An Evaluation of Substring Algorithms that Determine Similarity Between Surnames

G de V de Kock C du Plessis
Department of Computer Science, University of Port Elizabeth, P.O. Box 1600, Port Elizabeth, 6000

Abstract

The problem investigated in this study is, given a surname, determine similar surnames in a genealogical database. There exist a number of algorithms to determine the similarity between two strings based on their common substrings. The surnames in an existing genealogical database were used in an evaluation process to determine the relative success of these algorithms. The methods used to evaluate the performance of the algorithms and the algorithms are discussed briefly.

Keywords: genealogical database, word, string and surname matching

Computing Review Categories: H.5, I.1.7

1 Introduction

The primary problem is, given a surname, determine similar surnames in a genealogical database. More information on the origin of the problem is given in [2]. Other algorithms considered are given in [4] and [3]. In this paper applicable substring algorithms are briefly discussed, in some cases adapted and then evaluated. The norms and statistics used for the evaluation are also stated.

All the surnames in the UPE genealogical database at the time of the study formed the test dataset. Prepositions such as “de”, “van der”, “le” etc. have been dropped from the surnames in the test dataset.

Surnames with the same origin, spelling variations and aliases have been grouped together in mutually exclusive equivalence classes, the so-called ideal classes or partition.

A mathematical formulation of the primary problem follows:

Given a set of surnames, \( V \), (the test dataset)

\[ V = \{v_1, v_2, \ldots, v_n\} \]

and the ideal partition, \( P = \{V_1, V_2, \ldots, V_p\} \)

with \( V = \bigcup_{i=1}^{p} V_i \) and \( V_i \cap V_j = \emptyset \forall i \neq j \).

Given a surname, \( v \), determine:

1. \( i \) such that \( v \) is similar to the surnames in \( V_i \), or
2. the set of surnames in \( V \) which has a similarity to \( v \) greater than a predetermined value using some other norm (criterion).

The basic statistics of the test dataset are given in Table 1.

2 Success norms

The success norms discussed in [2] are stated briefly. Let \( A^* \) be the set of all possible strings over the alphabet \( A \), then \( V \subset A^* \). The similarity between two strings is usually a function, \( G : A^* \times A^* \rightarrow E \), where \( E \) is normally the interval \([0,1]\) on the real number line. Most of the similarity norms between two strings, \( u \) and \( v \), are of the form:

\[ G(u, v) = \frac{T(u, v)}{N(u, v)} \]

where \( T(u, v) \) is a function of the common substrings in \( u \) and \( v \), and \( N(u, v) \) is a normalising function to ensure that \( G(u, u) = 1 \). Furthermore, it is required that \( G(u, v) = G(v, u) \) and \( G(u, v) = G(u^R, v^R) \), where \( u^R \) is the reverse of \( u \).

A distance \( d = 1 - G \), can be defined between two strings. The function \( d \) is not necessarily a metric, since in most cases the triangular inequality is not satisfied.

For each algorithm evaluated, a centre \( c_i \) and a radius \( r_i \) are defined for each \( V_i \):

\[ r_i = \min_{w \in V_i} d(v, w) \]

The centre, \( c_i \), of \( V_i \), is the element of \( V_i \), such that

1. \( r_i = \max_{w \in V_i} d(c_i, w) \)
2. \( \{|w : w \in V, d(c_i, w) \leq r_i\} \cap (V - V_i)\} \), is a minimum.

Define the so-called “circle” with centre \( c_i \), and radius \( r_i + \gamma \), with \( \gamma \geq 0 \), as follows:

\[ C_i(\gamma) = \{w : w \in V \text{ and } d(c_i, w) \leq r_i + \gamma\} \]

Thus, \( V_i \subseteq C_i(\gamma) \forall i \).
Any set \( U = \{U_1, U_2, \ldots, U_p\} \), such that \( V_i \subseteq U_i, \forall V_i \in V \) is called a class cover or a \( C \)-cover of \( V \).

It follows that \( \bigcup_{i=1}^{p} U_i = V \).

For any \( \gamma \geq 0 \), \( C_\gamma = \{C_1(\gamma), C_2(\gamma), \ldots, C_p(\gamma)\} \) is a C-cover of \( V \). Here the discussion is restricted to \( U = C_0 \).

