Theoretical Computer Science 16 (1981) 187–198
North-Holland Publishing Company

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 $S_1, S_2, \ldots, S_u$ 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 $\geqslant 2$.

1. Introduction

Given a string S over an alphabet Σ, we define a *supersequence* S' of S to be any string $S' = w_0x_1w_1x_2w_2 \cdots x_kw_k$ over Σ such that $S = x_1x_2 \cdots x_k$ and each $w_i \in \Sigma^*$. A *common supersequence* of a set of strings $R = \{S_1, S_2, \ldots, S_u\}$ is a string S over Σ 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 Σ, a finite set R of strings from Σ^*, and a positive integer k, is there a common supersequence of R of length $\leqslant k$? 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 Σ is $\geqslant 2$. The LCS problem is, of course, trivially solvable in polynomial time when Σ 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 Σ is $\geqslant 5$.

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 $\geqslant 2$, 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 $\leqslant 2$ [3], or if the size of Σ is 1.

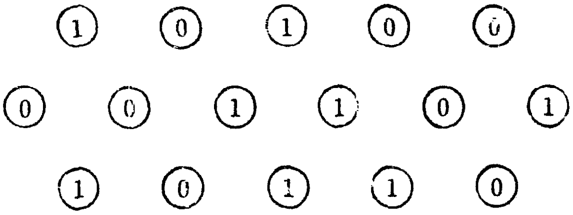
* The work of this author was supported by the Academy of Finland.

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 Σ. 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:



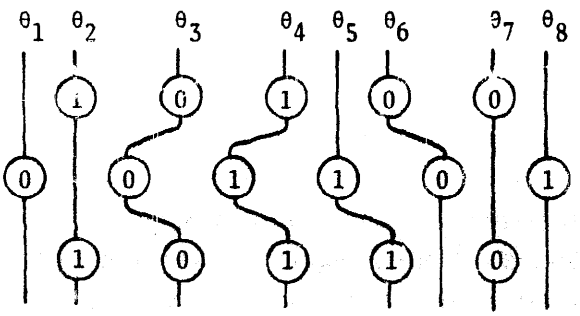
To construct a common supersequence all the beads are threaded so that

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

(ii) all beads on a thread must have the same label from Σ, 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 ≥ 8:



It is convenient to refer to a thread by its type or by the terms of the strings it threads. Thus in the example θ_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* Σ *of size* $\geqslant 2$.

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 $N' \subset N$ 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 $N = \{v_1, v_2, \ldots, v_t\}$, $E = \{(x_1, y_1), (x_2, y_2), \ldots, (x_r, y_r)\}$. We construct a set R of $r + 1$ 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 $\geqslant 5$.

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 ith and $(i + 1)$st blocks of $\bar{N}$. The *multiple node code* $\bar{N}[i_1, i_2, \ldots, i_s]$ has a zero in the i_1st, i_2nd, $\ldots$, and i_sth spots. The special case of $\bar{N}[1, 2, \ldots, t]$ 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* $\bar{E}[j_1, j_2, \ldots, j_s]$ 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: $\bar{E}$; $\bar{N}_s$; $\bar{E}_s$; $\bar{N}$; $\bar{E}_s$; $\bar{N}_s$; $\bar{E}$. 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

