Journal Subline

LNBI 3680

Transactions on Computational Systems Biology II

Corrado Priami
Editor-in-Chief

Springer

Corrado Priami Alexander Zelikovsky (Eds.)

Transactions on Computational Systems Biology II

Springer

Volume Editor

Alexander Zelikovsky
Georgia State University
Computer Science Department
33 Gilmer Street, Atlanta, GA, USA
E-mail: alexz@cs.gsu.edu

Library of Congress Control Number: 2005933892

CR Subject Classification (1998): J.3, H.2.8, F.1

ISSN 0302-9743
ISBN-10 3-540-29401-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-29401-6 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11567752 06/3142 5 4 3 2 1 0

Lecture Notes in Bioinformatics 3680

Edited by S. Istrail, P. Pevzner, and M. Waterman

Subseries of Lecture Notes in Computer Science

Preface

It gives me great pleasure to present the Special Issue of LNCS Transactions on Computational Systems Biology devoted to considerably extended versions of selected papers presented at the International Workshop on Bioinformatics Research and Applications (IWBRA 2005). The IWBRA workshop was a part of the International Conference on Computational Science (ICCS 2005) which took place in Emory University, Atlanta, Georgia, USA, May 22–24, 2005. See http://www.cs.gsu.edu/pan/iwbra.htm for more details.

The 10 papers selected for the special issue cover a wide range of bioinformatics research. The first papers are devoted to problems in RNA structure prediction: Blin et al. contribute to the arc-preserving subsequence problem and Liu et al. develop an efficient search of pseudoknots. The coding schemes and structural alphabets for protein structure prediction are discussed in the contributions of Lei and Dai, and Zheng and Liu, respectively. Song et al. propose a novel technique for efficient extraction of biomedical information. Nakhleh and Wang discuss introducing hybrid speciation and horizontal gene transfer in phylogenetic networks. Practical algorithms minimizing recombinations in pedigree phasing are proposed by Zhang et al. Kolli et al. propose a new parallel implementation in OpenMP for finding the edit distance between two signed gene permutations. The issue is concluded with two papers devoted to bioinformatics problems that arise in DNA microarrays: improved tag set design for universal tag arrays is suggested by Mandoiu et al. and a new method of gene selection is discussed by Xu and Zhang.

I am deeply thankful to the organizer and co-chair of IWBRA 2005 Prof. Yi Pan (Georgia State University). We were fortunate to have on the Program Committee the following distinguished group of researchers:

Piotr Berman, Penn State University, USA
Paola Bonizzoni, Università degli Studi di Milano-Bicocca, Italy
Liming Cai, University of Georgia, USA
Jake Yue Chen, Indiana University & Purdue University, USA
Bhaskar Dasgupta, University of Illinois at Chicago, USA
Juntao Guo, University of Georgia, USA
Tony Hu, Drexel University, USA
Bin Ma, University of West Ontario, Canada
Ion Mandoiu, University of Connecticut, USA
Kayvan Najarian, University of North Carolina at Charlotte, USA
Giri Narasimhan, Florida International University, USA
Jun Ni, University of Iowa, USA
Mathew Palakal, Indiana University & Purdue University, USA
Pavel Pevzner, University of California at San Diego, USA

Gwenn Volkert, Kent State University, USA
Kaizhong Zhang, University of West Ontario, Canada
Wei-Mou Zheng, Chinese Academy of Sciences, China

June 2005 Alexander Zelikovsky

LNCS Transactions on Computational Systems Biology – Editorial Board

Lecture Notes in Bioinformatics

Vol. 3695: M.R. Berthold, R. Glen, K. Diederichs, O. Kohlbacher, I. Fischer (Eds.), Computational Life Sciences. XI, 277 pages. 2005.

Vol. 3692: R. Casadio, G. Myers (Eds.), Algorithms in Bioinformatics. X, 436 pages. 2005.