The success of the algorithm is defined as the percentage of the elements in \( V \) which appear in only one \( U_i \). It can formally be calculated in two ways, viz:

\[
S_1(U) = \frac{100}{n} \left( |V| - \left( \sum_{i=1}^{p} \sum_{j \neq i} |V_i \cap U_j| \right) \right) \quad (1)
\]

Secondly, the frequencies of the surnames are taken into account.

Let \( D_i = \{v : v \in \bigcup_{j \neq i} V_i \cap U_j \text{ and } v \in V_i\} \).

\( D_i \) is the set of elements of \( V_i \) which appear in other ideal classes’ “circles”.

\[
S_2(U) = \frac{100}{N} \left( N - \left( \sum_{i=1}^{p} \sum_{v \in D_i} \rho(v) \right) \right) \quad (2)
\]

**Table 1. Test dataset – constants**

<table>
<thead>
<tr>
<th>Explanation</th>
<th>Symbol/ formula</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>The number of different surnames in the test dataset</td>
<td>( n =</td>
<td>V</td>
</tr>
<tr>
<td>The number of ideal classes in the ideal partition</td>
<td>( p )</td>
<td>4,093</td>
</tr>
<tr>
<td>The average number of surnames per ideal class</td>
<td>( y_p = \frac{n}{p} )</td>
<td>1,75</td>
</tr>
<tr>
<td>The ideal classes with more than one element, i.e ( B = {V_i : V_i \in P \text{ and }</td>
<td>V_i</td>
<td>&gt; 1} )</td>
</tr>
<tr>
<td>The average number of surnames per class in ( B )</td>
<td>( y_B = \frac{\sum_{V_i \in B}</td>
<td>V_i</td>
</tr>
<tr>
<td>The number of surnames in the test dataset (( \rho(v) ) is the frequency of occurrence for a surname, ( v ))</td>
<td>( N = \sum_{v \in V} \rho(v) )</td>
<td>92,327</td>
</tr>
<tr>
<td>The average length of a surname in characters</td>
<td>( \ell )</td>
<td>6,93</td>
</tr>
</tbody>
</table>

Practical problems

In applying these success norms a number of practical problems have been experienced resulting in the following adjustments:

a. A surname, \( u \in V_i \), may have such a small similarity to the other surnames in \( V_i \), that the radius \( r_i \) is very large. If it is close to one then \( U_i \) includes most of the surnames in \( V \).

Therefore, it has been decided that in the event of surname \( u \in V_i \) having a smaller similarity to each of the other surnames \( v \in V_i \) than a predetermined cut-off value, \( l_1 \), then such a surname is considered an outlier. The “covering” circle of \( V_i := V_i - \{u\} \), is then determined instead, i.e. \( u \) is excluded from \( V_i \).

b. For some ideal class, \( V_k \), it may happen that each surname in \( V_k \) has such a small similarity to all the other surnames in \( V_k \), that the radius is close to one. Thus, all the surnames of \( V_k \) can be considered as outliers.

This case has been handled as follows: whenever the radius \( r_k > (1 - l_2) \), a predetermined cut-off value, then all the surnames in \( V_k \) excluding the one with the highest frequency of occurrence are taken as outliers. The result is a circle with a very small radius, \( \epsilon \), say, containing a group consisting of one surname. (To simplify further discussion such a circle will be referred to as a “circle with a zero radius”)

c. For some ideal classes more than one surname may be a candidate for the centre of the circle (e.g. a class containing only two elements).

Let \( K_m \) be the set of candidates for the centre of \( V_m \).

Let \( X = \max \left( \min \{d(v, w)\} \right) \).

The element \( v_y \in K_m \) such that \( \min \{d(v_y, w)\} = X \), is chosen as the centre.

After the sets \( U_i \)'s have been determined, each surname \( v \in V \), is an element of one and only one of the following sets:

\( W_U : \) The set of outliers.
\( W_D : \) The set of surnames appearing in more than one circle, \( U_i \), which do not belong to \( W_U \).
\( W_G : \) The set of surnames appearing in only one circle which do not belong to \( W_U \).

\( W_U, W_D \) and \( W_G \) form a partition of \( V \).

In order to cater for the outliers the success norms of an algorithm as given in equations 1 and 2, are adapted as follows:

\[
S_3(U) = \frac{100}{n} \left( |V| - |W_D| - |W_U| \right) = \frac{100}{n} |W_G| \quad (3)
\]
\[ S_n(U) = \frac{100}{N} \left( \sum_{v \in W_0} \rho(v) \right) \]  

(4)

Statistics determined

The adjustments made may impact the functionality of the success norms. To guard against this and to gain a better understanding of the algorithms and the effects of the parameters \( l_1 \) and \( l_2 \), the following statistics have been determined.

