

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7393358号
(P7393358)

(45)発行日 令和5年12月6日(2023.12.6)

(24)登録日 令和5年11月28日(2023.11.28)

(51)国際特許分類

F I

G 0 6 F 16/332 (2019.01)

G 0 6 F 16/332

G 0 6 F 3/0482 (2013.01)

G 0 6 F 3/0482

請求項の数 19 (全54頁)

(21)出願番号	特願2020-569146(P2020-569146)	(73)特許権者	502303739
(86)(22)出願日	令和1年6月12日(2019.6.12)		オラクル・インターナショナル・コーポレーション
(65)公表番号	特表2021-527268(P2021-527268 A)		アメリカ合衆国カリフォルニア州94065レッドウッド・シティー, オラクル・パークウェイ500
(43)公表日	令和3年10月11日(2021.10.11)	(74)代理人	110001195
(86)国際出願番号	PCT/US2019/036834		弁理士法人深見特許事務所
(87)国際公開番号	WO2019/241428	(72)発明者	マラック, マイケル
(87)国際公開日	令和1年12月19日(2019.12.19)		アメリカ合衆国、80209コロラド州、デンバー、サウス・ワシントン・ストリート、573
審査請求日	令和4年5月24日(2022.5.24)	(72)発明者	リーバス, ルイス・イー
(31)優先権主張番号	62/684,498		アメリカ合衆国、80237コロラド州、デンバー、イースト・フロンティア
(32)優先日	平成30年6月13日(2018.6.13)		最終頁に続く
(33)優先権主張国・地域又は機関	米国(US)		
(31)優先権主張番号	62/749,001		
(32)優先日	平成30年10月22日(2018.10.22)		
	最終頁に続く		

(54)【発明の名称】 正規表現生成のためのユーザインターフェイス

(57)【特許請求の範囲】

【請求項1】

正規表現を生成する方法であって、
1つまたは複数のプロセッサを備える正規表現生成器が、1つまたは複数のデータソースからデータを取り出すことと、
前記正規表現生成器が、前記取り出されたデータを、ユーザインターフェイス内でテーブル形式で表示することとを備え、前記表示されるテーブル形式データは複数のデータセルを含み、前記方法はさらに、
前記正規表現生成器が、前記ユーザインターフェイスを介して、前記表示されたテーブル形式データの中からデータの第1の選択を受け取ることと、
前記正規表現生成器が、前記表示されたテーブル形式データの中からの前記データの第1の選択に基づいて、第1の正規表現を生成することと、
前記正規表現生成器が、前記ユーザインターフェイス内で前記第1の正規表現を表示することとを備える、正規表現を生成する方法。

【請求項2】

前記データの第1の選択を受け取るとは、前記ユーザインターフェイスを介して第1のデータセルの選択を受け取ることを含み、前記第1の正規表現を生成することは、前記第1の正規表現が前記第1のデータセル内のデータとマッチすることを確認することを含む、請求項1に記載の方法。

【請求項3】

前記データの第1の選択を受け取ることは、前記ユーザインターフェイスを介して第1のデータセル内のキャラクタのサブシーケンスの選択を受け取ることを含み、前記第1のデータセル内の1つまたは複数の追加のキャラクタは、前記データの第1の選択内において選択されない、請求項1に記載の方法。

【請求項4】

前記第1の正規表現を生成することは、前記第1の正規表現が前記第1のデータセル全体とマッチすることを確認することを含み、前記方法はさらに、

前記表示されたテーブル形式データ内において、前記第1の正規表現にマッチする、1つまたは複数の追加のデータセルを判断することと、

前記第1の正規表現にマッチする前記追加のデータセルの各々内で、前記第1のデータセル内の選択されたキャラクタのサブシーケンスに対応するキャラクタのサブシーケンスを判断することと、

前記ユーザインターフェイスを介して、前記追加のデータセル内の各判断されたキャラクタのサブシーケンスの選択を示すものを表示することとを備える、請求項3に記載の方法。

【請求項5】

前記方法はさらに、

前記ユーザインターフェイスを介して、前記表示されたテーブル形式データ内におけるバルク修正操作の選択を受け取ることを備え、前記方法はさらに、

前記選択されたバルク修正操作を、前記第1のデータセル内の前記キャラクタのサブシーケンス、および前記第1の正規表現にマッチする前記追加のデータセル内の前記対応するキャラクタのサブシーケンスに対して実行することを備え、

前記選択されたバルク修正操作は、前記第1のデータセル内の前記キャラクタのサブシーケンス、および前記第1の正規表現にマッチする前記追加のデータセル内の前記対応するキャラクタのサブシーケンスに対する抽出操作、置換操作、または削除操作の1つに対応する、請求項4に記載の方法。

【請求項6】

前記方法はさらに、

前記正規表現生成器が、前記ユーザインターフェイスを介して、前記表示されたテーブル形式データの中からデータの第2の選択を受け取ることと、

前記正規表現生成器が、前記データの第2の選択に応答して第2の正規表現を生成することとを備え、前記第2の正規表現は、前記表示されたテーブル形式データ内の前記データの第1の選択と前記データの第2の選択との両方にマッチし、前記方法はさらに、

前記正規表現生成器が、前記ユーザインターフェイス内に前記第2の正規表現を表示することとを備える、請求項1～5のいずれか1項に記載の方法。

【請求項7】

前記方法はさらに、

前記正規表現生成器が、前記ユーザインターフェイスを介して、前記表示されたテーブル形式データの中からデータの第2の選択を受け取ることと、

前記データの第1の選択および前記データの第2の選択に基づいて、前記データの第1の選択が、生成された正規表現によってマッチされるべきデータに対応し、前記データの第2の選択が、前記生成された正規表現によってマッチされるべきでないデータに対応する、と判断することと、

前記正規表現生成器が、前記データの第2の選択に応答して第2の正規表現を生成することとを備え、前記第2の正規表現は、前記表示されたテーブル形式データ内の前記データの第1の選択に対応のデータとマッチするが、前記データの第2の選択に対応のデータとマッチせず、前記方法はさらに、

前記正規表現生成器が、前記ユーザインターフェイス内に第2の正規表現を表示することとを備える、請求項1～5のいずれか1項に記載の方法。

【請求項8】

前記データの第1の選択を受け取ることは、前記ユーザインターフェイスを介して、第1のデータセルまたは前記第1のデータセル内のキャラクタのサブシーケンスの選択を受け取ることを含み、前記方法はさらに、

前記表示されたテーブル形式データ内において、前記第1の正規表現にマッチする、データセルの第1のサブセットを識別することと、

前記ユーザインターフェイス内で、前記第1の正規表現にマッチする前記データセルの第1のサブセットの各々に関連付けられる第1のグラフィカルインジケータを表示することとを備える、請求項1～7のいずれか1項に記載の方法。

【請求項9】

前記方法はさらに、

前記表示されたテーブル形式データ内において、前記第1の正規表現にマッチしない、データセルの第2のサブセットを識別することと、

前記ユーザインターフェイス内で、前記第1の正規表現にマッチしない前記データセルの第2のサブセットの各々に関連付けられる第2の異なるグラフィカルインジケータを表示することとを備える、請求項8に記載の方法。

【請求項10】

正規表現を生成するためのシステムであって、

1つまたは複数のプロセッサを含む処理ユニットと、

命令を記憶するメモリとを備え、前記命令は、前記処理ユニットによって実行されると、前記システムに、

1つまたは複数のデータソースからデータを取り出させ、

前記取り出されたデータを、ユーザインターフェイス内でテーブル形式で表示させ、前記表示されるテーブル形式データは複数のデータセルを含み、前記命令は、さらに、前記処理ユニットによって実行されると、前記システムに、

前記ユーザインターフェイスを介して、前記表示されたテーブル形式データの中からデータの第1の選択を受け取らせ、

前記表示されたテーブル形式データの中からの前記データの第1の選択に基づいて、第1の正規表現を生成させ、

前記ユーザインターフェイス内に前記第1の正規表現を表示させる、正規表現を生成するためのシステム。

【請求項11】

前記データの第1の選択を受け取ることは、前記ユーザインターフェイスを介して第1のデータセルの選択を受け取ることを含み、前記第1の正規表現を生成することは、前記第1の正規表現が前記第1のデータセル内のデータとマッチすることを確認することを含む、請求項10に記載のシステム。

【請求項12】

前記データの第1の選択を受け取ることは、前記ユーザインターフェイスを介して第1のデータセル内のキャラクタのサブシーケンスの選択を受け取ることを含み、前記第1のデータセル内の1つまたは複数の追加のキャラクタは、前記データの第1の選択内において選択されない、請求項11に記載のシステム。

【請求項13】

前記第1の正規表現を生成することは、前記第1の正規表現が前記第1のデータセル全体とマッチすることを確認することを含み、前記メモリは、さらなる命令を記憶し、前記さらなる命令は、前記処理ユニットによって実行されると、前記システムに、

前記表示されたテーブル形式データ内において、前記第1の正規表現にマッチする、1つまたは複数の追加のデータセルを判断させ、

前記第1の正規表現にマッチする前記追加のデータセルの各々内で、前記第1のデータセル内の選択されたキャラクタのサブシーケンスに対応するキャラクタのサブシーケンスを判断させ、

前記ユーザインターフェイスを介して、前記追加のデータセル内の各判断されたキャラ

10

20

30

40

50

クタのサブシーケンスの選択を示すものを表示させる、請求項 12 に記載のシステム。

【請求項 14】

前記メモリは、さらなる命令を記憶し、前記さらなる命令は、前記処理ユニットによって実行されると、前記システムに、

前記ユーザインターフェイスを介して、前記表示されたテーブル形式データ内におけるバルク修正操作の選択を受け取らせ、前記さらなる命令は、さらに、前記処理ユニットによって実行されると、前記システムに、

前記選択されたバルク修正操作を、前記第 1 のデータセル内の前記キャラクタのサブシーケンス、および前記第 1 の正規表現にマッチする前記追加のデータセル内の前記対応するキャラクタのサブシーケンスに対して実行させ、

前記選択されたバルク修正操作は、前記第 1 のデータセル内の前記キャラクタのサブシーケンス、および前記第 1 の正規表現にマッチする前記追加のデータセル内の前記対応するキャラクタのサブシーケンスに対する抽出操作、置換操作、または削除操作の 1 つに対応する、請求項 13 に記載のシステム。

【請求項 15】

前記メモリは、さらなる命令を記憶し、前記さらなる命令は、前記処理ユニットによって実行されると、前記システムに、

前記ユーザインターフェイスを介して、前記表示されたテーブル形式データの中からデータの第 2 の選択を受け取らせ、

前記データの第 2 の選択に回答して第 2 の正規表現を生成させ、前記第 2 の正規表現は、前記表示されたテーブル形式データ内の前記データの第 1 の選択と前記データの第 2 の選択との両方にマッチし、前記さらなる命令は、さらに、前記処理ユニットによって実行されると、前記システムに、

前記ユーザインターフェイス内に前記第 2 の正規表現を表示させる、請求項 10 ～ 14 のいずれか 1 項に記載のシステム。

【請求項 16】

前記データの第 1 の選択を受け取ることは、前記ユーザインターフェイスを介して、第 1 のデータセルまたは前記第 1 のデータセル内のキャラクタのサブシーケンスの選択を受け取ることを含み、前記メモリは、さらなる命令を記憶し、前記さらなる命令は、前記処理ユニットによって実行されると、前記システムに、

前記第 1 の正規表現にマッチする、前記表示されたテーブル形式データ内のデータセルの第 1 のサブセットを識別させ、

前記ユーザインターフェイス内で、前記第 1 の正規表現にマッチする前記データセルの第 1 のサブセットの各々に関連付けられる第 1 のグラフィカルインジケータを表示させる、請求項 10 ～ 15 のいずれか 1 項に記載のシステム。

【請求項 17】

前記メモリは、さらなる命令を記憶し、前記さらなる命令は、前記処理ユニットによって実行されると、前記システムに、

前記表示されたテーブル形式データ内において、前記第 1 の正規表現にマッチしない、データセルの第 2 のサブセットを識別させ、

前記ユーザインターフェイス内で、前記第 1 の正規表現にマッチしない前記データセルの第 2 のサブセットの各々に関連付けられる第 2 の異なるグラフィカルインジケータを表示させる、請求項 16 に記載のシステム。

【請求項 18】

前記メモリは、さらなる命令を記憶し、前記さらなる命令は、前記処理ユニットによって実行されると、前記システムに、

前記ユーザインターフェイスを介して、前記表示されたテーブル形式データの中からデータの第 2 の選択を受け取らせ、

前記データの第 1 の選択および前記データの第 2 の選択に基づいて、前記データの第 1 の選択が、生成された正規表現によってマッチされるべきデータに対応し、前記データの

10

20

30

40

50

第2の選択が、前記生成された正規表現によってマッチされるべきでないデータに対応する、と判断させ、

前記データの第2の選択に回答して第2の正規表現を生成させ、前記第2の正規表現は、前記表示されたテーブル形式データ内の前記データの第1の選択に対応のデータにマッチし、前記データの第2の選択に対応のデータにマッチせず、前記さらなる命令は、さらに、前記処理ユニットによって実行されると、前記システムに、

前記ユーザインターフェイス内に前記第2の正規表現を表示させる、請求項10～17のいずれか1項に記載のシステム。

【請求項19】

請求項1～9のいずれか1項に記載の方法をプロセッサに実行させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願の相互参照

本出願は、米国特許法第119条(e)に基づき、2018年6月13日に提出された「AUTOMATED GENERATION OF REGULAR EXPRESSIONS (自動化された正規表現生成)」と題される米国仮特許出願第62/684,498号に対する優先権を主張し、および米国特許法第119条(e)に基づき、2018年10月22日に提出された「AUTOMATED GENERATION OF REGULAR EXPRESSIONS (自動化された正規表現生成)」と題される米国仮特許出願第62/749,001号に対する優先権を主張する。米国仮特許出願第62/684,498号および第62/749,001号の全内容は、あらゆる目的のために参照により本明細書に組み込まれる。

【背景技術】

【0002】

背景

ビッグデータ解析システムは、予測解析、ユーザ挙動解析、および他の高度なデータ解析に使用することができる。しかしながら、有用な結果を提供するために任意のデータ解析が効果的に実行される前に、最初のデータセットは、クリーンかつキュレートされたデータセットにフォーマットされる必要があり得る。このデータオンボード化は、多くの場合、さまざまな異なるデータソースおよび／またはデータストリームからのデータが単一のデータリポジトリにコンパイルされ得るクラウドベースのデータリポジトリならびに他のビッグデータシステムに対する課題を提示する。そのようなデータは、複数の異なるフォーマットの構造化データ、異なるデータモデルに従った半構造化データ、およびさらには非構造化データを含み得る。そのようなデータのリポジトリは、多くの場合、さまざまな異なるフォーマットおよび構造内のデータ表現を含み、重複データおよび誤ったデータを含む場合もある。これらのデータリポジトリが報告、予測モデリング、および他の解析タスクのために解析されるとき、初期データセットの低信号対雑音比は、不正確であるかまたは有用でない結果につながり得る。

【0003】

データフォーマッティングおよび前処理の問題に対する多くの現在の解決策は、データ解析を実行する前にデータを共通のフォーマットに操作するために、データを浄化およびキュレートするための手動およびアドホック処理を含む。これらの手動処理は、特定のより小さいデータセットに対しては有効であり得るが、そのような処理は、大規模なデータセットを前処理およびフォーマットを試みる場合は、非効率的かつ非実用的であり得る。

【発明の概要】

【課題を解決するための手段】

【0004】

概要

本明細書で説明される態様は、正規表現を生成するためのさまざまな技法を提供する。

本明細書で使用される場合、「正規表現」は、より長い入力テキストストリング内のマッチを検索するために使用され得る、パターンを定義するキャラクタのシーケンスを指し得る。いくつかの実施形態では、正規表現は、記号のワイルドカードマッチング言語を使用して構成されてもよく、正規表現によって定義されたパターンは、キャラクタストリングとマッチするよう、および／または入力として与えられるキャラクタストリングから情報を抽出するよう、使用されてもよい。本明細書で説明するさまざまな実施形態では、データ処理システムとして実現される正規表現生成器を使用して、入力テキストデータを受信および表示し、クライアントユーザインターフェイスを介して入力テキストの特定のキャラクタサブセットの選択を受け取り、次いで、選択されたキャラクタサブセットに基づいて1つまたは複数の正規表現を生成することができる。1つまたは複数の正規表現を生成した後、正規表現エンジンを使用して、正規表現のパターンを1つまたは複数のデータセットに対してマッチさせることができる。さまざまな実施形態において、正規表現にマッチするデータは、抽出、再フォーマット、または修正などされてもよい。場合によっては、正規表現にマッチするデータに基づいて、追加の列、テーブル、または他のデータセットが作成されてもよい。

【0005】

本明細書で説明するいくつかの態様によれば、データ処理システムを介して実現される正規表現生成器は、1つまたは複数の正規表現コードの、異なるセットによって共有される、判断された最長共通サブシーケンス(LCS)に基づいて、正規表現を生成することができる。正規表現コード(カテゴリコードとも称され得る)は、たとえば、英語アルファベットの文字に対するL、数字に対するN、空白に対するZ、句読点に対するP、および他の記号に対するSを含み得る。1つまたは複数の正規表現コードの各セットは、ユーザインターフェイスを介して入力データとして受信された1つまたは複数のキャラクタの異なるシーケンスから変換され得る。LCSから除外される正規表現コードは、任意選択および／または代替として表され得る。いくつかの実施形態では、正規表現コードは、正規表現コードの最小発生数に関連付けられてもよい。追加または代替として、正規表現コードは、正規表現コードの最大発生数に関連付けられてもよい。たとえば、あるカテゴリコードのセットは、LCSの特定の部分が、ある文字を、もしあったとしてもせいぜい1回含む旨を示すよう、 $L \langle 0, 1 \rangle$ を含んでもよい。以下でより詳細に説明するように、入力データを中間正規表現コード(IREC)として一般化することは、非常にわずかな入力データを使用することを含むさまざまな技術的利点を提供し得、それは、まだ見られていないデータにおける偽陽性マッチまたは偽陰性マッチに屈しない正規表現のほぼ即時的な生成を可能にする。

【0006】

本明細書で説明される追加の態様によれば、正規表現は、3つ以上のキャラクタシーケンスを含む入力データに基づいて生成され得る。3つ以上のキャラクタシーケンスが入力データとして識別される場合、キャラクタシーケンスのLCSを識別する正規表現生成器は、ランタイムの指数関数的な増加をもたらし得る。すべてのキャラクタシーケンスのLCSを十分に機能する態様で識別するために、正規表現生成器は、2つのキャラクタシーケンスの各別個の組み合わせに対してLCSアルゴリズムを実行してもよい。LCSアルゴリズムの結果に基づいて、全結合グラフを生成してもよく、各グラフノードは異なるキャラクタシーケンスを表し、各グラフエッジの長さは、グラフエッジを定義するノードのLCSに対応する。次いで、全結合グラフに対して最小スパニングツリーの深さ優先のトラバースの走査を実行することによって、キャラクタシーケンスを選択する順序を判断してもよい。

【0007】

本明細書で説明されるさらなる態様は、陽性キャラクタシーケンス例および陰性キャラクタシーケンス例の両方を含む入力に基づいて正規表現を生成することに関する。陽性例は、生成されるべき正規表現にマッチするキャラクタのシーケンスを指し得、陰性例は、生成されるべき正規表現にマッチしないキャラクタのシーケンスを指し得る。いくつかの

10

20

30

40

50

実施形態では、陽性例および陰性例の両方が受け取られた場合、正規表現生成器は、弁別子、つまり陽性例を陰性例から区別する1つまたは複数のキャラクタの最短サブシーケンスを識別してもよい。選択された弁別子は、（たとえばカテゴリコードで表現された）最短シーケンスであってもよく、陽性または陰性のいずれかであってもよく、したがって、陽性例はマッチし、陰性例はマッチしないことになる。次いで、弁別子は、正規表現生成器によって生成される正規表現にハードコード化されてもよい。場合によっては、最短サブシーケンスは、陰性例のプレフィックス部分またはサフィックス部分に含まれてもよい。

【0008】

本明細書で説明されるさらなる態様は、正規表現を生成するために入力データが提供され得る1つまたは複数のユーザインターフェイスに関する。いくつかの実施形態では、ユーザインターフェイスは、正規表現生成サーバに通信可能に結合されたクライアントデバイスに表示されてもよい。ユーザインターフェイスは、サーバによって、クライアントデバイスによって、またはサーバおよびクライアントにおいて実行されるソフトウェアコンポーネントの組み合わせによってプログラマ的に生成され得る。ユーザインターフェイスを介して受信された入力データは、陽性例または陰性例を表し得る1つまたは複数のキャラクタシーケンスのユーザ選択に対応し得る。場合によっては、ユーザインターフェイスは、第2のキャラクタシーケンス内の第1のキャラクタシーケンスの選択を含む入力データをサポートしてもよい。たとえば、ユーザは、より大きな、以前に強調表示されたキャラクタシーケンス内の、1つまたは複数のキャラクタを強調表示することができ、第2のユーザ選択は、より大きな第1のユーザ選択のためのコンテキストを提供することができる。これは、入力データが、より高い特異性で正規表現生成器に提供されることを可能にし、正規表現生成器に「コンテキスト」を提供して、それが、偽陽性を回避する正規表現を生成できるようにすることを可能にする。ユーザインターフェイスを介してユーザがキャラクタシーケンスを選択することに応答して、正規表現生成器は正規表現を生成して表示してもよい。たとえば、ユーザがキャラクタの第1のシーケンスを強調表示すると、正規表現生成器は、キャラクタの第1のシーケンスにマッチする正規表現、ならびに他の同様のキャラクタシーケンス（たとえば、マッチするシーケンスについてユーザの意図と整合する）を生成し、表示することができる。ユーザがキャラクタの第2のシーケンスを強調表示すると、正規表現生成器は、キャラクタの第1のシーケンスとキャラクタの第2のシーケンスとの両方を包含する更新された正規表現を生成してもよい。次いで、ユーザが（たとえば第1のシーケンスまたは第2のシーケンスのいずれか内で）キャラクタの第3のシーケンスを強調表示すると、正規表現生成器は正規表現を再び更新してもよいなどとなる。

【0009】

本明細書で説明される追加の態様によれば、正規表現は、1つまたは複数の入力シーケンス例からの最長共通サブシーケンスに基づいて生成され得るが、例のいくつかのみに存在するキャラクタを取り扱うこともできる。いくつかの入力例においてのみ存在するキャラクタを取り扱うために、正規表現コードの最小発生数および最大発生数の両方が追跡されるスパンを定義してもよい。所与の入力例のすべてにスパンが存在しない可能性がある場合、最小発生数はゼロにセットされてもよい。次いで、これらの最小数および最大数は、正規表現マルチプリシティ構文にマッピングされ得る。最長共通サブシーケンス（LCS）アルゴリズムを、すべての入力例には現れない「任意選択の」スパン（たとえばゼロの最小長さ）を含む、入力例から導出されたキャラクタのスパン上で実行してもよい。以下で説明するように、連続するスパンは、LCSアルゴリズムの実行中にマージされてもよい。そのような場合において、一緒に担持されている追加の任意選択のスパンが連続して出現することに終わるとき、LCSアルゴリズムは、それらの任意選択のスパン上でも同様に再帰的に実行されてもよい。

【0010】

本明細書で説明されるさらなる態様は、正規表現生成器によって実行されるLCSアルゴリズムが複数回実行されて、「正しい」正規表現（たとえば、すべての所与の陽性例と

10

20

30

40

50

適切にマッチし、すべての所与の陰性例を適切に除外する正規表現)を生成してもよく、および／または最も望ましいもしくは最適な正規表現が選択され得る複数の正しい正規表現を生成してもよい、コンビナトリック探索に関する。いくつかの実施形態では、LCSアルゴリズムは一般に、正規表現を生成するために、入力例において右から左に実行され得る。しかしながら、比較の目的のために、および代替的な正規表現を見つけるために、LCSアルゴリズムは、入力例において逆方向に(たとえば左から右への方向で)別途実行されてもよい。たとえば、ユーザ入力として受信された例示的なキャラクターシーケンスは、それらがLCSアルゴリズムを通過する前に反転されてもよく、次いで、LCSアルゴリズムからの結果を、(元のテキストフラグメントを含んで)反転して戻してもよい。さらに、いくつかの実施形態では、LCSアルゴリズムは、正規表現生成器によって、複数回、通常のキャラクターシーケンス順序および逆の順序の両方で、行の始まりでの位置指定、行の終わりでの位置固定、行の始まりまたは終わりでの位置指定なしで、実行されてもよい。したがって、場合によっては、LCSアルゴリズムは、少なくともこれら6回実行されてもよく、最短の成功裡な正規表現が、これらの実行から選択されてもよい。

