

Sequence Similarity

A Nonaligning Technique

CEES H. ELZINGA
Vrije Universiteit Amsterdam

This article reviews objections to optimal-matching (OM) algorithms in sequence analysis and reformulates the concept of sequence similarity in terms of a binary precedence relation. This precedence relation is then used to develop a new quantification of sequence similarity. The new measure is used to reanalyze the life history data that were previously discussed by Dijkstra and Taris (1995). The reanalysis demonstrates the new measure to be superior to the OM algorithm and the alternatives proposed by Dijkstra and Taris. A new algorithm is presented to enumerate matching k -tuples from pairs of sequences in polynomial time.

Keywords: *sequence analysis; sequence similarity; OM; life history*

1. INTRODUCTION

Some 18 months ago, this journal dedicated an entire issue to a methodology of research that has become known as sequence analysis. Andrew Abbott, having been the senior advocate of this type of research for more than 15 years now, and Angela Tsay prompted a discussion with their review article about the subject (Abbott and Tsay 2000). Joel Levine (2000) and Lawrence Wu (2000) responded to this provocative article by seriously questioning the payoff, validity, and potential of the methodology. Abbott (2000) then answered Levine and Wu in a way that eventually could lead us to a sensible broadening of the subject, but this hardly contributed to a clarification of the main issue: the legitimacy and usefulness of employing a particular type of dynamic programming to determine sequence distance or sequence similarity.

At least all involved in the debate seem to agree on what one does in sequence analysis: collect social scientific data, encode these data in the form of sequences of tokens from an alphabet that represents some taxonomy, calculate distances or similarities among these sequences, and then further reduce the resulting similarity or distance matrix with some clustering or scaling technique.

Apparently, it is common practice to determine distance or similarity (preferably interpreted as proximity: the reverse of distance) by using one of a family of algorithms, which has come to be known as *optimal matching* (OM) since Abbott and Forrest (1986) first introduced it into the social sciences.

Basically, an OM algorithm seeks to find the minimal number of edit operations (inserts, deletes, and substitutions) necessary to turn one member of a pair of sequences into an exact copy of the other sequence; these operations result in a perfectly aligned pair.

Mathematically, this alignment problem is known as the string-to-string correction problem and can be solved by dynamic programming techniques (e.g., Wagner 1983; Kruskal 1983).

The use of OM algorithms has become widespread through various disciplines outside the social sciences, especially in biomedics, phylogenetics, and pharmaceuticals. Application of OM-like algorithms has also proliferated into, for example, risk evaluation by moneylenders through comparing payback profiles.

Essentially, sequence analysis is a descriptive technique, an approach to data reduction that may or may not lead to the unveiling of pattern and structure in sequence data.

In this approach, the calculation of distances or similarities is only one step that, in principle, could be done with any suitable technique. Apparently, OM has become the standard technique—so much so, that sequence analysis and OM-like techniques are commonly regarded as being almost synonymous. Abbot, Wu, Levine, and many other authors use these concepts as interchangeable equivalents.

The result of this confounding terminology has been that criticism of and objections to the application of OM have been regarded and received as criticism of sequence analysis as an approach to data reduction. The application of OM-like techniques has been the main source of skepticism and objection to sequence analysis. Therefore, we will start this article by reviewing the main objections to the

application of OM in the social sciences. Then we will discuss an alternative to OM algorithms—the Dijkstra and Tavis (DT) coefficients—and its limitations. We use the latter discussion to develop some basic principles or axioms on which any measure of sequence similarity should be constructed. In the next two sections, we will interpret these axioms and develop a new similarity measure. Two further sections confront this new measure with some well-known examples and sequence data. In the last section, we present a very efficient algorithm to compute this similarity index.

2. OM ALGORITHMS: BASIC SHORTCOMINGS

In biochemistry, inserts, deletes, and substitutions concern sequences of tokens, each of which represents a chemical building block of the substance or species at hand. Making an insert, delete, or substitution of a token or even a subsequence of considerable length is justified by arguments rooted in a theory about the electrochemical, mechanical, or functional properties of what is deleted, inserted, or substituted for. Oftentimes, there is a plausible theory or credible hypothesis about the probability that such a set of operations really took place or could have taken place in the course of evolution. Such probabilities or degrees of chemical or functional equivalence then give rise to the employment of cost matrices associated with the edit operations, in which case the OM algorithm is adapted to maximize cost-efficiency instead of minimizing the number of operations.

The bulk of the criticisms concerning the application of sequence analysis just concerns this basic mechanism. We will therefore discuss these criticisms:

1. The taxonomies employed in the social sciences are too weak or fuzzy to permit encoding events or attributes (e.g., Levine 2000). Essentially, this criticism applies to any taxonomy in any branch of science. In all sciences, taxonomies are continually changing, and changing them is one of the major goals of those active in the field.

2. In the life sciences, the edit operations have a very nice interpretation of theoretical concepts: chemical and geometrical changes and perturbations of chains of amino acids, enzymes, and peptidic

structures. The cost matrices in OM algorithms reflect their evolutionary or physical probabilities of occurrence. OM then generates a representation of the sequences in a metric space (evolutionary theory specifying properties of its metric), and life scientists use this representation as a model of, for example, the phylogenesis of the species involved. Therefore, the way the representation, in the guise of measures of distance or similarity, is constructed is of vital importance. Optimal alignment of a pair of sequences can be obtained when certain elements are displaced, removed, or substituted, and the numbers of these operations and their relative costs determine similarity or distance. Had the cost functions been different, the resulting representation (and, therefore, the phylogenetic model) would have been different.

Levine (2000:37) writes about careers, “The analog to DNA is not obvious.” Wu (2000:46) wonders, “In what sense might careers or childbearing be usefully analyzed in a manner like DNA?”

In the social sciences, the situation is different indeed—different because those using or advocating sequence analysis are not primarily interested in the phylogenesis of a particular sequence or class of sequences. Rather, they are interested in classifying or scaling the sequences: parsimonious description, data reduction, or “fishing for patterns” (Abbott 2000:69). Specifying a cost function embodies a very precise, numerical, (proto-) theoretical notion of the relative importance of the events that make up the sequence and the order in which they do or do not appear in a sequence. Therefore, the spatial representation of the sequences, as well as the metric of the space itself, is to be considered as a precise, geometrical model of a sociological theory. Abbott’s (2000:67) claim that OM algorithms are not models is correct: Only the application of such an algorithm, and therewith specifying a cost function, is or implies a model. Unfortunately, I do not know of sociological theories that permit such precise and rich geometrical models.

So, the question is not so much (Wu 2000; Levine 2000) whether the edit operations mimic sociological processes. Instead, the question is whether the geometrical model, resulting from the edit operations and their associated cost function, reflects an equally rich and precise sociological theory. I’m afraid it does not.

