Forbidden pairs combinatorics

B. Litow

Department of Computer Science, James Cook University
Townsville, Qld. 4811, Australia
bruce@cs.jcu.edu.au

Narsingh Deo

School of Computer Science, University of Central Florida
Orlando FL, 32816, USA
deo@cs.ucf.edu

Abstract

We define a formalism, forbidden pairs problems, in which many combinatorial constructions can be expressed. The formalism highlights the basic nature of a large number of combinatorial constraints. We also define an algebraic-arithmetic problem to which all forbidden pairs problems can be PTIME reduced.

1 Introduction

We introduce a combinatorial problem which can be specialised to particular problems in many different ways. We will call this the *forbidden pairs* problem. In particular, we represent packing problems and Hamiltonian circuit as forbidden pairs problems. We then show that every forbidden pairs problem can in turn be formulated in terms of a simple, NP-hard algebraic-arithmetic type of problem that we designate as $\mathcal{F}$.

Let w be either a monomial over commuting indeterminates $x_1, \ldots, x_n$ or a string over $a_1, \ldots, a_n$. The Parikh vector $\vec{w}$ of w is defined to be the tuple $(k_1, \ldots, k_n)$ of nonnegative integers, such that k_i is the number of occurrences of x_i (a_i) in w. Note that $k_1 + \cdots + k_n$ is the total degree in case w is a monomial, and the length in case w is a string. We will refer to this sum as the total degree of the monomial-string pair.

A forbidden pairs problem is specified by choosing a set of commuting indeterminates $X = \{x_1, \ldots x_n\}$, a set of noncommuting indeterminates $A = \{a_1, \ldots, a_n\}$, a set B of monomials over X having total degree 2, and a set $C \subset AA$. The sets B

and C are called forbidden pair sets. A monomial y over X is said to be forbidden if it can be written as $y = u \cdot z$, where $u \in B$, otherwise it is said to be admissible. Likewise, a string $w \in A^*$ is forbidden if it can be written $w = uvz$ where $v \in C$, otherwise it is admissible. A monomial-string pair y, w is admissible iff both of the following two conditions are satisfied.

- y and w are both admissible.
- $\vec{y} = \vec{w}$.

The forbidden pairs problem, designated by (X, B, A, C) is to determine the maximum total degree over all admissible monomial-string pairs.

Many classical combinatorial configurations fall under the special subtype of forbidden pairs in which the noncommutative constraint C is empty. We give two familiar examples here. The first, packings, assumes a quite simple form in terms of forbidden pairs.

Combinatorial designs are a classical part of combinatorics. An account of fundamental ideas can be found in [3]. A packing design (called simply a packing) involves three positive integers p, q, r. A p, q, r-packing is a collection $\mathcal{X}$ of q-element subsets of a p-element set Y, such that if x and y are any two distinct elements of $\mathcal{X}$, $x \cap y$ contains fewer than r elements. Define $\#(p, q, r)$ to be the maximum cardinality of any p, q, r-packing. It is easy to show that

$$\#(p,q,r) \leq \frac{\binom{p}{r}}{\binom{q}{r}} .$$

First, no r-element subset of Y can be contained in two distinct elements of a p, q, r-packing. This implies

$$\#(p,q,r) \leq \binom{p}{r} .$$

Second, there is a multiplicity of $\binom{q}{r}$ residing in the quantity

$$\binom{p}{r} ,$$

and the desired upper bound on $\#(p, q, r)$ follows by dividing out this multiplicity. Rödl has shown that

$$\lim_{p \to \infty} \#(p,q,r) \cdot \frac{\binom{q}{r}}{\binom{p}{r}} = 1$$

for every fixed r and q such that $r < q$. See [6].

We proceed to express $\#(p, q, r)$ in terms of a forbidden pairs problem. Let $n = \binom{p}{q}$ and let $X = \{x_1, \ldots, x_n\}$. Let x_i correspond to the i-th element of some listing of all the q-element subsets of the p-element set Y. The set B consists of all $x_i \cdot x_j$ such that the corresponding q-element subsets of Y have an intersection whose cardinality is at least r. We let $C = \emptyset$. It should be clear that an admissible monomial represents a p, q, r-packing so that the maximum total degree of any admissible monomial is $\#(p, q, r)$. We regard p as the size of the problem of computing $\#(p, q, r)$ so that for q regarded as fixed we clearly have a PTIME reduction to an instance of $\mathcal{F}$. If q is regarded as a parameter, then $\binom{p}{q}$ cannot be bounded above by $p^{O(1)}$.