Vol. 3680: C. Priami, A. Zelikovsky (Eds.), Transactions on Computational Systems Biology II. IX, 153 pages. 2005.

Vol. 3678: A. McLysaght, D.H. Huson (Eds.), Comparative Genomics. VIII, 167 pages. 2005.

Vol. 3615: B. Ludäscher, L. Raschid (Eds.), Data Integration in the Life Sciences. XII, 344 pages. 2005.

Vol. 3594: J.C. Setubal, S. Verjovski-Almeida (Eds.), Advances in Bioinformatics and Computational Biology. XIV, 258 pages. 2005.

Vol. 3500: S. Miyano, J. Mesirov, S. Kasif, S. Istrail, P. Pevzner, M. Waterman (Eds.), Research in Computational Molecular Biology. XVII, 632 pages. 2005.

Vol. 3388: J. Lagergren (Ed.), Comparative Genomics. VII, 133 pages. 2005.

Vol. 3380: C. Priami (Ed.), Transactions on Computational Systems Biology I. IX, 111 pages. 2005.

Vol. 3370: A. Konagaya, K. Satou (Eds.), Grid Computing in Life Science. X, 188 pages. 2005.

Vol. 3318: E. Eskin, C. Workman (Eds.), Regulatory Genomics. VII, 115 pages. 2005.

Vol. 3240: I. Jonassen, J. Kim (Eds.), Algorithms in Bioinformatics. IX, 476 pages. 2004.

Vol. 3082: V. Danos, V. Schachter (Eds.), Computational Methods in Systems Biology. IX, 280 pages. 2005.

Vol. 2994: E. Rahm (Ed.), Data Integration in the Life Sciences. X, 221 pages. 2004.

Vol. 2983: S. Istrail, M.S. Waterman, A. Clark (Eds.), Computational Methods for SNPs and Haplotype Inference. IX, 153 pages. 2004.

Vol. 2812: G. Benson, R.D. M. Page (Eds.), Algorithms in Bioinformatics. X, 528 pages. 2003.

Vol. 2666: C. Guerra, S. Istrail (Eds.), Mathematical Methods for Protein Structure Analysis and Design. XI, 157 pages. 2003.

Table of Contents

What Makes the ARC-PRESERVING SUBSEQUENCE Problem Hard?*

Guillaume Blin[1], Guillaume Fertin[1], Romeo Rizzi[2], and Stéphane Vialette[3]

[1] LINA - FRE CNRS 2729 Université de Nantes,
2 rue de la Houssinière BP 92208 44322 Nantes Cedex 3 - France
{blin, fertin}@univ-nantes.fr

[2] Universit degli Studi di Trento Facolt di Scienze - Dipartimento di Informatica e Telecomunicazioni Via Sommarive, 14 - I38050 Povo - Trento (TN) - Italy
Romeo.Rizzi@unitn.it

[3] LRI - UMR CNRS 8623 Faculté des Sciences d'Orsay, Université Paris-Sud
Bât 490, 91405 Orsay Cedex - France
vialette@lri.fr

Abstract. In molecular biology, RNA structure comparison and motif search are of great interest for solving major problems such as phylogeny reconstruction, prediction of molecule folding and identification of common functions. RNA structures can be represented by arc-annotated sequences (primary sequence along with arc annotations), and this paper mainly focuses on the so-called *arc-preserving subsequence* (APS) problem where, given two arc-annotated sequences (S, P) and (T, Q), we are asking whether (T, Q) can be obtained from (S, P) by deleting some of its bases (together with their incident arcs, if any). In previous studies, this problem has been naturally divided into subproblems reflecting the intrinsic complexity of the arc structures. We show that APS(CROSSING, PLAIN) is **NP**-complete, thereby answering an open problem posed in [11]. Furthermore, to get more insight into where the actual border between the polynomial and the **NP**-complete cases lies, we refine the classical subproblems of the APS problem in much the same way as in [19] and prove that both APS($\{\sqsubset, \emptyset\}, \emptyset$) and APS($\{<, \emptyset\}, \emptyset$) are **NP**-complete. We end this paper by giving some new positive results, namely showing that APS($\{\emptyset\}$, $\emptyset$) and APS($\{\emptyset\}$,$\{\emptyset\}$) are polynomial time.

