Applied Mathematical Sciences, Vol. 6, 2012, no. 58, 2851 - 2855

The Longest Common Parameterized Subsequence Problem

Anna Gorbenko

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
gorbenko.ann@gmail.com

Vladimir Popov

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
Vladimir.Popov@usu.ru

Abstract

In this paper we describe an approach to solve the problem of the longest common parameterized subsequence. This approach is based on constructing logical models for LCPS.

Mathematics Subject Classification: 03D15, 68Q17

Keywords: parameterized pattern matching, logical models, **NP**-complete

1 Introduction

The problem of the longest common subsequence (LCS), of two strings of lengths n and m respectively, is well-known. It is solvable in $O(nm)$-time. It is a classical distance measure for strings. Another well-studied string comparison measure is that of parameterized matching. In this problem two equal-length strings are a parameterized-match if there exists a bijection on the alphabets such that one string matches the other under the bijection [1]. Note that most of the works associated with parameterized pattern matching present polynomial time algorithms (e.g. [1] – [4]). There have been several attempts to accommodate parameterized matching along with some other distance measures, as these turn out to be natural problems. In particular, Hamming distance [3] and a bounded version of edit-distance [4] are considered. Several

algorithms have been proposed for these problems. In this paper we consider the problem of the longest common parameterized subsequence (LCPS) which combines the LCS measure with parameterized matching. This problem was first considered in [5]. In [5] proved that LCPS is **NP**-hard. In this paper we describe an approach to solve the problem.

2 Problem Definition

A model of parameterized pattern matching was introduced in [1]. The main motivation for this scheme lies in software maintenance, where programs are to be considered "identical" even if variable names are different. Therefore, strings under this model are comprised of symbols from two disjoint sets Σ and Π containing fixed symbols and variable/parameter symbols respectively. Formally, parameterized pattern matching is as follows (see e.g. [2]). A parameterized string is a string over $\Sigma \cup \Pi$. Two parameterized strings S_1 and S_2 of same length are said to parameterized match if there exists a bijection $f : \Pi_1 \rightarrow \Pi_2$, where Π_1 and Π_2 are the symbols from Π in S_1 and S_2 respectively, such that the following holds: S_1 (S_2, resp.) equals S_2 (S_1, resp.) when any occurrence $x \in \Pi_1$ (Π_2, resp.) is replaced by $f(x)$ ($f^{-1}(x)$, resp.). Given two sequences S and T over some fixed alphabet Ξ, the sequence T is a subsequence of S if T can be obtained from S by deleting some letters from S. Notice that the order of the remaining letters of S bases must be preserved. Similarly, given two sequences S and T over some fixed alphabet $\Sigma \cup \Pi$, the sequence T is a parameterized subsequence of S if T parameterized match U where U can be obtained from S by deleting some letters from S. The length of a sequence S is the number of letters in it and is denoted as $|S|$. For simplicity, we use $S[i]$ to denote the ith letter in sequence S, and $S[i, j]$ to denote the substring of S consisting of the ith letter through the jth letter.

Given sequences S_1 and S_2 over some fixed alphabet $\Sigma \cup \Pi$, the LCPS problem asks for a longest sequence T that is a parameterized subsequence of S_1 and S_2. In the decision version LCPS can be formulated as following:

INSTANCE: *An alphabet $\Sigma \cup \Pi$, sequences S_1, S_2, and positive integer k.*

QUESTION: *Is there a sequence T, $|T| \geq k$, that is a parameterized subsequence of S_1 and S_2?*

