On Measuring Similarity for Sequences of Itemsets

Elias Egho, Chedy Raïssi, Toon Calders, Nicolas Jay, Amedeo Napoli

On Measuring Similarity for Sequences of Itemsets

Elias Egho , Chedy Raïssi , Toon Calders , Thomas Bourquard , Nicolas Jay , Amedeo Napoli

RESEARCH REPORT

N° 8086

October 2012

Project-Teams Orpailleur

ISSN 0249-6399 ISRN INRIA/RR--8086--FR+ENG

On Measuring Similarity for Sequences of Itemsets

Elias Egho [*], Chedy Raïssi [†], Toon Calders [‡], Thomas Bourquard [†], Nicolas Jay [*], Amedeo Napoli [*]

Project-Teams Orpailleur

Research Report n° 8086 — October 2012 — 19 pages

Abstract: Computing the similarity between sequences is a very important challenge for many different data mining tasks. There is a plethora of similarity measures for sequences in the literature, most of them being designed for sequences of items. In this work, we study the problem of measuring the similarity ratio between sequences of itemsets. We present new combinatorial results for efficiently counting distinct and common subsequences. These theoretical results are the cornerstone for an effective dynamic programming approach to deal with this problem. Experiments are realized on biological protein and synthetic dataset, showing that our measure of similarity produces competitive scores and indicates that our method is relevant for real-world sequential data analysis.

Key-words: Similarity measure, Clustering, Sequence mining

[*] LORIA, Vandoeuvre-les-Nancy, France
[†] INRIA, Nancy Grand Est, France
[‡] Université Libre de Bruxelles

RESEARCH CENTRE
NANCY – GRAND EST

615 rue du Jardin Botanique
CS20101
54603 Villers-lès-Nancy Cedex

Vers une mesure de similarité pour les séquences complexes

Résumé : Le calcul de la similarité entre les séquences est un défi très important pour de nombreuses tâches d'exploration de données. Il existe une pléthore de mesures de similarité de séquences dans la littérature, la plupart d'entre eux étant conçu pour les séquences d'items. Dans ce travail, nous étudions le problème de la mesure du taux de similitude entre les séquences d'itemsets. Nous vous présentons de nouveaux résultats combinatoires pour compter efficacement touts les sous-séquences distinctes et communes. Ces résultats théoriques sont la pierre angulaire d'une approche de programmation dynamique efficace pour traiter ce problème. Les expériences sont réalisées sur les données biologiques et ur les jeux de données synthétiques, montrant que notre mesure de similarité produit scores compétitifs et indique que notre méthode est pertinente pour le monde réel l'analyse de données séquentielles.

Mots-clés : Mesure de imilarité, Feuille de données, Feuille de données séquentielles

1 Introduction

Sequential data is widely present and used in many applications such as matching of time series in databases [1], DNA or amino-acids protein sequences analysis [2, 3], as well as web log analysis [4], and music sequences matching [5]. Consequently, analyzing sequential data has become an important data mining and machine learning task with a special focus on the examination of pairwise relationships between sequences. For example, some clustering and kernel-based learning methods depend on computing distances or similarity ratios between sequences [6, 7]. However, for a large part of literature, similarity measures on sequential data remains limited to *simple sequences*, which are ordered lists of items (i.e., symbols) [8, 9, 10, 11]. By contrast, in modern life sciences [12], sequential data sets are represented as ordered lists of itemsets (i.e., sets of symbols). This singularity is in itself a challenge as it implies to carefully take into account complex combinatorial aspects to compute similarities between sequences.

In this study, we focus on the notion of common subsequences as a mean to define a distance or similarity ratio between a pair of sequences composed of a list of itemsets. We make two significant contributions. Firstly, we start by answering two fundamental theoretical open problems: (i) given a sequence of itemsets, can we count, *without enumerating*, the number of distinct subsequences? (ii) for a pair of sequences, can we *efficiently* count the number of common subsequences? We present two theorems that positively answer these questions. Secondly, we discuss and present a dynamic programming algorithm for counting all common subsequences (ACS) between two given sequences. This dynamic programming algorithm allows us to define in a simple and intuitive manner our similarity measure which is a ratio between the number of common subsequences from two sequences S and T divided by the maximal number of distinct subsequences.

We believe that the results reported in this work are a useful contribution with direct practical applications to different discriminative approaches, and in particular kernel methods, as new complex sequence kernels can be devised based on the theoretical results provided in this work. Moreover, the method is completely general in that it can be used (with slight modifications) for a broad spectrum of sequence-based classification or clustering problems. We report extensive empirical study on synthetic datasets and a qualitative experiment on the comparison of the 3D-structures of two sets of proteins, namely haemoglobin and myoglobin protein families.

The rest of the report is organized as follows. Section 2 reviews the related work. Section 3 briefly reviews the preliminaries needed in our development. Section 4 and 5 introduces our new combinatorial results. Two experimental studies are reported in Section 6 and we conclude our work in Section 7.

2 Related Work

Since Levenshtein [8] proposed the edit distance measure to compute a distance between strings, many studies focused on developing efficient approaches for sequences similarities. The Levenshtein distance between strings s and t is defined as the minimum number of edit operations needed to transform s into t. The edit operations are either an insertion, a deletion, or a substitution of a symbol. Many other approaches are built on this seminal result but with notable differences like weighting the symbols or the edit operations [9], or using stochastic processes [13]. For time series, another important approach is the Dynamic Time Warping (DTW) technique for finding an optimal alignment between two sequences [10]. Intuitively, the sequences are warped in a nonlinear fashion to match each other. DTW technique had a huge impact and has been used to compare multiple patterns in automatic speech recognition to cope with different speaking speeds [14]. Zaki et al. [15] and Vlachos et al. [16] followed a radically different approach by developing longest common subsequences approaches for the comparison and similarity measure. However, the common information shared between two sequences is more than the longest common subsequence. In fact counting all possible common information between sequences provides a good idea about the similarity relationship between the sequences and their overall *complexity.* In addition, the common subsequences problem is related to the problem of counting the number of all distinct common subsequences between two sequences. Wang et al. [11] studied all common subsequences (ACS) as a similarity measure between two sequences of items. Elzinga et al. [17] used a dynamic programming algorithm to count a distinct common subsequences between two sequences of items.

In this work, we extend and generalize the previous work of [11, 17] for the complex structure of sequence of itemsets.