【図面の簡単な説明】

【0011】

【図1】さまざまな実施形態が実現され得る、正規表現を生成するための例示的な分散システムの構成要素を示すブロック図である。

【図2】本明細書で説明する1つまたは複数の実施形態による、ユーザインターフェイスを介して受信された入力に基づいて正規表現を生成するための処理を示すフローチャートである。

【図3】本明細書で説明する1つまたは複数の実施形態による、正規表現コードのセット上で最長共通サブシーケンス(LCS)アルゴリズムを使用して正規表現を生成するための処理を示すフローチャートである。

【図4】本明細書で説明する1つまたは複数の実施形態による、正規表現コードのセット上で最長共通サブシーケンス(LCS)アルゴリズムを使用して、2つのキャラクターシーケンス例に基づいて正規表現を生成するための例示的な図である。

【図5】本明細書で説明する1つまたは複数の実施形態による、より大きな正規表現コードのセット上で最長共通サブシーケンス(LCS)アルゴリズムを使用して正規表現を生成するための処理を示すフローチャートである。

【図6】本明細書で説明する1つまたは複数の実施形態による、正規表現コードのセット上で最長共通サブシーケンス(LCS)アルゴリズムを使用して、5つのキャラクターシーケンス例に基づいて正規表現を生成するための例示的な図である。

【図7】本明細書で説明する1つまたは複数の実施形態による、より大きな正規表現コードのセット上で最長共通サブシーケンス(LCS)アルゴリズムについて実行の順序を判断するための処理を示すフローチャートである。

【図8A】本明細書に記載する1つまたは複数の実施形態による、より大きな正規表現コードのセット上で最長共通サブシーケンス(LCS)アルゴリズムについて実行の順序を判断するために用いられる、全結合グラフを示す。

【図8B】本明細書に記載する1つまたは複数の実施形態による、より大きな正規表現コードのセット上で最長共通サブシーケンス(LCS)アルゴリズムについて実行の順序を判断するために用いられる、全結合グラフの最小スパニングツリー表現を示す。

【図9】本明細書に記載する1つまたは複数の実施形態による、陽性キャラクターシーケンス例および陰性キャラクターシーケンス例に基づいて正規表現を生成するための処理を示すフローチャートである。

【図10A】本明細書に記載する1つまたは複数の実施形態による、陽性キャラクターシーケンス例に基づく正規表現の生成を示す例示的なユーザインターフェイス画面である。

【図10B】本明細書に記載する1つまたは複数の実施形態による、陽性キャラクターシーケンス例および陰性キャラクターシーケンス例に基づく正規表現の生成を示す例示的なユーザインターフェイス画面である。

【図 1 1】本明細書で説明する 1 つまたは複数の実施形態による、ユーザインターフェイス内で受信されるユーザデータ選択に基づいて正規表現を生成するための処理を示すフローチャートである。

【図 1 2】本明細書で説明する 1 つまたは複数の実施形態による、ユーザインターフェイス内で受信されるユーザデータ選択を介して、正規表現を生成し、キャプチャグループに基づいてデータを抽出する処理を示すフローチャートである。

【図 1 3】本明細書に記載する 1 つまたは複数の実施形態によるテーブル形式データディスプレイを示す例示的なユーザインターフェイス画面である。

【図 1 4】本明細書に記載する 1 つまたは複数の実施形態による、テーブル形式ディスプレイからのデータの選択に基づく正規表現およびキャプチャグループの生成を示す例示的なユーザインターフェイス画面である。

10

【図 1 5】本明細書に記載する 1 つまたは複数の実施形態による、テーブル形式ディスプレイからのデータの選択に基づく正規表現およびキャプチャグループの生成を示す例示的なユーザインターフェイス画面である。

【図 1 6 A】本明細書に記載する 1 つまたは複数の実施形態による、テーブル形式ディスプレイからの陽性例および陰性例の選択に基づく正規表現の生成を示す例示的なユーザインターフェイス画面である。

【図 1 6 B】本明細書に記載する 1 つまたは複数の実施形態による、テーブル形式ディスプレイからの陽性例および陰性例の選択に基づく正規表現の生成を示す例示的なユーザインターフェイス画面である。

20

【図 1 7】本明細書に記載する 1 つまたは複数の実施形態による、テーブル形式ディスプレイからのデータの選択に基づく正規表現およびキャプチャグループの生成を示す別の例示的なユーザインターフェイス画面である。

【図 1 8】本明細書で説明する 1 つまたは複数の実施形態による、最長共通サブシーケンス (LCS) アルゴリズムを使用して、任意選択のスパンを含む、正規表現を生成するための処理を示すフローチャートである。

【図 1 9】本明細書で説明する 1 つまたは複数の実施形態による、最長共通サブシーケンス (LCS) アルゴリズムを使用して、任意選択のスパンを含む、正規表現を生成するための例示的な図である。

【図 2 0】本明細書で説明する 1 つまたは複数の実施形態による、最長共通サブシーケンス (LCS) アルゴリズムのコンビナトリックな実行に基づいて正規表現を生成するための処理を示すフローチャートである。

30

【図 2 1】本発明のさまざまな実施形態が実現され得る例示的な分散システムの構成要素を示すブロック図である。

【図 2 2】本発明の実施形態によって提供されるサービスがクラウドサービスとして提供され得るシステム環境の構成要素を示すブロック図である。

【図 2 3】本発明の実施形態が実現され得る例示的なコンピュータシステムを示すブロック図である。

【発明を実施するための形態】

【0012】

40

詳細な説明

以下の記載では、説明を目的として、本発明のさまざまな実施形態の完全な理解のために、多数の具体的な詳細が記載される。しかしながら、本発明の実施形態は、これらの具体的な詳細のいくつかを伴わずに実施され得ることが当業者には明白であろう。他の例では、周知の構造およびデバイスがブロック図の形で示される。

【0013】

以下の説明は、例示的な実施形態のみを提供するものであり、本開示の範囲、適用可能性、または構成を限定することを意図したものではない。むしろ、例示的な実施形態の以下の説明は、例示的な実施形態を実施するための実施可能な説明を当業者に提供するであろう。特許請求の範囲に記載されている本発明の精神および範囲から逸脱することなく、

50

要素の機能および構成にさまざまな変更を加えることができることを理解されたい。

【0014】

具体的な詳細が、以下の説明において、実施の形態の十分な理解を与えるよう与えられる。しかしながら、当業者には、実施の形態はこれらの具体的な詳細なしに実施されてもよいことが理解される。たとえば、回路、システム、ネットワーク、プロセスおよび他のコンポーネントは、実施の形態を不必要な詳細で曖昧にしないように、ブロック図形式におけるコンポーネントとして示され得る。他の例では、周知の回路、プロセス、アルゴリズム、構造および技術は、実施の形態を曖昧にすることを回避するために、不必要な詳細なしに示され得る。

【0015】

さらに、個々の実施の形態は、フローチャート、フロー図、データフロー図、構造図またはブロック図として示されるプロセスとして記載されることが注記される。フローチャートは動作をシーケンシャルなプロセスとして記載するかもしれないが、動作の多くは並列または同時に実行され得る。加えて、動作の順序は再構成されてもよい。プロセスは、その動作が完了されるときに終結されるが、図に含まれない追加のステップを含み得る。プロセスは、方法、関数、プロシージャ、サブルーチン、サブプログラムなどに対応し得る。プロセスが関数に対応する場合は、その終結は、その関数が呼出関数または主関数に戻ることに対応し得る。

【0016】

「コンピュータ読取可能媒体」という語は、命令および／もしくはデータを記憶するか、含むか、または担持することができるポータブルまたは固定された記憶装置、光記憶装置、ならびにさまざまな他の媒体のような非一時的媒体を含むが、それらに限定はされない。コードセグメントまたはコンピュータ実行可能な命令は、プロシージャ、関数、サブプログラム、プログラム、ルーチン、サブルーチン、モジュール、ソフトウェアパッケージ、クラス、または、命令、データ構造もしくはプログラム文の任意の組合せを表し得る。コードセグメントは、情報、データ、引数、パラメータまたはメモリコンテンツを受け渡すおよび／または受け取ることによって、別のコードセグメントまたはハードウェア回路に結合されてもよい。情報、引数、パラメータ、データなどは、メモリ共有、メッセージ受渡し、トークン受渡し、ネットワーク伝送などを含む任意の好適な手段を介して渡されるか、転送されるか、または伝送されてもよい。

【0017】

さらに、実施の形態は、ハードウェア、ソフトウェア、ファームウェア、ミドルウェア、マイクロコード、ハードウェア記述言語、またはそれらの任意の組み合わせによって実現されてもよい。ソフトウェア、ファームウェア、ミドルウェアまたはマイクロコードにおいて実現される場合には、必要なタスクを実行するプログラムコードまたはコードセグメントを機械読取可能媒体に記憶してもよい。プロセッサは必要なタスクを実行してもよい。

【0018】

本明細書では、1つまたは複数の入力データ例内で識別されるパターンに対応する正規表現を生成するためのさまざまな技法（たとえば、方法、システム、1つまたは複数のプロセッサによって実行可能な複数の命令を記憶する非一時的なコンピュータ読取可能記憶メモリなど）について説明する。特定の実施形態では、入力データの選択を受け取ることに応答して、入力データ内の1つまたは複数のパターンが自動的に識別され、識別されたパターンを表すよう、正規表現（または短く「レギュクス」）が自動的にかつ効率的に生成され得る。そのようなパターンは、キャラクタのシーケンス（たとえば、文字、数字、空白、句読点、記号等のシーケンス）に基づくことができる。本明細書では、方法、システム、1つもしくは複数のプロセッサによって実行可能なプログラム、コード、または命令を記憶する非一時的コンピュータ読取可能記憶媒体などを含むさまざまな実施形態について説明する。

【0019】

いくつかの実施形態では、正規表現は、キャラクタストリングにマッチするために、および／または入力として与えられるキャラクタストリングから情報を抽出するために、記号的なワイルドカードマッチング言語を使用して構成され得る。たとえば、第1の例示的な正規表現[A-Za-z]{3} /d?/d, /d/d/d/dは、ある日付（たとえばApril 3, 2018）にマッチしてもよく、第2の例示的な正規表現[A-Za-z]{3} /d?/d, (/d/d/d/d)を使用して、マッチする日付から年を抽出してもよい。正規表現生成システムによって受信される入力データは、たとえば、1つもしくは複数の「陽性」データ例および／または1つもしくは複数の「陰性」データ例を含み得る。本明細書で使用される場合、陽性例は、入力として受信され、その入力に基づいて生成される正規表現によってマッチされることになるキャラクタシーケンスを指し得る。一方、陰性例は、その入力に基づいて生成される正規表現によってマッチされない、入力されたキャラクタシーケンスを指し得る。

【0020】

いくつかの技術的利点が、本明細書に記載されるさまざまな実施形態および例内で実現され得る。たとえば、本開示で説明するいくつかの技法は、正規表現生成処理の速度および効率を向上させ得る（たとえば、レグックス解は、1秒未満で生成され得、ユーザインターフェイスは、対話型リアルタイム使用に適し得る）。本明細書で説明されるさまざまな技法はまた、判断論的であってもよく、トレーニングデータを必要としなくてもよく、いかなる初期正規表現入力も必要とせず解を生成してもよく、完全に自動化されてもよい（たとえば、任意の人的介入を必要とする範囲内で正規表現を生成する）。さらに、本明細書で説明されるさまざまな技法は、効果的に処理され得るデータ入力のタイプに関して限定される必要はなく、そのような技法は、結果として得られる正規表現の人間による可読性を改善し得る。

【0021】

本明細書で説明するいくつかの実施形態は、最長共通サブシーケンス（LCS）アルゴリズムの1つまたは複数の実行を含む。LCSアルゴリズムは、ある状況においては、2つのテキストファイル間の差分を判断し示すように構成された差分エンジン（たとえば、Unix “diff” utilityの背後にあるエンジン）として使用され得る。いくつかの実施形態では、入力データ（たとえば、ストリングおよび他のキャラクタシーケンス）は、抽象的なトークンに変換され得、次いで、それらは、LCSアルゴリズムへの入力として提供され得る。そのような抽象的なトークンは、たとえば、正規表現キャラクタクラスを表す正規表現コード（たとえば、Loogleコードまたは他のキャラクタクラスコード）に基づくトークンであってもよい。そのようなコードのさまざまな異なる例が考えられ、本明細書では「正規表現コード」または「中間正規表現コード」（IREC）と称され得る。たとえば、入力されたキャラクタシーケンス「May 3」は、IRECコード「LLLZN」に変換されてもよく、その後、トークン化されたストリングは、他のトークン化されたストリングとともにLCSアルゴリズムに供されてもよい。いくつかの実施形態では、入力されたキャラクタシーケンスが共通に有さないIREC（たとえば正規表現コード）が、最終的に生成された正規表現において、任意選択（たとえば任意選択のスパン）として現れてもよい。特定の实施形態では、正規表現コードは、<https://www.regular-expressions.info/unicode.html#category>で示されるユニコードカテゴリコードに基づくカテゴリコードであってもよい。たとえば、コードLは文字を表してもよく、コードNは数字を表してもよく、コードZは空白を表してもよく、コードSは記号を表してもよく、コードPは句読点を表してもよい等である。たとえば、コードLは、ユニコード/p{L}に対応し、コードNは、ユニコード/p{N}に対応し得る。これは、LCS出力から正規表現への1対1のマッピングが働くことを可能にし（たとえば、/pN/pN/pZ/pL/pLは「10 am」にマッチすることができる）、これは、人間による可読性の利点を提供し得る。加えて、これらの異なるカテゴリは、互いに素、つまり相互に排他的であってもよい。すなわち、この例では、カテゴリL、N、Z、PおよびSは、カテゴリのメンバ間に重複がないように、互いに素にされてもよい。

【0022】

さらなる技術的利点は、正規表現コード（たとえばカテゴリコード）、スパンなどの使用に基づく正規表現のより効率的な生成を含むさまざまな実施形態において実現され得る。そのようなコードを使用することによって、LCSアルゴリズムが入力ストリング内のキャラクタのすべてまたは実質的にすべてを異なるものとして首尾よく識別する場合には、計算資源を無駄にする必要がない。本明細書においてさまざまな実施形態によって提供されるさらなる技術的利点は、生成された正規表現の改善された可読性、ならびに陽性例および陰性例の両方を入力データとしてサポートすること、およびさまざまな有利なユーザインターフェイス特徴を提供すること（たとえば、ユーザが、抽出のために、より大きなキャラクタシーケンスまたはデータセル内のテキストフラグメントを強調表示することを可能にする）を含む。

10

【0023】

I. 全体の概要

本明細書に開示されるさまざまな実施形態は、正規表現の生成に関連する。いくつかの実施形態では、正規表現生成器として構成されるデータ処理システムは、正規表現コード（たとえばカテゴリコード）の異なるセットによって共有される最長共通サブシーケンス（LCS）を識別することによって正規表現を生成することができる。正規表現コードの各セットは、ユーザインターフェイスを介して入力データとして受け取られるキャラクタのシーケンスから変換され得る。本明細書で説明する技術的利点のうち、入力データを中間コード（たとえば、正規表現コード、スパンなど）として抽象化することにより、非常に少ない入力データを使用して正規表現を効率的に生成することができる。

20

【0024】

図1は、さまざまな実施形態が実現され得る、正規表現を生成するための例示的な分散システムの構成要素を示すブロック図である。この例に示されるように、クライアントデバイス120は、正規表現生成サーバ110（または正規表現生成器）と通信し、ユーザインターフェイスと対話してテーブル形式データを取り出し表示し、ユーザインターフェイスを介した入力データ（たとえば例）の選択に基づいて正規表現を生成することができる。いくつかの実施形態では、クライアントデバイス120は、クライアントウェブブラウザ121および／またはクライアント側正規表現アプリケーション122（たとえば、サーバ110によって生成された正規表現を受信／消費するクライアント側アプリケーション）を介して正規表現生成器110と通信してもよい。正規表現生成器110内で、クライアントデバイス120からの要求は、ネットワークインターフェイスにおいてさまざまな通信ネットワークを介して受信され、REST API 112などのアプリケーションプログラミングインターフェイス（API）によって処理され得る。ユーザインターフェイスデータモデル生成器114コンポーネントは正規表現生成器110とともに、サーバ側プログラミングコンポーネントおよびロジックを提供して、本明細書で説明されるさまざまなユーザインターフェイス特徴を生成し、レンダリングすることができる。そのような特徴は、ユーザがデータリポジトリ130からテーブル形式データを取り出し、表示すること、入力データ例を選択して正規表現の生成を開始し、生成された正規表現に基づいてデータを修正および／または抽出することを可能にする機能を含み得る。この例では、正規表現生成器コンポーネント116は、入力キャラクタシーケンスを正規表現コードおよび／またはスパンに変換すること、入力データに対してアルゴリズム（たとえばLCSアルゴリズム）を実行すること、および正規表現を生成／単純化することを含んで、正規表現を生成するように実現され得る。正規表現生成器116によって生成された正規表現は、RESTサービス112によってクライアントデバイス120に送信されてもよく、そこで、クライアントブラウザ121（または対応するクライアント側アプリケーションコンポーネント122）上のJavascriptコードは、次いで、ブラウザ内でレンダリングされたスプレッドシート列内のすべてのセルに対して正規表現を適用することができる。他の場合では、サーバ側でマッチするデータ／マッチしないデータを識別するために、サーバ側で別個の正規表現エンジンコンポーネントを実現して、生成された正規表現を、ユーザインターフェイス上に表示されるテーブル形式データおよび／またはデータリポジト

30

40

50

リ 1 3 0 に格納された他のデータ内で比較してもよい。さまざまな実施形態において、マッチする／マッチしないデータは、ユーザインターフェイス内で自動的に選択（たとえば強調表示）されてもよく、抽出、修正、削除などのために選択されてもよい。正規表現の生成に基づいて、ユーザインターフェイスを介して抽出または修正される任意のデータは、1 つまたは複数のデータリポジトリ 1 3 0 に記憶され得る。さらに、いくつかの実施形態では、生成された正規表現（および／または L C S アルゴリズムへの対応する入力）は、将来の検索および使用のために正規表現ライブラリ 1 3 5 に記憶されてもよい。いくつかの実施形態では、生成された正規表現は、実際に「ライブラリ」に格納される必要はなく、「変換スクリプト」に組み込まれてもよい。たとえば、E T . S . 特許番号 1 0 , 2 1 0 , 2 4 6（すべての目的のために、ここに引用により援用する）においてより詳細に記載されるように、そのような変換スクリプトは、受信されたデータを変換するために 1 つまたは複数の処理ユニットによって実行可能であり得るプログラム、コード、または命令を含み得る。変換スクリプトの他の考えられ得る例は、「改名列」、「大文字列データ」、または「ファーストネームから性別を推測し性別を伴う新たな列を作成する」などを含み得る。

【 0 0 2 5 】

図 2 は、本明細書に記載される 1 つまたは複数の実施形態に係る、ユーザインターフェイスを介して受信される入力に基づいて正規表現を生成するための処理を示すフローチャートである。ステップ 2 0 1 において、正規表現生成器 1 1 0 は、正規表現生成器ユーザインターフェイスにアクセスし、ユーザインターフェイスを介して特定のデータを閲覧するための要求をクライアントデバイス 1 2 0 から受信することができる。ステップ 2 0 1 における要求は、R E S T A P I 1 1 2、および／またはウェブサーバ、認証サーバなどを介して受信されてもよく、ユーザの要求はパズおよび認証されてもよい。たとえば、ビジネスまたは組織内のユーザは、取引データ、顧客データ、実績データ、予測データ、および／または組織のデータリポジトリ 1 3 0 に記憶され得るデータの任意の他のカテゴリを解析および／または修正するために、正規表現生成器 1 1 0 にアクセスすることができる。ステップ 2 0 2 において、正規表現生成器 1 1 0 は、選択された入力データに基づく正規表現の生成をサポートするユーザインターフェイスを介して、要求されたデータを取り出し、表示することができる。そのようなユーザインターフェイスのさまざまな実施形態および例は、以下で詳細に説明される。

【 0 0 2 6 】

ステップ 2 0 3 において、ユーザは、正規表現生成器 1 1 0 によって提供されるユーザインターフェイスに表示されたデータから 1 つまたは複数の入力キャラクターケンスを選択し得る。いくつかの実施形態では、データは、特定のデータタイプおよび／またはデータのカテゴリを有するラベル付き列を含んで、ユーザインターフェイス内にテーブル形式で表示されてもよい。そのような場合、ステップ 2 0 3 における入力データの選択は、ユーザがデータセルを選択すること、またはデータセル内の個々のテキストフラグメントを選択すること（たとえば強調表示すること）に対応し得る。しかしながら、他の実施形態では、正規表現生成器 1 1 0 は、ユーザインターフェイスを介して半構造化および非構造化データの検索ならびに表示をサポートしてもよく、ユーザは、半構造化または非構造化データからキャラクターケンスを選択することによって正規表現生成のための入力データを選択してもよい。後述する例で説明されるように、ユーザが表示されたテーブル形式データから入力キャラクターケンスを選択することは、単なる使用ケースの一例である。他の例では、ユーザ（たとえば、おそらく、Linux（登録商標）コマンドラインツール grep、sed、または awk などのために正規表現を構築することを試みるソフトウェア開発者またはパワーユーザ）は、スプレッドシートから例を拾うのではなく、例において初めからタイプしてもよい。

【 0 0 2 7 】

ステップ 2 0 4 において、正規表現生成器 1 1 0 は、ステップ 2 0 3 においてユーザにより選択された入力データに基づいて、1 つまたは複数の正規表現を生成してもよい。ス

10

20

30

40

50

テップ 205 において、正規表現生成器 110 は、たとえば、生成された正規表現を表示するように、および／または表示されたデータ内のマッチする／マッチしないデータを強調表示するように、ユーザインターフェイスを更新してもよい。いくつかの実施形態では任意選択であり得るステップ 206 において、ユーザインターフェイスは、ユーザが生成された正規表現に基づいて基礎となるデータを修正することを可能にする機能をサポートしてもよい。たとえば、ユーザインターフェイスは、ユーザが、テーブル形式データから、特定のデータフィールドを、それらフィールドが正規表現とマッチするか否かに基づいて、フィルタリング、修正、削除、または抽出することを可能にする特徴をサポートしてもよい。データをフィルタリングまたは修正することは、リポジトリ 130 に格納された基礎となるデータを修正することを含むことができ、場合によっては、抽出したデータを新たな列および／または新たなテーブルとしてリポジトリ 130 に格納することができる。

10

【0028】

これらのステップは、正規表現生成器 110 のユーザインターフェイスとの例示的なユーザ対話の一般的かつハイレベルの概要を示すが、他の実施形態では、さまざまな追加の特徴および機能性をサポートしてもよい。たとえば、いくつかの実施形態では、正規表現コード（またはカテゴリコード）は、コードの最小発生数に関連付けられ得る。追加的または代替的に、正規表現コードは、コードの最大発生数に関連付けられてもよい。一例として、正規表現コードのセットは、LCS の特定の部分がある文字を少なくとも 0 回、および最大でも 1 回のいずれかで含むことを示すコード $L \langle 0,1 \rangle$ を含むことができる。

【0029】

さらに、いくつかの実施形態では、入力データは、3 つ以上のキャラクターシーケンスを含み得る。そのような実施形態では、さまざまな技法を用いて、3 つ以上のキャラクターシーケンスに対して LCS アルゴリズムを実行するための順序を判断し、結果として生じる正規表現が十分に機能する態様で生成され得るようにして、3 つ以上の入力キャラクターシーケンスによって引き起こされるランタイムの指数関数的な増加を回避することができる。なお、正規表現生成器 110 は、かわりに、一度に 2 つのキャラクターシーケンス上で LCS アルゴリズムを実行し、グラフに基づいてキャラクターシーケンスのペアを選択するための順序を判断してもよい。たとえば、全結合グラフは、LCS アルゴリズムの第 1 の実行（たとえば LCS 1）は、シーケンス 1 およびシーケンス 3 に対して実行されるべきであることを示し、次いで、LCS アルゴリズムの第 2 の実行（たとえば LCS 2）は、LCS 1 およびシーケンス 2 に対して実行されるべきであることなどを示し得る。グラフは、全結合グラフであってもよく、ノードは、キャラクターシーケンスを表し、エッジはノードを接続して、接続されたノードによって共有される LCS の長さを表す。グラフ内の各ノードは、グラフ内の他のすべてのノードに接続されてもよく、キャラクターシーケンスを選択する順序は、グラフについて最小スパニングツリーの深さ優先トラバースを実行することによって判断されてもよい。