3 Logical models of LCPS

The satisfiability problem (SAT) was the first known **NP**-complete problem. The problem SAT is the problem of determining if the variables of a given boolean function in conjunctive normal form (CNF) can be assigned in such a way as to make the formula evaluate to true. Considered also different variants

of SAT. The problem SAT remains **NP**-complete even if all expressions are written in conjunctive normal form with 3 variables per clause (3-CNF). The problem 3SAT is the problem of determining if the variables of a given 3-CNF can be assigned in such a way as to make the formula evaluate to true. In practice, the satisfiability problem is fundamental in solving many problems in automated reasoning, computer-aided design, computer-aided manufacturing, machine vision, database, robotics, integrated circuit design, computer architecture design, and computer network design. In recent years, many optimization methods, parallel algorithms, and practical techniques have been developed for solving the satisfiability problem. It is natural to use a reduction to different variants of the satisfiability problem to solve computational hard problems. Encoding problems as Boolean satisfiability and solving them with very efficient satisfiability algorithms has recently caused considerable interest (e.g. [6, 7, 8, 9, 10]). In this paper we consider reductions from LCPS to SAT and 3SAT. Let $\Sigma = \{a_1, a_2, \ldots, a_{|\Sigma|}\}$, $\Pi = \{b_1, b_2, \ldots, b_{|\Pi|}\}$,

$$\varphi_1 = \wedge_{1\le i\le k}((\vee_{1\le j\le|\Sigma\cup\Pi|}x[i,j]) \wedge (\wedge_{1\le j[1]<j[2]\le|\Sigma\cup\Pi|}(\neg x[i,j[1]] \vee \neg x[i,j[2]]))),$$

$$\varphi_2 = \wedge_{1\le i\le|S_2|}((\vee_{1\le j\le|\Sigma\cup\Pi|}y[i,j]) \wedge (\wedge_{1\le j[1]<j[2]\le|\Sigma\cup\Pi|}(\neg y[i,j[1]] \vee \neg y[i,j[2]]))),$$

$$\varphi_3 = \wedge_{1\le i\le|S_2|,S_2[i]\in\Pi} \wedge_{1\le j\le|\Sigma|} \neg y[i,j],$$

$$\psi_1 = \wedge_{1\le i[1]<i[2]\le|S_2|,S_2[i[1]]=S_2[i[2]]\in\Pi,1\le j\le|\Sigma\cup\Pi|}y[i[1],j] = y[i[2],j],$$

$$\psi_2 = \wedge_{1\le i[1]<i[2]\le|S_2|,S_2[i[1]]\neq S_2[i[2]],S_2[i[1]],S_2[i[2]]\in\Pi,1\le j\le|\Sigma\cup\Pi|}\neg y[i[1],j] \vee \neg y[i[2],j],$$

$$\rho_1 = \wedge_{1\le i\le|S_1|,1\le j\le k,1\le l\le|\Sigma\cup\Pi|,S_1[i]\neq a_l,S_1[i]\neq b_{l-|\Sigma|}}z[1,i,j] \to \neg x[j,l],$$

$$\rho_2 = \wedge_{1\le i\le|S_2|,1\le j\le k,1\le l\le|\Sigma\cup\Pi|,S_2[i]\neq a_l,S_2[i]\in\Sigma}z[2,i,j] \to \neg x[j,l],$$

$$\rho_3 = \wedge_{1\le i\le|S_2|,1\le j\le k,1\le l\le|\Sigma|,S_2[i]\in\Pi}z[2,i,j] \to \neg x[j,l],$$

$$\rho_4 = \wedge_{1\le i\le|S_2|,1\le j\le k,|\Sigma|+1\le l\le|\Sigma\cup\Pi|,S_2[i]\in\Pi}z[2,i,j] \to y[i,l] = x[j,l],$$

$$\tau_1 = \wedge_{1\le i\le2,1\le j\le|S_i|,1\le l[1]<l[2]\le k}\neg z[i,j,l[1]] \vee \neg z[i,j,l[2]],$$

$$\tau_2 = \wedge_{1\le i\le2,1\le l\le k} \vee_{1\le j\le|S_i|} z[i,j,l],$$

$$\tau_3 = \wedge_{1\le i\le2,1\le j[1]\le|S_i|,1\le l[1]\le k,j[2]>j[1],l[2]<l[1]}z[i,j[1],l[1]] \to \neg z[i,j[2],l[2],$$

$$\xi = \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \psi_1 \wedge \psi_2 \wedge \rho_1 \wedge \rho_2 \wedge \rho_3 \wedge \rho_4 \wedge \tau_1 \wedge \tau_2 \wedge \tau_3.$$