Keywords: RNA structures, Arc-Preserving Subsequence problem, Computational complexity.

1 Introduction

At a molecular state, the understanding of biological mechanisms is subordinated to the discovery and the study of RNA functions. Indeed, it is established that the

* This work was partially supported by the French-Italian PAI Galileo project number 08484VH and by the CNRS project ACI Masse de Données "NavGraphe". A preliminary version of this paper appeared in the Proc. of IWBRA'05, Springer, V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3515, pp. 860-868, 2005.

C. Priami, A. Zelikovsky (Eds.): Trans. on Comput. Syst. Biol. II, LNBI 3680, pp. 1–36, 2005.

conformation of a single-stranded RNA molecule (a linear sequence composed of ribonucleotides A, U, C and G, also called primary structure) partly determines the function of the molecule. This conformation results from the folding process due to local pairings between complementary bases ($A-U$ and $C-G$, connected by a hydrogen bond). The secondary structure of an RNA (a simplification of the complex 3-dimensional folding of the sequence) is the collection of folding patterns (stem, hairpin loop, bulge loop, internal loop, branch loop and pseudo-knot) that occur in it.

RNA secondary structure comparison is important in many contexts, such as:

- identification of highly conserved structures during evolution, non detectable in the primary sequence which is often slightly preserved. These structures suggest a significant common function for the studied RNA molecules [16,18,13,8],
- RNA classification of various species (phylogeny)[4,3,21],
- RNA folding prediction by considering a set of already known secondary structures [24,14],
- identification of a consensus structure and consequently of a common role for molecules [22,5].

Structure comparison for RNA has thus become a central computational problem bearing many challenging computer science questions. At a theoretical level, the RNA structure is often modeled as an *arc-annotated sequence*, that is a pair (S, P) where S is the sequence of ribonucleotides and P represents the hydrogen bonds between pairs of elements of S. Different pattern matching and motif search problems have been investigated in the context of arc-annotated sequences among which we can mention the *arc-preserving subsequence* (APS) problem, the EDIT DISTANCE problem, the *arc-substructure* (AST) problem and the *longest arc-preserving subsequence* (LAPCS) problem (see for instance [6,15,12,11,2]). For other related studies concerning algorithmic aspects of (protein) structure comparison using *contact maps*, refer to [10,17].

In this paper, we focus on the *arc-preserving subsequence* (APS) problem: given two arc-annotated sequences (S, P) and (T, Q), this problem asks whether (T, Q) can be exactly obtained from (S, P) by deleting some of its bases together with their incident arcs, if any. This problem is commonly encountered when one is searching for a given RNA pattern in an RNA database [12]. Moreover, from a theoretical point of view, the APS problem can be seen as a restricted version of the LAPCS problem, and hence has applications in the structural comparison of RNA and protein sequences [6,10,23]. The APS problem has been extensively studied in the past few years [11,12,6]. Of course, different restrictions on arc-annotation alter the computational complexity of the APS problem, and hence this problem has been naturally divided into subproblems reflecting the complexity of the arc structure of both (S, P) and (T, Q): PLAIN, CHAIN, NESTED, CROSSING or UNLIMITED (see Section 2 for details). All of them but one have been classified as to whether they are polynomial time solvable or **NP**-complete. The problem of the existence of a polynomial time algorithm for the APS(CROSSING,PLAIN) problem was mentioned in [11] as the last open problem

Table 1. APS problem complexity where $n = |S|$ and $m = |T|$. $\star$ result from this paper.