Let \( s_v \) be the number of circles containing the surname, \( v \). Note that the radius of a circle is zero, not only under the circumstances previously explained, but also when an ideal class contains only one element. Let:

\[ G = \{U_i : r_i > 0 \text{ and } U_i \in U \} \]

(5)

\[ I_G = \{i : U_i \in G\} \]

(6)

1. The percentage of surnames, \( pv_u \), considered as outliers:

\[ pv_u = \left( \frac{|W_U|}{|V|} \right) \times 100 \]

(7)

2. The average number of foreign circles \(^1\) in which an outlier occurs:

\[ g_{su} = \frac{\sum_{v \in W_U} s_v}{|W_U|} \]

Note that if a surname, \( v_k \in V_i \), has been taken as an outlier then \( v_k \notin U_i \).

3. The percentage of surnames, excluding outliers, appearing in more than one circle:

\[ pv_d = \left( \frac{|W_D|}{|V-W_U|} \right) \times 100 \]

(8)

4. The average number of circles in which each of the elements of \( W_D \) appears:

\[ gs_d = \frac{\sum_{v \in W_D} s_v}{|W_D|} \]

(9)

Note that \( gs_d > 1 \).

5. To get an indication of how many of these surnames appear in 2, 3, 4, . . . circles, a frequency distribution has been determined. The frequencies are expressed as percentages and denoted by \( pv_{di} \) with \( i = 2, 3, 4, . . . \).

6. Percentage of ideal classes left out \( pg_u \). See the practical problem (b) mentioned above.

7. The average number of surnames used to determine a circle:

\[ g_{v_s} = \frac{\sum_{i \notin I_G} |V_i-W_U|}{|G|} \]

(10)

8. The average number of foreign surnames in each of the elements of \( G \), \( gn_s \) (excluding outliers).

\[ T_G = \sum_{U_i \in G} |U_i \cap (V-V_i)-W_U| \]

(11)

\[ gn_s = \frac{T_G}{|G|} \]

(12)

9. The average number of outliers in each element of \( G \):

\[ T_2 = \sum_{U_i \in G} |U_i \cap W_U| \]

(13)

\[ gu_s = \frac{T_2}{|G|} \]

(14)

10. The average number of elements of foreign groups in each element of \( G \), \( gg_s \).

\[ X_i = \{V_j : i \neq j, U_i \cap V_j \neq \emptyset \} \]

(15)

\[ g_{gs} = \frac{T_3}{|G|} \]

(16)

3 Algorithms and evaluation

For each algorithm the results have been determined for three different combinations of values for \((l_1; l_2)\), viz (0.1; 0.1), (0.2; 0.1) and (0.2; 0.2). A detailed discussion of the evaluation is given in [4]. Here the main results are summarised.

Position-independent algorithms

Findler and Van Leeuwen [5] investigated a class of similarity norms and proposed the following:

\[ G_1(u, v) = \frac{T_1(u, v)}{N_1(u, v)} \] where

\[ T_1(u, v) = \sum_{\alpha \in (u^+v^+)} \min\{p(u : \alpha), p(v : \alpha)\} \cdot |\alpha| \]

\[ N_1(u, v) = \sum_{\alpha \in (u^+v^+)} \max\{p(u : \alpha), p(v : \alpha)\} \cdot |\alpha| \]

and

\[ G_2(u, v) = \frac{T_2(u, v)}{N_2(u, v)} \] where

\[ T_2(u, v) = T_1(u, v) \]

\[ N_2(u, v) = \left[ \sum_{\alpha \in (u^+v^+)} p(u : \alpha) \cdot |\alpha| \right]^{1/2} \cdot \left[ \sum_{\alpha \in (u^+v^+)} p(v : \alpha) \cdot |\alpha| \right]^{1/2} \]

Where

---

\(^1\) A surname appears in a foreign circle, if it lies in the circle of another ideal class.
The main drawback of Method 2 is the large values for $g_s$, and $g_g$.

In both cases substrings of all lengths are used which require an unacceptably large number of processing steps to determine the similarity. The main objection against these methods is that common substrings are counted independently of their relative position.

