Open Journal of Social Sciences, 2015, 3, 8-15
Published Online November 2015 in SciRes. http://www.scirp.org/journal/jss
http://dx.doi.org/10.4236/jss.2015.311002

Discovering Complex Incomplete Periodic Patterns through Logical Derivations

Janusz R. Getta[1], Marcin Zimniak[2]

[1]School of Computing and Information Technology, University of Wollongong, Wollongong, Australia
[2]Faculty of Computer Science, TU Chemnitz, Chemnitz, Germany
Email: jrg@uow.edu.au, marcin.zimniak@cs.tu-chemnitz.de

Received October 2015

Abstract

Discovering complex and incomplete periodic patterns in the logs of events is a complicated and time consuming task. This work shows that it is possible to discover complex and incomplete periodic patterns through finding simple patterns first and through logical derivations of complex and incomplete patterns later on. The paper defines a syntax and semantics of a class of periodic patterns that frequently occur in the logs of events. A system of derivation rules proposed in the paper can be used to transform a set of periodic patterns into a logically equivalent set of patterns. The rules are used in the algorithms that derive complex and incomplete periodic patterns. A prototype implementation of the algorithms that discover complex and incomplete periodic patterns in the logs of events is presented.

Keywords

Periodic Pattern, Complex Periodic Pattern, Incomplete Periodic Pattern, Derivation Rules, Data Mining Algorithms

1. Introduction

It is well known that precise estimation of the future workloads can be used to eliminate many performance related problems in database systems [1]. When the future operations on a database are known to a database administrator then it is possible to apply database performance tuning techniques such as indexing, clustering, partitioning of data containers, caching, relocation of data containers to faster persistent storage devices, materialization of the results of expected computations and the others. It is also well known that workload characteristics periodically change due to the repetitive nature of the real world processes implemented by in the database systems. Therefore, information about the workloads recorded in the past can be analyzed in order to anticipate the workloads in the future. The characteristics of database workloads can be quite easily collected at run-time and it can be saved in a form of log, audit trail, processing traces, etc. Unfortunately detection of the periodic changes in the workloads is a difficult problem due to the large amounts of collected data and due to a high level and non-deterministic nature of the periodically repeated processes [2]. A single process may consist of hundreds of events and it may overlap in time with many other processes. Additionally, periodicity of the processes may be incomplete due to the random events as well as due to the slight differences in length of the adjacent periods.

How to cite this paper: Getta, J.R. and Zimniak, M. (2015) Discovering Complex Incomplete Periodic Patterns through Logical Derivations. *Open Journal of Social Sciences*, **3**, 8-15. http://dx.doi.org/10.4236/jss.2015.311002

A problem of finding periodic patterns in the recorded workloads can be solved in a different way from the computationally intensive generation of candidate patterns and their subsequent verification in the logs of events. An important property of periodically repeated processes says that no matter how long and how complex a process is, all its elementary operations are also processed periodically. It leads to an idea where discovery of complex and incomplete periodic patterns can be done through discovery of periodic patterns of individual operations and later on through composition of simple patterns into the complex and incomplete ones. Discovery of simple and complete periodic patterns based on one operation or event can be done in a relatively simple way after partitioning historical data into the subsets that record activities of only one operation or event. The outcomes are the elementary and complete periodic patterns. Then, such homogeneous and complete patterns are "stitched" into the homogeneous and incomplete patterns with certain predefined maximum number of cycles missing. In the next stage, the sets of homogeneous and incomplete periodic patterns are union and all pairs of patterns that satisfy the predefined composition constraints such as minimal length, maximal carrier length are created and composed into complex and incomplete periodic patterns. The procedure is repeated until no new pairs can be found.

To implement a method described above we need a system of derivation rules that transforms the sets of periodic patterns into the logically equivalent sets of patterns and that allows for synthesis of longer patterns and composition of more complicated patterns. The main objective of this paper is to propose a system of derivation rules for complex and incomplete periodic patterns and to show how such system can be used in the algorithms and in a simple prototype implementation that discovers the incomplete periodic patterns from the logs of event.

