

(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
2022年1月27日(27.01.2022)



(10) 国際公開番号

WO 2022/019124 A1

(51) 国際特許分類:
G06F 11/36 (2006.01)

(21) 国際出願番号: PCT/JP2021/025689

(22) 国際出願日: 2021年7月7日(07.07.2021)

(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 2020-124296 2020年7月21日(21.07.2020) JP

(71) 出願人: 日立 Astemo 株式会社(HITACHI
ASTEMO, LTD.) [JP/JP]; 〒3128503 茨城県ひ
たちなか市高場 2 5 2 0 番地 Ibaraki (JP).

(72) 発明者: ロペス ハイロ (LOPEZ, Jairo);
〒1008280 東京都千代田区丸の内一丁目 6 番 6

号 株式会社日立製作所内 Tokyo (JP). 長野 岳彦
(NAGANO, Takehiko); 〒1008280 東京都千代田
区丸の内一丁目 6 番 6 号 株式会社日立製作所内
Tokyo (JP). 芹沢 一(SERIZAWA, Kazuyoshi);
〒3128503 茨城県ひたちなか市高場 2 5 2 0 番
地 日立 Astemo 株式会社内 Ibaraki (JP).

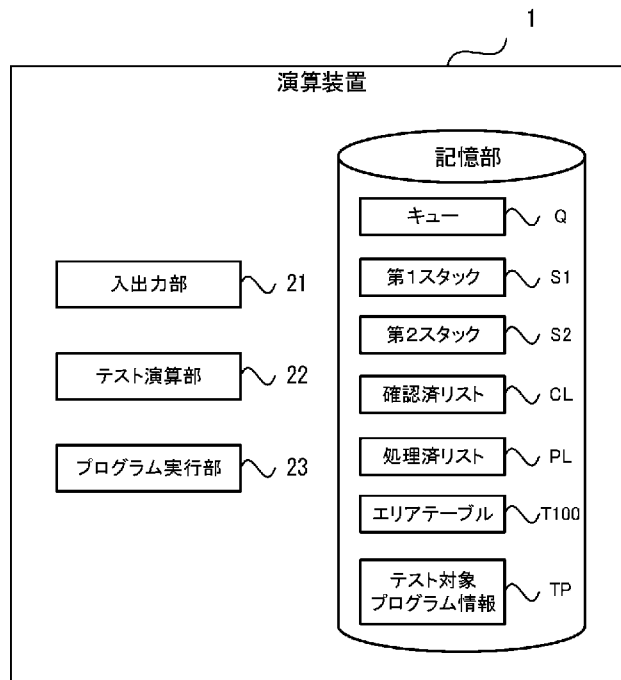
(74) 代理人: 特許業務法人サンネクスト国際特許
事務所(SUNNEXT INTERNATIONAL PATENT
OFFICE); 〒1400002 東京都品川区東品川二丁
目 3 番 1 2 号 シーフォートスクエア センタ
ービルディング 1 6 階 Tokyo (JP).

(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, DJ, DK, DM, DO,

(54) Title: TEST METHOD AND COMPUTATION DEVICE

(54) 発明の名称: テスト方法、演算装置

図2



1 Computation device
21 Input/output unit
22 Test computation unit
23 Program execution unit
CL Confirmed list
PL Processed list
Q Cue
S1 First stack
S2 Second stack
T100 Area table
TP Test target program
information

(57) Abstract: A test method according to the present invention is used by a computer to test a program including a plurality of macro tasks for which there can be many orders of execution. The test method includes: setting any of the plurality of macro tasks as a focus task; specifying an ancestor list which is a collection of the macro tasks which affect the focus task; separating the macro tasks included in the ancestor list into a nonparallel group which does not contribute to parallelization and a parallel contribution group which contributes to parallelization; generating a plurality of test patterns

DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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, ST, 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:

- 一 国際調査報告 (条約第21条(3))

for the parallel contribution group; and executing the plurality of test patterns by using results obtained by executing the macro tasks belonging to the nonparallel group.

(57) 要約: 実行される順番が複数とおり存在し得る複数のマクロタスクを含むプログラムをコンピュータがテストするテスト方法であって、複数のマクロタスクのいずれかを注目タスクに設定することと、注目タスクに影響を与えるマクロタスクの集合である先祖リストを特定することと、先祖リストに含まれるマクロタスクを、並列化に寄与しない非並列群および並列化に寄与する並列寄与群に分別することと、並列寄与群についてテストパターンを複数生成することと、非並列群に属するマクロタスクを実行して得られた結果を用いて、複数のテストパターンを実行することと、を含むテスト方法。

明 細 書

発明の名称： テスト方法、演算装置

技術分野

[0001] 本発明は、テスト方法および演算装置に関する。

背景技術

[0002] 複雑なプログラムは一般に検証が困難である。ソフトウェアへの要求レベルが高い分野では、単体テストレベルで100%のテストカバレッジを取得する必要がある。プログラム内の処理の順番が常に一定である順次プログラムでは、テスト時に相互依存が問題になることはほとんどなく、テストの困難性は限定的である。その一方で、プログラム内の処理の順番が一定ではない非順次プログラムではテストが困難である。

[0003] プログラムの規模が大きいシステムや高速な処理が必要なシステム、たとえば、自動車の高度運転支援システム（ADAS）や自動運転システム（ADS）などのリアルタイムシステムでは、複数のプロセッサが用いられる。また、従来のシングルコア用システムがマルチコアのハードウェア上に構築された場合には、自動並列化処理により複数のシステムを統合することができる。このどちらの場合にも、プログラム内の処理が実行される順番は膨大なため、安全に統合することは容易ではない。従来のシングルコアソフトウェアに対して自動並列化コンパイラを適用すると、並列化されたコードが元のコードと同じ機能と動作を持っていることを実際に確認できるかどうかという別の問題も発生する。

[0004] コンパイラにより正しく実行されたことの確認を行うというアイデアは、特にプログラム言語からアセンブリコードへの変換のレベルで開発された。このレベルでは、プログラム依存グラフという名前で有向非巡回グラフを使用することで、バイナリコードの効率化と自動並列化の向上に大きく貢献している。

[0005] 特許文献1には、複数のテストノード及び該テストノード間のプロセスを

定義したテストシナリオに関する情報を記憶するシナリオ情報記憶手段と、テストノードごとに、次に遷移すべきテストノードを識別する情報を含むテストノードに関する情報、プロセスが分岐する分岐パスを含むか否かに関する情報を記憶するテストノード情報記憶手段と、テストノードごとに前記分岐パスが存在するか否かを判断する第一の判断手段と、該第一の判断手段で前記分岐パスが存在すると判断したテストノードにおいて、スナップショットを取得した場合にテストを完了するまでに要する実行予測時間を算出する実行予測時間算出手段と、スナップショットを取得することなくテストを完了するまでに要する実行時間を算出する実行時間算出手段と、スナップショットを取得したテストノードごとに、実行時間と実行予測時間との差分を短縮時間として算出する短縮時間算出手段と、前記短縮時間が最も大きいテストノードを選択する選択手段と、選択したテストノードを識別する情報を出力する出力手段とを備えるテスト支援装置が開示されている。

先行技術文献

特許文献

[0006] 特許文献1：日本国特開2011-129028号公報

発明の概要

発明が解決しようとする課題

[0007] 特許文献1に記載されている発明では、実行される順番が複数とおり存在し得る複数のマクロタスクを含むプログラムのテストにおいて特定のマクロタスクを対象としたテストが効率的に実行できない。

課題を解決するための手段

[0008] 本発明の第1の態様によるテスト方法は、実行される順番が複数とおり存在し得る複数のマクロタスクを含むプログラムをコンピュータがテストするテスト方法であって、前記複数のマクロタスクのいずれかを注目タスクに設定することと、前記注目タスクに影響を与える前記マクロタスクの集合である先祖リストを特定することと、前記先祖リストに含まれる前記マクロタス

クを、並列化に寄与しない非並列群および並列化に寄与する並列寄与群に分別することと、前記並列寄与群についてテストパターンを複数生成することと、前記非並列群に属する前記マクロタスクを実行して得られた結果を用いて、前記複数の前記テストパターンを実行することと、を含む。

本発明の第2の態様による演算装置は、実行される順番が複数とおり存在し得る複数のマクロタスクを含むプログラムを格納する記憶部と、前記複数のマクロタスクのいずれかを注目タスクに設定し、前記注目タスクに影響を与える前記マクロタスクの集合である先祖リストを特定し、前記先祖リストに含まれる前記マクロタスクを、並列化に寄与しない非並列群および並列化に寄与する並列寄与群に分別し、前記並列寄与群についてテストパターンを複数生成するテスト演算部と、前記非並列群に属する前記マクロタスクを実行して得られた結果を用いて、前記複数の前記テストパターンを実行するプログラム実行部と、を含む。

発明の効果

[0009] 本発明によれば、実行される順番が複数とおり存在し得る複数のマクロタスクを含むプログラムのテストにおいて、特定のマクロタスクを対象としたテストが効率的に実行できる。

図面の簡単な説明

[0010] [図1]演算装置のハードウェア構成図

[図2]演算装置の機能構成図

[図3]プログラムを示す有向グラフ

[図4]エリアテーブルの一例を示す図

[図5]プログラムに含まれる各エリアに対応するテストケースを示す図

[図6]テスト演算部による構造解析処理を示すフローチャート

[図7]テスト演算部による構造解析処理を示すフローチャート

[図8]テスト演算部による構造解析処理を示すフローチャート

[図9]テスト演算部によるテーブル作成処理を示すフローチャート

[図10]テスト演算部によるメモリダンプ処理を示すフローチャート

[図11]テスト演算部によるプログラムテストを示すフローチャート

[図12]テスト演算部によるプログラムテストを示すフローチャート

[図13]テスト演算部によるプログラムテストを示すフローチャート

発明を実施するための形態

[0011] 一第1の実施の形態一

以下、図1～図13を参照して、演算装置が実行するテスト方法の第1の実施の形態を説明する。

[0012] (ハードウェア構成)

図1は、本発明に係るテスト方法を実行する演算装置1のハードウェア構成図である。演算装置1は、中央演算装置であるCPU2と、読み出し専用の記憶装置であるROM3と、読み書き可能な記憶装置であるRAM4と、不揮発性の記憶装置である記憶部5と、外部インタフェース6とを備える。CPU2は、ROM3に格納されるプログラムをRAM4に展開して実行することで後述する複数の機能を実現する。記憶部5はたとえば、ハードディスクドライブやフラッシュメモリなどである。ただし記憶部5が演算装置1の内部に格納されることは必須の構成要件ではなく、演算装置1の外部に設置された記憶部5に対して演算装置1が外部インタフェース6を介してアクセスしてもよい。

[0013] 外部インタフェース6は、演算装置1の外部とのインタフェースであり、対象は人間およびコンピュータのいずれでもよい。たとえば、人間を対象とするマウス、キーボード、ディスプレイ、およびプリンタなどの装置そのものやこれら装置の接続インタフェースでもよい。また他のコンピュータを対象とする通信インタフェース、およびリムーバブルメディアの読み書き装置でもよい。以下では、演算装置1を操作する人間を「オペレータ」と呼ぶ。

[0014] (機能構成)

図2は、演算装置1が有する機能を機能ブロックとして表した機能構成図である。ただし図2には記憶部5に格納される情報も示している。演算装置1はその機能として、入出力部21と、テスト演算部22と、プログラム実

行部 23 とを備える。記憶部 5 には、キュー Q と、第 1 スタック S1 と、第 2 スタック S2 と、確認済リスト CL と、処理済リスト PL と、エリアテーブル T100 と、テスト対象プログラム情報 TP とが格納される。

[0015] キュー Q、第 1 スタック S1、第 2 スタック S2、確認済リスト CL、および処理済リスト PL は、テスト演算部 22 が読み書きする対象であり、テスト演算部 22 が処理の過程で利用するデータの格納場所である。これらの情報は本実施の形態では演算装置 1 の外部に対して出力されない。キュー Q は、プログラムの分野で用いられる一般的なキューであり、最初に入力されたデータが最初に出力される性質を有する。第 1 スタック S1 および第 2 スタック S2 は、プログラムの分野で用いられる一般的なスタックであり、最初に入力されたデータが最後に出力される性質を有する。確認済リスト CL および処理済リスト PL は要素の集合であり要素間の順序は存在しない。

[0016] 本実施の形態では、キュー Q、第 1 スタック S1、および第 2 スタック S2 は、入力された順番にその要素を記載する。たとえば第 1 スタック S1 がカラ、すなわち空集合の状態において、最初に「A」、次に「B」、最後に「C」が第 1 スタック S1 に入力、すなわちプッシュされた場合は、第 1 スタック S1 のデータを「A、B、C」と記載する。

[0017] テスト対象プログラム情報 TP は、演算装置 1 がテストを行う対象のプログラムに関する情報である。以下では、演算装置 1 がテストを行う対象のプログラムを「プログラム P1」と呼ぶ。テスト対象プログラム情報 TP には、少なくともプログラム P1 を作成可能な情報、たとえばソースコードやオブジェクトファイルが含まれる。なお後述するように、プログラム P1 の動作は非循環有向グラフを用いて表現ができる。プログラム P1 の動作を示す非循環有向グラフは、テスト対象プログラム情報 TP に含まれてもよいし、逆アセンブルなどの既知の手法を用いて演算装置 1 が作成してもよい。エリアテーブル T100 には、テスト演算部 22 がプログラム P1 を解析して得られた情報が格納される。詳しくは後述する。

[0018] 入出力部 21 は、演算装置 1 の外部から演算に必要な情報、たとえばプロ

グラム P 1 に関する情報やオペレータの指示を取得し、必要に応じて演算装置 1 の外部に演算結果を出力する。入出力部 2 1 は、CPU 2 による演算に加えて、外部インタフェース 6 をあわせて用いることで実現される。

[0019] テスト演算部 2 2 は、プログラム P 1 のテストを行うために必要な様々な演算を行う。以下では主にテスト演算部 2 2 の動作を説明する。テスト演算部 2 2 は初めに、処理対象であるプログラム P 1 の構造を解析する構造解析処理を実行する。構造解析処理ではプログラム P 1 を、並列化に寄与しない非並列群および並列化に寄与する並列寄与群に分別する。ただし厳密には構造解析処理では、プログラム P 1 を構成するマクロタスクのうち、並列寄与群を特定する第 2 スタック S 2 を作成する。すなわちプログラム P 1 を構成するマクロタスクのうち第 2 スタック S 2 により特定されないマクロタスクが非並列群である。

[0020] 構造解析処理の次にテスト演算部 2 2 は、構造解析処理の結果を利用してプログラム P 1 に含まれる各エリアの情報をエリアテーブル T 1 0 0 にまとめるテーブル作成処理を行う。さらにテスト演算部 2 2 は、効率的なテストのためにメモリダンプを取得するメモリダンプ処理を行う。最後にテスト演算部 2 2 は、所定のエリアを対象としてプログラムテストを実行する。なおテーブル作成処理とメモリダンプ処理は実行の順番は不問である。