To this criticism, we have not seen an answer yet. Certainly, if it exists at all, such an answer will not be found by looking at more and more sophisticated algorithms and clever costing schemes (e.g., Abbott and Tsay 2000): Someone will have to decide that such algorithms or costing schemes are “better” or even “best” in terms of criteria that cannot be found within the algorithms themselves or the results they produce.

Neither can this criticism be dismissed by referring to (e.g., Abbott 2000) biomedics having computational problems also or to money-lenders employing dynamic programming algorithms on a massive scale: Our problem is not primarily computational, and our purpose is not profitable money lending. Our challenge, within the context of sequence analysis as a descriptive, exploratory tool, is to find a representation of the sequences and their similarities that is free of any sociological or historical theory—one that just relies on the basic properties of a sequence: its constituting elements and the order in which they appear.

3. Dijkstra and Taris (1995) already objected to OM because of the way it handles the order of the tokens in sequences. Wu (2000) also objected to OM algorithms because they do not recognize the significance of specific orders in which tokens appear in a sequence: If “1” and “0” represent “having a job” and “being unemployed,” then the (sub)sequence $\{0, 1\}$ has a behaviorally, socially, and culturally quite different meaning from the subsequence $\{1, 0\}$ (e.g., Wu 2000). Therefore, whatever the meaning of the tokens “2,” “3,” “4,” and “5,” the distance between the sequences $\{2, 0, 1, 3\}$ and $\{2, 1, 0, 3\}$ could be much more significant than the distance between either of these and the sequence $\{2, 4, 5, 3\}$, even though they are the same number of edit operations apart.

Furthermore, Wu (2000) argues, OM algorithms are insensitive to “direction of time”: According to OM algorithms, the distance between $\{a, c, b, d\}$ and $\{a, b, c, d\}$ is the same as that between $\{d, b, c, a\}$ and $\{d, c, b, a\}$. Wu considers such symmetries to be a serious problem (p. 52). I cannot detect the problem since the similarity between $\{a, b, c, d\}$ and $\{d, c, b, a\}$ is, for a unit-cost OM algorithm, exactly 0.

I’m afraid that these objections will never be answered in a satisfactory way. Such objections apply not only to OM algorithms in

particular but also to any implementation of the concept of similarity since that concept itself presupposes symmetry: If A is similar to B, then B is similar to A and to the same degree. Such symmetry is inherent to the very concept of similarity, just as it is to concepts such as distance and covariation.

In fact, these objections are red herrings. Since if, in Wu's (2000) employment example, the subsequence $\{0, 1\}$ is so much different from the subsequence $\{1, 0\}$, then why encode the events "finding a job" and "losing one's job" in such a way that the difference is just in the precedence of two otherwise equivalent tokens? Why not encode "losing a job" as $\{lj\}$ and "finding a job" as $\{fj\}$? A similar argument pertains to Wu's "direction of time" objection.

Such problems should be properly handled in the encoding phase, when the researcher decides about what his or her data are and what they are about. Algorithms will never make such decisions.

Indeed, the importance of events is encoded in the data and the flow of time in the precedence of codes within sequences.

However, as argued above, objections to OM algorithms do not pertain to the potential usefulness of sequence analysis as a data reduction strategy: These objections pertain to just one of the items in the toolkit labeled *sequence analysis*. If we could replace this tool by another, to which the second objection does not apply, we still would have a potentially useful toolkit.

3. DT COEFFICIENTS

In 1995, Dijkstra and Taris presented an alternative to the OM algorithms that, curiously enough, received little attention. Abbott (1995) rightly criticized a particular aspect of it, and, unjustly, van Driel and Oosterveld (2001) commented on a supposed nonoptimality of their proposal.

In view of the debate we saw, it is at least remarkable that we have not seen any study yet (i.e., I couldn't find one) that employs this alternative.

Dijkstra and Taris (1995) make three slightly different proposals on how to quantify sequence similarity. Although their proposals suffer from the same shortcomings, it is worthwhile to consider them in

some detail since the principles on which they are constructed are very useful and comprise a substantial leap in the right direction.

Dijkstra and Taris (1995) start² from what I consider to be four simple “axioms” that are hard to reject since these lie at the heart of what we consider to be sequences and what makes them similar or dissimilar:

- DT 1: Sequences that have no tokens in common are maximally dissimilar.
- DT 2: Sequences that have the same elements in the same order are maximally similar.
- DT 3: The more tokens that sequences have in common, the more similar these sequences are.
- DT 4: The more common order there is among common tokens, the more similar sequences are.

Given two sequences—say, $\mathbf{x} = \{a, b, b, c, d\}$ and $\mathbf{y} = \{b, a, c, c\}$ with sequence lengths $l_x = 5$ and $l_y = 4$, respectively—Dijkstra and Taris (1995) then implement these axioms by first discarding all tokens that are noncommon to the sequences involved. Furthermore, from the common tokens, they discard all but one of the possible repetitions from each of the sequences.³ So, they are then left with two reduced sequences, $\mathbf{x}^* = \{a, b, c\}$ and $\mathbf{y}^* = \{b, a, c\}$, of equal length and containing the same tokens but not necessarily in the same order. Then they count the number $C_{\mathbf{x}^*, \mathbf{y}^*}$ of ordered pairs common to these reduced sequences and normalize this count to the maximum number of common ordered pairs given the length l of the reduced sequences. This maximum will be obtained if the reduced sequences are identical. Since the sequences are reduced sequences, they comprise l different tokens, from which one can pick precisely $\binom{l}{2}$ different ordered pairs. If the sequences are identical, each pair of tokens from the one sequence will have one and only one matching counterpart in the other sequence. Therefore, this maximum equals $\binom{l}{2}$. Then Dijkstra and Taris compute

$$r_{\mathbf{x}^*, \mathbf{y}^*} \equiv \frac{C_{\mathbf{x}^*, \mathbf{y}^*} + 1}{\binom{l}{2} + 1} \quad (1)$$

as a measure of similarity between the reduced sequences \mathbf{x}^* and \mathbf{y}^* and define the coefficients

$$\begin{aligned}\alpha_{x,y} &\equiv \frac{l^2}{l_x \cdot l_y} r_{x^*,y^*}, \\ \beta_{x,y} &\equiv \frac{l^2}{l_x + l_y - 1} r_{x^*,y^*}, \quad \text{and} \\ \gamma_{x,y} &\equiv \frac{l}{l_x + l_y} r_{x^*,y^*}\end{aligned}\tag{2}$$

to quantify the similarity between the original sequences. Obviously, the different DT coefficients differently use the original sequence lengths and the number of discarded tokens. The important point is that each of them uses the concept of order in the same way.