The paper is organized in the following way. The next section reviews the previous research works related to discovering periodic patterns in historical information. Section 3 defines the concepts of multisets, time units and it shows how a log of events is transformed into a workload trace. Section 4 defines the syntax and semantics of complete and incomplete periodic patterns. A system of derivation rules for incomplete periodic patterns of is proposed in Section 5. Section 6 presents the algorithms that apply the system of derivation rules to find complex and incomplete periodic patterns. Section 7 describes a prototype implementation of the algorithms. Finally, Section 8 concludes the paper.

2. Previous Work

The works on frequent episodes [3] and its extensions on mining complex episodes [4] inspired the works on cyclic patterns. A starting point to many research studies on discovering cyclic patterns is a work [5] that defines the principle concepts of cycle pruning, cycle skipping, cycle elimination heuristics. Discovering periodic patterns in event logs appears to be quite similar to periodicity mining in time series [6] where the long sequences of elementary data items partitioned into a number of ranges and associated with the timestamps are analyzed to find the cyclic trends. The latest works on discovering periodic patterns address the concepts of full periodicity, partial periodicity, perfect and imperfect periodicity [7] and the most recently asynchronous periodicity [8] and [9]. A class of periodic patterns considered in this paper is a variation of periodic patterns earlier investigated in [10] and [11].

3. Workload Trace

Let e be a unique identifier of an event, for example identifier of query processing plan in a database system, or an identifier of flight booking routine in a flight reservation system, etc. A *log of events* is a sequence of pairs $<e_1:t_1>,\ldots,<e_n:t_n>$ where each e_i is a unique identifier of an event, t_i is a timestamp when the processing of an event e_i has started, and $t_1 \leq \ldots \leq t_n$ .A definition of a workload trace is based on a concept of *multiset* [12]. A multiset *M* is a pair $<S, f>$ where *S* is a finite set and $f : S \rightarrow N^+$ is a function such that $\Sigma_{s \in S} f(s) < \infty$ called as a *cardinality* and denoted by $|M|$. Let $M' =< S, f' >$ be a multiset on *S*, then we say that M' is a *submultiset* of *M* and we denote it by $M' \subseteq M$ if $f'(x) \leq f(x)$ for all $x \in S$. In the rest of this paper we shall denote a multiset $<\{e_1,\ldots,e_m\}, f>$ where $f(e_i) = k_i$ for $i = 1,\ldots,m$ as $(e_1^{k_1},\ldots,e_m^{k_m})$. We shall denote an empty multiset $<\varnothing, f>$ as $\varnothing$ and we shall abbreviate a single element multiset (e^k) as e^k.

At a data preparation stage a log of events is transformed into a *workload trace* in the following way. A period of time $<t_{start}, t_{end}>$ over which a log of events is recorded is divided into a sequence U of *n* disjoint time units $<t^i, \tau^i> i = 1,\ldots,n$ where t^i is a timestamp when a time unit starts and τ^i is a length of the unit. All

time units satisfy the following properties: $t_{start} \leq t^1$ and $t^i + \tau^i \leq t^{i+1}$ and $t^n + \tau^n \leq t_{end}$. Let n be the total number of time units in U and let $U[i]$ denotes the i-th time unit in U where i changes from 1 to n. A *workload trace of an event e* is a sequence W_e of n multisets of events such that $W_e[i] = (e^{k_i})$ or $W_e[i] = \varnothing$ for $i = 1,\ldots,n$ and $k_i \geq 1$ equal to the total number of times processing of an event e started in the i-th time unit $U[i]$. Let E be a set of all events whose occurrences are recorded in a log $L(E)$ over time units U and saved in a reduced event table. A *workload trace of a log* $L(E)$ is denoted by W_L and $W_{L[i]} = \uplus_{e \in E} W_{e[i]}$, $\forall i = 1,\ldots,|U|$ *i.e.* it is a multiset union over the respective time units of workload traces of all events included in E.