### Table 2. Results for Methods 1 and 2

<table>
<thead>
<tr>
<th>Method</th>
<th>$1a$</th>
<th>$1b$</th>
<th>$1c$</th>
<th>$2a$</th>
<th>$2b$</th>
<th>$2c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_1$</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>$l_2$</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>$p_vu$</td>
<td>16.4</td>
<td>24.0</td>
<td>28.7</td>
<td>5.6</td>
<td>9.2</td>
<td>18.5</td>
</tr>
<tr>
<td>$g_s$</td>
<td>2.00</td>
<td>0.75</td>
<td>0.34</td>
<td>11.1</td>
<td>5.76</td>
<td>1.74</td>
</tr>
<tr>
<td>$p_d$</td>
<td>6.60</td>
<td>13.9</td>
<td>15.7</td>
<td>1.78</td>
<td>4.86</td>
<td>7.84</td>
</tr>
<tr>
<td>$p_d$</td>
<td>55.4</td>
<td>23.5</td>
<td>8.5</td>
<td>95.5</td>
<td>85.8</td>
<td>52.2</td>
</tr>
<tr>
<td>$p_d$</td>
<td>32.1</td>
<td>2.30</td>
<td>2.08</td>
<td>8.87</td>
<td>5.22</td>
<td>2.90</td>
</tr>
<tr>
<td>$p_d$</td>
<td>45.7</td>
<td>76.0</td>
<td>92.6</td>
<td>7.40</td>
<td>20.0</td>
<td>53.2</td>
</tr>
<tr>
<td>$p_d$</td>
<td>24.0</td>
<td>19.4</td>
<td>7.14</td>
<td>8.99</td>
<td>18.3</td>
<td>24.1</td>
</tr>
<tr>
<td>$S_3$</td>
<td>37.3</td>
<td>58.2</td>
<td>65.2</td>
<td>4.28</td>
<td>12.9</td>
<td>39.0</td>
</tr>
<tr>
<td>$S_4$</td>
<td>33.4</td>
<td>61.4</td>
<td>80.7</td>
<td>2.34</td>
<td>9.45</td>
<td>35.2</td>
</tr>
<tr>
<td>$p_d$</td>
<td>73.3</td>
<td>80.6</td>
<td>82.4</td>
<td>68.5</td>
<td>71.6</td>
<td>74.5</td>
</tr>
<tr>
<td>$g_r$</td>
<td>2.73</td>
<td>2.71</td>
<td>2.41</td>
<td>3.07</td>
<td>3.07</td>
<td>2.68</td>
</tr>
<tr>
<td>$g_n$</td>
<td>8.16</td>
<td>2.83</td>
<td>0.90</td>
<td>41.9</td>
<td>22.4</td>
<td>7.03</td>
</tr>
<tr>
<td>$g_n$</td>
<td>1.29</td>
<td>0.62</td>
<td>0.22</td>
<td>2.23</td>
<td>1.92</td>
<td>1.33</td>
</tr>
<tr>
<td>$g_g$</td>
<td>6.72</td>
<td>2.30</td>
<td>0.81</td>
<td>31.7</td>
<td>17.7</td>
<td>5.87</td>
</tr>
</tbody>
</table>

The results for $G_1$ and $G_2$ for the three combinations of values for $l_1$ and $l_2$ are given in Table 2 under the columns Method 1a–c and 2a–c respectively.

Although Method 1 seems to be good, note that

a) too many surnames have been taken as outliers (cf. $p_vu$);

b) for too many ideal classes a too small similarity between the elements of the class has been determined (cf. $p_d$).

The main drawback of Method 2 is the large values for $g_s$, and $g_g$.

An evaluation has been made using five different values of the parameters $k$ and $s$ in the range 1 to 3, and the combinations of $l_1$ and $l_2$ mentioned earlier. This report is restricted to the best two combinations which are called Method 3.1 and 3.2.

The statistical results are given in Table 3. Method 3.1 yields the smallest value for $p_vu$ and $p_d$.

An evaluation was made using five different values of the parameters $k$ and $s$ in the range 1 to 3, and the combinations of $l_1$ and $l_2$ mentioned earlier. This report is restricted to the best two combinations which are called Method 3.1 and 3.2.

### Position-dependent algorithms

Whenever a common substring appears in two strings, but the position where it appears in the two strings differs too much, then it is not used in the calculation of $T(u, v)$.