20

30

【0030】

さらなる実施形態では、入力データは、いくつかの異なる方法でユーザインターフェイスを介して提供されてもよい。たとえば、入力データは、キャラクタのセットの第 2 のユーザ選択内の、1 つまたは複数のキャラクタの第 1 のユーザ選択を示してもよい。たとえば、ユーザは、以前に強調表示されたキャラクタのセット内のうちのあるキャラクタを強調表示してもよい。したがって、第 2 のユーザ選択は、第 1 のユーザ選択のためのコンテキストを提供し得、これは、入力データが、より高い特異性で正規表現生成器 110 に提供されることを可能にし得る。いくつかの実施形態では、正規表現生成器 110 は、各ユーザ選択に応答して、ほぼリアルタイムで正規表現を生成し、表示することができる。たとえば、ユーザがキャラクタの第 1 の範囲を強調表示する場合、正規表現生成器 110 は、キャラクタの第 1 の範囲を表す正規表現を表示してもよい。そして、ユーザが、キャラクタの第 1 の範囲内の、キャラクタの第 2 の範囲を強調表示すると、正規表現生成器 110 は、表示される正規表現を更新してもよい。

40

【0031】

50

さらに、いくつかの実施形態では、正規表現生成器 110 は、陽性および陰性の両方の例を含む入力に基づいて正規表現を生成することができる。上述したように、陽性例は、正規表現に包含されるべきキャラクタのシーケンスを指し得、陰性例は、正規表現に包含されるべきでないキャラクタのシーケンスを指し得る。そのような場合、正規表現生成器 110 は、特定の位置において、陽性例を陰性例から区別する、1 つまたは複数のキャラクタからなる最短サブシーケンスを識別することができる。次いで、最短サブシーケンスは、正規表現生成器 110 によって生成された正規表現内でハードコード化され得る。さまざまな例では、最短サブシーケンスは、プレフィックス／サフィックス部分、または陰性例内のミッドスパンに含まれ得る。

【0032】

特定の実施形態による、正規表現を自動的に生成するさらなる例を以下に記載する。これらの例は、図 2 の一般的技法のさまざまな具体的な考えられ得る実現例に対応し得、それぞれのシステムの 1 つまたは複数の処理ユニット（たとえば、プロセッサ、コア）によって実行されるソフトウェア（たとえば、コード、命令、プログラムなど）、ハードウェア、またはそれらの組み合わせにおいて実現され得る。ソフトウェアは、非一時的記憶媒体上に（たとえばメモリデバイス上に）記憶され得る。以下で説明するさらなる例は、例示的かつ非限定的であることを意図している。これらの例は、特定の順番または順序で生じるさまざまな処理ステップを示すが、これは限定することを意図するものではない。いくつかの代替実施形態では、ステップは、ある異なる順序で実行されてもよく、またはいくつかのステップは、並行して実行されてもよい。

【0033】

いくつかの例では、ユーザインターフェイス（たとえばステップ 203）を介して受信されるユーザ入力は、正規表現出力によってマッチされる 1 つまたは複数の「陽性例」と、正規表現出力によってマッチされない 0 またはそれより多い「陰性例」とを含み得る。任意選択で、陽性例のうちの 1 つまたは複数の強調表示して、キャラクタの特定の範囲（またはサブシーケンス）を選択することができる。場合によっては、ステップ 204 において、ユーザインターフェイスを介して受信された陽性例は、正規表現コード（たとえば、ユニコードカテゴリコードのようなキャラクタカテゴリコード）のスパンに変換され得る。各陽性例に対して、スパンのシーケンスが生成され得る。いくつかの実施形態では、各頂点が、スパンのシーケンスのうちの 1 つに対応し、エッジ重みが、エッジのエンドポイントに対応するスパンのそれらの 2 つのシーケンス上で実行される LCS アルゴリズムからの出力の長さに等しい、グラフが生成されてもよい。グラフについて、最小スパニングツリーを判断することができる。たとえば、いくつかの実施形態では、Prim のアルゴリズムを用いて、最小スパニングツリーを得てもよい。深さ優先トラバースを最小スパニングツリー上で実行してトラバース順序を判断してもよく、その後、LCS アルゴリズムを、トラバースの最初の 2 つの要素上で実行してもよい。次いで、1 つずつ、トラバースの各追加要素が、前の LCS 反復の出力および次の現在のトラバース要素に対して LCS アルゴリズムを再び実行することによって、現在の LCS 出力に順番にマージされてもよい。次いで、スパンのシーケンスであり得る、LCS アルゴリズムの最終出力が、正規表現に変換されてもよい。この変換は、いくつかの実施形態では、1 対 1 変換であり得るが、本明細書で説明される特定の任意選択的な実施形態は、1 対 1 変換に対応しないかもしれない。最後に、得られた正規表現は、ステップ 203 でユーザインターフェイスを介して受信されたすべての陽性例および陰性例に対してテストされてもよい。テストのいずれかが失敗した場合、すべての陽性例および失敗した任意の陰性例を使用して、上述の処理を繰り返してもよい。

【0034】

II. 正規表現コード上で最長共通サブシーケンスアルゴリズムを用いた正規表現生成
上述したように、本明細書で説明するいくつかの態様は、入力データに対応する正規表現コードの異なるセットによって共有される最長共通サブシーケンス（LCS）の計算に基づく正規表現の生成に関する。

【0035】

図3は、本明細書で説明する1つまたは複数の実施形態による、正規表現コードのセットに対してLCSアルゴリズムを使用して正規表現を生成するための処理を示すフローチャートである。ステップ301において、正規表現生成器110は、入力データとして1つまたは複数のキャラクタシーケンスを受け取り得る。上述したように、いくつかの例では、入力データは、ユーザインターフェイスに表示されたテーブル形式データ内から選択された陽性例データに対応し得るが、いくつかの実施形態ではユーザインターフェイスは任意選択であり、入力データは、さまざまな例では任意の他の通信チャネル（たとえば非ユーザインターフェイス）を介して受信される任意のキャラクタシーケンスに対応し得ることを理解されたい。

10

【0036】

ステップ302において、ステップ301において受け取られた各キャラクタシーケンスは、対応する正規表現コードに変換され得る。さまざまな実施形態において、正規表現コードは、Loogleコード、ユニコードカテゴリコード、または正規表現キャラクタクラスを表す任意の他のキャラクタクラスコードであり得る。たとえば、ある入力キャラクタシーケンス「May 3」をLoogleコード「LLLZN」に変換してもよい。いくつかの実施形態では、正規表現コードは、<https://www.regular-expressions.info/unicode.html#category>に示されるユニコードカテゴリコードに基づくカテゴリコードであり得る。たとえば、コードLは文字を表してもよく、コードNは数字を表してもよく、コードZは空白を表してもよく、コードSは記号を表してもよく、コードPは句読点を表してもよい、等である。たとえば、コードLは、ユニコード/p{L}に対応し、コードNは、ユニコード/p{N}に対応し得る。

20

【0037】

ステップ303では、ステップ302で生成された正規表現コードのセットの中から最長共通サブシーケンスを判断することができる。いくつかの実施形態では、LCSアルゴリズムは、入力として正規表現コードの2つのセットを使用して実行され得る。LCSアルゴリズム（たとえば、処理の方向、位置指定、空白のプッシュ、低濃度スパンの合体（coalescing）、共通トークン上の整列など）の実行のさまざまな異なる特性が、異なる実施形態において使用され得る。ステップ304では、LCSアルゴリズムの出力に基づいて正規表現を生成することができる。場合によっては、ステップ304は、LCSアルゴリズムの出力を正規表現コードで捕捉すること、および正規表現コードを正規表現に変換することを含み得る。ステップ305において、正規表現は、たとえば、ユーザインターフェイスを介してユーザに正規表現を表示することによって、単純化され、出力されてもよい。

30

【0038】

図4は、2つのキャラクタシーケンス例に基づき、正規表現コードのセットに対して最長共通サブシーケンス（LCS）アルゴリズムを使用して、正規表現を生成するための例示的な図である。したがって、図4は、図3において上述した処理を適用する例を示している。図4に示すように、本例の正規表現は、2つの入力ストリング「iPhone 5」および「iPhone X」に基づいて生成される。この例における各シーケンスは、正規表現コードのそれぞれのセットに変換され得る。したがって、iPhone 5は「LLLLLLZN」に変換されてもよく、iPhone Xは「LLLLLLZL」に変換されてもよい。図4に示されるように、これらのカテゴリコードは、次いで、LCSアルゴリズムへの入力として提供され、LCSアルゴリズムは、IREC（またはカテゴリコード）の両方のセットは6つのLおよび1つのZを含む、と判断する。LCSから除外されるZカテゴリコードは、任意選択および／または代替として表され得る。したがって、両方のキャラクタシーケンスを包含する正規表現は、/pL{6}/pZ/pN?/pL?のように表され得る。この例において、正規表現はユニコードカテゴリコード（たとえば、文字については/pL、空白については/pZ、数字については/pN）を含む。数字6を含む中括弧は、文字の6つのインスタンスを示し、疑問符は、最後の数字／文字が任意選択であることを示す。最後に、正規表現生成器によって単純化処

40

50

理を実行することができ、その間に、正規表現は、共通テキストフラグメント「iPhone」を最終正規表現に挿入し戻すことによって単純化され、正規表現のより広い「/pL{6}/」部分を置き換える。

【0039】

この例に示されるように、正規表現生成器110によって受け取られた入力ストリングは、正規表現ブロードカテゴリ（これは、「カテゴリコード」とも呼ばれ得る）を表す「正規表現コード」に変換されてもよく、LCSアルゴリズムは、それらの正規表現コード上で実行されてもよい。いくつかの実施形態では、正規表現コードのために、ユニコードカテゴリコードを使用し得る。たとえば、入力テキストストリングは、レゲックスユニコードブロードカテゴリ（たとえば、文字については/pL、句読点については/pP等である。）を表すコードに変換され得る。図3および図4によって示されるこのアプローチは、間接的アプローチと称され得る。しかしながら、他の実施形態では、LCSアルゴリズムが入力として受け取られたキャラクタシーケンスに対して直接実行される直接的アプローチが使用されてもよい。

【0040】

いくつかの実施形態では、間接的アプローチは、大量のトレーニングデータを必要としないという点で、追加の技術的利点を提供し得、比較的より少ない数の入力例で有効な正規表現を生成し得る。これは、間接的なアプローチが、正規表現生成における不確実性を低減するために、および潜在的な偽陽性および偽陰性を除去するために、ヒューリスティックを使用するためである。たとえば、入力ストリング「May 3」および「Apr 11」に基づいて正規表現を生成する際に、直接的アプローチは、日付パターンにマッチする有効な正規表現を生成するために月毎に少なくとも1つの例を必要とし得る。それら2つの例のみに依拠して、直接的アプローチは、「[AM][ap][yr][13]1?」のレゲックスを生成してもよい。対照的に、間接的アプローチは、ユニコードブロードカテゴリに基づいて、「/pL{3}/d{1,2}」の、より効果的な正規表現を生成してもよい。加えて、上述のように、本明細書で説明される技術的利点の1つは、場合によっては単一の例からさえ、非常にわずかな入力データを使用して正規表現を効率的に生成することを含む。たとえば、単一の例「am」からの正規表現の生成に関して、あるヒューリスティックは、正規表現のために「am」を生成するか「/pL/pL」を生成するかを判断することができる。どちらも、おそらくは正しいが、プログラムされたヒューリスティックは、最適な正規表現を生成する方法（たとえば、それが「pm」にもマッチすべきか否か）を判断するために、ユーザ選好および／または基準を実現し得る。

【0041】

加えて、間接的アプローチは、さらに、生成された正規表現「/pL{3}/d{1,2}」～「[A-Za-z]{3}/d{1,2}」を単純化して、それをより人間によって可読なものにすることができる。これは、いくつかの実施形態において、たとえば、正規表現のためのユニコード表現に精通していない可能性がある高度な知識のない正規表現ユーザに出力する場合に有益であり得る。

【0042】

さらに、いくつかの実施形態では、LCSアルゴリズムを実行する際に各キャラクタを独立して扱う代わりに、シーケンシャルかつ等しい正規表現コードが、スパンデータ構造（スパンとも称され得る）に変換され得る。場合によっては、スパンは、単一の正規表現コード（たとえばユニコードブロードカテゴリコード）の表現を、繰り返しカウント範囲（たとえば最小数および／または最大数）とともに含み得る。正規表現コードからスパンへの変換は、代替（たとえば分離）を認識するなど、以下に説明されるいくつかのさまざまな追加の特徴を容易にすることができ、また、生成された正規表現をさらに単純化するために、隣接する任意選択のスパンのマージを容易にすることもできる。

【0043】

上述したように、LCSアルゴリズムは、図4のストリング「iPhone」のような、最終的な正規表現に潜在的に挿入して戻すことができる、入力キャラクタシーケンス内の基礎

10

20

30

40

50

となるテキストフラグメントを記憶し、保持するように構成することができる。そのスパンに割り当てられたカテゴリコードを元々もたらしたテキストフラグメントを追跡することによって、そのような実施形態は、リテラルテキスト（たとえばamおよびpm）が、生成された正規表現に直接含まれることを可能にし、偽陽性を低減し、正規表現出力をより人間により可読なものにすることができる。

【0044】

III. 正規表現コードの組み合わせ上で最長共通サブシーケンスアルゴリズムを用いた正規表現生成

本明細書で説明するさらなる態様は、3つ以上のストリング（たとえば3つ以上の別個のキャラクタシーケンス）を含む入力データに基づく正規表現の生成に関する。3つ以上のストリングが入力データとして識別される場合、正規表現生成器110は、LCSアルゴリズム実行のシーケンスに対して最適な順序が判断される性能最適化特徴を使用し得る。以下で説明するように、3つ以上のストリングに関する性能最適化機能は、各ストリングに対応する頂点と、各ストリングと他のすべてのストリングとの間のLCS出力のサイズに基づき得るエッジ長さ／重みとでグラフを構築することを含み得る。次いで、それらのエッジ重みを使用して最小スパニングツリーが導出され得、入力ストリングの順序を判断するために、深さ優先トラバースが実行され得る。最後に、判断された入力ストリングの順序を使用して、一連のLCSアルゴリズムが行われてもよい。

【0045】

図5は、正規表現コードの、より大きなセット（たとえば3つ以上のキャラクタシーケンス）に対して、最長共通サブシーケンス（LCS）アルゴリズムを使用して、正規表現を生成するための処理を示すフローチャートである。したがって、この例におけるステップ502～505は、図3で上述したステップ303に対応し得る。しかしながら、この例は、3つ以上の入力キャラクタシーケンスに基づいて正規表現を生成することに関するため、LCSアルゴリズムは複数回実行されてもよい。たとえば、3つ以上の入力ストリングのランタイムの指数関数的な増加を避けるために、LCSアルゴリズムは複数回実行されてもよく、各実行は2つの入力ストリングだけに対して実行される。たとえば、正規表現生成器110は、2つのストリング（たとえば2つの入力キャラクタシーケンスまたは2つの変換された正規表現コード）に対してLCSアルゴリズムの最初の実行を実行してもよく、次いで、第1のLCSアルゴリズムの出力および第3のストリングに対してLCSアルゴリズムの第2の実行を実行してもよく、次いで、第2のLCSアルゴリズムの出力および第4のストリングに対してLCSアルゴリズムの第3の実行を実行してもよい等となる。

【0046】

そのような実施形態の性能を改善および／または最適化するために、LCSアルゴリズムのシーケンスを実行する入力ストリング（たとえば入力キャラクタシーケンスまたは正規表現コード）の最適な順序を判断することが望ましい場合がある。たとえば、入力ストリングを取り込むための良好な順序は、任意選択のスパンの数を最小にすることなどによって、生成された正規表現の可読性に影響を及ぼし得る。生成されたレゲックスを簡潔に保つために、現在のレゲックスにLCSされる追加のストリングは、現在のレゲックス（既に見られたストリングをLCSしたことからの中間結果）に既にいくらか類似していることが好ましい。

【0047】

これにより、ステップ501では、複数（たとえば3つ以上）の入力キャラクタシーケンスが正規表現コードに変換される。ステップ502では、LCSアルゴリズムを使用して正規表現コードを処理する順序を判断する。ステップ502における順序の判断については、図7を参照して後述する。ステップ503において、判断された順序における最初の2つの正規表現コードが（ステップ503の最初の反復のために）選択されるか、または判断された順序における次の正規表現コードが（ステップ503の後続の反復のために）選択される。ステップ504において、LCSアルゴリズムは、正規表現コードのフォ

ーマットに対応する2つの入力ストリングに対して実行される。ステップ504の第1の反復では、LCSアルゴリズムは、判断された順序における最初の2つの正規表現コードに対して実行され、ステップ504の後続の反復では、LCSアルゴリズムは、判断された順序における次の正規表現コードおよび前のLCSアルゴリズムの出力（同じ正規表現コードのフォーマットであってもよい）に対して実行される。ステップ505において、正規表現生成器110は、LCSアルゴリズムへの入力としてまだ提供されていない、判断された順序における追加の正規表現コードがあるか否かを判断する。判断された順序におけるそのような追加の正規表現コードがある場合、処理は、LCSアルゴリズムの別の実行のためにステップ503に戻る。そうでない場合には、ステップ506において、LCSアルゴリズムの最後の実行の出力に基づいて、正規表現が生成される。

10

【0048】

図6は、5つの入力キャラクターシーケンス例に基づいて正規表現を生成するための例示的な図である。この例では、各入力されたキャラクターシーケンスを正規表現コードに変換し、次いで、判断された正規表現コードの順序に基づいてLCSアルゴリズムを繰り返し実行する。したがって、図6は、図5において上述した処理を適用する一例を示す。この例では、5つの正規表現コードについて判断された順序は、コード#1～コード#5であり、各コードは、判断された順序でLCSアルゴリズムに入力されて、正規表現出力が生成される。最終の正規表現出力（Reg Ex #4）は、入力キャラクターシーケンスの5つすべてに基づいて生成された最終正規表現に対応する。

【0049】

20

図7は、正規表現コードの、より大きいセット（たとえば3つ以上）上における最長共通サブシーケンス（LCS）アルゴリズムについての実行順序を判断するための処理を示すフローチャートである。したがって、この例に示されるように、ステップ701～704は、上述のステップ502における順序判断に対応し得る。ステップ701において、LCSアルゴリズムは、入力データに対応する正規表現コードの各他とは異なるペア上で実行されてもよく、得られた出力LCSは、実行ごとに記憶されてもよい。したがって、 k 個の入力データについて、これは、LCSアルゴリズムを通して実行されるストリングのすべての $(k(k-1))/2$ 個の可能なペア形成、またはいくつかの実施形態では $k(k-1)$ を表し得る。たとえば、 $k=3$ の入力キャラクターシーケンスが受け取られた場合、LCSアルゴリズムは、ステップ701において3回実行されてもよく； $k=4$ の入力キャラクターシーケンスが受け取られた場合、LCSアルゴリズムは、ステップ701において6回実行されてもよく； $k=5$ の入力キャラクターシーケンスが受け取られた場合、LCSアルゴリズムは、ステップ701において10回実行されてもよい等となる。ステップ702において、全結合グラフは、 $(k(k-1))/2$ 個のエッジのエッジ重みが2つのノード間の未処理LCS出力の長さでストリングを表す k 個のノードから構築され得る。ステップ703では、ステップ702における全結合グラフから最小スパニングツリーを導出し得る。ステップ704では、深さ優先トラバースが最小スパニングツリー上で実行され得る。このトラバースの出力は、正規表現コードがLCSアルゴリズム実行のシーケンスに入力される順序に対応し得る。

30

【0050】

40

図8Aおよび図8Bを簡単に参照すると、図5に、受け取られた $k=5$ の入力キャラクターシーケンスに基づいて生成された全結合グラフの例が示され、図8Bに、全結合グラフに対する最小スパニングツリー表現が示されている。

【0051】

いくつかの実施形態では、図5～図8Bに記載されるアプローチは、性能に関して追加の技術的利点を提供し得る。たとえば、LCSアルゴリズムの、特定の従来の実現例は、 $O(n^2)$ のランタイム性能を示すことができ、ここで、 n はストリングの長さである。そのような実現例を2だけの代わりに k 個のストリングに拡張すると、指数関数的なランタイム性能 $O(n^k)$ をもたらす得、なぜならば、LCSアルゴリズムは k 次元空間を探索するよう求められ得るからである。LCSアルゴリズムのそのような従来の実現例は、

50

リアルタイムのオンラインユーザ体験に対しては、十分に機能しないか、または十分に適していないかもしれない。

【0052】

上述したように、LCSアルゴリズムは、 $(k(k-1))/2$ 回実行されてもよく、その場合、時として重複は以前に見られたのと全く同じであり、なぜならば、LCSアルゴリズムは、ユーザからの生の入力例がレグックスカテゴリコードに変換された場合であり得るためである。したがって、いくつかの場合においては、記憶化が実現され得、キャッシュを使用して、以前に見られたLCS問題を以前に機能したLCS解にマッピングすることができる。

【0053】

IV. 陽性および陰性パターンマッチング例に基づく正規表現の生成

本明細書で説明するさらなる態様は、陽性および陰性の両方の例に対応する入力データに基づいて正規表現を生成することに関する。上述したように、陽性例は、正規表現生成器によって生成される正規表現にマッチするべき例のストリングとして指定される入力データキャラクターシーケンスを指し得る。一方、陰性例は、正規表現生成器によって生成される正規表現とマッチするべきでない例のストリングとして指定される入力データキャラクターシーケンスを指し得る。以下で説明するように、いくつかの実施形態では、正規表現生成器110は、位置、およびその位置で陽性例を陰性例から区別するキャラクタの最短サブシーケンスを識別するように構成され得る。次いで、最短サブシーケンスは、生成された正規表現にハードコード化され得、陽性例は正規表現とマッチすることになり、陰性例は正規表現によって除外される（たとえばマッチしない）ことになる。

【0054】

図9は、陽性のキャラクターシーケンス例および陰性のキャラクターシーケンス例に基づいて正規表現を生成するための処理を示すフローチャートである。ステップ901において、正規表現生成器110は、陽性例に対応する1つまたは複数の入力データキャラクターシーケンスを受け取ることができる。ステップ902において、正規表現生成器110は、受け取った陽性例に基づいて正規表現を生成し得る。したがって、ステップ901～902は、入力データキャラクターシーケンスに基づいて正規表現を生成するために、上で論じた図3または図5において実行されるステップの一部またはすべてを含み得る。

【0055】

ステップ903において、正規表現生成器110は、陰性例に対応する1つの追加入力データキャラクターシーケンスを受け取ることができる。したがって、ステップ902で生成された正規表現とマッチしないように、陰性例を具体的に指定する。いくつかの実施形態では、ステップ903において受け取られた陰性例は、ステップ902において生成された正規表現に対して最初にテストされてもよく、陰性例が正規表現とマッチしないと判断された場合、さらなるアクションはとられない。しかしながら、この例では、ステップ903において受け取られた陰性例のうちの少なくとも1つはステップ902において生成された正規表現とマッチする、と仮定されてもよい。したがって、ステップ904では、ステップ902で生成された正規表現内で曖昧性除去位置を判断してもよい。いくつかの実施形態では、曖昧性除去位置は、（たとえば正規表現の始まりにおける）プレフィックス位置または（たとえば正規表現の終わりにおける）サフィックス位置のいずれかとして選択され得る。たとえば、正規表現生成器110は、陽性例を陰性例から区別するためにプレフィックスで必要とされるであろう第1の数のキャラクタと、陽性例を陰性例から区別するためにサフィックスで必要とされるであろう第2の数のキャラクタとを判断してもよい。次いで、正規表現生成器110は、必要とされる置換キャラクタの最短数に基づいてサフィックスまたはプレフィックスを選択し得る。場合によっては、曖昧性除去位置としてプレフィックスを使用することが、可読性のために好まれる（たとえば重み付けされる）ことがある。さらに他の例では、曖昧性除去位置は、正規表現のプレフィックスまたはサフィックスに対応しないミッドスパン位置であり得る。

【0056】

ステップ905において、正規表現生成器110は、判断された位置で正規表現に挿入されると、陽性例を陰性例から区別することができるカスタムキャラクタクラスの置換シーケンスを判断することができる。いくつかの実施形態では、ステップ905において、正規表現生成器110は、陽性例および陰性例の各々から、曖昧性除去位置（または置換位置）に対応するテキストフラグメントを取り出し、次いで、それらテキストフラグメントを使用して、陽性例を陰性例から区別する置換シーケンスとして使用される弁別子を判断することができる。さらに、ステップ905で判断される弁別子置換シーケンスは、カスタムキャラクタクラスの、複数の異なる置換シーケンスを含むことができ、それらは、正規表現内の同じ位置または異なる位置のいずれかで置換されることができる。