4. Periodic Patterns

In this work we consider periodic patterns that belong to a wider class of *CRP* periodic patterns defined as a triple $< C, R, P >$ whose individual components have the following meanings.

- A *carrier C* defines a structure of periodically repeated events, computations, queries, etc.
- A *range R* determines a time scope of periodic repetitions of a *carrier* measured in time units, for example from one time unit to another or starting in a given time unit and continuing over several cycles.
- A *periodicity P* determines location of the next cycle of periodic pattern, for example after a given number of time units from the latest cycle with possible delay by one or more time units.

In the previous works, for example in [10], a carrier C is a nonempty, finite sequence of multisets of syntax trees of relational algebra expressions implementing SQL statements, a range R is a pair of the ordinal numbers of time units, and a periodicity P is a total number of time units between two adjacent cycles. In [11] a carrier C is defined as a multiset e^k of an event e, R is defined as a pair of numbers $f : t$ that determine the first and the last repetition of a carrier and P is defined as a pair of numbers $n : x$ that determine the minimal and maximum distance between any two adjacent repetitions of a carrier.

In this work we consider a subclass of *CRP* periodic patterns defined as a triple $< C, f : t, p : g >$ whose individual components have the following meanings.

- A *carrier C* is a nonempty sequence of at least one nonempty multisets of events.
- A *range* $f : t$ is a pair of natural numbers where f determines a location of the first cycle and t is the total number of cycles in the pattern.
- A *periodicity* is a pair of natural numbers $p : g$ where p determines a period of a cycle, *i.e.* a distance between every two adjacent cycles and g determines the longest gap between the adjacent cycles, *i.e.* the maximum total number of adjacent cycles that can be missing from the pattern.
- The values of $f : t$ and $p : g$ must satisfy the conditions $f \geq 1$ and $t \geq 1$ and $p \geq 0$ and $g \geq 0$ and $t \geq g$ and $f + (t-1) * p + |C| - 1 \leq |U|$ and if $t = 1$ then $p = 0$ and $g = 0$ or $g = 1$.
- If $g = 0$ then a periodic pattern is called as a *complete periodic pattern* otherwise if $g \neq 0$ it is called as an *incomplete periodic pattern*.
- It is possible, that $g = t$, *i.e.* the largest total number of missing cycles is equal to the total number of cycles in the pattern. Such a "ghost" periodic pattern can be interpreted as information about planned periodical processing of events, which actually has never been implemented and such that it is still possible in the future.

The following sequence of definitions leads to validation of periodic pattern in a workload trace. Let C be a sequence of multisets where $|C| \leq n$. A *trace of carrier C* spanning over n multisets and starting at a time unit f where $f + |C| - 1 \leq n$ is denoted by $tr(C, f, n)$ and it is defined as sequence of $f - 1$ empty multisets followed by a sequence of multisets C and followed by $n - (f + 1) - |C|$ empty multisets. For example, $trace(e_1 e_2^2, 3, 5)$ is a sequence of multisets $\varnothing\varnothing e_1 e_2^2 \varnothing$.

A *trace of a complete periodic pattern* $< C, f : t, p : 0 >$ over n time units where $f + (t-1) * p + |C| - 1 \leq n$ is denoted by $TRC(< C, f : t, p : 0 >, n)$ and it is defined as a multiset union of traces $tr(C, f + (i-1) * p, n)$ for $\forall i \in \{1,\ldots,t\}$, *i.e.* $TRC(< C, f : t, p : 0 >, n) = \uplus_{i \in \{1,\ldots,t\}} tr(C, f + (i-1) * p, n)$. In the other words, a *trace of a complete periodic pattern* is a union of traces of its carrier over n multisets such that each trace starts at the time units $f, f + p, \ldots, f + (t-1) * p$. For example, a trace of periodic pattern $< e_1 e_2^2, 2 : 2, 1 : 0 >$ over 5 time

