

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6107495号
(P6107495)

(45) 発行日 平成29年4月5日 (2017.4.5)

(24) 登録日 平成29年3月17日 (2017.3.17)

(51) Int.Cl. F I
G06F 17/30 (2006.01)
 G06F 17/30 220Z
 G06F 17/30 170Z

請求項の数 5 (全 30 頁)

(21) 出願番号	特願2013-147955 (P2013-147955)	(73) 特許権者	000005223
(22) 出願日	平成25年7月16日 (2013.7.16)		富士通株式会社
(65) 公開番号	特開2015-22370 (P2015-22370A)		神奈川県川崎市中原区上小田中4丁目1番1号
(43) 公開日	平成27年2月2日 (2015.2.2)	(74) 代理人	100089118
審査請求日	平成28年4月5日 (2016.4.5)		弁理士 酒井 宏明
		(72) 発明者	多湖 真一郎
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72) 発明者	河東 孝
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72) 発明者	稲越 宏弥
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

最終頁に続く

(54) 【発明の名称】 照合方法および照合プログラム

(57) 【特許請求の範囲】

【請求項 1】

コンピュータが実行する照合方法であって、

第1ウィンドウにイベント条件が対応付けられ、前記第1ウィンドウに所定のウィンドウ幅条件が設定され、複数の前記第1ウィンドウが所定のインターバル条件によって直列に接続されたクエリに基づいて、前記ウィンドウ幅条件よりも大きいウィンドウ幅の第2ウィンドウを、イベントと前記イベントの出現時刻とを対応付けた照合データに設定し、

前記第2ウィンドウに含まれる前記照合データのイベントおよび各第2ウィンドウのインターバルが、前記第1ウィンドウのイベント条件および前記インターバル条件を満たすか否かを判定する

各処理を実行することを特徴とする照合方法。

【請求項 2】

前記第2ウィンドウを前記照合データに設定する処理は、前記第2ウィンドウの開始時刻を起点として、前記第2ウィンドウの幅を2倍した時刻よりも後に出現する初めのイベントの時刻から、前記第2ウィンドウの幅だけ前の時刻に戻り、戻った時刻の次の時刻よりも後に出現する初めのイベントの時刻を開始時刻とする次の第2ウィンドウを設定することを特徴とする請求項1に記載の照合方法。

【請求項 3】

前記インターバル条件は最小インターバル条件を含み、前記判定する処理は、先行の第2ウィンドウに含まれるイベントのうち、前記イベント条件を満たす最初のイベントの組

みの中で一番遅いイベントの第1時刻と、後続の第2ウィンドウに含まれるイベントのうち、前記イベント条件を満たす最後のイベントの組み合わせの中で一番早いイベントの第2時刻とを特定し、前記第1時刻と前記第2時刻とのインターバルと、前記最小インターバル条件とを基にして、先行の第2ウィンドウと後続の第2ウィンドウとが、前記インターバル条件を満たすか否かを判定することを特徴とする請求項2に記載の照合方法。

【請求項4】

前記インターバル条件は最大インターバル条件を含み、前記判定する処理は、先行の第2ウィンドウに含まれるイベントのうち、前記イベント条件を満たす最後のイベントの組み合わせの中で一番早いイベントの第3時刻と、後続の第2ウィンドウに含まれるイベントのうち、前記イベント条件を満たす最初のイベントの組み合わせの中で一番遅いイベントの第4時刻とを特定し、前記第3時刻と前記第4時刻とのインターバルと、前記最大インターバル条件と、先行の第2ウィンドウの幅と、後続の第2ウィンドウの幅とを基にして、先行の第2ウィンドウと後続の第2ウィンドウとが、前記インターバル条件を満たすか否かを判定することを特徴とする請求項2に記載の照合方法。

10

【請求項5】

コンピュータに、

第1ウィンドウにイベント条件が対応付けられ、前記第1ウィンドウに所定のウィンドウ幅条件が設定され、複数の前記第1ウィンドウが所定のインターバル条件によって直列に接続されたクエリに基づいて、前記ウィンドウ幅条件よりも大きいウィンドウ幅の第2ウィンドウを、イベントと前記イベントの出現時刻とを対応付けた照合データに設定し、

20

前記第2ウィンドウに含まれる前記照合データのイベントおよび各第2ウィンドウのインターバルが、前記第1ウィンドウのイベント条件および前記インターバル条件を満たすか否かを判定する

各処理を実行させる照合プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、照合方法等に関する。

【背景技術】

【0002】

30

規則性を持つ複数のイベントによる事象を分析、検知し、リアクションすることで、ビジネス活動を効率化していくことが重要である。イベントの検知に関し、従来技術では、クエリを定義し、クエリと照合データとをマッチングすることで、所定の規則性を持つイベントの組み合わせを照合データから検出する。クエリは、イベントを一意に識別するイベント条件と、イベント条件間の間隔を示すインターバル条件とによって定義される。

【0003】

図29は、従来技術を説明するためのクエリの一例を示す図(1)である。クエリ10は、クーポンを発行してから3日以内に商品を購入した事象を検出するクエリの一例である。図29に示す例では、クエリ10は、ノード11、12を有し、ノード11はイベント条件「クーポン発行」を示し、ノード12は、イベント条件「商品購入」を示す。また、ノード11、12間のインターバル条件は「0-3日」となる。

40

【0004】

図30は、従来技術を説明するための照合データの一例を示す図(1)である。図30の照合データ20は、イベントが発生した日付と、該当するイベントとが対応付けられる。図30の横軸は、日付を表す。例えば、図30に示す例では、日付「1/2、1/5、1/12」に、イベント「クーポン発行」が発生した旨が示される。

【0005】

従来技術により、クエリ10と照合データ20とをマッチングすると、日付「1/5」のイベント「クーポン発行」と、日付「1/6」のイベント「商品購入」との組み合わせがヒットする。また、イベント「クーポン発行」とイベント「商品購入」との間隔が「0」であ

50

り、インターバル条件「0-3日」を満たすためである。このため、従来技術は、日付「1/5」のイベント「クーポン発行」と、日付「1/6」のイベント「商品購入」との組みを検出結果として出力する。

【0006】

続いて、ウィンドウ型直列パターンのクエリについて説明する。ウィンドウ型直列パターンのクエリは、イベントの出現順序を規定しないウィンドウを直列に接続したものである。図31は、ウィンドウ型直列パターンのクエリの一例を示す図である。図31に示すクエリ30は、ウィンドウ31およびウィンドウ32が定義され、ウィンドウ31およびウィンドウ32が直列に接続される。ウィンドウ31とウィンドウ32との間隔は、インターバル条件「5-8分」が設定されている。

10

【0007】

ウィンドウ31は、イベント条件31a、31b、31cを有する。イベント条件31aは、イベント「商品を手にする」が出現するという条件である。イベント条件31bは、イベント「店員が声をかける」が出現するという条件である。イベント条件31cは、「タイムセールを行う」が出現するという条件である。

【0008】

ウィンドウ31のウィンドウ幅条件は、4分である。このため、ある開始時刻から4分後までの時間帯に、イベント条件31a、31b、31cを満たすイベントが出現する場合には、ウィンドウ31の条件を満たすことになる。

【0009】

20

ウィンドウ32は、イベント条件32a、32bを有する。イベント条件32aは、イベント「商品を購入する」が出現するという条件である。イベント条件32bは、イベント「店員が値下げする」が出現するという条件である。

【0010】

ウィンドウ32のウィンドウ幅条件は、3分である。このため、ある開始時刻から3分後までの時間帯に、イベント条件32a、32bを満たすイベントが出現する場合には、ウィンドウ32の条件を満たすことになる。

【0011】

従来技術は、照合データ上に、ウィンドウを設定し、ウィンドウをずらしながら、ウィンドウ31、32の条件を満たすイベントの組みを照合データから検出する。そして、従来技術は、ウィンドウ31の条件を満たす時間帯と、ウィンドウ32の条件を満たす時間帯との間隔が、インターバル条件を満たす場合に、ウィンドウ31、32に含まれる照合データのイベントの組みを、クエリの検索結果として出力する。

30

【0012】

図32は、従来技術の手法の一例を示す図である。ここでは、図31に示したクエリ30の条件を満たす照合データ40上のイベントの組みを検出する場合について説明する。図31の照合データ40は、イベントが発生した時刻と、該当するイベントを対応付ける。照合データ40の横軸は時刻を示す。また便宜上、照合データ40では、イベント「商品を手にする」を「A」、「店員が声をかける」を「B」、「タイムセールを行う」を「C」、「商品を購入する」を「D」、「店員が値下げする」を「E」と表す。

40

【0013】

従来技術は、照合データ40上に、クエリ30のウィンドウ31の条件に対応するウィンドウ50と、ウィンドウ32の条件に対応するウィンドウ60を設定する。従来技術は、ウィンドウ50、60を1単位時間ずつずらしながら、クエリ30のウィンドウ31、32の条件を満たすか否かを判定する。

【0014】

ウィンドウ50の開始時刻がそれぞれ1, 2, 7, 8, 9となった時点で、クエリ30のウィンドウ31の条件を満たす。ウィンドウ50の開始時刻が1, 2, 7, 8, 9となった時点のウィンドウをそれぞれ、50a, 50b, 50c, 50d, 50eと表記する。また、ウィンドウ60の開始時刻が13, 14, 17, 18となった時点で、クエリ3

50

0のウィンドウ32の条件を満たす。ウィンドウ60の開始時刻が13、14、17、18となった時点のウィンドウをそれぞれ、60a、60b、60c、60dと表記する。

【0015】

従来技術は、ウィンドウ50a～50eと、ウィンドウ60a～60との組み合わせについてそれぞれクエリ30のインターバル条件を満たすか否かを判定し、インターバル条件を満たすウィンドウの組みに含まれるイベントを検出する。検出されたイベントの組みは、クエリ30の条件を満たす。図32に示す例では、ウィンドウ50a、60aの組み、ウィンドウ50b、60bの組みがクエリ30の条件を満たす。また、ウィンドウ50c、60cの組み、ウィンドウ50d、60cの組み、ウィンドウ50d、60dの組み、ウィンドウ50e、60dの組みがクエリ30の条件を満たす。

10

【先行技術文献】

【特許文献】

【0016】

【特許文献1】特開2004-46601号公報

【特許文献2】特開2009-134689号公報

【特許文献3】特開2010-108073号公報

【特許文献4】特開2013-8329号公報

【発明の概要】

【発明が解決しようとする課題】

【0017】

20

しかしながら、上述した従来技術では、クエリと照合データとのマッチングを高速に行うことができないという問題がある。

【0018】

例えば、従来技術では、クエリのウィンドウの条件を満たすウィンドウの数が多くなり、これによって、各ウィンドウの組み合わせの量も大きくなる。そして、従来技術では、各組み合わせについて、クエリのインターバル条件を満たすか否かを判定するため、マッチングを高速に行うことができない。

【0019】

1つの側面では、クエリと照合データとのマッチングを高速に行うことができる照合方法および照合プログラムを提供することを目的とする。

30

【課題を解決するための手段】

【0020】

第1の案では、コンピュータが、クエリに基づいて、ウィンドウ幅条件よりも大きいウィンドウ幅の第2ウィンドウを、イベントとイベントの出現時刻とを対応付けた照合データに設定する。また、コンピュータが、第2ウィンドウに含まれる照合データのイベントおよび各第2ウィンドウのインターバルが、第1ウィンドウのイベント条件およびインターバル条件を満たすか否かを判定する。

