互いに一致する一致部分を検出する。この一致部分は、上述したタイプ１あるいはタイプ２のコードクローンに該当する。一致部分検出部１１は、検出した一致部分ごとに一致部分情報を生成し、生成した一致部分情報を一致部分情報ＤＢ２１に記録する。一致部分検出部１１の機能を実現するには、例えば非特許文献２が開示する技術およびソフトウェアを用いることができる。本実施形態においては、非特許文献２が開示するソフトウェアのひとつであるＣＣｆｉｎｄｅｒＸを使用する。一致部分検出部１１は、ソースプログラム１を行ごとに解析し、行単位で一致部分を検出し、検出した一致部分の位置をソースプログラム上の開始行番号および終了行番号で特定する。ここで行とは、ソースプログラム上において、改行を示すコードに挟まれた部分のコードの集合を指す。

[0028] 実行単位標識検出部１２は、文法定義情報ＤＢ２２に記録された文法定義に基づき、入力されたソースプログラムの実行単位の開始あるいは終了を示す標識を検出し、実行単位標識情報を生成し、実行単位標識情報ＤＢ２３に記録する。ここで実行単位とは、各プログラム言語の構文規則で定義されているセクション（節）、ディビジョン(部）、段落、手続き（サブルーチン）、関数、ブロック等、プログラムを分割する上で基準となる単位を指す。

[0029] 類似部分検出部１３は、一致部分情報ＤＢ２１に記録された一致部分情報に基づいて、ソースプログラムから基準に合致する類似部分を検出し、検出

した類似部分に関する類似部分情報を生成し、生成した類似部分情報を類似部分情報ＤＢ２４に記録する。類似部分検出部１３は、検出した類似部分の位置をソースプログラム上の開始行番号および終了行番号で特定する。類似部分検出部１３は、以下に示す２パターンのいずれかに合致する部分を、基準に合致する類似部分として検出する。

[0030] 第１のパターンの類似部分は、複数の一致部分と、その一致部分に挟まれた部分からなるものである。この類似部分は、両端の一致部分と、その間の内容が一致しない非一致部分を有し、文の挿入や削除、変更が施されているコードクローン（タイプ３のコードクローン）に該当する。第１のパターンの類似部分を検出するとき、類似部分検出部１３は、一致部分情報ＤＢ２１に記録された一致部分情報を参照し、複数の一致部分が、間に他の一致部分を挟まずに同じ出現順番でソースプログラムに配置されているものを検出する。そして、検出した複数の一致部分および当該一致部分の間にある部分を、ひとつの類似部分（第１のパターンの類似部分）として抽出する。

[0031] 第２のパターンの類似部分は、一致部分と、ソースプログラム上その一致部分の直前にある非一致部分（前部分）と、その一致部分の直後にある非一致部分（後部分）とからなるものである。この類似部分は、一致部分と、その前あるいは後の非一致部分（前部分あるいは後部分）から構成されるため、文の挿入や削除、変更が施されているコードクローンに該当する。類似部分検出部１３は、一致部分情報ＤＢ２１に記録された一致部分情報、および実行単位標識情報ＤＢ２３に記録された実行単位標識情報を参照し、各一致部分の前部分および後部分を検出する。ここで、前部分とは、所定の区切りからその一致部分の開始行までの部分を指し、後部分とは、その一致部分の終了行から所定の区切りまでの部分を指す。所定の区切りとは、実行単位標識または他の一致部分の開始行もしくは終了行の中で、その一致部分に最も近いものを指す。第２のパターンの類似部分を検出するとき、類似部分検出部１３は、前部分あるいは後部分が検出された一致部分について、前部分から一致部分を経て後部分までの範囲を、ひとつの類似部分として抽出する。

なお、類似部分検出部１３は、前部分のみが検出された一致部分があると、前部分からその一致部分までを類似部分として抽出する。類似部分検出部１３は、後部分のみが検出された一致部分があると、一致部分から後部分までを類似部分として抽出する。

[0032] 非一致部分分析部１４は、類似部分情報ＤＢ２４に記録された類似部分を分析し、当該類似部分と互いに一致する部分を有する他の類似部分とどれだけ類似しているかを示す指標を非一致構造情報ＤＢ２５に記録する。具体的には、非一致部分分析部１４は、類似部分情報ＤＢ２４から類似部分の位置と、その類似部分に含まれる一致部分リストを読み出し、また一致部分情報ＤＢ２１から一致部分の位置を読み出し、類似部分の非一致部分の位置を特定する。非一致部分分析部１４は、特定した非一致部分の位置のプログラム記述をソースプログラム１から読み出す。非一致部分分析部１４は、読み出したプログラム記述を、文法定義情報ＤＢ２２に記録された文法定義に従って構文解析し、構文構造を取得する。非一致部分分析部１４は、取得した構文構造を比較し、各類似部分の非一致部分で共通する構文構造（非一致部分共通構文構造）を検出する。非一致部分分析部１４は、各類似部分の非一致部分の構文構造と、非一致部分共通構文構造とを比較し、類似する度合いを構文類似度として算出する。構文解析の際、非一致部分分析部１４は、ＬＲ法（Left-to-Right scanning）によって非一致部分のプログラム記述を解析し、構文木構造で表された構文構造を取得する。

[0033] また、図３に示す一致部分情報ＤＢ２１は、図４に示すように、コードクローンＩＤ（Ｔ２１０１）、グループＩＤ（Ｔ２１０２）、ファイル名（Ｔ２１０３）、開始行番号（Ｔ２１０４）、終了行番号（Ｔ２１０５）を有する。コードクローンＩＤは、一致部分検出部１１が、ソースプログラム１から検出した一致部分（タイプ１あるいはタイプ２のコードクローン）にユニークに付ける識別情報である。グループＩＤは、その一致部分が属するグループを示す識別情報である。同一のグループに属する一致部分は、互いに内容が一致する。ファイル名は、ソースプログラム１のファイル名を指す。開

始行番号は、ソースプログラム１上で当該一致部分が存在する部分の最初の行番号を示す。終了行番号は、ソースプログラム１上で当該一致部分が存在する部分の最後の行番号を示す。

[0034] 図３に示す文法定義情報ＤＢ２２は、図５に示すように、プログラム言語種別（Ｔ２２０１）、文法定義（Ｔ２２０２）を有する。プログラム言語種別は、後述する文法定義がどのプログラム言語に対応するものであるかを示す情報である。例えば「Ｃ（Ｃ言語）」「ＪＡＶＡ（登録商標）」などといった形で表現される。文法定義は、プログラム言語種別で示されたプログラム言語の文法を定義する情報である。文法定義は、例えば、関数の開始を示すコード、関数の終了を示すコード、変数の範囲、数値の範囲、演算子の種類やコード、代入演算子を示すコード、式の形式、代入式の形式等が、対象のプログラム言語ではどのように規定されているかを示す。

[0035] 図３に示す実行単位標識情報ＤＢ２３は、図６に示すように、実行単位標識ＩＤ（Ｔ２３０１）、実行単位種別（Ｔ２３０２）、開始終了属性（Ｔ２３０３）、ファイル名（Ｔ２３０４）、開始行番号（Ｔ２３０５）、終了行番号（Ｔ２３０６）を有する。実行単位標識ＩＤは、実行単位標識検出部１２が、ソースプログラム１から検出した実行単位の開始あるいは終了を示すコード（標識）に付ける識別情報である。実行単位標識種別は、検出されたコードが、セクション（節）、ディビジョン(部）、段落、手続き（サブルーチン）、関数、ブロック等の実行単位のどれに対応するものであるかを示す情報である。開始終了属性は、検出されたコードが実行単位の開始に対応するものであるか、終了に対応するものであるかを表す情報である。ファイル名は、ソースプログラム１のファイル名を示す。開始行番号は、検出されたコードが開始する行を示す。終了行番号は、検出されたコードが終了する行を示す。

[0036] 図３に示す類似部分情報ＤＢ２４は、図７に示すように、類似部分ＩＤ（Ｔ２４０１）、類似部分グループＩＤ（Ｔ２４０２）、ファイル名（Ｔ２４０３）、開始行番号（Ｔ２４０４）、終了行番号（Ｔ２４０５）、一致部分

リスト（Ｔ２４０６）からなる類似部分情報を記録する。類似部分ＩＤ（Ｔ２４０１）は、類似部分検出部１３が、ソースプログラム１から検出した類似部分にユニークに付ける識別情報である。類似部分グループＩＤ（Ｔ２４０２）は、類似部分が属する類似部分グループを示す情報である。同一の類似部分グループに属する類似部分は、同一のグループに属する一致部分を含む。ファイル名（Ｔ２４０３）は、ソースプログラム１のファイル名を示す。開始行番号（Ｔ２４０４）は、検出された類似部分が開始する行を示す。終了行番号（Ｔ２４０５）は、検出された類似部分が終了する行を示す。一致部分リスト（Ｔ２４０６）は、当該類似部分に含まれる一致部分のコードクローンＩＤを示す。一致部分リストは、一致部分に存在するコードクローンＩＤを出現する順にリスト化された形で保有する。

[0037] 図３に示す非一致構造情報ＤＢ２５は、図８に示すように、類似部分ＩＤ（Ｔ２５０１）、類似部分グループＩＤ（Ｔ２５０２）、非一致部分構文構造（Ｔ２５０３）、非一致部分共通構文構造（Ｔ２５０４）非一致部分構文類似度（Ｔ２５０５）からなる非一致構造情報を記録する。類似部分ＩＤ（Ｔ２５０１）は、類似部分を識別するための識別情報である。類似部分グループＩＤ（Ｔ２５０２）は、類似部分のグループを識別するための識別情報である。非一致部分構文構造（Ｔ２５０３）は、類似部分が保持する非一致部分の構文構造を示す。非一致部分構文構造は、ＸＭＬ（eXtensible Markup Language）、あるいはＤＯＭ（Document Object Model）等の方式によって階層構造で表現されたデータとして、あるいはバイナリデータとして記録される。非一致部分共通構文構造（Ｔ２５０４）は、同一の類似部分グループＩＤを有する類似部分の非一致部分が共通に保持する構文構造（非一致部分共通構文構造）を示す。非一致部分共通構文は、ＸＭＬ（eXtensible Markup Language）、あるいはＤＯＭ（Document Object Model）等の方式によって階層構造で表現されたデータとして、あるいはバイナリデータとして記録される。非一致部分構文類似度（Ｔ２５０５）は、類似部分の非一致部分が、非一致部分共通構文構造とどの程度類似しているかを示す数値指標である。

具体的には、階層構造で表現された非一致部分構文構造において、全体のノード数のうちどれだけの割合のノードが非一致部分共通構文構造と共通しているかをパーセンテージで示す。

[0038] 次に、ソースプログラム解析システム１００がソースプログラムを解析する動作について、図９に示すソースプログラムを解析する場合を例に説明する。

[0039] 操作者が、操作部１３０を介してソースプログラム解析システム１００に、ソースプログラム１の解析処理を開始する指示を入力すると、ソースプログラム解析システム１００は、図１０に示すソースプログラム解析処理を開始する。

[0040] 最初に、制御部１１０は、一致部分検出処理を開始する（ステップＳ１００）。

[0041] この一致部分検出処理（ステップＳ１００）の詳細について、図１１を参照して説明する。制御部１１０は、図１１に示すように、ソースプログラム１に存在する一致部分を検出する（ステップＳ１０１）。具体的には、制御部１１０は図９に示すソースプログラム１のプログラム記述を解析し、第０５行から第０６行のソースコードと、第１１行から第１２行のソースコードとが一致することを検出する。これを受けて、制御部１１０は、第０５行から第０６行のソースコードと、第１１行から第１２行のソースコードを一致部分として検出する。同様に、制御部１１０は、第０８行から第１０行、および第１４行から第１６行の各ソースコードも一致部分として検出する。

[0042] 次に制御部１１０は、ステップＳ１０１で検出した各一致部分に識別番号を付与する（ステップＳ１０２）。具体的には、制御部１１０は第０５行から第０６行、第０８行から第１０行、第１１行から第１２行、および第１４行から第１６行の各ソースコードに識別番号としてコードクローンＩＤ、グループＩＤを付与する。コードクローンＩＤを付与する際、制御部１１０は、検出した一致部分間でコードクローンＩＤが重複しないように採番する。グループＩＤを付与する際、制御部１１０は、内容が一致する一致部分間で

グループＩＤが一致するように、また内容が一致しないコードクローン間ではグループＩＤが重複しないように採番する。以上を踏まえ、制御部１１０は第０５行から第０６行のソースコードにコードクローンＩＤ「１」、グループＩＤ「１」を付与し、第０８行から第１０行のソースコードにコードクローンＩＤ「２」、グループＩＤ「２」を付与する。また、第１１行から第１２行のソースコードは第０５行から第０６行のソースコードと内容が一致する。そこで、制御部１１０は第１１行から第１２行のソースコードにはコードクローンＩＤ「３」を付与するとともに、グループＩＤは第０５行から第０６行のソースコードと一致する「１」を付与する。同様に、制御部１１０は第１４行から第１６行のソースコードにはコードクローンＩＤ「４」、グループＩＤ「２」を付与する。