units is the following union of sequences of multisets $\varnothing e_1 e_2^2 \varnothing\varnothing \uplus \varnothing\varnothing e_1 e_2^2 \varnothing\varnothing = \varnothing e_1 \left(e_1, e_2^2\right) e_2^2 \varnothing$. In a special case when a value of $t = 1$, *i.e.* when a pattern consists of only one cycle, the values of parameters *p* and *g* must be equal to 0 for example, a trace of periodic pattern $< e_1 e_2^2, 2:1,0:0 >$ over 5 time units is equal to $\varnothing e_1 e_2^2 \varnothing\varnothing$. In another case when the values of parameters *p* and *g* are equal to 0 and a value of parameter $t > 1$, a trace of periodic pattern $< e_1 e_2^2, 2:2,0:0 >$ over 5 time units is equal to $\varnothing e_1^2 e_2^4 \varnothing\varnothing$.

A *trace of an incomplete periodic pattern* $< C, f:t, p:g >$ over *n* time units where $f + (t-1)*p + |C| - 1 \leq n$ is denoted by $TRI(< C, f:t, p:g >, n)$ and it is defined as a multiset union of traces $tr\left(C, f + (i-1)*p, n\right)$ for some of $i \in \{1,\ldots,t\}$ and $\nexists j \in \{1,\ldots,t\}$ such that the traces $tr\left(C, f + (j-1)*p+1, n\right), tr\left(C, f + (j-1)*p+1, n\right), \ldots, tr\left(C, f + (j-1)*p+g, n\right)$ are missing from the multiset union.

An *incomplete periodic pattern* $< C, f:t, p:g >$ is *valid in a workload histogram* W_L recorded over *n* time units if $TRI(< C, f:t, p:g >, n)[i] \subseteq W\_L[i]$ for $\forall i \in \{1,\ldots,t\}$. Of course a complete periodic pattern is valid *in a workload histogram* W_L recorded over *n* time units if $TRC\left(< C, f:t, p:0 >, n\right)[i] \subseteq W_L[i]$ for $\forall i \in \{1,\ldots,t\}$. In the other words a periodic pattern is valid in a workload trace that spans over n time units if some of the elements of its trace over *n* time units are included in the respective elements of a workload W_L and it never happens that the elements of its trace that are not included in a workload form a contiguous sequence longer than *g* elements, *i.e.* the gaps in the cycles are no longer that *g*.

For example, a periodic pattern $< e_1 e_2^2, 2:2,1:0 >$ is valid in a workload trace $e_1^3 e_1 \left(e_1^2, e_2^2\right) e_2^2 \varnothing$ because every element of its trace $\varnothing e_1 \left(e_1, e_2^2\right) e_2^2 \varnothing$ is included in the respective element of the workload trace. The periodic pattern is not valid in a workload trace $e_1^3 e_1 \left(e_1^2, e_2\right) e_2^2 \varnothing$ because an element $\left(e_1, e_2^2\right)$ of its trace is not included in $\left(e_1^2, e_2\right)$. However, an incomplete periodic pattern $< e_1 e_2^2, 2:2,1:1 >$ is valid in a workload trace $e_1^3 e_1 \left(e_1^2, e_2\right) e_2^2 \varnothing$ because a sequence of at most one contiguous elements of its trace is not included in the workload.

5. Derivation Rules

A system of derivation rules presented below allows for creation of new periodic patterns valid in a workload trace W_L that spans over *n* time units from a set of periodic patterns already valid in W_L.

5.1. Discovery Rule

Let *C* be a multiset of events such that $C \subseteq W_L[f]$ for $f \in \{1,\ldots,n\}$. Then, a periodic pattern $< C, f:1,0:0 >$ is valid in W_L. A *discovery rule* creates a single cycle periodic pattern valid in W_L from any non-empty submultiset of an element in a workload trace.

5.2. Incompleteness Rule

If a periodic pattern $< C, f:t, p:g >$ is valid in a workload W_L then a periodic pattern $< C, f:t, p:g' >$ such that $t \geq g' \geq g$ is valid in W_L. An incompleteness rule increases the maximum size of gaps acceptable for a periodic pattern.