【0057】

上述したように、場合によっては、ステップ905における置換シーケンスの判断は、ステップ904における曖昧性除去位置（または置換位置）の判断との関連で実行されてもよい。たとえば、正規表現生成器110は、第1の可能な置換位置において陽性例を陰性例から区別し得る1つまたは複数の置換シーケンスを判断し得る。正規表現生成器110はまた、第2の異なる可能な置換位置において陽性例を陰性例から区別し得る1つまたは複数の他の置換シーケンスも判断し得る。この例において、異なる可能な置換位置および対応する置換シーケンスの間で選択するとき、正規表現生成器110は、ヒューリスティック式を適用して、置換位置のキャラクタのサイズ、ならびに対応する置換シーケンスの数および／またはサイズ、のうちの1つまたは複数の基づいて選択を実行することができる。最後に、ステップ906において、正規表現は、1つまたは複数の判断された置換シーケンスを判断された位置に挿入して正規表現の以前の部分を置換することによって、修正され得る。場合によっては、ステップ906における正規表現の修正に続いて、陽性例および／または陰性例を、修正された正規表現に対してテストして、陽性例が正規表現とマッチし、陰性例が正規表現とマッチしないことを確認してもよい。

【0058】

図10Aおよび図10Bは、陽性および陰性のキャラクタシーケンスの例に基づく正規表現の生成を示す例示的なユーザインターフェイス画面である。したがって、図10Aおよび図10Bに示される例は、上述の図9の処理の実行中に表示されるユーザインターフェイスに対応し得る。図10Aにおいて、ユーザは、データ入力キャラクタシーケンスの3つの陽性例1001を提供し、正規表現生成器110は、陽性例の各々にマッチする正規表現1002を生成する。次に、図4Bにおいて、ユーザは、1つの陰性例1004を提供し、正規表現生成器110は、陽性例の現在のセット1003および陰性例の現在のセット1004の両方に基づく修正された正規表現1005を生成する。

【0059】

上で注記したように、いくつかの実施形態では、陽性例および陰性例の両方が受け取られると、正規表現生成器110は、弁別子、つまり陽性例を陰性例から区別する1つまたは複数のキャラクタの最短サブシーケンスを識別してもよい。選択された弁別子は、最短シーケンス（たとえばカテゴリコードで表現される）であってもよく、陽性または陰性のいずれかであってもよく、陽性例はマッチし、陰性例はマッチしないことになる。場合によっては、弁別子は、次いでステップ905で正規表現にハードコード化され得る置換サブシーケンスに対応してもよい。一例として、「[AL][a-z]+」において、[AL]は、それが街路サフィックスに適用されると仮定すると、「Alley（小路）」、「Avenue（大通り）」、および「Lane（横町）」とはマッチする（またはそれらを認める）であろうが、他のすべてのものとはマッチしない（または許容しない）であろう陽性の弁別子である。別の例として、「[BC][o][a-z]+」において、[BC][o]は、「Boulevard（並木道）」および「Court（袋小路）」とマッチするであろう2つのキャラクタクラスのシーケンスからなる陽性の弁別子である。さらに別の例として、「[^A][a-z]+」において、[LA]は、「Alley」および「Avenue」を許容しないであろう陰性の弁別子であってもよい。場合によっては、アルゴリズムは、正しく弁別するために陰性後読みを生成してもよい。たとえば、(?<!Av)[A-Za-z]+は、「Avenue」は除外するが、「Alley」は認めるであろう。

10

20

30

40

50

【0060】

別の例として、ユーザが陽性例「202-456-7800」および「313-678-8900」ならびに陰性例「404-765-9876」および「515-987-6570」を供給する場合、ある実施形態では、正規表現生成器110は、正規表現「/d/d/d-/d/d/d-/d/d00」を生成してもよい。すなわち、置換キャラクタサブシーケンスは、正規表現のサフィックスについて、（たとえば、目的が、ビジネス電話番号とマッチする正規表現であると仮定して、）00で終わる電話番号は陽性例を陰性例から区別するという判断に基づいて、識別され得る。これは、サフィックスによる陰性例の例（より具体的に言えば、陽性のサフィックスを使用することによって陰性例に対処する例）であるが、さまざまな他の実施形態は、プレフィックス、サフィックス、またはミッドスパン位置のいずれかでの置換をサポートしてもよい。ミッドスパン位置での置換の例では、スパン内にオフセットされるキャラクタが追跡され、ミッドスパンポイントで分割され得る。

10

【0061】

プレフィックスまたはサフィックスを使用するかどうかを判断するために、いくつかの実施形態では、 k_a およびプレフィックス／サフィックスのすべての組み合わせにわたって最小スコアが選択されるヒューリスティックが使用される：

【0062】

【数1】

$$\text{スコア} = k_a \min^2 \left\{ \frac{|F_p|}{1 + |E_p|}, \frac{|F_n|}{1 + |E_n|} \right\} + \begin{cases} 0.0 & (\text{プレフィックスの場合}) \\ 0.1 & (\text{サフィックスの場合}) \end{cases}$$

20

【0063】

k_a ＝アフィックス（プレフィックスまたはサフィックス）の曖昧性を除去すると考えられるキャラクタの数

$|F_p|$ ＝アフィックスの曖昧性を除去するために必要とされる陽性例からの一意のテキストフラグメントの数

$|F_n|$ ＝アフィックスの曖昧性を除去するために必要とされる陰性例からの一意のテキストフラグメントの数

$|E_p|$ ＝ユーザによって提供される（完全な）陽性例の数

$|E_n|$ ＝ユーザによって提供される（完全な）陰性例の数

30

上記の例では、ヒューリスティックは、より長い曖昧性除去テキストフラグメント（したがって、たとえば、 k_a による乗算）よりもより短い曖昧性除去テキストフラグメントを好むように設計される。ヒューリスティックはまた、可読性を改善するよう、サフィックス（したがって、たとえば、サフィックスに対する0.1のペナルティ）よりもプレフィックスを好むように設計される。最後に、ヒューリスティックは、より長いプレフィックスまたはサフィックスの曖昧性除去（たとえば置換）を、より多数のストリングフラグメント（したがって、たとえば、置換されるべきストリングフラグメントの数を二乗）を使用することによる曖昧性除去よりも好むように設計される。

【0064】

40

上述したように、いくつかの実施形態はまた、陰性のミッドスパン例、ならびに陰性の後読み例および陰性の先読み例をサポートしてもよい。

【0065】

いったんプレフィックス／サフィックスおよび k （曖昧性除去すべきキャラクタの数）が判断されると、正規表現生成器110は、さらに、その曖昧性除去を、生成された正規表現においてどのように表現するかを判断することができる。生成された正規表現は、陽性例のように見えるアフィックス（たとえばプレフィックスまたはサフィックス）について許容的（permissive）であってもよく、または陰性例のように見えるアフィックスを除外してもよい。

【0066】

50

【数 2】

$$usePermissive = \frac{|E_p|}{|F_p|} - \frac{|E_n|}{|F_n|}$$

【0067】

usePermissiveがゼロよりも大きい場合、陽性例のように見えるものは、（各キャラクタ位置）について1つずつ、陽性例から取られたキャラクタを許容する正規表現を生成することによって通される。他の場合では、正規表現生成器110は（各キャラクタ位置について）1つずつ、陰性例から取られたキャラクタを許容しない正規表現を生成することにより、陰性例のように見えるものを許容しないアプローチをとってもよい。

10

【0068】

別の例として、陽性例8amおよび陰性例9pmについて生成された正規表現は、/d[[^]p]mであるかもしれない。これは、カレット構文を使用する。場合によっては、正規表現生成器110は、より短い正規表現を好むように構成されてもよく、それは、ユーザにとってより可読であり得るだけでなく、より正しい可能性があり得る。その原理は、今後、頻繁に出現するキャラクタは、今後再び出現する可能性がよりありそうであり、したがって、頻繁に出現するキャラクタに強調がおかれるべきである、というものである。一意のキャラクタ |_p| がより少ない場合（出現するものはより頻繁に出現するため、一意性はより少ない）、これは、分母内にそれを有することによって、ヒューリスティックにおいて報酬を与えられる。

20

【0069】

再び上記のusePermissive例のヒューリスティックを参照すると、1つの一意の陽性アフィックスを判断することは、ユーザから1つの陽性例しかない場合、大きな特色ではない。したがって、このヒューリスティックでは低い |_p| は、分子内にそれを有することによってペナルティを科せられる（すなわち、高い |_p| は、このヒューリスティックにおいて報酬を与えられる）。

【0070】

さらに、いくつかの実施形態では、陰性例は、後読みおよび／または先読みに基づいてもよい。たとえば、ユーザは、「323-1234」の陽性例および「202-754-9876」の陰性例を提供し、その場合、それはエリアコードを有する電話番号を除外するためにレゲックス後読み構文(? <!)の使用を伴う。

30

【0071】

場合によっては、陰性例は、任意選択のスパンに基づいてもよい。たとえば、ユーザは、「ab」および「a2b」の陽性例および「a3b」の陰性例を提供してもよい。この場合、ある例示的な実現例は失敗し得、なぜならば、それは、必要とされるスパンのみに基づいて弁別しようとし得、「2」の桁は任意選択のスパンにあるからである。この例において、失敗とは、生成された正規表現が、陽性例のすべてに（正しく）マッチし、また、陰性例の1つまたは複数に（誤って）マッチする状況を指し得る。そのような場合、ユーザは、失敗に対して警告されることができ、生成された正規表現を手動で修復するために、および／または陰性例の一部を除去するために、ユーザインターフェイスを介して選択肢を提供されることができる。

40

【0072】

V. 正規表現生成のためのユーザインターフェイス

本明細書で説明する追加の態様は、正規表現の生成に関連するグラフィカルユーザインターフェイス内にいくつかの異なる特徴および機能を含む。以下で説明するように、これらの特徴のうちのあるものは、ユーザ選択のためのさまざまな選択肢、および陽性例および陰性例の強調表示、陽性例および陰性例のカラーコード化、ならびにデータセル内の複数の重複／ネストされた強調表示を含み得る。

【0073】

50

図 1 1 は、ユーザインターフェイス内で受信されたユーザデータ選択に基づいて正規表現を生成するための処理を示すフローチャートである。図 1 1 の例示的な処理は、入力データキャラクタシーケンスに基づいて正規表現を生成する前述の例のいずれかに対応し得る。しかしながら、図 1 1 は、クライアントデバイス 1 2 0 上で生成および表示され得るユーザインターフェイスに関する処理を説明する。ステップ 1 1 0 1 において、ユーザインターフェイスを介したユーザからの要求に回答して、正規表現生成器 1 1 0 は、（たとえばデータリポジトリ 1 3 0 から）データを取り出し、そのデータをグラフィカルユーザインターフェイス内においてテーブル形式でレンダリング／表示してもよい。なお、この例ではテーブル形式データを用いているが、他の例ではテーブル形式データを使用および表示しなくてもよいことを理解されたい。たとえば、場合によっては、ユーザは、（ユーザインターフェイスからデータを選択するのではなく）生データを直接タイプすることができる。さらに、データがユーザインターフェイスを介して提示されるとき、データはテーブル形式である必要はなく、構造化されていないデータ（たとえばドキュメント）または半構造化（たとえば、ツイートまたはポストのような非フォーマット化／非構造化データアイテムのスプレッドシート）であってもよい。さまざまな例では、テーブル形式データは、取引データ、顧客データ、実績データ、予測データ、および／またはビジネスもしくは他の組織のためにデータリポジトリ 1 3 0 に記憶され得るデータの任意の他のカテゴリに対応し得る。ステップ 1 1 0 2 において、入力データのユーザ選択がユーザインターフェイスを介して受信され得る。選択された入力データは、たとえば、ユーザによって選択されたあるデータセル全体、またはあるデータセル内のキャラクタのサブシーケンスに対応し得る。ステップ 1 1 0 3 において、正規表現生成器 1 1 0 は、ステップ 1 1 0 2 において受信された入力データ（たとえばデータセルまたはその一部分）に基づいて正規表現を生成し得る。ステップ 1 1 0 4 において、ユーザインターフェイスは、正規表現の生成に回答して更新され得る。場合によっては、ユーザインターフェイスは、単に、生成された正規表現をユーザーに表示するために更新されてもよく、一方、他の場合では、ユーザインターフェイスは、以下で説明するさまざまな他の方法で更新されてもよい。この例に示されるように、ユーザは、ユーザインターフェイスを介して複数の異なる入力データキャラクタシーケンスを選択してもよく、受信された各新たな入力データに回答して、正規表現生成器 1 1 0 は、キャラクタシーケンスの第 1 および第 2 の（陽性の）例の両方を包含する更新された正規表現を生成してもよい。次いで、ユーザがキャラクタの第 3 のシーケンスを（たとえば、両方のキャラクタシーケンス外、または第 1 もしくは第 2 のキャラクタシーケンス内で）を強調表示すると、正規表現生成器 1 1 0 は、正規表現を再び更新してもよい等となる。いくつかの実施形態では、正規表現生成器 1 1 0 は、アルゴリズムをリアルタイム（またはほぼリアルタイム）で実行し得、全面的に新たな正規表現が、ユーザによってなされた各新たなキーストロークまたは各新たな強調表示されたセクションに回答して生成され得る。

【 0 0 7 4 】

したがって、図 1 1 に示すように、ユーザインターフェイスを介するキャラクタシーケンスのユーザ選択に回答して、正規表現生成器 1 1 0 は、正規表現を生成および表示し得る。たとえば、ユーザがキャラクタの第 1 のシーケンスを強調表示すると、正規表現生成器は、キャラクタの第 1 のシーケンスを表す正規表現を生成し、表示し得る。ユーザがキャラクタの第 2 のシーケンスを強調表示すると、正規表現生成器は、キャラクタの第 1 のシーケンスとキャラクタの第 2 のシーケンスとの両方を包含する更新された正規表現を生成し得る。次いで、ユーザがキャラクタの第 3 のシーケンスを（たとえば第 1 のシーケンスまたは第 2 のシーケンスのいずれか内で）強調表示すると、正規表現生成器は正規表現を再び更新し得る等となる。

【 0 0 7 5 】

図 1 2 は、ユーザインターフェイス内で受信されたユーザデータ選択を介して、正規表現を生成し、キャプチャグループに基づいてデータを抽出するための処理を示す別のフローチャートである。ステップ 1 2 0 1 において、ステップ 1 1 0 1 で上述したように、正

正規表現生成器 110 は、（たとえばデータリポジトリ 130 から）データを取り出し、そのデータをグラフィカルユーザインターフェイス内においてテーブル形式でレンダリング／表示することができる。ステップ 1202 において、正規表現生成器 110 は、特定のデータセル内のテキストフラグメントのユーザ強調表示の選択を受け取ることができる。ステップ 1203 において、正規表現生成器 110 は、選択されたデータセルの陽性例に基づいて正規表現を生成することができ、ステップ 1204 において、セル内で強調表示されたテキストフラグメントに基づいて正規表現キャプチャグループを作成することができる。ステップ 1205 において、正規表現生成器 110 は、生成された正規表現とマッチする、表示されたテーブル形式データ内の 1 つまたは複数の追加セルを判断することができ、ステップ 1206 において、生成された正規表現とマッチする追加セル内の対応するテキストフラグメントを抽出することができる。

10

【0076】

したがって、陽性例を供給することに加えて、ユーザは、（たとえばマウステキスト強調表示を介して、）選択された陽性例のいずれか内でテキストフラグメントを選択してもよい。これに応答して、正規表現生成器 110 は、例から、そのテキストフラグメントを抽出し、正規表現が適用されているテキスト中の他のすべてのマッチから、対応するフラグメントを抽出するために、正規表現キャプチャグループを作成してもよい。マッチするデータセルからテキストフラグメントを抽出することは、削除および修正も含み得、場合によっては、半構造化または非構造化テキストの既存の列からデータの新たな列を作成するために用いられ得る。

20

【0077】

ユーザが陽性のデータ例を選択する例を用いて、ユーザが年を強調表示した場合、正規表現生成器 110 は正規表現 $(?:[A-Z]\{3\}/s+/d/d,/s+|/d/d//d/d)(/d/d/d/d)$ を生成することができる。この例に示すように、正規表現生成器 110 は、年の周りに括弧を付けてあり、また、月および日の周りの古い括弧（代替のために使用）を?:レグックス構文の使用により「非キャプチャ」グループに変換している。いくつかの実施形態では、抽出／キャプチャグループは、スパン境界上にあることが要求され得、そのような実施形態では、正規表現生成器 110 は、強調表示されたキャラクタ範囲を入力として取り得、最も近い位置指定子スパン境界を包含するようにそれを拡張し得る。しかしながら、他の例では、ミッドスパン抽出／キャプチャは、ユーザインターフェイスによってサポートされてもよい。

30

【0078】

いくつかの実施形態では、ユーザインターフェイスは、第 2 のキャラクタシーケンス内の第 1 のキャラクタシーケンスの選択を含むユーザからの入力データをサポートしてもよい。たとえば、ユーザは、より大きな以前に強調表示されたキャラクタシーケンス内の 1 つまたは複数のキャラクタを強調表示することができ、第 2 のユーザ選択は、より大きな第 1 のユーザ選択のためのコンテキストを提供してもよい。そのような実施形態は、入力データが、より高い特異性で正規表現生成器 110 に提供されることを可能にし得る。

【0079】

さらに、いくつかの例では、ユーザがユーザインターフェイス内で選択する（たとえばテキストを強調表示する）ことに応答して、動作を開始し、ダイアログを開くことができる。場合によっては、ダイアログは、メイン画面とのユーザ対話を妨げないフローティングツールボックスウィンドウなどの非モデルダイアログであってもよい。ダイアログはまた、ユーザがどのような主要な操作を行っているかに応じて外観および／または機能性を変化させることもできる。したがって、そのような場合、ユーザは、キャプチャグループテキストフラグメントの修正、抽出などを開始するために、選択されたテキストを強調表示した後にさらなるメニューアイテムを検索する必要がない。さらに、特定の実施形態では、正規表現を生成するために提供されるユーザインターフェイスは、3 つの強調表示モード、すなわちネスト化自動、ネスト化手動、および単一レベルを含み得る。場合によっては、デフォルト動作モードは、セル全体が強調表示された領域として識別されることで

40

50

あってもよく、ユーザは、強調されたセル内の1つまたは複数の追加のサブシーケンスをさらに強調表示してもよい。他のモードでは、ユーザは、テーブル形式データディスプレイのデータセル内で両方の強調表示を手動で指定することを許可されてもよい。さらに他のモードでは、ユーザは、内側強調表示なしで外側強調表示を手動で指定することを許可されてもよい。これらのその他のモードは、半構造化データ、たとえば、ブラウザ「ユーザエージェント」ストリングなどのツイートまたは他の長いストリングからなるデータの列に、より適し得る。「半構造化」データは、ユーザインターフェイス内でテーブル形式で表示され得るデータを指すが、テーブル内の列は非構造化テキストからなる。

【0080】

いくつかのそのような実施形態では、ユーザインターフェイスを介したユーザによる内側および外側の選択（たとえば強調表示）は、カラーコード化によって区別され得る。たとえば、陽性例の外側強調表示は、第1のテキスト／背景色の組み合わせで示されてもよく、陽性例の内側強調表示は、異なる対照的なテキスト／背景色の組み合わせで示されてもよい。

【0081】

上述したように、ユーザは、キャラクタサブシーケンスの選択を介してキャプチャグループの選択を指定することができる。GUIを用いて、強調表示（または他の表示）を介するユーザ選択を容易にしてもよい。一例を図13に示し、例示的ユーザインターフェイス画面がテーブル形式データディスプレイとともに示される。この例では、図13は、たとえば、ユーザが列値の1つまたは複数の所望の要素を横切ってマウスをドラッグすることによって引き起こされる、列値内の強調表示を示す。なお、ユーザ強調表示が実行される「セル」は、列値の選択を示す色変化を示してもよい。この色変化は、ユーザ強調表示に応答する自動化された強調表示と解釈されてもよい。

【0082】

図14および図15は、テーブル形式ディスプレイからのデータの選択に基づく正規表現およびキャプチャグループの生成を示す例示的なユーザインターフェイス画面である。これらの例では、図14および図15は、テーブル形式データディスプレイ内のユーザ強調表示1401の検出が自動的に表示される追加のユーザインターフェイスウィンドウを示す。ウィンドウは、陽性例を表示するためのフィールド1402と、陰性例を表示するためのフィールドと、テーブル形式データディスプレイからの陽性例の選択に応答して動的に（およびほぼ瞬時に）生成される正規表現を表示するためのフィールドとを含む。これらの例では、列値1401内のユーザ強調表示は、自動化された強調表示内のユーザ強調表示と同等であり得る。したがって、エリアコードのユーザ強調表示により、ユーザ強調表示されたエリアコード1401だけでなく、電話番号の残りの部分も陽性例フィールド1402にポピュレートされる。

【0083】

しかしながら、ユーザ強調表示は、自動強調表示内の性能に限定されないことを理解されたい。たとえば、ユーザ強調表示は、代替的に、他のユーザ強調表示内で実行されてもよい。別の例として、ユーザ強調表示は、代替的に、内側強調表示（たとえば、強調表示されたテキスト内でのさらに強調表示）なしで実行されてもよい。これらの代替例は、「ツイート」または他の長いストリング（たとえばブラウザ「ユーザエージェント」ストリング）を含むデータの列などの半構造化データに特に適している。

【0084】

さらに、対応する正規表現が生成されると、正規表現にマッチする他の列値1402が、追加の自動化された強調表示に基づいて識別され得る。図14および図15に示される例では、追加の自動化された強調表示は、生成された正規表現のキャプチャグループにマッチする、これらの他の列値の要素を示す。追加の自動化された強調表示は、ユーザ強調表示に使用される色とは異なる色を使用して実行されてもよい。

【0085】

図15に示すように、他の例のユーザ選択を示すために、追加のユーザ強調表示が示さ

10

20

30

40

50

れる。追加のユーザ強調表示は、上述の方法と同様の方法で実行され得る。したがって、図15のユーザインターフェイスは、陽性例を表示するためのフィールド1502における他の例のポピュレーションを示す。これは、追加のユーザ強調表示の検出に応答して起こり得る。さらに、生成された正規表現1503は、それが陽性例1502のすべてにマッチするように、動的にかつほぼ瞬時に更新されてもよい。更新された正規表現の生成に応答して、更新された正規表現にマッチする他の列値1504の自動化された強調表示も更新され得る。いくつかの実現形態では、動的カラーコード化も使用され得る。たとえば、マッチは、第1の色（たとえば青色）を使用してカラーコード化されてもよく、陽性例は、第2の色（たとえば緑色）を使用してカラーコード化されてもよく、陰性例は、第3の色（たとえば赤色）を使用してカラーコード化されてもよい。

10

【0086】

図16Aおよび図16Bは、テーブル形式ディスプレイからの陽性例および陰性例の選択に基づく正規表現の生成を示すユーザインターフェイス画面例である。図16A～図16Bでは、陽性例フィールド1602からの個々の例は、陽性例フィールド1603から除去され得、および／または陰性例フィールド1603に移され得る。ユーザインターフェイス内で、これは、たとえば、ユーザが例の1つをクリック（たとえば右クリック）してそれを選択することによって実行されてもよい。選択は、ユーザインターフェイスに、削除オプションおよび変更オプションを含むメニュー1602を表示させることができる。その後、オプションをクリックすると、対応する機能が実行される。

【0087】

20

図16Aおよび図16Bに示される例において、変更オプションのユーザ選択の結果は、選択された例を陰性例フィールド1603に移動させ、正規表現1601を正規表現1604に更新させ、この正規表現1604は、動的にかつほぼ瞬時に生成され得る（たとえばある実施形態では、30ms～9000msの間である）。更新された正規表現1604の生成に応答して、更新された正規表現にマッチする他の列値の自動化された強調表示も、テーブル形式データディスプレイ内で更新され得る。さらに、自動化された強調表示は、陰性例に対応する任意の列値を含む、陰性例の一部または全部に対して実行されてもよく、それは、上で使用された色のいずれとも異なる色を使用して強調表示されてもよく、または他の態様では他の視覚的技法を使用してユーザインターフェイス内で区別されてもよい。

30

【0088】

いくつかの実施形態では、ユーザインターフェイスを介して陰性例を指定することは、図16Aおよび図16Bに示されるように、最初にその例を陽性例として指定し、次いでそれを陰性例に変換することを必要とする必要はない。むしろ、陰性例は、さまざまな方法で指定され得る。たとえば、ユーザは、ユーザインターフェイスを介して列値（たとえば、自動化された強調表示が実行されて、生成された正規表現とマッチすることを示した、他の列値のうちの1つ）を選択する（たとえば右クリックする）ことができ、それによって、オプション（たとえば「新たな反例を作成する」）を含むメニューのディスプレイに、選択された列値を陰性例として指定させることができる。

【0089】

40

したがって、図16Aおよび図16Bに示される例を使用して、更新された正規表現1604の生成に応答して、更新された正規表現にマッチする他の列値の自動化された強調表示も更新され得る。これらの例では、更新された正規表現は、「9」で終わる電話番号を指定する。