There is a direct connection between the forbidden pairs formulation of the packing problem and the maximum clique problem in a graph. A clique is a subgraph that is isomorphic to a complete graph. Determining the largest order of any clique in a given graph is the maximum clique problem, which is known to be NP-complete. See [2]. Let $(X, B, A, \emptyset)$ be the forbidden pairs problem to which a p, q, r-packing problem has been reduced. The undirected graph G has X as its set of vertices and edge set $X \times X - \{(x_i, x_j) \mid x_i \cdot x_j \in B\}$. It is evident that the maximum order of any clique is exactly $\#(p, q, r)$. If q is fixed, the order of G is polynomial in p. Now, for certain values of p, e.g., powers of primes belonging to various number theoretic families, PTIME constructions of packings of size $\#(p, q, r)$ are known. See [3]. It follows that for the corresponding graphs G, maximum clique can be solved in PTIME.

Ramsey problems also belong to the $C = \emptyset$ type of forbidden pairs problem. We illustrate this with the simplest of Ramsey problems, the determination of the Ramsey number $p(q, r)$. Recall that p is the least integer such for any partition into two pieces of the set S of all q-element subsets of a p-element set T, there exists an r-element subset of T all of whose q-element subsets are in one of the pieces.

We proceed to formulate this Ramsey problem in terms of forbidden pairs. Let R be the set of all r-element subsets of T. Define X to be the set of all $x_{i,j,k} = (y_i, j, z_k)$, where $y_1, \ldots, y_f$ is a list of all of the elements of S, $1 \leq j \leq g$, where g is the number of bipartite partitions of S, and $z_1, \ldots, z_h$ is a list of all of the elements of R, such that $y_i \subseteq z_k$. We regard the elements of X as mutually commuting. Define B to be the set of degree-two monomials consisting of all $x_{i,j,k} \cdot x_{i',j,k'}$ such that $k \neq k'$, all $x_{i,j,k} \cdot x_{i,j,k}$, and all $x_{i,j,k} \cdot x_{i',j,k}$ such that y_i and $y_{i'}$ are in different pieces of the j-th partition. An admissible monomial of highest total degree will have for each partition index j a single element z_k of R and a list of elements of S constituting a partition of all of the q-element subsets of z_k. This total degree will be exactly

$$\binom{r}{q} \cdot \binom{\binom{p}{q}}{2} .$$

In this way, for given data p, q, r, we can use the forbidden pairs problem $(X, B, A, \emptyset)$ to check whether $p \geq p(q, r)$. Note that A plays no part here.

We point out the role played by the string component of a monomial-string pair. The string encodes a regular language constraint. In fact, the edge set of any digraph can be faithfully represented as the set C of ordered pairs of vertices which are not edges. Any path in the graph is a string over the vertices which is admissible w.r.t. C. In fact, one can use this elementary observation to establish an homomorphic characterisation of regular languages. See [7]. As we will see with Hamiltonian circuit, the constraint imposed by the string component of a monomial-string pair can be automatically enforced without explicit use of monomial-string pairs.

An instance of the algebraic-arithmetic problem $\mathcal{F}$ is given in terms of arithmetic expressions. An arithmetic expression E is either a symbol in $\{0, 1, x_1, \ldots, x_n\}$, or can be written as $(F + G)$ or $F \cdot G$, where F and G are arithmetic expressions. All of the symbols commute and the rules of ordinary arithmetic apply to sums $(F + G)$ and products $F \cdot G$. The full sum-of-products expansion of E is designated by $\tilde{E}$. Let D_i^k designate formal k-fold differentiation w.r.t. x_i, followed by setting x_i to zero. Given an expression E and nonnegative integers $k_1, \ldots, k_n$, the question is whether $D_1^{k_1} \cdots D_n^{k_n} E \neq 0$. This is tantamount to asking whether the monomial $x_1^{k_1} \cdots x_n^{k_n}$ occurs with a nonzero coefficient in $\tilde{E}$.