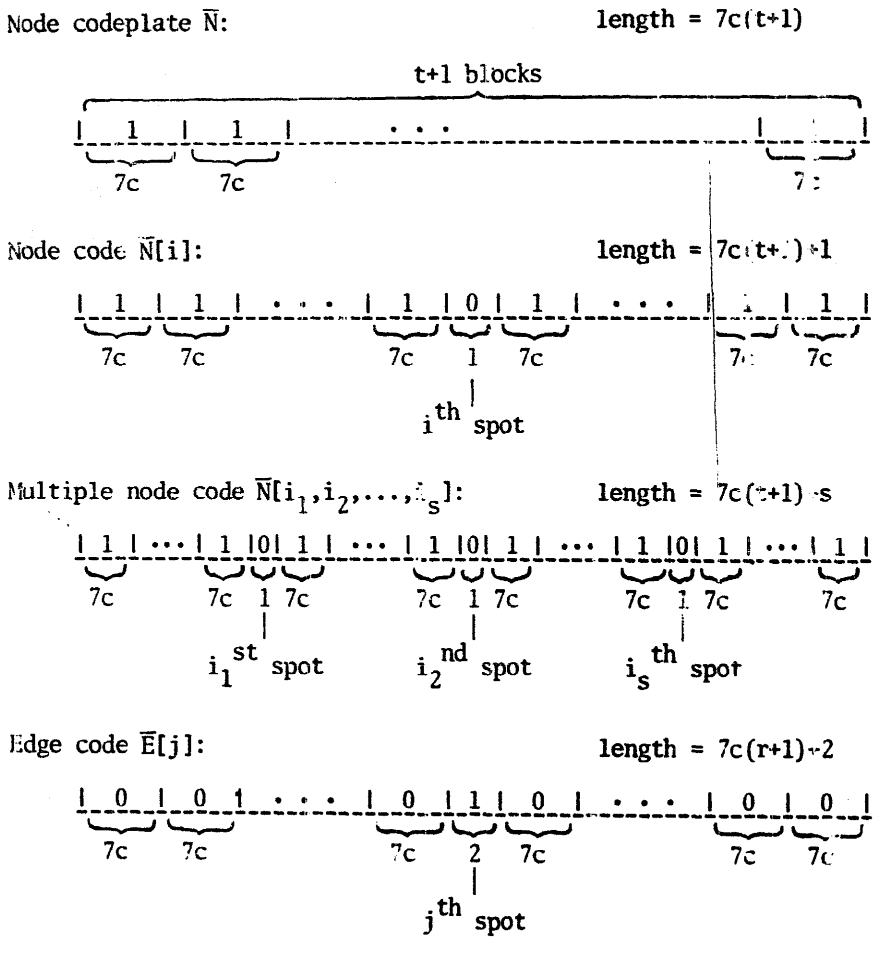


Fig. 1.

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

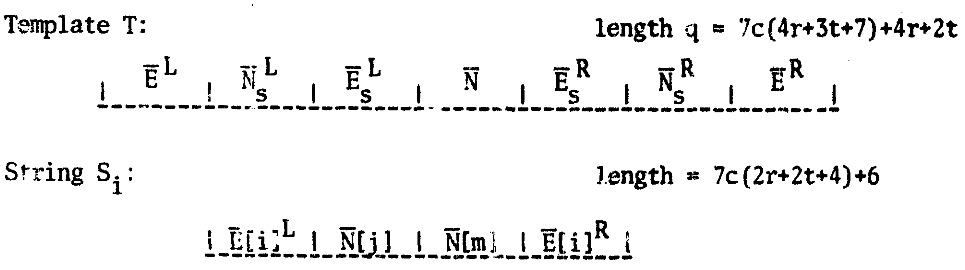

Template T:
length $q = 7c(4r+3t+7)+4r+2t$
$\bar{E}^L$ $\bar{N}_s^L$ $\bar{E}_s^L$ $\bar{N}$ $\bar{E}_s^R$ $\bar{N}_s^R$ $\bar{E}^R$
String S_i:
length = $7c(2r+2t+4)+6$
$\bar{E}[i]^L$ $\bar{N}[j]$ $\bar{N}[m]$ $\bar{E}[i]^R$

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 $W = \{e_i \mid e_i = (x_i, y_i)$ 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 $T' = \bar{E}[U]; \bar{N}_s; E_s; \bar{N}[N']; \bar{E}_s; \bar{N}_s; \bar{E}[W]$. 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).