[0021] プログラム実行部 2 3 は、テスト演算部 2 2 の動作指令に従い、プログラム P 1 を実行する。ただしプログラム実行部 2 3 は、プログラム P 1 の全体を実行する場合もあるが、プログラム P 1 の一部だけを実行する場合もある。さらにプログラム実行部 2 3 は、テスト演算部 2 2 により指定された位置でプログラムの実行を停止する機能を有する。またプログラム実行部 2 3 は、プログラム P 1 を実行するコンピュータに内蔵されるプロセッサをソフトウェアでエミュレートしてもよいし、CPU 2 とは異なるプロセッサを演算装置 1 が有しておりプログラム実行部 2 3 はそのプロセッサを利用してもよい。

[0022] (プログラムの例示)

プログラム P 1 は、複数の処理単位に分解可能であり、非巡回有向グラフによりその動作が表される。すなわち演算装置 1 は、非巡回有向グラフを用いて動作が表現できるプログラムであれば、あらゆるプログラムを分析対象とできる。以下では処理単位のそれぞれを「マクロタスク」と呼ぶ。マクロタスクは、内部に分岐処理を含まない 1 または複数の処理である。

[0023] 図 3 は、本実施の形態において処理対象とするプログラム P 1 を示す有向グラフである。プログラム P 1 は、MT 6、MT 9、MT 14、MT 16、MT 17、MT 18、MT 19、MT 21、MT 22、および MT 23 の合計 10 のマクロタスクにより構成される。図 3 に示すマクロタスクの位置関係は、矢印の向きとも関係があり、いずれの矢印も図示上部から図示下部に向かっており、先祖から子孫に向かうとも言う。すなわち、図示上部に記載されているほど先祖であり、図示下部に記載されているほど子孫である。

[0024] プログラム P 1 の処理が開始すると最初の先祖である MT 14 の実行が開始され、最後の子孫である MT 23 の実行が完了するとプログラム P 1 が終了する。図 3 は、MT 14 が完了すると、順不同で MT 16～MT 19 がそれぞれ実行されることを示している。以下では、所定の範囲における最初の先祖を「ルートノード」とも呼び、所定の範囲における最後の子孫を「末端ノード」と呼ぶ。たとえば図 3 に示す全体の範囲ではルートノードは MT 6 であり、末端ノードは MT 23 である。

[0025] 図 3 において矩形で囲むエリア A 11 は、MT 6、MT 9、および MT 14 を含む。破線で囲むエリア A 12 は、MT 14、MT 18、MT 19 および MT 21 を含む。一点鎖線で囲むエリア A 13 は、MT 14、MT 17～MT 19、MT 21、および MT 22 を含む。エリア A 11 において、ルートノードは MT 6、末端ノードは MT 14 である。エリア A 12 において、ルートノードは MT 14、末端ノードは MT 21 である。

[0026] (エリアテーブル T 100)

図 4 は、エリアテーブル T 100 の一例を示す図である。エリアテーブル T 100 には、テスト演算部 22 が算出した並列寄与群の情報が記載される

。詳しくは後述するが、エリアテーブルT100に算出される並列寄与群は、設定される注目タスクにより異なる。図4に示す例は、図3に示したプログラムP1を対象とし、注目タスクをT22とした場合の例である。なお図4は概念的にエリアテーブルT100を表形式で表現しているにすぎず、表形式で記録しなくてもよいし、表の各セルにはデータの実態へのリンクを記載してもよい。

[0027] エリアテーブルT100は1または複数のレコードを有し、各レコードはプログラムT101、エリアID T102、並列化済バイナリT103、開始タスクT104、終了タスクT105、開始点ダンプT106、終了点ダンプT107、および有向グラフT108のフィールドを有する。プログラムT101には、解析対象となるプログラムの識別子、たとえばプログラムの名称が格納される。エリアID T102には、エリアの識別子、たとえばエリアの名称が格納される。

[0028] 並列化済バイナリT103には、そのレコードのエリアに含まれる処理を実行するバイナリファイルであって、可能な限り並列化処理が施されたバイナリファイルが格納される。すなわちそのエリアにおいて並列化処理が可能な場合には並列化処理が施されたバイナリファイルが格納され、そのエリアにおいて並列化処理が不可能な場合には並列化されていないバイナリファイルが格納される。

[0029] 開始タスクT104には、そのレコードのエリアに含まれる最も先祖のマクロタスクの名称が格納される。終了タスクT105には、そのレコードのエリアに含まれる最も子孫のマクロタスクの名称が格納される。開始点ダンプT106には、開始タスクT104の最後の処理におけるメモリのダンプ情報が格納される。なおメモリのダンプ情報は、「メモリのスナップショット」とも呼ばれる。

[0030] 終了点ダンプT107には、終了タスクT105の最初の処理におけるメモリのダンプ情報が格納される。有向グラフT108には、そのレコードのエリアに含まれるマクロタスクの有向グラフ、たとえば図3に示す有向グラ

フの情報が格納される。

[0031] (テストケース)

図5は、図3に示す各エリアに対応するテストケースを示す図である。テストケースは、テスト演算部22により作成される。テストケースは、エリア内に含まれるすべてのマクロタスクを実行順に並べたリストであり、図5に示す例では実現可能なすべての実行順番を列挙している。なお本実施の形態ではエリアA11のテストケースは作成されないが、図5では参考のために記載している。エリアA11は3つのマクロタスクが含まれ、それらは直列に一直線で接続される。すなわちエリアA11ではマクロタスクの実行順番は選択の余地がなくケース「A11-1」の1とおりのみが存在する。エリアA11の各タスクは、順番に実行する必要があるので、並列化に寄与しない非並列群である。

[0032] エリアA12は4つのマクロタスクが含まれ、MT18とMT19が並列に存在する。そのため、MT18とMT19のいずれが先に実行される場合もありえるため、エリアA12は「A12-1」および「A12-2」の2つのケースが存在する。具体的には、最初に実行されるのはMT14で固定、最後に実行されるのはMT21で固定であり、2番目と3番目に実行されるマクロタスクがMT18およびMT19のいずれかである。

[0033] エリアA13は、6つのマクロタスクが含まれ、大きくはMT18、MT19、およびMT21の第1グループと、MT17のみからなる第2のグループが並列に存在する。さらに第1のグループの内部ではMT18とMT19が並列に存在する。並列に存在するマクロタスクは実行順番が不問であり、いずれのタイミングで実行されてもよい。そのためエリアA13では「A13-1」～「A13-8」の8つのケースが存在する。

[0034] (構造解析処理のフローチャート)

図6～図8は、テスト演算部22による構造解析処理を示すフローチャートである。まずステップS100ではテスト演算部22は、キューQ、第1スタックS1、第2スタックS2、および確認済リストCLを初期化する。

具体的には本ステップの初期化処理により、キューQ、第1スタックS1、第2スタックS2、および確認済リストCLがカラ、すなわち空集合に設定される。

[0035] 続くステップS101ではテスト演算部22は、プログラムP1に関する各種情報を読み込む。続くステップS102ではテスト演算部22は、ステップS101において読み込んだ情報から有向グラフD1を抽出する。本フローチャートにおける動作例の説明において、本ステップにおいて抽出した有向グラフD1は、図3に例示した有効グラフとする。

[0036] 続くステップS103ではテスト演算部22は、プログラムP1に含まれるいずれかのマクロタスクMTである注目点Xを決定する。注目点Xは、機械処理によりたとえば直前の実行によりエラーが発見された箇所が自動的に選択されてもよいし、オペレータにより手動で設定されてもよい。本フローチャートにおける動作例の説明において、注目点Xは「MT22」とする。続くステップS104ではテスト演算部22は、注目点Xの先祖リストC1を作成する。有向グラフD1が図3に示すもので、注目点Xが「MT22」の場合に先祖リストC1は「MT6、MT9、MT14、MT17、MT18、MT19、MT21」となる。

[0037] 続くステップS105ではテスト演算部22は、注目点Xを処理対象Xiに設定する。続くステップS106ではテスト演算部22は、処理対象Xiを確認済みに設定する。具体的にはテスト演算部22は、処理対象Xiである「MT22」を確認済リストCLに追加する。ステップS106の処理が完了するとテスト演算部22は、ステップS106Aに進む。

[0038] ステップS106Aではテスト演算部22は、先祖リストC1中に未確認の処理対象Xiの隣接ノードが存在するか否かを判断する。隣接ノードとは、子孫や先祖に関係なく、換言すると上下方向の位置に関係なく、1つだけ離れたノードである。たとえば図3に示す例において、「MT22」の隣接ノードは「MT17、MT21、MT23」である。

[0039] テスト演算部22は、先祖リストC1中に未確認の処理対象Xiの隣接ノ

ードが存在すると判断する場合にはステップS 1 0 7に進み、先祖リストC 1に含まれる処理対象X iの隣接ノードは全て確認済であると判断する場合にはステップS 1 0 7 Aに進む。具体的にはテスト演算部2 2は、まず先祖リストC 1に含まれる処理対象X iの隣接ノードを全て挙げ、それらが確認済リストC Lに記載されているか否かにより本ステップの判断ができる。

[0040] ステップS 1 0 7ではテスト演算部2 2は、処理対象X iの入次数が1より大きいであるか否かを判断する。テスト演算部2 2は、処理対象X iの入次数が1より大きいと判断する場合はステップS 1 1 0に進み、処理対象X iの入次数が1より大きくない、すなわち1以下であると判断する場合はステップS 1 0 7 Aに進む。

[0041] ステップS 1 0 7 Aではテスト演算部2 2は、先祖リストC 1中に確認済の処理対象X iの隣接ノードが存在するか否かを判断する。本ステップにおける判断をステップS 1 0 6 Aと比較すると、未確認か確認済かが異なる。テスト演算部2 2は、先祖リストC 1中に確認済の処理対象X iの隣接ノードが存在すると判断する場合にはステップS 1 0 8に進み、先祖リストC 1に含まれる処理対象X iの隣接ノードは全て未確認であると判断する場合にはステップS 1 0 9に進む。

[0042] ステップS 1 0 8ではテスト演算部2 2は、処理対象X iの出次数が1より大きいと否かを判断する。テスト演算部2 2は、処理対象X iの出次数が1より大きいと判断する場合はステップS 1 1 0に進み、処理対象X iの出次数が1以下と判断する場合はステップS 1 0 9に進む。入次数および出次数の具体例を挙げると、図3に示す例においてTM 1 4の入次数は「1」、出次数は「4」である。

[0043] ステップS 1 0 9ではテスト演算部2 2は、処理対象X iは先祖リストC 1におけるソースノードX Nに該当するか否かを判断する。テスト演算部2 2は、処理対象X iは先祖リストC 1におけるソースノードX Nに該当すると判断する場合はステップS 1 1 0に進み、処理対象X iは先祖リストC 1におけるソースノードX Nに該当しないと判断する場合はステップS 1 1 1

に進む。ステップS 1 1 1ではテスト演算部2 2は、処理対象X iの近傍ノードを抽出し、隣接グループNを定義する。たとえば図3に示す例において処理対象X iが「MT 2 2」の場合は隣接グループNは、「MT 1 7、MT 2 1、MT 2 3」である。ステップS 1 1 1の処理が完了するとテスト演算部2 2は、丸囲みのAを経由して図7におけるステップS 1 1 1 Aに進む。以上が図6の説明である。

[0044] 図7のステップS 1 1 1 Aではテスト演算部2 2は、処理済リストPLを初期化する。続くステップS 1 1 9ではテスト演算部2 2は、隣接グループNの全要素を処理したか否かを判断し、全要素を処理したと判断する場合はステップS 1 2 0に進み、未処理の要素が存在すると判断する場合はステップS 1 1 2に進む。ステップS 1 1 9ではテスト演算部2 2は、隣接グループNの要素と、処理済リストPLに記載されたマクロタスクとを比較することでステップS 1 1 9の判断ができる。

[0045] ステップS 1 1 2ではテスト演算部2 2は、隣接グループNから未処理のいずれかのマクロタスクを要素N iとして抽出する。続くステップS 1 1 2 Aではテスト演算部2 2は、ステップS 1 1 2において抽出した要素N iを処理済リストPLに追加する。換言するとテスト演算部2 2は、ステップS 1 1 2において要素N iとして抽出されたマクロタスクの名称を処理済リストPLに追加する。続くステップS 1 1 3ではテスト演算部2 2は、要素N iが先祖リストC 1に含まれるか否かを判断する。テスト演算部2 2は、要素N iが先祖リストC 1に含まれると判断する場合はステップS 1 1 4に進み、要素N iが先祖リストC 1に含まれないと判断する場合はステップS 1 1 9に戻る。

[0046] ステップS 1 1 4ではテスト演算部2 2は、要素N iは確認済みであるか否か、換言すると要素N iが確認済リストCLに記載されているか否かを判断する。テスト演算部2 2は、要素N iは確認済みであると判断する場合はステップS 1 1 5に進み、要素N iは確認済みではないと判断する場合はステップS 1 1 7に進む。ステップS 1 1 5ではテスト演算部2 2は、要素N

i はキューQに含まれるか否かを判断する。テスト演算部22は、要素N i はキューQに含まれると判断する場合はステップS 1 1 6に進み、要素N i はキューQに含まれないと判断する場合はステップS 1 1 9に戻る。

[0047] ステップS 1 1 7ではテスト演算部22は、キューQの末尾に要素N i を追加する。続くステップS 1 1 8ではテスト演算部22は、要素N i を確認済みに変更する。具体的にはテスト演算部22は、要素N i として抽出されたマクロタスクの名称を確認済リストCLに追加する。

[0048] ステップS 1 1 9において肯定判断されると実行されるステップS 1 2 0ではテスト演算部22は、キューQが空集合であるか否かを判断する。テスト演算部22は、キューQが空集合であると判断する場合は丸囲みのCを経由して図8のステップS 6 0 1に進み、キューQが空集合ではないと判断するとステップS 1 2 1に進む。なおステップS 1 2 0において肯定判断される場合とは、ステップS 1 2 0が実行される前にステップS 1 1 7が1回も実行されない場合である。ステップS 1 2 1ではテスト演算部22は、キューQの先頭を取り出して処理対象X i とし、丸囲みのBを経由して図6におけるステップS 1 0 6 Aに戻る。以上が図7の説明である。

[0049] 図8に示す最初の処理であるステップS 6 0 1ではテスト演算部22は、変数i にゼロを代入してステップS 6 0 2に進む。ステップS 6 0 2ではテスト演算部22は、第1スタックS 1の長さが「i + 1」の値よりも長いかな否かを判断する。なお第1スタックS 1の長さとは、第1スタックS 1に追加されたマクロタスクの数である。テスト演算部22は、第1スタックS 1の長さが「i + 1」の値よりも長いと判断する場合はステップS 6 0 3に進み、第1スタックS 1の長さが「i + 1」の値以下であると判断する場合は図8に示す処理を終了する。