2 Hamiltonian circuit in terms of forbidden pairs

We will work with the digraph version of Hamiltonian circuit for convenience. A simple spanning cycle in a graph is called a Hamiltonian circuit. Determining whether a graph has a Hamiltonian circuit is known to be NP-complete. See [2]. In this section, G will be a graph with vertex set $V = \{v_1, \ldots, v_n\}$ and edge set E. First, we exhibit a $2^{O(n)}$-time algorithm for Hamiltonian circuit. A more sophisticated analysis of the search space leads to a subexponential time algorithm (essentially $2^{n^{7/10}}$-time), which is presented in [5].

The naive algorithm for Hamiltonian circuit involves enumeration of all $n!$ permutations on V and checking whether each permutation is a cycle in G. This is a wasteful procedure because the search space can be reduced in size to 2^n.

We define a rightlinear grammar $\mathcal{G}$ as follows. Its terminal set is V, its nonterminal set is $\{y_1, \ldots, y_n\}$, and y_1 is the initial symbol. If $(v_i, v_j) \in E$, $y_i \to v_j y_j$ is a production, and for each (v_i, v_1) there is a production $y_i \to v_1$. These are the only productions. It is not difficult to see that $\mathcal{G}$ generates exactly the set of cycles in G which contain v_1 just once. Using the standard product construction of automaton theory it is straightforward to modify $\mathcal{G}$ to a grammar $\mathcal{G}_n$ which generates all cycles having length n and containing v_1 just once. This set is designated by $L(\mathcal{G}_n)$. The construction of $\mathcal{G}_n$ takes $(n \cdot |\mathcal{G}|)^{O(1)}$-time, where $|\mathcal{G}|$ is the size of $\mathcal{G}$.

An algorithm for Hamiltonian circuit can be obtained by designing a deterministic finite automaton which accepts those strings in $L(\mathcal{G}_n)$ which are simple. To do this the automaton needs states of the form $(s_1 \cdots s_n, q)$, where $s_i \in \{0,1\}$ and q is a state dictated by converting $\mathcal{G}_n$ into an automaton. If the first component of the state has $s_i = 0$ and v_i is read, s_i is set to 1. If $s_i = 1$ and v_i is read, the automaton quits in failure. It should be evident that an automaton with $(n \cdot |\mathcal{G}_n|)^n \cdot 2^n$ states can accept exactly those strings in V^* which represent simple, spanning cycles in G. Now, $|\mathcal{G}_n| = n^{O(1)}$ so the emptiness problem for this automaton can be solved in $(n^{O(1)} \cdot 2^n)^{O(1)} = 2^{O(n)}$-time.

Now we will formulate Hamiltonian circuit for the graph G as a forbidden pairs problem (X, B, V, C). We define $B = \{x_i \cdot x_i \mid i = 1, \ldots, n\}$, which is the smallest possible B. Note that B enforces the constraint that no vertex recur in a cycle. The set C, which represents the graph G is just $V \times V - E$. It is clear that G has a Hamiltonian circuit iff there exists a monomial-string pair whose total degree is n.

Theorem 1 *$\mathcal{F}$ is NP-hard, and is at least as hard as counting the number of Hamiltonian circuits.*

Proof : We use the grammar $\mathcal{G}_n$ described in the $2^{O(n)}$-time algorithm for Hamiltonian circuit. By regarding the terminals $v_1, \ldots, v_n$ of $\mathcal{G}_n$ and nonterminals $y_1, \ldots, y_n$ as mutually commuting indeterminates, $\mathcal{G}_n$ becomes a system of equations that is linear in the nonterminals. This system can be explicitly solved and the solution of each y_i will be a polynomial in the terminals. These solutions are polynomials and not rational functions because $L(\mathcal{G}_n)$ is a finite language. A detailed and much more general discussion of this method is contained in [4]. In particular, the solution for y_1 will actually be an expression E such that

$$\tilde{E} = \sum_{w \in L(\mathcal{G}_n)} w ,$$

where w is to be thought of as a monomial rather than a string. Notice that two strings w and v such that $\vec{w} = \vec{v}$ will give rise to the same monomial so that coefficients exceeding 1 are possible.