APS				
	CROSSING	NESTED	CHAIN	PLAIN
CROSSING	**NP**-complete [6]	**NP**-complete [12]		**NP**-complete $\star$
NESTED		$O(nm)$ [11]		
CHAIN			$O(nm)$ [11]	$O(n+m)$ [11]

in the context of arc-preserving subsequences (cf. Table 1). Unfortunately, as we shall prove in Section 4, the APS(CROSSING,PLAIN) problem is **NP**-complete even for restricted special cases.

In analyzing the computational complexity of a problem, we are often trying to define the precise boundary between the polynomial and the **NP**-complete cases. Therefore, as another step towards establishing the precise complexity landscape of the APS problem, it is of great interest to subdivide the existing cases into more precise ones, that is to refine the classical complexity levels of the APS problem, for determining more precisely what makes the problem hard. For that purpose, we use the framework introduced by Vialette [19] in the context of 2-intervals (a simple abstract structure for modelling RNA secondary structures). As a consequence, the number of complexity levels rises from 4 (not taking into account the UNLIMITED case) to 8, and all the entries of this new complexity table need to be filled. Previous known results concerning the APS problem, along with two **NP**-completeness and two polynomiality proofs, allow us to fill all the entries of this new table, therefore determining what exactly makes the APS problem hard.

The paper is organized as follows. In Section 2, we give notations and definitions concerning the APS problem. In Section 3 we introduce and explain the new refinements of the complexity levels we are going to study. In Section 4, we show that the APS($\{\sqsubset, \emptyset\}, \emptyset$) problem is **NP**-complete thereby proving that the (classical) APS(CROSSING, PLAIN) problem is **NP**-complete as well. As another refinement to that result, we prove that the APS($\{<, \emptyset\}, \emptyset$) problem is **NP**-complete. Finally, in Section 5, we give new polynomial time solvable algorithms for restricted instances of the APS(CROSSING, PLAIN) problem.

2 Preliminaries

An RNA structure is commonly represented as an arc-annotated sequence (S, P) where S is the sequence of ribonucleotides (or bases) and P is the set of arcs connecting pairs of bases in S. Let (S, P) and (T, Q) be two arc-annotated sequences such that $|S| \geq |T|$ (in the following, $n = |S|$ and $m = |T|$). The APS problem asks whether (T, Q) can be exactly obtained from (S, P) by deleting some of its bases together with their incident arcs, if any.

Since the general problem is easily seen to be intractable [6], the arc structure must be restricted. Evans [6] proposed four possible restrictions on P (resp. Q) which were largely reused in the subsequent literature:

1. there is no base incident to more than one arc,
2. there are no arcs crossing,
3. there is no arc contained in another,
4. there is no arc.

These restrictions are used progressively and inclusively to produce five different levels of allowed arc structure:

- UNLIMITED - the general problem with no restrictions
- CROSSING - restriction 1
- NESTED - restrictions 1 and 2
- CHAIN - restrictions 1, 2 and 3
- PLAIN - restriction 4

Guo proved in [12] that the APS(CROSSING, CHAIN) problem is **NP**-complete. Guo et al. observed in [11] that the **NP**-completeness of the APS(CROSSING, CROSSING) and APS(UNLIMITED, PLAIN) easily follows from results of Evans [6] concerning the LAPCS problem. Furthermore, they gave a $O(nm)$ time for the APS(NESTED, NESTED) problem. This algorithm can be applied to easier problems such as APS(NESTED, CHAIN), APS(NESTED, PLAIN), APS(CHAIN, CHAIN) and APS(CHAIN,PLAIN). Finally, Guo et al. mentioned in [11] that APS(CHAIN, PLAIN) can be solved in $O(n + m)$ time. Until now, the question of the existence of an exact polynomial algorithm for the problem APS(CROSSING, PLAIN) remained open. We will first show in the present paper that the problem APS(CROSSING,PLAIN) is **NP**-complete. Table 1 surveys known and new results for various types of APS. Observe that the UNLIMITED level has no restrictions, and hence is of limited interest in our study. Consequently, from now on we will not be concerned anymore with that level.

