THE SHORTEST COMMON SUPERSEQUENCE PROBLEM OVER BINARY ALPHABET IS NP-COMPLETE

Kari-Jouko RÄIHÄ* and Esko UKKONEN

Department of Computer Science, University of Helsinki, SF-00250 Helsinki 25, Finland

Communicated by A. Salomaa

Received May 1980

Abstract. We consider the complexity of the Shortest Common Supersequence (SCS) problem, i.e. the problem of finding for finite strings $S_1, S_2, \ldots, S_n$ a shortest string $S$ such that every $S_i$ can be obtained by deleting zero or more elements from $S$. The SCS problem is shown to be NP-complete for strings over an alphabet of size $2^2$.

Given a string $S$ over an alphabet $\Sigma$, we define a supersequence $S'$ of $S$ to be any string $S' = w_0x_1w_1x_2w_2 \cdots x_kw_k$ over $\Sigma$ such that $S = x_1x_2 \cdots x_k$ and each $w_i \in \Sigma^*$. A common supersequence of a set of strings $R = \{S_1, S_2, \ldots, S_n\}$ is a string $S$ over $\Sigma$ such that $S$ is a supersequence of each $S_i$. The Shortest Common Supersequence (SCS) problem can now be stated as follows: Given an alphabet $\Sigma$, a finite set $R$ of strings from $\Sigma^*$, and a positive integer $k$, is there a common supersequence of $R$ of length $\leq k$? If $S'$ is a supersequence of $S$, then $S$ is a subsequence of $S'$. The Longest Common Subsequence (LCS) problem can be defined in an obvious way.

The complexity of the SCS and LCS problems for an arbitrary set $R$ has been studied by Maier [5]. He is mainly interested in the LCS problem which he shows to be NP-complete when the size of the alphabet $\Sigma$ is $\geq 2$. The LCS problem is, of course, trivially solvable in polynomial time when $\Sigma$ is of size one. For a fixed $k$ or for a fixed size of $R$, the problem is also known to be solvable in polynomial time, see e.g. [1, 7]. Furthermore, it was shown in [5] that the SCS problem is NP-complete when the size of $\Sigma$ is $\geq 5$.

In this paper we improve the result of [5] on the SCS problem by showing that the problem is NP-complete already for alphabet size $\geq 2$, i.e. for the binary alphabet. The SCS problem is therefore in this respect similar to the LCS problem. Again, the SCS problem is trivially solvable in polynomial time when the size of the set $R$ is $2$ (by first computing the longest common subsequence), or if all $S_i \in R$ are of length $\leq 2$ [3], or if the size of $\Sigma$ is $1$.

* The work of this author was supported by the Academy of Finland.

0304-3975/81/0000-0000/$02.50 © 1981 North-Holland
Our proof technique and notations are developed from those of [5]. We use the convenient concept of threading schemes, introduced in Section 2. The result is proved in Section 3.

Our result has found applications in the field of evaluation of attribute grammars. In fact, the NP-completeness of the SCS problem over binary alphabet leads to the result that the problem of finding an optimal multi-pass evaluator for an attribute grammar is NP-complete, too [6]. The SCS problem may also have applications to data compression techniques.

2. Threading schemes

Following [5] we analyze the SCS problem in terms of so-called threading schemes. We think of a string in \( R \) as a row of beads with labels from \( \Sigma \). The process of constructing a common supersequence is then equivalent to threading the beads in a certain manner. As an example we consider a set \( R \) having three strings over \( \Sigma = \{0, 1\} \): \( S_1 = 10100, S_2 = 001101, S_3 = 10110 \). The strings are represented as rows of beads:

\[
\begin{array}{cccccc}
1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 \\
\end{array}
\]

To construct a common supersequence all the beads are threaded so that

(i) each thread contains at most one bead from each row,

(ii) all beads on a thread must have the same label from \( \Sigma \), called the type of the thread, and

(iii) threads may not cross.