3 Preliminaries

Definition 1 (Sequence) *Let $\mathcal{I}$ be a finite set of* items. *An* itemset X *is a non-empty subset of $\mathcal{I}$. A* sequence S *over $\mathcal{I}$ is an ordered list $\langle X_1 \cdots X_n \rangle$, where X_i ($1 \leq i \leq n$, $n \in N$) is an itemset. S^l denotes the l-prefix $\langle X_1, \ldots, X_l \rangle$ of sequence S with $1 \leq l \leq n$. The j-th itemset X_j of sequence S is denoted $S[j]$ with $1 \leq j \leq n$.*

Definition 2 (Subsequence) *A sequence $T = \langle Y_1 \cdots Y_m \rangle$ is a **subsequence** of $S = \langle X_1 \ldots X_n \rangle$, denoted by $T \preceq S$, if there exist indices $1 \leq i_1 < i_2 < \cdots < i_m \leq n$ such that $Y_j \subseteq X_{i_j}$ for all $j = 1 \ldots m$ and $m \leq n$. S is said to be a **supersequence** of T.*

*$\varphi(S)$ denotes the **set of all subsequences** of a given sequence S and $\phi(S) = |\varphi(S)|$. For two sequences S and T, $\varphi(S,T)$ denotes the set of **all common subsequences** between two sequences S and T: $\varphi(S,T) = \varphi(S) \cap \varphi(T)$ and $\phi(S,T) = |\varphi(S,T)|$.*

We now define the following similarity measure between two sequences of itemsets S and T.

Definition 3 *The **similarity between two sequences** S **and** T, denoted sim(S, T) is defined as the number of common subsequences divided by the maximal number of subsequences of S and T; that is:*

$$sim(S,T) = \frac{\phi(S,T)}{\max\{\phi(S), \phi(T)\}}$$

From this point on, the rest of the report, up to the experiments Section, will be dedicated to efficient techniques for computing $\phi(S)$ and $\phi(S,T)$, as these form the backbone of our new similarity measure. As the explanation and the proofs of correctness of these computations involve complicated manipulations of sequences, we introduce the following operators on sets of sequences.

Definition 4 (Concatenation) *Let $S = \langle X_1 \cdots X_n \rangle$ and $T = \langle Y_1 \cdots Y_m \rangle$ be two sequences. The **concatenation of** S **and** T, denoted $S \circ T$, is the sequence $\langle X_1 \cdots X_n \ Y_1 \cdots Y_m \rangle$.*

Given two sets of sequences $\mathcal{S}$ and $\mathcal{T}$, $\mathcal{S} \circ \mathcal{T} = \{S \circ T \mid S \in \mathcal{S}, T \in \mathcal{T}\}$.

For ease of notation we will denote a non-empty itemset X with the singleton sequence $\langle X \rangle$, the empty set $\emptyset$ with the empty sequence $\langle \rangle$. As usual, the powerset of an itemset X will be denoted by $\mathcal{P}(X)$, and $\mathcal{P}_{\geq 1}(X)$ denotes all nonempty subsets of X; that is, $\mathcal{P}_{\geq 1}(X) = \mathcal{P}(X) \backslash \{\emptyset\}$.

Example 1 *We use the sequence database $\mathcal{D}_{ex}$ in Table 1 as a running example. It contains* 4 *data sequences over the set of items $\mathcal{I} = \{a, b, c, d\}$. Sequence $\langle \{a\}\{b\}\{c, d\} \rangle$ is a subsequence of*

$\mathcal{D}_{ex} =$

S_1	$\langle\{a\}\{a,b\}\{e\}\{c,d\}\{b,d\}\rangle$
S_2	$\langle\{a\}\{b,c,d\}\{a,d\}\rangle$
S_3	$\langle\{a\}\{b,d\}\{c\}\{a,d\}\rangle$
S_4	$\langle\{a\}\{a,b,d\}\{a,b,c\}\{b,d\}\rangle$

Table 1: The sequence database used as the running example

$S_1 = \langle\{a\}\{a,b\}\{e\}\{c,d\}\{b,d\}\rangle$*. The* 3*-prefix of* S_1*, denoted* S_1^3*, is* $\langle\{a\}\{a,b\}\{e\}\rangle$ *and* $S_1[2]$*, the second itemset in sequence* S_1*, is* $\{a,b\}$*.*

The set of all subsequences of S_4^2 *is*

$$\begin{aligned}\varphi(S_4^2) \quad = \quad & \{\langle\rangle, \langle\{a\}\rangle, \langle\{b\}\rangle, \langle\{d\}\rangle, \langle\{a,b\}\rangle, \langle\{a,d\}\rangle, \langle\{b,d\}\rangle, \langle\{a,b,d\}\rangle, \langle\{a\}\{a\}\rangle, \langle\{a\}\{a,b\}\rangle, \\ & \langle\{a\}\{d\}\rangle, \langle\{a\}\{b\}\rangle, \langle\{a\}\{a,d\}\rangle, \langle\{a\}\{b,d\}\rangle, \langle\{a\}\{a,b,d\}\rangle\}\end{aligned}$$

Hence, $\phi(S_4^2) = 15$*.*

The concatenation of the sequence S_4^2 *with the itemset* $\{a,b,c\}$*, denoted as* $S_4^2 \circ \{a,b,c\}$*, is the sequence* $\langle\{a\}\{a,b,d\}\{a,b,c\}\rangle$*.* $\langle\{a,b\}\rangle \circ \mathcal{P}_{\geq 1}(\{c,d\})$ *denotes the set of sequences* $\{\langle\{a,b\}\{c\}\rangle, \langle\{a,b\}\{d\}\rangle, \langle\{a,b\}\{c,d\}\rangle\}$*.*

The set of all common subsequences of S_1^4 *and* S_2^3 *is*

$$\begin{aligned}\varphi(S_1^4, S_2^3) \quad = \quad & \{\langle\rangle, \langle\{a\}\rangle, \langle\{b\}\rangle, \langle\{d\}\rangle, \langle\{c\}\rangle, \langle\{c,d\}\rangle, \langle\{a\}\{a\}\rangle, \langle\{a\}\{b\}\rangle, \langle\{a\}\{c\}\rangle, \langle\{a\}\{d\}\rangle, \\ & \langle\{a\}\{c,d\}\rangle, \langle\{b\}\{d\}\rangle, \langle\{a\}\{b\}\{d\}\rangle\}\end{aligned}$$

4 Counting All Distinct Subsequences

In this section, we present an efficient technique *to count* the number $\phi(S)$ of all distinct subsequences for a given sequence S. We emphasize the fact that the studied sequences are not *simple sequences* that are discussed in length in the bio-informatics literature for which efficient approaches exist, but rather an ordered list of itemsets. As we will show, this is a highly non-trivial extension as it implies new combinatorial aspects. In this section, we introduce a dynamic programming algorithm to count the number of distinct subsequences for a given sequence S. Before stating the main result, we present the intuition behind the proposed dynamic programming approach. Suppose that we extend a given sequence $S = \langle X_1 \cdots X_n\rangle$ with an itemset Y and we observe the relation between $\phi(S)$ and $\phi(S \circ Y)$. Two cases may appear:

1. Y is disjoint with any itemset in S; i.e., for all $i = 1 \ldots n$, $Y \cap X_i = \emptyset$, then the number of distinct subsequences of $S \circ Y$ equals $|\varphi(S)| \cdot 2^{|Y|}$, since for all $T \in \varphi(S)$, and $Y_1, Y_2 \in \mathcal{P}(Y)$, $T \circ Y_1 \neq T \circ Y_2$. For example, $\phi(\langle\{a,b\}\{c\}\rangle \circ \{d,e\}) = 8 \cdot 2^2 = 32$.