Now, G has a Hamiltonian circuit iff at least one of the monomials w in $\tilde{E}$ is admissible w.r.t. the forbidden pair set $B = \{v_i \cdot v_i \mid i = 1, \ldots, v_n\}$, In turn, this condition is equivalent to

$$D_1^1 \cdots D_n^1 E \neq 0 ,$$

i.e., the monomial $v_1 v_2 \cdots v_n$ has a coefficient of at least 1. This implies that evaluation of $D_1^1 \cdots D_n^1 E$ will provide the number of Hamiltonian circuits.

What is the time complexity of computing the polynomials $y_1, \ldots, y_n$? The solutions can be obtained by summing the products of powers of the system matrix and the vector whose i-th entry is v_1 if $y_i \to v_1$, otherwise 0. We need powers up to the n - 1-st. In evaluating each power, we do not expand anything. This means that the solution of each y_i will be an expression with a high degree of nesting. Provided this is done, the time required is certainly $n^{O(1)}$. □

3 Reduction of forbidden pairs to $\mathcal{F}$

We show that forbidden pairs problems can be PTIME reduced to $\mathcal{F}$. We employ the Hadamard product to do this. The coefficient of a monomial z in a polynomial p in the commuting indeterminates $x_1, \ldots, x_n$ is written $[z]p$. If p and q are polynomials in the commuting indeterminates $x_1, \ldots, x_n$, their Hadamard product r is the polynomial given by $[z]r = [z]p \cdot [z]q$ for all monomials z. We write $r = p \odot q$. Of course, $p \odot q = q \odot p$.

It may be helpful for the reader to see the reduction of a particular forbidden pairs problem to $\mathcal{F}$. Consider the following forbidden pairs problem (X, B, A, C) where $X = \{x_1, x_2, x_3, x_4\}$, $C = \emptyset$ and $B = \{x_1^2, \ldots, x_4^2, x_1 \cdot x_2, x_2 \cdot x_3, x_2 \cdot x_4, x_3 \cdot x_4\}$. Define four expressions as follows.

$$\begin{aligned} E_1 &= (x_1 + x_2) \cdot (1 + x_3) \cdot (1 + x_4) \\ E_2 &= (x_2 + x_3) \cdot (1 + x_1) \cdot (1 + x_4) \\ E_3 &= (x_2 + x_4) \cdot (1 + x_1) \cdot (1 + x_3) \\ E_4 &= (x_3 + x_4) \cdot (1 + x_2) \cdot (1 + x_3) \end{aligned}$$

Let $J = E_1 \odot E_2 \odot E_3 \odot E_4$. Note that $\tilde{J}$ contains a monomial w iff w is admissible w.r.t. B. We now investigate an explicit way to compute J. This method is an application of the residue theorem and was used by Jungen in his proof of the elementary part of what is now known as the Jungen-Schützenberger theorem. See [8] for a discussion of the generalisation of Jungen's theorem. A related topic is discussed in [1].

To faciliate the exposition, we introduce some auxiliary notation. For $i = 1, \ldots, 4$ let ν_i, μ_i, η_i be new variables which commute with everything. Define E_1' to be E_1, but with every occurrence of x_i replaced by $x_i \cdot \nu_i \cdot \eta_i$. Define E_2' to be E_2, but with every occurrence of x_i replaced by $1/\nu_i$. Define E_3' to be E_3, but with every occurrence of x_i replaced by μ_i/η_i. Define E_4' to be E_4, but with every occurrence of x_i replaced by $1/\mu_i$. Define $F_1 = E_1' \cdot E_2'$ and $F_2 = E_3' \cdot E_4'$. Finally define

$$K = \frac{1}{(2\pi\sqrt{-1})^{12}} \cdot \int_\gamma F_1 \cdot F_2 \cdot \frac{d\nu_1 \cdots d\nu_4 \cdot d\mu_1 \cdots d\mu_4 \cdot d\eta_1 \cdots d\eta_4}{\mu_1 \cdots \mu_4 \cdot \nu_1 \cdots \mu_4 \cdot \eta_1 \cdots \eta_4} .$$

The integral is around a circle γ of suitable radius, concentric with the origin in the complex plane. By the residue theorem, $\tilde{K}$ includes all and only those monomials that are in common to $\tilde{E}_1, \ldots, \tilde{E}_4$, i.e., $K = J$.