[0043] 次に制御部１１０は、ステップＳ１０１で検出した各コードクローンのそれぞれについて、一致部分情報を一致部分情報ＤＢ２１に記録する（ステップＳ１０３）。一致部分情報は、各一致部分のコードクローンＩＤ、グループＩＤ、ファイル名、開始行番号、終了行番号からなる情報である。具体的には、第０５行から第０６行のソースコードは、コードクローンＩＤ「１」、グループＩＤ「１」であり、ファイル名はソースプログラム１のファイル名から「src.c」であり、開始行番号は「０５」であり、終了行番号「０６」である。制御部１１０は、これらを連結し、一致部分情報ＤＢ２１に記録する。また制御部１１０は、第０８行から第１０行、第１１行から第１２行、および第１４行から第１６行の各ソースコードについても同様の処理を行う。

[0044] 以上の処理を経て、一致部分情報ＤＢ２１には、図１２に示す情報が記録される。また、制御部１１０は、以上の処理において一致部分検出部１１として機能する。

[0045] 以上の処理が終了すると、制御部１１０は一致部分検出処理（ステップＳ１００）を終了し、図１０に示すフローチャートに従い、次に実行単位標識検出処理（ステップＳ２００）を開始する。ここから、制御部１１０が実行

する実行単位標識検出処理について、図１３を参照しながら説明する。

[0046] 実行単位標識検出処理の最初に、制御部１１０は、文法定義情報ＤＢ２２から、ソースプログラム１の記述言語に対応する文法定義情報を読み込む（ステップＳ２０１）。図９に示すようにソースプログラム１はＣ言語で記載されているため、制御部１１０は、図１４に示す文法定義情報ＤＢ２２から、Ｃ言語に対応する文法定義情報を読み込む。文法定義情報ＤＢ２２は、図１４に示すとおり、Ｃ言語に対応する文法定義情報を記録している。この文法定義情報には、関数の開始を示す標識として「｛」記号、関数の終了を示す標識として「｝」記号が定義されている。制御部１１０はこの文法定義情報に従い、ソースプログラム１の中で「｛」を関数の開始を示す記号として、「｝」を関数の終了を示す記号として扱う。

[0047] 次に、制御部１１０は、ソースプログラム１から実行単位の開始あるいは終了を示す標識を検出する（ステップＳ２０２）。具体的には、制御部１１０は、図９に示されたソースプログラム１を解析し、第０１行の「｛」記号、および第１８行の「｝」記号を実行単位の開始あるいは終了を示す標識として検出する。

[0048] 次に制御部１１０は、ステップＳ２０１で検出した実行単位の開始あるいは終了を示す標識のそれぞれについて、識別番号を付与する（ステップＳ２０３）。具体的には、制御部１１０は、ソースプログラム１の第０１行にある「｛」記号、および第１８行にある「｝」記号に、それぞれ実行単位標識ＩＤとして「１」「２」を付与する。実行単位標識ＩＤを付与する際、制御部１１０は、実行単位標識ＩＤが重複しないように採番する。

[0049] 次に制御部１１０は、ステップＳ２０１で検出した実行単位の開始あるいは終了を示す情報のそれぞれについて、実行単位標識情報を実行単位標識情報ＤＢ２３に記録する（ステップＳ２０４）。実行単位標識情報は、上述したとおり実行単位標識ＩＤ、実行単位種別、開始終了属性、ファイル名、開始行番号、終了行番号からなる情報である。具体的な状況としては、ソースプログラム１の第０１行にある「｛」記号は、実行単位標識ＩＤ「１」を付

与されており、「関数」の「開始」を示す属性を有しており、ファイル名はソースプログラム１のファイル名から「src.c」であり、開始行番号は「０１」であり、終了行番号「０１」である。制御部１１０は、これらの情報を連結し、実行単位標識情報ＤＢ２３に記録する。また制御部１１０は、ソースプログラム１の第１８行にある「}」記号についても同様の処理を行う。

[0050] 以上の処理を経て、実行単位標識情報ＤＢ２３には、図１５に示す情報が記録される。また、制御部１１０は、以上の処理において実行単位標識検出部１２として機能する。

[0051] 以上の処理が終了すると、制御部１１０は実行単位標識検出処理（ステップＳ２００）を終了する。次に、図１０に示すフローチャートに従い、制御部１１０は類似部分検出処理を開始する（ステップＳ３００）。ここから、制御部１１０が実行する類似部分検出処理について、図１６を参照しながら説明する。

[0052] 制御部１１０は、類似部分検出処理の最初に、一致部分の出現順による類似部分検出処理（ステップＳ３１０）を行い、一致部分情報ＤＢ２１から第１のパターンの類似部分を検出する。一致部分の出現順による類似部分は、複数のタイプ１あるいはタイプ２のコードクローンと、そのコードクローンに挟まれた部分からなるものであり上述の第１のパターンの類似部分に該当する。ここから、図１７を参照し、一致部分の出現順による類似部分検出処理について説明する。

[0053] 制御部１１０は、一致部分の出現順による類似部分検出処理の最初に、一致部分情報ＤＢ２１から一致部分情報を読み出す（ステップＳ３１１）。具体的には、制御部１１０は、一致部分情報ＤＢ２１からコードクローンＩＤ１～４の一致部分情報を読み出す。

[0054] 次に制御部１１０は、読み出した一致部分情報から、複数の一致部分が、間に他の一致部分を挟まずに同じ出現順番でソースプログラムに配置されているものを検出する（ステップＳ３１２）。具体的には、制御部１１０は、読み出したコードクローンＩＤ１～４の一致部分情報から、コードクローン

ＩＤ１および２の一致部分がある部分と、コードクローンＩＤ３および４の一致部分がある部分とは、どちらもグループＩＤ１と２のコードクローンが順に現れる構造を有していることを検出する。制御部１１０は、コードクローンＩＤ１の一致部分から、コードクローンＩＤ２の一致部分までを、類似部分１として検出する。同様に制御部１１０は、コードクローンＩＤ３の一致部分からコードクローンＩＤ４の一致部分までを、類似部分２として検出する。類似部分１は、コードクローンＩＤ１の一致部分の開始位置から開始するため、開始行番号は第０５行となる。また類似部分１は、コードクローンＩＤ２のコードクローンの終了位置で終了するため、終了行番号は第１０行となる。また、類似部分１は、その両端のコードクローンが、それぞれグループＩＤ「１」と「２」を有するため、一致部分リストとして「１、２」を有する。類似部分２は、コードクローンＩＤ３の一致部分の開始位置から開始するため、開始行番号は第１１行となる。また類似部分２は、コードクローンＩＤ４の一致部分の終了位置で終了するため、終了行番号は第１６行となる。また、類似部分２は、その両端のコードクローンが、それぞれグループＩＤ「１」と「２」を有するため、一致部分リストとして「１、２」を有する。

[0055] 以上の処理が終了すると、制御部１１０は一致部分の出現順による類似部分検出処理（ステップＳ３１０）を終了する。次に制御部１１０は、図１６に示すフローに従い、一致部分の前後部分を含む類似部分検出処理を開始する（ステップＳ３２０）。ここから、図１８を参照し、一致部分の前後部分を含む類似部分検出処理について説明する。

[0056] 制御部１１０は、一致部分の前後部分を含む類似部分検出処理の最初に、一致部分情報ＤＢ２１から一致部分情報を読み出す（ステップＳ３２１）。具体的には、制御部１１０は、一致部分情報ＤＢ２１からコードクローンＩＤ１～４の一致部分情報を読み出す。

[0057] 次に制御部１１０は、実行単位標識情報ＤＢ２１から実行単位標識情報を読み出す（ステップＳ３２２）。具体的には、制御部１１０は、一致部分情

報ＤＢ２１から実行単位標識ＩＤ１～２の実行単位標識情報を読み出す。

[0058] 次に制御部１１０は、読み出した一致部分の前部分を検出する（ステップＳ３２３）。一致部分の前部分とは、ソースコード上にある所定の区切り部分からその一致部分の開始行までの部分のことをいう。ここで所定の区切り部分とは、（１）その一致部分直前の他の一致部分の終了行、あるいは（２）その一致部分直前の実行単位の開始あるいは終了を示すコードのある行を指す。制御部１１０は、この（１）（２）のうち後にある方からその一致部分の開始行までの部分を前部分として抽出する。

[0059] 具体的には、制御部１１０は、一致部分情報ＤＢ２１から読み出したコードクローンＩＤ１の一致部分について、（１）その直前には他の一致部分がないこと、および（２）その直前の実行単位の開始あるいは終了を示すコードは、第０１行の「｛」であることを検出する。（１）では他の一致部分を検出できなかったため、制御部１１０は（２）の検出に基づいて前部分を判別する。その結果、制御部１１０はコードクローンＩＤ１のコードクローンの前部分として、第０２行（記号「｛」がある行の次行）から第０４行まで（その一致部分開始行の前行）を検出する。

[0060] 同様に制御部１１０は、コードクローンＩＤ２の一致部分についても前部分を検出する。この場合、制御部１１０は、（１）コードクローンＩＤ２の一致部分の直前の一致部分はコードクローンＩＤ１の一致部分であり、その終了行は第０６行であること、（２）コードクローンＩＤ２の一致部分の直前の実行単位の開始あるいは終了を示すコードは、第０１行の「｛」であることを検出する。この結果、制御部１１０は（１）で検出した第０６行のほうが後にあるため、（１）の検出に基づいて前部分を抽出する。従って、制御部１１０は、コードクローンＩＤ２の一致部分の前部分として、第０７行（直前コードクローンの終了行の次行）から第０７行まで（そのコードクローン開始行の前行）を検出する。

[0061] 続いて制御部１１０は、コードクローンＩＤ３の一致部分についても前部分を検出する。この場合、制御部１１０は、（１）コードクローンＩＤ３の

一致部分の直前の他の一致部分はコードクローンＩＤ２の一致部分であり、その終了行は第１０行であること、（２）コードクローンＩＤ２の一致部分の直前の実行単位の開始あるいは終了を示すコードは、第０１行の「｛」であることを検出する。この結果、制御部１１０は（１）で検出した第１０行のほうが後にあるため、（１）の検出に基づいて前部分を検出する。ここで、制御部１１０は、（１）に基づいて検出した所定の区切り部分が、コードクローンＩＤ３の一致部分の開始行（第１１行）の前行であることを検出する。所定の区切り部分と、コードクローンＩＤ３の一致部分の開始行との間にソースコードが存在しないため、制御部１１０は、コードクローンＩＤ３の一致部分は前部分を持たないと検出する。

[0062] 続いて制御部１１０は、コードクローンＩＤ４の一致部分についても前部分を検出する。上記同様の処理により、制御部１１０は、コードクローンＩＤ４の一致部分の前部分として、第１３行（直前の一致部分の終了行の次行）から第１３行まで（その一致部分開始行の前行）を検出する。

[0063] 次に制御部１１０は、読み出した各一致部分の後部分を検出する（ステップＳ３２４）。一致部分の後部分とは、その一致部分の終了行から、ソースコード上にある所定の区切り部分までの部分のことをいう。ここで所定の区切り部分とは、（１）その一致部分直後の他の一致部分の開始行、あるいは（２）その一致部分直後の実行単位の開始あるいは終了を示すコードのある行を指し、制御部１１０はその一致部分の終了行から（１）（２）のうち前にある方までの部分を後部分として検出する。

[0064] 具体的には、制御部１１０は、一致部分情報ＤＢ２１から読み出したコードクローンＩＤ１のコードクローンについて、（１）コードクローンＩＤ１の一致部分の直後の一致部分はコードクローンＩＤ２の一致部分であり、その開始行は第０８行であること、（２）コードクローンＩＤ１の一致部分の直後の実行単位の開始あるいは終了を示すコードは、第１８行の「｝」であることを検出する。この結果、制御部１１０は（１）で検出した第０８行のほうが前にあるため、（１）の検出に基づいて後部分を抽出する。従って、

制御部１１０は、コードクローンＩＤ１の一致部分の後部分として、第０７行（その一致部分終了行の次行）から第０７行まで（直後にある他の一致部分開始行の前行）を検出する。

[0065] 続いて制御部１１０は、同様の処理によりコードクローンＩＤ２～４の一致部分について後部分を検出する。制御部１１０は、コードクローンＩＤ２の一致部分は後部分を持たないと検出する。また制御部１１０は、コードクローンＩＤ３の一致部分の後部分は第１３行から第１３行まで、コードクローンＩＤ４の一致部分の後部分は第１７行から第１７行までと検出する。