【0090】

図14および図15を簡単に参照すると、「抽出」ボタンがユーザによってクリックされるか、または他の態様で選択されると、現在の正規表現1403または1503にマッチするすべてのセル内の強調表示されたテキストフラグメントを抽出するための動作が、開始され得る。図14および図15には示されていないが、いくつかの実施形態では、ユーザインターフェイスは、「抽出」ボタンに加えて、またはその代わりに、他の選択可能

50

なボタンを提供してもよい。たとえば、「置換」ボタンが、ユーザ強調表示された要素をユーザ指定された要素に置換するためのオプションとして提示されてもよい。追加または代替として、1つまたは複数の「削除」ボタンが、事実上、ユーザ強調表示された要素を何にも置換しないオプションとして提示されてもよい。たとえば、「フラグメントを削除」操作および／または「行を削除」操作の一方または両方が実現されてもよく、それは、それぞれ、ユーザ強調表示されたテキストフラグメントまたはいずれかの行のいずれかを削除することになる。さまざまな実施形態において実現され得る追加の操作は、「行を保持」操作、「分割」操作（たとえば、コンマを強調表示し、次いで、コンマ分離成分を別々の複数の新たな列に抽出する）、および「難読化」操作（たとえば、強調表示されたテキスト／キャプチャグループを「#」または他の記号のシーケンスで置き換える）を含み得る。この例では、「抽出」ボタンが選択されたことに応答して、抽出操作が、下流の操作によって実行されるべき変換スクリプトのリストに追加され得る。いくつかの実施形態では、変換スクリプトのリストは、ユーザによるレビュー／修正のためにユーザインターフェイスの一部に表示されてもよい。代替的に、抽出操作は、その場で実行されて、レゲックスキャプチャグループの内容（たとえば、陽性例のユーザ強調表示部分に対応する要素）を含む新たな列を生成してもよい。図14および図15に示される例では、エリアコードの新たな列および／または新たなテーブルが、「抽出」ボタンの選択に応答して生成されてもよい。

10

【0091】

図17は、本明細書に記載する1つまたは複数の実施形態による、テーブル形式ディスプレイからのデータの選択に基づく正規表現およびキャプチャグループの生成を示す別の例示的なユーザインターフェイス画面である。

20

【0092】

VI. スパン上で最長共通サブシーケンサルゴリズムを用いた正規表現生成

本明細書で説明されるさらなる態様は、1つまたは複数のデータ入力キャラクタシーケンスからのLCSアルゴリズムに基づく正規表現の生成に関するが、正規表現生成器110は、例のいくつかのみに存在するキャラクタを取り扱うこともできる。いくつかの入力例においてのみ存在するキャラクタを取り扱うために、正規表現コードの最小発生数および最大発生数の両方が追跡されるスパンが定義され得る。たとえば、「9pm」および「9 pm」のキャラクタシーケンス入力については、数字と「pm」テキストとの間に任意選択の空白が存在する。そのような場合、所与の入力例のすべてに一定のスパン（たとえば「9」と「pm」との間の単一の空白）が存在しないかもしれない場合、最小発生数はゼロに設定されてもよい。次いで、これらの最小数および最大数は、正規表現マルチプリシティ構文にマッピングされ得る。最長共通サブシーケンス（LCS）アルゴリズムを、すべての入力例に現れない「任意選択の」スパン（たとえばゼロの最小長さ）を含む、入力例から導出されたキャラクタのスパン上で実行してもよい。以下で説明するように、連続するスパンは、LCSアルゴリズムの実行中にマージされてもよい。そのような場合において、一緒に担持される追加の任意選択のスパンが連続して出現することに終わるとき、LCSアルゴリズムは、それらの任意選択のスパン上でも同様に再帰的に実行されてもよい。すなわち、LCSアルゴリズムの実行は、その性質上、再帰的であるが、これらの場合、LCSアルゴリズム全体を、再帰的に実行してもよい（たとえば、再帰的LCSアルゴリズムを再帰的に実行する）。他の技術的利点の中でもとりわけ、これは、より短く、よりクリーンで、より可読性のある正規表現生成を可能にし得る。たとえば、(am | am)（すなわち、amの前に任意選択の空白を有する）は、LCSアルゴリズムを再帰的に実行せずに生成されるかもしれず、一方、LCSアルゴリズムを再帰的に実行すると、正規表現は、より短く、よりクリーンな(?am)として生成される結果となり得る。

30

40

【0093】

図18は、本明細書で説明する1つまたは複数の実施形態による、最長共通サブシーケンス（LCS）アルゴリズムを使用して、任意選択のスパンを含む正規表現を生成するための処理を示すフローチャートである。ステップ1801において、正規表現生成器11

50

0 は、陽性の正規表現例に対応する 1 つまたは複数のキャラクタシーケンスを入力データとして受け取ってもよい。ステップ 1802 において、正規表現生成器 110 は、キャラクタシーケンスを正規表現コードに変換してもよい。したがって、ステップ 1801 およびステップ 1802 は、上述の、先の対応する例と同様でも同一でもよい。次いで、ステップ 1802 において、正規表現コードは、スパンデータ構造（またはスパン）にさらに変換され得る。上述したように、各スパンは、キャラクタクラスコード（たとえばレグエクスクード）および繰り返しカウント範囲（たとえば最小カウントおよび／または最大カウント）を記憶するデータ構造を含むことができる。ステップ 1804 において、正規表現生成器 110 は、LCS アルゴリズムを実行して、アルゴリズムへの入力としてスパンのセットを提供することができる。この例における LCS アルゴリズムの出力は、LCS アルゴリズムの出力内の任意選択のスパンに対応する、ゼロに等しい最小繰り返しカウント範囲を有する少なくとも 1 つのスパンを含むスパンの出力セットを含むことができる。最後に、ステップ 1805 において、正規表現生成器 110 は、任意選択のスパンを含む LCS アルゴリズムの出力の出力に基づいて正規表現を生成することができる。

【0094】

図 19 は、最長共通サブシーケンス（LCS）アルゴリズムを用いた正規表現の生成を示す例示的な図であり、生成された正規表現は、任意選択のスパンを含む。この例では、2 つの入力データキャラクタシーケンスは、「8am」および「9 pm」である。入力データキャラクタシーケンスは、上述したように、まず正規表現コードに変換され（ステップ 1802）、次にスパンに変換される（ステップ 1803）。スパンは、LCS アルゴリズムへの入力として提供され得（ステップ 1804）、LCS 出力は、任意選択のスパン $\langle 0, 1 \rangle$ を含み、任意選択の単一の空白が数字および 2 文字のテキストシーケンスであり得ることを示す。すなわち、この例における上付き表記は、先行するコード（たとえば $Z =$ 空白）に適用される 2 つの数字、最小繰り返しカウント範囲（たとえば 0）および最大繰り返しカウント範囲（たとえば 1）を含み得る。最後に、正規表現は、LCS アルゴリズムの出力スパンに基づいて生成されてもよく、任意選択のスパンは、対応する正規表現コード「 pZ^* 」に変換されてもよい。

【0095】

いくつかの実施形態では、LCS アルゴリズムの実行中の正規表現生成器 110 による任意選択の空白の描出および使用は、性能および可読性に関してさらなる技術的利点を提供し得る。たとえば、正規表現を生成するとき、場合によっては、すべての所与の例の間で共通するキャラクタと、それらの例のうちのいくつかにおいてのみ存在するキャラクタとの両方を扱うことができることが望ましい。

【0096】

ある実施形態では、各スパンデータ構造について、カテゴリコードの最小発生数およびカテゴリコードの最大発生数の両方が追跡され得る。所与の例の 1 つまたは複数においてスパンが全く存在しない場合、最小はゼロに設定される。別の例として、繰り返しで示された月を扱うための正規表現を生成するために、最小数および最大数を、次いで、中括弧を伴う正規表現マルチプリシティ構文（たとえば $[A-Za-z]\{3,9\}$ ）にマッピングしてもよい。

【0097】

いくつかの実施形態では、正規表現生成器 110 は、各スパンについて最小発生数および最大発生数を追跡してもよいが、追加の実施詳細を処理してもよい。たとえば、任意選択のスパンを取り扱うこととキャラクタのスパン上で LCS を実行することとの組み合わせの結果として、正規表現生成器 110 は、LCS アルゴリズムの実行を通して、連続的なスパンを検出し、マージするように構成され得る。加えて、一緒に担持されている任意の追加の任意選択のスパンが、時々、連続的に現れ、LCS アルゴリズムがそれら上でも同様に再帰的に実行されることが望ましい場合がある。たとえば、場合によっては、正規表現生成器 110 は、任意選択のシーケンス要素と必要なシーケンス要素（たとえばスパン）との間のより少ない遷移を好む（または重み付けする）よう、LCS アルゴリズムを修正および／または拡張する。たとえば、任意選択のスパンを一緒にグループ化すること

は、正規表現内で使用されなければならないグループ化括弧の数を最小にすることができ、したがって、生成された正規表現の人間の可読性を改善することができる。場合によっては、結果として生じる長さが、任意選択のスパンを考慮した後でさえ等しい場合、正規表現生成器 110 は、任意選択のスパンと必要なスパンとの間の遷移がより少ない代替物に対する選好を示してもよい。たとえば、場合によっては、ある標準 L C S アルゴリズムは、その判断点でより長いシーケンスの選択を好むように実現され得る。しかしながら、選択肢が等しい長さのものである判断点では、構成選好が正規表現生成器 110 にプログラムされてもよい。1つのそのような構成選好は、たとえば、（任意選択のスパンが考慮されると）より短いシーケンスを好むことであり得る。したがって、この構成内のカスタマイズされた L C S は、（必要なスパンの）より長いシーケンスおよび（必要なスパンおよび任意選択のスパンの合計の）より短いシーケンスを同時に最適化することができる。

10

【0098】

いくつかの実施形態では、生成された正規表現は、任意選択のスパンで正規表現を開始するのではなく、必要なスパン（これは、人間の読者に対する精神的拠り所としても働き得る）で開始する場合、より可読性があり得る。したがって、場合によっては、結果として得られる選択肢が等しい数の遷移を有する場合、より早期の非任意選択のスパンを有する選択肢が選択され得る。加えて、正規表現生成器 110 によって実行される L C S アルゴリズムは、いくつかの実施形態では、正規表現内においてすべての空白（空白に対応する任意選択のスパンを含む）を右にプッシュするように構成されてもよい。すべての空白を右にプッシュすることによって、空白のスパンが一緒にマージされ得る機会が増大する可能性があり、これは、結果として生じる正規表現を単純化し、かつ可読性を改善し得る。このように、L C S アルゴリズムの実行中に、サブストリングの 2 つのセットが同じ L C S を有すると判断された場合、サブストリングの 2 つのセットのうちの 1 つのセットを恣意的に選択する代わりに、可読性の改善を容易にするセットを選択してもよい。さらに、いくつかの実施形態では、L C S アルゴリズムは、可読性を改善するために、より多くの数の必要なスパンおよび／またはより少ない数の任意選択のスパンを好むように構成されてもよい。

20

【0099】

上述したように、場合によっては、陰性例は、任意選択のスパンに基づいてもよい。たとえば、ユーザは、「ab」および「a2b」の陽性例ならびに「a3b」の陰性例を提供してもよい。この場合、ある例示的な実現例は失敗し得、なぜならば、それは、必要とされるスパンのみに基づいて区別しようとし得、「2」の桁は任意選択のスパンにあるからである。そのような場合、ユーザは、失敗に対して警告されることができ、生成された正規表現を手動で修復するために、および／または陰性例の一部を除去するために、ユーザーインターフェイスを介して選択肢を提供されることができる。

30

【0100】

いくつかの実施形態では、R E S T サービスから戻って来る J S O N の一部として返される isSuccess が存在し得る。いくつかの実施形態では、生成されたレゲックスは、isSuccess = 偽のとき、異なる色（たとえば赤色）となってもよい。

【0101】

V I I . コンビナトリック最長共通サブシーケンスアルゴリズムを用いた正規表現生成
本明細書で説明されるさらなる態様は、正規表現生成器 110 によって実行される L C S アルゴリズムが複数回実行されて、「正しい」正規表現（たとえば、すべての所与の陽性例と適切にマッチし、すべての所与の陰性例を適切に除外する正規表現）を生成し得る、および／または最も望ましいもしくは最適な正規表現が選択され得る複数の正しい正規表現を生成し得るコンビナトリック探索に関する。たとえば、コンビナトリック探索中、全 L C S アルゴリズムおよび正規表現生成処理は、テキスト処理方向の異なる組み合わせ／置換、異なる位置指定、および L C S アルゴリズムの他の異なる特性を含めて、複数回実行されてもよい。

40

【0102】

50

図20は、最長共通サブシーケンス（LCS）アルゴリズムのコンビナトリックな実行に基づいて正規表現を生成するための処理を示すフローチャートである。ステップ2001において、正規表現生成器110は、陽性例に対応する入力データキャラクタシーケンスを受け取り得る。ステップ2002において、正規表現生成器110は、LCSアルゴリズムのための実行技法のさまざまな異なる組み合わせに対して反復することができる。この例に示されるように、ステップ2002の各反復の間に、正規表現生成器110は、以下のLCSアルゴリズム実行パラメータ（または特性）、すなわち位置指定子（すなわち、位置指定なし、行の始まりで位置指定、行の終わりで位置指定）、処理方向（すなわち、右から左の順序、左から右の順序）、空白プッシュ（すなわち、空白プッシュを行うかまたは行わない）、およびスパンを隠す（collapse）（すなわち、スパンを隠すことを行うかまたは行わない）、の異なる組み合わせを選択し得る。ステップ2003において、LCSアルゴリズムは、入力データキャラクタシーケンスにおいて（または、入力キャラクタシーケンスが最初に変換された場合には正規表現コードにおいて）実行され、LCSアルゴリズムは、ステップ2002において選択されたパラメータ／特性に基づいて構成される。ステップ2004において、LCSアルゴリズムの出力は、正規表現生成器110によって格納されてもよく、アルゴリズムによってLCSが成功裡に識別されたか否か、および対応する正規表現の長さなどのデータを含み得る。ステップ2005において、処理は、LCSアルゴリズムがコンビナトリック探索のパラメータ／特性のすべての可能な組み合わせで実行されるまで、反復してもよい。最後に、ステップ2006において、LCSの1つからの特定の出力が、最適出力（たとえば、成功および正規表現長に基づく）として選択され、正規表現が、選択されたLCSアルゴリズム出力に基づいて生成され得る。

【0103】

さまざまな実施形態において、図20を参照して上述したもののようなコンビナトリック探索は、パラメータ／特性のさまざまな異なる組み合わせに対して実行されてもよい。たとえば、幾つかの実施形態では、LCSアルゴリズムは、正規表現をテキストの始まりに位置指定するためにカレット記号^を使用し、および／または正規表現をテキストの終わりに位置指定するためにドル記号\$を使用してもよい。場合によっては、そのような位置指定は、より短い正規表現を生成する結果となり得る。位置指定子は、ユーザがストリングの始まりおよび／または終わりに特定のパターンを発見することを望む場合に特に有用であり得る。たとえば、ユーザは始まりに製品名を望む場合がある。LCSアルゴリズムを、製品名を記述するさまざまな数の単語と混同するのを避けるために、下の画像に示されるように、カレットを使用して、レゲックスをストリングの始まりに位置指定することができる。

【0104】

さらに、いくつかの実施形態では、LCSアルゴリズムは、順方向または逆方向のいずれかである入力データを用いて実行され得る（または同様に、LCSアルゴリズムは、通常の順序で入力データを受け取り、次いでアルゴリズムを実行する前に順序を逆にするように構成されてもよい）。したがって、いくつかの実施形態では、入力キャラクタシーケンスまたはコードのペアに対して実行され得るLCSアルゴリズムのコンビナトリック探索は、以下のものであってもよい。

【0105】

1. 通常（右から左へ）の順序、開始または終了に対して位置指定しない
 2. 通常（右から左へ）の順序、カレット^を使用して行の始まりに対して位置指定する
 3. 通常（右から左へ）の順序、ドル\$を使用して行の終わりに対して位置指定する
 4. 逆（左から右へ）の順序、始まりまたは終わりに対して位置指定しない
 5. 逆（左から右へ）の順序、カレット^を使用して行の始まりに対して位置指定する
 6. 逆（左から右へ）の順序で、ドル\$を使用して行の終わりに対して位置指定する
- この例では、LCSの6つの実行のうち、最も短い結果の正規表現が選択されてもよい（ステップ2006）。

10

20

30

40

50

【0106】

幾つかの実施形態では、LCSアルゴリズムのコンビナトリック探索は、greedy量指定子「?」および非greedy量指定子「??」に対して反復してもよい。たとえば、デフォルトでは、任意選択のスパンが存在する場合、1つの疑問符が発せられ、たとえば、任意選択のミドルイニシャルを有するファーストネームおよびラストネームについては[A-Z]+(?:[A-Z]/.)*[A-Z]+である。greedy量指定子を使用する場合に満足のいく正規表現が見つからない場合には、コンビナトリック探索は、すべての疑問符量指定子を二重疑問符量指定子（たとえば[A-Z]+(?:[A-Z]/.)*?[A-Z]+）に置き換えることを試みることができる。二重疑問符は、非greedy量指定子に対応し、それは、マッチを見つけるために、下流の正規表現マッチャーにバックトラッキングモードに入るように命令することができる。

10

【0107】

加えて、いくつかの実施形態では、LCSアルゴリズムのコンビナトリック探索は、右側の空白を好むかどうかに対して反復することもできる。たとえば、上記のように、空白を右にプッシュするいくつかの実施形態において、たとえば、LCSアルゴリズムが、他の態様であれば等しい選択肢の恣意的な選択に直面する場合、空白スパンがともにマージされ、全体のスパンの数がより少なくなる結果となることを期待して、ある戦略が使用されてもよい。この特徴は、別の選択肢をコンビナトリック探索に追加し、すなわち、空白を右にプッシュするか、または判断を任意のままにする従来のLCSアプローチに従って実行するかのいずれかにするようにする。

【0108】

20

さらに、いくつかの実施形態では、LCSアルゴリズムのコンビナトリック探索はまた、元のストリング上でLCSを実行することによって、すべての例で共通のリテラルに対する走査／非走査に対して反復してもよい。そのような実施形態では、LCSアルゴリズムは、共通単語を識別し、整列するように構成され得る。本明細書で使用される場合、「共通単語」は、すべての陽性例において現れる単語を指し得る。いったん共通単語が識別されると、そのスパンタイプは、LETTERからWORDに変換されてもよく、次いで、LCSアルゴリズムを介する後続の実行は、それに自然に整列してもよい。

【0109】

したがって、以下の例においては、コンビナトリック探索は、完全なLCSアルゴリズムが実行される96回に達するように、いくつかのパラメータ／特性に対して反復してもよい。この例において反復されるべきさまざまなパラメータ／特性は以下のとおりである：

30

- ・位置指定子（3）（値＝^, \$, またはどちらもない）
- ・空白をプッシュ（2）（値＝YesまたはNo）
- ・低濃度スパンのワイルドカードへの合体（2）（値＝YesまたはNo）
- ・Greedy量指定子?（2）（値＝YesまたはNo）
- ・LCSアルゴリズムの共通トークン上での整列（2）（値＝YesまたはNo）
- ・別のスパンとして扱われる文字「/pL」および数字「/pN」を保持することに対して、英数字を表すよう「/w」を使用（2）（値＝YesまたはNo）

上述したように、この例では、完全なLCSアルゴリズムは96回（たとえば $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$ ）実行される。

40

【0110】

しかしながら、他の実施形態では、正規表現生成器110は、性能向上を提供してもよく、それによって、上記のリストのうちの最初の3つの特性のみ（位置指定子、空白のプッシュ、および低濃度スパンのワイルドカードへの合体）が、コンビナトリック探索に加わってもよい。これは、遙かにより少ない数の完全なLCSアルゴリズムが実行されることになる結果となり得る（たとえば $3 \times 2 \times 2 = 12$ 回）。そのような実施形態では、上記リストの最後の3つの特性（Greedy量指定子、LCSアルゴリズムの共通トークン上での整列、ならびに別のスパンとして扱われる文字「/pL」および数字「/pN」を保持することに対して、英数字を表すよう「/w」を使用）は、コンビナトリック探索に加わらないが、これらの特性は、最後に、個々におよび逐次、テストされ得る。そのような実施形態

50

においては技術的利点の実現され得、なぜならば、探索空白をこのように分割することは、それでも、満足のいく正規表現が、性能において約 8 倍の高速化を伴って、見いだされる結果となり得るからである。

【0111】

説明すると、コンビナトリック探索の以下の例は、前の例よりも性能上の利点を提供し得る。この例では、コンビナトリック探索は、反復されるべき以下のパラメータ／特性に基づいて実行され得る：

- ・位置指定 (3) : BEGINNING_OF_LINE_MODE (行の始まりモード), END_OF_LINE_MODE (行の終わりモード), NO_EOL_MODE (行の終わりなしモード)
- ・順序／方向 (2) : 右から左 (通常) L C S 対左から右 (逆) L C S
- ・プッシュ (2) : L C S アルゴリズム内で空白を右にプッシュしようとするか否か
- ・ワイルドカードに圧縮 (2) : 時々発生するにすぎないスパンの長いシーケンスをワイルドカード.*?に圧縮しようとするか否か

この例におけるコンビナトリックは、完全なアルゴリズムを $3 * 2 * 2 * 2 = 24$ 回実行する結果となり得る。次いで、正規表現生成器 110 は、L C S アルゴリズムの 24 個の結果のうち最良のものを取り得、ここで、「最良」とは、(a) L C S アルゴリズムが成功したこと、および (b) 最短正規表現が生成されたこと、を意味し得る。次いで、正規表現生成器 110 は、以下の 3 つの追加のタスクを実行することができる：

1. 空白、句読点、または記号によって途切れない文字および数字のシーケンスを、/w の生成されたレゲックスに対応する、ALPHANUMERIC と呼ばれる新たなスパンタイプ E まで圧縮することを試みる。これは、クリックストリームログからの I P v 6 アドレスに見られる 16 進数に対して有用であり得る (2019 年 4 月からのノベルティ 64 を参照されたい)。

【0112】

2. greedy 量指定子?の代わりに非greedy量指定子??を使用することを試みる。
3. リテラル上で整列を試みる。

【0113】

ハードウェア概要

図 21 は、ある実施形態を実現するための分散型システム 2100 の簡略図を示す。図示される実施形態において、分散型システム 2100 は、1 つ以上の通信ネットワーク 2110 を介してサーバ 2112 に結合された 1 つ以上のクライアントコンピューティングデバイス 2102、2104、2106、および 2108 を含む。クライアントコンピューティングデバイス 2102、2104、2106、および 2108 は、1 つ以上のアプリケーションを実行するように構成され得る。

【0114】

さまざまな実施形態において、サーバ 2112 は、本開示に記載される正規表現の自動化された生成を可能にする 1 つ以上のサービスまたはソフトウェアアプリケーションを実行するように適合され得る。たとえば、特定の実施形態では、サーバ 2112 は、クライアントデバイスから送信されたユーザ入力データを受信することができ、ユーザ入力データは、クライアントデバイスで表示されたユーザインターフェイスを介して、クライアントデバイスによって受信される。次いで、サーバ 2112 は、ユーザ入力データを、ユーザインターフェイスを介して表示するためにクライアントデバイスに送信される正規表現に変換することができる。

【0115】

特定の実施形態では、サーバ 2112 はまた、非仮想環境および仮想環境を含み得る他のサービスまたはソフトウェアアプリケーションを提供し得る。いくつかの実施形態では、これらのサービスは、クライアントコンピューティングデバイス 2102、2104、2106 および／または 2108 のユーザに対して、サービスとしてのソフトウェア (Software as a Service : SaaS) モデルのようなウェブベースのサービスまたはクラウドサービスとして提供され得る。クライアントコンピューティングデバイス 2102、2

104、2106および／または2108を操作するユーザは、1つ以上のクライアントアプリケーションを利用してサーバ2112とやり取りすることで、これらのコンポーネントによって提供されるサービスを利用し得る。

【0116】

図21に示される構成では、サーバ2112は、サーバ2112によって実行される機能を実現する1つ以上のコンポーネント2118、2120および2122を含み得る。これらのコンポーネントは、1つ以上のプロセッサ、ハードウェアコンポーネント、またはそれらの組合わせによって実行され得るソフトウェアコンポーネントを含み得る。分散型システム2100とは異なり得る多種多様なシステム構成が可能であることが認識されるはずである。したがって、図21に示される実施形態は、実施形態のシステムを実現するための分散型システムの一例であり、限定するよう意図されたものではない。