Theorem 3.1 *Given a fixed alphabet $\Sigma \cup \Pi$, sequences S_1 and S_2, and positive integer k. There is a sequence T, $|T| \ge k$, that is a parameterized subsequence of S_1 and S_2 if and only if ξ is satisfiable.*

PROOF. Given a fixed alphabet $\Sigma \cup \Pi$, sequences S_1 and S_2, and positive integer k. Suppose that there is a sequence T, $|T| \geq k$, that is a parameterized subsequence of S_1 and S_2. Without loss of generality we can assume that $|T| = k$ and is a subsequence of S_1. Let $x[i,j] = 1$ where $1 \leq i \leq k$, $1 \leq j \leq |\Sigma \cup \Pi|$, $T[i] = a_j$ or $T[i] = b_{j-|\Sigma|}$. Let $x[i,j] = 0$ where $1 \leq i \leq k$, $1 \leq j \leq |\Sigma \cup \Pi|$, $T[i] \neq a_j$ and $T[i] \neq b_{j-|\Sigma|}$. Since T is a parameterized subsequence of S_2, there is a bijection $f : \Pi_1 \to \Pi_2$ which transforms S_2 into a sequence S_3 such that T is a subsequence of S_3. Let $y[i,j] = 1$ where $1 \leq i \leq |S_2|$, $1 \leq j \leq |\Sigma \cup \Pi|$, $S_3[i] = a_j$ or $S_3[i] = b_{j-|\Sigma|}$. Let $y[i,j] = 0$ where $1 \leq i \leq |S_2|$, $1 \leq j \leq |\Sigma \cup \Pi|$, $S_3[i] \neq a_j$ and $S_3[i] \neq b_{j-|\Sigma|}$. Let $z[i,j,k] = 1$ if and only if $S_i[j]$ parameterized match $T[k]$. It can be verified directly that in case of such values of variables $\xi = 1$. Now suppose that $\xi = 1$. Since $\varphi_1 = 1$, it is easy to see that, for all i, there is only one value of j such that $x[i,j] = 1$. So, we can consider values of $x[i,j]$ as a definition of T. Similarly, in view of φ_2, we can consider values of $y[i,j]$ as a definition of S_3. In view of φ_3, $S_2[i] \in \Pi$ if and only if $S_3[i] \in \Pi$. Let $S|_\Pi$ denotes the sequence which obtained from S by deleting all letters from Σ. Since $\psi_1 = 1$ and $\psi_2 = 1$, $S_2|_\Pi$ parameterized match $S_3|_\Pi$. It is easy to check that $\tau_1 = 1$, $\tau_2 = 1$, and $\tau_3 = 1$ guarantee the preservation of order. Since $\rho_1 = 1$, it is clear that T is a subsequence of S_1. In view of $\rho_2 = 1$, $\rho_3 = 1$, $\rho_4 = 1$, T is a parameterized subsequence of S_2. □

Note that

$$\alpha \vee ((\beta \vee \neg\gamma) \wedge (\neg\beta \vee \gamma)) \Leftrightarrow (\alpha \vee \beta \vee \neg\gamma) \wedge (\alpha \vee \neg\beta \vee \gamma), \quad (1)$$

$$\alpha \Leftrightarrow (\alpha \vee \beta_1 \vee \beta_2) \wedge (\alpha \vee \neg\beta_1 \vee \beta_2) \wedge (\alpha \vee \beta_1 \vee \neg\beta_2) \wedge (\alpha \vee \neg\beta_1 \vee \neg\beta_2), \quad (2)$$