[0066] 次に制御部１１０は、読み出した各一致部分に基づいて、検出した前部分、一致部分、検出した後部分から構成される部分を類似部分として検出する（ステップＳ３２５）。具体的には、制御部１１０は、コードクローンＩＤ１の一致部分に基づいて、前部分（第０２行から第０４行）、一致部分（第０５行から第０６行）、後部分（第０７行から第０７行）を連結し、類似部分３として検出する。類似部分３は、開始行番号は第０２行であり、終了行番号は第０７行である。また、類似部分３は、グループＩＤ「１」を有する一致部分に基づいて検出されており、一致部分リストとして「１」を有する。同様にコードクローンＩＤ２～４に基づき、類似部分４～６を検出する。ただし、コードクローンＩＤ２の一致部分は後部分を有さず、またコードクローンＩＤ３の一致部分は前部分を有さないため、それぞれ一致部分自身の終了行、開始行が類似部分の終了行、開始行となる。この結果、制御部１１０は、類似部分４（第０７行から第１０行まで、一致部分リストは「２」）、類似部分５（第１１行から第１３行まで、一致部分リストは「１」）、類似部分６（第１３行から第１７行まで、一致部分リストは「２」）を検出する。

[0067] 次に制御部１１０は、図１６に示すとおり、ステップＳ３１０およびステップＳ３２０で検出した各類似部分について、識別番号を付与する（ステップＳ３３０）。具体的には、制御部１１０は類似部分（類似部分１～６）に、類似部分ＩＤおよび類似部分グループＩＤを付与する。類似部分ＩＤを付

与する際、制御部１１０は、検出した類似部分間で類似部分ＩＤが重複しないように採番する。本実施例においては、制御部１１０は、類似部分１に類似部分ＩＤ「１」を付与し、以降類似部分６まで順に類似部分ＩＤを付与する。また類似部分グループＩＤを付与する際、制御部１１０は、同一のグループＩＤを有するコードクローンに基づいて、同一のステップ（ステップＳ３１０またはステップＳ３２０）で検出された類似部分は同一の類似部分グループＩＤを有するように採番する。ステップＳ３１０で制御部１１０が検出した類似部分はすべて第１のパターンの類似部分であり、すべて同一のパターンで検出されたものである。同様に、ステップＳ３２０で制御部１１０が検出した類似部分はすべて第２のパターンの類似部分であり、同一のパターンで検出されたものである。具体的には、上記ステップＳ３１０において、類似部分１と２が、同一のグループＩＤ（１および２）を有するコードクローンから検出されているため、制御部１１０は、類似部分１と２に同一の類似部分グループＩＤ「１」を付与する。また上記ステップＳ３２０において、類似部分３と５が、同一のグループＩＤ「１」を有するコードクローンから検出されているため、制御部１１０は、類似部分３と５に同一の類似部分グループＩＤ「２」を付与する。同様に制御部１１０は、類似部分４と６に同一の類似部分グループＩＤ「３」を付与する。

[0068] 次に制御部１１０は、図１６に示すとおり、各類似部分に基づいて類似部分情報を生成し、類似部分情報ＤＢ２４に記録する（ステップＳ３４０）。具体的には、制御部１１０は、ステップＳ３１０で検出した類似部分１に基づいて、類似部分情報（類似部分ＩＤ「１」、類似部分グループＩＤ「１」、ファイル名「src.c」、開始行番号「０５」、終了行番号「１０」、一致部分リストとして「１、２」を記録する。同様に制御部１１０は、類似部分２～６についても類似部分情報ＤＢ２４に記録する。

[0069] 以上の処理を経て、類似部分情報ＤＢ２４には、図１９に示す情報が記録される。また、制御部１１０は、以上の処理において類似部分検出部１３として機能する。

[0070] 以上の処理が終了すると、制御部１１０は図１０に示す類似部分検出処理（ステップＳ３００）を終了する。次に、図１０に示すフローチャートに従い、制御部１１０は非一致構造分析処理（ステップＳ４００）を開始する。ここから、非一致構造分析処理について、図２０を参照しながら説明する。

[0071] 図２０に示す非一致構造分析処理の最初に、制御部１１０は、類似部分情報ＤＢ２４に格納されている類似部分の中から、同じ類似部分グループＩＤを持つ類似部分を取得する（ステップＳ４０１）。以降では、類似部分グループＩＤとして「１」を有する類似部分（類似部分１および２）を例として説明する。

[0072] 次に制御部１１０は、同一の類似部分グループＩＤを有する各類似部分が含む非一致部分を特定し、そのプログラム記述を前記ソースプログラム１から読み出す（ステップＳ４０２）。具体的には、制御部１１０は、類似部分１に関する類似部分情報を類似部分ＤＢ２４から読み出し、開始行番号が第０５行、終了行番号が第１０行であることを読み出す。また制御部１１０は、類似部分１の一致部分リストが「１、２」であることから、ソースプログラム１の第０５行から第１０行までのソースコードのうち、コードクローンＩＤ１のコードクローンに該当する部分（第０５行から第０６行）、およびコードクローンＩＤ２のコードクローンに該当する部分（第０８行から第１０行）、を除いた部分を抽出する。すなわち、制御部１１０は、ソースプログラム１の第０７行を、類似部分１における非一致部分として特定し、抽出する。同様に制御部１１０は、ソースプログラム１の第１３行を、類似部分２における非一致部分として特定し、抽出する。

[0073] 次に制御部１１０は、ステップＳ４０２で特定した非一致部分の構文を基づいて分析し、構文構造を取得する（ステップＳ４０３）。具体的には、制御部１１０は、類似部分１の非一致部分（第０７行）のプログラム記述をソースプログラム１から読み出し、文法定義情報ＤＢ２２に格納されている文法定義情報に基づいて構文を分析する。本例では、上述したＬＲ法によりソースプログラム１の第０７行を構文分析する。ソースプログラム１の第０７

行には、変数（ｃ）、代入演算子（＝）、数値（３０）を順に接続した式が記載されている。この式を構文木構造に基づいて分析することにより、制御部１１０は、図２１Ａに示す構文構造を取得する。また同様に、制御部１１０は、類似部分２の非一致部分（第１３行）より、ソースプログラム１の第１３行を構文解析する。ソースプログラム１の第１３行には、変数（ｃ）、代入演算子（＝）、変数（ａ）、演算子（＋）、変数（ｂ）、演算子（＋）、変数（ｃ）を順に接続した式が記載されている。この式を構文木構造に基づいて分析することにより、制御部１１０は、図２１Ｂに示す構文構造を取得する。

[0074] 次に制御部１１０は、同一類似部分グループＩＤを有する類似部分間で、ステップＳ４０３で取得した構文構造を比較する（ステップＳ４０４）。具体的には、制御部１１０は、同一類似部分ＩＤを有する類似部分１および２の非一致部分の構文構造（図２１Ａ、図２１Ｂ）を比較し、両者が共通に有する構文構造（以下、非一致部分共通構文構造と表記）を抽出する。制御部１１０は、両者の構文構造を構文木の上部から順にたどることにより比較する。制御部１１０は、両者において代入演算子（＝）およびその左辺（変数（ｃ））は共通するが、その右辺は違う構造を有することを識別し、かつ、その右辺は「式」として共通する性質を有すると識別する。その結果、制御部１１０は、図２１Ｃに示すとおり、変数（ｃ）、代入演算子（＝）、式を順に接続した構文構造を非一致部分共通構文構造として抽出する。

[0075] 次に制御部１１０は、各類似部分の非一致部分の構文構造と、ステップＳ４０４で抽出した非一致部分共通構文構造とを比較し、構文類似度を算出する（ステップＳ４０５）。具体的には、制御部１１０は、類似部分１および２の各非一致部分の構文構造と、非一致部分共通構文構造とを比較し、どれだけのノードが一致するかを示す構文類似度を算出する。制御部１１０は、図２１Ａに示す類似部分１の非一致部分の構文構造と図２１Ｃに示す非一致部分共通構文構造とを比較する。その結果、制御部１１０は、非一致部分の構文構造が有する３ノードのうち、代入演算子（＝）のノードと、変数（ｃ

）のノードとは非一致部分共通構文構造と一致し、数値（３０）のノードも非一致部分共通構文構造のノード（「式」）と一致することを検出する。非一致部分の構文構造が有する３ノードのうち、３ノードが一致することから、制御部１１０は、類似部分１の構文類似度を３／３＝１００％と算出する。同様に制御部１１０は、類似部分２の非一致部分の構文構造と非一致部分共通構文構造とを比較する。その結果、制御部１１０は、非一致部分の構文構造が有する７ノードのうち、３ノードが一致することから、制御部１１０は、類似部分１の構文類似度を３／７＝４２．８％と算出する。

[0076] 次に制御部１１０は、非一致構造情報を生成し、非一致構造情報ＤＢ２５に記録する（ステップＳ４０６）。制御部１１０は、上述したステップＳ４０１からステップＳ４０５までの過程で検出した情報から、類似部分ＩＤ、類似部分グループＩＤ、非一致部分の構文構造、非一致部分共通構文構造、構文類似度を抽出し、非一致構造情報として非一致構造情報ＤＢ２５に記録する。制御部１１０は、非一致部分の構文構造および非一致部分共通構文構造を、構文の階層構造を示すバイナリデータとして記録する。制御部１１０は、類似部分１について、類似部分ＩＤ「１」、類似部分グループＩＤ「１」、非一致部分の構文構造の情報（図２１Ａに示す構文の階層構造を示すバイナリデータ）、非一致部分共通構文構造の情報（図２１Ｃに示す構文の階層構造を示すバイナリデータ）、構文類似度「１００％」からなる非一致構造情報を生成し、非一致構造情報ＤＢ２５に記録する。同様に、類似部分２についても、非一致構造情報を生成し、非一致構造情報ＤＢ２５に記録する。この結果、非一致構造情報ＤＢ２５には、図２２に示す非一致構造情報が記録される。

[0077] 制御部１１０は、以上のステップＳ４０１からＳ４０６までの処理を、類似部分３～６についても同様に実施する。その結果、抽出された６つの類似部分に対応する非一致構造情報が、非一致構造情報ＤＢ２５に記録される。

[0078] 以上の処理の結果、非一致構造情報ＤＢ２５に非一致構造情報が記録される。また、制御部１１０は、以上の処理において類似部分検出部１３として

機能する。

[0079] 制御部１１０は、ステップＳ４０６の処理を完了すると、図２０に示す非一致構造分析処理を完了する。制御部１１０は、非一致構造分析処理を完了すると、図１０に示すソースプログラム解析処理を完了する。

[0080] 以上に示したとおり、制御部１１０は、入力されたソースプログラム１にあるタイプ３のコードクローンを、類似部分として非一致構造情報ＤＢ２５に記録する。従って操作者は、非一致構造情報ＤＢ２５のデータを閲覧することにより、ソースプログラム１のどこにタイプ３のコードクローンがあるかを知ることができる。また、制御部１１０は、各類似部分の構文類似度を非一致構造情報ＤＢ２５に記録する。この構文類似度は、プログラムを保守する観点で重要となる構造的、構文的な類似性を数値化したものである。従って、操作者は、構文類似度を参照することにより、プログラムを修正する上での作業量や所要時間を見積もることができる。

[0081] 以上のとおり、本発明によれば、ソースプログラム１からタイプ３のコードクローンを検出し、そのプログラムの修正コストを算出するために有効な情報を出力するソースプログラム解析システム１００を実現できる。上記の処理によって得られた構文類似度を参照することにより、例えばシステムの移行にあたって提案された移行方法の作業量や所要時間を見積もることが可能となり、移行方法を評価することができる。また、既存システムのソースコードのリファクタリングにおいて、必要な箇所を特定し作業量を見積もることができる。

[0082] 以上、本発明の実施形態について説明したが、本発明の実施形態はこれに限定されない。

[0083] 本実施形態では、制御部１１０は一致部分検出処理を行う際、非特許文献２が開示するソフトウェアであるＣＣｆｉｎｄｅｒＸにより一致部分を検出したが、これに限定されない。ソースプログラムから、タイプ１あるいは２のコードクローンを検出することができるソフトウェアであれば、任意のものによって一致部分検出処理を行うことができる。

[0084] 類似部分検出部１３は、ステップＳ３１１において、複数の一致部分が、間に他の一致部分を挟まずに同じ出現順番でソースプログラムに配置されているものを類似部分として検出している。しかし、類似部分検出部１３は、間に他の一致部分を挟んで同じ出現順番でソースプログラムに配置されているものも、類似部分として検出してもよい。このようにした場合、例えば上述の実施形態では、コードクローンＩＤ１の一致部分からコードクローンＩＤ４の一致部分までを、類似部分グループＩＤ１の類似部分として検出する。

[0085] 本実施形態においては、ソースプログラム解析システム１００は、ソースプログラム１を行単位に解析した。しかし、解析する単位は行に限定されず、改行コードとは異なるコードにより区分けされた部分ごとに一致部分を検出してもよい。例えば、単語あるいは文字ごとに解析するようにしてもよい。

[0086] 本実施形態においては、類似部分検出部１３は、類似部分として第１のパターンおよび第２のパターンの類似部分を検出した。これに限られず、類似部分検出部１３は、第１のパターンの類似部分のみを検出してもよいし、第２のパターンの類似部分のみを検出してもよい。また、他のスキームによって類似部分を検出してもよい。