【発明の効果】

【0021】

本発明の1実施形態によれば、クエリと照合データとのマッチングを高速に行うことができるという効果を奏する。

40

【図面の簡単な説明】

【0022】

【図1】図1は、本実施例1に係る照合装置の構成を示す機能ブロック図である。

【図2】図2は、本実施例1に係るクエリのデータ構造の一例を示す図である。

【図3】図3は、本実施例1に係る照合データのデータ構造の一例を示す図である。

【図4】図4は、照合データのデータイメージを示す図である。

【図5】図5は、本実施例1に係る仮ウィンドウ構築処理の処理手順を示すフローチャートである。

【図6】図6は、本実施例1に係る仮ウィンドウ構築処理を説明するための図(1)であ

50

る。

【図 7】図 7 は、本実施例 1 に係る仮ウィンドウ構築処理を説明するための図 (2) である。

【図 8】図 8 は、本実施例 1 に係る仮ウィンドウ構築処理を説明するための図 (3) である。

【図 9】図 9 は、本実施例 1 に係る仮ウィンドウ構築処理を説明するための図 (4) である。

【図 10】図 10 は、本実施例 1 に係る仮ウィンドウ構築処理の処理結果の一例を示す図である。

【図 11】図 11 は、本実施例 1 に係る照合装置の処理手順を示すフローチャートである 10

。【図 12】図 12 は、本実施例 2 に係る照合装置の構成を示す機能ブロック図である。

【図 13】図 13 は、本実施例 2 に係るウィンドウ処理部の処理の一例を説明するための図である。

【図 14】図 14 は、本実施例 2 に係る仮ウィンドウ構築処理の処理手順を示すフローチャートである。

【図 15】図 15 は、本実施例 2 に係る仮ウィンドウ構築処理を説明するための図 (1) である。

【図 16】図 16 は、本実施例 2 に係る仮ウィンドウ構築処理を説明するための図 (2) である。 20

【図 17】図 17 は、本実施例 2 に係る仮ウィンドウ構築処理を説明するための図 (3) である。

【図 18】図 18 は、最小インターバル条件を説明するための図である。

【図 19】図 19 は、最大インターバル条件を説明するための図である。

【図 20】図 20 は、インターバル条件判定処理の処理手順を示すフローチャート (1) である。

【図 21】図 21 は、インターバル条件判定処理の処理手順を示すフローチャート (2) である。

【図 22】図 22 は、本実施例 2 に係るインターバル条件判定処理を説明するための図 (1) である。 30

【図 23】図 23 は、本実施例 2 に係るインターバル条件判定処理を説明するための図 (2) である。

【図 24】図 24 は、本実施例 2 に係るインターバル条件判定処理を説明するための図 (3) である。

【図 25】図 25 は、本実施例 2 に係るインターバル条件判定処理を説明するための図 (4) である。

【図 26】図 26 は、組み合わせ展開の結果の一例を示す図である。

【図 27】図 27 は、本実施例 2 に係る照合装置の処理手順を示すフローチャートである。

【図 28】図 28 は、照合プログラムを実行するコンピュータの一例を示す図である。 40

【図 29】図 29 は、従来技術を説明するためのクエリの一例を示す図 (1) である。

【図 30】図 30 は、従来技術を説明するための照合データの一例を示す図 (1) である。

【図 31】図 31 は、ウィンドウ型直列パターンのクエリの一例を示す図である。

【図 32】図 32 は、従来技術の手法の一例を示す図である。

【発明を実施するための形態】

【0023】

以下に、本願の開示する照合方法および照合プログラムの実施例を図面に基づいて詳細に説明する。なお、この実施例によりこの発明が限定されるものではない。

【実施例 1】

【0024】

本実施例1に係る照合装置の構成の一例について説明する。図1は、本実施例1に係る照合装置の構成を示す機能ブロック図である。図1に示すように、この照合装置100は、通信部110と、入力部120と、表示部130と、記憶部140と、制御部150とを有する。

【0025】

通信部110は、ネットワークを介して他の外部装置と通信を実行する処理部である。通信部110は、通信装置に対応する。

【0026】

入力部120は、照合装置100に入力するための入力装置である。例えば、入力部120は、キーボードやマウス、タッチパネル等に対応する。 10

【0027】

表示部130は、制御部150から出力される情報を表示する表示装置である。表示部130は、液晶モニターやタッチパネル等に対応する。

【0028】

記憶部140は、クエリ141および照合データ142を有する。記憶部140は、RAM(Random Access Memory)、ROM(Read Only Memory)、フラッシュメモリ(Flash Memory)などの半導体メモリ素子などの記憶装置に対応する。

【0029】

クエリ141は、ウィンドウ型直列パターンのクエリである。ウィンドウ型直列パターンのクエリは、イベントの出現順序を規定しないウィンドウを直列に接続したものである。図2は、本実施例1に係るクエリのデータ構造の一例を示す図である。図2に示すように、クエリ141は、ウィンドウ141a、141bが定義され、ウィンドウ141aおよびウィンドウ141bが直列に接続される。ウィンドウ141aとウィンドウ141bとのインターバル条件を「5-8分」とする。 20

【0030】

ウィンドウ141aは、イベント「A」が出現するというイベント条件と、イベント「B」が出現するというイベント条件と、イベント「C」が出現するというイベント条件とを含む。また、ウィンドウ141aのウィンドウ幅条件は、4分である。このため、ある開始時刻から4分後までの時間帯に、順不同で、イベントA、B、Cが出現する場合には、ウィンドウ141aの条件を満たすことになる。 30

【0031】

ウィンドウ141bは、イベント「D」が出現するというイベント条件と、イベント「E」が出現するというイベント条件とを含む。また、ウィンドウ141bのウィンドウ幅条件は、3分である。このため、ある開始時刻から3分後までの時間帯に、順不同で、イベントD、Eが出現する場合には、ウィンドウ141bの条件を満たすことになる。

【0032】

照合データ142は、クエリ141の照合対象となる情報であり、イベントの出現時刻の情報を含む。図3は、本実施例1に係る照合データのデータ構造の一例を示す図である。図3に示すように、照合データ142は、時刻と、顧客IDと、場所と、イベントとをそれぞれ対応付ける。時刻は、イベントが出現した時刻を示す情報である。顧客IDは、イベントに関連する顧客を一意に識別する情報である。場所は、イベントが出現した場所を一意に識別する情報である。イベントは、イベントの種別を示す。例えば、図3の時刻「0」分において、場所「3F」でイベント「A」が出現し、顧客ID「432」の顧客に関連する旨のレコードが登録される。 40

【0033】

なお、図3の照合データ142のデータイメージは、図4に示すものとなる。図4は、照合データのデータイメージを示す図である。後述する説明では、図4に示すようなデータイメージを利用して、適宜説明を行う。図4の横軸は時刻に対応する。図4において、例えば、時刻「0」において、イベント「A」が出現している。 50

【0034】

図1の説明に戻る。制御部150は、ウィンドウ処理部151および照合部152を有する。制御部150は、例えば、ASIC (Application Specific Integrated Circuit) や、FPGA (Field Programmable Gate Array) などの集積装置に対応する。また、制御部150は、例えば、CPU (Central Processing Unit) やMPU (Micro Processing Unit) 等の電子回路に対応する。

【0035】

ウィンドウ処理部151は、クエリ141を参照し、クエリ141のウィンドウのウィンドウ幅条件よりも大きい幅のウィンドウを、照合データ142上に設定する処理部である。例えば、ウィンドウ処理部151は、式(1)に基づく時間幅のウィンドウを、照合データ142に設定する。ウィンドウ処理部151は、照合データ142上に設定したウィンドウが、クエリ141のウィンドウのイベント条件を満たす場合には、設定したウィンドウを仮ウィンドウとする。

10

【0036】

ウィンドウ処理部151が、仮ウィンドウを構築する処理手順について説明する。図5は、本実施例1に係る仮ウィンドウ構築処理の処理手順を示すフローチャートである。図5に示すように、ウィンドウ処理部151は、照合データ142の時間を各ウィンドウ幅条件の長さで分割し、分割領域に対して時刻順に番号を付与する(ステップS101)。

【0037】

ウィンドウ処理部151は、連続する2つの分割領域から最後の時刻を除いた期間を、ウィンドウに対する予備ウィンドウに設定する(ステップS102)。ウィンドウ処理部151は、1つのイベントを検知し(ステップS103)、ウィンドウのイベント条件で振り分ける(ステップS104)。

20

【0038】

ウィンドウ処理部151は、イベントの時刻をウィンドウ幅条件で割った値を分割領域番号とし、分割領域番号に一致する予備ウィンドウにイベントを追加する(ステップS105)。ただし、ステップS105において、ウィンドウ処理部151は、ステップS102で取り除いた最後の時刻と、イベントの時刻が一致する場合には、イベントを追加しない。

【0039】

ウィンドウ処理部151は、予備ウィンドウがウィンドウのイベント条件を満たすかを判定する(ステップS106)。ウィンドウ処理部151は、予備ウィンドウがウィンドウのイベント条件を満たさない場合には(ステップS106, No)、ステップS108に移行する。

30

【0040】

一方、ウィンドウ処理部151は、予備ウィンドウがウィンドウのイベント条件を満たす場合には(ステップS106, Yes)、予備ウィンドウを仮ウィンドウに設定する(ステップS107)。

【0041】

ウィンドウ処理部151は、処理を終了する場合には(ステップS108, Yes)、ウィンドウを設定する処理を終了する。一方、ウィンドウ処理部151は、処理を終了しない場合には(ステップS108, No)、ステップS103に移行する。

40

【0042】

ここで、図2に示したクエリ141と、図4に示した照合データ142とを例にして、ウィンドウ処理部151が実行する仮ウィンドウ構築処理について説明する。図6～図9は、本実施例1に係る仮ウィンドウ構築処理を説明するための図である。

【0043】

図6について、ウィンドウ処理部151は、クエリ141のウィンドウ141aのウィンドウ幅条件「4」によって照合データ142の時間を分割し、分割領域70a, 70b, 70c, 70d, 70eを設定する。ウィンドウ処理部151は、分割領域70a～7

50

0 eに対して、分割領域番号0～4をそれぞれ設定する。ウィンドウ処理部151は、クエリ141のウィンドウ141bのウィンドウ幅条件「3」によって照合データ142の時間を分割し、分割領域80a, 80b, 80c, 80d, 80e, 80f, 80gを設定する。ウィンドウ処理部151は、分割領域80a～80gに対して、分割領域番号0～6をそれぞれ設定する（ステップS10）。

【0044】

ウィンドウ処理部151は、連続する分割領域から、最後の時刻を除いた期間を予備ウィンドウに設定する。ウィンドウ処理部151は、分割領域70a, 70bについて、最後の時刻「7」を除いた期間を予備ウィンドウ71として設定する。ウィンドウ処理部151は、分割領域80a, 80bについて、最後の時刻「5」を除いた期間を予備ウィンドウ81として設定する（ステップS11）。 10

【0045】

ウィンドウ処理部151は、他の連続する分割領域についても、上記ステップS11と同様の処理を実行し、予備ウィンドウ72, 73, 74, 75および予備ウィンドウ82, 83, 84, 85, 86, 87を設定する（ステップS12）。

【0046】

ウィンドウ処理部151は、予備ウィンドウを設定した後に、照合データ142に対してイベントの走査を開始する。以下の説明では、時刻Xにおいて、イベントYを検知することを、イベント(X, Y)を検知すると適宜表記する。また、予備ウィンドウ71～75は、図2のクエリ141のウィンドウ141aに定義されたイベント条件A, B, Cに対応するイベントA, B, Cが追加された場合に、イベント条件を満たすことになる。予備ウィンドウ81～87は、クエリ141のウィンドウ141bに定義されたイベント条件D, Eに対応するイベントD, Eが追加された場合に、イベント条件を満たすことになる。 20

【0047】

ウィンドウ処理部151は、イベント(0, A)を検知する。イベントAの時刻が「0」であり、ウィンドウ幅条件が「4」であるため、分割領域番号は「 $0 \div 4 = 0$ 」となる。分割領域番号と同一の番号が割り当てられた分割領域を含む予備ウィンドウ71に、イベントAを追加する。なお、予備ウィンドウ71は、クエリ141のウィンドウ141aのイベント条件を満たしていない（ステップS13）。ステップS13に示す処理は、図5のステップS105、S106に示した処理に対応する。以下の処理についても同様である。 30

【0048】

図7の説明に移行する。ウィンドウ処理部151は、イベント(1, B)、(2, A)を検知する。イベント(1, B)、(2, A)に対する分割領域番号は「0」となる。このため、ウィンドウ処理部151は、予備ウィンドウ71にイベントA, Bを登録する。なお、予備ウィンドウ71は、クエリ141のウィンドウ141aのイベント条件を満たしていない（ステップS14）。

【0049】

ウィンドウ処理部151は、イベント(4, B)、(4, C)、(5, C)を検知する。イベント(4, B)、(4, C)、(5, C)に対する分割領域番号は「1」となる。ウィンドウ処理部151は、分割領域番号「1」に対応する予備ウィンドウ71、72にイベントB, C, Cを登録する。ウィンドウ処理部151は、予備ウィンドウ71が、クエリ141のウィンドウ141aのイベント条件を満たすので、予備ウィンドウ71を仮ウィンドウに登録する。なお、予備ウィンドウ72は、クエリ141のウィンドウ141aのイベント条件を満たしていない（ステップS15）。 40

【0050】

ウィンドウ処理部151は、イベント(9, B)、(10, A)、(10, C)を検知する。イベント(9, B)、(10, A)、(10, C)に対する分割領域番号は「2」となる。ウィンドウ処理部151は、分割領域番号「2」に対応する予備ウィンドウ72 50

、73にイベントB、A、Cを登録する。ウィンドウ処理部151は、ウィンドウ72、73が、クエリ141のウィンドウ141aのイベント条件を満たすので、予備ウィンドウ72、73を仮ウィンドウに登録する（ステップS16）。