For the SCS problem, we want to find if \( k \) threads are sufficient to thread all the beads. In the example we have, among others, the following threading with 8 threads; in fact, in this case \( k \) must always be \( \geq 8 \):

\[
\begin{array}{cccccc}
0_1 & 0_2 & 0_3 & 0_4 & 0_5 & 0_6 & 0_7 & 0_8 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 \\
\end{array}
\]

It is convenient to refer to a thread by its type or by the terms of the strings it threads. Thus in the example \( \theta_2 \) is a 1-thread threading strings \( S_1 \) and \( S_3 \).
A threading scheme for the SCS problem is a list (from left to right) of threads \( \theta_1, \theta_2, \ldots, \theta_p \) which fulfill the rules and thread all the beads. Given a threading scheme \( \Theta = (\theta_1, \theta_2, \ldots, \theta_p) \) for a set of strings \( R \), we can obtain a common supersequence of \( R \) by concatenating the types of \( \theta_1, \theta_2, \ldots, \theta_p \). In our example, 01011001 is a common supersequence. Clearly, for a given threading scheme the implied supersequence is unique, but the same supersequence may have several threading schemes.

3. The result

The purpose of this section is to prove the following result.

**Theorem.** The SCS problem is NP-complete for an alphabet \( \Sigma \) of size \( \geq 2 \).

**Proof.** The SCS problem is clearly in class NP. To prove the completeness we must therefore give a polynomial time transformation from some known NP-complete problem to the SCS problem over binary alphabet. The transformation we give will be, as in [5], from the node cover problem [2, 4]. Given a graph \( G = (N, E) \) and an integer \( k \), the node cover problem is to determine if there is a subset \( N' \subseteq N \) such that \( N' \) has at most \( k \) elements and, for each edge \( (x, y) \in E \), at least one of \( x \) and \( y \) belongs to \( N' \).

Let \( G = (N, E) \) and \( k \) constitute an instance of the node cover problem where \( N = \{v_1, v_2, \ldots, v_t\} \), \( E = \{(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t)\} \). We construct a set \( R \) of \( t + 1 \) strings over the binary alphabet \( \Sigma = \{0, 1\} \). Basically, our construction is a simplified version of the transformation used in [5] to prove the result of the theorem for alphabet size \( \geq 5 \).

The first string in \( R \) is the template \( T \). In addition, \( R \) contains a string \( S_i \) for each edge \( e_i = (x_i, y_i) \) in \( E \). In these strings, the nodes and edges of \( G \) are encoded using the alphabet \( \{0, 1\} \). We first describe the encoding, shown in Fig. 1. The node codeplate \( \bar{N} \) is defined as \( t + 1 \) blocks of \( 7c \) ones, where \( c = \max(r, t) \). Any \( v_i \) in \( N \) we encode with node code \( \bar{N}[i] \) which is obtained by inserting a zero between the \( i \)th and \((i+1)\)st blocks of \( \bar{N} \). The multiple node code \( \bar{N}[i_1, i_2, \ldots, i_t] \) has a zero in the \( i_1 \)st, \( i_2 \)nd, \ldots, and \( i_t \)th spots. The special case of \( \bar{N}[1, 2, \ldots, r] \) will be denoted by \( \bar{N}_s \) and referred to as the node sink, since it is a supersequence of all the node codes. The edge codeplate \( \bar{E} \), the edge code \( \bar{E}[j] \), and the multiple edge code \( \bar{E}[j_1, j_2, \ldots, j_t] \) are defined similarly with blocks of \( 7c \) zeros and pairs of ones. (Only the code \( \bar{E}[j] \) is shown in Fig. 1.) The code \( \bar{E}[1, 2, \ldots, r] \) is called the edge sink and denoted by \( \bar{E}_s \).