5.3. Normalization Rule

If a periodic pattern $< C, f:t, p:g >$ is valid in a workload W_L then a periodic pattern $< C', f':t, p:g >$ such that C' is obtained from *C* through elimination of all *i* leading empty multisets and all trailing empty multisets and such that $f' = f + i$ is valid in W_L. *Normalization rule* allows for elimination of leading and/or trailing empty multisets from a carrier of a periodic pattern.

5.4. Split Rule

If a periodic pattern $< C, f:t, p:g >$ is valid in a workload W_L then the following three cases are possible.

- If $t = 2$ then the periodic patterns $< C, f:1,0:1 >$ and $< C, f+p:1,0:1 >$ are valid in a workload trace W_L.

- If $t > 2$ and then the periodic patterns $< C, f : 1, 0 : 1 >$ and $< C, f + p : t - 1, p : g >$ are valid in a workload trace W_L or the periodic patterns $< C, f : t - 1, p : g >$ and $< C, f + (t-1) * p : 1, 0 : 1 >$ are valid in a workload trace W_L.
- If $t > 3$ and $f_{split} = f + i * p$ for $3 \leq i \leq t - 1$ then the periodic patterns $< C, f : i - 1, p : g >$ and $< C, f_{split} : t - i + 1, p : g >$ are valid in a workload trace W_L.

The first case of a *split rule* divides a pattern that consists of two cycles into two single cycle patterns. The second case "cuts of" a single cycle periodic pattern from either left or right side of a pattern that consist of more than two cycles. Finally, the last case splits a periodic pattern that has more than three cycles into two patterns with more than one cycle.

5.5. Synthesis Rule

If the periodic patterns $< C, f_i : t_i, p_i : g_i >$ and $< C, f_j : t_j, p_j : g_j >$ are valid in a workload trace W_L and $f_i < f_j$ then the following four cases are possible.

- If $t_i = t_j = 1$ and $f_i < f_j$ then a periodic pattern $< C, f_i : 2, f_j - f_i : g_i + g_j >$ is valid in a workload trace W_L.
- If $t_i = 1$ and $t_j \neq 1$ and $f_j - f_i = p_j$ then a periodic pattern $< C, f_i : t_j + 1, p_j : g_i + g_j >$ is valid in a workload trace W_L.
- If $t_j = 1$ and $t_i \neq 1$ and $f_j = f_i + t_i * p_i$ then a periodic pattern $< C, f_i : t_i + 1, p_i : g_i + g_j >$ is valid in a workload trace W_L.
- If $t_j \neq 1$ and $t_i \neq 1$ and $p_i = p_j$ and $f_j = f_i + t_i * p_i$ then a periodic pattern $< C, f_i : t_i + t_j, p_i : g_i + g_j >$ is valid in a workload trace W_L.

In the first case of a *synthesis rule* merges two single cycle pattern into one pattern. In the next two cases a single cycle pattern is added at the left/right end of another pattern. The last case concatenates two patterns such that both of them consist of more than one cycle.

5.6. Decomposition Rule

If a periodic pattern $< C, f : t, p : g >$ is valid in a workload trace W_L then a periodic pattern $< C', f : t, p : g >$ where a carrier C' is obtained by elimination of any multiset from any element of a carrier C is valid in W_L. For example, if a periodic pattern $< e_1 e_2^2, 2 : 5, 1 : 1 >$ is valid in W_L then a periodic pattern $< \varnothing e_2, 2 : 5, 1 : 1 >$ obtained through the elimination e_1 from the first element and e_2 from the second element of a carrier $e_1 e_2^2$ is valid in W_L. Then, a *normalization rule* can be used to eliminate a leading empty multiset from a carrier to get $< e_2, 3 : 5, 1 : 1 >$.

5.7. Composition Rule

If the periodic patterns $< C_i, f_i : t, p : g_i >$ and $< C_j, f_j : t, p : g_j >$ are valid in a workload trace W_L and $f_i \leq f_j$ then a periodic pattern $< C_k, f_i : t, p : g_i + g_j >$ where $C_k = tr\left(C_i, 1, f_j - f_i + |C_j|\right) \uplus tr\left(C_j, f_j - f_i, f_j - f_i + |C_j|\right)$ is valid in W_L. For example, if the periodic patterns $< e_1 e_2^2, 1 : 3, 4 : 1 >$ and $< e_1, 4 : 3, 4 : 1 >$ are valid in a workload trace W_L then a periodic pattern $< e_1 e_2^2 \varnothing e_1, 1 : 3, 4 : 2 >$ is valid in W_L.