3 Refinement of the APS Problem

In this section, we propose a refinement of the APS problem. We first state formally our approach and explain why such a refinement is relevant for both theoretical and experimental studies. We end the section by giving easy properties of the proposed refinement that will prove extremely useful in Section 5.

3.1 Splitting the Levels

As we will show in Section 4, the APS(CROSSING, PLAIN) problem is **NP**-complete. That result answers the last open problem concerning the computational complexity of the APS problem with respect to classical complexity levels, *i.e.*, PLAIN, CHAIN, NESTED and CROSSING (cf. Table 1). However, we are mainly interested in the elaboration of the precise border between **NP**-complete

and polynomially solvable cases. Indeed, both theorists and practitioners might naturally ask for more information concerning the hard cases of the APS problem in order to get valuable insight into what makes the problem difficult.

As a next step towards a better understanding of what makes the APS problem hard, we propose to refine the models which are classically used for classifying arc-annotated sequences. Our refinement consists in splitting those models of arc-annotated sequences into more precise relations between arcs. For example, such a refinement provides a general framework for investigating polynomial time solvable and hard restricted instances of APS(CROSSING, PLAIN), thereby refining in many ways Theorem 1 (see Section 5).

We use the three relations first introduced by Vialette [19,20] in the context of 2-*intervals* (a simple abstract structure for modelling RNA secondary structures). Actually, his definition of 2-intervals could almost apply in this paper (the main difference lies in the fact that Vialette used 2-intervals for representing sets of contiguous arcs). Vialette defined three possible relations between 2-intervals that can be used for arc-annotated sequences as well. They are the following: for any two arcs $p_1 = (i, j)$ and $p_2 = (k, l)$ in P, we will write $p_1 < p_2$ if $i < j < k < l$ (*precedence* relation), $p_1 \sqsubset p_2$ if $k < i < j < l$ (*nested* relation) and $p_1 \between p_2$ if $i < k < j < l$ (*crossing* relation). Two arcs p_1 and p_2 are τ-comparable for some $\tau \in \{<, \sqsubset, \between\}$ if $p_1 \tau p_2$ or $p_2 \tau p_1$. Let $\mathcal{P}$ be a set of arcs and R be a non-empty subset of $\{<, \sqsubset, \between\}$. The set $\mathcal{P}$ is said to be *R-comparable* if any two distinct arcs of $\mathcal{P}$ are τ-comparable for some $\tau \in R$. An arc-annotated sequence (S, P) is said to be an R-arc-annotated sequence for some non-empty subset R of $\{<, \sqsubset, \between\}$ if P is R-comparable. We will write $R = \emptyset$ in case $P = \emptyset$. Observe that our model cannot deal with arc-annotated sequences which contain only one arc. However, having only one arc or none can not really affect the computational complexity of the problem. Just one guess reduces from one case to the other. Details are omitted here.

As a straightforward illustration of the above definitions, classical complexity levels for the APS problem can be expressed in terms of combinations of our new relations: PLAIN is fully described by $R = \emptyset$, CHAIN is fully described by $R = \{<\}$, NESTED is fully described by $R = \{<, \sqsubset\}$ and CROSSING is fully described by $R = \{<, \sqsubset, \between\}$. The key point is to observe that our refinement allows us to consider new structures for arc-annotated sequences, namely $R = \{\sqsubset\}$, $R = \{\between\}$, $R = \{<, \between\}$ and $R = \{\sqsubset, \between\}$, which could not be considered using the classical complexity levels. Although other refinements may be possible (in particular well-suited for parameterized complexity analysis), we do believe that such an approach allows a more precise analysis of the complexity of the APS problem.