2. At least one item of Y appears in an itemset of S; i.e., $\exists i \in [1,n] : Y \cap X_i \neq \emptyset$. In this case, $|\varphi(S \circ X)|$ is smaller than $|\varphi(S)| \cdot 2^{|Y|}$, because not every combination of a sequence in $\varphi(S)$ with an element from the power set of Y results in a unique subsequence. For example, if $S = \langle\{a,b\}\rangle$ and $Y = \{a,b\}$, the sequence $\langle\{a\}\rangle$ can be obtained by either extending the empty sequence $\langle\rangle$ with the itemset {a}, or by extending $\langle\{a\}\rangle$ with $\emptyset$.

   Therefore, we need to define a method to *remove the repetitions from the count.* Formally, $|\varphi(S \circ Y)| = |\varphi(S)| \cdot 2^{|Y|} - R(S,Y)$ where $R(S,Y)$ represents a *correction term* that equals the number of repetitions of subsequences that should be suppressed for a given S concatenated with the itemset Y.

We illustrate the second case with an example.

Example 2 *Consider sequence S_4 from our toy data set. $S_4^2 = \langle\{a\}\{a,b,d\}\rangle$ is the 2-prefix of S_4. Recall from Example 1 that the total number of subsequences of S_4^2 is $\phi(S_4^2) = 15$. Now suppose that we extend this sequence S_4^2 with the itemset $Y = \{a,b,c\}$. Clearly, concatenating each sequence from $\varphi(S_4^2)$ with each element in the power set of $\{a,b,c\}$ will generate some subsequences multiple times. For instance, the subsequence $\langle\{a\}\{b\}\rangle$ is generated twice: $\langle\{a\}\rangle\circ\{b\}$ and $\langle\{a\}\{b\}\rangle\circ\emptyset$. The same applies to other subsequences $\langle\{a\}\rangle$, $\langle\{b\}\rangle$, $\langle\{a,b\}\rangle$, $\langle\{a\}\{a\}\rangle$ and $\langle\{a\}\{ab\}\rangle$. Thus, making a total of* 6 *subsequences that are counted twice. In this case, the correct number of distinct subsequences for $S_4^2 \circ Y = \langle\{a\}\{a,b,d\}\{a,b,c\}\rangle$ is $|\varphi(S_4^2)| \cdot 2^{|Y|} - R(S_4^2, Y) = 15 \cdot 2^3 - 6 = 114$.*

As illustrated by the above example, the actual challenge is the computation of the value of the *correction term* $R(S,Y)$. The general idea is to compensate the repeated concatenation of subsequences from S by the power set of Y. The problem occurs with sequences in $\varphi(S)\circ\mathcal{P}_{\geq 1}(Y)$ that are already in $\varphi(S)$. Suppose T is such a sequence, then T must be decomposable as $T'\circ Y'$, where $T' \in \varphi(S^i)$ for some $i = 0\ldots n-1$, and $Y' \subseteq Y \cap S[j]$, for some $j \in i+1\ldots n$. The following definition introduces the *position set* that will capture those positions in S that generate duplicates:

Definition 5 (Position set) *Given an itemset Y and a sequence $S = \langle X_1 \cdots X_n\rangle$, $L(S,Y)$ is the set of all* ***maximal positions*** *where the itemset Y has a maximal intersection with the different itemsets X_i, $i = 1\ldots n$. Formally,*

$$L(S,Y) = \{i \mid Y \cap X_i \neq \emptyset,\ \textit{and}\ (\nexists\ j\ ;\ j > i\ \textit{and}\ \ Y \cap X_i \subseteq Y \cap X_j)\} \tag{1}$$

Notice that if there are multiple positions that generate the same duplicates, we only consider the last one.

Example 3 *Let $S_4 = \langle\{a\}\{a,b,d\}\{a,b,c\}\{b,d\}\rangle$ be the studied sequence.*
$L(\langle\rangle, \{a\})$=$\emptyset$, $L(\langle\{a\}\rangle, \{a,b,d\})$=$\{1\}$, $L(\langle\{a\}\{a,b,d\}\rangle, \{abc\})$=$\{2\}$,
$L(\langle\{a\}\{a,b,d\}\{a,b,c\}\rangle, \{b,d\})$=$\{2,3\}$.

The following lemma now formalizes the observation that we only need to consider the sets X_i for i in the position set.

Lemma 1 *Let S be a sequence, and Y an itemset. Then $\phi(S\circ Y) = \phi(S)\cdot 2^{|Y|} - R(S,Y)$, with*

$$R(S,Y) \ = \ \left|\bigcup_{\ell\in L(S,Y)} \{\varphi(S^{\ell-1})\circ\mathcal{P}_{\geq 1}(S[\ell]\cap Y)\}\right|$$

See Appendix.

Notice, however, that the sets $\varphi(S^{\ell-1})\circ\mathcal{P}_{\geq 1}(S[\ell]\cap Y)$ are not necessarily disjoint; consider, e.g., $S = \langle\{a,b\},\{b,c\}\rangle$ and $Y = \{a,b,c\}$. Then $L(S,Y) = \{1,2\}$, and $\langle\{b\}\rangle$ appears in both $\varphi(S^0)\circ\mathcal{P}_{\geq 1}(S[1]\cap Y)$ and $\varphi(S^1)\circ\mathcal{P}_{\geq 1}(S[2]\cap Y)$. To incorporate this overlap, we compute the cardinality of the union in Lemma 1 using the inclusion-exclusion principle, leading to the following theorem:

Theorem 1 *Let $S = \langle X_1 ... X_n\rangle$ and Y an itemset. Then,*

$$\phi(S\circ Y) = 2^{|Y|}\cdot\phi(S) - R(S,Y) \tag{2}$$

with

$$R(S,Y) = \sum_{K \subseteq L(S,Y)} (-1)^{|K|+1} \cdot D(S,Y,K)$$

where

$$D(S,Y,K) = \phi(S^{\min(K)-1}) \cdot \left(2^{\left|\left(\bigcap_{j \in K} X_j\right) \cap Y\right|} - 1\right)$$

See Appendix.