6. Discovering Periodic Patterns

A process of discovering periodic patterns in the workload traces is implemented through systematic application of the derivation rules. In each step the rules transform a set of periodic patterns into an equivalent set of patterns. The objectives of the transformations are to find the periodic patterns that have complex carriers, that are long, that have short periods, and that have smallest length of gaps allowed.

We say that a periodic pattern $< C, f : t, p : g >$ is *homogeneous* when its carrier C is a sequence of multisets that contains occurrences of only one and always the same event. The concept of homogenous periodic pattern

given above is generalization of [13]. In the first stage we discover and we transform only *homogenous* periodic patterns. In the second step the sets of *homogeneous* periodic patterns are combined into complex and incomplete patterns.

The process is controlled by the values of parameters $p_{\max}$ that determines the maximal length of any periodic pattern discovered, $g_{\max}$ that determines the longest gap allowed for a periodic patterns, $t_{\min}$, that determines the shortest periodic patterns allowed, and $c_{\max}$ that determines the longest possible length of carrier for any periodic patterns found.

6.1. Discovering Homogeneous Periodic Patterns

A process of finding homogeneous periodic patterns consist of four steps in which the derivation rules are applied to a workload W_L partitioned into n workloads $W_L(e_1),\ldots,W_L(e_n)$. Each, $W_L(e_i)$ contains only occurrences of an event e_i for $i=1,\ldots,n$ extracted from W_L. We repeat the following steps for each $W_L(e_i)$.

Step 1

We start from the application of *discovery rule* to $W_{L(e_i)}$ to create the single cycle and complete periodic patterns like $<e_i, f:1,0:0>$.

Step 2

For each $p=1,\ldots,p_{\max}$ we use a *synthesis rule* to find the longest complete periodic patterns consistent with a form $<e_i, f:t, p:0>$. Then, for all patterns that have the same values of parameters *f* and *t* we apply a composition rule to create the patterns like $<e_i^k, f:t, p:0>$.

Step 3

The *split* and *composition rules* are used to transform the patterns like $<e_i^k, f:t, p:0>$ and such that $t<t_{\min}$ into single cycle pattern $<C, f:1,0:0>$ where a carrier *C* is equal to is a sequence of e_i^k separated with $p-1$ empty sets and repeated $t-1$ times and ended with e_i^k. A *synthesis rule* is applied to assemble the single cycle complete periodic patterns into multicycle complete patterns. For example, the patterns $<e,1:2,2;0>$, $<e,6:2,2;0>$, $<e,9:2,2:0>$ are first grouped into $<e\varnothing e,1;1,0:0>$, $<e\varnothing e,6:1,0:0>$, and $<e\varnothing e,9:1,0:0>$ and later on are synthesized into $<e\varnothing e,1:3,5:0>$.

Step 4

Finally, we apply a *synthesis rule* to the periodic patterns created so far in order to create longer and incomplete patterns with the length of gaps limited by a value of parameter $g_{\max}$. The rule is iteratively applied for $g=1,\ldots,g_{\max}$ the pairs of periodic patterns that have the same carrier. A pair of periodic patterns that can be synthesized into a longer and incomplete one is replaced with the results of synthesis rule. At the end of this process we obtain *n* sets of homogenous and incomplete periodic patterns $H_1,\ldots,H_n$ one set per each one of *n* events $e_1,\ldots,e_n$ recorded in a workload trace W_L.

Complexity of the algorithm depends on the length *n* of a workload trace W_L, on the maximum period size $p_{\max}$ and on the total number *h* of homogeneous patterns discovered in the first three steps. The complexity of the first there steps is $O(p_{\max} * n)$ the complexity of Step 4 is $O(h^2)$.