[0087] 本実施形態においては、一致部分検出部１１は一致部分情報を一致部分情報ＤＢ２１に記録し、類似部分検出部１３は、一致部分情報ＤＢ２１から一致部分情報を読み出した。しかし、必ずしも一旦データベースに一致部分情報を記録しなくてもよく、一致部分検出部１１から類似部分検出部１３へ直接的に一致部分情報を供給してもよい。また、本実施形態において一致部分情報ＤＢ２１は記憶部１２０に存在するが、記憶部１２０上に限られず、例えば制御部１１０のＲＡＭに存在するものであってもよいし、またネットワークを介してアクセス可能な他機器に存在するものであってもよい。同様のことは、他のＤＢ（文法定義情報ＤＢ２２、実行単位標識情報ＤＢ２３、類似部分情報ＤＢ２４、非一致構造情報ＤＢ２５）にも当てはまる。すなわち

、各ＤＢが存在する場所は任意であり、また各情報が、当該情報を必要とする機能部あるいは操作者に伝わるのであればＤＢとして存在する必要はない。また、各ＤＢのフォーマットは任意であり、一般的なファイルでも、データベース構造を有するデータセット等でも実現することができる。

[0088] 本発明の特徴は、ソースプログラムから内容が互いに一致する一致部分を検出すること、検出された一致部分に基づいてソースプログラムに存在する類似部分を検出すること、検出された類似部分にある非一致部分の類似性を示す指標を出力することにある。本発明の対象はソースプログラムに限られず、以下のような応用も可能である。すなわち、本発明のソースプログラム解析システムにより、例えば提出された課題の論文等における類似部分とその類似度を取得することができる。これにより、提出者間でのテキストのコピーの有無を判断する材料とすることも可能である。

[0089] 本発明の実施形態にかかるソースプログラム解析システムは、専用のシステムによらず、通常のコンピュータシステムを用いても実現可能である。例えば、コンピュータに、上記動作を実行するためのプログラムを、フレキシブルディスク、ＣＤ－ＲＯＭ（Compact Disk-Read Only Memory）、ＤＶＤ（Digital Versatile Disk）、ＭＯ（Magnet Optical Disk）などのコンピュータ読み取り可能な記憶媒体に記憶して配布し、これをコンピュータシステムにインストールすることにより、上述の処理を実行するソースプログラム解析システムを構成してもよい。さらに、インターネット上のサーバ装置が有するディスク装置等にプログラムを記憶しておき、例えば、搬送波に重畳させて、コンピュータにダウンロード等するものとしてもよい。

[0090] 本発明の実施形態にかかるソースプログラム解析システム１００は、図２３に示すように、一致部分検出部１１、類似部分検出部１３、非一致部分解析部１４から構成されうるものである。一致部分検出部１１は、ソースプログラム１から内容が互いに一致する一致部分を検出する。類似部分検出部１３は、一致部分検出部１１が検出した一致部分に基づいて、ソースプログラム１から内容が互いに一致する部分と一致しない部分とで構成される類似部

分を検出する。非一致部分解析部１４は、類似部分検出部１３が検出した類似部分の一致しない部分の構文を分析し、他の類似部分とどれだけ類似しているかを示す指標を分析結果２として出力する。操作者は、非一致部分解析部１４が生成した分析結果２により、ソースプログラム１にある文の挿入や削除、変更が施されているコードクローンの位置や数、構文類似度を知ることができる。すなわち、操作者は、ソースプログラム１の修正コストを算出するために有効な情報を得ることができる。このように、上記の構成により、本発明の課題は達成され、また、本発明の効果が奏される。

[0091] なお、本発明は、本発明の広義の精神と範囲を逸脱することなく、様々な実施形態および変形が可能とされるものである。また、上述した実施形態は、本発明を説明するためのものであり、本発明の範囲を限定するものではない。つまり、本発明の範囲は、実施形態ではなく、特許請求の範囲によって示される。そして、特許請求の範囲内およびそれと同等の発明の意義の範囲内で施される様々な変形が、本発明の範囲内とみなされる。

[0092] なお、上記の実施例の一部または全部は、以下の付記のようにも記載されうるが、以下には限られない。

[0093] （付記１）

ソースプログラムの中の、内容が互いに一致する一致部分を検出する一致部分検出手段と、

前記一致部分検出手段が検出した一致部分に基づいて、ソースプログラムの中の、内容が互いに一致する部分と一致しない部分とで構成され、かつ、基準に合致する類似部分を検出する類似部分検出手段と、

前記類似部分検出手段が検出した類似部分を分析し、当該類似部分と互いに一致する部分を有する他の類似部分とどれだけ類似しているかを示す指標を出力する非一致部分分析手段と、を備える、

ことを特徴とするソースプログラム解析システム。

[0094] （付記２）

前記類似部分検出手段は、前記一致部分検出手段が検出した一致部分のう

ち、間に他の一致部分を挟まずに同じ出現順番で配置されている複数の一致部分からなる組を検出し、同一の組に含まれる前記複数の一致部分と、当該複数の一致部分に挟まれた一致しない部分とで構成される類似部分を検出する、

ことを特徴とする付記１に記載のソースプログラム解析システム。

[0095] （付記３）

前記非一致部分分析手段は、前記類似部分検出手段が検出した類似部分の一致しない部分の構文が、当該類似部分と互いに一致する部分を有する他の類似部分の一致しない部分の構文とどれだけ類似しているかを示す構文類似度を算出する、

ことを特徴とする付記１または２に記載のソースプログラム解析システム。

[0096] （付記４）

前記非一致部分分析手段は、前記類似部分検出手段が検出した類似部分の一致しない部分を構成要素に分割し、当該構成要素のうち当該類似部分と互いに一致する部分を有する他の類似部分の一致しない部分と共通する構成要素の割合から構文類似度を算出する、

ことを特徴とする付記３に記載のソースプログラム解析システム。

[0097] （付記５）

前記非一致部分分析手段は、前記類似部分検出手段が検出した類似部分の一致しない部分を構成要素に分割し、当該類似部分と互いに一致する部分を有する他の類似部分の一致しない部分と共通する構成要素の構文構造を出力する、

ことを特徴とする付記３または４に記載のソースプログラム解析システム。

[0098] （付記６）

前記非一致部分分析手段は、前記類似部分検出手段が検出した類似部分の一致しない部分を構文木構造に基づいて分析することにより、構成要素ごと

に分割する、

ことを特徴とする付記４または５に記載のソースプログラム解析システム。

[0099] （付記７）

ソースプログラムの中の、実行単位の開始あるいは終了を示す標識を検出する実行単位標識検出手段をさらに備え、

前記類似部分検出手段は、前記一致部分検出手段が検出した一致部分の直前にある実行単位標識検出手段が検出した第１の標識および当該一致部分の直後にある実行単位標識検出手段が検出した第２の標識を検出し、

前記類似部分検出手段は、前記ソースプログラムにおける前記第１の標識の位置と、前記一致部分の直前にある他の一致部分の位置とのうち後にある方から前記一致部分の位置までの部分を前部分として抽出し、かつ、前記一致部分の位置から前記ソースプログラムにおける前記第２の標識の位置と、前記一致部分の直後にある他の一致部分の位置とのうち前にある方までの部分を後部分として抽出し、

前記類似部分検出手段は、前記前部分と、前記一致部分と、前記後部分とを有する類似部分を検出する、

ことを特徴とする付記１から６のいずれか１つに記載のソースプログラム解析システム。

[0100] （付記８）

前記類似部分検出手段は、前記前部分と前記一致部分とを有する類似部分、および前記一致部分と前記後部分とを有する類似部分を検出する、

ことを特徴とする付記７に記載のソースプログラム解析システム。

[0101] （付記９）

ソースプログラムの中の、内容が互いに一致する一致部分を検出する一致部分検出ステップと、

前記一致部分検出ステップで検出した一致部分に基づいて、ソースプログラムの中の内容が互いに一致する部分と一致しない部分とで構成され、かつ

、基準に合致する類似部分を検出する類似部分検出ステップと、

前記類似部分検出ステップで検出した類似部分を分析し、当該類似部分と互いに一致する部分を有する他の類似部分とどれだけ類似しているかを示す指標を出力する非一致部分分析ステップと、を備える、

ことを特徴とするソースプログラム解析方法。

[0102] （付記１０）

コンピュータに、

ソースプログラムの中の、内容が互いに一致する一致部分を検出する一致部分検出機能、

前記一致部分検出機能が検出した一致部分に基づいて、ソースプログラムの中の内容が互いに一致する部分と一致しない部分とで構成される所定の基準を満たす類似部分を検出する類似部分検出機能、

前記類似部分検出機能が検出した類似部分を分析し、当該類似部分と互いに一致する部分を有する他の類似部分とどれだけ類似しているかを示す指標を出力する非一致部分分析機能、

を実現させるプログラムを記録した記録媒体。

[0103] 本発明は、２０１３年３月２９日に出願された日本国特許出願２０１３－７４６２３号に基づく。本明細書中に日本国特許出願２０１３－７４６２３号の明細書、特許請求の範囲、図面全体を参照として取り込むものとする。

符号の説明

[0104] １：ソースプログラム

２：分析結果

１１：一致部分検出部

１２：実行単位標識検出部

１３：類似部分検出部

１４：非一致部分分析部

２１：一致部分情報ＤＢ

２２：文法定義情報ＤＢ

２３：実行単位標識情報ＤＢ

２４：類似部分情報ＤＢ

２５：非一致構造情報ＤＢ

１００：ソースプログラム解析システム

請求の範囲

[請求項1] ソースプログラムの中の、内容が互いに一致する一致部分を検出する一致部分検出手段と、

前記一致部分検出手段が検出した一致部分に基づいて、ソースプログラムの中の、内容が互いに一致する部分と一致しない部分とで構成され、かつ、基準に合致する類似部分を検出する類似部分検出手段と、

前記類似部分検出手段が検出した類似部分を分析し、当該類似部分と互いに一致する部分を有する他の類似部分とどれだけ類似しているかを示す指標を出力する非一致部分分析手段と、を備える、

ことを特徴とするソースプログラム解析システム。

[請求項2] 前記類似部分検出手段は、前記一致部分検出手段が検出した一致部分のうち、間に他の一致部分を挟まずに同じ出現順番で配置されている複数の一致部分からなる組を検出し、同一の組に含まれる前記複数の一致部分と、当該複数の一致部分に挟まれた一致しない部分とで構成される類似部分を検出する、

ことを特徴とする請求項１に記載のソースプログラム解析システム。

[請求項3] 前記非一致部分分析手段は、前記類似部分検出手段が検出した類似部分の一致しない部分の構文が、当該類似部分と互いに一致する部分を有する他の類似部分の一致しない部分の構文とどれだけ類似しているかを示す構文類似度を算出する、

ことを特徴とする請求項１または２に記載のソースプログラム解析システム。

[請求項4] 前記非一致部分分析手段は、前記類似部分検出手段が検出した類似部分の一致しない部分を構成要素に分割し、当該構成要素のうち当該類似部分と互いに一致する部分を有する他の類似部分の一致しない部分と共通する構成要素の割合から構文類似度を算出する、

ことを特徴とする請求項３に記載のソースプログラム解析システム。

[請求項5] 前記非一致部分分析手段は、前記類似部分検出手段が検出した類似部分の一致しない部分を構成要素に分割し、当該類似部分と互いに一致する部分を有する他の類似部分の一致しない部分と共通する構成要素の構文構造を出力する、

ことを特徴とする請求項３または４に記載のソースプログラム解析システム。

[請求項6] 前記非一致部分分析手段は、前記類似部分検出手段が検出した類似部分の一致しない部分を構文木構造に基づいて分析することにより、構成要素ごとに分割する、

ことを特徴とする請求項４または５に記載のソースプログラム解析システム。

[請求項7] ソースプログラムの中の、実行単位の開始あるいは終了を示す標識を検出する実行単位標識検出手段をさらに備え、

前記類似部分検出手段は、前記一致部分検出手段が検出した一致部分の直前にある実行単位標識検出手段が検出した第１の標識および当該一致部分の直後にある実行単位標識検出手段が検出した第２の標識を検出し、

前記類似部分検出手段は、前記ソースプログラムにおける前記第１の標識の位置と、前記一致部分の直前にある他の一致部分の位置とのうち後にある方から前記一致部分の位置までの部分を前部分として抽出し、かつ、前記一致部分の位置から前記ソースプログラムにおける前記第２の標識の位置と、前記一致部分の直後にある他の一致部分の位置とのうち前にある方までの部分を後部分として抽出し、

前記類似部分検出手段は、前記前部分と、前記一致部分と、前記後部分とを有する類似部分を検出する、

ことを特徴とする請求項１から６のいずれか１項に記載のソースプ

ログラム解析システム。

[請求項8] 前記類似部分検出手段は、前記前部分と前記一致部分とを有する類似部分、および前記一致部分と前記後部分とを有する類似部分を検出する、

ことを特徴とする請求項７に記載のソースプログラム解析システム。

[請求項9] ソースプログラムの中の、内容が互いに一致する一致部分を検出する一致部分検出ステップと、

前記一致部分検出ステップで検出した一致部分に基づいて、ソースプログラムの中の内容が互いに一致する部分と一致しない部分とで構成され、かつ、基準に合致する類似部分を検出する類似部分検出ステップと、