Ito [7] adapted Findler’s similarity measure, $G_1$, as follows:

1. Let $A = \phi$.
2. Let $B = v_{i,k}^+$.
3. For each substring $\alpha_u(r_u) \in u_{i,k}^+$, in increasing order of its position:
   a) Let $C = \{ \alpha_u(r_u) : \alpha_u(r_u) \in B \}$.
   b) Let $\alpha_v = \alpha_u$ and $r_v = [r_u - s, r_u + s]$.  

The statistical results are given in Table 3. Method 3.1 yields the smallest value for $p_vu$ and $p_d$.

After step 3 of Ito’s algorithm it follows:

$$G_3(u, v) = G_3(u, v)$$

For the surnames $v = \text{clerk}$ and $v = \text{klerk}$, $G_3(u, v)$ will be determined as follows for $s = 1$ and $k = 2$:

$$u_{i,k}^+ = \{ c (1), l (2), c (3), r (4), q (5), cl (1), le (2), er (3), rq (4) \}$$

$$v_{i,k}^+ = \{ k (1), l (2), c (3), r (4), k (5), kl (1), le (2), er (3), rk (4) \}$$

For the surnames $u = \text{clerk}$ and $v = \text{klerk}$, $G_3(u, v)$ will be determined as follows for $s = 1$ and $k = 2$:

The statistical results are given in Table 3. Method 3.1 yields the smallest value for $p_vu$ and $p_d$.

Although Method 3.2 also yields small values for $p_d$ and $p_d2$, the sum $p_d2 + p_d3$, i.e. the percentage of elements of $W_D$ appearing in two or less foreign circles is relatively large, viz 35.9; 59.6 and 82.3. Although more
surnames are classified as outliers (cf. $pv_u$), the values for $pv_u$ are not that much larger. The larger values for $S_3$ and smaller values for $gna$ and $gga$ indicate that Method 3.2 resulted in a better grouping of the surnames.

Table 3. Results for Methods 3.1 and 3.2

<table>
<thead>
<tr>
<th></th>
<th>Method 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1a</td>
</tr>
<tr>
<td>$l_1$</td>
<td>0.1</td>
</tr>
<tr>
<td>$l_2$</td>
<td>0.1</td>
</tr>
<tr>
<td>$s$</td>
<td>1</td>
</tr>
<tr>
<td>$k$</td>
<td>2</td>
</tr>
<tr>
<td>$pv_u$</td>
<td>3.08</td>
</tr>
<tr>
<td>$gs_u$</td>
<td>9.83</td>
</tr>
<tr>
<td>$gga_u$</td>
<td>0.93</td>
</tr>
<tr>
<td>$pv_d$</td>
<td>98.2</td>
</tr>
<tr>
<td>$gs_d$</td>
<td>7.83</td>
</tr>
<tr>
<td>$gga_d$</td>
<td>4.21</td>
</tr>
<tr>
<td>$pv_d3$</td>
<td>7.64</td>
</tr>
<tr>
<td>$S_1$</td>
<td>1.74</td>
</tr>
<tr>
<td>$S_2$</td>
<td>1.09</td>
</tr>
<tr>
<td>$p_{sv0}$</td>
<td>67.6</td>
</tr>
<tr>
<td>$gv_s$</td>
<td>3.15</td>
</tr>
<tr>
<td>$gn_s$</td>
<td>36.3</td>
</tr>
<tr>
<td>$gna_s$</td>
<td>1.05</td>
</tr>
<tr>
<td>$gga_s$</td>
<td>26.8</td>
</tr>
</tbody>
</table>

Other algorithms

Sidorov [8] investigated algorithms to determine the similarity between words in order to correct typing errors. His requirements were small memory usage and speed. Common substrings were only used if they appeared in the same order. This method proved to be unsuitable for solving the primary problem and will not be discussed. For details see [4].

Some algorithms are based on $n$-grams, an $n$-gram being a substring of length $n$. The most common values for $n$ are two and three, in which case the $n$-grams are called di- and tri-grams respectively. Freund, Angell and Willet [6] and [1] investigated similarity measures, the so-called “Dice” and “Overlap” coefficients. The author’s evaluation reported in [4] showed that tri-grams yielded better results for the test dataset. All the statistics however, indicate that these methods are not suitable for solving the primary problem.

In conclusion, then, Method 3.2 developed by Ito – although not ideal – appears to be the best of the various methods considered in this study.