We illustrate the counting process with sequence S_4^3. The position set of this sequence is given in Example 3.

$$\begin{aligned}
\phi(\langle\rangle) &= 1 \\
\phi(\langle\{a\}\rangle) &= 2^{|\{a\}|} \cdot \phi(\langle\rangle) \;=\; 2 \\
\phi(\langle\{a\}\{a,b,d\}\rangle) &= 2^{|\{a,b,d\}|}\phi(\langle\{a\}\rangle) \\
&\quad -(2^{|\{a,b,d\} \cap \{a\}|} - 1) \cdot \phi(\langle\rangle) \\
&= 2^3 \cdot 2 - (2^1 - 1) \cdot 1 = 15 \\
\phi(\langle\{a\}\{a,b,d\}\{a,b,c\}\rangle) & \\
&= 2^{|\{a,b,c\}|} \cdot \phi(\langle\{a\}\{a,b,d\}\rangle) \\
&\quad -(2^{|\{a,b,d\} \cap \{a,b,c\}|} - 1) \cdot \phi(\langle\{a\}\rangle) \\
&= 2^3 \cdot 15 - (2^2 - 1) \cdot 2 = 114
\end{aligned}$$

5 Counting All Common Subsequences

In this section, we will extend the previous results to count all common distinct subsequences between two sequences S and T. Again, we discuss the basic intuition and then present the main result. Suppose that we extend the sequence S with an itemset Y and we observe the relation between $\varphi(S,T)$ and $\varphi(S \circ Y, T)$, two cases may appear:

1. If no items in Y appear in any itemset of S and T then the concatenation of the itemset Y with the sequence S *has no effect* on the the set $\varphi(S,T)$.

2. If at least an item in Y appears in either one of the sequences S or T (or both) then it can be observed that new common subsequences will appear in $\varphi(S,T)$. As for the counting method of the distinct subsequences of a unique sequence S, repetitions may occur and a generalized correction term for both S and T needs to be defined. Formally,

$$|\varphi(S \circ Y, T)| = |\varphi(S,T)| + A(S,T,Y) - R(S,T,Y) \tag{3}$$

   where $A(S,T,Y)$ represents the number of extra common subsequences that should be added and $R(S,T,Y)$ is the correction term.

Similarly to the distinct subsequence problem, the position set will index the positions that generate duplicate sequences. The following lemma formalizes this observation:

Lemma 2 *Let $S = \langle X_1 ... X_n \rangle$, $T = \langle X'_1 ... X'_m \rangle$ and Y an itemset.*

$$A(S,T,Y) \;=\; \left| \bigcup_{\ell \in L(T,Y)} \left\{ \varphi(S, T^{\ell-1}) \circ \mathcal{P}_{\geq 1}(T[\ell] \cap Y) \right\} \right|$$

$$R(S,T,Y) = \left| \bigcup_{\ell \in L(S,Y)} \left\{ \bigcup_{\ell' \in L(T,Y)} \varphi(S^{\ell-1}, T^{\ell'-1}) \circ \mathcal{P}_{\geq 1}(S[\ell] \cap T[\ell'] \cap Y) \right\} \right|$$

Example 4 *Consider the sequences S_1 and S_2 from our running example. Let $S_1^4 = \langle\{a\}\{a,b\}\{e\}\{c,d\}\rangle$ be the 4-prefix of S_1, and let $S_2^3 = \langle\{a\}\{b,c,d\}\{a,d\}\rangle$ be the 3-prefix of S_2. Suppose that we extend S_1^4 with the itemset $Y = \{b,d\}$ and count all distinct common subsequences between $S_1^4 \circ \{b,d\}$ and S_2^3. Notice that the itemset $\{b,d\}$ appears two times in the sequence S_2^3: in the itemsets $\{b,c,d\}$ and $\{a,d\}$. Thus, $L(S_2^3, \{b,d\}) = \{2,3\}$ and $A(S_1^4, S_2^3, \{b,d\}) = |\{\varphi(S_1^4, S_2^1) \circ \mathcal{P}_{\geq 1}(\{b,d\} \cap \{b,c,d\})\} \cup \{\varphi(S_1^4, S_2^2) \circ \mathcal{P}_{\geq 1}(\{b,d\} \cap \{a,d\})\}| = 14$. Notice also that $L(S_1^4, \{b,d\}) = \{2,4\}$. In this case, adding the values $A(S_1^4, S_2^3, \{b,d\})$ to $\phi(S_1^4, S_2^3)$ will overcount some subsequences. For instance, the subsequences $\langle\{a\}\{b\}\{d\}\rangle$ and $\langle\{b\}\{d\}\rangle$ are counted twice: once in $\varphi(S_1^4, S_2^3)$ and when all sequences of the set $\varphi(S_1^4, S_2^2)$ are extended with $\{b,d\} \cap \{a,d\}$. The same remark applies to other subsequences: $\langle\{b\}\rangle$, $\langle\{a\}\{b\}\rangle$ and $\langle\{a\}\{d\}\rangle$. In this case, the correct number of all common subsequences between $S_1^4 \circ \{b,d\}$ and S_2^3 is $|\varphi(S_1^4, S_2^2)| + A(S_1^4, S_2^3, \{b,d\}) - R(S_1^4, S_2^3, \{b,d\}) = 13 + 14 - R(S_1^4, S_2^3, \{b,d\})$ with $R(S,T,Y) = |\{\varphi(S_1^1, S_2^1) \circ \mathcal{P}_{\geq 1}(\{a,b\} \cap \{b,c,d\} \cap \{b,d\})\} \cup \{\varphi(S_1^1, S_2^2) \circ \mathcal{P}_{\geq 1}(\{a,b\} \cap \{a,d\} \cap \{b,d\})\} \cup \{\varphi(S_1^3, S_2^1) \circ \mathcal{P}_{\geq 1}(\{c,d\} \cap \{b,c,d\} \cap \{b,d\})\} \cup \{\varphi(S_1^3, S_2^2) \circ \mathcal{P}_{\geq 1}(\{c,d\} \cap \{a,d\} \cap \{b,d\})\}| = 6$. Thus, $\phi(S_1^4 \circ \{b,d\}, S_2^3) = 21$.*

Similarly to Lemma 1 and as illustrated in the above example, the computation of the cardinality of the unions in Lemma 2 implies the usage of the inclusion-exclusion principle. This remark leads to the second theorem:

Theorem 2 *Let $S = \langle X_1 ... X_n\rangle$, $T = \langle X'_1 ... X'_m\rangle$ and Y an itemset. Then,*

$$\phi(S \circ Y, T) = \phi(S,T) + A(S,T,Y) - R(S,T,Y) \tag{4}$$

with

$$A(S,T,Y) = \sum_{K \subseteq L(T,Y)} (-1)^{|K|+1} \cdot \phi(S, T^{\min(K)-1}) \cdot \left(2^{\left|\left(\bigcap_{j \in K} X'_j\right) \cap Y\right|} - 1\right) \tag{5}$$

and

$$R(S,T,Y) = \sum_{K \subseteq L(S,Y)} (-1)^{|K|+1} \cdot \sum_{K' \subseteq L(T,Y)} (-1)^{|K'|+1} \cdot D(S,T,Y,K,K') \tag{6}$$

where

$$D(S,T,Y,K,K') = \phi(S^{\min(K)-1}, T^{\min(K')-1}) \cdot 2^{\left|\left(\bigcap_{j \in K} X_j\right) \cap \left(\bigcap_{j' \in K'} X'_{j'}\right) \cap Y\right|} - 1$$

See Appendix.