[0050] ステップS 6 0 3ではテスト演算部22は、第1スタックS 1におけるi + 1番目の要素を有向グラフD 1から抽出し、抽出したマクロタスクの先祖リストを一時先祖リストA 0 0として作成する。前述のとおり本実施の形態では、第1スタックS 1の要素番号は、最後に追加された要素を「0」番目

として定義する。仮に第1スタックS1に、「MT22, MT21, MT14」の順番に要素が追加され、変数iの値がゼロの場合には、第1スタックS1の「i+1」番目の要素は「MT21」である。＜正しいかをコメントで聞く＞この「MT21」の先祖リストが一時先祖リストA00として作成される。この場合の一時先祖リストA00は「MT14, MT18, M19」である。なおステップS603において作成される一時先祖リストA00の有効期限は、次にステップS603が実行されるとき、および図8の処理が終了するとき、のいずれか早い方までである。

[0051] 続くステップS604ではテスト演算部22は、変数jに変数iの値を代入してステップS605に進む。ステップS605ではテスト演算部22は、変数jの値が「-1」よりも大きいか否かを判断し、変数jが「-1」よりも大きいと判断する場合はステップS606に進み、変数jが「-1」以下であると判断する場合はステップS614に進む。ステップS606ではテスト演算部22は、第1スタックS1におけるj番目の要素であるマクロタスクを特定し、そのマクロタスクの子孫を先祖リストC1から抽出して一時子孫リストD00を作成する。たとえば第1スタックS1におけるj番目の要素が「MT14」であり、先祖リストC1が「MT14, MT17, MT18, M19, MT21」の場合に一時子孫リストD00は「MT17, MT18, MT19, MT21」となる。

[0052] 続くステップS607ではテスト演算部22は、ステップS606において作成した一時子孫リストD00と、ステップS603において作成した一時先祖リストA00と共通する要素を抽出して重複リストIS00を作成する。たとえば一時子孫リストD00が「MT17, MT18, MT19, MT21」、一時先祖リストA00が「MT14, MT18, M19」の場合に、重複リストIS00は「MT18, M19」である。続くステップS608ではテスト演算部22は、重複リストIS00の各要素において、入次数から出次数を引いた値を算出してその総和をSD00として算出する。

[0053] 続くステップS609ではテスト演算部22は、第1スタックS1におけ

る j 番目の要素のマクロタスクにおける出次数を $OD00$ として算出する。続くステップ $S610$ ではテスト演算部 22 は、第 1 スタック $S1$ における j 番目の要素のマクロタスクにおける入次数を $ID00$ として算出する。続くステップ $S610$ ではテスト演算部 22 は、ステップ $S608 \sim S610$ の算出結果を用いて、 $OD00$ と $SD00$ の和から $ID00$ を引いた値がゼロ以上であるか否かを判断する。テスト演算部 22 は、ステップ $S610$ を肯定判断する場合はステップ $S611$ に進み、否定判断する場合はステップ $S613$ に進む。

[0054] ステップ $S611$ ではテスト演算部 22 は、第 2 スタック $S2$ に第 1 スタック $S1$ の $i+1$ 番目の要素を挿入する。続くステップ $S612$ ではテスト演算部 22 は、第 2 スタック $S2$ に第 1 スタック $S1$ の j 番目の要素を挿入してステップ $S614$ に進む。なおステップ $S611$ および $S612$ は、第 2 スタック $S2$ に対するいわゆるプッシュ操作である。ステップ $S613$ ではテスト演算部 22 は、変数 j の値を「 1 」だけ減少させてステップ $S605$ に戻る。ステップ $S614$ ではテスト演算部 22 は、変数 i の値を「 1 」だけ増加させてステップ $S602$ に戻る。ステップ $S613$ は、いわゆるデクリメント処理であり、ステップ $S614$ は、いわゆるインクリメント処理である。以上が図 8 の説明である。

[0055] なお図 $6 \sim 8$ の処理により生成される第 1 スタック $S1$ には、分岐を含むエリアを示す情報が含まれる。また第 2 スタック $S2$ には、第 1 スタック $S1$ に含まれるエリア内における分岐を含むエリアを示す情報が含まれる。

[0056] (構造解析処理の動作例)

構造解析処理の動作例を説明する。以下に説明する動作例では動作主体の記載を省略するが、動作主体は全てテスト演算部 22 である。ステップ $S102$ において抽出される有向グラフ $D1$ が図 3 に示すものであり、注目点 X が「 $MT22$ 」の場合の例を説明する。この場合に、ステップ $S104$ において生成される先祖リスト $C1$ は「 $MT6$ 、 $MT9$ 、 $MT14$ 、 $MT17$ 、 $MT18$ 、 $MT19$ 、 $MT21$ 」である。ステップ $S106A$ ではテスト演

算部22は、処理対象 X_i である「MT22」の隣接ノードが先祖リストC1に存在し、かつ確認済リストCLには含まれていないので肯定判断する。そしてステップS107では「MT22」の入り次数が2なので、テスト演算部22は肯定判断してステップS110に進み、第1スタックS1が「MT22」に更新される。

[0057] そしてステップS111ではテスト演算部22は、隣接グループNを「MT17、MT21、MT23」と定義する。図7では、隣接グループNの全要素を処理するまでS112～S118のループが回り続けるので、隣接グループNに含まれる全要素を順番に要素 N_i としてS113～S118の処理が行われる。

[0058] 要素 N_i が「MT17」の場合には、ステップS113が肯定判断され、ステップS114が否定判断され、ステップS117およびステップS118が実行されてステップS119に戻る。要素 N_i が「MT21」の場合には、ステップS113が肯定判断され、ステップS114が否定判断され、ステップS117およびステップS118が実行されてステップS119に戻る。要素 N_i が「MT23」の場合には、ステップS113が否定判断されてステップS119に戻る。

[0059] したがって、初回にステップS119において肯定判断をする際には、キューQは「MT17、MT21」、確認済リストCLは「MT17、MT21、MT22」、第1スタックS1は「MT22」である。続くステップS120ではテスト演算部22は否定判断をしてステップS121に進み、キューQから先頭の「MT17」を取り出して処理対象 X_i として、丸囲みのBを経由して図6のステップS106Aに戻る。すなわちS106Aに戻った時点でキューQの要素は「MT21」のみとなる。

[0060] (構造解析処理の動作例 | 初回にBを経由した後)

処理対象 X_i である「MT17」について、ステップS106A、S107Aは肯定判断されるが、「MT17」の入次数および出次数は「1」なのでステップS107、S108は否定判断される。さらに「MT17」は先

祖リストC 1におけるソースノードではないのでステップS 1 1 1に進む。
ここで「MT 1 7」の近傍である「MT 1 4、MT 2 2」が新たな隣接グループNとして定義されて図7に進む。

[0061] 図7では、まず隣接グループNの要素である「MT 1 4、MT 2 2」を順番に要素N iとしてS 1 1 3～S 1 1 8の処理が行われる。要素N iが「MT 1 4」の場合には、ステップS 1 1 3が肯定判断され、ステップS 1 1 4が否定判断され、ステップS 1 1 7およびステップS 1 1 8が実行されてステップS 1 1 9に戻る。要素N iが「MT 2 2」の場合には、ステップS 1 1 3が肯定判断され、ステップS 1 0 6における処理が理由でステップS 1 1 4が肯定判断される。「MT 2 2」はキューQには含まれないので否定判断されてステップS 1 1 9に戻る。

[0062] すなわちステップS 1 1 9が2回目に肯定判断をする際には、キューQは「MT 2 1、MT 1 4」、確認済リストCLは「MT 1 7、MT 2 1、MT 2 2、MT 1 4」、第1スタックS 1は「MT 2 2」である。ここでもキューQは空集合でないのでステップS 1 2 0からステップS 1 2 1に進み、キューQから先頭の「MT 2 1」を取り出して処理対象X iとして図6のステップS 1 0 6 Aに戻る。すなわちS 1 0 6 Aに戻った時点でキューQの要素は「MT 1 4」のみとなる。

[0063] (構造解析処理の動作例 | 2回目にBを経由した後)

処理対象X iである「MT 2 1」について、ステップS 1 0 6 Aは肯定判断され、「MT 2 1」の入次数が「2」なのでステップS 1 0 7が肯定判断されてS 1 1 0に進み第1スタックS 1に「MT 2 1」が追加される。すなわち第1スタックS 1は「MT 2 2、MT 2 1」となる。続くステップS 1 1 1では「MT 2 1」の近傍である「MT 1 8、MT 1 9、MT 2 2」が新たな隣接グループNとして定義されて図7に進む。

[0064] 図7では、まず隣接グループNの要素である「MT 1 8、MT 1 9、MT 2 2」のそれぞれを要素N iとしてS 1 1 3～S 1 1 8の処理が行われる。要素N iが「MT 1 8」の場合には、ステップS 1 1 3が肯定判断され、ス

テップS 1 1 4 が否定判断され、ステップS 1 1 7 およびステップS 1 1 8 が実行されてステップS 1 1 9 に戻る。

[0065] 要素N i が「MT 1 9」の場合には、ステップS 1 1 3 が肯定判断され、ステップS 1 1 4 が否定判断され、ステップS 1 1 7 およびステップS 1 1 8 が実行されてステップS 1 1 9 に戻る。要素N i が「MT 2 2」の場合には、ステップS 1 1 3 が肯定判断され、ステップS 1 1 4 が肯定判断され、「MT 2 2」がキューQには含まれないので否定判断されてステップS 1 1 9 に戻る。

[0066] すなわちステップS 1 1 9 が3回目に肯定判断をする際には、キューQは「MT 1 4、MT 1 8、MT 1 9」、確認済リストCLは「MT 1 7、MT 1 8、MT 1 9、MT 2 1、MT 2 2、MT 1 4」、第1スタックS 1 は「MT 2 2、MT 2 1」である。ここでもキューQは空集合でないのでステップS 1 2 0 からステップS 1 2 1 に進み、キューQから先頭の「MT 1 4」を取り出して処理対象X i として図6のステップS 1 0 6 Aに戻る。すなわちS 1 0 6 Aに戻った時点でキューQの要素は「MT 1 8、MT 1 9」となる。

[0067] (構造解析処理の動作例 | 3回目にBを経由した後)

処理対象X i である「MT 1 4」について、先祖リストC 1 に含まれる隣接ノードである「MT 1 7、MT 1 8、MT 1 9」は全て確認済リストCLに含まれるのでステップS 1 0 6 Aは否定判断され、S 1 0 7 Aでは肯定判断される。そして「MT 1 4」の出次数が1より大きいのでステップS 1 0 8 が肯定判断されてS 1 1 0 に進み第1スタックS 1 に「MT 1 4」が追加される。すなわち第1スタックS 1 は「MT 2 2、MT 2 1、MT 1 4」となる。続くステップS 1 1 1 では「MT 1 4」の近傍である「MT 9、MT 1 6、MT 1 7、MT 1 8、MT 1 9」が新たな隣接グループNとして定義されて図7に進む。

[0068] 図7では、まず隣接グループNの要素である「MT 9、MT 1 6、MT 1 7、MT 1 8、MT 1 9」のそれぞれを要素N i としてステップS 1 1 3 ~

S 1 1 8 の処理が行われる。要素 N i が「MT 9、MT 1 6」の場合には、先祖リスト C 1 に含まれないのでステップ S 1 1 3 が否定判断されてステップ S 1 1 9 に戻る。要素 N i が「MT 1 7」の場合には、ステップ S 1 1 3 が肯定判断され、「MT 1 7」が確認済みなのでステップ S 1 1 4 が肯定判断され、「MT 1 7」はキュー Q に含まれないので否定判断されてステップ S 1 1 9 に戻る。

[0069] 要素 N i が「MT 1 8、MT 1 9」の場合にはいずれも、ステップ S 1 1 3、S 1 1 4、および S 1 1 9 が肯定判断され、ステップ S 1 1 6 においてキュー Q の末尾に要素 N i が追加される。すなわちこの時点でキュー Q は「MT 1 8、MT 1 9」、第 1 スタック S 1 は「MT 2 2、MT 2 1、MT 1 4」である。ここでもキュー Q は空集合でないのでステップ S 1 2 0 からステップ S 1 2 1 に進み、キュー Q から先頭の「MT 1 8」を取り出して処理対象 X i として図 6 のステップ S 1 0 6 A に戻る。すなわち S 1 0 6 A に戻った時点でキュー Q の要素は「MT 1 9」となる。

[0070] (構造解析処理の動作例 | 4 回目に B を経由した後)

処理対象 X i である「MT 1 8」について、隣接ノードである「MT 1 4、MT 2 1」は全て確認済リスト C L に含まれるのでステップ S 1 0 6 A は否定判断され、S 1 0 7 A では肯定判断される。ステップ S 1 0 8 では、「MT 1 4」の出次数は「1」なので否定判断をして S 1 0 9 に進み、「MT 1 4」はソースノードではないので否定判断してステップ S 1 1 1 に進む。ステップ S 1 1 1 では「MT 1 9」の近傍である「MT 1 4、MT 2 1」が新たな隣接グループ N として定義されて図 7 に進む。

[0071] 図 7 では、まず隣接グループ N の要素である「MT 1 4、MT 2 1」のそれぞれを要素 N i としてステップ S 1 1 3 ~ S 1 1 8 の処理が行われる。要素 N i が「MT 1 4、MT 2 1」の場合にはいずれも、先祖リスト C 1 に含まれるのでステップ S 1 1 3 が肯定判断され、確認済リスト C L に含まれるのでステップ S 1 1 4 が肯定判断され、キュー Q に含まれないので否定判断されてステップ S 1 1 9 に戻る。そしてステップ S 1 1 9 では肯定判断がさ

れる。

[0072] この場合には、キューQには何ら追加がされず、4回目にBを経由した直後の状態である、キューQには「MT 1 9」のみが含まれる状態である。また第1スタックS 1は「MT 2 2、MT 2 1、MT 1 4」である。ステップS 1 2 0では、キューQは空集合でないのでステップS 1 2 1に進み、キューQから先頭の「MT 1 9」を取り出して処理対象X iとして図6のステップS 1 0 6 Aに戻る。すなわちS 1 0 6 Aに戻った時点でキューQの要素は空集合となる。

[0073] (構造解析処理の動作例 | 5回目にBを経由した後)

処理対象X iである「MT 1 9」について、隣接ノードである「MT 1 4、MT 2 1」は全て確認済リストCLに含まれるのでステップS 1 0 6 Aは否定判断され、S 1 0 7 Aでは肯定判断される。ステップS 1 0 8では、「MT 1 4」の出次数は「1」なので否定判断をしてS 1 0 9に進み、「MT 1 9」はソースノードではないので否定判断してステップS 1 1 1に進む。ステップS 1 1 1では「MT 1 9」の近傍である「MT 1 4、MT 2 1」が新たな隣接グループNとして定義されて図7に進む。