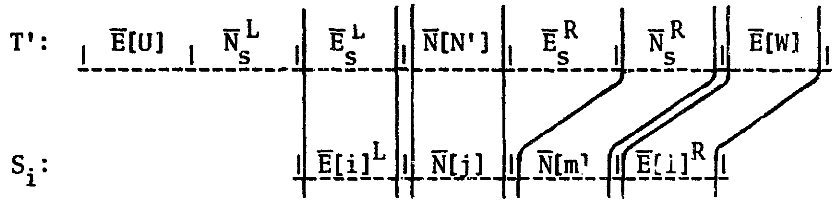


Fig. 3.

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

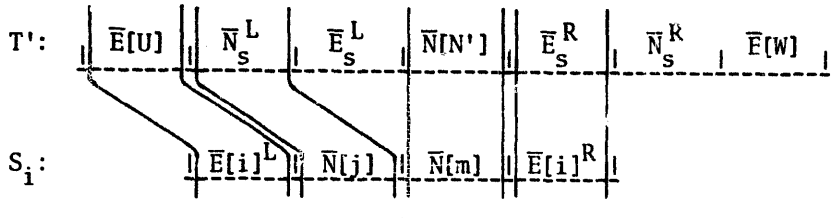


Fig. 4.

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 $\leqslant k$.

Proof. The set N is trivially a node cover for G. Therefore, if $k \geqslant t$, 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 Θ_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 Θ_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 Θ_4, we may decide that a node cover of size $\leqslant k$ for G exists. Each new T_i and Θ_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 *T-thread* and the other threads are called *extra threads*. Scheme Θ_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 $\geqslant 3c$. 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 θ_i and a term b of (possibly another) string be threaded by θ_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 Θ. 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 $\bar{E}[i]^{\mathrm{L}}$ is to the left of $\bar{E}_{\mathrm{s}}^{\mathrm{L}}$ or block $\bar{E}[i]^{\mathrm{R}}$ is to the right of $\bar{E}_{\mathrm{s}}^{\mathrm{R}}$ in Θ_0*

Proof. If the lemma is not true, then the block $\bar{N}[j]$; $\bar{N}[m]$ of S_i must be to the right of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$ and to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{R}}$ in Θ_0. So we have the situation given in Fig. 5. Since $\bar{N}[j]$ $\bar{N}[m]$ contains $14c(t+1)$ ones and there are only $7c(t+1)+4r$ ones between $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$ and $\bar{N}_{\mathrm{s}}^{\mathrm{R}}$ in T, this means that $7c(t+1)-4r>3c$ ones in $\bar{N}[j]$; $\bar{N}[m]$ must be on extra threads, a contradiction.

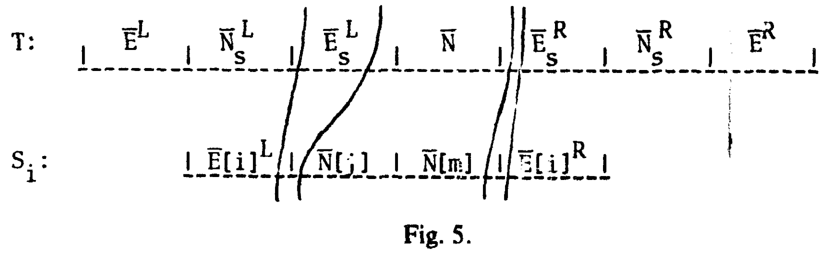


Fig. 5.

Lemma 2. *There is a common supersequence T_1 of R and a threading scheme Θ_1 for T_1 such that $|T_1|\leqslant|T_0|$ and for each S_i, block $\bar{E}[i]^{\mathrm{L}}$ is to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$ and the pair of ones in $\bar{E}[i]^{\mathrm{L}}$ is threaded by extra threads in Θ_1, or block $\bar{E}[i]^{\mathrm{R}}$ is to the right of $\bar{N}_{\mathrm{s}}^{\mathrm{R}}$ and the pair of ones in $\bar{E}[i]^{\mathrm{R}}$ is threaded by extra threads in Θ_1. Moreover, if $\bar{E}[i]^{\mathrm{L}}$ is not to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$, then it is not to the left of $\bar{E}_{\mathrm{s}}^{\mathrm{L}}$, and if $\bar{E}[i]^{\mathrm{R}}$ is not to the right of $\bar{N}_{\mathrm{s}}^{\mathrm{R}}$, then it is not to the right of $\bar{E}_{\mathrm{s}}^{\mathrm{R}}$ in Θ_1.*