6.2. Discovering Complex Periodic Patterns

A process of finding complex periodic patterns initially applies a composition rule to the sets of homogeneous and incomplete patterns obtained in the previous steps. Then, a composition rule is applied to the results of compositions until no new complex and incomplete patterns can be derived. The process is limited by a threshold value $c_{\max}$ that determines the longest possible carrier of periodic pattern obtained from application of a composition rule, by a threshold value. The process is also limited by the maximal allowed length of "gaps" $g_{\max}$ and the minimal length of the results of composition $t_{\min}$. The following two steps are repeated until no more new periodic patterns can be created with a composition rule.

Step 1

The sets $H_1,\ldots,H_n$ of homogenous and incomplete periodic patterns are assembled into $G = H_1 \cup \ldots \cup H_n$.

Step 2

Next, we find in *G* all pairs of periodic patterns $<C_i, f_i:t_i, p:g_i>$ and $<C_j, f_j:t_j, p:g_j>$ such that $f_i <= f_j$ and $\max(f_i+|C_i|, f_j+|C_j|)-f_i \le c_{\max}$ and $\min(t_i,t_j) \ge t_{\min}$ and $g_i+g_j \le g_{\max}$. Then if $t_i \ne t_j$ we apply a *split rule* to adjust the length of the patterns. A new pattern obtained from a split is appended to *G*. Additionally each periodic pattern must be a member of at most one pair. If a periodic pattern is involved in

more than one pair then we pick a pair that maximizes the length of the patterns obtained from the future composition. It may happen that it is impossible to find any pair of periodic patterns that satisfy the conditions listed above. Then, we end the process of creating complex and incomplete periodic patterns.

Step 3

For each pair of periodic patterns $< C_i, f_i : t, p : g_i >$ and $< C_j, f_j : t, p : g_j >$ found in the previous step we apply a *composition rule* to create a new periodic pattern. Then, we replace the composed patterns in G with the result of composition. When all pairs are processed we return to Step 2.

The complexity of the algorithm is equal to $O\left(k * h^2\right)$ where *h* is the total number of patterns included in an input set *G* and *k* is the total number of patterns obtained from split in Step 2 of the algorithm.

7. Prototype Implementation

The algorithms described in the previous section are implemented in an environment of a commercial relational database management system. We save an audit trail from processing of a sequence of SQL statements against a sample TPC-H benchmark database. Then, we apply EXPLAIN PLAN statement to transform each SQL statement into an expression of extended relational algebra. The computations of individual relational algebra operations are considered as individual events in a log. Suchlog of events together with the times tamps is transformed into a workload trace where the individual operations are grouped within predefined time units. A synthetic workload generator is used to implement periodic processing of sequences of SQL statements. A number of casually processed SQL statements are incorporated into the workload to evaluate an impact of randomly processed statements on discovery of periodic patterns. All software is implemented in SQL embedded into a host language of a database management system used.

Application of a synthetic workload generator allows for precise estimation of the quality of results obtained from the algorithms through the comparison of pre-programmed iterative processing of SQL statements with the periodic patterns obtained from the algorithms. The algorithms are applied several times to the same log of events partitioned each time into the time units of different size. In all cases when a period of iterative processing of SQL statement is a multiplicity of the length of time units the algorithms return almost perfect results and are able to precisely detect the expected patterns. In the cases when a period of iteratively processed sequence of SQL statements was not consistent with the length of time units the algorithm return a larger number of shorter and simpler periodic patterns than expected, however the results were still within the acceptable quality range. Quality of the results also strongly depends on a careful choice of the parameters that restrict carrier length, period length etc. Selection of too large or too small configuration parameters contributes to identification of accidental patterns not planned within a synthetic workload.