10

【0117】

ユーザは、クライアントコンピューティングデバイス2102、2104、2106および／または2108を用いて、1つまたは複数のアプリケーションを実行し、それは、本開示の教示に従って正規表現を生成してもよい。クライアントデバイスは、当該クライアントデバイスのユーザが当該クライアントデバイスと対話することを可能にするインターフェイスを提供し得る。クライアントデバイスはまた、このインターフェイスを介してユーザに情報を出力してもよい。図21は4つのクライアントコンピューティングデバイスだけを示しているが、任意の数のクライアントコンピューティングデバイスがサポートされ得る。

20

【0118】

クライアントデバイスは、ポータブルハンドヘルドデバイス、パーソナルコンピュータおよびラップトップのような汎用コンピュータ、ワークステーションコンピュータ、ウェアラブルデバイス、ゲームシステム、シンクライアント、各種メッセージングデバイス、センサまたはその他のセンシングデバイスなどの、さまざまな種類のコンピューティングシステムを含み得る。これらのコンピューティングデバイスは、さまざまな種類およびバージョンのソフトウェアアプリケーションおよびオペレーティングシステム（たとえばMicrosoft Windows（登録商標）、Apple Macintosh（登録商標）、UNIX（登録商標）またはUNIX系オペレーティングシステム、Linux（登録商標）またはLinux系オペレーティングシステム、たとえば、各種モバイルオペレーティングシステム（たとえばMicrosoft Windows Mobile（登録商標）、iOS（登録商標）、Windows Phone（登録商標）、Android（登録商標）、BlackBerry（登録商標）、Palm OS（登録商標））を含むGoogle Chrome（登録商標）OS）を含み得る。ポータブルハンドヘルドデバイスは、セルラーフォン、スマートフォン（たとえばiPhone（登録商標））、タブレット（たとえばiPad（登録商標））、携帯情報端末（PDA）などを含み得る。ウェアラブルデバイスは、Google Glass（登録商標）ヘッドマウントディスプレイおよびその他のデバイスを含み得る。ゲームシステムは、各種ハンドヘルドゲームデバイス、インターネット接続可能なゲームデバイス（たとえばKinect（登録商標）ジェスチャ入力デバイス付き／無しのMicrosoft Xbox（登録商標）ゲーム機、Sony PlayStation（登録商標）システム、Nintendo（登録商標）が提供する各種ゲームシステムなど）を含み得る。クライアントデバイスは、各種インターネット関連アプリケーション、通信アプリケーション（たとえばEメールアプリケーション、ショートメッセージサービス（SMS）アプリケーション）のような多種多様なアプリケーションを実行可能であってもよく、各種通信プロトコルを使用してもよい。

30

40

【0119】

ネットワーク2110は、利用可能な多様なプロトコルのうちのいずれかを用いてデータ通信をサポートできる、当該技術の当業者には周知のいずれかの種類のネットワークであればよく、上記プロトコルは、TCP/IP（伝送制御プロトコル／インターネットプ

50

ロトコル)、SNA(システムネットワークアーキテクチャ)、IPX(インターネットパケット交換)、AppleTalk(登録商標)などを含むがこれらに限定されない。単に一例として、ネットワーク2110は、ローカルエリアネットワーク(LAN)、Ethernet(登録商標)に基づくネットワーク、トークンリング、ワイドエリアネットワーク(WAN)、インターネット、仮想ネットワーク、仮想プライベートネットワーク(VPN)、イントラネット、エクストラネット、公衆交換電話網(PSTN)、赤外線ネットワーク、無線ネットワーク(たとえば電気電子学会(IEEE)802.11プロトコルスイートのいずれかの下で動作する無線ネットワーク、Bluetooth(登録商標)および/または任意の他の無線プロトコル)、および/またはこれらおよび/または他のネットワークの任意の組み合わせを含み得る。

10

【0120】

サーバ2112は、1つ以上の汎用コンピュータ、専用サーバコンピュータ(一例としてPC(パーソナルコンピュータ)サーバ、UNIX(登録商標)サーバ、ミッドレンジサーバ、メインフレームコンピュータ、ラックマウント型サーバなどを含む)、サーバファーム、サーバクラスタ、またはその他の適切な構成および/または組み合わせで構成されてもよい。サーバ2112は、仮想オペレーティングシステムを実行する1つ以上の仮想マシン、または仮想化を伴う他のコンピューティングアーキテクチャを含み得る。これはたとえば、サーバに対して仮想記憶装置を維持するように仮想化できる論理記憶装置の1つ以上のフレキシブルプールなどである。各種実施形態において、サーバ2112を、上記開示に記載の機能を提供する1つ以上のサービスまたはソフトウェアアプリケーションを実行するように適合させてもよい。

20

【0121】

サーバ2112内のコンピューティングシステムは、上記オペレーティングシステムのうちのいずれかを含む1つ以上のオペレーティングシステム、および、市販されているサーバオペレーティングシステムを実行し得る。また、サーバ2112は、HTTP(ハイパーテキスト転送プロトコル)サーバ、FTP(ファイル転送プロトコル)サーバ、CGI(コモンゲートウェイインターフェイス)サーバ、JAVA(登録商標)サーバ、データベースサーバなどを含むさまざまなさらに他のサーバアプリケーションおよび/または中間層アプリケーションのうちのいずれかを実行し得る。例示的なデータベースサーバは、Oracle(登録商標)、Microsoft(登録商標)、Sybase(登録商標)、IBM(登録商標)(International Business Machines)などから市販されているものを含むが、それらに限定されない。

30

【0122】

いくつかの実現例において、サーバ2112は、クライアントコンピューティングデバイス2102、2104、2106および2108のユーザから受信したデータフィードおよび/またはイベントアップデートを解析および整理統合するための1つ以上のアプリケーションを含み得る。一例として、データフィードおよび/またはイベントアップデートは、センサデータアプリケーション、金融株式相場表示板、ネットワーク性能測定ツール(たとえば、ネットワークモニタリングおよびトラフィック管理アプリケーション)、クリックストリーム解析ツール、自動車交通モニタリングなどに関連するリアルタイムのイベントを含んでもよい、1つ以上の第三者情報源および連続データストリームから受信される、Twitter(登録商標)フィード、Facebook(登録商標)アップデートまたはリアルタイムのアップデートを含み得るが、それらに限定されない。サーバ2112は、データフィードおよび/またはリアルタイムのイベントをクライアントコンピューティングデバイス2102、2104、2106および2108の1つ以上の表示デバイスを介して表示するための1つ以上のアプリケーションも含み得る。

40

【0123】

分散型システム2100はまた、1つ以上のデータリポジトリ2114、2116を含み得る。特定の実施形態において、これらのデータリポジトリを用いてデータおよびその他の情報を格納することができる。たとえば、データリポジトリ2114、2116のう

50

ちの1つ以上を用いて、システムにより生成された正規表現とマッチする新たなデータの列のような情報を格納することができる。データリポジトリ2114、2116は、さまざまな場所に存在し得る。たとえば、サーバ2112が使用するデータリポジトリは、サーバ2112のローカル位置にあってもよく、またはサーバ2112から遠隔の位置にあってもよく、ネットワークベースの接続または専用接続を介してサーバ2112と通信する。データリポジトリ2114、2116は、異なる種類であってもよい。特定の実施形態において、サーバ2112が使用するデータリポジトリは、データベース、たとえば、Oracle Corporation（登録商標）および他の製造業者が提供するデータベースのようなりレーショナルデータベースであってもよい。これらのデータベースのうちの1つ以上を、SQLフォーマットのコマンドに応じて、データの格納、アップデート、およびデータベースとの間での取り出しを可能にするように適合させてもよい。

10

【0124】

特定の実施形態では、データリポジトリ2114、2116のうちの1つ以上は、アプリケーションデータを格納するためにアプリケーションによって用いられてもよい。アプリケーションが使用するデータリポジトリは、たとえば、キー値ストアリポジトリ、オブジェクトストアリポジトリ、またはファイルシステムがサポートする汎用ストレージリポジトリのようなさまざまな種類のものであってもよい。

【0125】

特定の実施形態において、本開示に記載される機能は、クラウド環境を介してサービスとして提供され得る。図22は、特定の例に係る、各種サービスをクラウドサービスとして提供し得るクラウドベースのシステム環境の簡略化されたブロック図である。図22に示される例において、クラウドインフラストラクチャシステム2202は、ユーザが1つ以上のクライアントコンピューティングデバイス2204、2206および2208を用いて要求し得る1つ以上のクラウドサービスを提供し得る。クラウドインフラストラクチャシステム2202は、サーバ2112に関して先に述べたものを含み得る1つ以上のコンピュータおよび／またはサーバを含み得る。クラウドインフラストラクチャシステム2202内のコンピュータは、汎用コンピュータ、専用サーバコンピュータ、サーバファーム、サーバクラスタ、またはその他任意の適切な配置および／または組み合わせとして編成され得る。

20

【0126】

ネットワーク2210は、クライアント2204、2206、および2208と、クラウドインフラストラクチャシステム2202との間におけるデータの通信および交換を容易にし得る。ネットワーク2210は、1つ以上のネットワークを含み得る。ネットワークは同じ種類であっても異なる種類であってもよい。ネットワーク2210は、通信を容易にするために、有線および／または無線プロトコルを含む、1つ以上の通信プロトコルをサポートし得る。

30

【0127】

図22に示される例は、クラウドインフラストラクチャシステムの一例にすぎず、限定を意図したものではない。なお、その他いくつかの例において、クラウドインフラストラクチャシステム2202が、図22に示されるものよりも多くのコンポーネントもしくは少ないコンポーネントを有していてもよく、2つ以上のコンポーネントを組み合わせてもよく、または、異なる構成または配置のコンポーネントを有していてもよいことが、理解されるはずである。たとえば、図22は3つのクライアントコンピューティングデバイスを示しているが、代替例においては、任意の数のクライアントコンピューティングデバイスがサポートされ得る。

40

【0128】

クラウドサービスという用語は一般に、サービスプロバイダのシステム（たとえばクラウドインフラストラクチャシステム2202）により、インターネット等の通信ネットワークを介してオンデマンドでユーザにとって利用可能にされるサービスを指すのに使用される。典型的に、パブリッククラウド環境では、クラウドサービスプロバイダのシステム

50

を構成するサーバおよびシステムは、顧客自身のオンプレミスサーバおよびシステムとは異なる。クラウドサービスプロバイダのシステムは、クラウドサービスプロバイダによって管理される。よって、顧客は、別途ライセンス、サポート、またはハードウェアおよびソフトウェアリソースをサービスのために購入しなくても、クラウドサービスプロバイダが提供するクラウドサービスを利用できる。たとえば、クラウドサービスプロバイダのシステムはアプリケーションをホストし得るとともに、ユーザは、アプリケーションを実行するためにインフラストラクチャリソースを購入しなくても、インターネットを介してオンデマンドでアプリケーションをオーダーして使用し得る。クラウドサービスは、アプリケーション、リソースおよびサービスに対する容易でスケーラブルなアクセスを提供するように設計される。いくつかのプロバイダがクラウドサービスを提供する。たとえば、ミドルウェアサービス、データベースサービス、Java（登録商標）クラウドサービスなどのいくつかのクラウドサービスが、カリフォルニア州レッドウッド・ショアーズのOracle Corporation（登録商標）から提供される。

10

【0129】

特定の実施形態において、クラウドインフラストラクチャシステム2202は、ハイブリッドサービスモデルを含む、サービスとしてのソフトウェア（SaaS）モデル、サービスとしてのプラットフォーム（PaaS）モデル、サービスとしてのインフラストラクチャ（IaaS）モデルなどのさまざまなモデルを使用して、1つ以上のクラウドサービスを提供し得る。クラウドインフラストラクチャシステム2202は、各種クラウドサービスのプロビジョンを可能にする、アプリケーション、ミドルウェア、データベース、およびその他のリソースのスイートを含み得る。

20

【0130】

SaaSモデルは、アプリケーションまたはソフトウェアを、インターネットのような通信ネットワークを通して、顧客が基本となるアプリケーションのためのハードウェアまたはソフトウェアを購入しなくても、サービスとして顧客に配信することを可能にする。たとえば、SaaSモデルを用いることにより、クラウドインフラストラクチャシステム2202がホストするオンデマンドアプリケーションに顧客がアクセスできるようにし得る。Oracle Corporation（登録商標）が提供するSaaSサービスの例は、人的資源／資本管理のための各種サービス、カスタマー・リレーションシップ・マネジメント（CRM）、エンタープライズ・リソース・プランニング（ERP）、サプライチェーン・マネジメント（SCM）、エンタープライズ・パフォーマンス・マネジメント（EPM）、解析サービス、ソーシャルアプリケーションなどを含むがこれらに限定されない。

30

【0131】

IaaSモデルは一般に、インフラストラクチャリソース（たとえばサーバ、ストレージ、ハードウェアおよびネットワーキングリソース）を、クラウドサービスとして顧客に提供することにより、柔軟な計算およびストレージ機能を提供するために使用される。各種IaaSサービスがOracle Corporation（登録商標）から提供される。

【0132】

PaaSモデルは一般に、顧客が、環境リソースを調達、構築、または管理しなくても、アプリケーションおよびサービスを開発、実行、および管理することを可能にするプラットフォームおよび環境リソースをサービスとして提供するために使用される。Oracle Corporation（登録商標）が提供するPaaSサービスの例は、Oracle Java Cloud Service（JCS）、Oracle Database Cloud Service（DBCS）、データ管理クラウドサービス、各種アプリケーション開発ソリューションサービスなどを含むがこれらに限定されない。

40

【0133】

クラウドサービスは一般に、オンデマンドのセルフサービスベースで、サブスクリプションベースで、柔軟にスケーラブルで、信頼性が高く、可用性が高い、安全なやり方で提

50

供される。たとえば、顧客は、サブスクリプションオーダーを介し、クラウドインフラストラクチャシステム 2202 が提供する 1 つ以上のサービスをオーダーしてもよい。次いで、クラウドインフラストラクチャシステム 2202 は、処理を実行することにより、顧客のサブスクリプションオーダーで要求されたサービスを提供する。クラウドインフラストラクチャシステム 2202 を、1 つまたは複数のクラウドサービスを提供するように構成してもよい。

【0134】

クラウドインフラストラクチャシステム 2202 は、さまざまなデプロイメントモデルを介してクラウドサービスを提供し得る。パブリッククラウドモデルにおいて、クラウドインフラストラクチャシステム 2202 は、第三者クラウドサービスプロバイダによって所有されていてもよく、クラウドサービスは一般のパブリックカスタマーに提供される。このカスタマーは個人または企業であってもよい。プライベートクラウドモデルでは、クラウドインフラストラクチャシステム 2202 がある組織内で（たとえば企業組織内で）機能してもよく、サービスはこの組織内の顧客に提供される。たとえば、この顧客は、人事部、給与部などの企業のさまざまな部署であってもよく、企業内の個人であってもよい。コミュニティクラウドモデルでは、クラウドインフラストラクチャシステム 2202 および提供されるサービスは、関連コミュニティ内のさまざまな組織で共有されてもよい。上記モデルの混成モデルなどのその他各種モデルが用いられてもよい。

【0135】

クライアントコンピューティングデバイス 2204、2206、および 2208 は、異なるタイプであってもよく（たとえば図 21 に示されるデバイス 2102、2104、2106 および 2108）、1 つ以上のクライアントアプリケーションを操作可能であってもよい。ユーザは、クライアントデバイスを用いることにより、クラウドインフラストラクチャシステム 2202 が提供するサービスを要求することなど、クラウドインフラストラクチャシステム 2202 とのやり取りを行い得る。

【0136】

いくつかの実施形態において、クラウドインフラストラクチャシステム 2202 が、管理関連サービスを提供するために実行する処理は、ビッグデータ解析を含み得る。この解析は、大きなデータセットを使用し、解析し、処理することにより、このデータ内のさまざまな傾向、挙動、関係などを検出し可視化することを含み得る。この解析は、1 つ以上のプロセッサが、場合によっては、データを並列に処理し、データを用いてシミュレーションを実行するなどして、実行してもよい。たとえば、自動化された態様で正規表現を決定するために、ビッグデータ解析がクラウドインフラストラクチャシステム 2202 によって実行されてもよい。この解析に使用されるデータは、構造化データ（たとえばデータベースに格納されたデータもしくは構造化モデルに従って構造化されたデータ）および／または非構造化データ（たとえばデータブロッブ（blob）（binary large object：バイナリ・ラージ・オブジェクト））を含み得る。

【0137】

図 22 の例に示されるように、クラウドインフラストラクチャシステム 2202 は、クラウドインフラストラクチャシステム 2202 が提供する各種クラウドサービスのプロビジョンを容易にするために利用されるインフラストラクチャリソース 2230 を含み得る。インフラストラクチャリソース 2230 は、たとえば、処理リソース、ストレージまたはメモリリソース、ネットワーキングリソースなどを含み得る。

【0138】

特定の実施形態において、異なる顧客に対しクラウドインフラストラクチャシステム 2202 が提供する各種クラウドサービスをサポートするためのこれらのリソースを効率的にプロビジョニングし易くするために、リソースを、リソースのセットまたはリソースモジュール（「ポッド」とも処される）にまとめてもよい。各リソースモジュールまたはポッドは、1 種類以上のリソースを予め一体化し最適化した組み合わせを含み得る。特定の実施形態において、異なるポッドを異なる種類のクラウドサービスに対して予めプロビジ

10

20

30

40

50

ョニングしてもよい。たとえば、第1のポッドセットをデータベースサービスのためにプロビジョニングしてもよく、第1のポッドセット内のポッドと異なるリソースの組み合わせを含み得る第2のポッドセットをJavaサービスなどのためにプロビジョニングしてもよい。いくつかのサービスについて、これらのサービスをプロビジョニングするために割り当てられたリソースをサービス間で共有してもよい。

【0139】

クラウドインフラストラクチャシステム2202自体が、クラウドインフラストラクチャシステム2202の異なるコンポーネントによって共有されるとともにクラウドインフラストラクチャシステム2202によるサービスのプロビジョニングを容易にするサービス2232を、内部で使用してもよい。これらの内部共有サービスは、セキュリティ・アイデンティティサービス、統合サービス、エンタープライズリポジトリサービス、エンタープライズマネージャサービス、ウィルススキャン・ホワイトリストサービス、高可用性、バックアップリカバリサービス、クラウドサポートを可能にするサービス、Eメールサービス、通知サービス、ファイル転送サービスなどを含み得るが、これらに限定されない。

【0140】

クラウドインフラストラクチャシステム2202は複数のサブシステムを含み得る。これらのサブシステムは、ソフトウェア、またはハードウェア、またはそれらの組み合わせで実現され得る。図22に示されるように、サブシステムは、クラウドインフラストラクチャシステム2202のユーザまたは顧客がクラウドインフラストラクチャシステム2202とやり取りすることを可能にするユーザインターフェイスサブシステム2212を含み得る。ユーザインターフェイスサブシステム2212は、ウェブインターフェイス2214、クラウドインフラストラクチャシステム2202が提供するクラウドサービスが宣伝広告され消費者による購入が可能なオンラインストアインターフェイス2216、およびその他のインターフェイス2218などの、各種異なるインターフェイスを含み得る。たとえば、顧客は、クライアントデバイスを用いて、クラウドインフラストラクチャシステム2202がインターフェイス2214、2216、および2218のうちの1つ以上を用いて提供する1つ以上のサービスを要求（サービス要求2234）してもよい。たとえば、顧客は、オンラインストアにアクセスし、クラウドインフラストラクチャシステム2202が提供するクラウドサービスをブラウズし、クラウドインフラストラクチャシステム2202が提供するとともに顧客が申し込むことを所望する1つ以上のサービスについてサブスクリプションオーダーを行い得る。このサービス要求は、顧客と、顧客が申し込むことを所望する1つ以上のサービスを識別する情報を含んでいてもよい。たとえば、顧客は、クラウドインフラストラクチャシステム2202によって提供される正規表現の自動生成関連サービスの申し込み注文を出すことができる。

【0141】

図22に示される例のような特定の実施形態において、クラウドインフラストラクチャシステム2202は、新しいオーダーを処理するように構成されたオーダー管理サブシステム（order management subsystem：OMS）2220を含み得る。この処理の一部として、OMS2220は、既に作成されていなければ顧客のアカウントを作成し、要求されたサービスを顧客に提供するために顧客に対して課金するのに使用する課金および／またはアカウント情報を顧客から受け、顧客情報を検証し、検証後、顧客のためにこのオーダーを予約し、各種ワークフローを調整することにより、プロビジョニングのためにオーダーを準備するように、構成されてもよい。

【0142】

適切に妥当性確認がなされると、OMS2220は、処理、メモリ、およびネットワークリソースを含む、このオーダーのためのリソースをプロビジョニングするように構成されたオーダープロビジョニングサブシステム（OPS）2224を呼び出し得る。プロビジョニングは、オーダーのためのリソースを割り当てることと、顧客オーダーが要求するサービスを容易にするようにリソースを構成することとを含み得る。オーダーのためにリソースをプロビジョニングするやり方およびプロビジョニングされるリソースのタイ

プは、顧客がオーダーしたクラウドサービスのタイプに依存し得る。たとえば、あるワークフローに従うと、O P S 2 2 2 4 を、要求されている特定のクラウドサービスを判断し、この特定のクラウドサービスのために予め構成されたであろうポッドの数を特定するように構成されてもよい。あるオーダーのために割り当てられるポッドの数は、要求されたサービスのサイズ／量／レベル／範囲に依存し得る。たとえば、割り当てるポッドの数は、サービスがサポートすべきユーザの数、サービスが要求されている期間などに基づいて決定してもよい。次に、割り当てられたポッドを、要求されたサービスを提供するために、要求している特定の顧客に合わせてカスタマイズしてもよい。

【0143】

クラウドインフラストラクチャシステム 2 2 0 2 は、要求されたサービスがいつ使用できようになるかを示すために、レスポンスまたは通知 2 2 4 4 を、要求している顧客に送ってもよい。いくつかの例において、顧客が、要求したサービスの利益の使用および利用を開始できるようにする情報（たとえばリンク）を顧客に送信してもよい。特定の実施形態では、正規表現の自動生成関連サービスを要求する顧客に対して、応答は、実行されるとユーザインターフェイスの表示を引き起こす命令を含み得る。

【0144】

クラウドインフラストラクチャシステム 2 2 0 2 はサービスを複数の顧客に提供し得る。各顧客ごとに、クラウドインフラストラクチャシステム 2 2 0 2 は、顧客から受けた 1 つ以上のサブスクリプションオーダーに関連する情報を管理し、オーダーに関連する顧客データを維持し、要求されたサービスを顧客に提供する役割を果たす。また、クラウドインフラストラクチャシステム 2 2 0 2 は、申し込まれたサービスの顧客による使用に関する使用統計を収集してもよい。たとえば、統計は、使用されたストレージの量、転送されたデータの量、ユーザの数、ならびにシステムアップタイムおよびシステムダウンタイムの量などについて、収集されてもよい。この使用情報を用いて顧客に課金してもよい。課金はたとえば月ごとに行ってもよい。

【0145】

クラウドインフラストラクチャシステム 2 2 0 2 は、サービスを複数の顧客に並列に提供してもよい。クラウドインフラストラクチャシステム 2 2 0 2 は、場合によっては著作権情報を含む、これらの顧客についての情報を格納してもよい。特定の実施形態において、クラウドインフラストラクチャシステム 2 2 0 2 は、顧客の情報を管理するとともに管理される情報を分離することで、ある顧客に関する情報が別の顧客に関する情報からアクセスされないようにするように構成された、アイデンティティ管理サブシステム（IMS）2 2 2 8 を含む。IMS 2 2 2 8 は、アイデンティティサービス、情報アクセス管理、認証および許可サービス、顧客のアイデンティティおよび役割ならびに関連する能力などを管理するためのサービスなどの、各種セキュリティ関連サービスを提供するように構成されてもよい。

【0146】

図 2 3 は、コンピュータシステム 2 3 0 0 の例を示す。いくつかの実施形態では、コンピュータシステム 2 3 0 0 は、上述のシステムのいずれかを実現するために用いられ得る。図 2 3 に示されるように、コンピュータシステム 2 3 0 0 は、バスサブシステム 2 3 0 2 を介して他のいくつかのサブシステムと通信する処理サブシステム 2 3 0 4 を含むさまざまなサブシステムを含む。これらの他のサブシステムは、処理加速ユニット 2 3 0 6、I/O サブシステム 2 3 0 8、ストレージサブシステム 2 3 1 8、および通信サブシステム 2 3 2 4 を含み得る。ストレージサブシステム 2 3 1 8 は、記憶媒体 2 3 2 2 およびシステムメモリ 2 3 1 0 を含む非一時的なコンピュータ読取り可能記憶媒体を含み得る。

【0147】

バスサブシステム 2 3 0 2 は、コンピュータシステム 2 3 0 0 のさまざまなコンポーネントおよびサブシステムに意図されるように互いに通信させるための機構を提供する。バスサブシステム 2 3 0 2 は単一のバスとして概略的に示されているが、バスサブシステムの代替例は複数のバスを利用してもよい。バスサブシステム 2 3 0 2 は、さまざまなバス