Proof. Suppose that $\bar{E}[i]^{\mathrm{L}}$ is to the left of $\bar{E}_{\mathrm{s}}^{\mathrm{L}}$ in Θ_0. To prove the lemma we show that it is possible to transform T_0 and Θ_0 so that in the new threading $\bar{E}[i]^{\mathrm{L}}$ is to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$. This transformation (and a symmetric transformation for those $\bar{E}[i]^{\mathrm{R}}$ that are to the right of $\bar{E}_{\mathrm{s}}^{\mathrm{R}}$ in Θ_0) can be done successively for each $\bar{E}[i]^{\mathrm{L}}$ to the left of $\bar{E}_{\mathrm{s}}^{\mathrm{L}}$ and each $\bar{E}[i]^{\mathrm{R}}$ to the right of $\bar{E}_{\mathrm{s}}^{\mathrm{R}}$ in Θ_0. The resulting supersequence T_1 and threading scheme Θ_1 are as required in the lemma because by Lemma 1, either $\bar{E}[i]^{\mathrm{L}}$ or $\bar{E}[i]^{\mathrm{R}}$ satisfies the condition of the transformation for every S_i. Note that if $\bar{E}[i]^{\mathrm{L}}$ or $\bar{E}[i]^{\mathrm{R}}$ does not satisfy this condition, then it necessarily satisfies the last assertion of the lemma.

Before modifying T_0 and Θ_0 for $\bar{E}[i]^{\mathrm{I}}$ we show that the two ones in $\bar{E}[i]^{\mathrm{L}}$ are to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$. In fact, if this is not true, then all zeros of $\bar{E}[i]^{\mathrm{L}}$ to the right of the pair of ones must be threaded in Θ_0 to the right of $\bar{E}^{\mathrm{L}}$ but to the left of $\bar{E}_{\mathrm{s}}^{\mathrm{L}}$. This means that Θ_0 has at least $7c-t>3c$ extra threads because $\bar{E}[i]^{\mathrm{L}}$ has at least $7c$ zeros to the right of the two ones but $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$ contains only t zeros, a contradiction. Hence the two ones of $\bar{E}[i]^{\mathrm{L}}$ must be to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$ in Θ_0. This clearly implies that these ones are threaded by extra threads in Θ_0.

Therefore, if $\bar{E}[i]^{\mathrm{L}}$ is to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$ already in Θ_0, no modification of Θ_0 is needed for $\bar{E}[i]^{\mathrm{L}}$. Otherwise $\bar{E}[i]^{\mathrm{I}}$ has zeros that are not to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$ in Θ_0. We will move them on suitable threads which already thread $\bar{E}^{\mathrm{L}}$. In more detail, the $7c\cdot i$ zeros to the left of the pair of ones in $\bar{E}[i]^{\mathrm{L}}$ are first threaded with the $7c\cdot i$ leftmost zeros of $\bar{E}^{\mathrm{L}}$. This introduces no extra threads. Then two extra threads are added to thread the two ones of $\bar{E}[i]^{\mathrm{L}}$ 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 α_h be a prefix of S_h such that the last term of α_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 $\bar{E}[i]^{\mathrm{L}}$ is to the left of $\bar{E}_{\mathrm{s}}^{\mathrm{L}}$ in Θ_0, that string α_h must always be of the form $0^{7c\cdot h}11$ 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 θ in S_h and between θ and θ' in S_i.