At first sight, each of these coefficients is a perfect implementation, although not the only one possible, of the axioms mentioned above. Abbott (1995) commented on these proposals by stating that discarding tokens as a new type of edit operation could perhaps be justified if the sequences have a common “backbone” structure, enriched or contaminated by “extraneous” tokens. But this, of course, is only true when the hypothesized backbone itself does not contain repetitions of tokens.

However, the real point is not the question of discarding or retaining tokens. Instead, the important issue is the meaning of the word *order* in the axioms DT 2, DT 3, and DT 4. In the next section, we use a different interpretation of this concept to quantify similarity anew.

4. SIMILARITY AND PRECEDENCE

In mathematics, the concept of (strict or strong) order refers to a binary relation, defined on a Cartesian product set, that satisfies the properties of irreflexivity, asymmetry, and transitivity.

Generally, this concept does not apply to the order we have in mind when dealing with sequences. What we have in mind when dealing with order in sequences is a binary relation that is best called *precedence*: We say that token a precedes token b in a sequence \mathbf{x} whenever we, when reading from left to right, first encounter token a

and later encounter token b . For short, we write aPb when we observe that token a precedes token b . Note that aPb does not necessarily imply that a and b are consecutive elements; they may be several positions apart.

We extend the notion of precedence a little by applying it to itself: We write, for example, $(aPb)Pc = aPbPc$ to describe precedences as they occur within a triple of tokens. In sequences wherein tokens repeat themselves (i.e., the same token occurs on two or more different positions in one and the same sequence), we have, for example, aPa . Therefore, the binary relation P is reflexive.

Also, we may encounter a (sub)sequence of the form $\{a, b, c, a\}$, which means that we have aPb , bPc , but also cPa , the latter fact implying that the relation P is not only reflexive because of aPa but also intransitive. Furthermore, we have both⁴ aPc and cPa ; hence, P is symmetric. Therefore, within the context of sequence analysis or sequence similarity, the concept of order generally refers to a binary relation that is reflexive, symmetric, and intransitive. Only in the rare cases in which repetitions of tokens do not and cannot occur does P have the properties of a mathematical (strict) order.

Only when P has the properties of a strict order do all “higher order” precedences such as $aPbPc$ directly follow from the complete set of “simple” precedences of the form aPb .

If we interpret the word *order* in the axioms DT 2, DT 3, and DT 4 as referring to a relation P that is, in general, reflexive, symmetric, and intransitive, it is immediate that the DT coefficients force P to be a strict order in the mathematical sense by discarding everything that violates irreflexivity, asymmetry, and/or transitivity. Below, we will see that this produces similarity rankings of pairs of sequences that Dijkstra and Taris (1995) probably did not intend at all.

Apparently, we are left with the DT coefficients as tools that could be useful in the very unlikely situation in which repetition of tokens within sequences cannot occur.

5. SIMILARITY AS ENUMERATING COMMON PRECEDENCE

Defining a precedence relation for a set of tokens is exactly what generates a sequence. Furthermore, if we interpret *order* in the DT

axioms as *precedence*, these axioms are hard to reject as a basis for constructing a similarity coefficient.

It is therefore tempting to see if we can use these concepts as a basis to quantitatively define sequence similarity. In doing so, we use a somewhat formal language to stress the importance and power of the concept of a precedence relation and to clarify where we do and do not make arbitrary choices.

Let $X = \{\mathbf{x}, \mathbf{y}, \dots\}$ be a set of sequences, constructed from an alphabet $A = \{a, b, c, \dots\}$. We write $aP_x b$ if, in sequence \mathbf{x} , we encounter aPb . Let $s_{x,y}$ be a function from $X \times X$ onto the nonnegative real numbers. If $s_{x,y}$ is to be interpreted as a similarity index, it is convenient to choose it such that

$$0 \leq s_{x,y} \leq 1 \quad (3)$$

So, if we adhere to the DT axioms, we must have at least

$$x \cap y = \emptyset \Leftrightarrow s_{x,y} = 0, \quad (4)$$

$$s_{x,x} = 1, \quad (5)$$

because of DT 1 and DT 2. Now we have to use the full set of axioms to see if we can specify at least one $s_{x,y}$ that satisfies equations (3) through (5) and the axioms themselves. Therefore, we first define an ordered k -tuple \mathbf{x}_k from a sequence \mathbf{x} with length l_x as a subsequence from \mathbf{x} , containing exactly k tokens with $k \geq 1$. Intentionally, we do not presume that $k \leq l_x$ but instead adopt the convention to say that the number of different k -tuples from \mathbf{x} with $k > l_x$ equals zero. It is important to stress the point that the k -tuples are ordered, meaning that if we have, for any pair of tokens, $a, b \in A$, $a, b \in \mathbf{x}$, and $a, b \in \mathbf{x}_k$, it implies $aP_x b \Leftrightarrow aP_{\mathbf{x}_k} b$.

Now we say that two sequences \mathbf{x} and \mathbf{y} have a matching or common k -tuple \mathbf{x}_k , precisely when the k -tuple could have been taken both from sequence \mathbf{x} and from sequence \mathbf{y} . Let $m_{x,y}(k)$ denote the number of matching k -tuples of \mathbf{x} and \mathbf{y} for any positive value of k . If $k = 1$, then $m_{x,y}(1)$ simply counts the number of tokens shared by \mathbf{x} and \mathbf{y} ; if $k = 2$, $m_{x,y}(2)$ counts the number of pairs of tokens appearing in both \mathbf{x} and \mathbf{y} with the property that $aP_x b \Leftrightarrow aP_y b$, etc. (note that $m_{X,y}(2) = c_{X^*,y^*}$ when \mathbf{x} and \mathbf{y} do not contain repeating tokens).

There is nothing preventing us from considering the quantity $m_{x,x}(k)$: It simply counts the number of matching k -tuples when comparing x with itself. Obviously, if x does not contain repeated tokens, we have $m_{x,x}(k) = \binom{l_x}{k}$.

Now clearly, DT 3 and DT 4 require that

$$s_{x,y} = F\{M_{x,y}(1), m_{x,y}(2), \dots\}, \quad (6)$$

where F is a function that is strictly increasing in each of its arguments. Unfortunately, there is nothing in these axioms that further specifies the properties of F . So, from this point on, all decisions on properties of F are arbitrary but valid as long as it is ascertained that equations (3) through (6) are satisfied.