[0074] 図7では、まず隣接グループNの要素である「MT 1 4、MT 2 1」のそれぞれを要素N iとしてステップS 1 1 3～S 1 1 8の処理が行われる。要素N iが「MT 1 4、MT 2 1」の場合にはいずれも、先祖リストC 1に含まれるのでステップS 1 1 3が肯定判断され、確認済リストCLに含まれるのでステップS 1 1 4が肯定判断され、キューQに含まれないので否定判断されてステップS 1 1 9に戻る。そしてステップS 1 1 9では肯定判断がされる。

[0075] 5回目にBを経由した後の処理では、キューQには何ら追加がされないで、5回目にBを経由した直後の状態と同様に、キューQは空集合である。ステップS 1 2 0では、キューQは空集合なので初めて肯定判断をして、丸囲みのCを経由して図8のステップS 6 0 1に進む。この時点で、第1スタックS 1は「MT 2 2、MT 2 1、MT 1 4」である。

[0076] (構造解析処理の動作例 | 図8の初回処理)

第2スタックS2は、図6の冒頭において初期化されて以後、ここまで一切の操作が行われていないので、図8の処理が開始される時点では空集合のままである。第1スタックは図6～図7の処理において上述するようにいくつかの要素が追加されている。なお本実施の形態では、第1スタックS1に対するポップ操作は行われず、第1スタックS1の長さが短くなることはない。

[0077] ステップS601において変数iにゼロを代入する。ステップS602では、第1スタックS1の長さは「3」なので、「i+1」すなわち「1」よりも大きいことから肯定判断がされる。第1スタックS1における「i+1」すなわち「1」番目の要素は「MT21」なので、ステップS603では「MT21」の先祖リストである「MT14、MT18、MT19」を一時先祖リストA00とする。続くS604では変数jに変数iの値であるゼロを代入し、続くステップS605では変数jが「-1」よりも大きいことので肯定判断してステップS606に進む。

[0078] ステップS606では、第1スタックS1の「j」番目、すなわち「0」番目の要素である「MT14」の子孫を先祖リストC1から抽出し、「MT18、MT19、MT21」を一時子孫リストD00とする。続くステップS607では、一時子孫リストD00と一時先祖リストA00に共通する「MT18、MT19」を重複リストIS00とする。続くステップS608では、MT18およびMT19の入次数と出次数の差の総和であるゼロをSD00とする。

[0079] 続くステップS609では、第1スタックS1の「j」番目、すなわち「0」番目の要素である「MT14」の出次数「4」をOD00とする。続くステップS610では、第1スタックS1の「i+1」番目、すなわち「1」番目の要素である「MT21」の入次数「2」をID00とする。続くステップS610Aでは、 $OD00 + SD00 - ID00$ の値を「2」と算出し、これがゼロ以上なので肯定判断してステップS611に進む。

[0080] ステップS 6 1 1では、第2スタックS 2に第1スタックS 1の「 $i + 1$ 」番目、すなわち「1」番目の要素である「MT 2 1」を挿入する。続くステップS 6 1 2では、第2スタックS 2に第1スタックS 1の「 j 」番目、すなわち「0」番目の要素である「MT 1 4」を挿入する。続くステップS 6 1 4では変数 i をインクリメントして「1」としてステップS 6 0 2に戻る。以上説明した図8の1週目の処理により、第2スタックS 2は「MT 2 1、MT 1 4」に更新された。

[0081] (構造解析処理の動作例 | 図8の2週目)

2回目に実行されるステップS 6 0 2では、第1スタックS 1の長さ「3」が「 $i + 1$ 」すなわち「2」よりも長いので肯定判断してS 6 0 3に進む。ステップS 6 0 3では、第1スタックS 1の「 $i + 1$ 」番目、すなわち「2」番目の要素である「MT 2 2」の先祖リストである「MT 1 4、MT 1 7、MT 1 8、MT 1 9、MT 2 1」を一時先祖リストA 0 0とする。続くS 6 0 4では変数 j に変数 i の値である「1」を代入し、続くステップS 6 0 5では変数 j が「 -1 」よりも大きいので肯定判断してステップS 6 0 6に進む。

[0082] ステップS 6 0 6では、第1スタックS 1の「 j 」番目、すなわち「1」番目の要素である「MT 2 1」の子孫は先祖リストC 1に存在しないので一時子孫リストD 0 0は空集合となる。そのためステップS 6 0 7における重複リストI S 0 0も空集合となり、S D 0 0はゼロである。

[0083] 続くステップS 6 0 9では、第1スタックS 1の「 j 」番目、すなわち「1」番目の要素である「MT 2 1」の出次数「1」をO D 0 0とする。続くステップS 6 1 0では、第1スタックS 1の「 $i + 1$ 」番目、すなわち「2」番目の要素である「MT 2 2」の入次数「2」をI D 0 0とする。続くステップS 6 1 0 Aでは、 $O D 0 0 + S D 0 0 - I D 0 0$ の値を「 -1 」と算出し、これがゼロ未満なので否定判断してステップS 6 1 3に進む。ステップS 6 1 3では変数 j をデクリメントして「1」から「0」に更新する。そしてステップS 6 0 5に戻る。以上説明した図8の2週目の処理では、第2

スタック S 2 に変動はない。

[0084] (構造解析処理の動作例 | 図 8 の 3 週目)

3 回目の実行となるステップ S 6 0 5 では、変数 j の値がゼロであり「-1」よりも大きいので肯定判断してステップ S 6 0 6 に進む。

[0085] ステップ S 6 0 6 では、第 1 スタック S 1 の「j」番目、すなわち「0」番目の要素である「MT 1 4」の子孫を先祖リスト C 1 から抽出し、「MT 1 8、MT 1 9、MT 2 1」を一時子孫リスト D 0 0 とする。続くステップ S 6 0 7 では、一時子孫リスト D 0 0 と一時先祖リスト A 0 0 に共通する「MT 1 8、MT 1 9」を重複リスト I S 0 0 とする。続くステップ S 6 0 8 では、MT 1 8 および MT 1 9 の入次数と出次数の差の総和であるゼロを S D 0 0 とする。

[0086] 続くステップ S 6 0 9 では、第 1 スタック S 1 の「j」番目、すなわち「0」番目の要素である「MT 1 4」の出次数「4」を O D 0 0 とする。続くステップ S 6 1 0 では、第 1 スタック S 1 の「i + 1」番目、すなわち「2」番目の要素である「MT 2 2」の入次数「2」を I D 0 0 とする。続くステップ S 6 1 0 A では、 $OD00 + SD00 - ID00$ の値を「2」と算出し、これがゼロ以上なので肯定判断してステップ S 6 1 1 に進む。

[0087] ステップ S 6 1 1 では、第 2 スタック S 2 に第 1 スタック S 1 の「i + 1」番目、すなわち「2」番目の要素である「MT 2 2」を挿入する。続くステップ S 6 1 2 では、第 2 スタック S 2 に第 1 スタック S 1 の「j」番目、すなわち「0」番目の要素である「MT 1 4」を挿入する。続くステップ S 6 1 4 では変数 i をインクリメントして「2」としてステップ S 6 0 2 に戻る。以上説明した図 8 の 3 週目の処理により、第 2 スタック S 2 は「MT 2 1、MT 1 4、MT 2 2、MT 1 4」に更新された。

[0088] ステップ S 6 0 2 では、第 1 スタック S 1 の長さ「3」が「i + 1」すなわち「3」よりも長くないので、否定判断して構造解析処理を終了する。以上説明した処理により、第 2 スタック S 2 として「MT 2 1、MT 1 4、MT 2 2、MT 1 4」が得られる。

[0089] (テーブル作成処理のフローチャート)

図9は、テスト演算部22によるテーブル作成処理を示すフローチャートである。テーブル作成処理は、構造解析処理が完了していれば実行可能であり、メモリダンプ処理と実行の前後は不問である。ただしテーブル作成処理は、プログラムテストよりも先に実行が完了している必要がある。

[0090] 図9ではまずテスト演算部22は、ステップS200において図6～図8に示した構造解析処理により作成された第2スタックS2をコピーする。後述するようにテーブル作成処理では第2スタックS2に対してポップ操作を行い第2スタックS2が変化するので、これがメモリダンプ処理などの他の処理に影響を与えないように、コピーした第2スタックS2を以下の処理では用いる。テスト演算部22は、ステップS200の処理が完了するとステップS201～S203を並列処理する。ただしステップS201～S203は並列に処理することは必須ではなく、3つを順番に処理してもよい。

[0091] ステップS201ではテスト演算部22は、プログラムP1を読み込む。ステップS202ではテスト演算部22は、プログラムP1の有向グラフD1を取得する。ステップS203ではテスト演算部22は、注目点Xの先祖リストC1を作成する。なおステップS203における注目点Xは、構造解析処理のステップS103において決定された注目点Xである。

[0092] テスト演算部22は、ステップS201～S203の全ての実行が完了するとステップS204に進む。ステップS204ではテスト演算部22は、第2スタックS2の長さは2よりも長いかな否かを判断する。テスト演算部22は、第2スタックS2の長さが2以上であると判断するとステップS205に進み、第2スタックS2の長さが2未満であると判断すると図9に示す処理を終了する。

[0093] ステップS205ではテスト演算部22は、第2スタックS2に対してポップ処理を行い、取得したTMを第1要素Gと名付ける。続くステップS205ではテスト演算部22は、第2スタックS2に対して再度のポップ処理を行い、取得したTMを第2要素Hと名付ける。すなわちステップS205

およびS 2 0 6 の処理により第2スタックS 2 から2つの要素が削除される。続くステップS 2 0 7 ではテスト演算部2 2 は、プログラムP 1 の第1要素Gから第2要素Hまでを対象にして、並列化アルゴリズムを適用する。本ステップにおいて適用する並列化アルゴリズムに制限はなく、公知の様々なアルゴリズムを適用できる。

[0094] 続くステップS 2 0 8 ではテスト演算部2 2 は、先祖リストC 1 に基づき第1要素Gから第2要素Hまでの部分先祖リストP D i を作成する。すなわち先祖リストC 1 における第1要素Gから第2要素Hまでの範囲を部分先祖リストP D i と名付ける。続くステップS 2 0 9 ではテスト演算部2 2 は、プログラムP 1 の第1要素Gから第2要素Hまでの並列化されたバイナリB i を生成する。

[0095] 続くステップS 2 1 0 ではテスト演算部2 2 は、ステップS 2 0 8 において作成した部分先祖リストP D i をエリアテーブルT 1 0 0 のT 1 0 8 の欄に保存し、ステップS 2 0 9 において作成したバイナリB i をエリアテーブルT 1 0 0 のT 1 0 3 の欄に保存する。続くステップS 2 1 0 Aではテスト演算部2 2 は、第1要素Gから第2要素HまでをエリアA i としてエリアテーブルT 1 0 0 のT 1 0 2 の欄に保存する。

[0096] 続くステップS 2 1 1 ではテスト演算部2 2 は、エリアテーブルT 1 0 0 のT 1 0 5 の欄に、第2要素Hの最初の命令を記録する。続くステップS 2 1 2 ではテスト演算部2 2 は、エリアテーブルT 1 0 0 のT 1 0 4 の欄に第1要素Gの最後の命令を記録してステップS 2 0 4 に戻る。ここで記録した2つの命令の間は、状況によりマクロタスクの実行順番が入れ替わるので、後のテストでは繰り返し実行を行う。そのため、繰り返し実行すべき始点と終点を記録するためにステップS 2 1 1 ~ S 2 1 2 が実行される。第2スタックS 2 は、ステップS 2 0 5 ~ S 2 1 2 の処理を1回行うごとに要素が2ずつ減少し、第2スタックS 2 の長さが2未満になると図9の処理が終了する。

[0097] なおステップS 2 1 1 において第2要素Hの最後の命令を記録してもよい

し、ステップS 2 1 2において第1要素Gの最初の命令を記録してもよい。ただし第1要素Gおよび第2要素Hの内部における処理の順番は固定なので、順番が固定の箇所まで繰り返し実行することはテストの効率を悪化させる。したがって、フローチャートに記載したように、第1要素Gの最後の命令を記録し、第2要素Hの最初の命令を記録することが望ましい。

[0098] (テーブル作成処理の動作例)

上述した構造解析処理の動作例において作成した第2スタックS 2を用いるテーブル作成処理の動作例を説明する。以下に説明する動作例では動作主体の記載を省略するが、動作主体は全てテスト演算部22である。ステップS 2 0 0では、先頭から順番に「MT 2 1、MT 1 4、MT 2 2、MT 1 4」の4つが格納された第2スタックS 2をコピーする。ステップS 2 0 1～S 2 0 2の説明は省略する。ステップS 2 0 3では注目点Xが上述した動作例と同様に「MT 2 2」なので、作成される先祖リストC 1は上記と同様に「MT 6、MT 9、MT 1 4、MT 1 7、MT 1 8、MT 1 9、MT 2 1」である。

[0099] 次に実行される1回目のステップS 2 0 4では、第2スタックS 2の長さは「4」なので2以上であることから肯定判断する。そしてステップS 2 0 5では最後に入力された「MT 1 4」をポップして第1要素Gとし、ステップS 2 0 6では「MT 2 2」をポップして第2要素Hとする。1回目のステップS 2 0 5～S 2 0 6の処理により、第2スタックS 2の長さは「4」から「2」になる。

[0100] 続くステップS 2 0 7では、MT 1 4～MT 2 2に並列化アルゴリズムを適用し、ステップS 2 0 8ではMT 1 4～MT 2 2の部分先祖リストPD iを作成する。なお部分先祖リストPD iには、人間が理解しやすいように図式化した有向グラフの情報が含まれてもよいし、コンピュータによる処理が容易なように論理的に記載した有向グラフの情報が含まれてもよい。

[0101] 続くステップS 2 0 9ではMT 1 4～MT 2 2の並列化されたバイナリB iを作成する。ステップS 2 1 0では、部分先祖リストPD iとバイナリB

i をエリアテーブル T 1 0 0 に保存する。ステップ S 2 1 0 A では、MT 1 4 ~ MT 2 2 に新たなエリア ID を付してエリアテーブル T 1 0 0 の最初のレコードに保存する。続くステップ S 2 1 1 では、MT 2 2 の最初の命令を、続くステップ S 2 1 2 では MT 1 4 の最後の命令を、エリアテーブル T 1 0 0 の最初のレコードに保存する。

[0102] 2 回目のステップ S 2 0 4 では、第 2 スタック S 2 の長さは「2」なので肯定判断してステップ S 2 0 5 に進む。ステップ S 2 0 5 では「MT 1 4」が第 1 要素 G に設定され、ステップ S 2 0 6 では「MT 2 1」が第 2 要素 H に設定される。1 回目のステップ S 2 0 5 ~ S 2 0 6 の処理により、第 2 スタック S 2 の長さは「2」から「0」になる。以下のステップ S 2 0 7 ~ S 2 1 2 では「MT 1 4」~「MT 2 1」を対象として、エリアテーブル T 1 0 0 の 2 行目に書き込みが行われる。個別の説明は省略する。