Now we can define the \( r + 1 \) strings of \( R \), shown in Fig. 2. The template \( T \) consists of the following codes in the given order: \( \bar{E}; \bar{N}_s; \bar{E}_s; \bar{N}; \bar{E}; \bar{N}_s; \bar{E} \). We denote the length of \( T \) by \( q = 7c(4r + 3t + 7) + 4r + 2t \). For each \( e_i = (x_i, y_i) \) we define \( S_i \) as: \( \bar{E}[i]; \bar{N}[j]; \bar{N}[m]; \bar{E}[i] \), where \( j \) and \( m \) are such that \( x_i = v_j, y_i = v_m \). To distinguish
Node codeplate \( N \): 
\[ \text{length} = 7c(t+1) \]

\[
\begin{array}{cccccccccc}
1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & 1 & 1 \\
7c & 7c & 7c & 7c & \cdots & 7c & 7c & 7c & 7c & 7c \\
\end{array}
\]

Node code \( \tilde{N}(i) \): 
\[ \text{length} = 7c(t+1)-1 \]

\[
\begin{array}{cccccccccc}
1 & 1 & 1 & \cdots & 1 & 0 & 1 & \cdots & 1 & 1 \\
7c & 7c & 7c & \cdots & 7c & 1 & 7c & \cdots & 7c & 7c \\
\end{array}
\]

Multiple node code \( \tilde{N}(i_1, i_2, \ldots, i_s) \): 
\[ \text{length} = 7c(t+1)-s \]

\[
\begin{array}{cccccccccc}
1 & 1 & \cdots & 1 & 101 & 1 & \cdots & 1 & 101 & 1 \\
7c & 7c & \cdots & 7c & 1 & 7c & \cdots & 7c & 1 & 7c \\
\end{array}
\]

Edge code \( E[j] \): 
\[ \text{length} = 7c(t+1)-2 \]

\[
\begin{array}{cccccccccc}
1 & 0 & 0 & 1 & \cdots & 1 & 0 & 0 & 1 & 0 \\
7c & 7c & \cdots & 7c & 2 & 7c & \cdots & 7c & 7c & 7c \\
\end{array}
\]

Fig. 1.

between the left and right occurrences of the same code in a string we use superscripts \( L \) and \( R \).

Template \( T \): 
\[ \text{length} q = 7c(4r+3t+7)+4r+2t \]

\[
\begin{array}{ccccccccccc}
E^L & N^L & E^L & S & N & E^R & N^R & E^R \\
\end{array}
\]

String \( S_1 \): 
\[ \text{length} = 7c(2r+2t+4)+6 \]

\[
\begin{array}{cccccccccc}
\end{array}
\]

Fig 2.

By proving the following two claims we show that the above transformation from the node cover problem to the SCS problem has the desired properties.

**Claim 1.** If \( G \) has a node cover of size \( k \), then \( R \) has a common supersequence of length \( q+(2r+k) \).
Proof. Let \( N' = \{ n_1, n_2, \ldots, n_k \} \) be a node cover of size \( k \). Let \( W = \{ e_i | e_i = (x_i, y_i) \) and \( x_i \in N' \} \) and \( U = E - W \). Now, if \( e_i \in U \), then \( y_i \in N' \). Let \( T' \) be the string \( T' = \bar{E}[U]; \bar{N}_i; E_i; \bar{N}[N']; \bar{E}_i; \bar{N}_i; E[W] \). Since \( U \cup W = E \), the length of \( T' = q + (2r + k) \). The string \( T' \) is a supersequence of \( T \), since each block of \( T' \) is the same as the corresponding block in \( T \) with possibly some zeros and ones added. Moreover, \( T' \) is a supersequence of each \( S_i \). The matching goes as follows:

Case a: \( S_i \) corresponds to an edge in \( W \) (see Fig. 3).

Case b: \( S_i \) corresponds to an edge in \( U \) (see Fig. 4).

Thus \( T' \) is a common supersequence of \( R \).

Claim 2. If \( R \) has a common supersequence of length \( q + (2r + k) \), then \( G \) has a node cover of size \( \leq k \).

Proof. The set \( N \) is trivially a node cover for \( G \). Therefore, if \( k \geq t \), the claim is clearly true.