Of course one might object that some of these subdivisions are unlikely to appear in RNA secondary structures. While this is true, it is also true that it is of great interest to answer, at least partly, the following question: Where is the precise boundary between the polynomial and the **NP**-complete cases? Indeed, such a question is relevant for both theoretical and experimental studies.

For one, many important optimization problems are known to be **NP**-complete. That is, unless $\mathbf{P} = \mathbf{NP}$, there is no polynomial time algorithm that optimally solves these on every input instance, and hence proving a problem to be **NP**-complete is generally accepted as a proof of its difficulty. However the problem to be solved may be much more specialized than the general one that was proved to be **NP**-complete. Therefore, during the past three decades, many studies have been devoted to proving **NP**-completeness results for highly restricted instances in order to precisely define the border between tractable and intractable problems. Our refinements have thus to be seen as another step towards establishing the precise complexity landscape of the APS problem.

For another, it is worthwhile keeping in mind that intractability must be coped with and problems must be solved in practical applications. Computer science theory has articulated a few general programs for systematically coping with the ubiquitous phenomena of computational intractability: average case analysis, approximation algorithm, randomized algorithm and fixed parameter complexity. Fully understanding where the boundary lies between efficiently solvable formulations and intractable ones is another important approach. Indeed, from an engineering point of view for which the emphasis is on efficiency, that precise boundary might be a good starting point for designing efficient heuristics or for exploring fixed-parameter tractability. The better our understanding of the problem, the better our ability in defining efficient algorithms for practical applications.

3.2 Immediate Results

First, observe that, as in Table 1, we only have to consider cases of APS(R_1,R_2) where R_1 and R_2 are compatible, i.e. $R_2 \subseteq R_1$. Indeed, if this is not the case, we can immediately answer negatively since there exists two arcs in T which satisfy a relation in R_2 which is not in R_1, and hence T simply cannot be obtained from S by deleting bases of S. Those incompatible cases are simply denoted by hatched areas in Table 2.

Table 2. Complexity results after refinement of the complexity levels. ////: incompatible cases. ?: open problems.

APS								
R_1 \ R_2	$\{<,\sqsubset,\emptyset\}$	$\{\sqsubset,\emptyset\}$	$\{<,\emptyset\}$	$\{\emptyset\}$	$\{<,\sqsubset\}$	$\{\sqsubset\}$	$\{<\}$	$\emptyset$
$\{<,\sqsubset,\emptyset\}$	NP-C [6]	?	NP-C [12]	?	NP-C [12]	?	NP-C [12]	?
$\{\sqsubset,\emptyset\}$		?	////	?	////	?	////	?
$\{<,\emptyset\}$			?	?	////	////	?	?
$\{\emptyset\}$				?	////	////	////	?
$\{<,\sqsubset\}$					$O(nm)$ [11]	$O(nm)$ [11]	$O(nm)$ [11]	$O(nm)$ [11]
$\{\sqsubset\}$						$O(nm)$ [11]	////	$O(nm)$ [11]
$\{<\}$							$O(nm)$ [11]	$O(n+m)$ [11]
$\emptyset$								$O(n+m)$ [11]

Some known results allow us to fill many entries of the new complexity table derived from our refinement. The remainder of this subsection is devoted to detailing these first easy statements. We begin with an observation concerning complexity propagation properties of the APS problems in our refined model.

Observation 1. *Let R_1, R_2, R'_1 and R'_2 be four subsets of $\{<, \sqsubset, \wr\}$ such that $R'_2 \subseteq R_2 \subseteq R_1$ and $R'_2 \subseteq R'_1 \subseteq R_1$. If* APS($R'_1$, R'_2) *is* **NP***-complete (resp.* APS(R_1, R_2) *is polynomial time solvable) then so is* APS(R_1, R_2) *(resp.* APS(R'_1, R'_2)*).*

On the positive side, Gramm *et al.* have shown that APS(NESTED, NESTED) is solvable in $O(nm)$ time [11]. Another way of stating this is to say that APS($\{<, \sqsubset\}$, $\{<, \sqsubset\}$) is solvable in $O(mn)$ time. That result together with Observation 1 may be summarized by saying that APS(R_1, R_2) for any compatible R_1 and R_2 such that $\wr \notin R_1$ and $\wr \notin R_2$ is polynomial time solvable.