5.1 Dynamic Programming

Theorem 2 implies a simple dynamic programming algorithm. For two given sequences S and T, such that $|S| = n$ and $|T| = m$, the program produces a $n \times m$ matrix, denoted $\mathcal{M}$, where the $\mathcal{M}_{i,j}$ cell corresponds to all common subsequences between S^i and T^j, $\mathcal{M}_{i,j} = \phi(S^i, T^j)$.

	$\{\emptyset\}$	$\{a\}$	$\{b,c,d\}$	$\{a,d\}$
$\{\emptyset\}$	1	1	1	1
$\{a\}$	1	2	2	2
$\{a,b\}$	1	2	4	5
$\{e\}$	1	2	4	5
$\{c,d\}$	1	2	10	13
$\{b,d\}$	1	2	12	21

Table 2: Matrix for counting all common subsequences between S_1 and S_2

Example 5 *Consider the two sequences* $S_1 = \langle\{a\}\{a,b\}\{e\}\{c,d\}\{b,d\}\rangle$ *and* $S_2 = \langle\{a\}\{b,c,d\}\{a,d\}\rangle$. $\phi(S_1,S_2) = 21$ *and the set of all common subsequences of* S_1 *and* S_2 *is:*

$$\begin{aligned}\varphi(S_1,S_2) &= \{\emptyset, \langle\{a\}\rangle, \langle\{b\}\rangle, \langle\{c\}\rangle, \langle\{d\}\rangle, \langle\{cd\}\rangle, \langle\{bd\}\rangle, \langle\{a\}\{a\}\rangle, \langle\{a\}\{b\}\rangle, \\ &\quad \langle\{a\}\{c\}\rangle, \langle\{a\}\{d\}\rangle, \langle\{b\}\{d\}\rangle, \langle\{c\}\{d\}\rangle, \langle\{d\}\{d\}\rangle, \langle\{a\}\{cd\}\rangle, \langle\{a\}\{bd\}\rangle, \\ &\quad \langle\{cd\}\{d\}\rangle, \langle\{a\}\{d\}\{d\}\rangle, \langle\{a\}\{b\}\{d\}\rangle, \langle\{a\}\{c\}\{d\}\rangle, \langle\{a\}\{cd\}\{d\}\rangle\}\end{aligned}$$

We detail the computation of the cell $\mathcal{M}_{2,1}$ *with the position set* $L(S_2^1, \{a,b\}) = \{1\}$ *and* $L(S_1^1, \{a,b\}) = \{1\}$*:*

$$\begin{aligned}\mathcal{M}(\{a,b\},\{a\}) &= \phi(\langle\{a\}\{a,b\}\rangle, \langle\{a\}\rangle) \\ &= \mathcal{M}(\{a\},\{a\}) \\ &\quad +(2^{|\{a\}\cap\{a,b\}|} - 1)\cdot\mathcal{M}(\{a\},\{\emptyset\}) \\ &\quad -(2^{|\{a\}\cap\{a\}\cap\{a,b\}|} - 1)\cdot\mathcal{M}(\{\emptyset\},\{\emptyset\}) \\ &= 2+1-1 = 2\end{aligned}$$

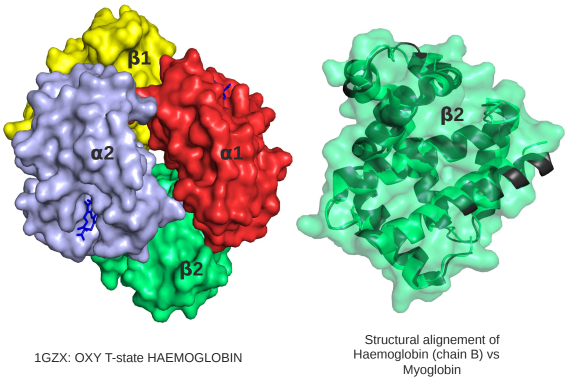
The entire computation for $\phi(S_1,S_2)$ *is illustrated in Table 2.*

6 Experiments

6.1 Protein folding: a relationship between sequence and structure

Protein-protein interactions are involved in almost all biological processes. Their function mostly depends on their 3-dimensional (3D) structure which helps understanding how proteins interact and how they can be regulated in case of pathology[12].

Our first batch of experiments focus on using our ACS similarity measure for the problem of homology modeling of proteins [3]. This problem can be stated as follows: can we compare the 3D-structure of two proteins based on the 2D-structure and additional structural information? The solution based on our similarity is novel from a biological point of view and intuitively easier than the usual methods for 3D-structures in bio-informatics. In this experiment we consider the primary structure (or sequence) corresponding to the linear assembling of amino-acid (20) residues along with their chemical properties. In addition, we use the secondary structure which



1GZX: OXY T-state HAEMOGLOBIN

Structural alignement of Haemoglobin (chain B) vs Myoglobin

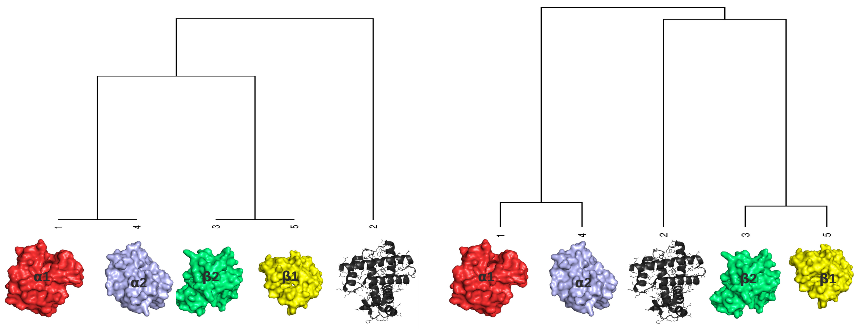
Figure 1: On the left, the 3D-structure of haemoglobin protein with surface representation of the four chains $\alpha1$, $\alpha2$, $\beta1$, $\beta2$ are represented. On the right, the overall structure superposition of chain $\beta2$ of haemoglobin and the single chain myoglobin protein is shown.

is a local refinement of the primary structure into periodic folders which corresponds to the itemsets in our sequence model.