In the rest of the proof we assume that \( k < t \). Let \( T_0 \) be a common supersequence of \( R \) of length \( q + (2r + k) \), and let \( \Theta_0 \) be a threading scheme for \( T_0 \). The proof is now continued with a sequence of lemmas in which we construct, starting from \( T_0 \) and \( \Theta_0 \), a sequence \( T_1, T_2, T_3, T_4 \) of supersequences of \( R \) and corresponding threading schemes \( \Theta_1, \Theta_2, \Theta_3, \Theta_4 \) such that the length of the \( T_i \)'s is decreasing. From the final result, \( T_4 \) and \( \Theta_4 \), we may decide that a node cover of size \( \leq k \) for \( G \) exists. Each new \( T_i \) and \( \Theta_i \) constructed in this process is more and more similar to the supersequence and the threading scheme used in the proof of Claim 1.

In the sequel we use the following convenient terminology. A thread which threads some term in the template \( T \) is called a \emph{T-thread} and the other threads are called \emph{extra threads}. Scheme \( \Theta_0 \) has \( q + (2r + k) \) threads including \( q \) T-threads and
$2r + k < 3c$ extra threads. The main argument used in the proofs of the lemmas will be the number of extra threads which is not allowed to be $\geq 3c$. An extra thread is called private if it threads a term in only one string of $R$. Otherwise an extra thread is called shared.

We also need terminology to refer to the relative ordering of terms of the strings in $R$ implied by a threading scheme $\Theta = (\theta_1, \theta_2, \ldots, \theta_p)$. Let a term $a$ of a string be threaded by $\theta_i$ and a term $b$ of (possibly another) string be threaded by $\theta_j$. If $i < j$, then we say that $a$ is to the left of $b$ (and $b$ is to the right of $a$) in scheme $\Theta$. More generally, if $A$ is a block of terms of one string and $B$ a block of terms of another string in $R$, then $A$ is said to be to the left of $B$ (and $B$ to the right of $A$) if for each term $a$ in $A$ and $b$ in $B$, $a$ is to the left of $b$.

The length of a string $S$ is denoted by $|S|$.

**Lemma 1.** For each string $S_i$, block $E[i]^L$ is to the left of $E[i]^L$ or block $E[i]^R$ is to the right of $E[i]^R$ in $\Theta_0$.

**Proof.** If the lemma is not true, then the block $\widetilde{N}[j]$; $\widetilde{N}[m]$ of $S_i$ must be to the right of $\widetilde{N}[s]^L$ and to the left of $\widetilde{N}[s]^R$ in $\Theta_0$. So we have the situation given in Fig. 5. Since $\widetilde{N}[j]$ $\widetilde{N}[m]$ contains $14c(t + 1)$ ones and there are only $7c(t + 1) + 4r$ ones between $\widetilde{N}[j]$ and $\widetilde{N}[m]$ in $T$, this means that $7c(t + 1) - 4r > 3c$ ones in $\widetilde{N}[j]$; $\widetilde{N}[m]$ must be on extra threads, a contradiction.

**Lemma 2.** There is a common supersequence $T_1$ of $R$ and a threading scheme $\Theta_1$ for $T_1$ such that $|T_1| = |T_0|$ and for each $S_i$, block $E[i]^L$ is to the left of $\widetilde{N}[s]^L$ and the pair of ones in $E[i]^L$ is threaded by extra threads in $\Theta_1$, or block $E[i]^R$ is to the right of $\widetilde{N}[s]^R$ and the pair of ones in $E[i]^R$ is threaded by extra threads in $\Theta_1$. Moreover, if $E[i]^L$ is not to the left of $\widetilde{N}[s]^L$, then it is not to the left of $E[i]^L$, and if $E[i]^R$ is not to the right of $\widetilde{N}[s]^R$, then it is not to the right of $E[i]^R$ in $\Theta_1$.

**Proof.** Suppose that $E[i]^L$ is to the left of $E[i]^L$ in $\Theta_0$. To prove the lemma we show that it is possible to transform $T_0$ and $\Theta_0$ so that in the new threading $E[i]^L$ is to the left of $\widetilde{N}[s]^L$. This transformation (and a symmetric transformation for those $E[i]^R$ that are to the right of $E[s]^R$ in $\Theta_0$) can be done successively for each $E[i]$ to the left of $\widetilde{N}[s]$ and each $E[i]^R$ to the right of $E[s]^R$ in $\Theta_0$. The resulting supersequence $T_1$ and threading scheme $\Theta_1$ are as required in the lemma because by Lemma 1, either $E[i]^L$ or $E[i]^R$ satisfies the condition of the transformation for every $S_i$. Note that
if $E[i]^L$ or $E[i]^R$ does not satisfy this condition, then it necessarily satisfies the last assertion of the lemma.