References


Notes for Contributors

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems, as well as shorter technical research papers. However, non-refereed review and exploratory articles of interest to the journal’s readers will be considered for publication under sections marked as Communications or Viewpoints. While English is the preferred language of the journal, papers in Afrikaans will also be accepted. Typed manuscripts for review should be submitted in triplicate to the editor.

Form of Manuscript

Manuscripts for review should be prepared according to the following guidelines.

- Use wide margins and 1½ or double spacing.
- The first page should include:
  - title (as brief as possible);
  - author’s initials and surname;
  - author’s affiliation and address;
  - an abstract of less than 200 words;
  - an appropriate keyword list;
  - a list of relevant Computing Review Categories.
- Tables and figures should be numbered and titled. Figures should be submitted as original line drawings/printouts, and not photocopies.
- References should be listed at the end of the text in alphabetic order of the (first) author’s surname, and should be cited in the text in square brackets [1–3]. References should take the form shown at the end of these notes.

Manuscripts accepted for publication should comply with the above guidelines (except for the spacing requirements), and may be provided in one of the following formats (listed in order of preference):

1. As (a) \\texttt{LaTeX} file(s), either on a diskette, or via e-mail/ftp – a \\texttt{LaTeX} style file is available from the production editor;
2. As an ASCII file accompanied by a hard-copy showing formatting intentions:
   - Tables and figures should be on separate sheets of paper, clearly numbered on the back and ready for cutting and pasting. Figure titles should appear in the text where the figures are to be placed.
   - Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin, if they are not clear in the text.

Further instructions on how to reduce page charges can be obtained from the production editor.

3. In camera-ready format – a detailed page specification is available from the production editor;
4. In a typed form, suitable for scanning.

Charges

Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect scanning, typesetting, reproduction and other costs. Currently, the minimum rate is R30-00 per final page for \\texttt{LaTeX} or camera-ready contributions and the maximum is R120-00 per page for contributions in typed format (charges include VAT).

These charges may be waived upon request of the author and at the discretion of the editor.

Proofs

Proofs of accepted papers in categories 2 and 4 above will be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Note that, in the case of camera-ready submissions, it is the author’s responsibility to ensure that such submissions are error-free. However, the editor may recommend minor typesetting changes to be made before publication.

Letters and Communications

Letters to the editor are welcomed. They should be signed, and should be limited to less than about 500 words.

Announcements and communications of interest to the readership will be considered for publication in a separate section of the journal. Communications may also reflect minor research contributions. However, such communications will not be refereed and will not be deemed as fully-fledged publications for state subsidy purposes.

Book reviews

Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

Advertisement

Placement of advertisements at R1000-00 per full page per issue and R500-00 per half page per issue will be considered. These charges exclude specialized production costs which will be borne by the advertiser. Enquiries should be directed to the editor.

References

Contents

GUEST CONTRIBUTIONS
Ideologies of Information Systems and Technology
  LD Introna ......................................................... 1

What is Information Systems?
  TD Crossman .................................................... 7

RESEARCH ARTICLES
Intelligent Production Scheduling: A Survey of Current Techniques and An Application in The Footwear Industry
  V Ram .......................................................... 11

Effect of System and Team Size on 4GL Software Development Productivity
  GR Finnie and GE Wittig ...................................... 18

EDI in South Africa: An Assessment of the Costs and Benefits
  G Harrington .................................................... 26

Metadata and Security Management in a Persistent Store
  S Berman ......................................................... 39

Markovian Analysis of DQDB MAC Protocol
  F Bause, P Kritzinger and M Sczitnack ......... 47

TECHNICAL NOTE
An evaluation of substring algorithms that determine similarity between surnames
  GdeV de Kock and C du Plessis .......................... 58

COMMUNICATIONS AND REPORTS
Ensuring Successful IT Utilisation in Developing Countries
  BR Gardner ...................................................... 63

Information Technology Training in Organisations: A Replication
  R Roets .......................................................... 68

The Object-Oriented Paradigm: Uncertainties and Insecurities
  SR Schach ....................................................... 77

A Survey of Information Authentication Techniques
  WB Smuts ........................................................ 84

Parallel Execution Strategies for Conventional Logic Programs: A Review
  PEN Lutu .......................................................... 91

The FRD Special Programme on Collaborative Software Research and Development: Draft Call for Proposals 99

Book review ...................................................... 102