For comparing two proteins and measuring their similarity, the alphabet of 20 residues was reduced to 6 elements referring to 6 categories related to their chemical similarities: small *(AGSTCP)*, aromatic *(YW)*, aliphatic *(IFMLV)*, polar *(NQ)*, negatively *(DE)* and positively *(HKR)* charged. An additional information was associated to each residue: the possible membership to a secondary structure motif. Accordingly, the primary structure of a protein is considered as a sequence $S = \{k_1, k_2 \ldots k_m\}$, where each k_i corresponds to one of the six categories introduced above and where each itemset of S is a representation of the secondary structure. The following problem has been chosen for testing the ACS similarity. A well-known limit of homology modeling concerns two proteins, namely myoglobin and haemoglobin (Hgb), both involved in oxygen transport of almost all vertebrates. Haemoglobin is composed of two chains commonly annotated α $(\alpha1, \alpha2)$ and two chains β $(\beta1, \beta2)$ as shown in Figure 1. The right part of Figure 1 shows that the $\beta2$ chain of haemoglobin fits the myoglobin structure. However, the usual measure of similarity separate the α and β chains of haemoglobin and myoglobin as shown in Figure 2a. In comparison, the result of ACS is reported on Figure 2b. The myoglobin structure is quite similar to both α and β chains of haemoglobin. In biological terms, this means that even if haemoglobin and myoglobin are different, their secondary 3D-structure and their chemical characteristics are in correspondence.

6.2 Protein Families

In this experiment, two sets of proteins were considered. Two families of proteins which are distinct in term of function, structurally different among the two sets but similar inside each set. The number of proteins for each family was set to 5. The ACS similarity measure was computed between the protein sequences and results are reported below Figure 3. The cross-similarity matrix reveals a correct separation between the two sets of proteins. The interest for biologists with this similarity measure is to focus on proteins which biologically unknown



(a) Primary Sequence Similarity

(b) ACS Similarity

Figure 2: Comparison of results based on primary sequence similarity only, and with ACS algorithm. On each tree, proteins 1 and 4 correspond to α proteins of haemoglobin, and proteins 3 and 5 to β proteins. Protein 2 corresponds to the myoglobin protein.

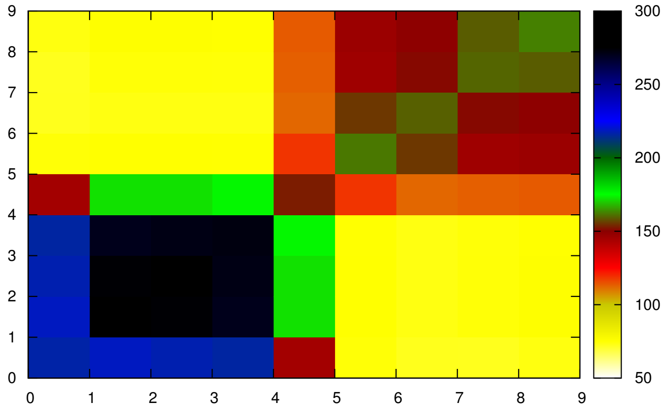


Figure 3: ACS similarity matrix. Gradient colors corresponding to the $\ln(ACS)$ between two sequences and to the $\ln(ADS)$ on the diagonal element of the matrix. The two sets of proteins appear along the diagonal matrix

Experiments	Dataset	Sequences Length	Itemset Length	Nb-Sequence
$stage_1$	$data_1$	50	25	1000
	$data_2$		20	
	$data_3$		15	
	$data_4$		10	
	$data_5$		5	
$stage_2$	$data_1$	80	15	1000
	$data_2$	60		
	$data_3$	40		
	$data_4$	20		
$stage_3$	$data_1$	25	15	1000
	$data_2$			2000
	$\vdots$			$\vdots$
	$data_{10}$			10000

Table 3: Synthetic Datasets

functions potentially infer them based on the number of common subsequences.

6.3 Experiments on Synthetic Datasets

In the following, we study the scalability of our measure computation. We assess the different runtimes with respect to three different parameters:

1. The average number of itemsets in a sequence.
2. The average number of items in each itemset of a sequence.
3. The total number of sequences that are processed through the similarity computation.

We carry out our experiments on three stages, each focusing on one of the previously listed parameters. Table 3 describes the synthetic datasets that were generated and used in our experiments.

The similarity matrix computation was parallelized, and up to 4 cores were fully allocated to compute the different subregions of the matrix. Each of our experiments was repeated five times and all the trials are plotted in Figures 4, 5 and 6.

Figure 4 represents the evolution of the runtime for each subregion of the similarity matrix w.r.t the average number of items in each itemset. We run this test on five types of sequences: the sequences with 5 items in their itemsets, with 10 , 15 , 20 and sequences with an average cardinality of 25 items in their itemsets. The boxplots represent the variations of the running time, considering the interval of sequences $[1, 50]$ then $[50, 100]$, $[100, 150]$, until the last block which represents the sequences in the interval $[950, 1000]$. For example, the average calculation time for the first subregion of similarity matrix between the 1^{st} and 50^{th} performs $\frac{50 \cdot 51}{2} + (1000 - 50) \cdot 50 = 48775$ similarity comparisons and needs 1231 seconds to do that. We can also notice that the running time decreases for each processed subregion in the matrix. For example, the average calculation time for the last part of matrix between the 950^{th} and 1000^{th} sequence is only 48 seconds.

In Figure 5, we show the performance of our similarity measure w.r.t the average length of the sequences. We run this test on four types of data sequences: sequences with $80, 60, 40$ and 20 itemsets. As expected, and noticed in the figure, the runtime increases with the length of the sequence. For example, the yellow boxplot represents the variations of the runtime for generating each subregion of the matrix for sequences of average length 60. The red boxplot represents the variations of the running time for sequences with length 80, the calculation time increases by a almost a factor of two, but remains acceptable, when we increase the average length of sequence.

Figure 6 shows the results of our experimnets on the total number of processed sequences. We simply calculate the similarity measure for 2 sequences. We note that the running time is increasing linearly. For example, the time needed to compute the similarities of 1000 pairs of sequences is 9.589 seconds and the calculation time for 2000 pairs is 18.467 seconds.

These experiments highlight the fact that our measure is efficient in term of runtime for a large panel of sequences with different varying parameters.

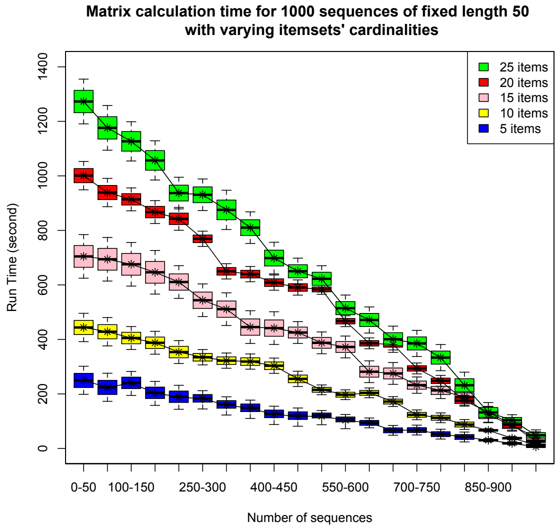


Figure 4: Boxplots showing the variations of the running time when calculating the similarity matrix for sequences depending on the number of items.