10

20

30

40

50

アーキテクチャのうちのいずれかを用いる、メモリバスまたはメモリコントローラ、周辺バス、ローカルバスなどを含むいくつかのタイプのバス構造のうちのいずれかであってもよい。たとえば、このようなアーキテクチャは、業界標準アーキテクチャ（Industry Standard Architecture：ISA）バス、マイクロチャネルアーキテクチャ（Micro Channel Architecture：MCA）バス、エンハンスドISA（Enhanced ISA：EISA）バス、ビデオ・エレクトロニクス・スタンダード・アソシエーション（Video Electronics Standards Association：VESA）ローカルバス、およびIEEE P1386.1規格に従って製造されるメザニンバスとして実現され得る周辺コンポーネントインターコネク（Peripheral Component Interconnect：PCI）バスなどを含み得る。

【0148】

処理サブシステム2304は、コンピュータシステム2300の動作を制御し、1つ以上のプロセッサ、特定用途向け集積回路（ASIC）、またはフィールドプログラマブルゲートアレイ（FPGA）を含み得る。プロセッサは、シングルコアまたはマルチコアプロセッサを含み得る。コンピュータシステム2300の処理リソースを、1つ以上の処理ユニット2332、2334などに組織することができる。処理ユニットは、1つ以上のプロセッサ、同一のまたは異なるプロセッサからの1つ以上のコア、コアとプロセッサとの組み合わせ、またはコアとプロセッサとのその他の組み合わせを含み得る。いくつかの実施形態において、処理サブシステム2304は、グラフィックスプロセッサ、デジタル信号プロセッサ（DSP）などのような1つ以上の専用コプロセッサを含み得る。いくつかの実施形態では、処理サブシステム2304の処理ユニットの一部または全部は、特定用途向け集積回路（ASIC）またはフィールドプログラマブルゲートアレイ（FPGA）などのカスタマイズされた回路を使用し得る。

【0149】

いくつかの実施形態において、処理サブシステム2304内の処理ユニットは、システムメモリ2310またはコンピュータ読取り可能記憶媒体2322に格納された命令を実行し得る。さまざまな例において、処理ユニットはさまざまなプログラムまたはコード命令を実行するとともに、同時に実行する複数のプログラムまたはプロセスを維持し得る。任意の所定の時点で、実行されるべきプログラムコードの一部または全部は、システムメモリ2310および／または潜在的に1つ以上の記憶装置を含むコンピュータ読取り可能記憶媒体2322に常駐していてもよい。適切なプログラミングを介して、処理サブシステム2304は、上述のさまざまな機能を提供し得る。コンピュータシステム2300が1つ以上の仮想マシンを実行している例において、1つ以上の処理ユニットが各仮想マシンに割り当ててもよい。

【0150】

特定の実施形態において、コンピュータシステム2300によって実行される全体的な処理を加速するように、カスタマイズされた処理を実行するために、または処理サブシステム2304によって実行される処理の一部をオフロードするために、処理加速ユニット2306を任意に設けることができる。

【0151】

I/Oサブシステム2308は、コンピュータシステム2300に情報を入力するための、および／またはコンピュータシステム2300から、もしくはコンピュータシステム2300を介して、情報を出力するための、デバイスおよび機構を含むことができる。一般に、「入力デバイス」という語の使用は、コンピュータシステム2300に情報を入力するためのすべての考えられ得るタイプのデバイスおよび機構を含むよう意図される。ユーザインターフェイス入力デバイスは、たとえば、キーボード、マウスまたはトラックボールなどのポインティングデバイス、ディスプレイに組み込まれたタッチパッドまたはタッチスクリーン、スクロールホイール、クリックホイール、ダイヤル、ボタン、スイッチ、キーパッド、音声コマンド認識システムを伴う音声入力デバイス、マイクロフォン、および他のタイプの入力デバイスを含んでもよい。ユーザインターフェイス入力デバイスは、ユーザが入力デバイスを制御しそれと対話することを可能にするMicrosoft

K i n e c t（登録商標）モーションセンサ、M i c r o s o f t X b o x（登録商標）360ゲームコントローラ、ジェスチャおよび音声コマンドを用いる入力を受信するためのインターフェイスを提供するデバイスなど、モーションセンシングおよび／またはジェスチャ認識デバイスも含んでもよい。ユーザインターフェイス入力デバイスは、ユーザから目の動き（たとえば、写真を撮っている間および／またはメニュー選択を行っている間の「まばたき」）を検出し、アイジェスチャを入力デバイス（たとえばG o o g l e G l a s s（登録商標））への入力として変換するG o o g l e G l a s s（登録商標）瞬き検出器などのアイジェスチャ認識デバイスも含んでもよい。また、ユーザインターフェイス入力デバイスは、ユーザが音声コマンドを介して音声認識システム（たとえばS i r i（登録商標）ナビゲータ）と対話することを可能にする音声認識感知デバイスを含んでもよい。

10

【0152】

ユーザインターフェイス入力デバイスの他の例は、三次元（3D）マウス、ジョイスティックまたはポインティングスティック、ゲームパッドおよびグラフィックタブレット、ならびにスピーカ、デジタルカメラ、デジタルカムコーダ、ポータブルメディアプレーヤ、ウェブカム、画像スキャナ、指紋スキャナ、バーコードリーダ3Dスキャナ、3Dプリンタ、レーザレンジファインダ、および視線追跡デバイスなどの聴覚／視覚デバイスも含んでもよいが、それらに限定されない。また、ユーザインターフェイス入力デバイスは、たとえば、コンピュータ断層撮影、磁気共鳴撮像、ポジションエミッショントモグラフィ、および医療用超音波検査デバイスなどの医療用画像化入力デバイスを含んでもよい。ユーザインターフェイス入力デバイスは、たとえば、M I D Iキーボード、デジタル楽器などの音声入力デバイスも含んでもよい。

20

【0153】

一般に、出力デバイスという語の使用は、コンピュータシステム2300からユーザまたは他のコンピュータに情報を出力するための考えられるすべてのタイプのデバイスおよび機構を含むことを意図している。ユーザインターフェイス出力デバイスは、ディスプレイサブシステム、インジケータライト、または音声出力デバイスなどのような非ビジュアルディスプレイなどを含んでもよい。ディスプレイサブシステムは、陰極線管（CRT）、液晶ディスプレイ（LCD）またはプラズマディスプレイを使うものなどのフラットパネルデバイス、計画デバイス、タッチスクリーンなどであってもよい。たとえば、ユーザインターフェイス出力デバイスは、モニタ、プリンタ、スピーカ、ヘッドフォン、自動車ナビゲーションシステム、プロッタ、音声出力デバイスおよびモデムなどの、テキスト、グラフィックスおよび音声／映像情報を視覚的に伝えるさまざまな表示デバイスを含んでもよいが、それらに限定されない。

30

【0154】

ストレージサブシステム2318は、コンピュータシステム2300によって使用される情報およびデータを格納するためのリポジトリまたはデータストアを提供する。ストレージサブシステム2318は、いくつかの例の機能を提供するための基本的なプログラミングおよびデータ構成を格納するための有形の非一時的なコンピュータ読取り可能記憶媒体を提供する。処理サブシステム2304によって実行されると上述の機能を提供するソフトウェア（たとえばプログラム、コードモジュール、命令）が、ストレージサブシステム2318に格納されてもよい。ソフトウェアは、処理サブシステム2304の1つ以上の処理ユニットによって実行されてもよい。ストレージサブシステム2318はまた、本開示の教示に従って使用されるデータを格納するためのリポジトリを提供してもよい。

40

【0155】

ストレージサブシステム2318は、揮発性および不揮発性メモリデバイスを含む1つ以上の非一時的メモリデバイスを含み得る。図23に示すように、ストレージサブシステム2318は、システムメモリ2310およびコンピュータ読取り可能記憶媒体2322を含む。システムメモリ2310は、プログラム実行中に命令およびデータを格納するための揮発性主ランダムアクセスメモリ（RAM）と、固定命令が格納される不揮発性読取

50

り専用メモリ（ROM）またはフラッシュメモリを含む、いくつかのメモリを含み得る。いくつかの実現例において、起動中などにコンピュータシステム2300内の要素間における情報の転送を助ける基本的なルーチンを含むベーシックインプット／アウトプットシステム（basic input/output system：BIOS）は、典型的には、ROMに格納されてもよい。典型的に、RAMは、処理サブシステム2304によって現在操作および実行されているデータおよび／またはプログラムモジュールを含む。いくつかの実現例において、システムメモリ2310は、スタティックランダムアクセスメモリ（SRAM）、ダイナミックランダムアクセスメモリ（DRAM）などのような複数の異なるタイプのメモリを含み得る。

【0156】

一例として、限定を伴うことなく、図23に示されるように、システムメモリ2310は、ウェブブラウザ、中間層アプリケーション、リレーショナルデータベース管理システム（RDBMS）などのような各種アプリケーションを含み得る、実行中のアプリケーションプログラム2312、プログラムデータ2314、およびオペレーティングシステム2316を、ロードしてもよい。一例として、オペレーティングシステム2316は、Microsoft Windows（登録商標）、Apple Macintosh（登録商標）および／またはLinuxオペレーティングシステム、市販されているさまざまなUNIX（登録商標）またはUNIX系オペレーティングシステム（さまざまなGNU／Linuxオペレーティングシステム、Google Chrome（登録商標）OSなどを含むがそれらに限定されない）、および／または、iOS（登録商標）、Windows（登録商標）Phone、Android（登録商標）OS、BlackBerry（登録商標）OS、Palm（登録商標）OSオペレーティングシステムのようなさまざまなバージョンのモバイルオペレーティングシステムなどを、含み得る。

【0157】

コンピュータ読取り可能記憶媒体2322は、いくつかの例の機能を提供するプログラミングおよびデータ構成を格納することができる。コンピュータ読取り可能記憶媒体2322は、コンピュータシステム2300のための、コンピュータ読取り可能命令、データ構造、プログラムモジュール、および他のデータのストレージを提供することができる。処理サブシステム2304によって実行されると上記機能を提供するソフトウェア（プログラム、コードモジュール、命令）は、ストレージサブシステム2318に格納されてもよい。一例として、コンピュータ読取り可能記憶媒体2322は、ハードディスクドライブ、磁気ディスクドライブ、CD ROM、DVD、Blu-Ray（登録商標）ディスクなどの光ディスクドライブ、またはその他の光学媒体のような不揮発性メモリを含み得る。コンピュータ読取り可能記憶媒体2322は、Zip（登録商標）ドライブ、フラッシュメモリカード、ユニバーサルシリアルバス（USB）フラッシュドライブ、セキュアデジタル（SD）カード、DVDディスク、デジタルビデオテープなどを含んでもよいが、それらに限定されない。コンピュータ読取り可能記憶媒体2322は、フラッシュメモリベースのSSD、エンタープライズフラッシュドライブ、ソリッドステートROMなどのような不揮発性メモリに基づくソリッドステートドライブ（SSD）、ソリッドステートRAM、ダイナミックRAM、スタティックRAMのような揮発性メモリに基づくSSD、DRAMベースのSSD、磁気抵抗RAM（MRAM）SSD、およびDRAMとフラッシュメモリベースのSSDとの組み合わせを使用するハイブリッドSSDも含み得る。

【0158】

特定の実施形態において、ストレージサブシステム2318は、コンピュータ読取り可能記憶媒体2322にさらに接続可能なコンピュータ読取り可能記憶媒体リーダ2320も含み得る。リーダ2320は、ディスク、フラッシュドライブなどのようなメモリデバイスからデータを受け、読取るように構成されてもよい。

【0159】

特定の実施形態において、コンピュータシステム2300は、処理およびメモリリソースの仮想化を含むがこれに限定されない仮想化技術をサポートし得る。たとえば、コンピ

10

20

30

40

50

ユータシステム 2300 は、1 つ以上の仮想マシンを実行するためのサポートを提供し得る。特定の実施形態において、コンピュータシステム 2300 は、仮想マシンの構成および管理を容易にするハイパーバイザなどのプログラムを実行し得る。各仮想マシンには、メモリ、演算（たとえばプロセッサ、コア）、I/O、およびネットワークリソースを割り当てられてもよい。各仮想マシンは通常、他の仮想マシンから独立して実行される。仮想マシンは、典型的には、コンピュータシステム 2300 によって実行される他の仮想マシンによって実行されるオペレーティングシステムと同じであり得るかまたは異なり得るそれ自体のオペレーティングシステムを実行する。したがって、潜在的に複数のオペレーティングシステムがコンピュータシステム 2300 によって同時に実行され得る。

【0160】

通信サブシステム 2324 は、他のコンピュータシステムおよびネットワークに対するインターフェイスを提供する。通信サブシステム 2324 は、他のシステムとコンピュータシステム 2300 との間のデータの送受のためのインターフェイスとして機能する。たとえば、通信サブシステム 2324 は、コンピュータシステム 2300 が、1 つ以上のクライアントデバイスとの間で情報を送受信するために、インターネットを介して 1 つ以上のクライアントデバイスへの通信チャネルを確立することを可能にし得る。

【0161】

通信サブシステム 2324 は、有線および／または無線通信プロトコルの両方をサポートし得る。ある実施形態において、通信サブシステム 2324 は、（たとえば、セルラー電話技術、3G、4G もしくは EDGE（グローバル進化のための高速データレート）などの先進データネットワーク技術、WiFi（IEEE 802.XX ファミリー規格、もしくは他のモバイル通信技術、またはそれらのいずれかの組み合わせを用いて）無線音声および／またはデータネットワークにアクセスするための無線周波数（RF）送受信機コンポーネント、グローバルポジショニングシステム（GPS）受信機コンポーネント、および／または他のコンポーネントを含み得る。いくつかの実施形態において、通信サブシステム 2324 は、無線インターフェイスに加えてまたはその代わりに、有線ネットワーク接続（たとえば Ethernet（登録商標））を提供し得る。

【0162】

通信サブシステム 2324 は、さまざまな形式でデータを受信および送信し得る。いくつかの実施形態において、通信サブシステム 2324 は、他の形式に加えて、構造化データフィードおよび／または非構造化データフィード 2326、イベントストリーム 2328、イベントアップデート 2330 などの形式で入力通信を受信してもよい。たとえば、通信サブシステム 2324 は、ソーシャルメディアネットワークおよび／または Twitter（登録商標）フィード、Facebook（登録商標）アップデート、Rich Site Summary（RSS）フィードなどのウェブフィード、および／または 1 つ以上の第三者情報源からのリアルタイムアップデートなどのような他の通信サービスのユーザから、リアルタイムでデータフィード 2326 を受信（または送信）するように構成されてもよい。

【0163】

特定の実施形態において、通信サブシステム 2324 は、連続データストリームの形式でデータを受信するように構成されてもよく、当該連続データストリームは、明確な終端を持たない、本来は連続的または無限であり得るリアルタイムイベントのイベントストリーム 2328 および／またはイベントアップデート 2330 を含んでもよい。連続データを生成するアプリケーションの例としては、たとえば、センサデータアプリケーション、金融株式相場表示板、ネットワーク性能測定ツール（たとえばネットワークモニタリングおよびトラフィック管理アプリケーション）、クリックストリーム解析ツール、自動車交通モニタリングなどを挙げることができる。

【0164】

通信サブシステム 2324 は、コンピュータシステム 2300 からのデータを他のコンピュータシステムまたはネットワークに伝えるように構成されてもよい。このデータは、

10

20

30

40

50

構造化および／または非構造化データフィールド 2 3 2 6、イベントストリーム 2 3 2 8、イベントアップデート 2 3 3 0 などのような各種異なる形式で、コンピュータシステム 2 3 0 0 に結合された 1 つ以上のストリーミングデータソースコンピュータと通信し得る 1 つ以上のデータベースに、伝えられてもよい。

【0165】

コンピュータシステム 2 3 0 0 は、ハンドヘルドポータブルデバイス（たとえば i P h o n e（登録商標）セルラーフォン、i P a d（登録商標）コンピューティングタブレット、P D A）、ウェアラブルデバイス（たとえば G o o g l e G l a s s（登録商標）ヘッドマウントディスプレイ）、パーソナルコンピュータ、ワークステーション、メインフレーム、キオスク、サーバラック、またはその他のデータ処理システムを含む、さまざまなタイプのうちの 1 つであればよい。コンピュータおよびネットワークの性質が常に変化しているため、図 2 3 に示されるコンピュータシステム 2 3 0 0 の記載は、具体的な例として意図されているに過ぎない。図 2 3 に示されるシステムよりも多くのコンポーネントまたは少ないコンポーネントを有するその他多くの構成が可能である。当業者であれば、本明細書における開示および教示に基づいて、さまざまな例を実現するための他の態様および／または方法を認識するだろう。

【0166】

特定の例について説明したが、さまざまな変形、変更、代替構成、および均等物が可能である。例は、特定のデータ処理環境内の動作に限定されず、複数のデータ処理環境内で自由に動作させることができる。さらに、例を特定の一連のトランザクションおよびステップを使用して説明したが、これが限定を意図しているのではないことは当業者には明らかであるはずである。いくつかのフローチャートは動作を逐次的プロセスとして説明しているが、これらの動作のうちの多くは並列または同時に実行されてもよい。加えて、動作の順序を再指定してもよい。プロセスは図に含まれない追加のステップを有し得る。上記の例の各種特徴および局面は、個別に使用されてもよく、またはともに使用されてもよい。

【0167】

さらに、特定の例をハードウェアとソフトウェアとの特定の組み合わせを用いて説明してきたが、ハードウェアとソフトウェアとの他の組み合わせも可能であることが理解されるはずである。特定の例は、ハードウェアでのみ、またはソフトウェアでのみ、またはそれらの組み合わせを用いて実現されてもよい。本明細書に記載されたさまざまなプロセスは、同じプロセッサまたは任意の組み合わせの異なるプロセッサ上で実現されてもよい。

【0168】

デバイス、システム、コンポーネントまたはモジュールが特定の動作または機能を実行するように構成されると記載されている場合、そのような構成は、たとえば、動作を実行するように電子回路を設計することにより、動作を実行するようにプログラミング可能な電子回路（マイクロプロセッサなど）をプログラミングすることにより、たとえば、非一時的なメモリ媒体に格納されたコードもしくは命令またはそれらの任意の組み合わせを実行するようにプログラミングされたコンピュータ命令もしくはコード、またはプロセッサもしくはコアを実行するなどにより、達成され得る。プロセスは、プロセス間通信のための従来の技術を含むがこれに限定されないさまざまな技術を使用して通信することができ、異なる対のプロセスは異なる技術を使用してもよく、同じ対のプロセスは異なる時間に異なる技術を使用してもよい。

【0169】

本開示では具体的な詳細を示すことにより例が十分に理解されるようにしている。しかしながら、例はこれらの具体的な詳細がなくとも実施し得るものである。たとえば、周知の回路、プロセス、アルゴリズム、構造、および技術は、例が曖昧にならないようにするために不必要な詳細事項なしで示している。本明細書は例示的な例のみを提供し、他の例の範囲、適用可能性、または構成を限定するよう意図されたものではない。むしろ、例の上記説明は、各種例を実現することを可能にする説明を当業者に提供する。要素の機能および構成の範囲内でさまざまな変更が可能である。

【0170】

したがって、明細書および図面は、限定的な意味ではなく例示的なものとみなされるべきである。しかしながら、請求項に記載されているより広範な精神および範囲から逸脱することなく、追加、削減、削除、ならびに他の修正および変更がこれらになされ得ることは明らかであろう。このように、具体的な例を説明してきたが、これらは限定を意図するものではない。さまざまな変形例および同等例は添付の特許請求の範囲内にある。

【0171】

上記の明細書では、本開示の局面についてその具体的な例を参照して説明しているが、本開示はそれに限定されるものではないということを当業者は認識するであろう。上記の開示のさまざまな特徴および局面は、個々にまたは一緒に用いられてもよい。さらに、例は、明細書のさらに広い精神および範囲から逸脱することなく、本明細書に記載されているものを超えて、さまざまな環境および用途で利用することができる。したがって、明細書および図面は、限定的ではなく例示的であると見なされるべきである。

10

【0172】

上記の説明では、例示の目的で、方法を特定の順序で記載した。代替の例では、方法は記載された順序とは異なる順序で実行されてもよいことを理解されたい。また、上記の方法は、ハードウェアコンポーネントによって実行されてもよいし、マシン実行可能命令であって、用いられると、そのような命令でプログラムされた汎用もしくは専用のプロセッサまたは論理回路などのマシンに方法を実行させてもよいマシン実行可能命令のシーケンスで具体化されてもよいことも理解されたい。これらのマシン実行可能命令は、CD-ROMもしくは他の種類の光ディスク、フロッピー（登録商標）ディスク、ROM、RAM、EPROM、EEPROM、磁気もしくは光学カード、フラッシュメモリのような、1つ以上の機械可読媒体、または電子命令を記憶するのに適した他の種類の機械可読媒体に保存できる。代替的に、これらの方法は、ハードウェアとソフトウェアとの組み合わせによって実行されてもよい。

20

【0173】

構成要素が特定の動作を実行するように構成されるとして記載されている場合、そのような構成は、たとえば、特定の動作を実行するよう電子回路もしくは他のハードウェアを設計すること、特定の動作を実行するようプログラミング可能な電子回路（たとえばマイクロプロセッサもしくは他の好適な電子回路）をプログラミングすること、またはそれらの任意の組み合わせによって達成されてもよい。

30

【0174】

本願の説明のための例をここに詳細に記載したが、本発明の概念は、他の態様で様々な具現化および採用され得ること、および特許請求の範囲は、先行技術によって制限される場合を除き、そのような変形を含むように解釈されるよう意図されることを理解されたい。

【0175】

構成要素が特定の動作を実行する「ように構成される」として記載されている場合、そのような構成は、たとえば、特定の動作を実行するよう電子回路もしくは他のハードウェアを設計すること、特定の動作を実行するようプログラミング可能な電子回路（たとえばマイクロプロセッサもしくは他の好適な電子回路）をプログラミングすること、またはそれらの任意の組み合わせによって達成されてもよい。