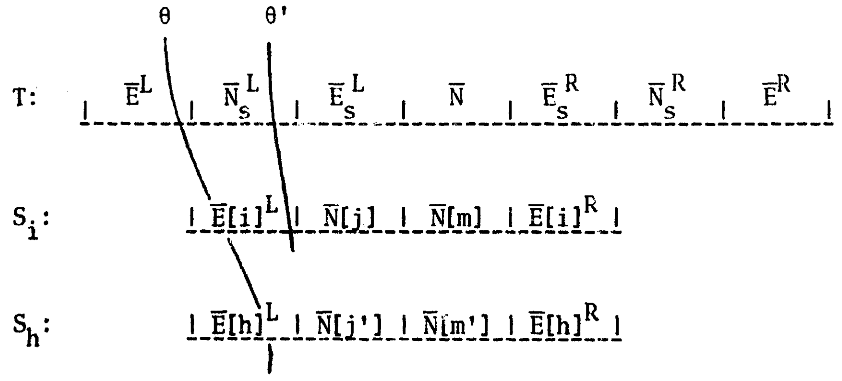


Fig. 6.

Thus α_h is a subsequence of the prefix $\beta=0^{7c\cdot i}11$ of S_i which we have threaded so far. Hence α_h can be threaded by the threads of β. 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 $\bar{E}[i]^{\mathrm{L}}$ are finally threaded with the $7c((r+1)-i)$ rightmost zeros of $\bar{E}^{\mathrm{L}}$. In the resulting threading $\bar{E}[i]^{\mathrm{L}}$ is to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$ and the pair of ones is threaded by extra threads. Moreover, the corresponding common supersequence is of length $\leqslant|T_0|$. Repeating this process for each S_i we arrive at T_1 and Θ_1 which are as required.

Lemma 3. *There is a common supersequence T_2 of R and a threading scheme Θ_2 for T_2 such that $|T_2| \leq |T_1|$ and Θ_2 is as Θ_1 except if for some string S_i block $\bar{E}[i]^{\mathrm{L}}$ is to the left of $\bar{N}_s^{\mathrm{L}}$ and $\bar{E}[i]^{\mathrm{R}}$ to the right of $\bar{N}_s^{\mathrm{R}}$ in Θ_1, then no term of $\bar{N}[j]$ or $\bar{N}[m]$ of S_i is threaded by extra threads in Θ_2.*

Proof. For every such string S_i, we may change Θ_1 so that $\bar{N}[j]$ is threaded with $\bar{N}_s^{\mathrm{L}}$ and $\bar{N}[m]$ with $\bar{N}_s^{\mathrm{R}}$. Since $\bar{N}[j]$ and $\bar{N}[m]$ are subsequences of $\bar{N}_s$, this can be done without introducing new threads but some threads of Θ_1 may become empty. These should be removed. When these changes are successively made for every possible S_i, the resulting threading scheme Θ_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 Θ_3 for T_3 such that $|T_3| \leq |T_2|$ and Θ_3 is as Θ_2 except if for some S_i block $\bar{E}[i]^{\mathrm{L}}$ is not to the left of $\bar{N}_s^{\mathrm{L}}$ in Θ_2, then $\bar{E}[i]^{\mathrm{L}}$ is to the right of $\bar{N}_s^{\mathrm{L}}$ in Θ_3, and symmetrically, if block $\bar{E}[i]^{\mathrm{R}}$ is not to the right of $\bar{N}_s^{\mathrm{R}}$ in Θ_2, then $\bar{E}[i]^{\mathrm{R}}$ is to the left of $\bar{N}_s^{\mathrm{R}}$ in Θ_3.*