On the other hand, (3) restricts F to be a mapping onto $[0, 1]$ and, together with (5), suggests that we relate the quantities $m_{x,y}(k)$ to what is maximally attainable in the case of identical sequences, that is, to specify F as, for example,

$$H \left[l_x, l_y, \frac{m_{x,y}(1)}{f\{m_{x,x}(1), m_{y,y}(1)\}}, \frac{m_{x,y}(2)}{f\{m_{x,x}(2), m_{y,y}(2)\}}, \dots \right], \quad (7)$$

where the function f is increasing in both of its arguments and such that it satisfies $\min\{u, v\} \leq f\{u, v\} \leq \max\{u, v\}$.

Now one obvious (but arbitrary) solution is to choose $f\{u, v\} = (u \cdot v)^{\frac{1}{2}}$ and H such that

$$s_{x,y} = \sum_{k=1}^L \frac{m_{x,y}(k)}{\sqrt{m_{x,x}(k) \cdot m_{y,y}(k)}} / L \quad (8)$$

where $L = \max\{l_x, l_y\}$.

Let us elaborate a little on this solution. To begin with, suppose that $l_x = l_y = L$ and that neither sequence contains repeated tokens. Then (8) reduces to

$$s_{x,y} = \sum_{k=1}^L \frac{m_{x,y}(k)}{\binom{L}{k}} / L \quad (9)$$

since then the precedence relation P has the properties of a strict order, and the number of differently ordered k -tuples that can be taken from each of the sequences clearly amounts to $m_{x,x}(k) = \binom{L}{k} = m_{y,y}(k)$. Thus, dividing each summand in (9) by $\binom{L}{k}$ ensures that the value of each summand will have a value somewhere between 0 and 1. Then, dividing by L forces $0 \leq s_{x,y} \leq 1$. When the sequences are identical, we will obtain $m_{x,y}(k) = \binom{L}{k}$ for each k (hence $s_{x,y} = 1$), and when the sequences do not have even one common token, each $m_{x,y}(k)$ will be equal to zero.

If the sequences are of unequal length (say, l_x and l_y with $l_x \neq l_y$), it is not immediately clear how to balance between $\binom{l_x}{k}$ and $v = \binom{l_y}{k}$ when normalizing the summands $m_{x,y}(k)$ to a maximum.

We solve this by arbitrarily choosing $f\{u, v\} = (u \cdot v)^{\frac{1}{2}}$, which balances for different lengths of the sequences.

Finally, the particular choice of L as an overall normalizer is arbitrary (remember that $m_{x,y}(k) = 0$ for at least $k > \min\{l_x, l_y\}$).

As soon as one considers the possibility that either or both x and y contain repeating tokens, then one could try to write explicit, closed expressions for $m_{x,x}(k)$ and $m_{y,y}(k)$. Such expressions will turn out to be extremely complicated, even for quite simple cases with few repeating tokens. All that such expressions do is illustrate that one primarily needs an efficient algorithm to actually compute $m_{x,x}(k)$ in the case of repeating tokens. The discussion of such an algorithm is postponed to the last section; for the moment, we just take the computability of $m_{x,x}(k)$ and $m_{x,y}(k)$ for granted in all cases (i.e., for sequences with or without repeating tokens). For short sequences with or without repeating tokens, $s_{x,y}$ is easily calculated with a paper and pencil.

Clearly, (8) is an expression that satisfies all the requirements that we could put up for an acceptable similarity index without speculating about the specific properties of the sequences involved or the processes that generated them. As can be expected under such weak conditions, (8) is arbitrary to a considerable extent. As will become

apparent in the next sections, it does serve the purpose of revealing characteristics of sequences much better than either the DT coefficients or an OM algorithm.

6. SOME EXAMPLES

To maintain some continuity in the ongoing discussion about sequence analysis and sequence similarity, we choose to use the same example sequences and the same sequence data that Dijkstra and Taris (1995) used. To illustrate some properties of their coefficients, Dijkstra and Taris used five example sequences that they took from Abbott and Hrycak (1990). These sequences are as follows:

- 1** = {1, 1, 2, 3, 4, 5, 6}
- 2** = {1, 2, 3, 4, 5, 6, 7, 8}
- 3** = {1, 1, 4, 5, 7, 8, 7, 7}
- 4** = {4, 5, 6, 7, 7, 7, 7, 7}
- 5** = {1, 2, 3, 4, 5}

Dijkstra and Taris (1995) then calculated the similarity for each of the 10 pairs of sequences, both with a unit-cost OM algorithm and with each of their coefficients. Here we reproduce a part of their results plus the calculated values of $s_{x,y}$ in Table 1.

TABLE 1: Similarity Indices Compared

Pairs:	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(2, 3)	(2, 4)	(2, 5)	(3, 4)	(3, 5)	(4, 5)
OM	0.625	0.250	0.000	0.714	0.375	0.125	0.625	0.375	0.250	0.000
γ	0.800	0.533	0.400	0.833	0.625	0.500	0.769	0.625	0.462	0.308
$s_{x,y}$	0.355	0.095	0.029	0.468	0.191	0.114	0.277	0.218	0.083	0.024

More interesting than the numbers themselves are the orderings⁵ of the pairs that different methods achieve. To show these orderings, we write, for example, $(1, 2) > (1, 3)$ if a particular method assigns more similarity to pair (1, 2) than it does to (1, 3), and we write $\begin{pmatrix} 1, 2 \\ 1, 3 \end{pmatrix}$

if a particular method cannot distinguish between the pairs. So, from Table 1, we find the orderings as

$$\text{OM} : (1, 5) > \begin{pmatrix} 1, 2 \\ 2, 5 \end{pmatrix} > \begin{pmatrix} 2, 3 \\ 3, 4 \end{pmatrix} > \begin{pmatrix} 1, 3 \\ 3, 5 \end{pmatrix} \\ > (2, 4) > \begin{pmatrix} 1, 4 \\ 4, 5 \end{pmatrix},$$

$$\gamma : (1, 5) > (1, 2) > (2, 5) > \begin{pmatrix} 2, 3 \\ 3, 4 \end{pmatrix} \\ > (1, 3) > (2, 4) > (3, 5) > (1, 4) > (4, 5),$$

$$s : (1, 5) > (1, 2) > (2, 5) > (3, 4) > (2, 3) > (2, 4) \\ > (1, 3) > (3, 5) > (1, 4) > (4, 5).$$

The ties in the ordering of the similarities as produced by the OM algorithm are remarkable. For example, OM generates $\begin{pmatrix} 1, 2 \\ 2, 5 \end{pmatrix}$, implying that **2** is as similar to **1** as it is similar to **5**. Clearly, the difference between the sequences **2** and **5** is three noncommon tokens, and the difference between **1** and **2** is two noncommon tokens and one noncommon repetition of a shared token. Apparently, OM is not too sensitive about small differences. $s_{x,y}$ disentangles the ties in the orderings as generated by the other indices: The closer one looks at things, the less similar they appear!