Before modifying $T$ and $\Theta$ for $E[i]^L$ we show that the two ones in $E[i]^L$ are to the left of $N_s^L$. In fact, if this is not true, then all zeros of $E[i]^L$ to the right of the pair of ones must be threaded in $\Theta$ to the right of $E^L$ but to the left of $E_{s}^L$. This means that $\Theta$ has at least $7c - t > 3c$ extra threads because $E[i]^L$ has at least $7c$ zeros to the right of the two ones but $N_s^L$ contains only $t$ zeros, a contradiction. Hence the two ones of $E[i]^L$ must be to the left of $N_s^L$ in $\Theta$. This clearly implies that these ones are threaded by extra threads in $\Theta$.

Therefore, if $E[i]^L$ is to the left of $N_s^L$ already in $\Theta$, no modification of $\Theta$ is needed for $E[i]^L$. Otherwise $E[i]^L$ has zeros that are not to the left of $N_s^L$ in $\Theta$. We will move them on suitable threads which already thread $E^L$. In more detail, the $7c \cdot i$ zeros to the left of the pair of ones in $E[i]^L$ are first threaded with the $7c \cdot i$ leftmost zeros of $E^L$. This introduces no extra threads. Then two extra threads are added to thread the two ones of $E[i]^L$ and the original threads of the two ones are removed. If either of the original threads for the ones was not private, some 1 in a string $S_h$, $h \neq i$, may now become unthreaded. Let $\alpha_h$ be a prefix of $S_h$ such that the last term of $\alpha_h$ is the rightmost unthreaded 1 of $S_h$. Then it is straightforward to see from the structure of the strings in $R$ and from the fact that $E[i]^L$ is to the left of $E_s^L$ in $\Theta$, that string $\alpha_h$ must always be of the form $0^{7c \cdot h}1$ or $0^{7c \cdot h}1$, where $h < i$. Otherwise the situation would be as in Fig. 6, where $h > i$ and too many extra threads are required for the zeros to the left of $\theta$ in $S_h$ and between $\theta$ and $\theta'$ in $S_i$.

Thus $\alpha_h$ is a subsequence of the prefix $\beta = 0^{7c \cdot i}11$ of $S_i$ which we have threaded so far. Hence $\alpha_h$ can be threaded by the threads of $\beta$. After performing this process for each $S_h$ there are no unthreaded terms in $R$. The $7c((r + 1) - i)$ zeros following the pair of ones in $E[i]^L$ are finally threaded with the $7c((r + 1) - i)$ rightmost zeros of $E^L$. In the resulting threading $E[i]^L$ is to the left of $N_s^L$ and the pair of ones is threaded by extra threads. Moreover, the corresponding common supersequence is of length $\leq |T_0|$. Repeating this process for each $S_i$ we arrive at $T_1$ and $\Theta_1$ which are as required.
Lemma 3. There is a common supersequence $T_2$ of $R$ and a threading scheme $\Theta_2$ for $T_2$ such that $|T_2| \leq |T_1|$ and $\Theta_2$ is as $\Theta_1$ except if for some string $S_i$ block $E[i]^L$ is to the left of $N[i]$ and $E[i]^R$ to the right of $N[i]$ in $\Theta_1$, then no term of $N[j]$ of $N[m]$ of $S_i$ is threaded by extra threads in $\Theta_2$.

Proof. For every such string $S_i$ we may change $\Theta_1$ so that $N[i]$ is threaded with $N[i]$ and $N[m]$ with $N[m]$. Since $N[i]$ and $N[m]$ are subsequences of $N[i]$, this can be done without introducing new threads but some threads of $\Theta_1$ may become empty. These should be removed. When these changes are successively made for every possible $S_i$, the resulting threading scheme $\Theta_2$ and the corresponding supersequence $T_2$ are as required.