Conversely, the **NP**-completeness of APS(CROSSING,CROSSING) has been proved by Evans [6]. A simple reading shows that her proof is concerned with $\{<, \sqsubset, \wr\}$-arc-annotated sequences, and hence she actually proved that APS($\{<, \sqsubset, \wr\}$, $\{<, \sqsubset, \wr\}$) is **NP**-complete. Similarly, in proving that APS(CROSSING, CHAIN) is **NP**-complete [12], Guo actually proved that APS($\{<, \sqsubset, \wr\}$, $\{<\}$) is **NP**-complete. Note that according to Observation 1, this latter result implies that APS($\{<, \sqsubset, \wr\}$, $\{<, \sqsubset\}$) and APS($\{<, \sqsubset, \wr\}$, $\{<, \wr\}$) are **NP**-complete.

Table 2 surveys known and new results for various types of our refined APS problem. Observe that this paper answers all questions concerning the APS problem with respect to the new complexity levels.

4 Hardness Results

We show in this section that APS($\{\sqsubset, \wr\}, \emptyset$) is **NP**-complete thereby proving that the (classical) APS(CROSSING, PLAIN) problem is **NP**-complete. That result answers an open problem posed in [11], which was also the last open problem concerning the computational complexity of the APS problem with respect to classical complexity levels, *i.e.*, PLAIN, CHAIN, NESTED and CROSSING (cf. Table 1). Furthermore, we prove that the APS($\{<, \wr\}, \emptyset$) is **NP**-complete as well.

We provide a polynomial time reduction from the 3-SAT problem: Given a set $\mathcal{V}_n$ of n variables and a set $\mathcal{C}_q$ of q clauses (each composed of three literals) over $\mathcal{V}_n$, the problem asks to find a truth assignment for $\mathcal{V}_n$ that satisfies all clauses of $\mathcal{C}_q$. It is well-known that the 3-SAT problem is **NP**-complete [9].

It is easily seen that the APS($\{\sqsubset, \wr\}, \emptyset$) problem is in **NP**. The remainder of the section is devoted to proving that it is also **NP**-hard. Let $\mathcal{V}_n = \{x_1, x_2, ...x_n\}$ be a finite set of n variables and $\mathcal{C}_q = \{c_1, c_2, \ldots, c_q\}$ a collection of q clauses. Observe that there is no loss of generality in assuming that, in each clause, the literals are ordered from left to right, *i.e.*, if $c_i = (x_j \vee x_k \vee x_l)$ then $j < k < l$. Let us first detail the construction of the sequences S and T:

$$S = S^s_{x_1} A\ S^s_{\overline{x_1}}\ S^s_{x_2} A\ S^s_{\overline{x_2}} \dots S^s_{x_n} A\ S^s_{\overline{x_n}}\ S_{c_1}\ S_{c_2} \dots S_{c_q}\ S^e_{x_1}\ S^e_{x_2} \dots S^e_{x_n}$$
$$T = T^s_{x_1}\ T^s_{x_2} \dots T^s_{x_n}\ T_{c_1}\ T_{c_2} \dots T_{c_q}\ T^e_{x_1}\ T^e_{x_2} \dots T^e_{x_n}$$

