TECHNICAL UNIVERSITY OF DENMARK - DTU COMPUTE -

[SUPERSTRINGWITHEXPANSION] Problem

Salik Lennert Pedersen
Anders Holmgaard Opstrup
Federico Bergamin

1 DESCRIPTION OF THE PROBLEM

In this decisional problem we have two different alphabet Σ and Γ, where $\Gamma = \{\gamma_1, ..., \gamma_m\}$ contains m symbol, a string s which is made up of symbols that belongs to the Σ alphabet (formally $s \in \Sigma^*$), k strings $t_1, ..., t_k$, which is made up of symbols from both alphabets, and in the end, we have m subsets $R_1, ..., R_m \subseteq \Sigma^*$. These subsets contain symbols of Σ alphabet, and as we can notice, they are as many as the symbols in the alphabet Γ. That's because for each γ_i there is a subset R_i. We can understand better this relation explaining the core of our problem. The output of the problem will be YES if exits a sequence of words $r_1 \in R_1$, $r_2 \in R_2$,...,$r_m \in R_m$, such that for all $1 \leq i \leq k$ the so-called *expansion* $e(t_i)$ is substring of s; the expansion of a string t_i consists in the replacement of every symbols $\gamma_j \in \Gamma$ that appears in the string t_i with its expansion, where the expansion of a symbol is defined as follow: $e(\gamma_j) := r_j$. That is, we have to choose for every symbol $\gamma_j \in \Gamma$ a symbol $r_j \in R_j$, and we know that $r_j \in \Sigma^*$. We have to replace the symbol γ_j in each string t_i with the symbol r_j that was chosen. We have to do that for each $\gamma \in \Gamma$. In this way, using these replacements, we are transforming our string t_i in its expansion $e(t_i)$.
In the end we will obtain that every expansion-string is made up of symbols of the alphabet Σ as our string s. Hence, if every new constructed string $e(t_i)$ is a substring of s, the answer for our problem will be YES, otherwise NO.
In other words, the answer of our problem will be YES if there is a selection of the $r_i \in R_i$ such that, replacing each symbols $\gamma_i \in \Gamma$ with the respective symbol r_i (we have to remember that for each γ_i we have to chose one symbol from the subset $R_i \subseteq \Sigma^*$) in every string $t \in T$, where $|T| = k$, we will have that every new string that we obtain with this replacement is a substring of the string s. If a selection of $r_i \in R_i$ like this doesn't exist, the answer will be NO.

1.1 Simple examples

With the text of the problem we receive also some problem instances on the alphabets $\Sigma = \{\mathtt{a,b,...,z}\}$, $\Gamma = \{\mathtt{A,B,...,Z}\}$. The first line of the file contains the number k, which is the number of strigs t we have. The second line contains the string s and the following k lines the strings $t_1,...,t_k$. Finally, the last lines (at most 26, the number of letters in the alphabet) start with a letter $\gamma_j \in \Gamma$ followed by a colon and the contents of the set R_j belonging to the letter, where the elements of the set are separated by commas. Hence, we can clearly see that for each symbols $\gamma_i \in \Gamma$, in this case letter, there is a subset R_i and in this subset we have to chose one symbol.

For example:

```
4
abdde
ABD
DDE
AAB
ABd
A:a,b,c,d,e,f,dd
B:a,b,c,d,e,f,dd
C:a,b,c,d,e,f,dd
D:a,b,c,d,e,f,dd
E:aa,bd,c,d,e
```

In this example we can observe that there is NO possible selection of symbols r_i such that every string t would be a substring of `s=abdde`. In fact if we have a look to t_2=`DDE` and t_3=`AAB`, we can notice that they have the same structure, and in the same time in our string s the only letter that is repeated twice is the `d`. So we conclude that `A=d` and `D=d`. Moreover from t_4=`ABd` we see that, since t_4 will be a substring, or `B=d` or `B=b`. If we try to choose our selection of r_i following these rules, we can noticed that there is no possible solution, such that at the same time, all $e(t_i)$ are substring of s.

2 SuperStringWithExpansion is in $\mathcal{NP}$

We have to show and prove that our problem is in $\mathcal{NP}$. To do that we use the so called *guess-verify algorithms* proof, so we have to start designing a deterministic algorithm A which takes as input a problem instance X and a random sequence R.

Proof. We have to show that there is a polynomial p and a randomized p-bounded algorithm A which satisfies the condition of the class $\mathcal{NP}$.

□