Apart from the fact that the methods produce different numbers of ties in the orderings, they also produce reversals of orderings when compared to each other, even if we accept ties as weak orderings. A reversal of similarity orderings implies quite different spatial representations of the sequences: OM generates $(3, 5) > (2, 4)$, implying that, spatially, **3** is closer to **5** than **2** is close to **4**, whereas γ and $s_{x,y}$ generate the opposite $(2, 4) > (3, 5)$.

These orderings are what really matters since scaling algorithms and clustering procedures decide on the basis of the spatial content of the information they are fed with and interpret ties and intransitivities such that some measure of fit or stress is optimized. Therefore, the planned data reduction in the last phase of a sequence analysis crucially depends on precisely this kind of information.

7. REAL DATA

To confront $s_{x,y}$ with some real data, we use the same sequence data that Dijkstra and Taris (1995) used. Essentially, these data comprise 494 sequences of life history events, taken from young adults, in three variables: living situation, education, and employment. The categories and codes are given in Table 2.

TABLE 2: Encoding Scheme of Life History Data

<i>Variable</i>	<i>Category</i>	<i>Code</i>
Living	With parents	H
	Alone	S
	With partner	P
	Married	M
	Other	O
Education	Full-time	F
	Part-time	P
	None	O
Employment	Full-time	F
	Part-time	P
	None	O

Thus, each sequence consists of three-character tokens such as HOF, SFO, and so on. Elementary characteristics of these sequences are given in Table 3 in terms of sequence lengths.

TABLE 3: Sequence Length Characteristics

	<i>Mean</i>	<i>Variance</i>	<i>n</i>
Females	8.61	9.49	244
Males	8.70	9.95	250
Total	8.66	9.72	494

Below, we present a more or less random sample of 10 sequences that happened to be consecutive in the data set. Note the high variation in the sequence lengths.

HFO HOO HOP HOO
 HFO HFP HOP HPP HOP HOO HOF HPF HOF SOF
 HFO HOO OOO OFO MFO MOO
 HFO SFO PFO POO POF
 HFO HOF HPF HOF POF MOF MPF MOF MPF MOF
 HFO SFO SOO
 HFO HOO HOF SOF POF
 HFO HOO HPO HPF HOF HPF HOF POF PPF HPF HOF SOF
 HOF SPF
 HFO OFO OFP SFP
 HFO HOO HFO HOO HOF SOF POO OOO PPO FPO FOO FPO
 FPF FPO FOO

To get a feel for these data, we take a closer look at the sequence {HFO, HOO, HOF, SOF, POF}, presuming the subject it represents is male. This person apparently starts living with his parents and going to school at least four days a week. Then he stops going to school while still living at home. After a while, he finds a job for at least four days per week but keeps living with his parents. Then he moves from the parental home to start living on his own, still being employed full-time. Finally, the situation again changes when he finds a partner to live with.

For Dijkstra and Taris (1995), a main research question was whether life histories of males and females were different. To find out, they identified, separately for males and females, the sequence with highest mean γ with all other sequences. We reproduce the essentials of their results in Table 4 and added the characteristic sequence for the whole group.

TABLE 4: Characteristic Sequences According to γ

	<i>Characteristic Sequence</i>	<i>Mean γ</i>
Females	HFO HOO HOF POF MOF MOO	0.497
Males	HFO HOO HOF HOO HOF POF MOF	0.521
Total	HFO HOO HOF POF MOF	0.502

With OM instead of γ , one obtains essentially the same results. Looking at Table 3 and the reproduced sample of 10 sequences above,

one is inclined to think that these sequences cannot be very similar. It is therefore amazing to observe that the mean γ in Table 4 roughly amounts to 0.5 on a scale ranging from 0 to 1.

We did the same to the data set: identify the sequence with the highest mean $s_{x,y}$, first for the total set of males and females, then separately for males and females. The results are shown in Table 5.

TABLE 5: Characteristic Sequences According to $s_{x,y}$

	<i>Characteristic Sequence</i>	<i>Mean γ</i>
Females	HFO HOO HOF MOF MOO	0.189
Males	HFO HOO HOF MOF	0.184
Total	FHO HOO HOF MOF	0.179

Dijkstra and Taris (1995:223) called the sequence [HFO, HOO, HOF, MOF] a “traditional life history.” However, it does not show up in these data as the sequence with highest average γ , neither for the group as a whole nor for the separate groups of males and females. Table 5 clearly shows that for the group as a whole and for the males separately, this is exactly the pattern with the highest average $s_{x,y}$. On the other hand, Dijkstra and Taris do report the extra token MOF at the end of the characteristic sequence for females, just as is found with $s_{x,y}$.

The “small” difference between the characteristic sequences for males and females also appears in Table 6: simple ANOVA results with gender as independent variable and the similarity between sequences and the “characteristic female life history” $z = \{\text{HFO, HOO, HOF, MOF, MOO}\}$ as the dependant variable. So, for each subject/sequence x , we computed $s_{x,z}$, and these similarities served as the dependant variable in the analysis of variance.

Of course, we know nothing about the sample distribution properties $s_{x,y}$, so the F statistic in Table 6 should not be taken too seriously. We just employ the ANOVA scheme because Dijkstra and Taris (1995) used it; strictly, a nonparametric scheme would have been more appropriate. However, the mean $s_{x,y}$ with the characteristic female pattern for males is 0.147 (cf. Table 5).

TABLE 6: ANOVA with $S_{x,y}$ for Characteristic Female Life History

	<i>Sum of Squares</i>	<i>df</i>	<i>Mean Square</i>	<i>F</i>
Groups	0.170	1	0.170	6.38
Error	13.094	492	0.027	
Total	15.263	493		

The principal result of the above is that the similarity index $s_{x,y}$, as defined by (8), not only satisfies what formal requirements can be put up but also seems to be a useful exploratory tool in that it unveils patterns that the DT coefficients or a unit-cost OM algorithm do not.

Apart from the life history sequence of each individual and his or her gender, there is also an assessment of socioeconomic status (SES) on a 10-point scale for each individual. Dijkstra and Taris (1995) then used one-way ANOVA to demonstrate that life histories of subjects with lower SES scores are more similar to the “traditional life history” than the life histories of subjects with a higher SES score. We performed the same analysis, only using $s_{x,y}$ as a similarity index. The results are shown in Table 7.