前記類似部分検出ステップで検出した類似部分を分析し、当該類似部分と互いに一致する部分を有する他の類似部分とどれだけ類似しているかを示す指標を出力する非一致部分分析ステップと、を備える、

ことを特徴とするソースプログラム解析方法。

[請求項10] コンピュータに、

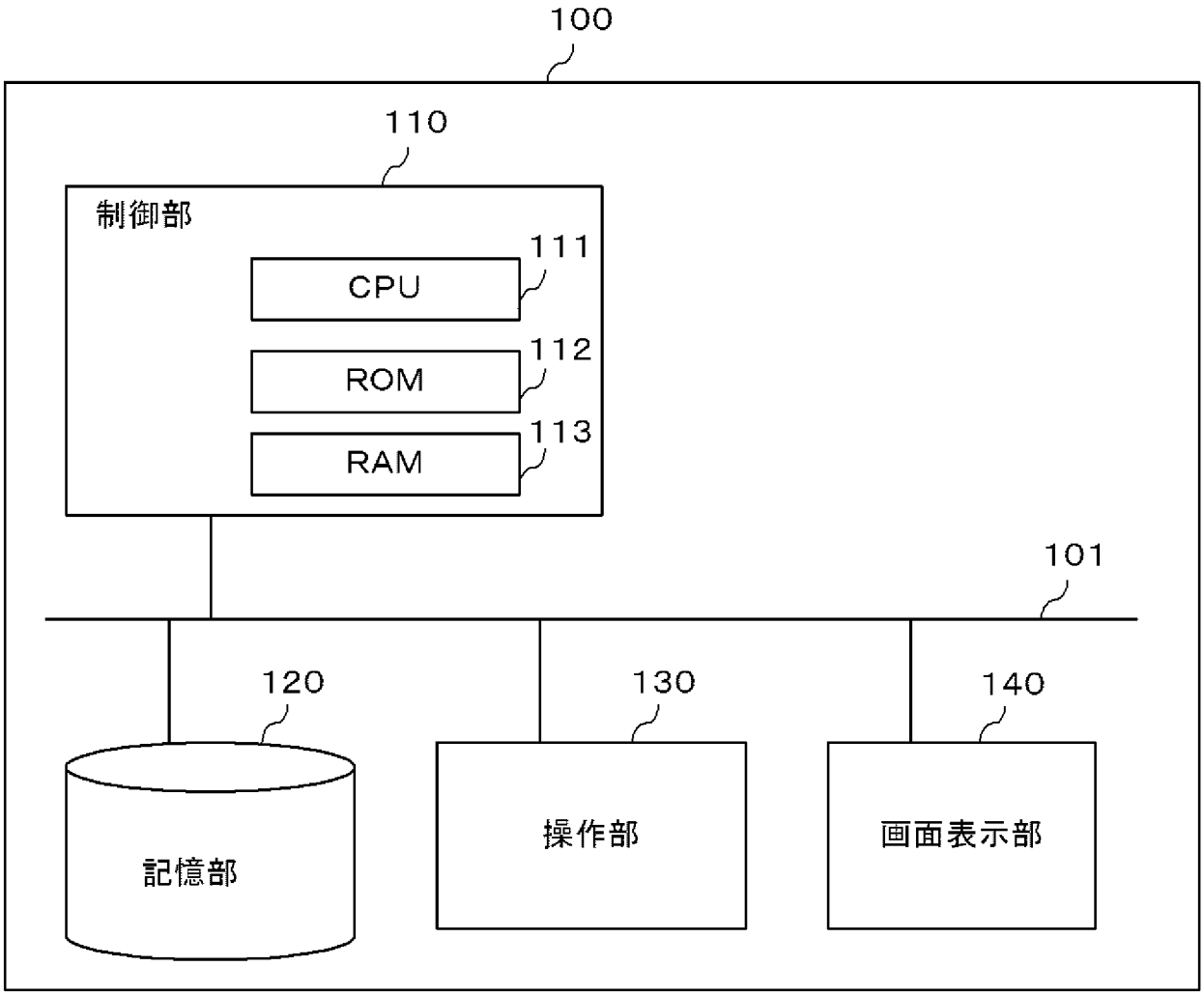
ソースプログラムの中の、内容が互いに一致する一致部分を検出する一致部分検出機能、

前記一致部分検出機能が検出した一致部分に基づいて、ソースプログラムの中の内容が互いに一致する部分と一致しない部分とで構成される所定の基準を満たす類似部分を検出する類似部分検出機能、

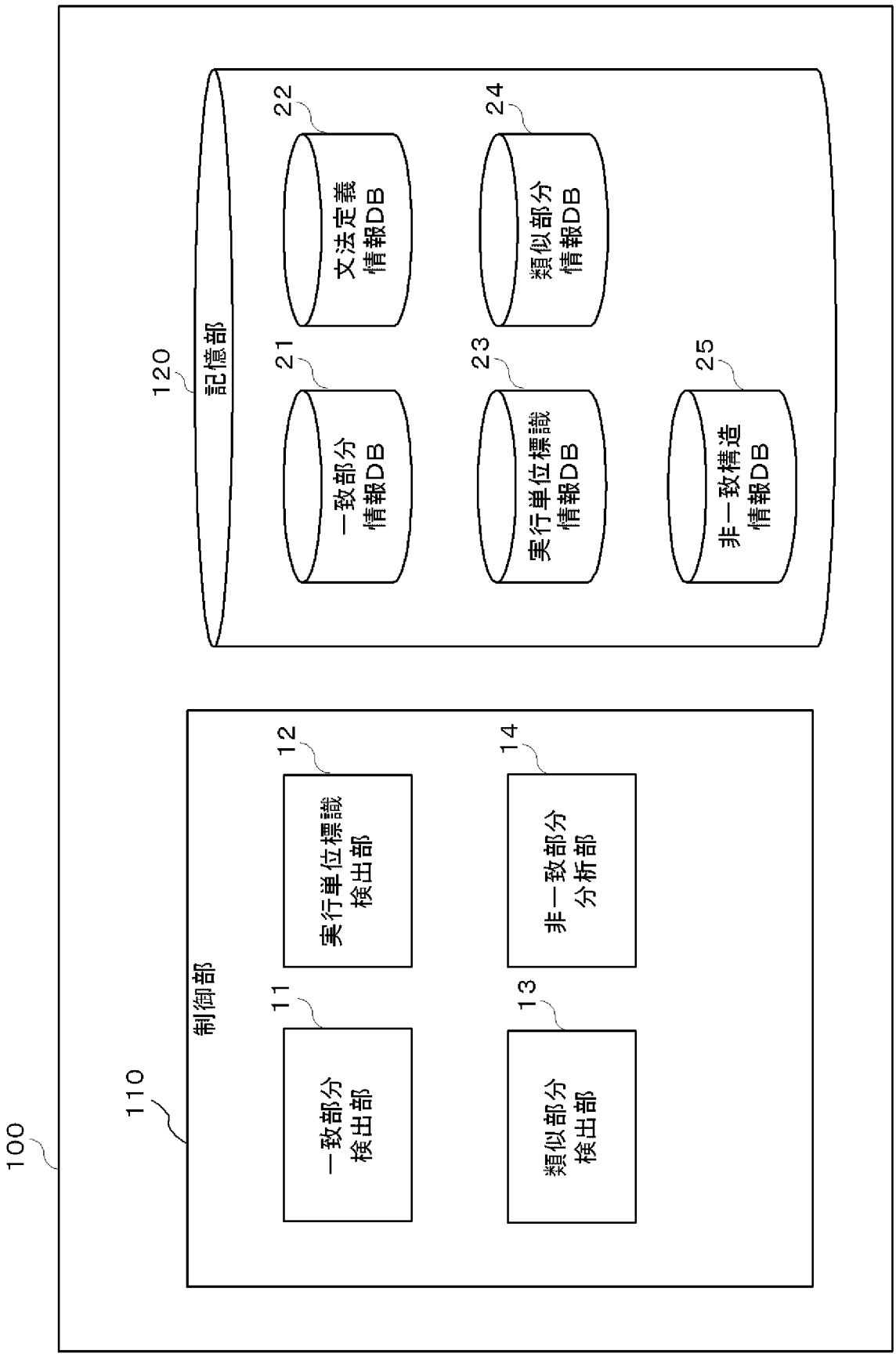
前記類似部分検出機能が検出した類似部分を分析し、当該類似部分と互いに一致する部分を有する他の類似部分とどれだけ類似しているかを示す指標を出力する非一致部分分析機能、

を実現させるプログラムを記録した記録媒体。

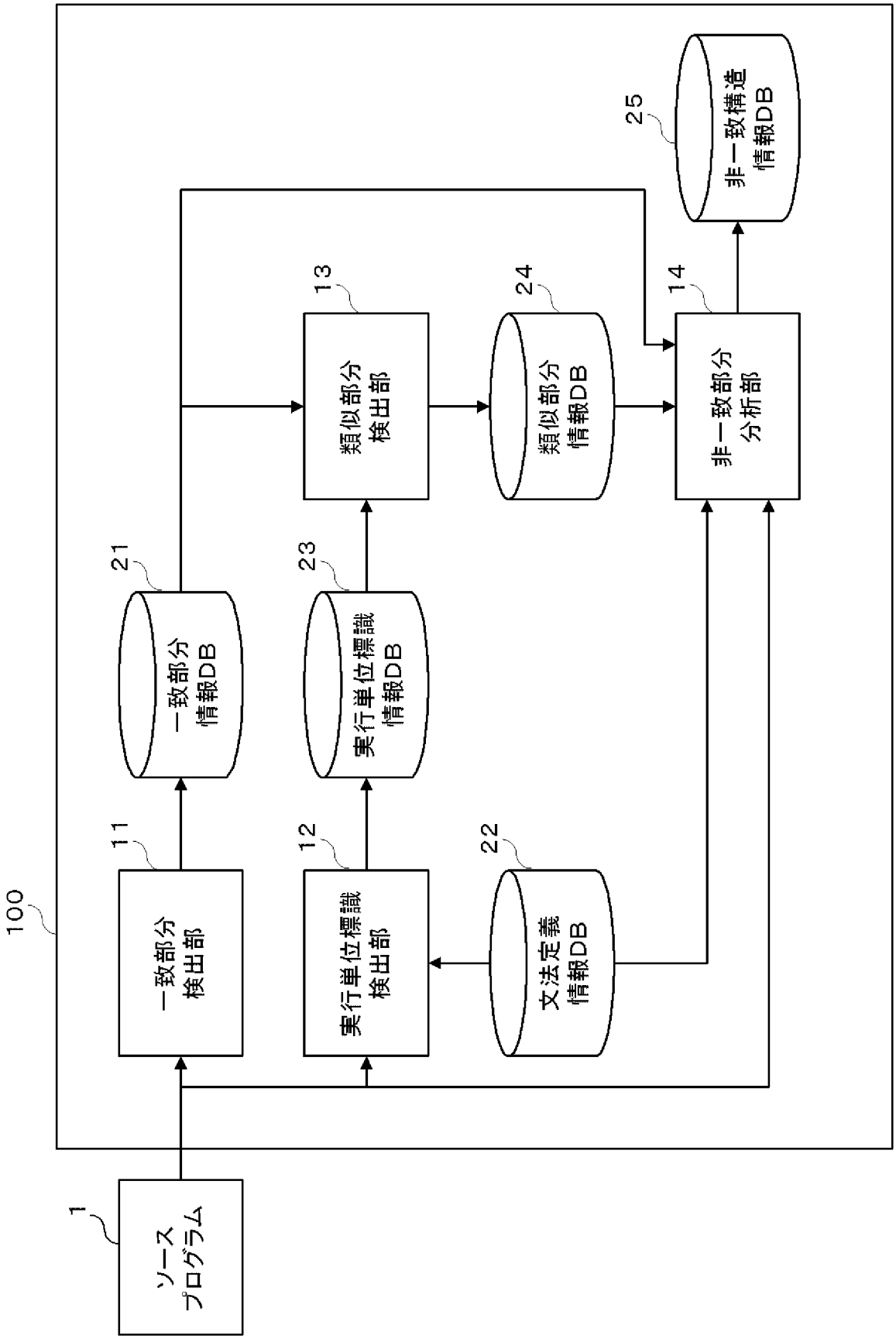
[図1]