Lemma 4. There is a common supersequence $T_3$ of $R$ and a threading scheme $\Theta_3$ for $T_3$ such that $|T_3| \leq |T_2|$ and $\Theta_3$ is as $\Theta_2$ except if for some $S_i$ block $E[i]^L$ is not to the left of $N[i]$ in $\Theta_2$, then $E[i]^R$ is to the right of $N[i]$ in $\Theta_3$, and symmetrically, if block $E[i]^R$ is not to the right of $N[i]$ in $\Theta_2$, then $E[i]^L$ is to the left of $N[i]$ in $\Theta_3$.

Proof. We first define $T_3$ and $\Theta_3$. Suppose that $E[i]^L$ is not to the left of $N[i]$ in $\Theta_2$. Then by Lemma 2, $E[i]^R$ must be to the right of $N[i]$ in $\Theta_2$. Scheme $\Theta_2$ is now modified such that $E[i]^L$ is threaded with $E[i]^L$, $N[j]$ with $N$ and $N[m]$ with $N[m]$; the threading of $E[i]^R$ remains unchanged. In this process the only extra thread is needed to thread the zero appearing in $N[i]$. This thread $\theta$ crosses $N$ between the $7c \cdot j$th and $(7c \cdot j + 1)$st one. However, if there is already an extra 0-thread $\theta'$ at this place we use $\theta' \forall$ for threading the zero of $N[i]$, and $\theta$ is not introduced. The symmetric case in which $E[i]^R$ is not to the right of $N[i]$ but $E[i]^L$ is to the left of $N[i]$ in $\Theta_2$ is handled analogously: $E[i]^R$ is threaded with $E[i]^R$, $N[m]$ with $N$ and $N[m]$ with $N[i]$. The only term which uses an extra thread is now the zero in $N[m]$.

Applied to every possible $S_i$ the above process yields a scheme $\Theta_3$ which obviously threads every $E[i]^L$ and $E[i]^R$ as required by the lemma. To complete the proof we must still show that $|T_3| \leq |T_2|$.

The only extra threads appearing in $\Theta_3$ but possibly not in $\Theta_2$ are the 0-threads threading the zeros of those $N[i]$ that are threaded with $N$ in $\Theta_3$. In addition, if $S_i$ and $S_j$ share such an extra thread, then both $S_i$ and $S_j$ must contain the block $N[g]$ and the extra thread threads the zeros of these blocks. Hence to prove $|T_3| \leq |T_2|$ it suffices to show that

(i) if $N[g]$ in $S_i$ has an extra 0-thread in $\Theta_3$, then the same $N[g]$ has an extra 0-thread in $\Theta_2$, too, and

(ii) If the extra 0-thread of such a block $N[g]$ is shared in $\Theta_2$ with the zero of another block $N[g']$, then it is shared with the same zero in $\Theta_3$, too.

To prove (i) suppose that $N[g]$ of $S_i$ has an extra thread in $\Theta_3$; the case where $N[m]$ has an extra thread can be considered symmetrically. Then, by the construction of $\Theta_3$, $E[i]^L$ cannot be to the left of $N[i]$ in $\Theta_2$ but $E[i]^R$ is to the right of $N[i]$. Lemma 2 implies that $E[i]^L$ is not to the left of $E[i]^L$ in $\Theta_2$. If the 0-thread $\theta$ threading the
zero of $\bar{N}[j]$ in $\Theta_2$ is not an extra thread, then it must thread a zero of $T$ either to the left or to the right of $\bar{N}$ because there are no zeros in $\bar{N}$. If the zero is to the right of $\bar{N}$ the solution in $\Theta_2$ is as shown in Fig. 7. We note that there are to the right of $\theta$ at most $7c(t + 1) + 2r$ ones in $T$ but at least $7c(t + 1) + 7c$ ones in $S_i$. Hence $\Theta_2$ should have at least $7c - 2r > 3c$ extra 1-threads, a contradiction. Similarly, if $\theta$ is to the left of $\bar{N}$, then again a contradiction follows. Thus we have proved (i).