[0103] 3 回目のステップ S 2 0 4 では、第 2 スタック S 2 の長さは「0」なので否定判断してテーブル作成処理を終了する。以上がテーブル作成処理の動作例の説明である。

[0104] (メモリダンプ処理のフローチャート)

図 1 0 は、テスト演算部 2 2 によるメモリダンプ処理を示すフローチャートである。メモリダンプ処理は、構造解析処理が完了していれば実行可能であり、テーブル作成処理と実行の前後は不問である。ただしメモリダンプ処理は、プログラムテストよりも先に実行が完了している必要がある。

[0105] 図 1 0 ではまずステップ S 3 0 0 において、テスト演算部 2 2 は構造解析処理により作成された第 2 スタック S 2 をコピーする。後述するように図 1 0 に示す処理では第 2 スタック S 2 に対してポップ操作を行い第 2 スタック S 2 が変化するので、これがテーブル作成処理などの他の処理に影響を与えないように、コピーした第 2 スタック S 2 を以下の処理では用いる。続くステップ S 3 0 1 ではテスト演算部 2 2 は、プログラム P 1 を読み込んでステップ S 3 0 2 に進む。

[0106] ステップ S 3 0 2 ではテスト演算部 2 2 は、第 2 スタック S 2 の長さが 2

よりも長いかな否かを判断する。テスト演算部 22 は、第 2 スタック S 2 の長さが 2 以上であると判断するとステップ S 303 に進み、第 2 スタック S 2 の長さが 2 未満であると判断するとステップ S 307 に進む。ステップ S 303 ではテスト演算部 22 は、第 2 スタック S 2 に対してポップ処理を行い、取得したマクロタスクを第 3 要素 M と名付ける。続くステップ S 304 ではテスト演算部 22 は、第 3 要素 M の最後をメモリダンプのための信号位置に設定する。続くステップ S 305 ではテスト演算部 22 は、第 2 スタック S 2 に対して再度のポップ処理を行い、取得した TM を第 4 要素 N と名付ける。続くステップ S 306 ではテスト演算部 22 は、第 4 要素 N の TM の最初をメモリダンプのための信号位置に設定してステップ S 302 に戻る。すなわちステップ S 303 および S 305 の処理により第 2 スタック S 2 から 2 つの要素が削除される。

[0107] ステップ S 307 ではテスト演算部 22 は、プログラム P 1 を実行する。そしてステップ S 308 ではテスト演算部 22 は、プログラム P 1 がステップ S 304 およびステップ S 306 において設定した信号位置に到達するたびにメモリダンプを取得する。続くステップ S 309 ではテスト演算部 22 は、ステップ S 308 において取得したメモリダンプをエリアテーブルの T 106 と T 107 の欄に保存して図 10 に示す処理を終了する。具体的には、ステップ S 304 に対応する信号位置で取得したメモリダンプは T 106 に記録し、ステップ S 306 に対応する信号位置で取得したメモリダンプは T 107 に記録する。

[0108] (メモリダンプ処理の動作例)

上述した構造解析処理の動作例において作成した第 2 スタック S 2 を用いるメモリダンプ処理の動作例を説明する。以下に説明する動作例では動作主体の記載を省略するが、動作主体は全てテスト演算部 22 である。ステップ S 300 では、先頭から順番に「MT 21、MT 14、MT 22、MT 14」の 4 つが格納された第 2 スタック S 2 をコピーする。ステップ S 301 ではプログラム P 1 を読み込む。

[0109] 1回目のステップS302では、第2スタックS2の長さは「4」なので2以上であることから肯定判断する。そしてステップS303では最後に入力された「MT14」をポップして第3要素Mとし、ステップS304では「MT14」の最後の命令を信号位置に設定する。ステップS305では「MT22」をポップして第4要素Nとし、ステップS306では「MT22」の最初の命令を信号位置に設定する。1回目のステップS303、S305の処理により、第2スタックS2の長さは「4」から「2」になる。

[0110] 2回目のステップS302では、第2スタックS2の長さは「2」なので2以上であることから肯定判断する。そしてステップS303では「MT14」をポップして第3要素Mとし、ステップS304では「MT14」の最後の命令を信号位置に設定する。ステップS305では「MT21」をポップして第4要素Nとし、ステップS306では「MT21」の最初の命令を信号位置に設定する。1回目のステップS303、S305の処理により、第2スタックS2の長さは「2」から「0」になる。

[0111] 3回目のステップS302では、第2スタックS2の長さは「0」なので2以上はないから否定判断してステップS307に進む。ステップS307ではプログラムP1を実行し、ステップS308では信号位置に到達するたびにメモリダンプ、換言するとメモリのスナップショットを取得する。そしてステップS309では取得したメモリダンプをエリアテーブルT100に保存してメモリダンプ処理を終了する。

[0112] (プログラムテストのフローチャート)

図11は、テスト演算部22によるプログラムテストを示すフローチャートである。図11ではテスト演算部22は、まずステップS401においてエリアテーブルT100から対象とするエリアA_iを選択する。この選択は、オペレータによるマニュアル操作に基づき行われてもよいし、所定のルールにより自動処理により選択されてもよい。所定のルールによる自動処理とはたとえば、エリアテーブルT100に記載されたエリアA_iを上から順番に1つつ選択することである。

- [0113] 続くステップS402ではテスト演算部22は、ステップS401において選択されたエリアA_iをヒートランするか否かを決定する。テスト演算部22は、ヒートランすると判断する場合はステップS403に進み、ヒートランしないと判断する場合は丸囲みのDを経由して図12のS501に進む。なおステップS402の判断はたとえば、オペレータによる事前の設定に基づき行われる。ステップS403ではテスト演算部22は、メモリダンプを使用するか否かを決定する。テスト演算部22は、メモリダンプを使用すると判断する場合はステップS404に進み、メモリダンプを使用しないと判断する場合はステップS410に進む。なおステップS403の判断はたとえば、オペレータによる事前の設定に基づき行われる。
- [0114] ステップS404ではテスト演算部22は、エリアテーブルT100のT106の欄を読み取りステップS405に進む。ステップS405ではテスト演算部22は、エリアテーブルT100のT103の欄から並列化済バイナリB_iを取得してステップS406に進む。ステップS406ではテスト演算部22は、メモリダンプを用いて並列化済バイナリB_iを初期化する。続くステップS407ではテスト演算部22は、並列化済バイナリB_iの状態、換言するとメモリに格納される情報に必要な改変を加える。この改変とは、バイナリB_iの実行結果を所定の閾値に達するようにするためのものであり、たとえば特定の変数の値を書き換えることである。書き換えではたとえば、初回にS407が実行される際には「1」だけ大きな値に変更し、2回目のS407では「2」だけ大きな値に変更し、それ以後もS407の実行回数に合わせて変数の値を更新する。
- [0115] 続くステップS408ではテスト演算部22は、並列化済バイナリB_iを実行し、続くステップS409ではテスト演算部22は、並列化済バイナリB_iの実行結果を取得する。続くステップS410ではテスト演算部22は、並列化済バイナリB_iの実行結果が所定の閾値に達したか否かを判断する。テスト演算部22は、所定の閾値に達したと判断する場合は図11に示す処理を終了し、所定の閾値に達していないと判断する場合はステップS40

6に戻る。

[0116] ステップS 4 0 3において否定判断されると実行されるステップS 4 1 0ではテスト演算部2 2は、プログラムP 1を読み込む。続くステップS 4 1 1ではテスト演算部2 2は、エリアテーブルT 1 0 0から選択したエリアA_iに対応するT 1 0 4の欄を読み取り、以下ではこの欄に記載のマクロタスクをループ開始要素Fと呼ぶ。

[0117] 続くステップS 4 1 2ではテスト演算部2 2は、ループ開始要素FにおいてプログラムP 1が停止するように設定する。続くステップS 4 1 3ではテスト演算部2 2はプログラムP 1を実行する。続くステップS 4 1 4ではテスト演算部2 2は、ループ開始要素Fにおいて停止したプログラムP 1のメモリダンプを取得してステップS 4 0 5に進む。以上が図1 1の説明である。

[0118] 図1 2は、図1 1におけるステップS 4 0 2において否定判断されると実行される処理を示すフローチャートである。ステップS 4 0 2において否定判断されると実行されるステップS 5 0 1ではテスト演算部2 2は、エリアテーブルT 1 0 0のT 1 0 8の欄の情報を読み込む。続くステップS 5 0 2ではテスト演算部2 2は、オペレータによる有向グラフに関する追加要求を読み込む。続くステップS 5 0 3ではテスト演算部2 2は、ステップS 5 0 2において読み込んだ追加要求に応じて有向グラフを加工する。

[0119] ステップS 5 0 2～S 5 0 3における第1の具体例は、マクロタスクの実行順番が特定、より具体的には「MT 1 7」が3番目に実行される実行順番だけに限定されることである。第2の具体例は、オペレータから指定されたマクロタスクを指定された位置に追加することである。

[0120] 続くステップS 5 0 4ではテスト演算部2 2は、ステップS 5 0 3において加工された有向グラフD 1から全ての経路を抽出する。続くステップS 5 0 5ではテスト演算部2 2は、それぞれの経路に対応するバイナリPB_iを作成し、丸囲みのEを経由して図1 3に進む。

[0121] 図1 3は、図1 2におけるステップS 5 0 5の後に実行される処理を示す

フローチャートである。ステップS 5 0 5の次に実行されるステップS 5 0 6ではテスト演算部2 2は、メモリダンプを使用するか否かを判断する。テスト演算部2 2は、メモリダンプを使用すると判断する場合はステップS 5 0 7に進み、メモリダンプを使用しないと判断する場合はステップS 5 1 3に進む。

[0122] ステップS 5 0 7ではテスト演算部2 2は、エリアテーブルT 1 0 0のT 1 0 6の欄からメモリダンプを取得してステップS 5 0 8に進む。ステップS 5 0 8ではテスト演算部2 2は、ステップS 5 0 7において取得したメモリダンプを用いてバイナリP B iを初期化する。続くステップS 5 0 9ではテスト演算部2 2は、必要に応じて入力进行操作する。この操作はたとえばメモリの値を書き換えることである。続くステップS 5 1 0ではテスト演算部2 2は、バイナリP B iを実行する。続くステップS 5 1 1ではテスト演算部2 2は、ステップS 5 1 0において実行したバイナリP B iの実行結果を確認する。

[0123] 続くステップS 5 1 2ではテスト演算部2 2は、実行すべきバイナリP B iがまだ存在するか否かを判断する。テスト演算部2 2は、実行すべきバイナリP B iがまだ存在すると判断する場合はステップS 5 0 8に戻り、実行すべきバイナリP B iが存在しないと判断する場合は図1 3に示す処理を終了する。

[0124] ステップS 5 0 6において否定判断されると実行されるステップS 5 1 3ではテスト演算部2 2は、エリアテーブルT 1 0 0におけるT 1 0 4の欄に記載されたマクロタスクを読み取り、要素Eとする。続くステップS 5 1 4ではテスト演算部2 2は、要素EにおいてプログラムP 1が停止するように設定する。続くステップS 5 1 5ではテスト演算部2 2は、プログラムP 1を実行する。続くステップS 5 1 6ではテスト演算部2 2は、要素Eにおいて停止したプログラムP 1のメモリダンプを取得してステップS 5 0 8に進む。以上が図1 3の処理である。

[0125] (プログラムテストの動作例)

以下ではプログラムテストの複数の動作例を説明する。ただし以下では動作の概略を説明するにとどめ、詳細な説明は省略する。また、ヒートランの有無、およびメモリダンプの使用の有無は、動作例ごとにあらかじめオペレータから指示されている。以下に説明する動作例では動作主体の記載を省略するが、動作主体は全てテスト演算部22である。

[0126] (プログラムテストの動作例1)

プログラムテストの動作例1を説明する。この動作例1では、ヒートランを行わず、メモリダンプを使用しない。動作例1では、ヒートランを行わないのでステップS402において否定判断し、丸囲みのDを経由して図12のステップS501に進む。そしてステップS501～S505を実行し、丸囲みのEを経由して図13のステップS506に進む。ステップS506では、メモリダンプを使用しないので否定判断してステップS513に進む。

[0127] ステップS513～S516では、エリアテーブルT100の開始タスクT104の欄を読み取り要素Fとし、プログラムP1が要素Fに達した際のメモリダンプを取得する。ステップS508～S511ではS516において取得したメモリダンプを用いてバイナリPB_iを初期化して実行する。実行すべきバイナリPB_iは、ステップS504～S505において有向グラフの全ての経路に対応して作成され、その全てを実行するまではステップS508～S511の処理が繰り返される。

[0128] (プログラムテストの動作例2)

プログラムテストの動作例2を説明する。この動作例2では、ヒートランを行わず、メモリダンプを使用する。動作例2は、図13に達するまでは動作例1と同様なのでステップS506までの処理の説明は省略する。ステップS506ではメモリダンプを使用するので肯定判断し、ステップS507においてエリアテーブルT100の開始点ダンプT106の欄からメモリダンプを取得する。ステップS508以降の処理は、動作例1と同様なので説明を省略する。

[0129] (プログラムテストの動作例3)

プログラムテストの第3の動作例を説明する。この第3の動作例では、ヒートランを行ない、メモリダンプを使用しない。動作例3では、ヒートランを行なうのでステップS402において肯定判断し、メモリダンプを使用しないのでステップS403では否定判断してステップS410に進む。

ステップS410～S414では、プログラムP1を読み込み、エリアテーブルT100の開始タスクT104の欄を読み取って要素Fとして、プログラムP1が要素Fに達した際のメモリダンプを取得する。続くステップS405～S06では、エリアテーブルT100から並列化したバイナリB_iを取得し、メモリダンプを使用してバイナリB_iを初期化する。ステップS407では必要な改変を加えてからバイナリB_iを実行し、その実行結果が所望の閾値に達するまでS407における改変を変更しながら繰り返し実行する。そして所望の閾値に達するとテストを終了する。

[0130] (プログラムテストの動作例4)

プログラムテストの第4の動作例を説明する。この第4の動作例では、ヒートランを行ない、メモリダンプを使用する。動作例4では、ヒートランを行なうのでステップS402において肯定判断し、メモリダンプを使用するのでステップS403では肯定判断してステップS404に進む。ステップS404ではエリアテーブルT100からメモリダンプを読み込んでステップS405に進む。ステップS405以降の処理は、動作例3と同様なので説明を省略する。