We now detail the subsequences that compose S and T. Let γ_m (resp. $\gamma_{\overline{m}}$) be the number of occurrences of literal x_m (resp. $\overline{x_m}$) in $\mathcal{C}_q$ and let $k_m = \max(\gamma_m, \gamma_{\overline{m}})$. For each variable $x_m \in \mathcal{V}_n$, $1 \leq m \leq n$, we construct words $S^s_{x_m} = AC^{k_m}$, $S^s_{\overline{x_m}} = C^{k_m}A$ and $T^s_{x_m} = AC^{k_m}A$ where C^{k_m} represents a word of k_m consecutive bases C. For each clause c_i of $\mathcal{C}_q$, $1 \leq i \leq q$, we construct words $S_{c_i} = UGGGA$ and $T_{c_i} = UGA$. Finally, for each variable $x_m \in \mathcal{V}_n$, $1 \leq m \leq n$, we construct words $S^e_{x_m} = UUA$ and $T^e_{x_m} = UA$.

Having disposed of the two sequences, we now turn to defining the corresponding two arc structures (see Figure 1). In the following, $\mathcal{S}eq[i]$ will denote the i^{th} base of a sequence $\mathcal{S}eq$ and, for any $1 \leq m \leq n$, $l_{\overline{m}} = |S^s_{\overline{x_m}}|$. For all $1 \leq m \leq n$, we create the two following arcs: $(S^s_{x_m}[1], S^e_{x_m}[1])$ and $(S^s_{\overline{x_m}}[l_{\overline{m}}], S^e_{x_m}[2])$. For each clause c_i of $\mathcal{C}_q$, $1 \leq i \leq q$, and for each $1 \leq m \leq n$, if the k^{th} (*i.e.* 1^{st}, 2^{nd} or 3^{rd}) literal of c_i is x_m (resp. $\overline{x_m}$) then we create an arc between any free (i.e. not already incident to an arc) base C of $S^s_{\overline{x_m}}$ (resp. $S^s_{x_m}$) and the k^{th} base G of S_{c_i} (note that this is possible by definition of $S^s_{\overline{x_m}}$, $S^s_{x_m}$ and S_{c_i}). On the whole, the instance we have constructed is composed of $3q + 2n$ arcs. We denote by APS-CP-construction any construction of this type. In the following, we will distinguish arcs between bases A and U, denoted by AU-arcs, from arcs between bases C and G, denoted by CG-arcs. An illustration of an APS-CP-construction is given in Figure 1. Clearly, our construction can be carried out in polynomial time. Moreover, the result of such a construction is indeed an instance of $\mathrm{APS}(\{\sqsubset, \wr\}, \emptyset)$, since $Q = \emptyset$ (no arc is added to T) and P is a $\{\sqsubset, \wr\}$-comparable set (since there are no arcs $\{<\}$-comparable.

We begin by proving a canonicity lemma of an APS-CP-construction.

Lemma 1. *Let (S, P) and (T, Q) be any two arc-annotated sequences obtained from an* APS-CP*-construction. If (T, Q) can be obtained from (S, P) by deleting*

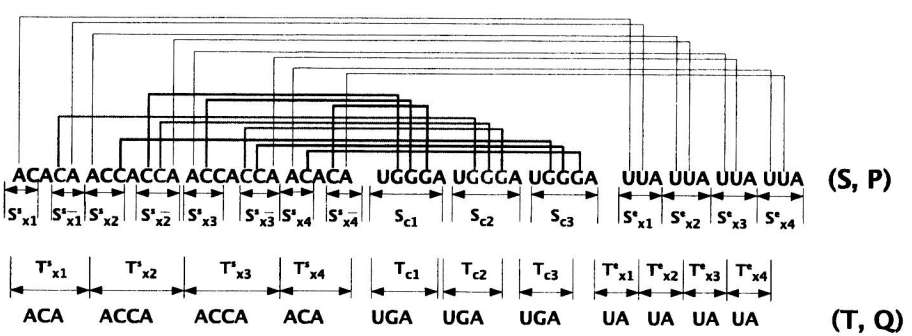

Fig. 1. Example of an APS-CP-construction with $\mathcal{C}_q = (x_2 \vee \overline{x_3} \vee x_4) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_2} \vee x_3 \vee \overline{x_4})$