【0051】

ウィンドウ処理部151は、イベント(11、A)、(11、B)を検知する。イベント(11、A)、(11、B)に対する分割領域番号は「2」となる。また、イベント(11、A)、(11、B)は、分割領域70cの最後の時刻である。このため、ウィンドウ処理部151は、仮ウィンドウ73にのみ、イベントA、Bを追加する（ステップS17）。

【0052】

図8の説明に移行する。ウィンドウ処理部151は、イベント(12、A)を検知する。イベント(12、A)に対する分割領域番号は「3」となる。ウィンドウ処理部151は、分割領域番号「3」に対応する仮ウィンドウ73および予備ウィンドウ74に、イベントAを追加する。予備ウィンドウ74は、クエリ141のウィンドウ141aのイベント条件を満たしていない（ステップS18）。

【0053】

ウィンドウ処理部151は、イベント(14、B)を検知する。イベント(14、B)に対する分割領域番号は「3」となる。ウィンドウ処理部151は、分割領域番号「3」に対応する仮ウィンドウ73および予備ウィンドウ74に、イベントBを追加する。予備ウィンドウ74は、クエリ141のウィンドウ141aのイベント条件を満たしていない（ステップS19）。

【0054】

ウィンドウ処理部151は、イベント(14、D)を検知する。イベント(14、D)に対する分割領域番号は「3」となる。ウィンドウ処理部151は、イベント(14、D)が、分割領域80eで最後の時刻であるため、予備ウィンドウ85にのみ、イベントDを追加する。ウィンドウ85は、クエリ141のウィンドウ141bのイベント条件を満たしていない（ステップS20）。

【0055】

ウィンドウ処理部151は、イベント(15、E)を検知する。イベント(15、E)に対する分割領域番号は「3」となる。ウィンドウ処理部151は、分割領域番号「3」に対応する予備ウィンドウ85、86に、イベントEを追加する。ウィンドウ処理部151は、予備ウィンドウ85が、クエリ141のウィンドウ141bのイベント条件を満たすので、予備ウィンドウ85を仮ウィンドウ85に登録する。予備ウィンドウ86は、クエリ141のウィンドウ141bのイベント条件を満たしていない（ステップS21）。

【0056】

図9の説明に移行する。ウィンドウ処理部151は、イベント(17、C)を検知する。イベント(17、C)に対する分割領域番号は「3」となる。ウィンドウ処理部151は、分割領域番号「3」に対応する予備ウィンドウ74、75に、イベントCを追加する。ウィンドウ処理部151は、予備ウィンドウ74が、クエリ141のウィンドウ141bのイベント条件を満たすので、予備ウィンドウ74を仮ウィンドウ74に登録する。予備ウィンドウ75は、クエリ141のウィンドウ141bのイベント条件を満たしていない（ステップS22）。

【0057】

ウィンドウ処理部151は、イベント(17、E)を検知する。イベント(17、E)に対する分割領域番号は「5」となる。ウィンドウ処理部151は、イベント(17、E)が、分割領域80fの最後の時刻であるため、予備ウィンドウ86にのみ、イベントEを登録する。予備ウィンドウ86は、クエリ141のウィンドウ141bのイベント条件を満たしていない（ステップS23）。

【0058】

ウィンドウ処理部151は、イベント(18、E)を検知する。イベント(18、E)

10

20

30

40

50

に対する分割領域番号は「6」となる。ウィンドウ処理部151は、分割領域番号「6」に対応する予備ウィンドウ86, 87に、イベントEを追加する。予備ウィンドウ86, 86は、クエリ141のウィンドウ141bのイベント条件を満たしていない（ステップS24）。

【0059】

ウィンドウ処理部151は、イベント（19, D）を検知する。イベント（19, D）に対する分割領域番号は「6」となる。ウィンドウ処理部151は、分割領域番号「6」に対応する予備ウィンドウ86, 87に、イベントDを追加する。ウィンドウ処理部151は、予備ウィンドウ86, 87が、クエリ141のウィンドウ141bのイベント条件を満たすので、予備ウィンドウ86, 87を仮ウィンドウに登録する（ステップS25）

10

【0060】

図6～図9に説明したように、ウィンドウ処理部151が、処理を実行すると、仮ウィンドウ71, 72, 73, 74と、仮ウィンドウ85, 86, 87が登録される。図10は、本実施例1に係る仮ウィンドウ構築処理の処理結果の一例を示す図である。ウィンドウ処理部151は、処理結果を、照合部152に出力する。

【0061】

照合部152は、ウィンドウ処理部151の処理結果を基にして、クエリ141の条件を満たすイベントの組み合わせを検出する処理部である。ウィンドウ処理部151の処理結果を図10に示すものであるとすると、照合部152は、仮ウィンドウ71～74と、仮ウィンドウ85～87との組み合わせについて、クエリ141のインターバル条件「5－8分」を満たす仮ウィンドウの組み合わせを判定する。

20

【0062】

クエリ141のインターバル条件「5－8分」を満たす仮ウィンドウの組み合わせは、仮ウィンドウ71と仮ウィンドウ85との組み、仮ウィンドウ72と仮ウィンドウ86との組み、仮ウィンドウ72と仮ウィンドウ87との組みとなる。

【0063】

このため、照合部152は、仮ウィンドウ71, 85の組みに含まれるイベントの種別と時刻との組みを、照合結果として出力する。また、照合部152は、仮ウィンドウ72, 86の組みに含まれるイベントの種別と時刻との組みを、照合結果として出力する。また、照合部152は、仮ウィンドウ72, 87の組みに含まれるイベントの種別と時刻との組みを、照合結果として出力する。

30

【0064】

次に、本実施例1に係る照合装置100の処理手順について説明する。図11は、本実施例1に係る照合装置の処理手順を示すフローチャートである。図11に示すように、照合装置100は、仮ウィンドウ構築処理を実行する（ステップS201）。このステップS201に示す仮ウィンドウ構築処理は、図5に示した処理手順に対応する処理である。

【0065】

照合装置100は、インターバル条件をチェックし、インターバル条件を満たす仮ウィンドウの組みとリンクを出力する（ステップS202）。照合装置100は、組み合わせ展開を実行する（ステップS203）。例えば、ステップS203において、照合装置100は、インターバル条件を満たす仮ウィンドウの組みに含まれるイベントの組みを出力する。

40

【0066】

なお、ここでは説明を省略するが、照合装置100は、インターバル条件を満たす仮ウィンドウに含まれるイベントの組み合わせ展開を行う。照合装置100は、各イベントの組み合わせにおいて、クエリ141の全ての条件を満たすか否かをチェックし、クエリ141の全ての条件を満たしていれば、かかるイベントの組み合わせを出力する。組み合わせ展開に関する詳しい説明は、実施例2において説明する。

【0067】

50

次に、本実施例 1 に係る照合装置 100 の効果について説明する。照合装置 100 は、クエリ 141 に定義されたウィンドウ幅条件よりも大きいウィンドウを、照合データ 142 に対して設定し、クエリ 141 に定義されたウィンドウのイベント条件を満たすウィンドウを、仮ウィンドウとして設定する。そして、照合装置 100 は、各仮ウィンドウの組みについて、インターバル条件をチェックし、インターバル条件を満たす仮ウィンドウの組みに含まれるイベントを照合結果として出力する。本実施例 1 に係る照合装置 100 によれば、従来技術と比較して、インターバル条件のチェック対象となるウィンドウの組み合わせを削減することができ、これにより、クエリと照合データとのマッチングを高速に行うことができる。なお、本実施例 1 では、クエリに 2 つのウィンドウが定義された最も基礎となる例にて説明したが、クエリに 3 つ以上のウィンドウが定義された場合にも適用可能である。

10

【実施例 2】

【0068】

次に、本実施例 2 に係る照合装置の構成の一例について説明する。図 12 は、本実施例 2 に係る照合装置の構成を示す機能ブロック図である。図 12 に示すように、この照合装置 200 は、通信部 110 と、入力部 120 と、表示部 130 と、記憶部 140 と、制御部 250 とを有する。通信部 110、入力部 120、表示部 130、記憶部 140 に関する説明は、実施例 1 に示した通信部 110、入力部 120、表示部 130、記憶部 140 に関する説明と同様であるため、同一の符号を付してここでは説明を省略する。

【0069】

20

制御部 250 は、ウィンドウ処理部 251 および照合部 252 を有する。制御部 250 は、例えば、ASIC や、FPGA などの集積装置に対応する。また、制御部 250 は、例えば、CPU や MPU 等の電子回路に対応する。

【0070】

ウィンドウ処理部 251 は、クエリ 141 を参照し、クエリ 141 のウィンドウのウィンドウ幅条件よりも大きい幅の仮ウィンドウを、順次設定する処理部である。図 13 は、本実施例 2 に係るウィンドウ処理部の処理の一例を説明するための図である。図 13 では一例として、ウィンドウ幅条件を「4」として説明を行う。

【0071】

ウィンドウ処理部 251 は、クエリ 141 のウィンドウ幅条件以上の値を「w」の値に設定する。ウィンドウ処理部 251 は、ウィンドウの開始時刻から「 $2 \times w$ 」過ぎてから、初めに出現するイベントを特定する。ウィンドウ処理部 251 は、初めに出現するイベントよりも、「 $w - 1$ 」時間前の時刻から次のイベントが出現する時刻を、次の仮ウィンドウの開始時刻として、順次仮ウィンドウを構築していく。

30

【0072】

ここでは一例として、w の値を「4」として、ウィンドウ処理部 251 の説明を行う。図 13 に示すように、ウィンドウ処理部 251 は、ウィンドウ 260 の開始時刻「0」から「 $2 \times w$ 」過ぎてから、初めに出現するイベント（9, B）を特定する。ウィンドウ処理部 251 は、特定したイベント（9, B）よりも、「 $w - 1 = 3$ 」時間前の時刻「6」から次のイベントが出現する時刻「9」を、次のウィンドウ 261 の開始時刻として特定する。

40

【0073】

ここで、ウィンドウ処理部 251 が仮ウィンドウを構築する処理手順について説明する。図 14 は、本実施例 2 に係る仮ウィンドウ構築処理の処理手順を示すフローチャートである。図 14 に示すように、ウィンドウ処理部 251 は、1 つのイベントを検知し（ステップ S301）、イベントを該当する仮ウィンドウに追加する（ステップ S302）。

【0074】

ウィンドウ処理部 251 は、仮ウィンドウが存在しない場合には（ステップ S303, No）、ステップ S312 に移行する。

【0075】

50

一方、ウィンドウ処理部251は、仮ウィンドウが存在する場合には（ステップS303, Yes）、イベントの時刻から最新の仮ウィンドウの開始時刻を減算した値が、「 $2w-1$ 」以上であるか否かを判定する（ステップS304）。ウィンドウ処理部251は、イベントの時刻から最新の仮ウィンドウの開始時刻を減算した値が、「 $2w-1$ 」以上でない場合には（ステップS304, No）、ステップS309に移行する。

【0076】

ウィンドウ処理部251は、イベントの時刻から最新の仮ウィンドウの開始時刻を減算した値が、「 $2w-1$ 」以上の場合には（ステップS304, Yes）、最新の仮ウィンドウが全てのイベント条件を満たすか否かを判定する（ステップS305）。ウィンドウ処理部251は、最新の仮ウィンドウが全てのイベント条件を満たさない場合には（ステップS305, No）、ステップS311に移行する。

10

【0077】

ウィンドウ処理部251は、最新の仮ウィンドウが全てのイベント条件を満たす場合には（ステップS305, Yes）、ステップS307に移行する。ウィンドウ処理部251は、最新の仮ウィンドウ内に「イベントの時刻 $-w+1$ 」の時刻以降にイベントがあるか否かを判定する（ステップS306）。ウィンドウ処理部251は、最新の仮ウィンドウ内に「イベントの時刻 $-w+1$ 」の時刻以降にイベントがない場合には（ステップS306, No）、ステップS309に移行する。

【0078】

ウィンドウ処理部251は、最新の仮ウィンドウ内に「イベントの時刻 $-w+1$ 」の時刻以降にイベントが有る場合には（ステップS306, Yes）、ステップS307に移行する。ウィンドウ処理部251は、最新の仮ウィンドウに含まれるイベントから「イベントの時刻 $-w+1$ 」の時刻から初めのイベントを取り出す（ステップS307）。

20

【0079】

ウィンドウ処理部251は、仮ウィンドウを新しく生成し、取り出したイベントの時刻を新しく生成した仮ウィンドウの開始時刻とし、開始時刻以降のイベントを新しい仮ウィンドウに追加する（ステップS308）。ウィンドウ処理部251は、イベント時刻をウィンドウの終了時刻とし、イベントを仮ウィンドウに追加する（ステップS309）。

【0080】

ウィンドウ処理部251は、処理を終了する場合には（ステップS310, Yes）、仮ウィンドウ構築処理を終了する。一方、ウィンドウ処理部251は、処理を終了しない場合には（ステップS310, No）、ステップS301に移行する。