To prove (ii) let $\bar{N}[j]$ and $\theta$ be as in the proof of (i) and let $\theta$ be shared with the zero of $\bar{N}[j']$ in $\Theta_2$. We prove that $j = j'$ and that $\bar{N}[j']$ has an extra 0-thread in $\Theta$. This proves (ii) because then the zeros of $\bar{N}[j]$ and $\bar{N}[j']$ must be threaded by the same thread in $\Theta_3$. Let $\bar{N}[j']$ be from $S_i$. Since $\bar{N}[j']$ has an extra thread, we know from Lemmas 2 and 3 that $\bar{E}[i']^L$ is not to the left of $\bar{E}_i^L$ or $\bar{E}[i']^R$ is not to the right of $\bar{E}_i^R$. Assume that $\bar{E}[i']^R$ is not to the right of $\bar{E}_i^R$; the other case can be considered similarly. Then $S_i$ must be of the form $\bar{E}[i']^L; N[m]; N[j']; \bar{E}[i']^R$, that is, $\bar{N}[j']$ is the rightmost $N$-block of $S_i$; otherwise the number of extra threads is easily seen to become too large. The situation is as shown in Fig. 8. Here the threads $\theta_1, \theta_2, \theta_3, \theta_4$ are the last threads of $\bar{E}[i']^L$ and $\bar{E}[i']^R$ and the first threads of $\bar{E}[i']^R$ and $\bar{E}[i']^R$, respectively.

The threading of $S_i$ is seen to be such that the zero of $\bar{N}[j']$ is threaded by an extra thread in $\Theta_3$, as required. To finally prove that $j = j'$ suppose that $j < j'$ or $j' < j$. If $j < j'$ there must be $7c((t + 1) + j')$ 1-threads between $\theta_1$ and $\theta$, and $7c((t + 1 - j) + (t + 1))$ 1-threads between $\theta$ and $\theta_4$, that is, there are $7c(3(t + 1) + j' - j) > 3c$
3 \cdot 7c(t + 1) + 7c 1\text{-}threads between \( \theta_1 \) and \( \theta_4 \). But the number of ones in \( T \) between \( \theta_1 \) and \( \theta_4 \) is \( 3 \cdot 7c(t + 1) + 4r \). Thus the number of extra threads is \( \geq 3c \), a contradiction. Similarly, if \( j' < j \), then we see that there must be \( 7c((t + 1) + j - j') \geq 7c(t + 1) + 7c \) 1\text{-}threads between \( \theta_2 \) and \( \theta_3 \), but this interval of \( T \) contains at most \( 7c(t - 1) - 4r \) ones, which again leads to a contradiction. The possibilities not shown in Fig. 8 can be considered similarly. Thus \( j = j' \) which completes the proof of (i) and the proof of the lemma.

The following lemma is an immediate consequence of the construction of threading scheme \( \Theta_3 \).

**Lemma 5.** In threading scheme \( \Theta_3 \), if for some \( S_i \) block \( E[i]^L \) is not to the left of \( \tilde{N}_s \) or \( E[i]^R \) is not to the right of \( \tilde{N}^R_s \), then either of the two zeros in node codes \( \tilde{N}[j], \tilde{N}[m] \) is threaded by an extra 0-thread and all the zeros on this thread belong to node codes for the same node.

**Lemma 6.** In threading scheme \( \Theta_3 \), if for some \( S_i \) block \( E[i]^L \) is to the left of \( \tilde{N}_s^L \), then the pair of ones in \( E[i]^L \) is threaded by private threads, and similarly, if \( E[i]^R \) is to the right of \( \tilde{N}_s^R \), then the pair of ones in \( E[i]^R \) is threaded by private threads.

**Proof.** We consider here only the case where \( E[i]^L \) is to the left of \( \tilde{N}_s^L \). From Lemma 2 we know that in \( \Theta_3 \) the two ones in \( E[i]^L \) are threaded by extra threads. Suppose that one of them, say \( \theta \), is shared with \( S_r \). Since \( \theta \) is to the left of \( \tilde{N}_s^L \), Lemma 4 implies that the part of \( S_r \), to the right of \( E[i]^L \) must be to the right of \( E^L \). Consequently, \( \theta \) must thread with a one in \( E[i]^L \). Then also \( E[i]^L \) must be to the left of \( \tilde{N}_s^L \). But this easily implies that there must be at least \( 7c \) extra 0-threads to the left of \( \tilde{N}_s^L \) in \( \Theta_3 \), a contradiction. Hence such an \( i' \) cannot exist.