TABLE 7: Similarity with Traditional Life History Per Socioeconomic Status (SES) Group

<i>SES</i>	0	1	2	3	4	5	6	7	8	9	<i>Total</i>
<i>n</i>	4	102	70	95	82	58	43	28	10	2	494
Mean $s_{x,y}$.273	.246	.209	.201	.165	.130	.120	.079	.086	.31	.181
Variance $s_{x,y}$.040	.049	.032	.038	.023	.013	.013	.008	.003	.000	.032
	Sum of Squares	<i>df</i>	Mean Square	<i>F</i>							
	1.319	9	0.147	4.88							
Error	14.538	484	0.030								
Total	15.857	493									

The data that we just discussed consist of three parallel sequences: status of household, education, and employment. Dijkstra and Taris (1995) defined *life history events* as a change in any of these variables, and we analyzed the so-defined life history sequences. Hence,

TABLE 8: Order Reversals with Various Similarity Coefficients

	α	β	γ	<i>Optimal Matching</i>	$s_{x,y}$
(x, y)	0.80	0.80	0.89	0.80	0.55
(z, y)	0.71	0.71	0.83	0.71	0.58

the above analysis will certainly have missed any patterns possibly present in each of the separate sequences. Since the main purpose of this analysis has been primarily to demonstrate our approach to and quantification of similarity, we do not consider this to be a complication.

A final example nicely illustrates the effect that discarding of tokens may have. Consider the following triple of sequences:

$$\begin{aligned} \mathbf{x} &= \{a, b, c, d\} \\ \mathbf{y} &= \{a, b, c, d, e\} \\ \mathbf{z} &= \{a, b, c, b, c, d, e\} \end{aligned}$$

In Table 8, the similarities between the pairs (x, y) and (z, y) are presented according to the various indices discussed so far.

Coincidentally, α , β , and OM produce the same results. Clearly, only for s do we have $(x, y) < (z, y)$.

At first sight, the result obtained with $s_{x,y}$ may seem counterintuitive: Perceptually, y seems to be more similar to x than to z . However, closer inspection reveals that the difference between x and y is a noncommon token e , implying many noncommon precedences. On the other hand, the difference between y and z is the occurrence of cPb and bPc in z , while only bPc occurs in y . Therefore, the result is in perfect accordance with the DT axioms.

Sequences with patterns identical or akin to those of this example frequently occur in the data set: the triple

$$\begin{aligned} \mathbf{x} &= \{\text{HFO, HOO, HOF, MOF}\} \\ \mathbf{y} &= \{\text{HFO, HOO, HOF, MOF, MOO}\} \\ \mathbf{z} &= \{\text{HFO, HOO, HOF, HOO, HOF, MOF, MOO}\} \end{aligned}$$

was easily fished out of the female life histories. Similar examples from this data set are abundant.

The difference between the women x and y is that y stopped working some time after becoming married. This is a decision made frequently by married young women, especially by those with lower educational levels (this is to be considered a speculative and suggestive remark since I know nothing about the woman represented by y).

The difference between the women represented by z and y is that z appears to have been unemployed for some time while living at home, and then she again finds a job.

I am not an expert in life histories. But to me, it seems only natural that we consider the life histories represented by x and y as being less similar than those represented by z and y .

Of course, this example does not prove that $s_{x,y}$ is a better similarity index than γ or any other index: Such proof (or a disproof) will never be found in any example. Rather, the example shows a consequence of an implicit theory of the data encoded: The encoder decided that the subsequence $\{HOO, HOF, HOO, HOF\}$ is, for some reason, substantially different from the subsequence $\{HOO, HOF\}$. In this context, Table 8 suits us well; it might not have been so if the meaning of the tokens in the sequences had been different. If one accepts an index such as $s_{x,y}$ as a reasonable quantification of similarity, counterintuitive results should focus our attention on the data encoding and its theory.

The example from Table 8 also nicely demonstrates that perceptual similarity between token sequences can, apparently, be quite different from similarity, as implied by the DT axioms.

By now, the potential of $s_{x,y}$ has been amply and sufficiently demonstrated, so it seems worthwhile to devote a section to an algorithm to compute it.

8. COMPUTING $s_{x,y}$

According to equation (8), computing $s_{x,y}$ seems a quite formidable task. Indeed, if one counts all the matching k -tuples by listing each and every k -tuple from a sequence and looking for a match among all the k -tuples from the other sequence, one faces a staggering amount of comparison and counting. Algorithmically, this problem has a complexity that is roughly exponential with sequence length.

Therefore, we describe a simple algorithm⁶ that avoids most of the listing and counting.

Let $\mathbf{x} = \{x_1, x_2, \dots\}$ and $\mathbf{y} = \{y_1, y_2, \dots\}$ be two sequences of lengths l_x and l_y , respectively.

We start by listing all the pairs of matching elements from \mathbf{x} and \mathbf{y} in a matrix \mathbf{Z} with two columns and at most $(l_x \times l_y)$ rows, defined by $[z(k, 1) = i \ \& \ z(k, 2) = j] \Leftrightarrow x_i = y_j$.

Next, we define an $l_x \times l_y$ -matrix $\mathbf{H}_1 = \{h_1(i, j)\}$, such that $h(i, j) = 1 \Leftrightarrow x_i = y_j$ and $h(i, j) = 0$ otherwise. Indeed, at the start of the algorithm, \mathbf{Z} and \mathbf{H}_1 do contain the same information. Obviously, $\sum_{i,j} h_1(i, j)$ equals the number of matches when each and every element of \mathbf{x} is compared with each and every element from \mathbf{y} .

From the matrix \mathbf{H}_1 , we construct a new $l_x \times l_y$ -matrix $\mathbf{V}_1 = \{v_1(i, j)\}$ through

$$v_1(i, j) = \sum_{a>i, b>j} h_1(a, b)$$

A small example is instructive: Let $x = \{a, a, b, c, b, d\}$ and $y = \{a, b, a, d, c\}$. Evidently, $l_x = 6$ and $l_y = 5$, and we have

$$\mathbf{Z}^T = \begin{pmatrix} 1 & 1 & 2 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 1 & 3 & 2 & 5 & 2 & 4 \end{pmatrix},$$

$$\mathbf{H}_1 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \text{ and } \mathbf{V}_1 = \begin{pmatrix} 5 & 3 & 2 & 1 & 0 \\ 4 & 2 & 2 & 1 & 0 \\ 3 & 2 & 2 & 1 & 0 \\ 2 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$m_{x,y}(1) = \sum_{i,j} h_1(i, j) = 8.$$

Now the elements $v_1(i, j)$ contain the number of matches found when comparing each and every element of \mathbf{x} with each element from \mathbf{y} , disregarding the first i elements of \mathbf{x} and disregarding the first j elements of \mathbf{y} .

If the pair $(i, j) \in \mathbf{Z}$ —that is, if (i, j) is a row of \mathbf{Z} —then we know that we can construct $v_1(i, j)$ matching 2-tuples from \mathbf{x} and \mathbf{y} , with the

first element of the 2-tuple from \mathbf{x} being the element x_i and the first element of the 2-tuple from \mathbf{y} being the element y_j .