30

【0081】

ところで、ステップS311において、ウィンドウ処理部251は、最新の仮ウィンドウを削除する（ステップS311）。ウィンドウ処理部251は、仮ウィンドウを新しく生成し、イベントの時刻を新しい仮ウィンドウの開始時刻に設定し（ステップS312）、ステップS309に移行する。

【0082】

ここで、図2に示したクエリ141と、図4に示した照合データ142を例にして、ウィンドウ処理部251が実行する仮ウィンドウ構築処理について説明する。図15～図17は、本実施例2に係る仮ウィンドウ構築処理を説明するための図である。図15～図17の説明では、イベントA～Cに対するウィンドウの w の値を「4」とし、イベントD, Eに対するウィンドウの w の値を「3」として説明を行う。

40

【0083】

図15について説明する。ウィンドウ処理部251は、イベント(0, A)を検知する。ウィンドウ処理部251は、仮ウィンドウ260を生成する。ウィンドウ処理部251は、イベントAを、仮ウィンドウ260に追加する（ステップS30）。

【0084】

ウィンドウ処理部251は、イベント(1, B)を検知する。ウィンドウ処理部251は、仮ウィンドウ260の終了時刻を、イベント(1, B)の時刻「1」に設定する。ウ

50

ウィンドウ処理部251は、イベントBを、仮ウィンドウ260に追加する（ステップS31）。

【0085】

ウィンドウ処理部251は、イベント(2, A)、(4, B)、(4, C)、(5, C)を検知する。ウィンドウ処理部251は、仮ウィンドウ260の終了時刻をイベント(5, C)の時刻「5」に設定する。ウィンドウ処理部251は、各イベントA, B, C, Cを、仮ウィンドウ260に追加する（ステップS32）。

【0086】

ウィンドウ処理部251は、イベント(9, B)を検知する。イベント(9, B)のイベント時刻「9」から最新の仮ウィンドウ260の開始時刻「0」を減算した値が、 $2w-1=7$ の値よりも大きくなる。また、最新の仮ウィンドウ260は、全てのイベント条件を満たす。また、「イベント(9, B)のイベント時刻「9」 $-w+1$ 」 $=$ 「 $9-4+1=6$ 」以降に、最新の仮ウィンドウ260のイベントが存在しない。このため、ウィンドウ処理部251は、新しい仮ウィンドウ261を生成し、仮ウィンドウ261の開始時刻および終了時刻を「9」に設定する（ステップS33）。

10

【0087】

ウィンドウ処理部251は、イベント(10, A)、(10, C)、(11, A)、(11, B)、(12, A)、(14, B)を検知する。ウィンドウ処理部251は、各イベントA, C, A, B, A, Bを、最新の仮ウィンドウ261に設定する。ウィンドウ処理部251は、仮ウィンドウの終了時刻を「14」に設定する（ステップS34）。

20

【0088】

ウィンドウ処理部251は、イベント(14, D)を検知する。ウィンドウ処理部251は、ウィンドウ270を生成する。ウィンドウ処理部251は、イベントDを、仮ウィンドウ270に追加する（ステップS35）。

【0089】

ウィンドウ処理部251は、イベント(15, E)を検知する。ウィンドウ処理部251は、イベントEを仮ウィンドウ270に追加する（ステップS36）。

【0090】

ウィンドウ処理部251は、イベント(17, C)を検知する。イベント(17, C)のイベント時刻「17」から最新の仮ウィンドウ261の開始時刻「9」を減算した値が、 $2 \times 2 - 1 = 7$ の値よりも大きくなる。最新の仮ウィンドウ261は、全てのイベント条件を満たす。また、「イベント(17, C)のイベント時刻「17」 $-w+1$ 」 $=$ 「 $17-4+1=14$ 」以降に、最新の仮ウィンドウ261のイベント(14, B)が存在する。このため、ウィンドウ処理部251は、新しい仮ウィンドウ262を生成し、仮ウィンドウ262の開始時刻をイベント(14, B)の開始時刻「14」に設定する。また、ウィンドウ処理部251は、イベント(17, C)を、仮ウィンドウ262に追加し、イベント(17, C)の時刻「17」を、仮ウィンドウ262の終了時刻に設定する（ステップS37）。

30

【0091】

図17の説明に移行する。ウィンドウ処理部251は、イベント(17, E)、(18, E)を検知する。ウィンドウ処理部251は、イベントE, Eを仮ウィンドウ270に追加する（ステップS38）。

40

【0092】

ウィンドウ処理部251は、イベント(19, D)を検知する。イベント(19, D)のイベント時刻「19」から最新の仮ウィンドウ270の開始時刻「14」を減算した値が、 $2 \times 3 - 1 = 5$ の値よりも大きくなる。最新の仮ウィンドウ270は、全てのイベント条件を満たす。また、「イベント(19, D)のイベント時刻「19」 $-w+1$ 」 $=$ 「 $17-3+1=15$ 」以降に、最新の仮ウィンドウ261のイベント(17, E)が存在する。このため、ウィンドウ処理部251は、新しい仮ウィンドウ271を生成し、仮ウィンドウ271の開始時刻をイベント(17, E)の開始時刻に設定する。また、ウィン

50

ドウ処理部 251 は、イベント (18, E)、(19, D) を、仮ウィンドウ 271 に追加し、イベント (19, D) の時刻「19」を、仮ウィンドウ 271 の終了時刻に設定する (ステップ S39)。

【0093】

ウィンドウ処理部 251 は、時刻 21 以降のイベントの検知を試みるが、イベントは存在しない。このため、最新の仮ウィンドウ 262 は、全てのイベント条件を満たさない。ウィンドウ処理部 251 は、仮ウィンドウ 262 を削除する (ステップ S40)。

【0094】

ウィンドウ処理部 251 は、上記図 15～図 17 に示す処理を実行することで、仮ウィンドウ 260, 261 および仮ウィンドウ 270, 271 を構築する。ウィンドウ処理部 251 は、構築した仮ウィンドウの情報を、照合部 252 に出力する。

10

【0095】

図 12 の説明に戻る。照合部 252 は、ウィンドウ処理部 251 によって構築された仮ウィンドウと、クエリ 141 とを基にして、クエリ 141 の条件を満たすイベントの組みを照合データ 142 から検出する処理部である。

【0096】

照合部 252 は、各仮ウィンドウの組みが、クエリ 141 のインターバル条件を満たすか否かを確認する場合に、仮ウィンドウの組みの終了時刻と開始時刻と時間間隔を利用する代わりに、仮ウィンドウ内のイベントの時刻を考慮して行う。以下の説明では、仮ウィンドウの組みについて、先行する仮ウィンドウを先行ウィンドウと表記し、後続する仮ウィンドウを後続ウィンドウと表記する。

20

【0097】

照合部 252 は、先行ウィンドウと後続ウィンドウとの関係が「最小インターバル条件」および「最大インターバル条件」を満たす場合に、先行ウィンドウと後続ウィンドウとの組みを出力する。

【0098】

まず、最小インターバル条件について説明する。照合部 252 は、先行ウィンドウと後続ウィンドウとの「最小間隔」が、インターバル条件の最小値より大となる場合に、「最小インターバル条件」を満たすと判定する。図 18 は、最小インターバル条件を説明するための図である。

30

【0099】

図 18 について、先行ウィンドウをウィンドウ 280 とし、後続ウィンドウをウィンドウ 290 とする。照合部 252 は、先行ウィンドウ 280 に含まれるイベントのうち、イベント条件を満たす最初のイベントの組みの中で一番遅いイベントの時刻を、第 1 基準時刻として特定する。そうすると、第 1 基準時刻は「10」となる。

【0100】

照合部 252 は、後続ウィンドウ 290 に含まれるイベントのうち、イベント条件を満たす最後のイベントの組みの中で一番早いイベントの時刻を、第 2 基準時刻として特定する。そうすると、第 2 基準時刻は「14」となる。

【0101】

40

照合部 252 は、第 1 基準時刻と第 2 基準時刻との間隔を最小間隔として算出する。図 18 に示す例では、最小間隔は「4」となり、インターバル条件の最小値「4」以下である。このため、照合部 252 は、仮ウィンドウ 280, 290 の組みについて、最小インターバルの条件を満たさないと判定する。なお、仮に、第 1 基準時刻が「9」より早い時刻であれば、最小インターバル条件を満たすことになる。

【0102】

続いて、最大インターバル条件について説明する。照合部 252 は、先行ウィンドウと後続ウィンドウとの「最大間隔」が、インターバル条件の最大値と、先行ウィンドウ幅条件と、後続ウィンドウ幅条件を加算した値以下となっている場合に、「最大インターバル条件」を満たすと判定する。図 19 は、最大インターバル条件を説明するための図である

50

。

【0103】

図19について、先行ウィンドウをウィンドウ281とし、後続ウィンドウをウィンドウ291とする。照合部252は、先行ウィンドウ281に含まれるイベントのうち、イベント条件を満たす最後のイベントの組みの中で一番早いイベントの時刻を、第3基準時刻として特定する。そうすると、第3基準時刻は「2」となる。

【0104】

照合部252は、後続ウィンドウ291に含まれる最初のイベントの組みの中で一番遅いイベントの時刻を、第4基準時刻として特定する。そうすると、第4基準時刻は「19」となる。

10

【0105】

照合部252は、第3基準時刻と第4基準時刻との間隔を最大間隔として算出する。図19に示す例では、最大間隔は「17」となる。一方、インターバル条件の最大値「8」と、先行ウィンドウ幅「4」と、後続ウィンドウ幅「3」を加算した値は「15」となる。このため、照合部252は、仮ウィンドウ281、291の組みについて、最大インターバル条件を満たさないと判定する。なお、仮に、第3基準時刻が「5」より遅い時刻であれば、最大インターバル条件を満たすことになる。

【0106】

ここで、照合部252がインターバル条件を判定する処理手順について説明する。図20および図21は、インターバル条件判定処理の処理手順を示すフローチャートである。図20に示すように、照合部252は、クエリ141の先行するウィンドウ条件に対応する先頭の仮ウィンドウを取り出し、先行ウィンドウに設定する（ステップS401）。

20

【0107】

照合部252は、クエリ141の後続ウィンドウ条件に対応する先頭の仮ウィンドウを取り出して後続ウィンドウに設定し、さらに、ベース後続ウィンドウに設定する（ステップS402）。

【0108】

照合部252は、先行ウィンドウと後続ウィンドウについて、インターバル条件をチェックする（ステップS403）。照合部252は、最小インターバル条件を満たさない場合には（ステップS404, No）、図21のステップS407に移行する。

30

【0109】

一方、照合部252は、最小インターバル条件を満たす場合には（ステップS404, Yes）、最大インターバル条件を満たすか否かを判定する（ステップS405）。照合部252は、最大インターバル条件を満たさない場合には（ステップS405, No）、図21のステップS411に移行する。

【0110】

一方、照合部252は、最大インターバル条件を満たす場合には（ステップS405, Yes）、ステップS406に移行する。照合部252は、該当する先行ウィンドウと後続ウィンドウとの組みを出力していなければ出力し、先行ウィンドウと後続ウィンドウとのリンクを出力し（ステップS406）、図20のステップS408に移行する。

40

【0111】

図21の説明に移行する。照合部252は、後続ウィンドウの次のウィンドウをベース後続ウィンドウに設定する（ステップS407）。照合部252は、後続ウィンドウの次の仮ウィンドウが存在する場合には（ステップS408, Yes）、後続ウィンドウの次の仮ウィンドウを取り出し、後続ウィンドウに設定し（ステップS409）、図20のステップS403に移行する。

【0112】

一方、照合部252は、後続ウィンドウの次の仮ウィンドウが存在しない場合には（ステップS408, No）、先行ウィンドウの次の仮ウィンドウが存在するか否かを判定する（ステップS410）。照合部252は、先行ウィンドウの次の仮ウィンドウが存在し

50

ない場合には（ステップS 4 1 0， N o）、処理を終了する。

【0 1 1 3】

一方、照合部2 5 2は、先行ウィンドウの次の仮ウィンドウが存在する場合には（ステップS 4 1 0， Y e s）、先行ウィンドウの次の仮ウィンドウを取り出し、先行ウィンドウに設定する（ステップS 4 1 1）。

【0 1 1 4】

照合部2 5 2は、ベース後続ウィンドウがN U L Lである場合には（ステップS 4 1 2， Y e s）、処理を終了する。照合部2 5 2は、ベース後続ウィンドウがN U L Lでない場合には（ステップS 4 1 2， N o）、ベース後続ウィンドウを後続ウィンドウに設定し（ステップS 4 1 3）、図2 0のステップS 4 0 3に移行する。