100
110
制御部
111
CPU
112
ROM
113
RAM
101
120
130
140
記憶部
操作部
画面表示部

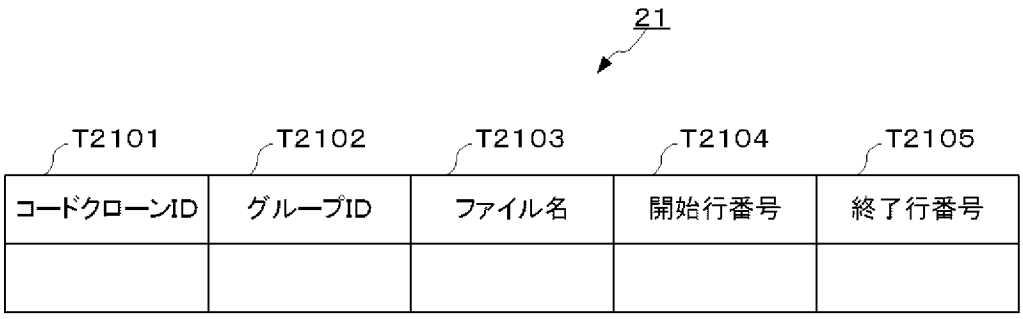
[図2]


100
110
制御部
一致部分
検出部
11
実行単位標識
検出部
12
類似部分
検出部
13
非一致部分
分析部
14
120
記憶部
一致部分
情報DB
21
文法定義
情報DB
22
実行単位標識
情報DB
23
類似部分
情報DB
24
非一致構造
情報DB
25

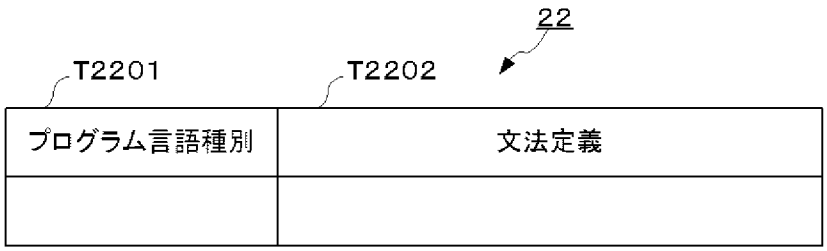
[図3]


100
1
ソース
プログラム
11
一致部分
検出部
12
実行単位標識
検出部
13
類似部分
検出部
14
非一致部分
分析部
21
一致部分
情報DB
22
文法定義
情報DB
23
実行単位標識
情報DB
24
類似部分
情報DB
25
非一致構造
情報DB

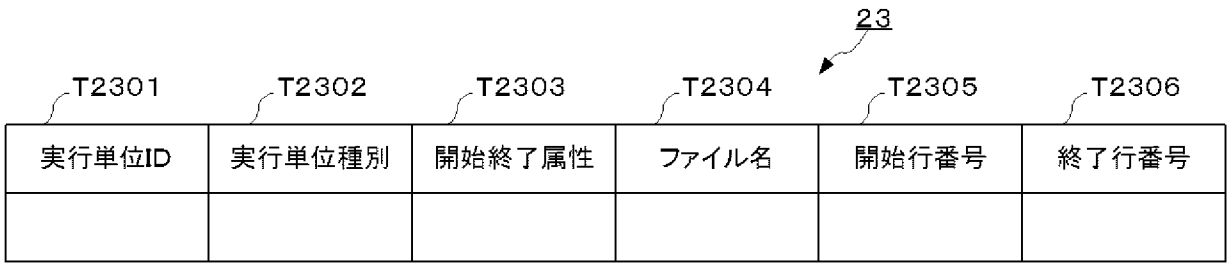
[図4]


21
T2101
T2102
T2103
T2104
T2105
コードクローンID
グループID
ファイル名
開始行番号
終了行番号

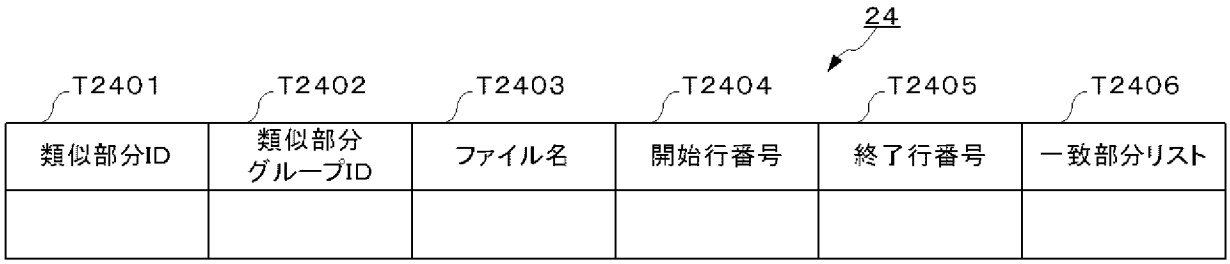
[図5]


22
T2201
T2202
プログラム言語種別
文法定義

[図6]


23
T2301
T2302
T2303
T2304
T2305
T2306
実行単位ID
実行単位種別
開始終了属性
ファイル名
開始行番号
終了行番号

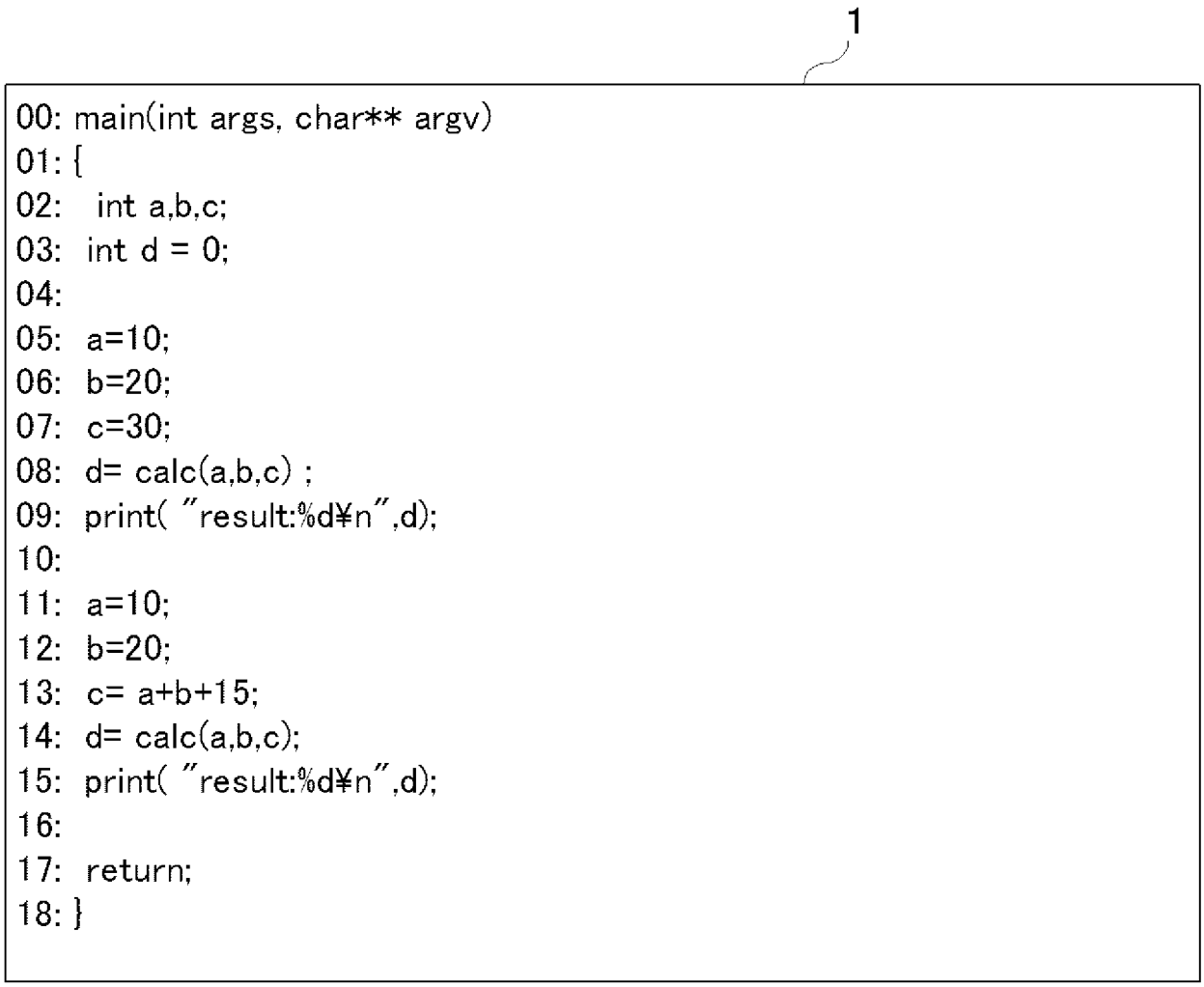
[図7]


24
T2401
T2402
T2403
T2404
T2405
T2406
類似部分ID
類似部分
グループID
ファイル名
開始行番号
終了行番号
一致部分リスト

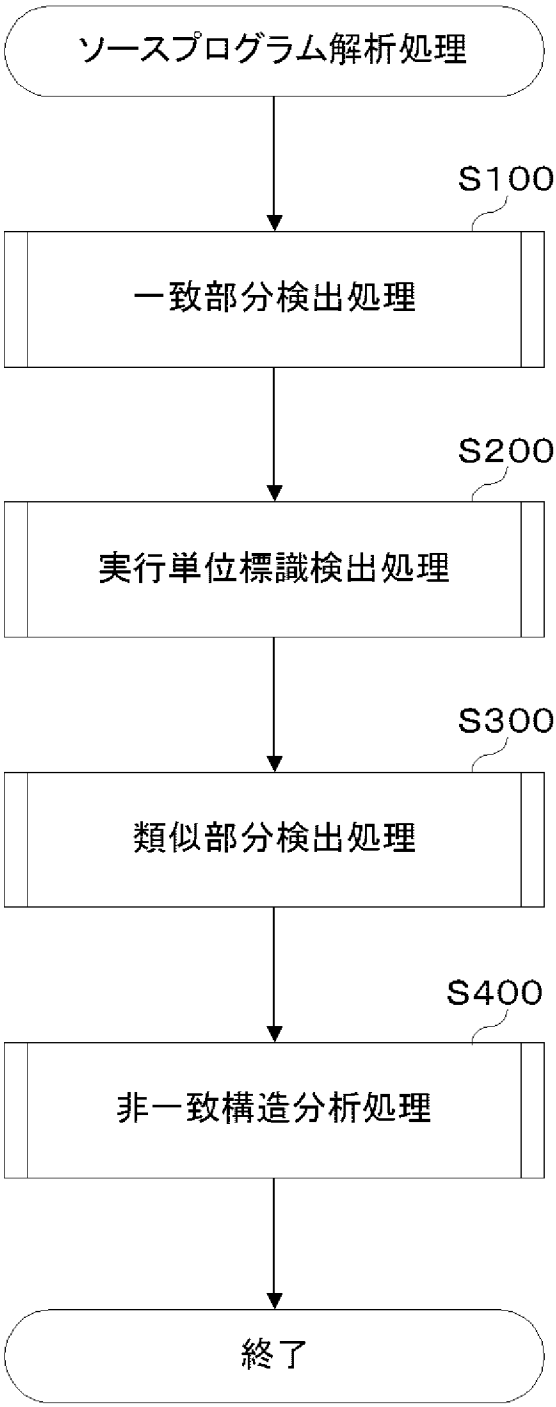
[図8]


25
T2501
T2502
T2503
T2504
T2505
類似部分ID
類似部分
グループID
非一致部分
構文構造
非一致部分
共通構文構造
非一致部分
構文類似度

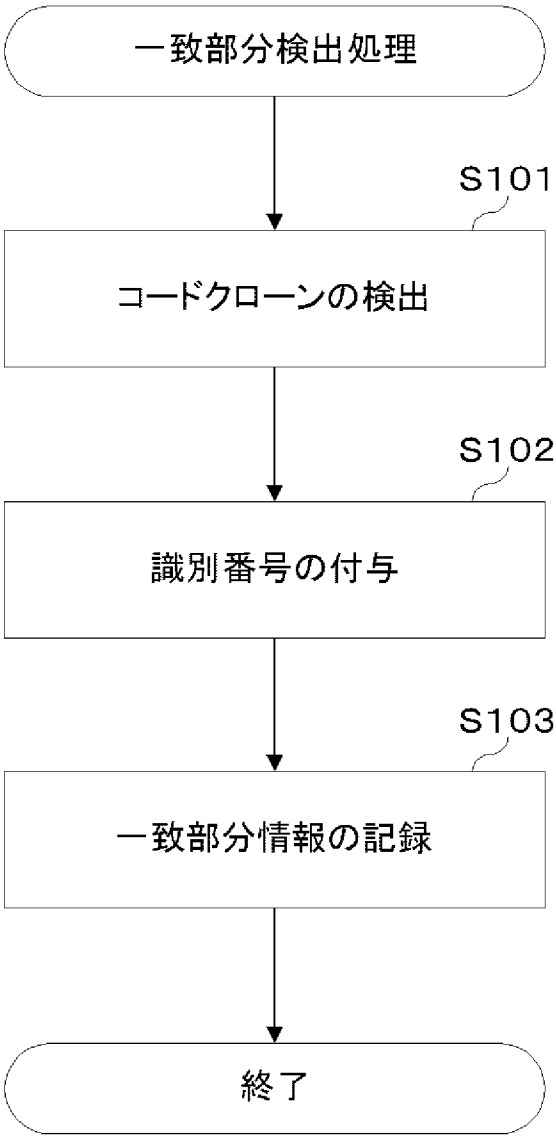
[図9]