Hence, if we sum those $v_1(i, j)$ with $(i, j) \in \mathbf{Z}$, we will obtain the number of matching 2-tuples from \mathbf{x} and \mathbf{y} . We implement this by defining a new $l_x \times l_y$ -matrix $\mathbf{H}_2 = \{h_2(i, j)\}$ by $h_2(i, j) = v_1(i, j) \Leftrightarrow (i, j) \in \mathbf{Z}$, and $h_2(i, j) = 0$ otherwise, and obtain $m_{x,y}(2) = \sum_{i,j} h_2(i, j)$.

From the matrix \mathbf{H}_2 , we define the new matrix \mathbf{V}_2 by

$$v_2(i, j) = \sum_{a>i, b>j} h_2(a, b).$$

For our example sequences, we thus obtain

$$H_2 = \begin{pmatrix} 5 & 0 & 2 & 0 & 0 \\ 4 & 0 & 2 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ and } V_2 = \begin{pmatrix} 5 & 2 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$m_{x,y}(2) = \sum_{i,j} h_2(i, j) = 16.$$

Now the $v_2(i, j)$ contain the number of matching 2-tuples in the sequences \mathbf{x} and \mathbf{y} , disregarding the first i elements of \mathbf{x} and disregarding the first j elements of \mathbf{y} . From each of these 2-tuples, we can construct two matching 3-tuples by adding the element $x_i \in \mathbf{x}$ to the one and the element $y_j \in \mathbf{y}$ to the other if $(i, j) \in \mathbf{Z}$.

So, we create \mathbf{H}_3 from \mathbf{V}_2 by $h_3(i, j) = v_2(i, j) \Leftrightarrow (i, j) \in \mathbf{Z}$, and $h_3(i, j) = 0$ otherwise, and obtain $m_{x,y}(3) = \sum_{i,j} h_3(i, j)$.

For our example sequences, we thus obtain

$$H_3 = \begin{pmatrix} 5 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad m_{x,y}(3) = 8$$

Again, from \mathbf{H}_3 , we construct \mathbf{V}_3 through

$$v_3(i, j) = \sum_{a>i, b>j} h_3(a, b).$$

For our example sequences, obviously each $v_3(i, j) = 0$, so no matching 4-tuples can be constructed.

In general, to obtain $m_{x,y}(k)$, we will create a matrix \mathbf{H}_k from the matrix \mathbf{V}_{k-1} by

$$h_k(i, j) = v_{k-1}(i, j) \Leftrightarrow (i, j) \in \mathbf{Z} \quad (10)$$

and create \mathbf{V}_k from \mathbf{H}_k through

$$v_k(i, j) = \sum_{a>i, b>j} h_k(a, b). \quad (11)$$

Obviously, the process will effectively terminate as soon as $m_{x,y}(k) = 0$ since if this is true for some k , $m_{x,y}(a)$ for all $a > k$.

In this basic form, the algorithm already achieves a tremendous reduction in labor when compared to the “listing and matching of k -tuples”: Its algorithmic complexity is now a third-degree polynomial of sequence length. Since, in an actual implementation, each \mathbf{H}_k will replace \mathbf{H}_{k-1} and each \mathbf{V}_k will replace \mathbf{V}_{k-1} , the algorithm will use only a very limited amount of memory.

Substantial further reduction in computation is achievable but is not presented for lack of space.

9. DISCUSSION

We constructed a similarity index that does not suffer from the shortcomings of OM algorithms or those of the DT coefficients: We do not need the classical edit operations or any discarding of tokens. On the other hand, we use all the information about the precedence of tokens: that which turns a set of tokens into a sequence. Furthermore, and because of the latter, we showed that this index is quite well discriminating between pairs of sequences (Table 1) and discriminating in the right way (Table 8).

Apparently, the index $s_{x,y}$ does a nice job in the reanalysis of the DT data in that it “spontaneously” uncovers the “traditional life history” that Dijkstra and Taris (1995) hypothesized.

Finally, we presented an algorithm that makes $s_{x,y}$ readily computable, although perhaps special measures have to be taken in

the case of long sequences because of the colossal numbers that will then arise as entrances of the matrices \mathbf{H}_k and \mathbf{V}_k .

We also stressed the fact that the specific mold of (8) represents an aggregate of choices that are partly arbitrary. An alternative not discussed so far could be

$$\frac{\sum_k m_{x,x}(k)}{\sqrt{\sum_k m_{x,x}(k) \sum_k m_{y,y}(k)}}. \quad (12)$$

At first sight, this alternative definitely appeals, not in the least because of its resemblance to Pearson's $r_{x,y}$. A little thought, however, reveals that since its denominator is such a big number, it would hardly discriminate between pairs of sequences that have lots of matching k -tuples for lower values of k .

An intermediate between (8) and (12) could be

$$\sum_{k=1}^L \frac{k \cdot m_{x,y}(k)}{\sqrt{m_{x,x}(k) \cdot m_{y,y}(k)}} / \sum_{k=1}^L (k^{-1}). \quad (13)$$

Since lots of matching k -tuples are impossible without many matching $(k-1)$ -tuples, we feel that there should be a very good theory about the sequences involved to justify it.

A natural generalization of (8) is given by

$$s_{x,y} = \sum_{k=1}^L \frac{w(k) \cdot m_{x,y}(k)}{\sqrt{m_{x,x}(k) \cdot m_{y,y}(k)}} / \sum_{k=1}^L (W(k)^{-1}). \quad (14)$$

with

$$w(k) = \begin{cases} 1 & \Leftrightarrow 1 \leq k \leq K \\ 0 & \Leftrightarrow k > K \end{cases}.$$

Of course, there is a much simpler equivalent to (14), but I present it in the above form to allow for other choices of the set $\{w(k)\}$. There are circumstances when at least the use of the weights as specified in (14) could be well justified. Consider the case when the researcher has a good theory about the sequences, implying that a particular pattern—say, $\{a, b, a, c, d, p\}$ —should best characterize his or her data. In such a case, there would be good reason not to consider k -tuples for $k > 6$ and take them along in the calculation of (8). Considering k -tuples for $k > 6$ could even prevent the researcher from finding that

$\{a, b, a, c, d, p\}$ is a good template. Hence, the choice of $K = 6$ could be well defensible.

There is one aspect⁷ of sequence analysis that we have not dealt with yet. This is the aspect of duration of states within sequences (i.e., the number of time units a certain state is kept “occupied”). For example, if state a is occupied for two uninterrupted units of time, we might encode this fact by having two adjacent a s in the sequence.

Consider the sequences and similarities as shown in Table 9.