[0131] 上述した第1の実施の形態によれば、次の作用効果が得られる。

(1) 演算装置1が実行するテスト方法は、実行される順番が複数とおり存在し得る複数のマクロタスクを含むプログラムをコンピュータである演算装置1がテストするテスト方法である。この方法は、複数のマクロタスクのいずれかを注目点に設定することと(図6のS103)、注目点Xに影響を与えるマクロタスクの集合である先祖リストC1を特定すること(図6のS104)と、先祖リストC1に含まれるマクロタスクを、並列化に寄与しない

非並列群および並列化に寄与する並列寄与群に分別すること（構造解析処理）と、並列寄与群についてテストパターンを複数生成すること（図12のS501～S504）と、非並列群に属するマクロタスクを実行して得られた結果を用いて、複数のテストパターンを実行すること（図11のS406～S408、図13のS508～S510）、を含む。そのため、実行される順番が複数とおり存在し得る複数のマクロタスクを含むプログラムP1のテストにおいて、注目点Xに設定された特定のマクロタスクを対象としたテストが効率的に実行できる。

[0132] （2）非並列群に属するマクロタスクを実行して得られた結果とは、マクロタスクを実行するコンピュータのメモリのダンプであり、1つのダンプを用いて、複数のテストパターンを実行する（図11のS410：NOのループ、図13のS512：YESのループ）。そのため、同一の処理を何度も実行する代わりにメモリダンプを利用するので、複数のテストパターンを実行する時間を短縮できる。

[0133] （3）演算装置1が実行するテスト方法は、生成されたテストパターンをオペレータの指令に基づき修正することを含む（図12のS502～S503）。そのため、自動生成されたテストパターンだけでなく、オペレータが気になる修正を任意に加えることができる。

[0134] （4）演算装置1が実行するテスト方法は、プログラムを実行してエラーが発生したマクロタスクを注目点Xに設定する（図6のS103）。そのため、オペレータによる指定を必須とせず、テスト全体を自動で実行できる。

[0135] 上述した各実施の形態および変形例において、機能ブロックの構成は一例に過ぎない。別々の機能ブロックとして示したいくつかの機能構成を一体に構成してもよいし、1つの機能ブロック図で表した構成を2以上の機能に分割してもよい。また各機能ブロックが有する機能の一部を他の機能ブロックが備える構成としてもよい。

[0136] 上述した各実施の形態および変形例において、演算装置1が実行するプログラムはROM3に格納されたとしたが、プログラムは記憶部5に格納され

ていてもよい。また、演算装置は、必要なときに外部インタフェース 6 が利用可能な媒体を介して、他の装置からプログラムが読み込まれてもよい。ここで媒体とは、例えば入出力インタフェースに着脱可能な記憶媒体、または通信媒体、すなわち有線、無線、光などのネットワーク、または当該ネットワークを伝搬する搬送波やデジタル信号、を指す。また、プログラムにより実現される機能の一部または全部がハードウェア回路や F P G A により実現されてもよい。

[0137] 上述した各実施の形態および変形例は、それぞれ組み合わせてもよい。上記では、種々の実施の形態および変形例を説明したが、本発明はこれらの内容に限定されるものではない。本発明の技術的思想の範囲内で考えられるその他の態様も本発明の範囲内に含まれる。

符号の説明

- [0138] 1 …演算装置
5 …記憶部
2 1 …入出力部
2 2 …テスト演算部
2 3 …プログラム実行部
D 1 …有向グラフ
Q …キュー
S 1 …第 1 スタック
S 2 …第 2 スタック
T 1 0 0 …エリアテーブル

請求の範囲

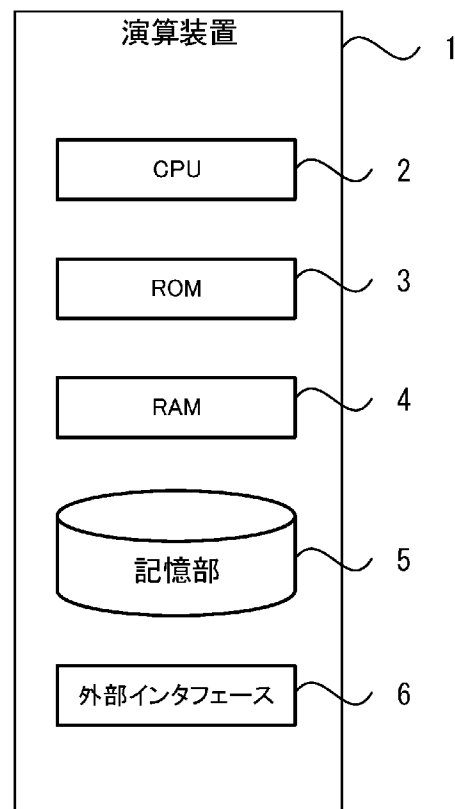
- [請求項1] 実行される順番が複数とおり存在し得る複数のマクロタスクを含むプログラムをコンピュータがテストするテスト方法であって、
前記複数のマクロタスクのいずれかを注目タスクに設定することと、
、
前記注目タスクに影響を与える前記マクロタスクの集合である先祖リストを特定することと、
前記先祖リストに含まれる前記マクロタスクを、並列化に寄与しない非並列群および並列化に寄与する並列寄与群に分別することと、
前記並列寄与群についてテストパタンを複数生成することと、
前記非並列群に属する前記マクロタスクを実行して得られた結果を用いて、前記複数の前記テストパタンを実行することと、を含むテスト方法。
- [請求項2] 請求項1に記載のテスト方法において、
前記非並列群に属する前記マクロタスクを実行して得られた結果とは、前記マクロタスクを実行するコンピュータのメモリのダンプであり、
1つの前記ダンプを用いて、前記複数のテストパタンを実行する、テスト方法。
- [請求項3] 請求項1に記載のテスト方法において、
生成された前記テストパタンをオペレータの指令に基づき修正することをさらに含む、テスト方法。
- [請求項4] 請求項1に記載のテスト方法において、
前記プログラムを実行してエラーが発生した前記マクロタスクを前記注目タスクに設定する、テスト方法。
- [請求項5] 実行される順番が複数とおり存在し得る複数のマクロタスクを含むプログラムを格納する記憶部と、
前記複数のマクロタスクのいずれかを注目タスクに設定し、前記注

目タスクに影響を与える前記マクロタスクの集合である先祖リストを特定し、前記先祖リストに含まれる前記マクロタスクを、並列化に寄与しない非並列群および並列化に寄与する並列寄与群に分別し、前記並列寄与群についてテストパターンを複数生成するテスト演算部と、

前記非並列群に属する前記マクロタスクを実行して得られた結果を用いて、前記複数の前記テストパターンを実行するプログラム実行部と、を含む演算装置。

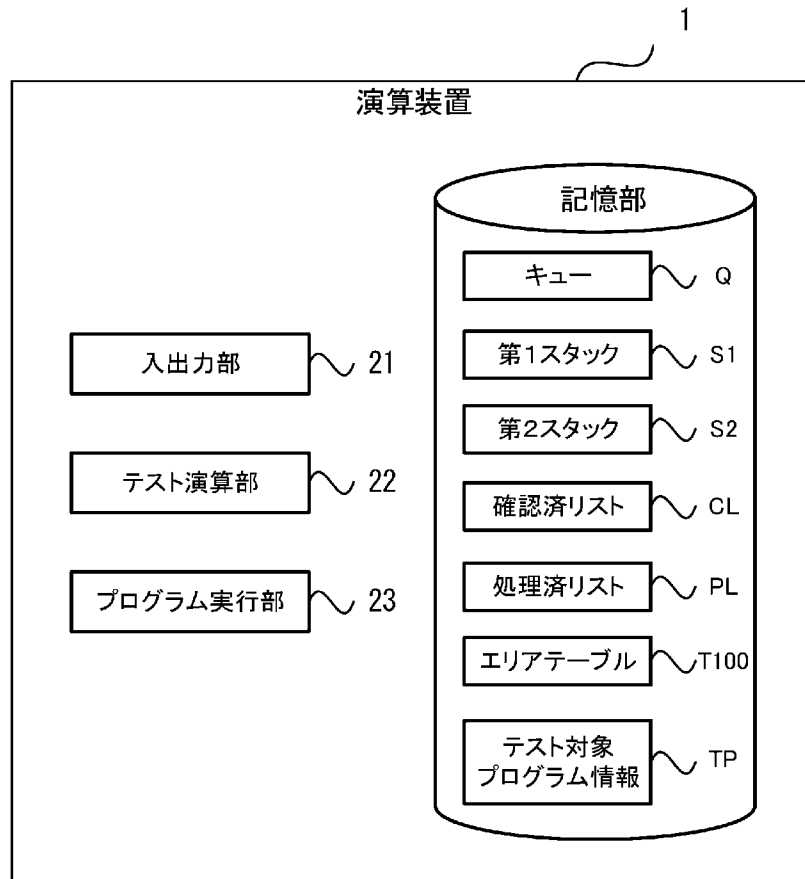
[図1]

図1



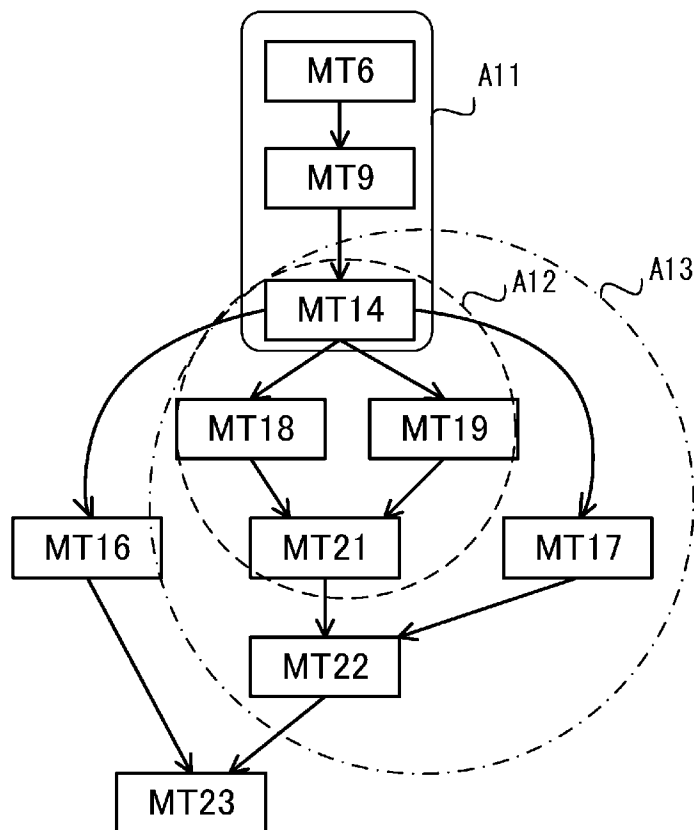
[図2]

図2



[図3]

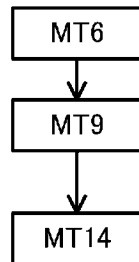
図3



[図5]

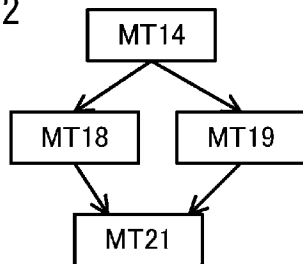
図5

A11



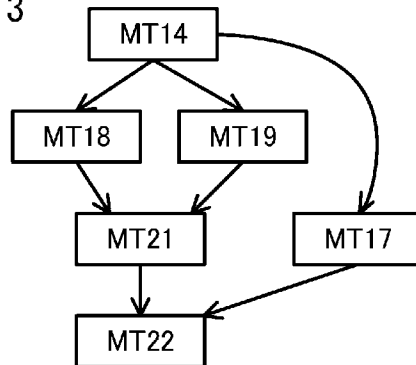
ケース	順位1	順位2	順位3
A11-1	MT6	MT9	MT14

A12



ケース	順位1	順位2	順位3	順位4
A12-1	MT14	MT18	MT19	MT21
A12-2	MT14	MT19	MT18	MT21

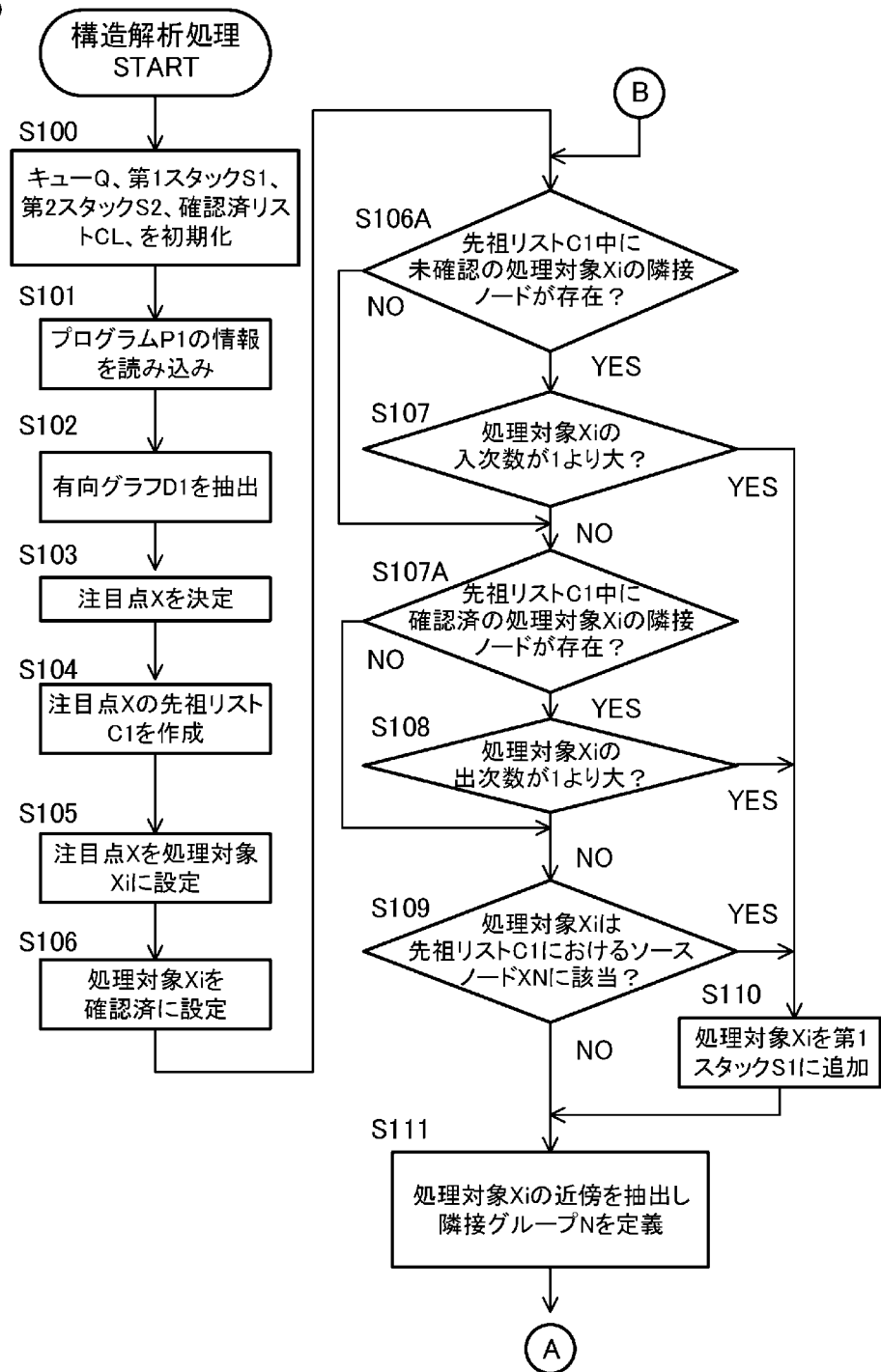
A13



ケース	順位1	順位2	順位3	順位4	順位5	順位6
A13-1	MT14	MT19	MT18	MT21	MT17	MT22
A13-2	MT14	MT19	MT21	MT17	MT21	MT22
A13-3	MT14	MT19	MT17	MT18	MT21	MT22
A13-4	MT14	MT18	MT17	MT19	MT21	MT22
A13-5	MT14	MT18	MT19	MT21	MT17	MT22
A13-6	MT14	MT18	MT19	MT17	MT21	MT22
A13-7	MT14	MT17	MT19	MT18	MT21	MT22
A13-8	MT14	MT17	MT18	MT19	MT21	MT22