Proof. We first define T_3 and Θ_3. Suppose that $\bar{E}[i]^{\mathrm{L}}$ is not to the left of $\bar{N}_s^{\mathrm{L}}$ in Θ_2. Then by Lemma 2, $\bar{E}[i]^{\mathrm{R}}$ must be to the right of $\bar{N}_s^{\mathrm{R}}$ in Θ_2. Scheme Θ_2 is now modified such that $\bar{E}[i]^{\mathrm{L}}$ is threaded with $\bar{E}_s^{\mathrm{L}}$, $\bar{N}[j]$ with $\bar{N}$ and $\bar{N}[m]$ with $\bar{N}_s^{\mathrm{R}}$; the threading of $\bar{E}[i]^{\mathrm{R}}$ remains unchanged. In this process the only extra thread is needed to thread the zero appearing in $\bar{N}[j]$. This thread θ crosses $\bar{N}$ between the $7c \cdot j$th and $(7c \cdot j+1)$st one. However, if there is already an extra 0-thread θ' at this place we use θ' for threading the zero of $\bar{N}[j]$, and θ is not introduced. The symmetric case in which $\bar{E}[i]^{\mathrm{R}}$ is not to the right of $\bar{N}_s^{\mathrm{R}}$ but $\bar{E}[i]^{\mathrm{L}}$ is to the left of $\bar{N}_s^{\mathrm{L}}$ in Θ_2 is handled analogously: $\bar{E}[i]^{\mathrm{R}}$ is threaded with $\bar{E}_s^{\mathrm{R}}$, $\bar{N}[m]$ with $\bar{N}$ and $\bar{N}[j]$ with $\bar{N}_s^{\mathrm{L}}$. The only term which uses an extra thread is now the zero in $\bar{N}[m]$.

Applied to every possible S_i the above process yields a scheme Θ_3 which obviously threads every $\bar{E}[i]^{\mathrm{L}}$ and $\bar{E}[i]^{\mathrm{R}}$ 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 Θ_3 but possibly not in Θ_2 are the 0-threads threading the zeros of those $\bar{N}[g]$ that are threaded with $\bar{N}$ in Θ_3. In addition, if S_i and $S_{i'}$ share such an extra thread, then both S_i and $S_{i'}$ must contain the block $\bar{N}[g]$ 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 $\bar{N}[g]$ in S_i has an extra 0-thread in Θ_3, then the same $\bar{N}[g]$ has an extra 0-thread in Θ_2, too, and

(ii) If the extra 0-thread of such a block $\bar{N}[g]$ is shared in Θ_2 with the zero of another block $\bar{N}[g']$, then it is shared with the same zero in Θ_3, too.

To prove (i) suppose that $\bar{N}[j]$ of S_i has an extra thread in Θ_3; the case where $\bar{N}[m]$ has an extra thread can be considered symmetrically. Then, by the construction of Θ_3, $\bar{E}[i]^{\mathrm{L}}$ cannot be to the left of $\bar{N}_s^{\mathrm{L}}$ in Θ_2 but $\bar{E}[i]^{\mathrm{R}}$ is to the right of $\bar{N}_s^{\mathrm{R}}$. Lemma 2 implies that $\bar{E}[i]^{\mathrm{L}}$ is not to the left of $\bar{E}_s^{\mathrm{L}}$ in Θ_2. If the 0-thread θ threading the zero of $\bar{N}[j]$ in Θ_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 situation in Θ_2 is as shown in Fig. 7. We note that there are to the right of θ at most $7c(t+1)+2r$ ones in T but at least $7c(t+1)+7c$ ones in S_i. Hence Θ_2 should have at least $7c-2r>3c$ extra 1-threads, a contradiction. Similarly, if θ is to the left of $\bar{N}$, then again a contradiction follows. Thus we have proved (i).

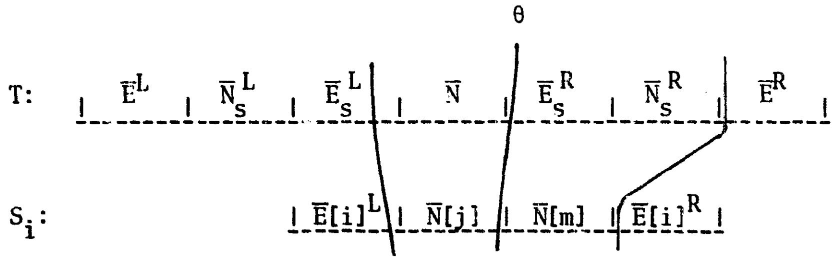


Fig. 7.