TABLE 9: Similarity According to $S_{x,y}$ (Lower Diagonal) and Unit-Cost Optimal Matching (Upper Diagonal) for the Sequences in the First Column

$[a, b, c]$	1	.750	.750	.750
$[a, a, b, c]$.668	1	.750	.750
$[a, b, b, c]$.668	.575	1	.750
$[a, c, b, c]$.565	.506	.478	1

These sequences share all their tokens; only in the second and third sequences do we have two identical, adjacent tokens, meaning that we have states occupied for two units of time instead of one unit of time. In the lower diagonal part of the table, we show the values of $s_{x,y}$ according to (8); in the upper diagonal part, we present the similarities according to a unit-cost OM algorithm.

The OM algorithm considers each of the sequences 2, 3, and 4 as equally similar to each other and equally similar to $[a, b, c]$. Apparently, this OM algorithm completely overlooks the duration aspect that should make $[a, a, b, c]$ and $[a, b, b, c]$ more similar to each other than to $[a, c, b, c]$. So, quite complicated cost functions are necessary to make an OM algorithm behave in this kind of application. Instead, $s_{x,y}$ exactly does what one hopes for in this case. This small example demonstrates that $s_{x,y}$ is potentially useful in describing duration patterns. A more elaborate and precise discussion of this topic is beyond the scope of this article.

Note that in discussing alternatives to (8), we almost unavoidably crossed the thin line between an exploratory or descriptive use of sequence analysis and a kind of use that is much closer to inferential application and hypothesis testing. Of course, sequence analysis will never corroborate any model: Descriptive techniques never do. However, models and hypotheses do not come out of the blue. They arise

from carefully exploring and relating all kinds of data. So, we will never do without descriptive techniques.

But these descriptive techniques should be as free as possible from any assumption about the data or their representation. OM-like algorithms are clearly not free of such assumptions: Their cost matrices imply the choice of particular spatial representations over and above others. Neither are the DT coefficients free of such assumptions. They presuppose or force the precedences to adhere to the properties of a strict order, and if they do not, this is accounted for in a totally different aspect of the sequences: their length.

The present approach does not assume any property of the precedences or anything about a spatial representation of the sequences. It just uses all of the information on precedences within sequences.

Therefore, we hope that a simple tool such as $s_{x,y}$ will provide sequence analysts with some new prospects to find their “decisive” application and take away from their opponents at least one reason to condemn sequence analysis as a strategy to learn something about social reality.

NOTES

1. In discussing the Dijkstra and Taris (DT) coefficients, I will use formulas and notation that are equivalent but not identical to those of Dijkstra and Taris (1995) to create some consistency of notation in this article.

2. Dijkstra and Taris (1995) never mentioned axioms. I translated their arguments into the axioms mentioned here.

3. Dijkstra and Taris (1995) called such repetitions “superfluous codes” and decided on which one to remove according to an optimality criterion. It was exactly on this point that van Driel and Oosterveld (2001) commented. Their criticism is unjustified but understandable: Dijkstra and Taris are not too explicit about their algorithm on this point, and Dijkstra (2001), in his comment on van Driel and Oosterveld, is even less clear about the issue.

4. Some authors (e.g., Wells 1971) would call our precedence relation P an *inconsistent* precedence relation since it is reflexive and symmetric.

5. Here, we use the word *order* as pertaining to an ordering of similarity indices, not as pertaining to order or precedence of tokens within single sequences. So, here the word *order* should be interpreted in the normal sense of a transitive, reflexive, and asymmetric relation.

6. Upon request (by e-mail to CH.Elzinga@fsw.vu.nl), a free and simple computer program is available as an executable. The program is written in Microsoft’s Visual C#.NET. Input to the program is a set of sequences, and output is a matrix of coefficients $s_{x,y}$ and/or a set of coefficients $s_{x,z}$, where z is a user-defined template sequence. The program will be e-mailed with a short user manual and a tutorial file of sequences.

7. The issue of duration and these example sequences were suggested to me by an anonymous reviewer.

REFERENCES

- Abbot, A. 1995. "A Comment on 'Measuring the Agreement Between Sequences.'" *Sociological Methods & Research* 24 (1): 232-42.
- . 2000. "Reply to Levine and Wu." *Sociological Methods & Research* 29 (1): 65-76.
- Abbot, A. and J. Forrest. 1986. "Optimal Matching Methods for Historical Sequences." *Journal of Interdisciplinary History* 15:471-94.
- Abbot, A. and A. Hrycak. 1990. "Measuring Resemblance in Sequence Data: An Optimal Matching Analysis of Musicians' Careers." *American Journal of Sociology* 96:144-85.
- Abbot, A. and A. Tsay. 2000. "Sequence Analysis and Optimal Matching Methods in Sociology: Review and Prospect." *Sociological Methods & Research* 29 (1): 3-33.
- Dijkstra, W. 2001. "How to Measure the Agreement Between Sequences: A Comment." *Sociological Methods & Research* 29 (4): 532-35.
- Dijkstra, W. and T. Taris. 1995. "Measuring the Agreement Between Sequences." *Sociological Methods & Research* 24 (2): 214-31.
- Kruskall, J. B. 1983. "An Overview of Sequence Comparison." Pp. 1-44 in *Time Warps, String Edits and Macromolecules: The Theory and Practice of String Comparison*, edited by D. Sankoff and J. B. Kruskall. Reading, MA: Addison-Wesley.
- Levine, J. H. 2000. "But What Have You Done for Us Lately." *Sociological Methods & Research* 29 (1): 35-40.
- van Driel, K. and P. Oosterveld. 2001. "Nonoptimal Alignment: A Comment on 'Measuring the Agreement Between Sequences by Dijkstra and Taris.'" *Sociological Methods & Research* 29 (4): 524-31.
- Wagner, R. A. 1983. "On the Complexity of the Extended String-to-String Correction Problem." Pp. 211-14 in *Time Warps, String Edits and Macromolecules: The Theory and Practice of String Comparison*, edited by D. Sankoff and J. B. Kruskall. Reading, MA: Addison-Wesley.
- Wells, M. B. 1971. *Elements of Combinatorial Computing*. New York: Pergamon.
- Wu, L. 2000. "Some Comments on 'Sequence Analysis and Optimal Matching Methods in Sociology: Review and Prospect.'" *Sociological Methods & Research* 29 (1): 41-64.

Cees H. Elzinga obtained his Ph.D. as a psychologist in 1985. During the past 17 years, he has been active in the management of marketing and sales in various industries. In January 2002, he was appointed as a researcher with the Department of Social Research Methodology of the Faculty of Social Cultural Sciences at the Vrije Universiteit Amsterdam. His main interest lies in the study of interaction sequences as they arise in structured interviews.