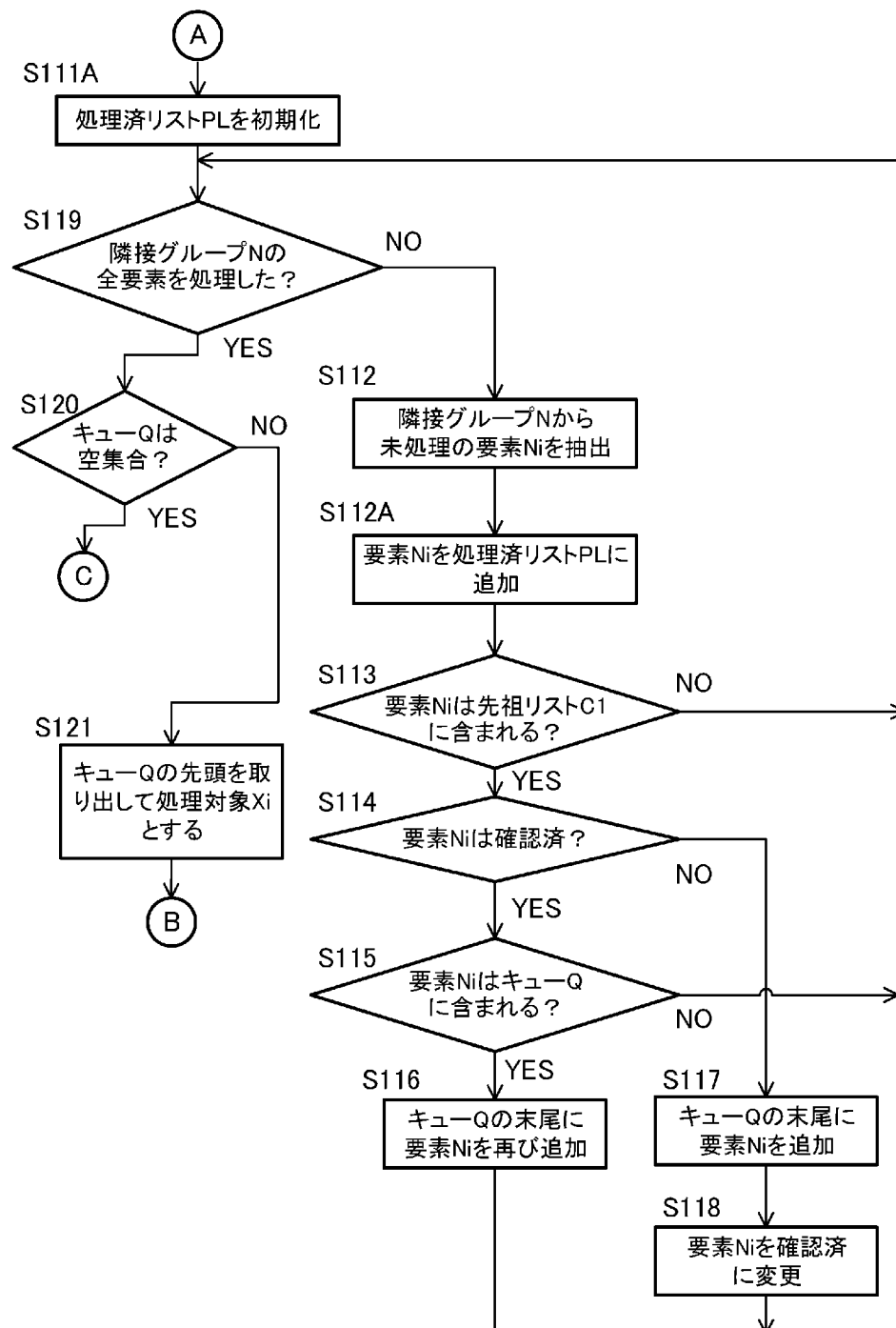
[図6]

図6

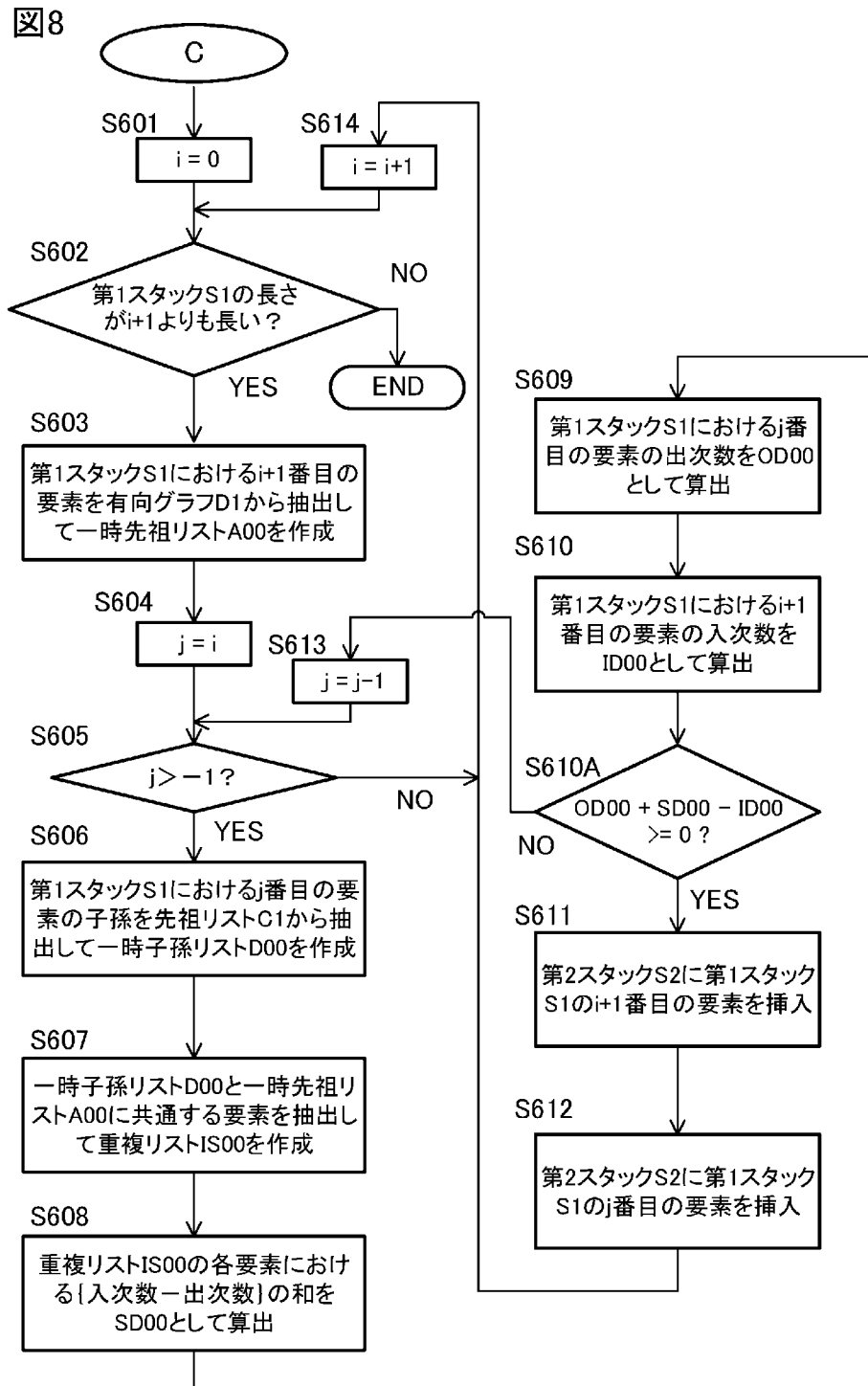


[図7]

図7

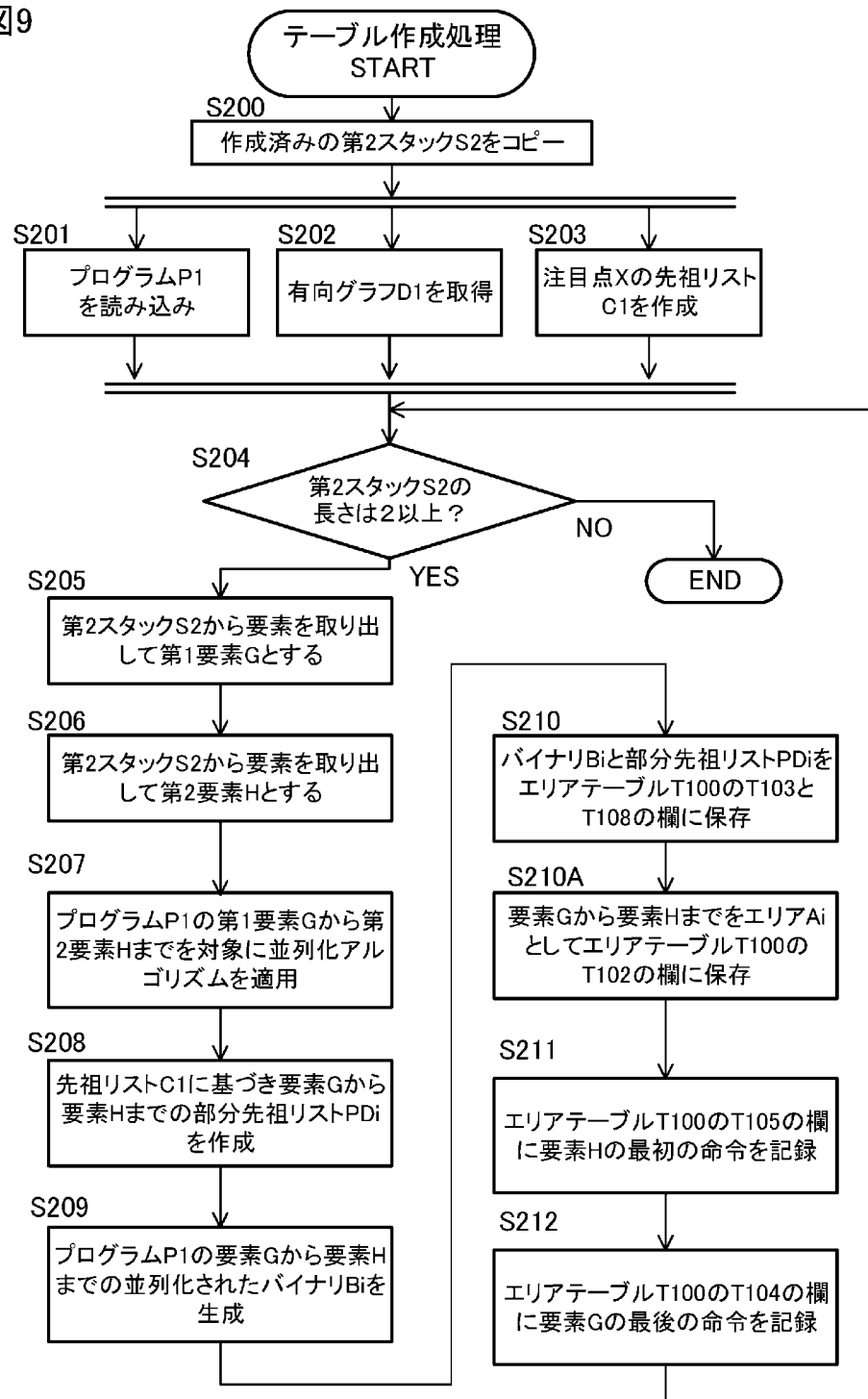


[図8]



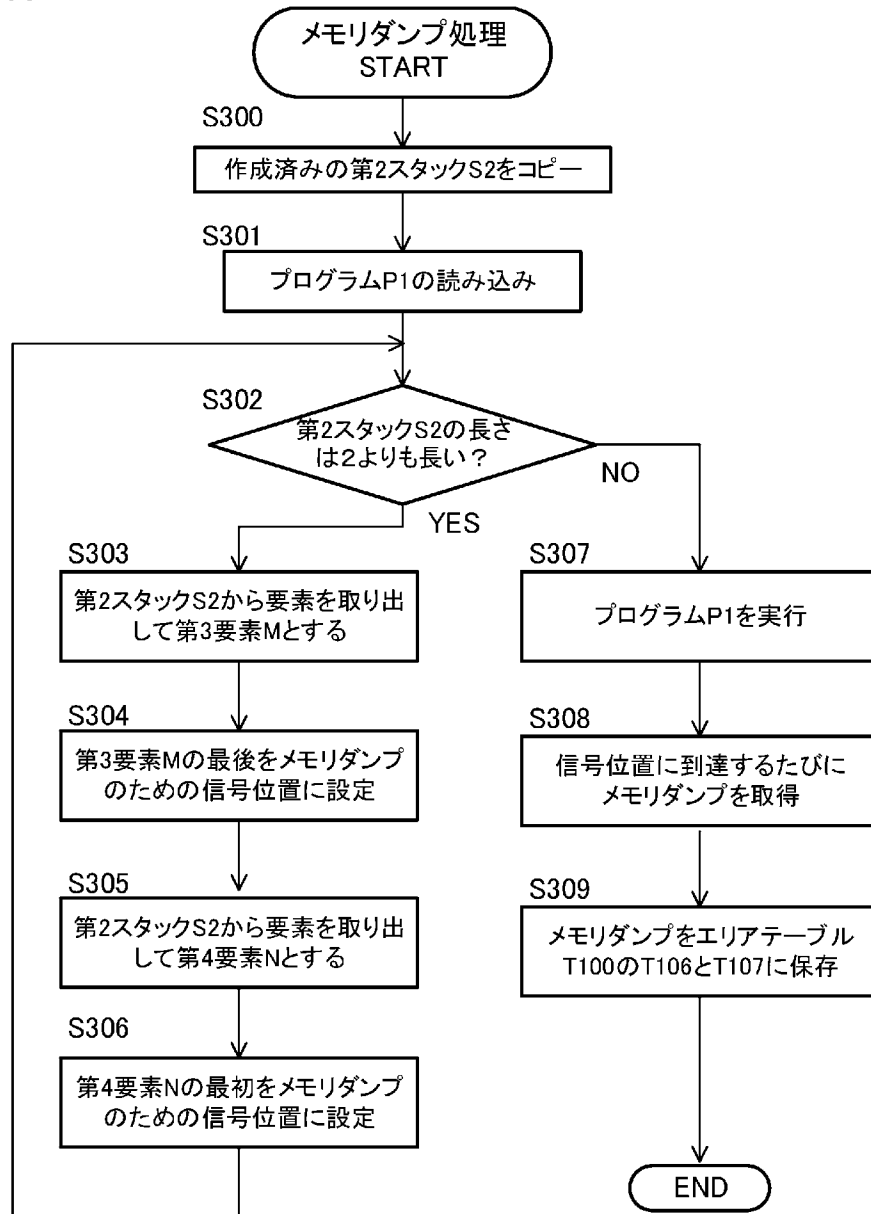
[図9]