8. Summary and Future Work

This work describes a new approach to discovery of complex and incomplete periodic patterns in the logs of events. The method is based on an idea that it is possible to create complex and incomplete periodic patterns through systematic discovery, transformation, and composition of the simpler patterns. The new approach requires a system of derivation rules for transformation of periodic patterns into the equivalent ones. Such system of rules is defined in the paper. We show how the rules can be applied in the algorithms that process a workload trace obtained from a log of events into initially simple and homogeneous patterns and later on into complex and incomplete ones. A prototype implementation of the algorithms is used to discover the periodic patterns among the events equivalent to the computations of individual extended relational algebra operations implementing SQL statements processed by a relational database system.

A number of interesting research problems remain to be solved. An important problem is an appropriate choice of time units used to partition a log of events into multisets of events in a workload trace due an observation that quality of the discovered patterns depends on the length of time units. The next interesting problem includes investigations on the other system of derivation rules that may lead to more efficient implementations. An interesting task is more efficient and more general implementation of the algorithms that can be used to discover periodic patterns in many other domains. Finally, a class of periodic patterns considered in the paper can be extended on the patterns with more sophisticated specification of period parameter allowing for slight variations from cycle to cycle.

References

[1] Bruno, N. (2011) Automated Physical Database Design and Tuning. CRC Press Taylor and Francis Group, Boca Raton.

[2] Van der Alst, W.M.P. (2011) Process Mining Discovery, Conformance and Enhancement of Business Processes. Springer, Berlin.

[3] Mannila, H., Toivonen, H. and Verkamo, A.I. (1997) Discovery of Frequent Episodes in Event Sequences. *Data Mining and Knowledge Discovery*, **1**, 259-289. http://dx.doi.org/10.1023/A:1009748302351

[4] Wojciechowski, M. (2000) Discovering Frequent Episodes in Sequences of Complex Events. *Proceedings of Enlarged Fourth East-European Conference on Advances in Databases and Information Systems* (*ADBIS-DASFAA*), 205-214.

[5] Ozden, B., Ramaswamy, S. and Silberschatz, A. (1998) Cyclic Association Rules. *Proceedings of the Fourteenth International Conference on Data Engineering*, 412-421. http://dx.doi.org/10.1109/ICDE.1998.655804

[6] Rasheeed, F., Alshalalfa, M. and Alhajj, R. (2011) Efficient Periodicity Mining in Time Series Databases Using Suffix Trees. *IEEE Transactions on Knowledge and Data Engineering*, **23**, 79-94. http://dx.doi.org/10.1109/TKDE.2010.76

[7] Huang, K.-Y. and Chang, C.-H. (2004) Asynchronous Periodic Patterns Mining in Temporal Databases. *Databases and Applications*, IASTED/ACTA Press, 43-48.

[8] Yang, J., Wang, W. and Yu, P.S. (2003) Mining Asynchronous Periodic Patterns in Time Series Data. *IEEE Transactions on Knowledge and Data Engineering*, **15**, 613-628. http://dx.doi.org/10.1109/TKDE.2003.1198394

[9] Yeh, J.-S., Lin, S.-C. and Hu, S.-C. (2013) Novel Algorithms for Asynchronous Periodic Pattern Mining Based on 2-d Linked List. *International Journal of Database Theory and Application*, **5**, 33-43.

[10] Zimniak, M., Getta, J. and Benn, W. (2014) Deriving Composite Periodic Patterns from Database Audit Trails. *The* 6*th Asian Conference on Intelligent Information and Database Systems*, 310-321. http://dx.doi.org/10.1007/978-3-319-05476-6_32

[11] Getta, J., Zimniak, M. and Benn, W. (2014) Mining Periodic Patterns from Nested Event Logs. *The* 14*th IEEE International Conference on Computer and Information Technology*, Xi'an, 160-167. http://dx.doi.org/10.1109/cit.2014.27

[12] Simovici, D.A. and Djeraba, C. (2008) Mathematical Tools for Data Mining: Set Theory, Partial Orders, Combinatorics. *Advanced Information and Knowledge Processing*, Springer.

[13] Zimniak, M., Getta, J.R. and Benn, W. (2014) Discovering Periodic Patterns in System Logs. *Proceedings of the LWA* 2014 *Workshops*: *KDML*, *IR*, *FGWM*, Aachen, 156-161.