**Lemma 7.** There is a common supersequence \( T_4 \) of \( R \) and a threading scheme \( \Theta_3 \) for \( T_4 \) such that

(i) \( |T_4| \leq |T_3| \),

(ii) every \( S_i \) in \( R \) has two private 1-threads, and

(iii) for every \( S_i \), either of the two zeros in node codes \( \tilde{N}[j] \) and \( \tilde{N}[m] \) is threaded by an extra 0-thread and all the zeros on this thread belong to node codes for the same node.

**Proof.** According to Lemmas 2, 5 and 6 scheme \( \Theta_3 \) satisfies conditions (ii) and (iii) except for those strings \( S_i \) where both \( E[i]^L \) is to the left of \( \tilde{N}_s^L \) and \( E[i]^R \) is to the right of \( \tilde{N}_s^R \) in \( \Theta_3 \). From Lemmas 3 and 6 we know that every such \( S_i \) has two private 1-threads in \( E[i]^L \) and no extra threads in \( \tilde{N}[j] ; \tilde{N}[m] \). Using the same matching as in constructing \( \Theta_3 \) we may now thread \( E[i]^L \) with \( \tilde{E}_s^L, \tilde{N}[j] \) with \( \tilde{N} \) and \( \tilde{N}[m] \) with \( \tilde{N}_s^R \). A new extra 0-thread is possibly needed for the zero of \( \tilde{N}[j] \). However, the two private threads for the ones in \( E[i]^L \) become empty and can be removed. Thus the
supersequence determined by the new threading is shorter than the original. When
the changes in \( \Theta_3 \) are done for all such \( S_i \), we finally obtain \( \Theta_4 \) and \( T_4 \) which satisfy
the conditions of the lemma.

**Proof of Claim 2 (continued).** To conclude the proof we must still show how \( T_4 \) and
\( \Theta_4 \) indicate that \( G \) has a node cover of size \( < k \). Let

\[
C = \{ N[g] \mid R \text{ has a string } \xi; \text{ containing node code } N[g] \text{ and the zero of } N[g] \\
\text{is threaded by an extra thread in } \Theta_4 \}.
\]

Lemma 7(iii) implies that the nodes having a code in \( C \) constitute a node cover of
\( G \). Let \( k' \) be the size of this cover. Then we have

\[
k' = \text{number of extra 0-threads in } \Theta_4 \\
= \text{number of all extra threads in } \Theta_4 - \text{number of extra 1-threads in } \Theta_4.
\]

Since the number of all extra threads equals \( |T_4| - q \), and by Lemma 7(ii), the number
of extra 1-threads is \( \geq 2r \), we obtain

\[
k' \leq |T_4| - q - 2r \leq q + (2r + k) - q - 2r = k.
\]

**Proof of Theorem (continued).** Claims 1 and 2 above suffice to show that \( R \) has a
common supersequence of length \( \leq q + (2r + k) \) if and only if \( G \) has a node cover of
size \( \leq k \). Clearly, strings \( R \) can be generated in a time which depends polynomially
on the size of the instance of the node cover problem. Thus we have a polynomial
time transformation of an NP-complete problem to the SCS problem over a binary
alphabet. This problem is therefore NP-complete, too.

4. Conclusions

The SCS problem over an alphabet with size \( \geq 5 \) has been shown to be NP-
complete by Maier [5], who also conjectured that his techniques could be used to
prove the result for alphabet size \( \geq 3 \). In this paper we have used a simplified form
of the encodings of Maier to prove NP-completeness of the SCS problem over any
alphabet with size \( \geq 2 \). The special case where the size of the alphabet is 2 has found
an interesting application in the field of evaluation of attribute grammars [6].

References