To prove (ii) let $\bar{N}[j]$ and θ be as in the proof of (i) and let θ be shared with the zero of $\bar{N}[j']$ in Θ_2. We prove that $j=j'$ and that $\bar{N}[j']$ has an extra 0-thread in Θ_3. This proves (ii) because then the zeros of $\bar{N}[j]$ and $\bar{N}[j']$ must be threaded by the same thread in Θ_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 $\bar{E}_s^L$ or $\bar{E}[i']^R$ is not to the right of $\bar{E}_s^R$. Assume that $\bar{E}[i']^R$ is not to the right of $\bar{E}_s^R$; the other case can be considered similarly. Then $S_{i'}$ must be of the form $\bar{E}[i']^L$; $\bar{N}[m']$; $\bar{N}[j']$; $\bar{E}[i']^R$, 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 $\bar{E}[i]^L$ and the first threads of $\bar{E}[i']^R$ and $\bar{E}[i]^R$, respectively.

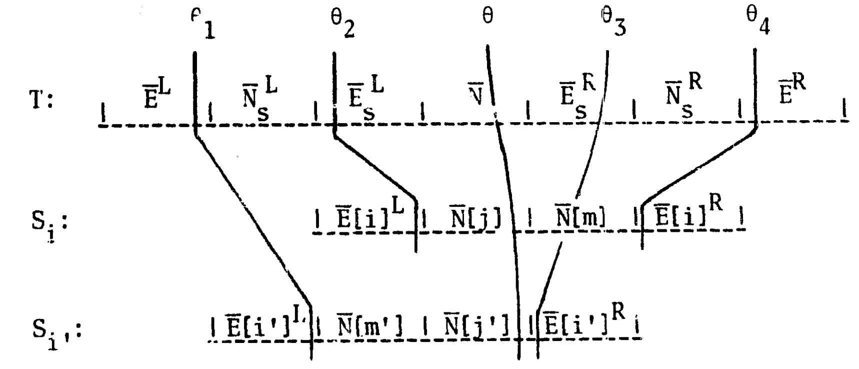


Fig. 8.

The threading of $S_{i'}$ is seen to be such that the zero of $\bar{N}[j']$ is threaded by an extra thread in Θ_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 θ_1 and θ, and $7c((t+1-j)+(t+1))$ 1-threads between θ and θ_4, that is, there are $7c(3(t+1)+j'-j)\geq$

$3\cdot 7c(t+1)+7c$ 1-threads between θ_1 and θ_4. But the number of ones in T between θ_1 and θ_4 is $3\cdot 7c(t+1)+4r$. Thus the number of extra threads is $\geqslant 3c$, a contradiction. Similarly, if $j'<j$, then we see that there must be $7c((t+1)+j-j')\geqslant 7c(t+1)+7c$ 1-threads between θ_2 and θ_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 (ii) and the proof of the lemma.

The following lemma is an immediate consequence of the construction of threading scheme Θ_3.

Lemma 5. *In threading scheme Θ_3, if for some S_i block $\bar{E}[i]^{\mathrm{L}}$ is not to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$ or $\bar{E}[i]^{\mathrm{R}}$ is not to the right of $\bar{N}_{\mathrm{s}}^{\mathrm{R}}$, then either of the two zeros in node codes $\bar{N}[j]$, $\bar{N}[m]$ 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 Θ_3, if for some S_i block $\bar{E}[i]^{\mathrm{L}}$ is to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$, then the pair of ones in $\bar{E}[i]^{\mathrm{L}}$ is threaded by private threads, and similarly, if $\bar{E}[i]^{\mathrm{R}}$ is to the right of $\bar{N}_{\mathrm{s}}^{\mathrm{R}}$, then the pair of ones in $\bar{E}[i]^{\mathrm{R}}$ is threaded by private threads.*