10

【0 1 1 5】

ここで、図1 7に示した仮ウィンドウの構築結果と、図2に示したクエリ1 4 1とを例にして、照合部2 5 2が実行するインターバル条件判定処理について説明する。図2 2～図2 5は、本実施例2に係るインターバル条件判定処理を説明するための図である。

【0 1 1 6】

図2 2について説明する。照合部2 5 2は、クエリ1 4 1の先行するウィンドウ1 4 1 aの条件に対応する先頭の仮ウィンドウ2 6 0を先行ウィンドウに設定する。照合部2 5 2は、クエリ1 4 1の後続ウィンドウ1 4 1 bの条件に対応する後続の仮ウィンドウ2 7 0を、後続ウィンドウに設定する。また、後続ウィンドウ2 7 0を、ベース後続ウィンドウに設定する（ステップS 5 0）。

20

【0 1 1 7】

照合部2 5 2は、先行ウィンドウ2 6 0と後続ウィンドウ2 7 0について、最小インターバル条件を満たすか否かを判定する。先行ウィンドウ2 6 0の第1基準時刻は、イベント（4， C）の「4」となる。後続ウィンドウ2 7 0の第2基準時刻は、イベント（1 4， D）の「1 4」となる。照合部2 5 2は、最小間隔が「1 0」となり、インターバル条件の最小値「4」よりも大となるため、先行ウィンドウ2 6 0と後続ウィンドウ2 7 0とは、最小インターバル条件を満たすと判定する（ステップS 5 1）。

【0 1 1 8】

照合部2 5 2は、先行ウィンドウ2 6 0と後続ウィンドウ2 7 0について、最大インターバル条件を満たすか否かを判定する。先行ウィンドウ2 6 0の第3基準時刻は、イベント（2， A）の「2」となる。後続ウィンドウ2 7 0の第4基準時刻は、イベント（1 5， E）の「1 5」となる。ここで、最大間隔が「1 3」となり、インターバル条件の最大値「8」と、先行ウィンドウ幅条件「4」と、後続ウィンドウ幅条件「3」を加算した値「1 5」以下となる。このため、照合部2 5 2は、先行ウィンドウ2 6 0と後続ウィンドウ2 7 0とが、最大インターバル条件を満たすと判定する（ステップS 5 2）。

30

【0 1 1 9】

先行ウィンドウ2 6 0および後続ウィンドウ2 7 0が最小インターバル条件および最大インターバル条件を満たす。このため、照合部2 5 2は、先行ウィンドウ2 6 0および後続ウィンドウ2 7 0の組みを出力し、リンクを出力する（ステップS 5 3）。

【0 1 2 0】

図2 3の説明に移行する。照合部2 5 2は、後続ウィンドウ2 7 0の次の仮ウィンドウ2 7 1を後続ウィンドウに設定する（ステップS 5 4）。

40

【0 1 2 1】

照合部2 5 2は、先行ウィンドウ2 6 0と後続ウィンドウ2 7 1とについて、最小インターバル条件を満たすか否かを判定する。先行ウィンドウ2 6 0の第1基準時刻は、イベント（4， C）の「4」となる。後続ウィンドウ2 7 1の第2基準時刻は、イベント（1 8， E）の「1 8」となる。照合部2 5 2は、最小間隔が「1 4」となり、インターバル条件の最小値「4」よりも大となるため、先行ウィンドウ2 6 0と後続ウィンドウ2 7 1とは、最小インターバル条件を満たすと判定する（ステップS 5 5）。

【0 1 2 2】

50

照合部 252 は、先行ウィンドウ 260 と後続ウィンドウ 271 について、最大インターバル条件を満たすか否かを判定する。先行ウィンドウ 260 の第 3 基準時刻は、イベント (2, A) の「2」となる。後続ウィンドウ 260 の第 4 基準時刻は、イベント (19, D) の「19」となる。ここで、最大間隔が「17」となり、インターバル条件の最大値「8」と、先行ウィンドウ幅条件「4」と、後続ウィンドウ幅条件「3」を加算した値「15」よりも大となる。このため、照合部 252 は、先行ウィンドウ 260 と後続ウィンドウ 271 とが、最大インターバル条件を満たさないと判定する (ステップ S56)。

【0123】

図 24 の説明に移行する。照合部 252 は、先行ウィンドウ 260 の次の仮ウィンドウ 261 を先行ウィンドウに設定する。照合部 252 は、後続ウィンドウ 270 をベース後続ウィンドウに設定する (ステップ S57)。

10

【0124】

照合部 252 は、先行ウィンドウ 261 と後続ウィンドウ 270 について、最小インターバル条件を満たすか否かを判定する。先行ウィンドウ 261 の第 1 基準時刻は、イベント (10, A) の「10」となる。後続ウィンドウ 270 の第 2 基準時刻は、イベント (14, D) の「14」となる。照合部 252 は、最小間隔が「4」となり、インターバル条件の最小値「4」以下となるため、先行ウィンドウ 261 と後続ウィンドウ 270 とは、最小インターバル条件を満たさないと判定する。なお、先行ウィンドウ 261 とベース後続ウィンドウ 270 とが、最小インターバル条件を満たさない場合には、先行ウィンドウ 261 と仮ウィンドウ 271 とともに最小インターバル条件を満たさない (ステップ S58)。

20

【0125】

照合部 252 は、後続ウィンドウ 270 の次の仮ウィンドウ 271 を後続ウィンドウおよびベース後続ウィンドウに設定する (ステップ S59)。

【0126】

照合部 252 は、先行ウィンドウ 261 と後続ウィンドウ 271 について、最小インターバル条件を満たすか否かを判定する。先行ウィンドウ 261 の第 1 基準時刻は、イベント (10, A) の「10」となる。後続ウィンドウ 271 の第 2 基準時刻は、イベント (18, E) の「18」となる。照合部 252 は、最小間隔が「8」となり、インターバル条件の最小値「4」よりも大となるため、先行ウィンドウ 261 と後続ウィンドウ 271 とは、最小インターバル条件を満たすと判定する (ステップ S60)。

30

【0127】

図 25 の説明に移行する。照合部 252 は、先行ウィンドウ 261 と後続ウィンドウ 271 について、最大インターバル条件を満たすか否かを判定する。先行ウィンドウ 261 の第 3 基準時刻は、イベント (10, C) の「10」となる。後続ウィンドウ 271 の第 4 基準時刻は、イベント (19, D) の「19」となる。ここで、最大間隔が「9」となり、インターバル条件の最大値「8」と、先行ウィンドウ幅条件「4」と、後続ウィンドウ幅条件「3」を加算した値「15」以下となる。このため、照合部 252 は、先行ウィンドウ 261 と後続ウィンドウ 271 とが、最大インターバル条件を満たすと判定する (ステップ S61)。

40

【0128】

先行ウィンドウ 261 および後続ウィンドウ 271 が最小インターバル条件および最大インターバル条件を満たす。このため、照合部 252 は、先行ウィンドウ 261 および後続ウィンドウ 271 の組みを出力し、リンクを出力する (ステップ S62)。

【0129】

照合部 252 は、先行ウィンドウ 261 の次の仮ウィンドウが存在しないため、インターバル条件判定処理を終了する。照合部 252 は、図 22～図 25 の処理を実行することで、先行ウィンドウ 260 と後続ウィンドウ 270 との組みと、先行ウィンドウ 261 と後続ウィンドウ 271 との組みを出力する。

【0130】

50

照合部 252 は、各仮ウィンドウについてインターバル条件判定処理を実行した後に、組み合わせ展開を行い、クエリ 141 の条件を満たすイベントの組みを判定する。図 26 は、組み合わせ展開の結果の一例を示す図である。

【0131】

図 26 において、イベント群 300 は、先行ウィンドウ 260 と後続ウィンドウ 270 との組み合わせ展開の結果である。イベント群 400 は、先行ウィンドウ 261 と後続ウィンドウ 271 の組み合わせ展開の結果である。照合部 252 は、イベント群 300、400 について、クエリ 141 の条件を満たすイベントの組みを特定する。

【0132】

イベント群 300 において、イベント 306、309 の組みが、クエリ 141 の条件を満たす。イベント群 400 において、イベント 401～404、406～409、411～414 の組みが、クエリ 141 の条件を満たす。このため、照合部 252 は、イベント 306、309、イベント 401～404、406～409、411～414 の組みを、クエリ 141 の照合結果として出力する。

【0133】

次に、本実施例 2 に係る照合装置 200 の処理手順について説明する。図 27 は、本実施例 2 に係る照合装置の処理手順を示すフローチャートである。図 27 に示すように、照合装置 100 は、仮ウィンドウ構築処理を実行する（ステップ S501）。ステップ S501 に示す仮ウィンドウ構築処理は、図 14 に示した処理手順に対応する処理である。

【0134】

照合装置 200 は、インターバル条件をチェックし、インターバル条件を満たす仮ウィンドウの組みとリンクを出力する（ステップ S502）。ステップ S502 に示す処理は、図 20 及び図 21 に示した処理手順に対応する処理である。照合装置 200 は、組み合わせ展開を実行し、クエリ 141 の条件を満たすイベントの組みを出力する（ステップ S503）。

【0135】

次に、本実施例 2 に係る照合装置 200 の効果について説明する。照合装置 200 は、クエリ 141 のウィンドウ幅条件以上の値を「w」の値に設定し、ウィンドウの開始時刻から「2×w」過ぎてから、初めに出現するイベント特定する。そして、照合装置 200 は、初めに出現するイベントよりも、「w-1」時間前の時刻から次のイベントが出現する時刻を、次の仮ウィンドウの開始時刻として、順次仮ウィンドウを構築し、クエリ 141 の条件を満たすか否かを判定する。このため、照合装置 200 によれば、インターバル条件のチェック対象となるウィンドウの組み合わせを削減することができ、これにより、クエリと照合データとのマッチングを高速に行うことができる。

【0136】

また、照合装置 200 は、先行ウィンドウおよび後続ウィンドウに含まれるイベントの時刻を基にして、クエリ 141 のインターバル条件が満たされるか否かを判定する。具体的に、照合装置 200 は、先行ウィンドウに含まれるイベントのうち、イベント条件を満たす最初のイベントの組みの中で一番遅いイベントの時刻を、第 1 基準時刻として特定する。照合装置 200 は、後続ウィンドウに含まれるイベントのうち、イベント条件を満たす最後のイベントの組みの中で一番早いイベントの時刻を、第 2 基準時刻として特定する。そして、照合装置 200 は、第 1 基準時刻および第 2 基準時刻のインターバルと、最小インターバル条件とを基にして、最小インターバル条件を満たすか否かを判定する。

【0137】

更に、照合装置 200 は、先行ウィンドウに含まれるイベントのうち、イベント条件を満たす最後のイベントの組みの中で一番早いイベントの時刻を、第 3 基準時刻として特定する。照合装置 200 は、後続ウィンドウに含まれる最初のイベントの組みの中で一番遅いイベントの時刻を、第 4 基準時刻として特定する。照合装置 200 は、第 3 基準時刻および第 4 基準時刻のインターバルと、最大インターバル条件、先行ウィンドウの幅、および後続ウィンドウの幅を基にして、最大インターバル条件を満たすか否かを判定する。

【0138】

そして、照合装置200は、最小インターバル条件を満たし、かつ、最大インターバル条件を満たす場合に、先行ウィンドウと後続ウィンドウとが、クエリ141のインターバル条件を満たすと判定する。このため、クエリ141の解となり得るイベントの組みを含んだウィンドウの組みをより正確に特定することができる。

【0139】

次に、上記の実施例に示した照合装置と同様の機能を実現する照合プログラムを実行するコンピュータの一例を説明する。図28は、照合プログラムを実行するコンピュータの一例を示す図である。

【0140】

図28に示すように、コンピュータ500は、各種演算処理を実行するCPU501と、ユーザからのデータの入力を受け付ける入力装置502と、ディスプレイ503を有する。また、コンピュータ500は、記憶媒体からプログラム等を読み取り装置504と、ネットワークを介して他のコンピュータとの間でデータの授受を行うインターフェース装置505とを有する。また、コンピュータ500は、各種情報を一時記憶するRAM506と、ハードディスク装置507を有する。そして、各装置501～507は、バス508に接続される。

【0141】

ハードディスク装置507は、ウィンドウ処理プログラム507a、照合プログラム507bを有する。CPU501は、各プログラム507a、507bを読み出してRAM506に展開する。

【0142】

ウィンドウ処理プログラム507aは、ウィンドウ処理プロセス506aとして機能する。照合プログラム507bは、照合プロセス506bとして機能する。

【0143】

例えば、ウィンドウ処理プロセス506aは、ウィンドウ処理部151、251に対応する。照合プロセス506bは、照合部152、252に対応する。

【0144】

なお、各プログラム507a、507bについては、必ずしも最初からハードディスク装置507に記憶させておかなくてもよい。例えば、コンピュータ500に挿入されるフレキシブルディスク(FD)、CD-ROM、DVDディスク、光磁気ディスク、ICカードなどの「可搬用の物理媒体」に各プログラムを記憶させておく。そして、コンピュータ500がこれらから各プログラム507a、507bを読み出して実行するようにしてもよい。