1
00: main(int args, char** argv)
01: {
02: int a,b,c;
03: int d = 0;
04:
05: a=10;
06: b=20;
07: c=30;
08: d= calc(a,b,c) ;
09: print("result:%d¥n",d);
10:
11: a=10;
12: b=20;
13: c= a+b+15;
14: d= calc(a,b,c);
15: print("result:%d¥n",d);
16:
17: return;
18: }

[図10]


ソースプログラム解析処理
S100
一致部分検出処理
S200
実行単位標識検出処理
S300
類似部分検出処理
S400
非一致構造分析処理
終了

[図11]

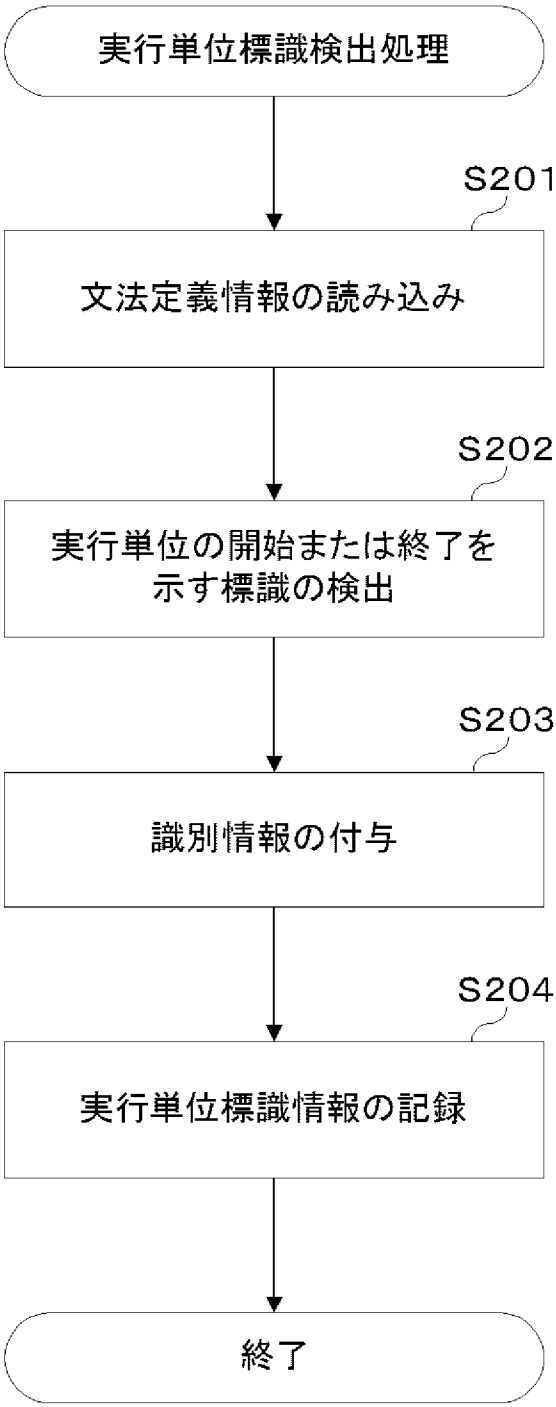


[図12]

21

T2101 コードクローンID	T2102 グループID	T2103 ファイル名	T2104 開始行番号	T2105 終了行番号
1	1	src.c	05	06
2	2	src.c	08	10
3	1	src.c	11	12
4	2	src.c	14	16

[図13]



[図14]

22

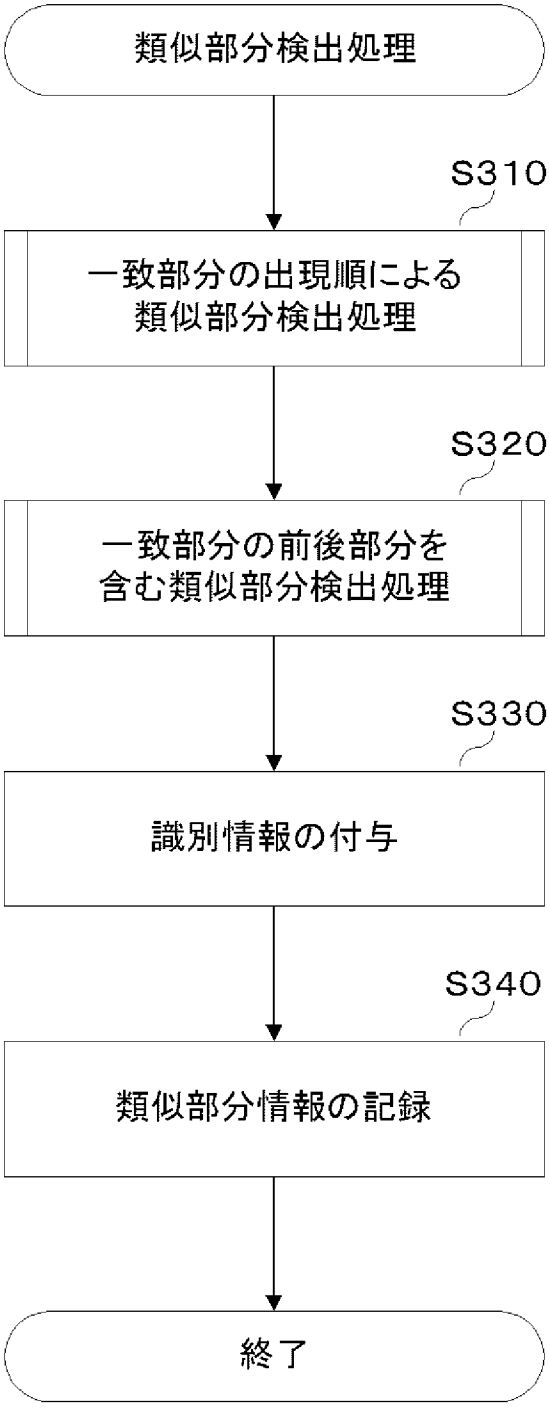
T2201 プログラム言語種別	T2202 文法定義
.c	関数の開始: { 関数の終了: } 変数: [a-z]* 数値: [0-9]* 演算子: + | - | * | / 代入演算子: = 式: 変数|数値 （演算子 変数|数値|式） 代入: 変数 代入演算子 式

[図15]

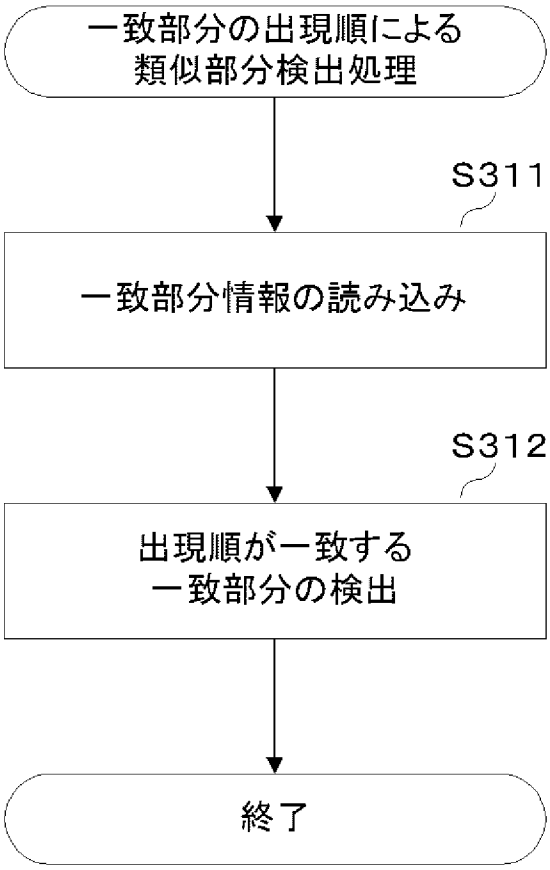
23

T2301 実行単位ID	T2302 実行単位種別	T2303 開始終了属性	T2304 ファイル名	T2305 開始行番号	T2306 終了行番号
1	関数	開始	src.c	01	01
2	関数	終了	src.c	18	18

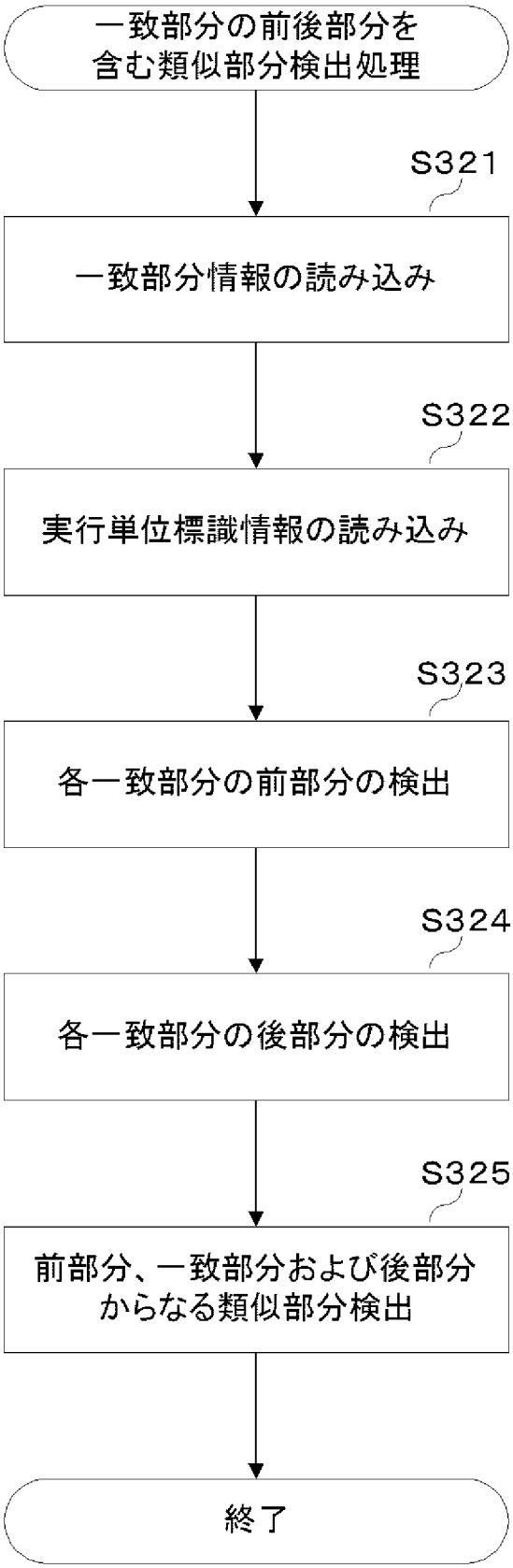
[図16]



[図17]


一致部分の出現順による
類似部分検出処理
S311
一致部分情報の読み込み
S312
出現順が一致する
一致部分の検出
終了

[図18]

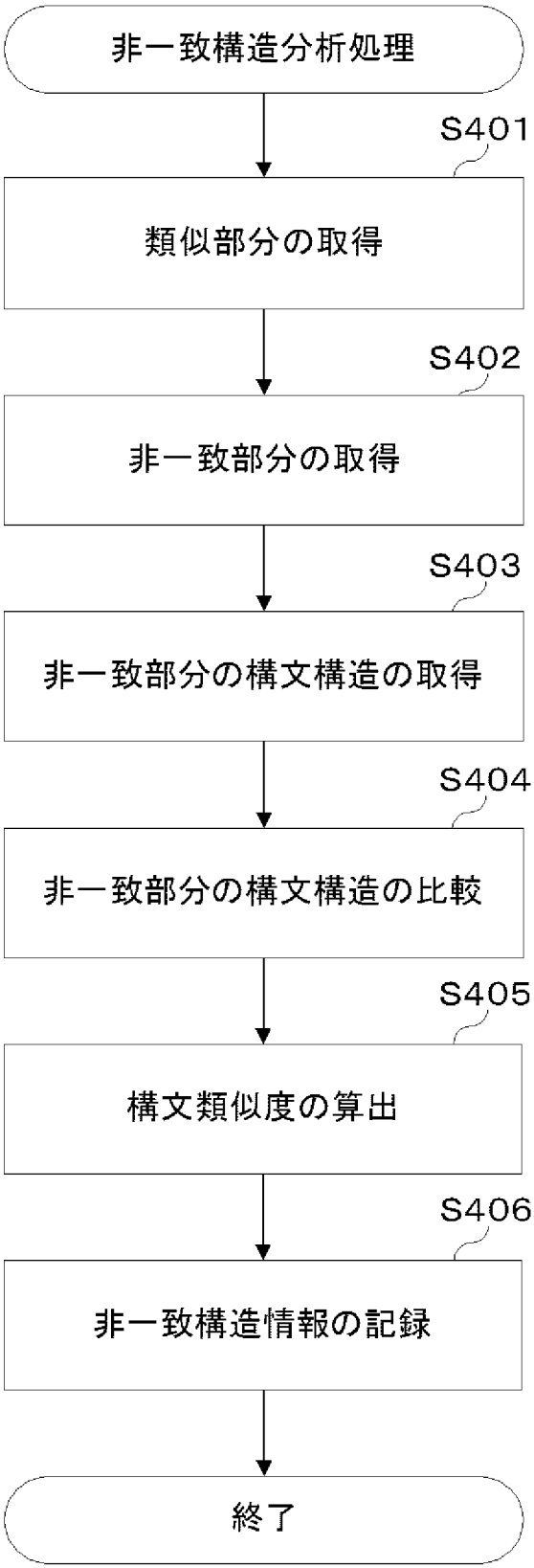

一致部分の前後部分を
含む類似部分検出処理
S321
一致部分情報の読み込み
S322
実行単位標識情報の読み込み
S323
各一致部分の前部分の検出
S324
各一致部分の後部分の検出
S325
前部分、一致部分および後部分
からなる類似部分検出
終了