7 Conclusion

In this paper, we study the problem of counting all common subsequences between two sequences of itemsets. We present an efficient dynamic programming algorithm (ACS) to count the number of common subsequences between two sequences. This solution allows us to define in a simple and intuitive manner a similarity measure between two sequences S and T. This similarity has been successfully applied for the analysis of real-world biological data and for synthetic datasets. An ongoing work is on the approximation of the number of all common subsequences to speed up the computation for long biological sequences.

References

[1] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," in *SIGMOD Conference*, R. T. Snodgrass and M. Winslett, Eds. ACM Press, 1994, pp. 419–429.

[2] C. Sander and R. Schneider, "Database of homology-derived protein structures and the structural meaning of sequence alignment," *Proteins*, vol. 1, no. 9, pp. 56–68, 1991.

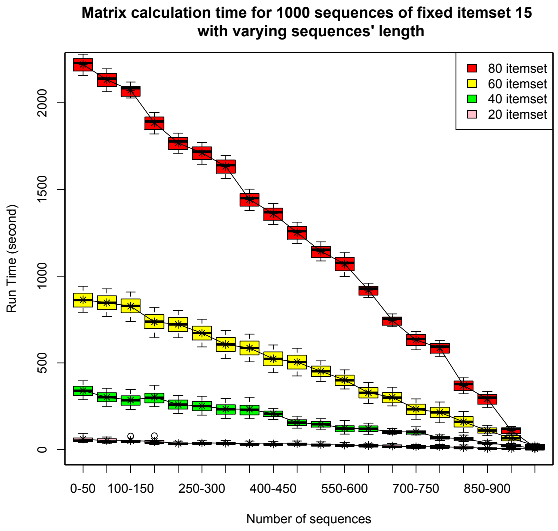


Figure 5: Boxplots showing the variations of the running time when calculating the similarity matrix for sequences depending on the number of itemsets.

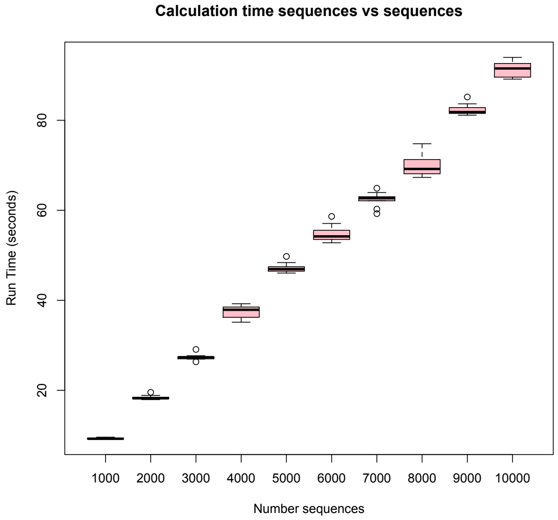


Figure 6: Calculation time of similarity measure for sequences vers sequences.

[3] C. Chothia and M. Gerstein, "Protein evolution. how far can sequences diverge?" *Nature*, vol. 6617, no. 385, pp. 579–581, 1997.

[4] Q. Yang and H. H. Zhang, "Web-log mining for predictive web caching," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 4, pp. 1050–1053, 2003.

[5] J. Serrà, H. Kantz, X. Serra, and R. G. Andrzejak, "Predictability of music descriptor time series and its application to cover song detection," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 20, no. 2, pp. 514–525, 2012.

[6] C. S. Leslie, E. Eskin, and W. S. Noble, "The spectrum kernel: A string kernel for svm protein classification," in *Pacific Symposium on Biocomputing*, 2002, pp. 566–575.

[7] T. Xiong, S. Wang, Q. Jiang, and J. Z. Huang, "A new markov model for clustering categorical sequences," in *ICDM*, D. J. Cook, J. Pei, W. Wang, O. R. Zaïane, and X. Wu, Eds. IEEE, 2011, pp. 854–863.

[8] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[9] J. Herranz, J. Nin, and M. Sole, "Optimal symbol alignment distance: A new distance for sequences of symbols," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, pp. 1541–1554, 2011.

[10] E. Keogh, "Exact indexing of dynamic time warping," in *Proceedings of the 28th international conference on Very Large Data Bases*, ser. VLDB '02, 2002, pp. 406–417.

[11] H. Wang, Z. Lin, and G. Gediga, "Counting all common subsequences to order alternatives," in *RSKT*, 2007, pp. 566–573.

[12] S. Wodak and J. Janin, "Structural basis of macromolecular recognition." *Adv Protein Chem*, vol. 61, pp. 9–73, 2002.

[13] J. Oncina and M. Sebban, "Learning stochastic edit distance: Application in handwritten character recognition," *Pattern Recogn.*, vol. 39, no. 9, pp. 1575–1587, Sep. 2006.

[14] F. Muzaffar, B. Mohsin, F. Naz, and L. F. Jawed, "Dsp implementation of voice recognition using dynamic time warping algorithm," *IEEE Explore*, pp. 1–7, 2005.

[15] M. J. Z. Karlton Sequeira, "Admit: Anomaly-base data mining for intrusions," in *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Jul 2002.

[16] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. J. Keogh, "Indexing multi-dimensional time-series with support for multiple distance measures," in *KDD*, 2003, pp. 216–225.

[17] C. Elzinga, S. Rahmann, and H. Wang, "Algorithms for subsequence combinatorics," *Theor. Comput. Sci.*, vol. 409, no. 3, pp. 394–404, 2008.

Proof of Lemma 1

Let $T = \langle T_1, \ldots, T_m \rangle$ be a sequence that is counted multiple times; i.e., $T \in (\varphi(S) \circ \mathcal{P}_{\geq 1}(Y)) \cap \varphi(S)$. Clearly $T_m \in \mathcal{P}_{\geq 1}(Y)$ as otherwise T would not have been in $\varphi(S) \circ \mathcal{P}_{\geq 1}(Y)$. Let k denote $\max\{j | T_m \subseteq S[j]\}$. Since $T \in \varphi(S)$, such k must exist. Then, $k \in L(S,Y)$, since k is the largest index for which $S[k] \cap Y$ includes T_m. Therefore, $T \in \varphi(S^{k-1}) \circ \mathcal{P}_{\geq 1}(S[k] \cap Y)$ for a $k \in L(S,Y)$.