【0145】

以上の各実施例を含む実施形態に関し、さらに以下の付記を開示する。

【0146】

(付記1) コンピュータが実行する照合方法であって、

第1ウィンドウにイベント条件が対応付けられ、前記第1ウィンドウに所定のウィンドウ幅条件が設定され、複数の前記第1ウィンドウが所定のインターバル条件によって直列に接続されたクエリに基づいて、前記ウィンドウ幅条件よりも大きいウィンドウ幅の第2ウィンドウを、イベントと前記イベントの出現時刻とを対応付けた照合データに設定し、

前記第2ウィンドウに含まれる前記照合データのイベントおよび各第2ウィンドウのインターバルが、前記第1ウィンドウのイベント条件および前記インターバル条件を満たすか否かを判定する

各処理を実行することを特徴とする照合方法。

【0147】

(付記2) 前記第2ウィンドウを前記照合データに設定する処理は、前記第2ウィンドウの開始時刻を起点として、前記第2ウィンドウの幅を2倍した時刻よりも後に出現する初めのイベントの時刻から、前記第2ウィンドウの幅だけ前の時刻に戻り、戻った時刻の次

10

20

30

40

50

の時刻より後に出現する初めのイベントの時刻を開始時刻とする次の第2ウィンドウを設定することを特徴とする付記1に記載の照合方法。

【0148】

(付記3) 前記インターバル条件は最小インターバル条件を含み、前記判定する処理は、先行の第2ウィンドウに含まれるイベントのうち、前記イベント条件を満たす最初のイベントの組みの中で一番遅いイベントの第1時刻と、後続の第2ウィンドウに含まれるイベントのうち、前記イベント条件を満たす最後のイベントの組みの中で一番早いイベントの第2時刻とを特定し、前記第1時刻と前記第2時刻とのインターバルと、前記最小インターバル条件とを基にして、先行の第2ウィンドウと後続の第2ウィンドウとが、前記インターバル条件を満たすか否かを判定することを特徴とする付記2に記載の照合方法。

10

【0149】

(付記4) 前記インターバル条件は最大インターバル条件を含み、前記判定する処理は、先行の第2ウィンドウに含まれるイベントのうち、前記イベント条件を満たす最後のイベントの組みの中で一番早いイベントの第3時刻と、後続の第2ウィンドウに含まれるイベントのうち、前記イベント条件を満たす最初のイベントの組みの中で一番遅いイベントの第4時刻とを特定し、前記第3時刻と前記第4時刻とのインターバルと、前記最大インターバル条件と、先行の第2ウィンドウの幅と、後続の第2ウィンドウの幅とを基にして、先行の第2ウィンドウと後続の第2ウィンドウとが、前記インターバル条件を満たすか否かを判定することを特徴とする付記2に記載の照合方法。

【0150】

20

(付記5) コンピュータに、

第1ウィンドウにイベント条件が対応付けられ、前記第1ウィンドウに所定のウィンドウ幅条件が設定され、複数の前記第1ウィンドウが所定のインターバル条件によって直列に接続されたクエリに基づいて、前記ウィンドウ幅条件よりも大きいウィンドウ幅の第2ウィンドウを、イベントと前記イベントの出現時刻とを対応付けた照合データに設定し、

前記第2ウィンドウに含まれる前記照合データのイベントおよび各第2ウィンドウのインターバルが、前記第1ウィンドウのイベント条件および前記インターバル条件を満たすか否かを判定する

各処理を実行させる照合プログラム。

【0151】

30

(付記6) 前記第2ウィンドウを前記照合データに設定する処理は、前記第2ウィンドウの開始時刻を起点として、前記第2ウィンドウの幅を2倍した時刻よりも後に出現する初めのイベントの時刻から、前記第2ウィンドウの幅だけ前の時刻に戻り、戻った時刻の次の時刻より後に出現する初めのイベントの時刻を開始時刻とする次の第2ウィンドウを設定することを特徴とする付記5に記載の照合プログラム。

【0152】

(付記7) 前記インターバル条件は最小インターバル条件を含み、前記判定する処理は、先行の第2ウィンドウに含まれるイベントのうち、前記イベント条件を満たす最初のイベントの組みの中で一番遅いイベントの第1時刻と、後続の第2ウィンドウに含まれるイベントのうち、前記イベント条件を満たす最後のイベントの組みの中で一番早いイベントの第2時刻とを特定し、前記第1時刻と前記第2時刻とのインターバルと、前記最小インターバル条件とを基にして、先行の第2ウィンドウと後続の第2ウィンドウとが、前記インターバル条件を満たすか否かを判定することを特徴とする付記6に記載の照合プログラム。

40

【0153】

(付記8) 前記インターバル条件は最大インターバル条件を含み、前記判定する処理は、先行の第2ウィンドウに含まれるイベントのうち、前記イベント条件を満たす最後のイベントの組みの中で一番早いイベントの第3時刻と、後続の第2ウィンドウに含まれるイベントのうち、前記イベント条件を満たす最初のイベントの組みの中で一番遅いイベントの第4時刻とを特定し、前記第3時刻と前記第4時刻とのインターバルと、前記最大インタ

50

ーバル条件と、先行の第2ウィンドウの幅と、後続の第2ウィンドウの幅とを基にして、先行の第2ウィンドウと後続の第2ウィンドウとが、前記インターバル条件を満たすか否かを判定することを特徴とする付記6に記載の照合プログラム。

【符号の説明】

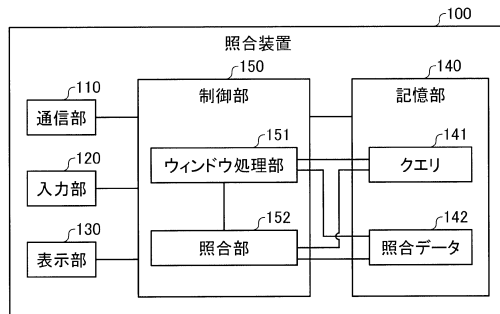
【0154】

- 100 照合装置
- 141 クエリ
- 142 照合データ
- 151, 251 ウィンドウ処理部
- 152, 252 照合部

10

【図1】

本実施例1に係る照合装置の構成を示す機能ブロック図



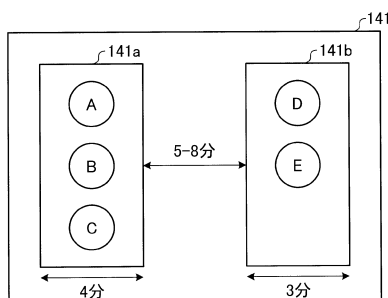
【図3】

本実施例1に係る照合データのデータ構造の一例を示す図

時刻	顧客ID	場所	イベント
0	432	3F	A
1	432	2F	B
2	432	2F	A
4	432	2F	B
4	432	2F	C
5	432	2F	C
9	432	2F	B
10	432	2F	A
10	432	2F	C
11	432	2F	A
11	432	2F	B
12	432	2F	A
14	432	2F	B
14	432	2F	D
15	432	2F	E
17	432	2F	C
17	432	2F	E
18	432	2F	E
19	432	2F	D

【図2】

本実施例1に係るクエリのデータ構造の一例を示す図



【図 4】

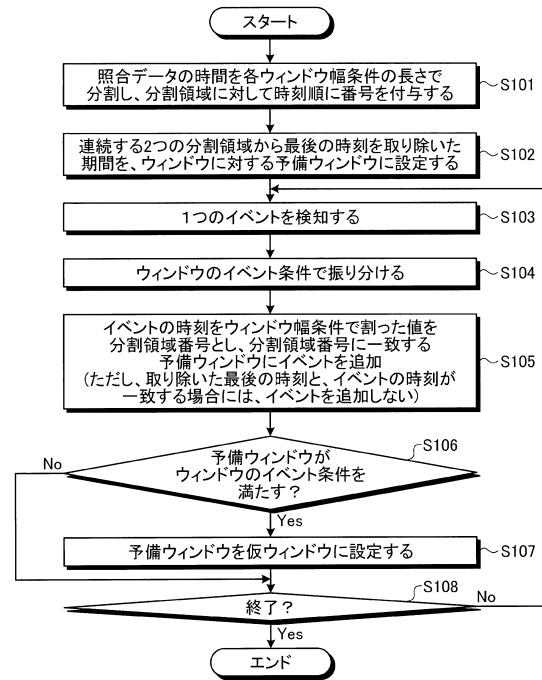
照合データのデータイメージを示す図

時刻	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	A									A	A	A									
B		B							B	B	B				B						
C					B	C	C				C										
D															D				D		
E																E	E	E			

142

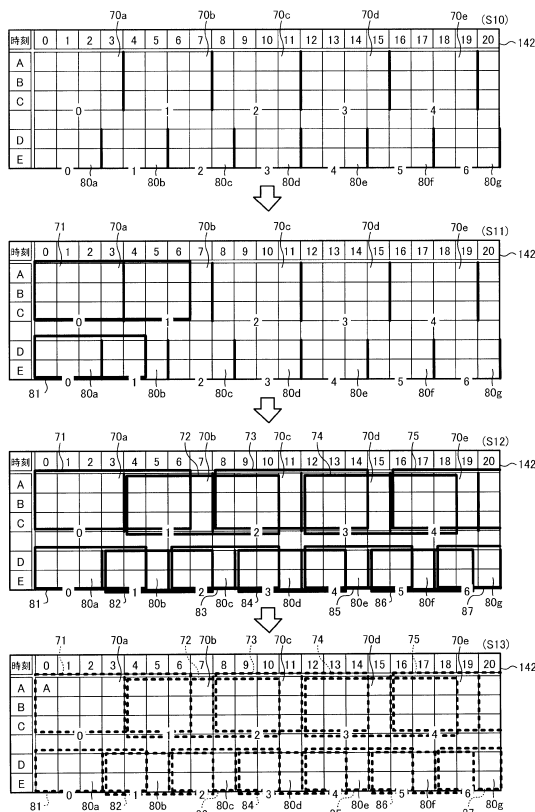
【図 5】

本実施例1に係る仮ウィンドウ構築処理の処理手順を示すフローチャート



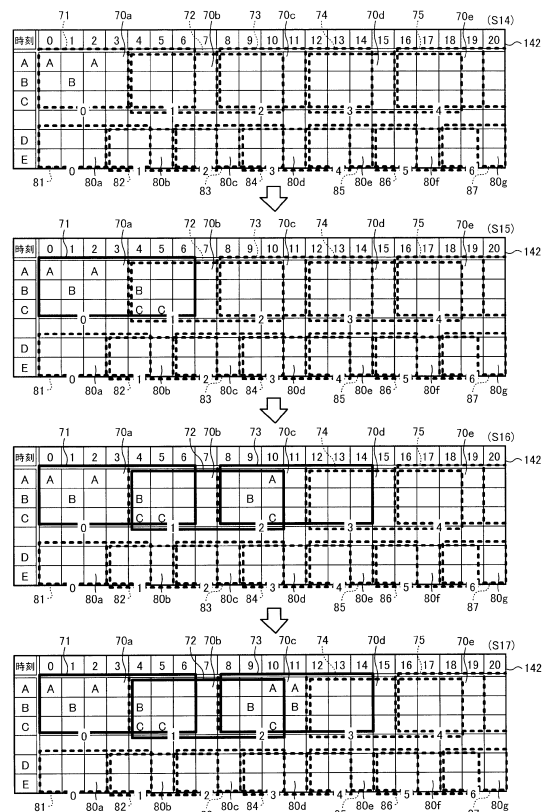
【図 6】

本実施例1に係る仮ウィンドウ構築処理を説明するための図(1)



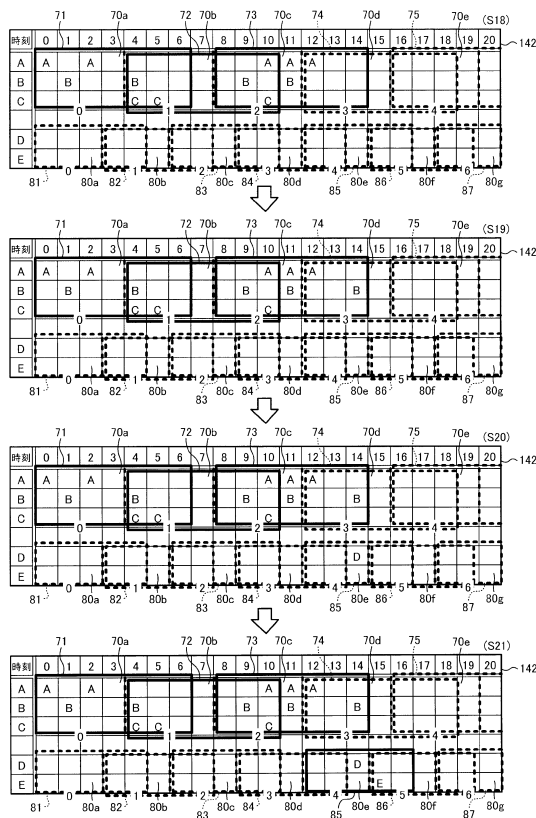
【図 7】

本実施例1に係る仮ウィンドウ構築処理を説明するための図(2)