[図19]

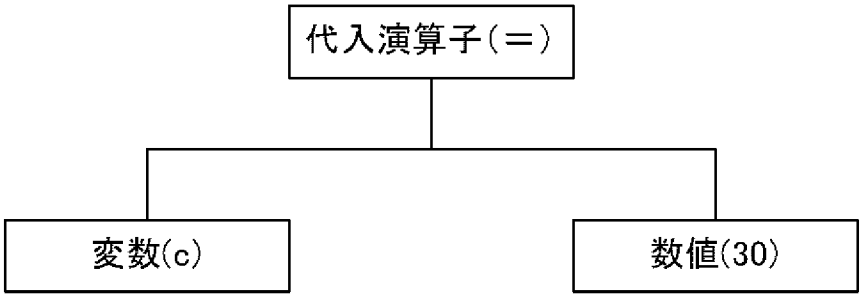
24

T2401	T2402	T2403	T2404	T2405	T2406
類似部分ID	類似部分グループID	ファイル名	開始行番号	終了行番号	一致部分リスト
1	1	src.c	05	10	1,2
2	1	src.c	11	16	3,4
3	2	src.c	02	07	1
4	3	src.c	07	10	2
5	2	src.c	11	13	3
6	3	src.c	13	16	4

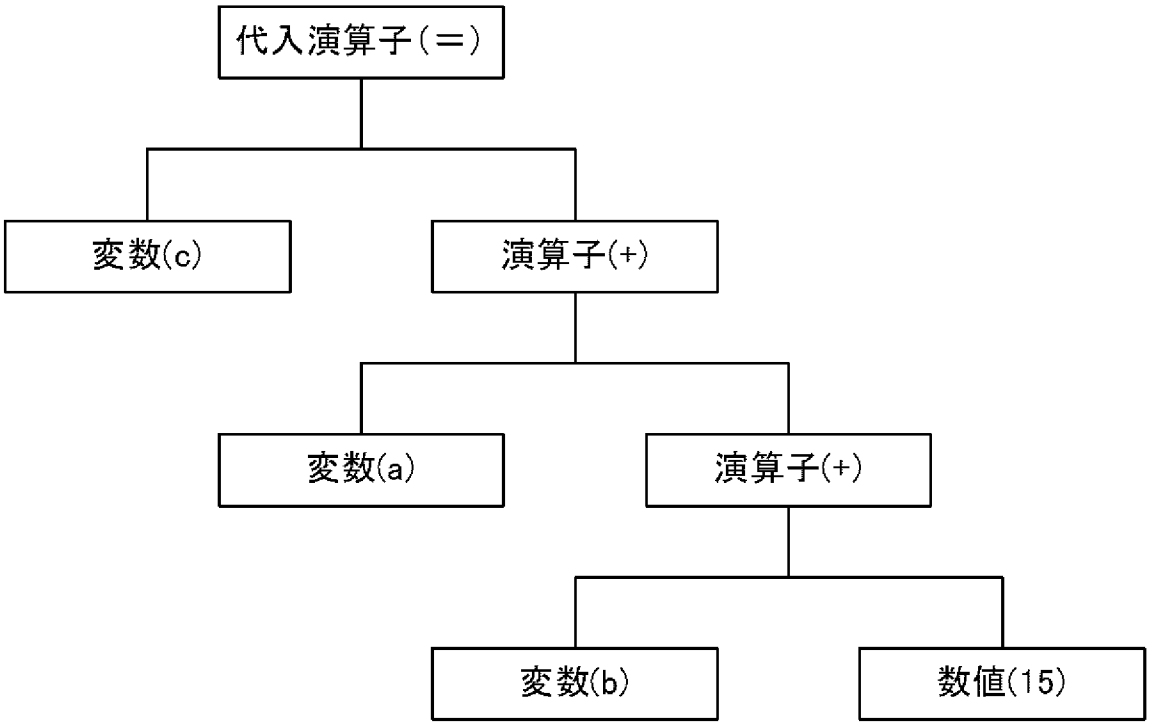
[図20]


非一致構造分析処理
S401
類似部分の取得
S402
非一致部分の取得
S403
非一致部分の構文構造の取得
S404
非一致部分の構文構造の比較
S405
構文類似度の算出
S406
非一致構造情報の記録
終了

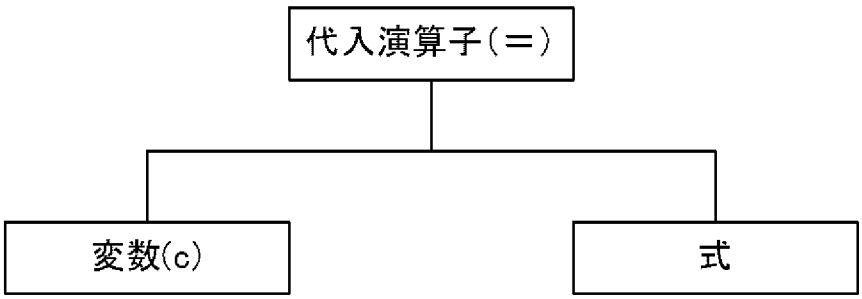
[図21A]



[図21B]



[図21C]

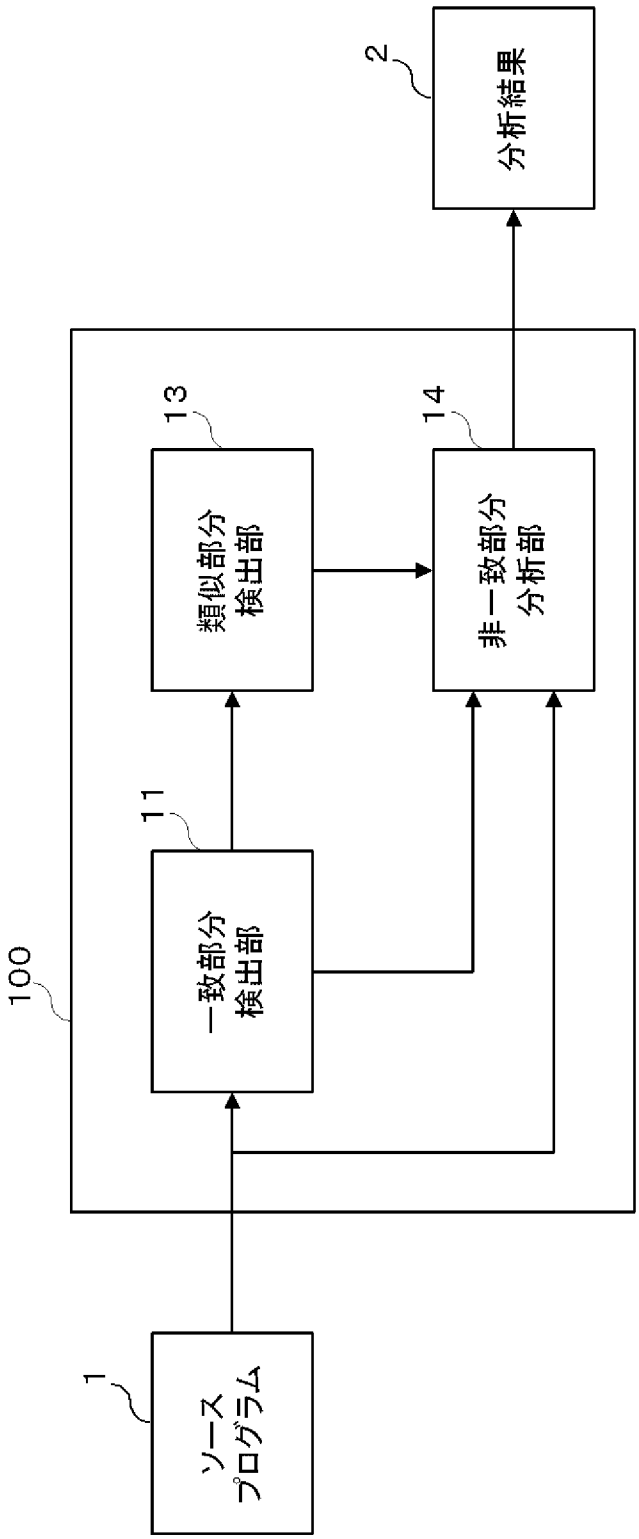


[図22]

25

T2501 類似部分ID	T2502 類似部分グループID	T2503 非一致部分構文構造	T2504 非一致部分共通構文構造	T2505 非一致部分構文類似度
1	1	図21Aに対応するデータ	図21Cに対応するデータ	100%
2	1	図21Bに対応するデータ	図21Cに対応するデータ	42.8%

[図23]



INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2014/058021

A. CLASSIFICATION OF SUBJECT MATTER
G06F9/44(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F9/44, G06F17/30

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2014
Kokai Jitsuyo Shinan Koho 1971-2014 Toroku Jitsuyo Shinan Koho 1994-2014

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	Yasushi UEDA et al. 'On Detection of Gapped Code Clones using Gap Locations' Proceedings of the Ninth Asia-Pacific Software Engineering Conference (APSEC'02), 2002.12.06, pp.327-336, particularly, passages 2 to 3.1	1-2,9-10 3-8
Y A	JP 2003-216425 A (Japan Science and Technology Corp.), 31 July 2003 (31.07.2003), paragraphs [0006] to [0010] (Family: none)	1-2,9-10 3-8
A	JP 2006-18693 A (Fujitsu Ltd.), 19 January 2006 (19.01.2006), paragraphs [0043], [0064] to [0078] & US 2006/0004528 A1	1-10

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26 May, 2014 (26.05.14)	10 June, 2014 (10.06.14)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2014/058021

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Yoshiki HIGO et al., "A Survey of Code Clone Detection and Its Related Techniques", The Transactions of the Institute of Electronics, Information and Communication Engineers, 01 June 2008 (01.06.2008), vol.J91-D, no.6, pages 1465 to 1481, particularly, passage 3	1-10
A	US 2011/0246968 A1 (Dongmei ZHANG, et al.), 06 October 2011 (06.10.2011), abstract; paragraphs [0033] to [0051] (Family: none)	1-10

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２０１４／０５８０２１

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　G06F9/44(2006.01)i

Ｂ．　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int.Cl.　G06F9/44, G06F17/30

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２０１４年
日本国実用新案登録公報	１９９６－２０１４年
日本国登録実用新案公報	１９９４－２０１４年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

Ｃ．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求項の番号
Y A	Yasushi UEDA et al. 'On Detection of Gapped Code Clones using Gap Locations' Proceedings of the Ninth Asia-Pacific Software Engineering Conference (APSEC'02), 2002.12.06, pp.327-336 特に、第２節－第３．１節の記載を参照。	1-2, 9-10 3-8
Y A	JP 2003-216425 A（科学技術振興事業団）2003.07.31, 段落【０００６】－【００１０】（ファミリーなし）	1-2, 9-10 3-8

☑　Ｃ欄の続きにも文献が列挙されている。

☐　パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー

「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの

「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「Ｏ」口頭による開示、使用、展示等に言及する文献

「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献

「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「＆」同一パテントファミリー文献

国際調査を完了した日	国際調査報告の発送日
２６．０５．２０１４	１０．０６．２０１４

国際調査機関の名称及びあて先
日本国特許庁（ＩＳＡ／ＪＰ）
郵便番号１００－８９１５
東京都千代田区霞が関三丁目４番３号

特許庁審査官（権限のある職員）　５Ｂ　４０５５
稲垣　良一
電話番号　０３－３５８１－１１０１　内線　３５４４

Ｃ（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求項の番号
A	JP 2006-18693 A (富士通株式会社) 2006.01.19, 段落【００４３】,【００６４】－【００７８】 & US 2006/0004528 A1	1-10
A	肥後 芳樹 外２名, コードクローン検出とその関連技術, 電子情報通信学会論文誌, 2008.06.01, Vol.J91-D, No.6, pp.1465-1481 特に、第３節の記載を参照。	1-10
A	US 2011/0246968 A1 (Dongmei ZHANG, et al.) 2011.10.06, Abstract, [0033]-[0051] (ファミリーなし)	1-10