$$\vee_{j=1}^{l} \alpha_j \Leftrightarrow (\alpha_1 \vee \alpha_2 \vee \beta_1) \wedge (\wedge_{i=1}^{l-4}(\neg\beta_i \vee \alpha_{i+2} \vee \beta_{i+1})) \wedge (\neg\beta_{l-3} \vee \alpha_{l-1} \vee \alpha_l), \quad (3)$$

$$\alpha_1 \vee \alpha_2 \Leftrightarrow (\alpha_1 \vee \alpha_2 \vee \beta) \wedge (\alpha_1 \vee \alpha_2 \vee \neg\beta), \quad (4)$$

$$\vee_{j=1}^{4} \alpha_j \Leftrightarrow (\alpha_1 \vee \alpha_2 \vee \beta_1) \wedge (\neg\beta_1 \vee \alpha_3 \vee \alpha_4) \quad (5)$$

where $l > 4$. Using (1) we can obtain an explicit transformation ξ in ξ_1 such that $\xi \Leftrightarrow \xi_1$ and ξ_1 is a CNF. Clearly, ξ_1 give us an explicit reduction from LCPS to SAT. Similarly, using relations (2) – (5) we can obtain an explicit transformation ξ_1 in ξ_2 such that $\xi_1 \Leftrightarrow \xi_2$ and ξ_2 is a 3-CNF. Clearly, ξ_2 give us an explicit reduction from LCPS to 3SAT.

4 Conclusion

In the previous section we have obtained explicit reductions from LCPS to some variants of satisfiability, SAT and 3SAT. There is a well known site on which solvers for SAT are posted [11]. We have created a generator of natural instances for LCPS. Also we used test problems from [11]. We have

designed our own genetic algorithm for SAT which is based on algorithms from [11]. We used heterogeneous cluster based on three clusters (Cluster USU, umt, um64) [12]. Each test was run on a cluster of at least 100 nodes. The maximum solution time was 11 hours. The average time to find a solution was 8.3 minutes. The best time was 12 seconds.

References

[1] B. S. Baker, Parameterized Pattern Matching: Algorithms and Applications, *Journal of Computer and System Sciences*, 52 (1996), 28-42.

[2] A. Amir, M. Farach, and S. Muthukrishnan, Alphabet Dependence in Parameterized Matching, *Information Processing Letters*, 49 (1994), 111-115.

[3] B. S. Baker, Parameterized diff, *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, (1999), 854-855.

[4] C. Hazay, M. Lewenstein, and D. Sokol, Approximate parameterized matching, *ACM Transactions on Algorithms*, 3 (2007), 29-43.

[5] O. Keller, T. Kopelowitz, and M. Lewenstein, On the Longest Common Parameterized Subsequence, *Theoretical Computer Science*, 410 (2009), 5347-5353.

[6] A. Gorbenko, M. Mornev, and V. Popov, Planning a Typical Working Day for Indoor Service Robots, *IAENG International Journal of Computer Science*, 38 (2011), 176-182.

[7] A. Gorbenko, M. Mornev, V. Popov, and A. Sheka, The problem of sensor placement for triangulation-based localisation, *International Journal of Automation and Control*, 5 (2011), 245-253.

[8] A. Gorbenko, V. Popov, and A. Sheka, Localization on Discrete Grid Graphs, *Proceedings of the CICA 2011*, (2012), 971-978.

[9] A. Gorbenko and V. Popov, On the Problem of Placement of Visual Landmarks, *Applied Mathematical Sciences*, 6 (2012), 689-696.

[10] A. Gorbenko and V. Popov, Programming for Modular Reconfigurable Robots, *Programming and Computer Software*, 38 (2012), 13-23.

[11] http://people.cs.ubc.ca/∼hoos/SATLIB/index-ubc.html

[12] http://parallel.imm.uran.ru/mvc_now/hardware/supercomp.htm

Received: January, 2012