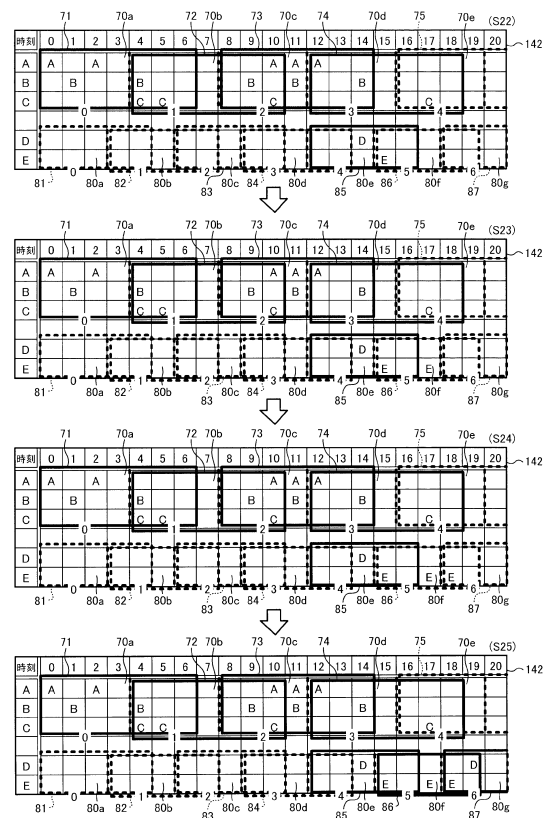
【図 8】

本実施例1に係る仮ウィンドウ構築処理を説明するための図(3)



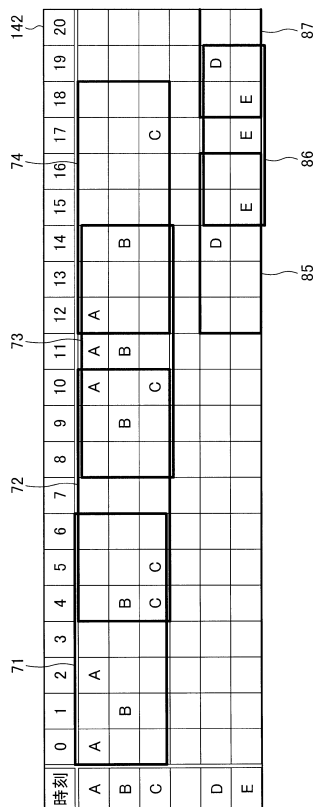
【図 9】

本実施例1に係る仮ウィンドウ構築処理を説明するための図(4)



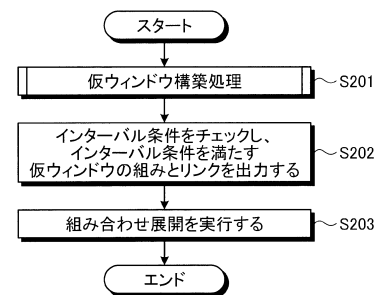
【図 10】

本実施例1に係る仮ウィンドウ構築処理の結果の一例を示す図

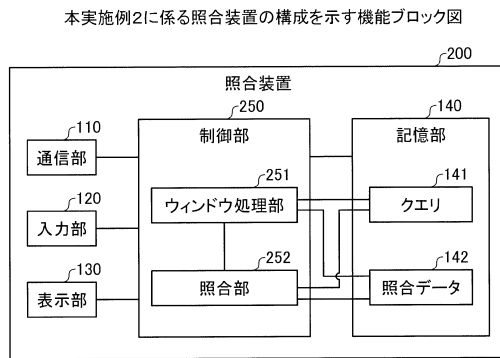


【図 11】

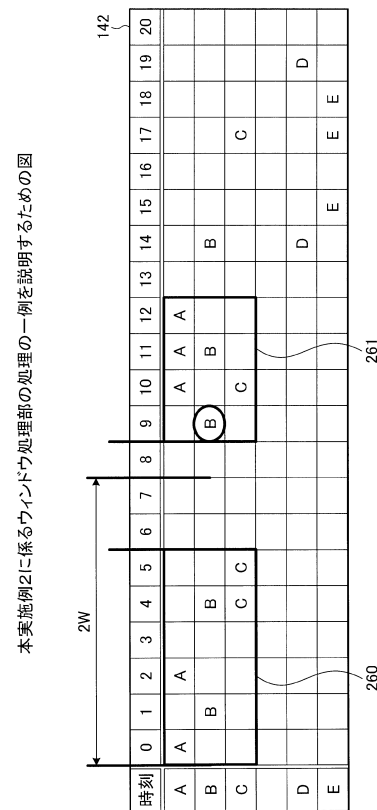
本実施例1に係る照合装置の処理手順を示すフローチャート



【図 1 2】

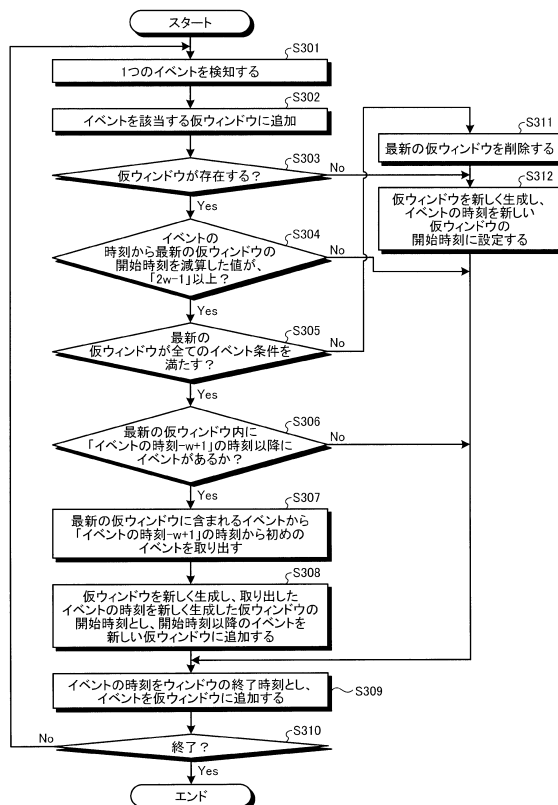


【図 1 3】



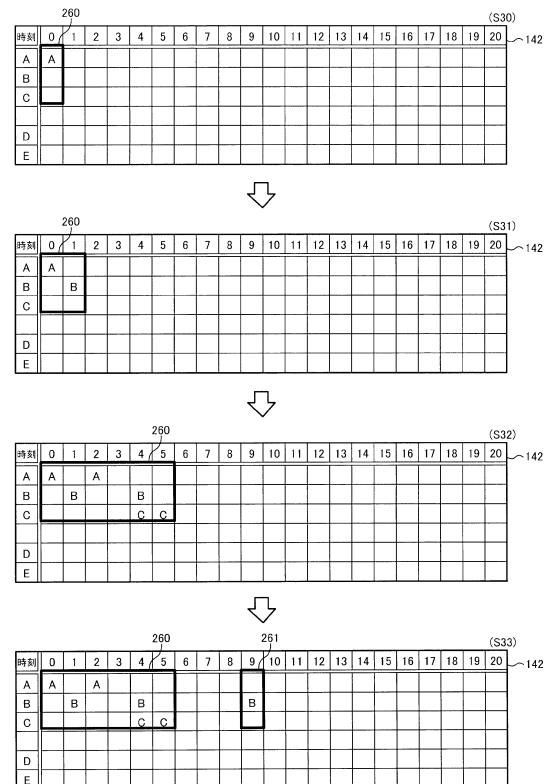
【図 1 4】

本実施例2に係る仮ウィンドウ構築処理の処理手順を示すフローチャート



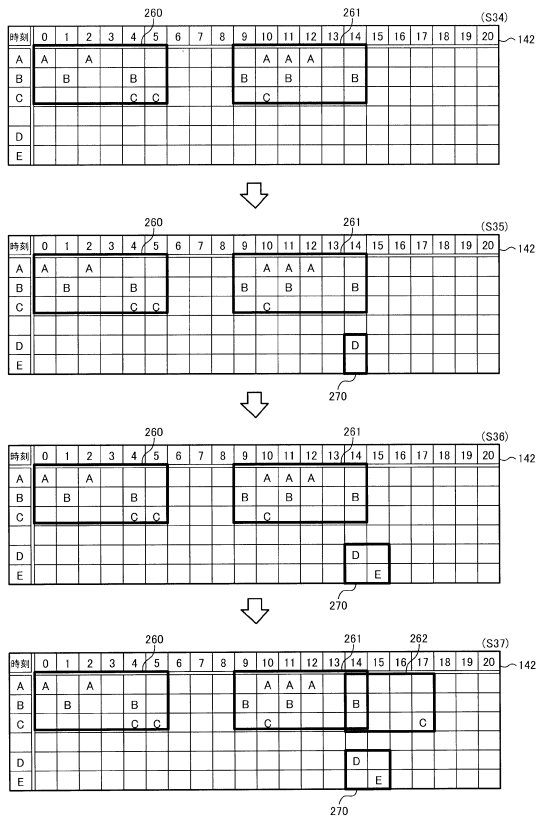
【図 1 5】

本実施例2に係る仮ウィンドウ構築処理を説明するための図(1)



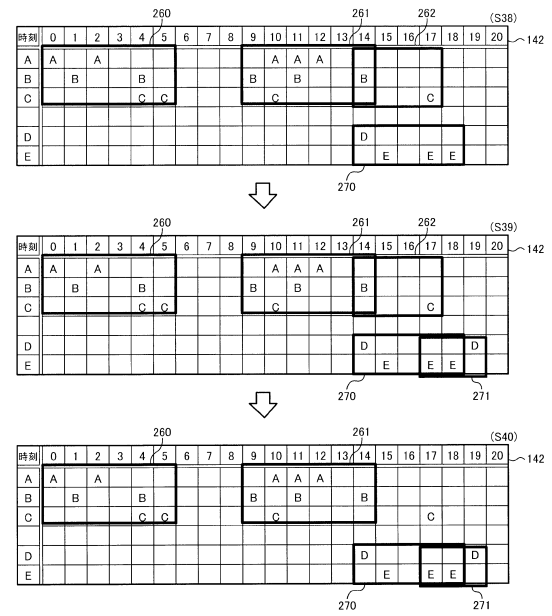
【図 16】

本実施例2に係る仮ウィンドウ構築処理を説明するための図(2)



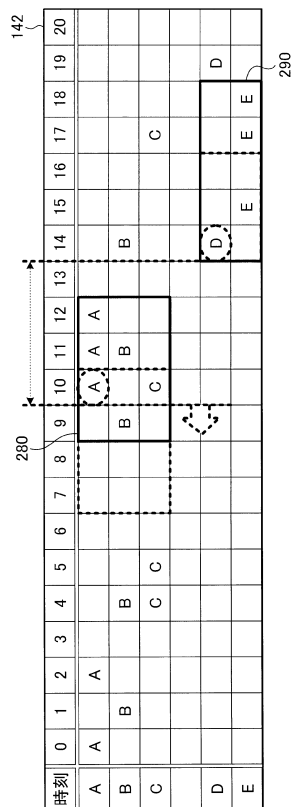
【図 17】

本実施例2に係る仮ウィンドウ構築処理を説明するための図(3)