Proof. We consider here only the case where $\bar{E}[i]^{\mathrm{L}}$ is to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$. From Lemma 2 we know that in Θ_3 the two ones in $\bar{E}[i]^{\mathrm{L}}$ are threaded by extra threads. Suppose that one of them, say θ, is shared with $S_{i'}$. Since θ is to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$, Lemma 4 implies that the part of $S_{i'}$, to the right of $\bar{E}[i']^{\mathrm{L}}$ must be to the right of $\bar{E}^{\mathrm{L}}$. Consequently, θ must thread with a one in $\bar{E}[i']^{\mathrm{L}}$. Then also $\bar{E}[i']^{\mathrm{L}}$ must be to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$. But this easily implies that there must be at least $7c$ extra 0-threads to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$ in Θ_3, a contradiction. Hence such an i' cannot exist.

Lemma 7. *There is a common supersequence T_4 of R and a threading scheme Θ_4 for T_4 such that*

(i) $|T_4|\leqslant|T_3|$,

(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 $\bar{N}[j]$ and $\bar{N}[m]$ 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 Θ_3 satisfies conditions (ii) and (iii) except for those strings S_i where both $\bar{E}[i]^{\mathrm{L}}$ is to the left of $\bar{N}_{\mathrm{s}}^{\mathrm{L}}$ and $\bar{E}[i]^{\mathrm{R}}$ is to the right of $\bar{N}_{\mathrm{s}}^{\mathrm{R}}$ in Θ_3. From Lemmas 3 and 6 we know that every such S_i has two private 1-threads in $\bar{E}[i]^{\mathrm{L}}$ and no extra threads in $\bar{N}[j]$; $\bar{N}[m]$. Using the same matching as in constructing Θ_3 we may now thread $\bar{E}[i]^{\mathrm{L}}$ with $\bar{E}_{\mathrm{s}}^{\mathrm{L}}$, $\bar{N}[j]$ with $\bar{N}$ and $\bar{N}[m]$ with $\bar{N}_{\mathrm{s}}^{\mathrm{R}}$. A new extra 0-thread is possibly needed for the zero of $\bar{N}[j]$. However, the two private threads for the ones in $\bar{E}[i]^{\mathrm{L}}$ become empty and can be removed. Thus the supersequence determined by the new threading is shorter than the original. When the changes in Θ_3 are done for all such S_i, we finally obtain Θ_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 Θ_4 indicate that G has a node cover of size $\leqslant k$. Let

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

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

$$\begin{aligned} k' &\leqslant \text{number of extra 0-threads in } \Theta_4 \\ &= \text{number of all extra threads in } \Theta_4 - \text{number of extra 1-threads in } \Theta_4. \end{aligned}$$

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

$$k' \leqslant |T_4| - q - 2r \leqslant q + (2r + k) - q - 2r = k.$$

Proof of Theorem (continued). Claims 1 and 2 above suffice to show that R has a common supersequence of length $\leqslant q + (2r + k)$ if and only if G has a node cover of size $\leqslant k$. 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 $\geqslant 5$ 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 $\geqslant 3$. 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 $\geqslant 2$. 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

[1] A.V. Aho, D.S. Hirschberg and J.D. Ullman, Bounds on the complexity of the longest common subsequence problem, *J. ACM* **23** (1) (1976) 1–12.

[2] S.A. Cook, The complexity of theorem proving procedures, *Proc. 3rd Annual ACM Symposium on Theory of Computing* (1971) 151–158.

[3] M.R. Garey and D.S. Johnson, *Computers and Intractability* (Freeman, San Francisco, CA, 1979).

[4] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller and J.W. Thatcher, Eds., *Complexity of Computer Computation* (Plenum, New York, 1972) 85–103.

[5] D. Maier, The complexity of some problems on subsequences and supersequences, *J. ACM* **25** (2) (1978) 322–336.

[6] K.-J. Räihä and E. Ukkonen, Minimizing the number of evaluation passes for attribute grammars, *SIAM J. Comput.*, to appear.

[7] R.A. Wagner and M.J. Fischer, The string-to-string correction problem, *J. ACM* **21** (1) (1974) 168–173.