40

【図面】

【図 1】

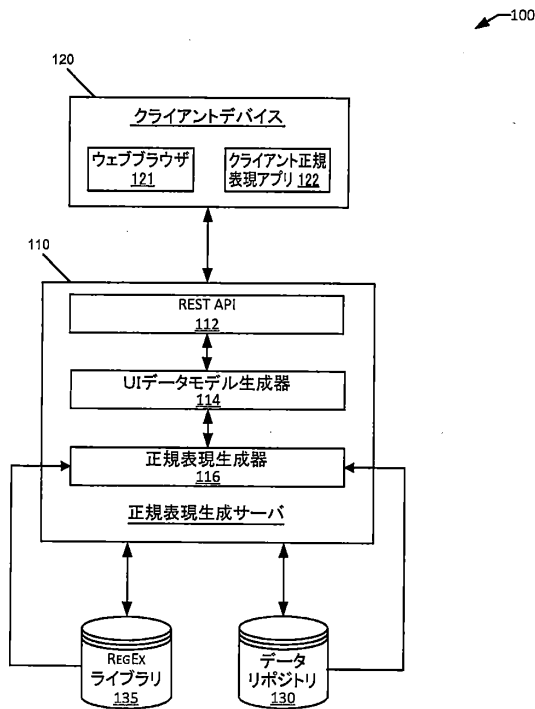


FIG. 1

【図 2】

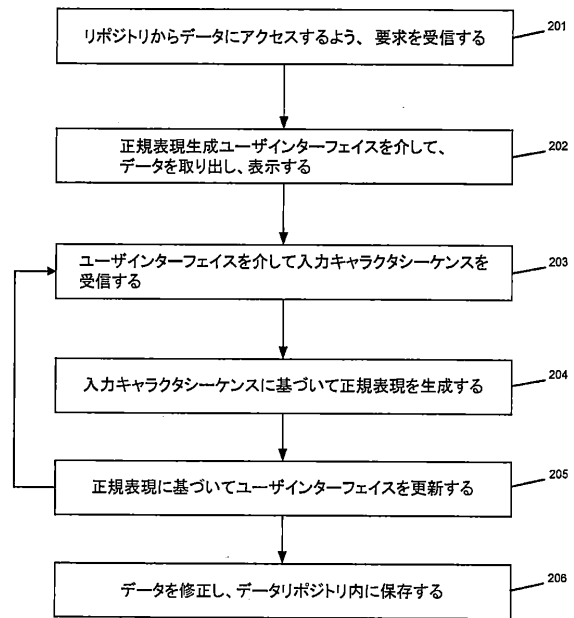


FIG. 2

【図 3】

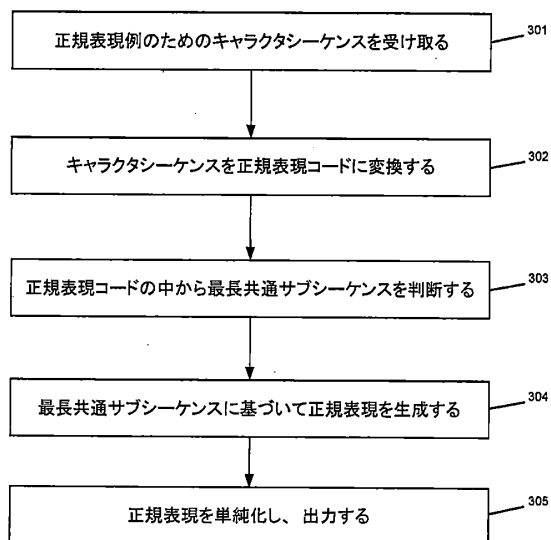


FIG. 3

【図 4】

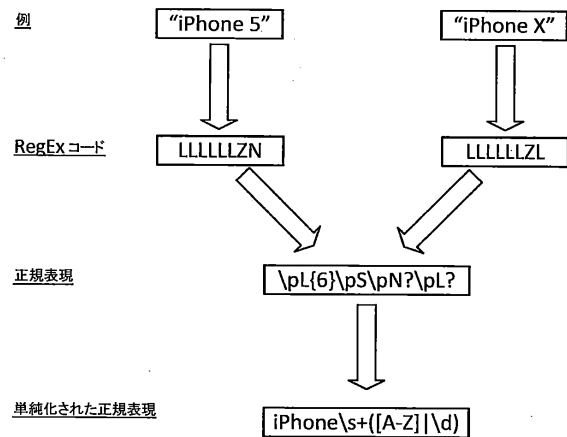


FIG. 4

10

20

30

40

50

【図 5】

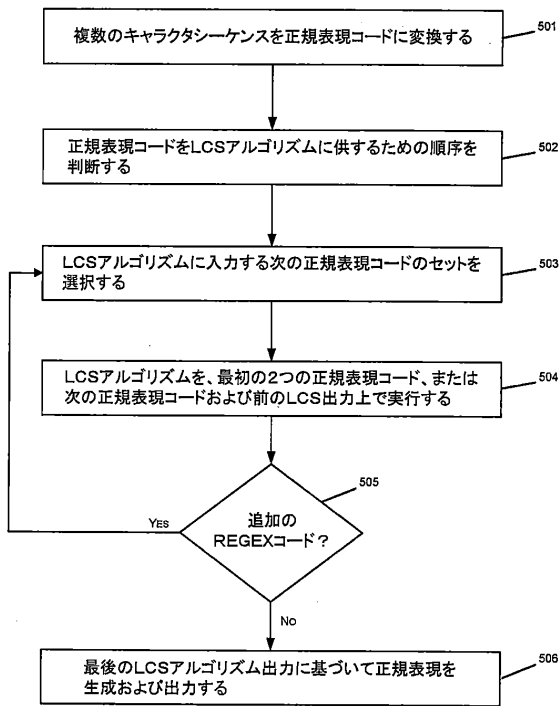


FIG. 5

【図 6】

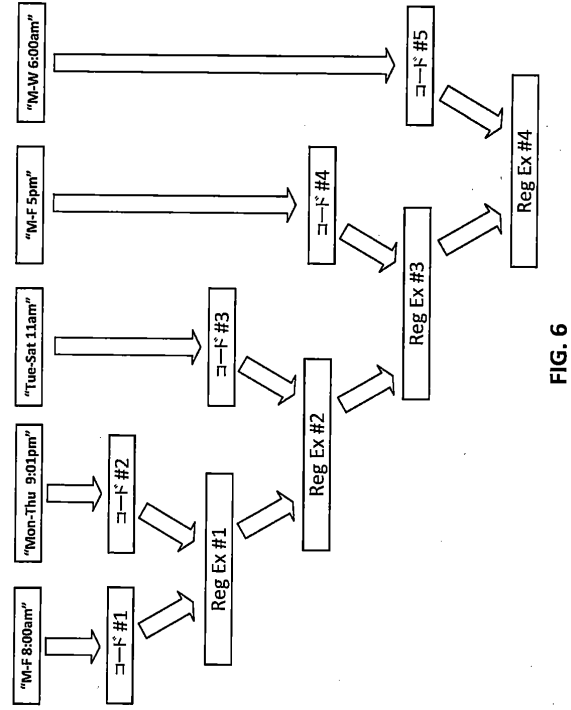


FIG. 6

【図 7】

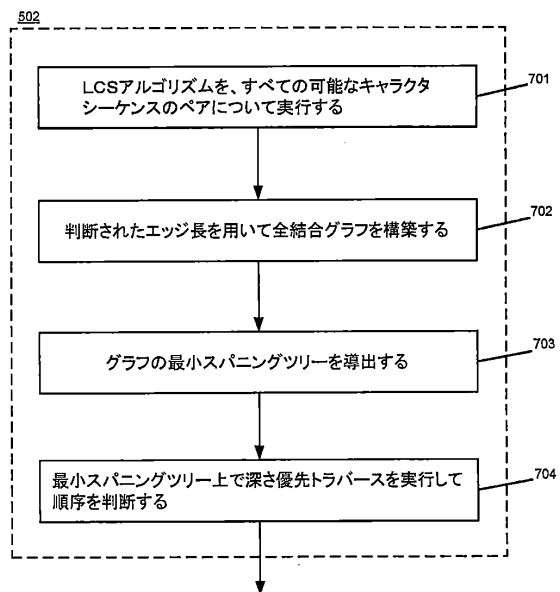


FIG. 7

【図 8 A】

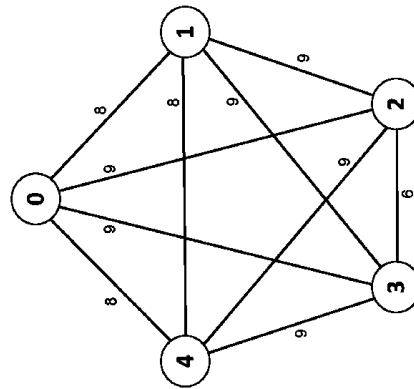


FIG. 8A

10

20

30

40

50

【図 8 B】

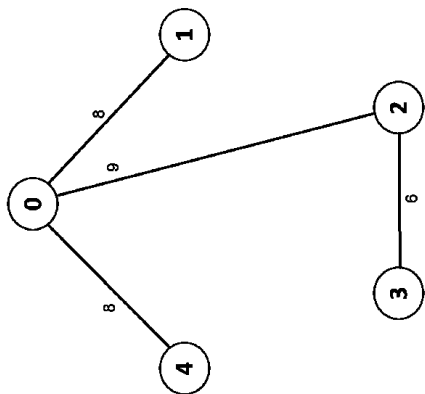


FIG. 8B

【図 9】

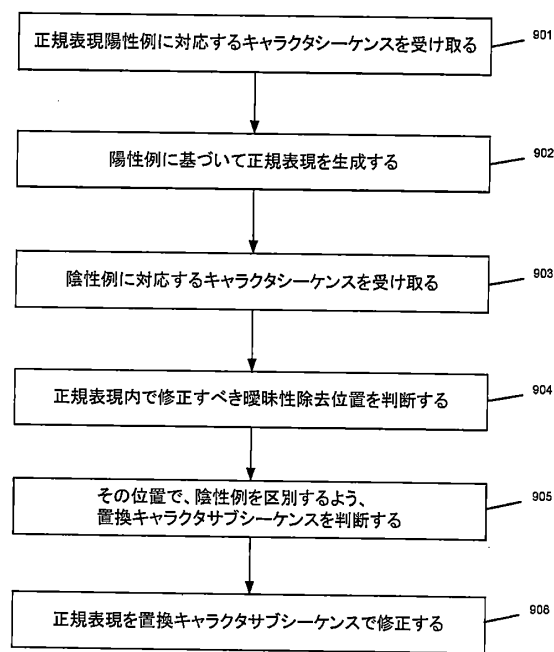


FIG. 9

【図 10 A】

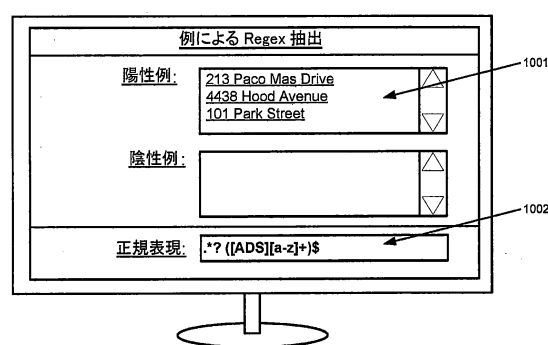


FIG. 10A

【図 10 B】

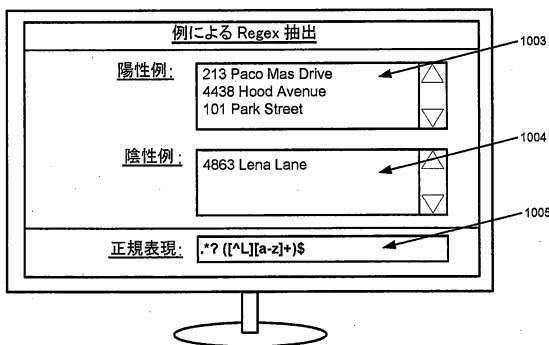


FIG. 10B

【図 1 1】

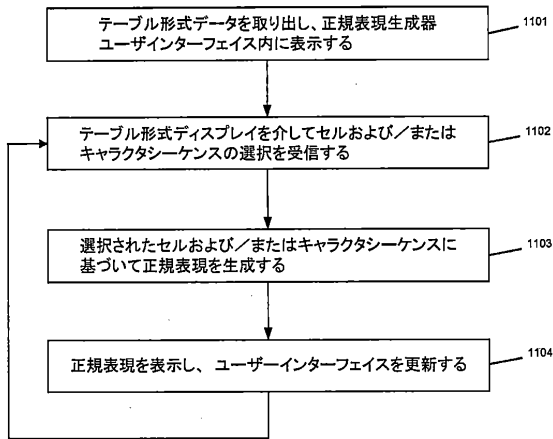


FIG. 11

【図 1 2】

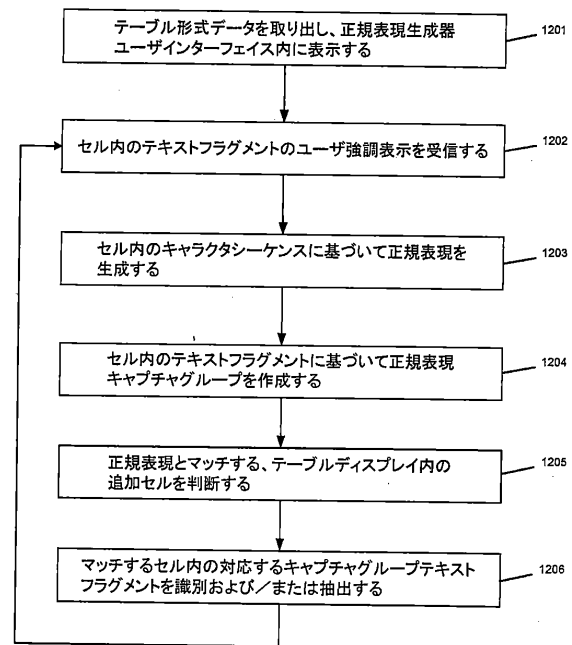


FIG. 12

【図 1 3】

電話 (919) 257-5648 918-915-8573 224-0539 (786) 201-858-7428 678-218-7856 (419) 675-9166

国	州	市	区	町	丁目	番地	郵便番号	電話番号	会社名
US	CA	San Jose	San Jose	San Jose	San Jose	San Jose	95128	(919) 257-5648	Apple Inc.
US	CA	San Jose	San Jose	San Jose	San Jose	San Jose	95128	918-915-8573	Apple Inc.
US	CA	San Jose	San Jose	San Jose	San Jose	San Jose	95128	224-0539	Apple Inc.
US	CA	San Jose	San Jose	San Jose	San Jose	San Jose	95128	(786) 201-858-7428	Apple Inc.
US	CA	San Jose	San Jose	San Jose	San Jose	San Jose	95128	678-218-7856	Apple Inc.
US	CA	San Jose	San Jose	San Jose	San Jose	San Jose	95128	(419) 675-9166	Apple Inc.

FIG. 13

【図 1 4】

例によるRegex抽出
指定された例
300-224-0539
指定された反例
300-224-0
生成されたRegex
(300-224-0539)

国	州	市	区	町	丁目	番地	郵便番号	電話番号	会社名
US	CA	San Jose	San Jose	San Jose	San Jose	San Jose	95128	(919) 257-5648	Apple Inc.
US	CA	San Jose	San Jose	San Jose	San Jose	San Jose	95128	918-915-8573	Apple Inc.
US	CA	San Jose	San Jose	San Jose	San Jose	San Jose	95128	224-0539	Apple Inc.
US	CA	San Jose	San Jose	San Jose	San Jose	San Jose	95128	(786) 201-858-7428	Apple Inc.
US	CA	San Jose	San Jose	San Jose	San Jose	San Jose	95128	678-218-7856	Apple Inc.
US	CA	San Jose	San Jose	San Jose	San Jose	San Jose	95128	(419) 675-9166	Apple Inc.

FIG. 14

【図 15】

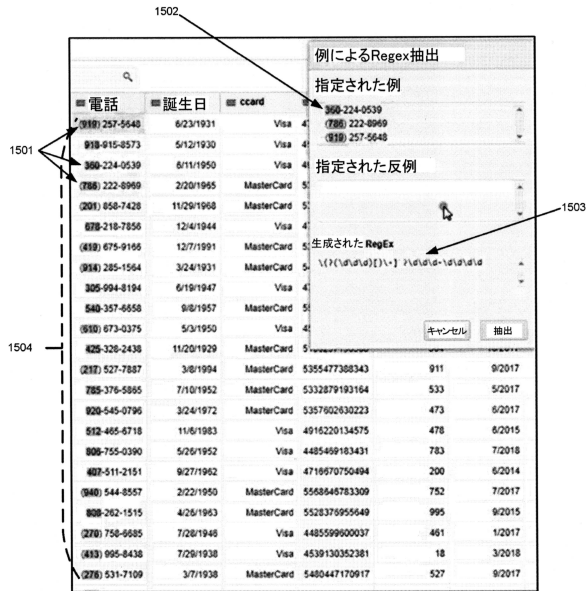


FIG. 15

【図 16 A】

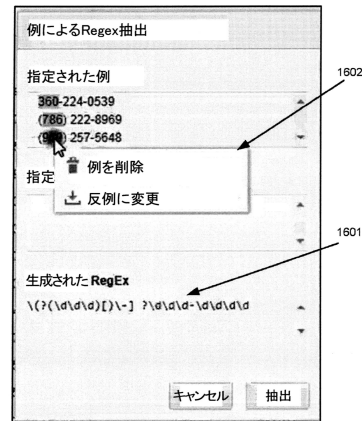


FIG. 16A

【図 16 B】

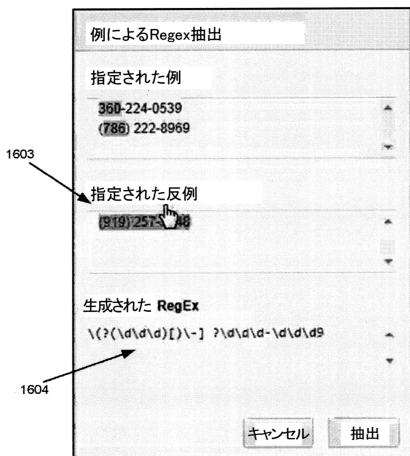


FIG. 16B

【圖 17】

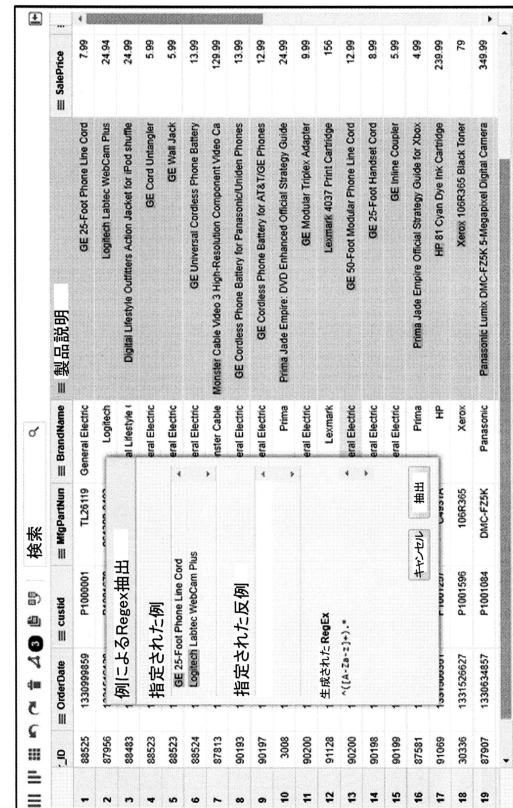


FIG. 17

【図 18】

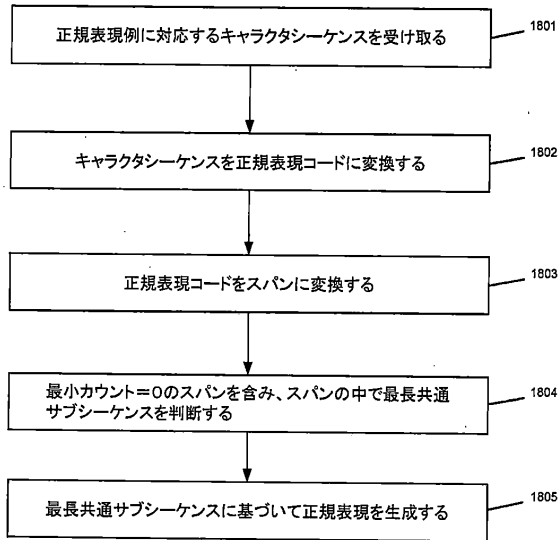


FIG. 18

【図 19】

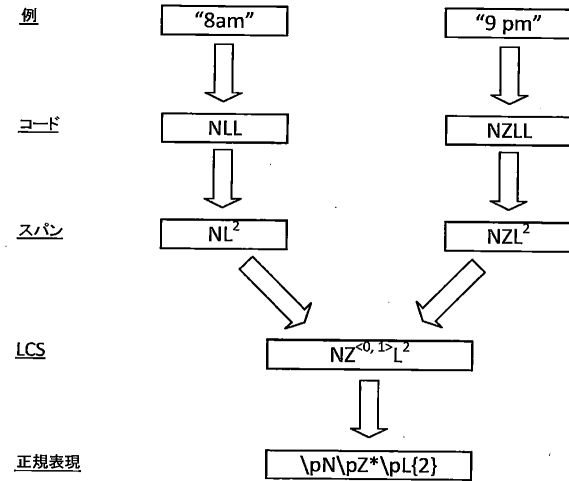


FIG. 19

【図 20】

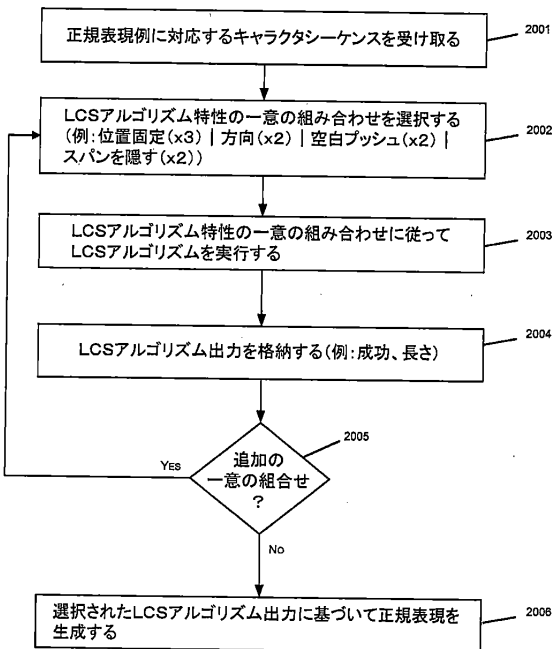


FIG. 20

【図 21】

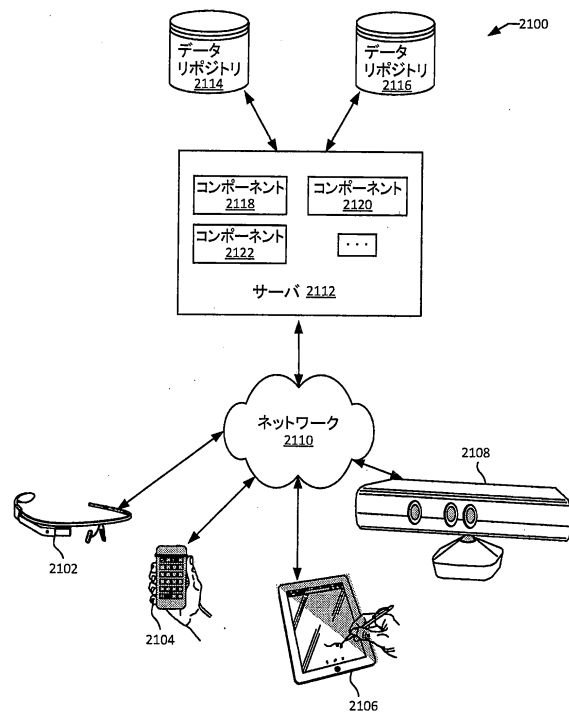


FIG. 21

【図 2 2】

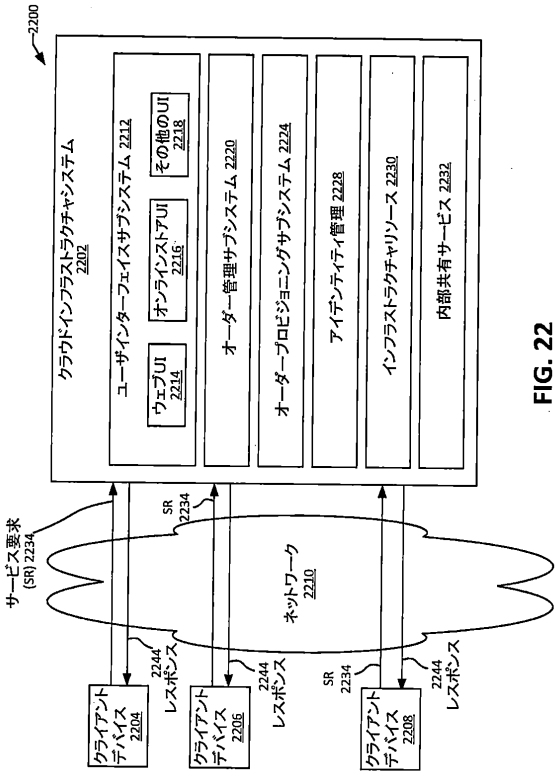


FIG. 22

【図 2 3】

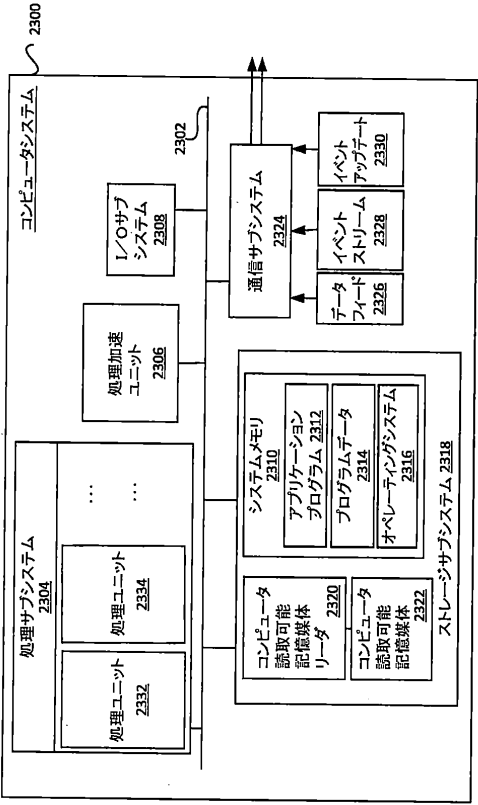


FIG. 23

10

20

30

40

50

フロントページの続き

(33)優先権主張国・地域又は機関

米国(US)

(31)優先権主張番号 16/438,327

(32)優先日 令和1年6月11日(2019.6.11)

(33)優先権主張国・地域又は機関

米国(US)

・プレイス、8717

(72)発明者 クライダー、マーク・エル

アメリカ合衆国、80007 コロラド州、アーバダ、ウェスト・セブンティエイス・ドライブ、
17868

審査官 和田 財太

(56)参考文献 特開2017-204815(JP, A)

特開2009-015395(JP, A)

特開2019-004402(JP, A)

(58)調査した分野 (Int.Cl., DB名)

G06F 16/00

G06F 3/00