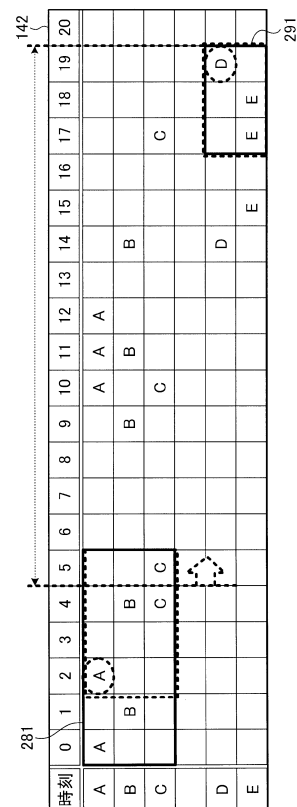
【図 18】

最小インターバル条件を説明するための図



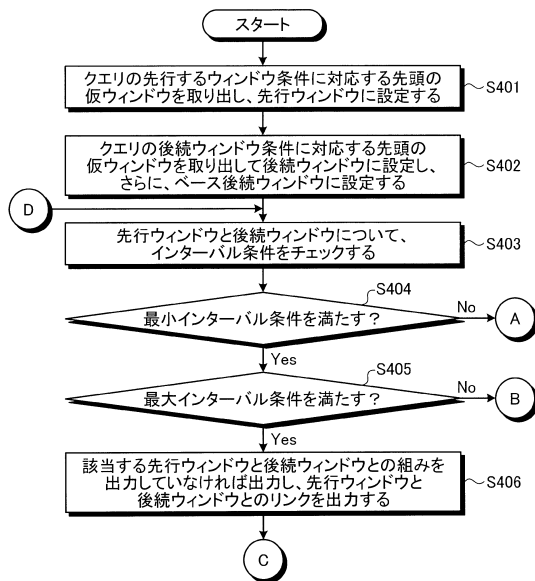
【図 19】

最大インターバル条件を説明するための図



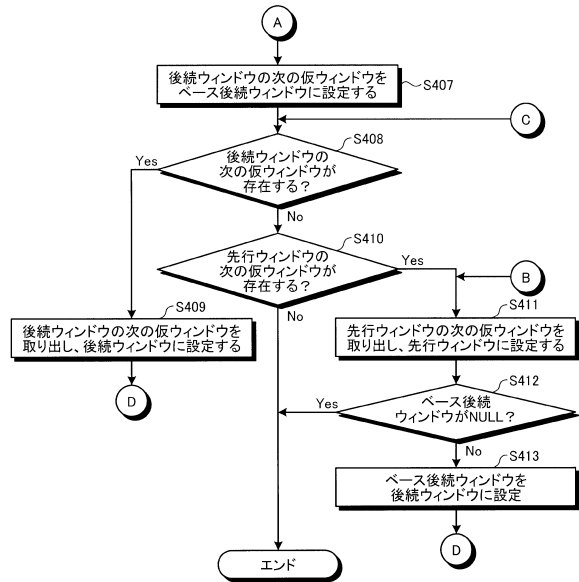
【図 20】

インターバル条件判定処理の処理手順を示すフローチャート(1)



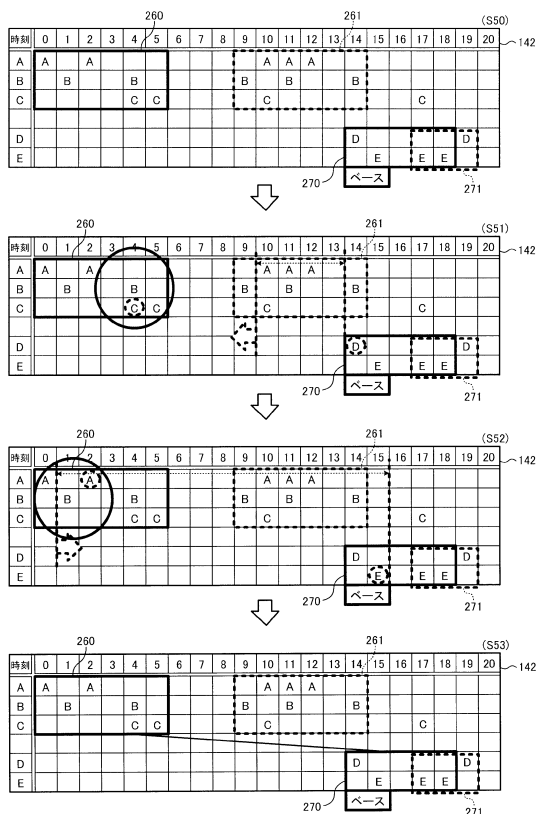
【図 21】

インターバル条件判定処理の処理手順を示すフローチャート(2)



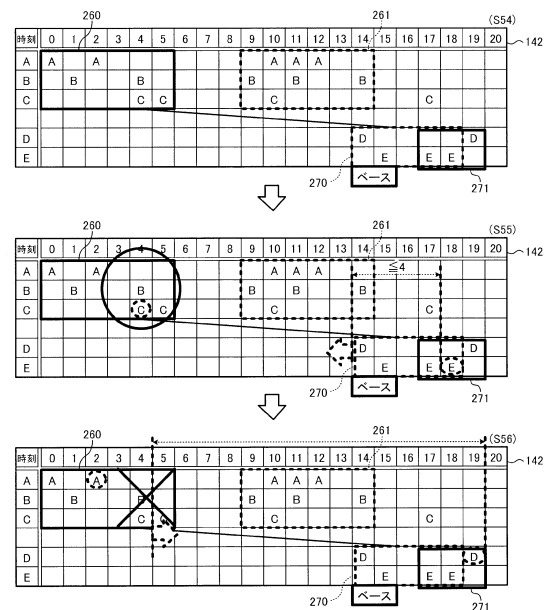
【図 22】

本実施例2に係るインターバル条件判定処理を説明するための図(1)



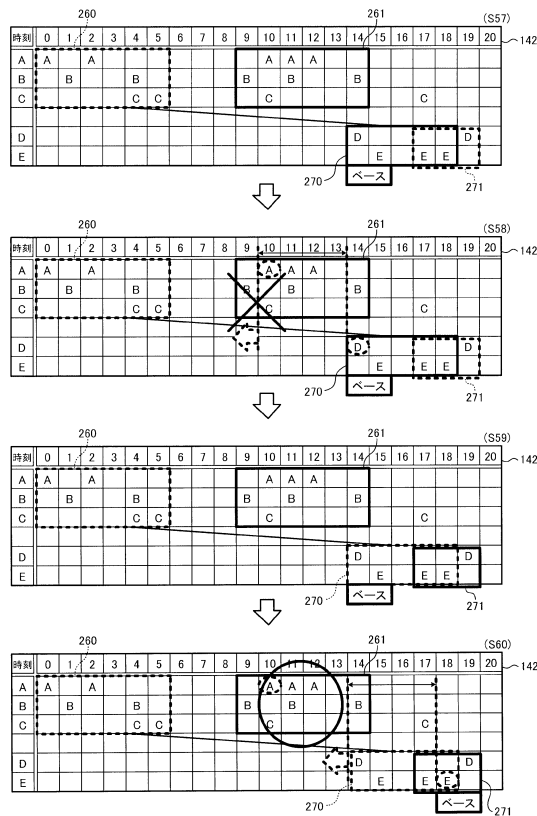
【図 23】

本実施例2に係るインターバル条件判定処理を説明するための図(2)



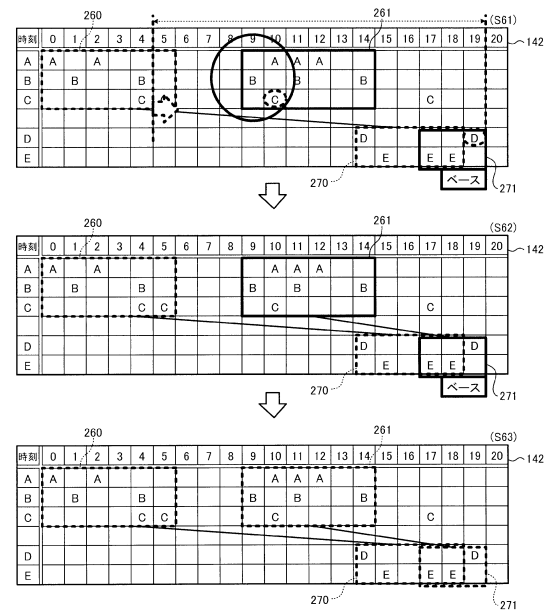
【図 24】

本実施例2に係るインターバル条件判定処理を説明するための図(3)



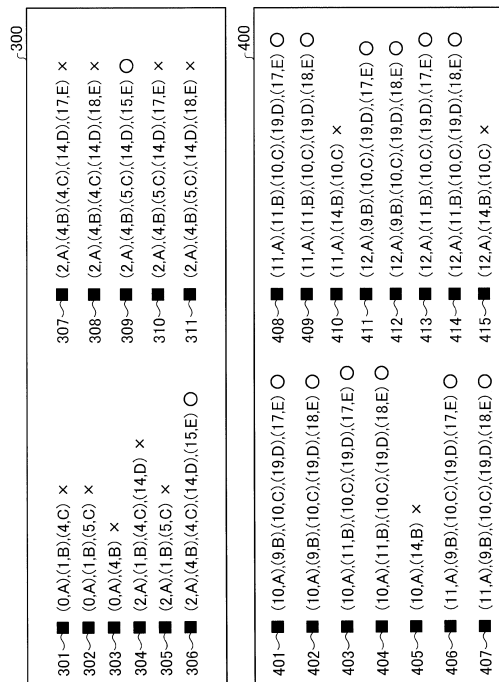
【図 25】

本実施例2に係るインターバル条件判定処理を説明するための図(4)



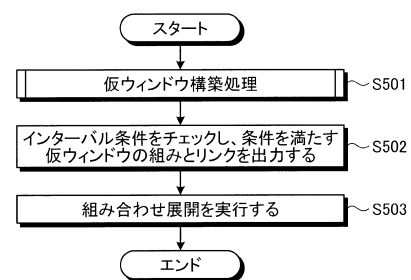
【図 26】

組み合わせ展開の結果の一例を示す図



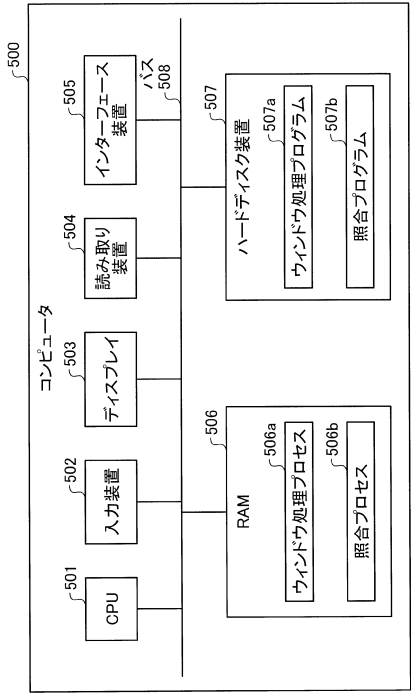
【図 27】

本実施例2に係る照合装置の処理手順を示すフローチャート



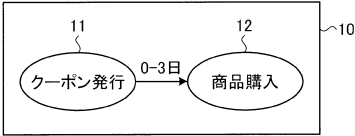
【図 2 8】

照合プログラムを実行するコンピュータの一例を示す図



【図 2 9】

従来技術を説明するためのクエリの一例を示す図(1)



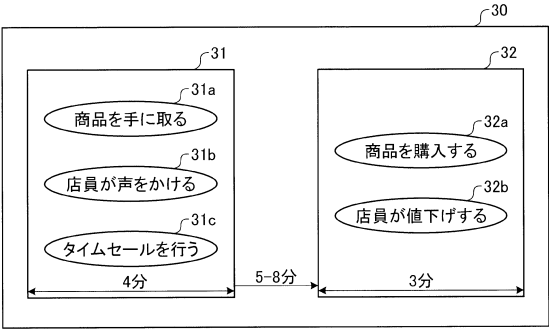
【図 3 0】

従来技術を説明するための照合データの一例を示す図(1)

	1/1	1/2	1/3	1/4	1/5	1/6	1/7	1/8	1/9	1/10	1/11	1/12	1/13
クーポン発行		ク			ク								
商品購入	購	来			来	来							
来店												来	
ダイレクトメール送付				送									
返品													

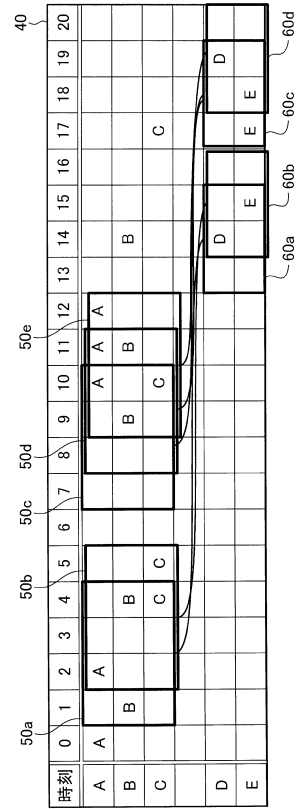
【図 3 1】

ウィンドウ型直列パターンのクエリの一例を示す図



【図 3 2】

従来技術の手法の一例を示す図



フロントページの続き

審査官 樋口 龍弥

(56)参考文献 国際公開第2013/069078 (WO, A1)

特開2003-141158 (JP, A)

特開2003-308333 (JP, A)

特開2006-155404 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 17/30