□

Proof of Theorem 1

The proof is a simple application of the inclusion-exclusion principle to compute the cardinality of the union of Lemma 1:

$$R(S,Y) = \left| \bigcup_{\ell \in L(S,Y)} \left\{ \varphi(S^{\ell-1}) \circ \mathcal{P}_{\geq 1}(S[\ell] \cap Y) \right\} \right|$$

$$R(S,Y) = \sum_{K \subseteq L(S,Y)} (-1)^{|K|+1} \left| \bigcap_{\ell \in K} \left\{ \varphi(S^{\ell-1}) \circ \mathcal{P}_{\geq 1}(S[\ell] \cap Y) \right\} \right|$$

The proof is completed by the following two observations:

$$\begin{aligned} set_K &:= \bigcap_{\ell \in K} \left\{ \varphi(S^{\ell-1}) \circ \mathcal{P}_{\geq 1}(S[\ell] \cap Y) \right\} \\ &= \varphi(S^{\min(K)-1}) \circ \mathcal{P}_{\geq 1}((\cap_{k \in K} S[k]) \cap Y) \end{aligned}$$

Indeed; any sequence of length m in set_K has $T^{m-1} \in S^{\min(K)-1}$, and $T_m \in \mathcal{P}_{\geq 1}(S[k] \cap Y)$, for all $k \in K$. And, the second observation:

$$|set_K| = \phi(S^{\min(K)-1}) \cdot \left(2^{|(\cap_{k \in K} S[k]) \cap Y|} - 1 \right)$$

□

Proof of Theorem 2

1. No items in Y appear in any itemset of S and T, in this case the set of all common distinct subsequences between $S \circ Y$ and T is exactly the same set of all common distinct subsequences between S and T. Hence, $\phi(S \circ Y, T) = \phi(S, T)$.

2. If at least an item in Y appears in either one of the sequences S or T (or both), then $\varphi(S \circ Y, T)$ is expressed as the union of the set of all common distinct subsequences between S and T with the set of added sequences $\mathcal{A}$ *without* the set of repeated sequences $\mathcal{R}$. Formally,

   $$\varphi(S \circ Y, T) = \varphi(S, T) \cup \mathcal{A} \backslash \mathcal{R} \tag{7}$$

   with

   $$\mathcal{A} = \left\{ \bigcup_{\ell' \in L(T,Y)} \varphi(S, T^{\ell'-1}) \circ \mathcal{P}_{\geq 1}(T[\ell'] \cap Y) \right\} \tag{8}$$

and

$$\mathcal{R} = \left\{ \bigcup_{\ell \in L(S,Y)} \left\{ \bigcup_{\ell' \in L(T,Y)} \varphi(S^{\ell-1}, T^{\ell'-1}) \circ \mathcal{P}_{\geq 1}(S[\ell] \cap T[\ell'] \cap Y) \right\} \right\} \tag{9}$$

Notice that because these three sets are disjoint, the cardinality of $\varphi(S \circ Y, T)$ can be simply expressed as $|\varphi(S \circ Y, T)| = |\varphi(S, T)| + |\mathcal{A}| - |\mathcal{R}|$. Using the inclusion-exclusion principle, $|\mathcal{A}|$, denoted as $A(S, T, Y)$ can be written as,

$$\begin{aligned} A(S,T,Y) &= \left| \bigcup_{\ell \in L(T,Y)} \left\{ \varphi(S, T^{\ell-1}) \circ \mathcal{P}_{\geq 1}(T[\ell] \cap Y) \right\} \right| \\ &= \sum_{K \subseteq L(T,Y)} (-1)^{|K|+1} \, |set_K| \end{aligned}$$

where

$$set_K = \bigcap_{\ell \in K} \left\{ \varphi(S, T^{\ell-1}) \circ \mathcal{P}_{\geq 1}(T[\ell] \cap Y) \right\}$$

$A(S, T, Y)$ is completed by the following two observations:

$$\begin{aligned} set_K &:= \bigcap_{\ell \in K} \left\{ \varphi(S, T^{\ell-1}) \circ \mathcal{P}_{\geq 1}(T[\ell] \cap Y) \right\} \\ &= \varphi(S, T^{\min(K)-1}) \circ \mathcal{P}_{\geq 1}((\cap_{k \in K} T[k]) \cap Y) \end{aligned}$$

And, the second observation:

$$|set_K| = \phi(S, T^{\min(K)-1}) \cdot \left(2^{|(\cap_{k \in K} T[k]) \cap Y|} - 1 \right)$$

$A(S, T, Y)$ can be written as,

$$A(S,T,Y) = \sum_{K \subseteq L(T,Y)} (-1)^{|K|+1} \cdot \phi(S, T^{\min(K)-1}) \cdot \left(2^{\left|\left(\cap_{j \in K} X'_j\right) \cap Y\right|} - 1 \right) \tag{10}$$

The same inclusion-exclusion reasoning applies to the cardinality of $\mathcal{R}$, denoted $R(S, T, Y)$

$$\begin{aligned} R(S,T,Y) &= \left| \left\{ \bigcup_{\ell \in L(S,Y)} \left\{ \bigcup_{\ell' \in L(T,Y)} \mathcal{D}_{\ell,\ell'} \right\} \right\} \right| \\ &= \sum_{K \subseteq L(S,Y)} (-1)^{|K|+1} \cdot \sum_{K' \subseteq L(T,Y)} (-1)^{|K'|+1} \cdot \left| set_{K,K'} \right| \end{aligned}$$

and

$$set_{K,K'} = \bigcap_{\ell \in K} \bigcap_{\ell' \in K'} \varphi(S^{\ell-1}, T^{\ell'-1}) \circ \mathcal{P}_{\geq 1}(S[\ell] \cap T[\ell'] \cap Y)$$

The final result follows after noticing that,

$$set_{K,K'} = \bigcap_{\ell \in K} \bigcap_{\ell' \in K'} \varphi(S^{\ell-1}, T^{\ell^{'}-1}) \circ \mathcal{P}_{\geq 1}(S[\ell] \cap T[\ell^{'}] \cap Y)$$

$$set_{K,K'} = \varphi(S^{\min(K)-1}, T^{\min(K^{'})-1}) \circ \mathcal{P}_{\geq 1} \left((\cap_{k \in K} S[k]) \cap (\cap_{k' \in K'} T[k^{'}]) \cap Y \right)$$

$R(S,T,Y)$ can be written as,

$$R(S,T,Y) = \sum_{K \subseteq L(S,Y)} (-1)^{|K|+1} \cdot \sum_{K' \subseteq L(T,Y)} (-1)^{|K^{'}|+1} \cdot D(S,T,Y,K,K^{'}) \tag{11}$$

where

$$D(S,T,Y,K,K^{'}) = \phi(S^{\min(K)-1}, T^{\min(K^{'})-1}) \cdot 2^{\left|(\bigcap_{j \in K} X_j) \cap \left(\bigcap_{j' \in K'} X^{'}_{j'}\right) \cap Y\right|} - 1$$

□

Contents

RESEARCH CENTRE
NANCY – GRAND EST

615 rue du Jardin Botanique
CS20101
54603 Villers-lès-Nancy Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399