図9



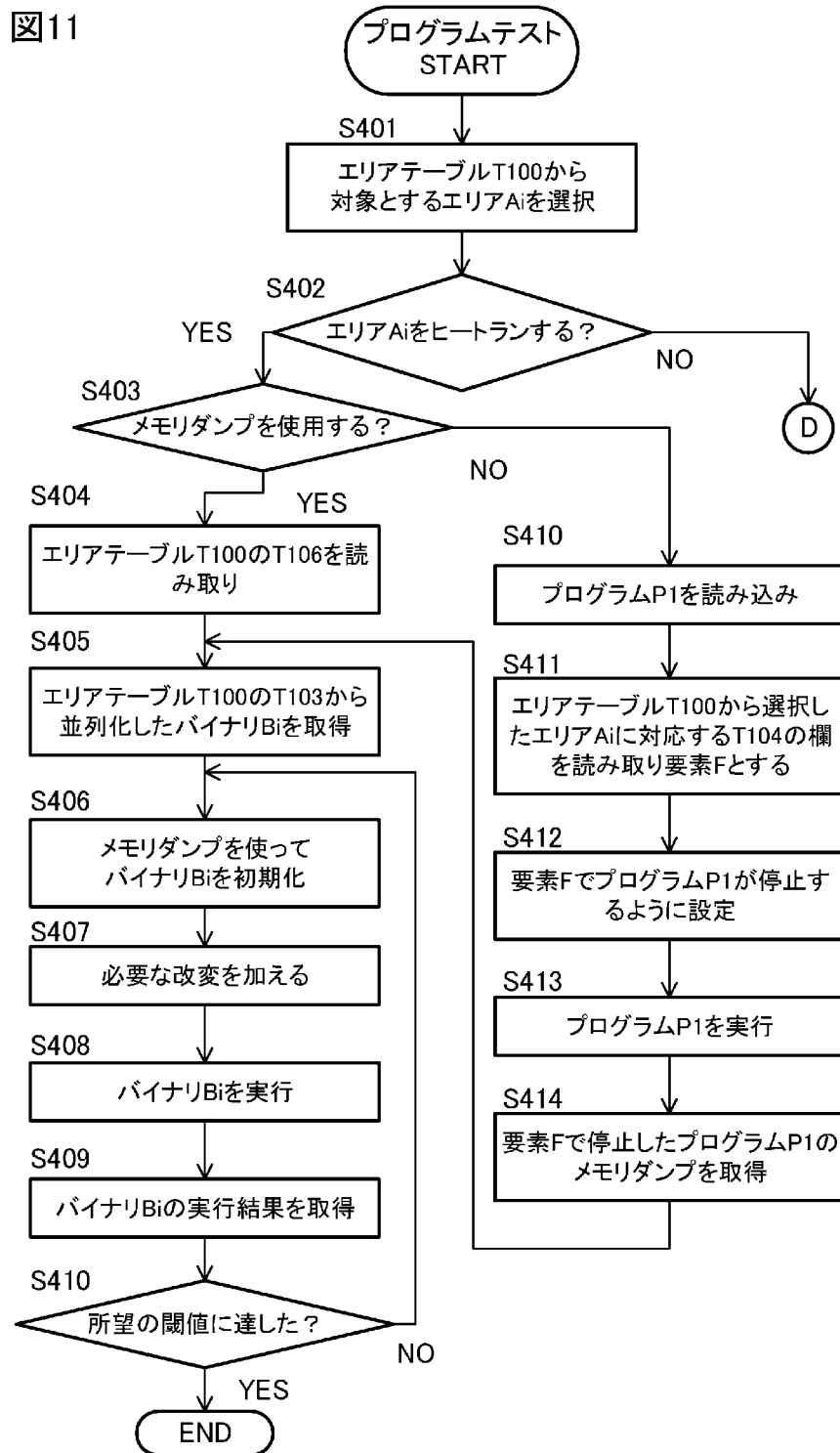
[図10]

図10



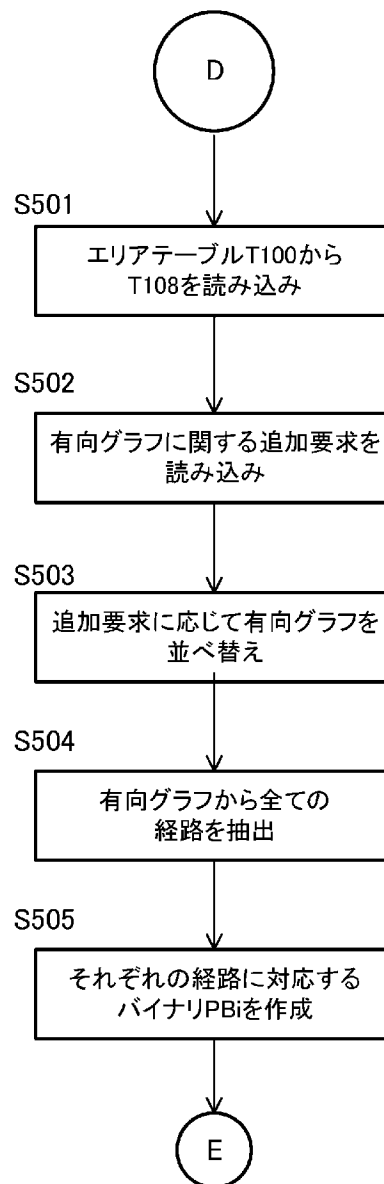
[図11]

図11



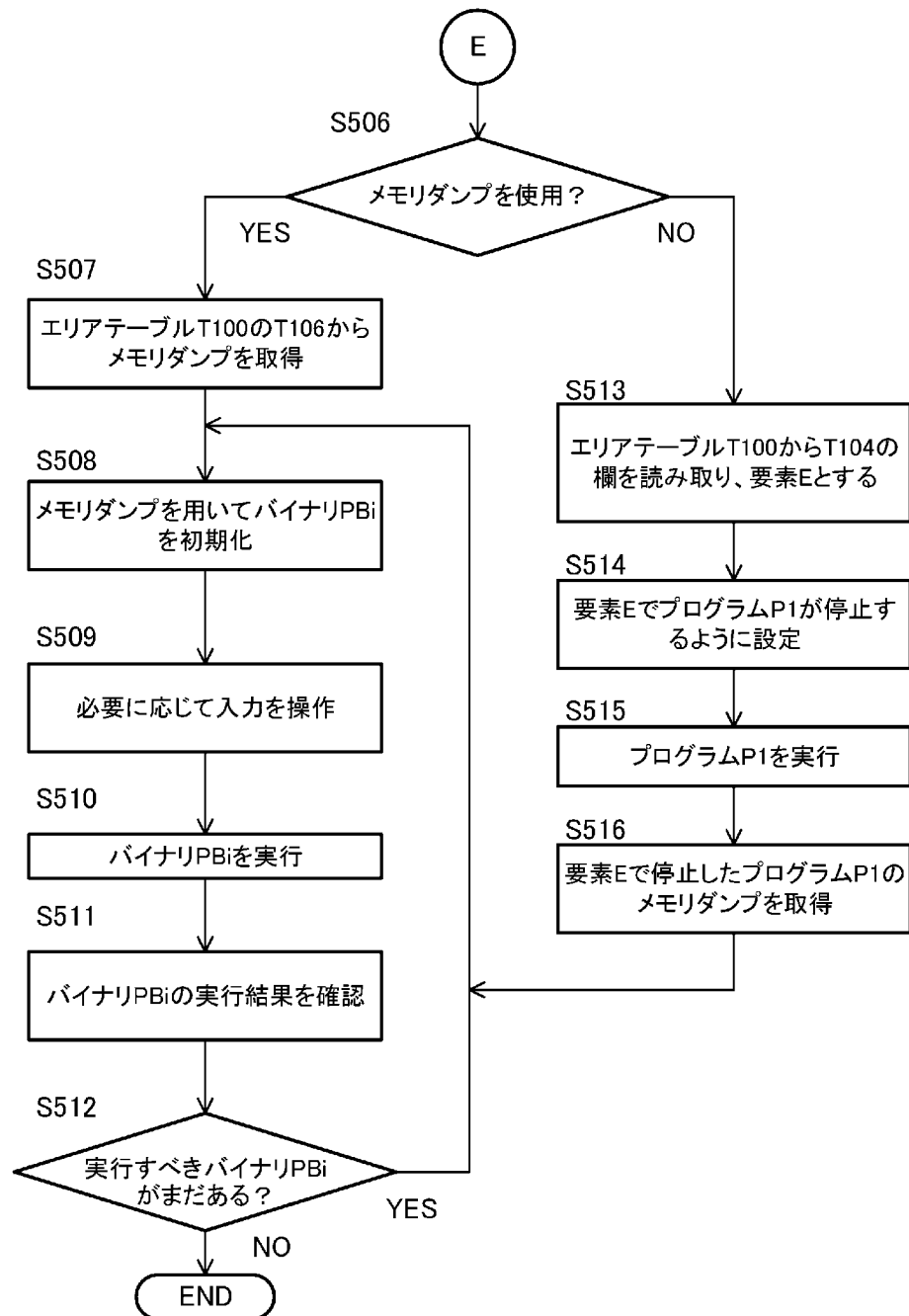
[図12]

図12



[図13]

図13



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2021/025689

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl. G06F11/36 (2006.01) i

FI: G06F11/36 184

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)

Int. Cl. G06F11/36

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan 1922-1996

Published unexamined utility model applications of Japan 1971-2021

Registered utility model specifications of Japan 1996-2021

Published registered utility model applications of Japan 1994-2021

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	JP 11-120030 A (TOSHIBA CORP.) 30 April 1999 (1999-04-30), paragraphs [0057]-[0068] (Family: none)	1-5
Y	本多弘樹ほか, Fortran プログラム粗粒度タスク間の並列性検出手法, 電子情報通信学会論文誌 D, 25 December 1990, vol. J73-D1, No. 12, pp. 951-960, in particular, 2.4, 3 (HONDA, Hiroki et al. Coarse grain parallelism detection scheme of a Fortran program. IEICE Transactions D.)	1-5
A	WO 2018/037787 A1 (HITACHI AUTOMOTIVE SYSTEMS, LTD.) 01 March 2018 (2018-03-01), entire text (Family: none)	1-5

☐

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
01.09.2021Date of mailing of the international search report
14.09.2021Name and mailing address of the ISA/
Japan Patent Office
3-4-3, Kasumigaseki, Chiyoda-ku,
Tokyo 100-8915, Japan

Authorized officer

Telephone No.

A. 発明の属する分野の分類（国際特許分類（IPC）） G06F 11/36(2006.01)i FI: G06F11/36 184														
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）） G06F11/36 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 <table border="0"> <tr> <td>日本国実用新案公報</td> <td>1922 - 1996年</td> </tr> <tr> <td>日本国公開実用新案公報</td> <td>1971 - 2021年</td> </tr> <tr> <td>日本国実用新案登録公報</td> <td>1996 - 2021年</td> </tr> <tr> <td>日本国登録実用新案公報</td> <td>1994 - 2021年</td> </tr> </table> 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）			日本国実用新案公報	1922 - 1996年	日本国公開実用新案公報	1971 - 2021年	日本国実用新案登録公報	1996 - 2021年	日本国登録実用新案公報	1994 - 2021年				
日本国実用新案公報	1922 - 1996年													
日本国公開実用新案公報	1971 - 2021年													
日本国実用新案登録公報	1996 - 2021年													
日本国登録実用新案公報	1994 - 2021年													
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	JP 11-120030 A（株式会社東芝） 30.04.1999（1999 - 04 - 30） 段落[0057]-[0068] （ファミリーなし）	1-5												
Y	本多 弘樹 ほか, Fortranプログラム粗粒度タスク間の並列性検出手法, 電子情報 通信学会論文誌 D, 1990.12.25, Vol. J73-D1 No.12, pp.951-960 特に2.4, 3.等参照	1-5												
A	W0 2018/037787 A1（日立オートモティブシステムズ） 01.03.2018（2018 - 03 - 01） 全文 （ファミリーなし）	1-5												
☐ 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。														
<table border="0"> <tr> <td>* 引用文献のカテゴリー</td> <td>“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抵触するものではなく、発明の原理又は理論の理解のために引用するもの</td> </tr> <tr> <td>“A” 特に関連のある文献ではなく、一般的な技術水準を示すもの</td> <td>“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</td> </tr> <tr> <td>“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</td> <td>“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</td> </tr> <tr> <td>“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）</td> <td>“&” 同一パテントファミリー文献</td> </tr> <tr> <td>“O” 口頭による開示、使用、展示等に言及する文献</td> <td></td> </tr> <tr> <td>“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</td> <td></td> </tr> </table>			* 引用文献のカテゴリー	“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抵触するものではなく、発明の原理又は理論の理解のために引用するもの	“A” 特に関連のある文献ではなく、一般的な技術水準を示すもの	“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	“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	“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	“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）	“&” 同一パテントファミリー文献	“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	
* 引用文献のカテゴリー	“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抵触するものではなく、発明の原理又は理論の理解のために引用するもの													
“A” 特に関連のある文献ではなく、一般的な技術水準を示すもの	“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													
“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	“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													
“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）	“&” 同一パテントファミリー文献													
“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														
国際調査を完了した日 01.09.2021	国際調査報告の発送日 14.09.2021													
名称及びあて先 日本国特許庁(ISA/JP) 〒100-8915 日本国 東京都千代田区霞が関三丁目4番3号	権限のある職員（特許庁審査官） 上田 翔太 5B 4449 電話番号 03-3581-1101 内線 3545													