Using the residue theorem, K can also be wriiten as

$$K = D_{\nu_1}^{g_1} \cdots D_{\nu_4}^{g_4} \cdot D_{\mu_1}^{h_1} \cdots D_{\mu_4}^{h_4} \cdot D_{\eta_1}^{k_1} \cdots D_{\eta_4}^{k_4} F_1 \cdot F_2 \ ,$$

where g_i, h_i, k_i count the number of occurrences in $F_1 \cdot F_2$ of $1/\nu_i, 1/\mu_i, 1/\eta_i$, respectively. Thus, this forbidden pairs problem can be cast as an instance of $\mathcal{F}$. We proceed to generalise this observation in the next theorem.

The size of a forbidden pairs problem (X, B, A, C) is just the sum of the cardinalities, $|X|, |B|, |A|, |C|$ of the four sets. However, since $|B| \leq |X|^2$ and $|C| \leq |A|^2$ and $|X| = |A|$ it suffices to identify the size of the problem with $|X| = n$. Thus PTIME means $n^{O(1)}$ time. We will say that a monomial y in the X-variables and a string $w \in A^*$ are commutatively equivalent if $\vec{y} = \vec{w}$. We also apply this term in case w and y are both over A or X.

Theorem 2 *Forbidden pairs can be PTIME reduced to $\mathcal{F}$.*

Proof : Let (X, B, A, C) be a forbidden pairs problem. Let L_1 be the regular language associated with (A, C). Let $a_{1_1,2_1}, a_{2_1,2_2}, \ldots, a_{q_1,q_2}$ be a listing of all of the strings that are commutatively equivalent to the monomials of B. For $i = 1, \ldots, q$ define $A_i = A - \{a_{i_1}, a_{i_2}\}$. If U is any regular expression (or language), define $U^{[n]} = \bigcup_{j=1}^{n} U^j$, where the superscript j indicates j-fold concatenation. For $i = 1, \ldots, q$, define the regular expression E_i as

$$E_i = A_i^{[n]} + A_i^{[n]} a_{i_1} A_i^{[n]} + A_i^{[n]} a_{i_2} A_i^{[n]} \ .$$

It is easy to see that $L(E_i)$ is the set of all strings that are commutatively equivalent to monomials that are admissible w.r.t. $\{x_{i_1}^2, x_{i_2}^2, x_{i_1} \cdot x_{i_2}\}$. It follows from this that the $\bigcap_{i=1}^{q} L(E_i)$ is the set of all strings in A^* that are commutatively equivalent to the admissible monomials w.r.t. the monomial set B. We have just established, then that $w \in L_1 \cap \bigcap_{i=1}^{q} L(E_i)$ iff w is admissible w.r.t. (A, C) (membership in L_1, and $\vec{w} = \vec{y}$, where y is an admissible monomial (membership in $\bigcap_{i=1}^{q} L(E_i)$), so by the definition of forbidden pairs admissibility, $w \in L_1 \cap \bigcap_{i=1}^{q} L(E_i)$ iff there exists a monomial y such that y, w is an admissible monomial-string pair.

The regular expressions E_i are modified so that each now yields the language $L(E_i) \cap L_1$. Each of these modified regular expressions can be constructed in PTIME from the original E_i and the set C (L_1) and it is evident that the entire process can be done in PTIME since C and the original E_i can be constructed in PTIME. Notice also that the intersection of all of the modified $L(E_i)$ equals the intersection of L_1 and the original $L(E_i)$.

It will be convenient to assume that $q = 2^r$. This will make the exposition cleaner, but the tree construction that we are about to describe can be carried out for any q. For $1 \leq k \leq 2^r = q$, define $F_{0,k} = E_k$. For $1 \leq h \leq r$, and $1 \leq k \leq 2^{r-h}$ define $F_{h,k} = F_{h-1,2k-1} \odot F_{h-1,2k}$. It is clear that $F_{r,1}$ is the Hadamard product of $E_1, \ldots, E_q$. Next, for $1 \leq i \leq q$ we introduce new variables $y_{h,i}$ in the following way. Construct the complete binary tree of depth r whose vertices at depth $r - h$ are labeled left to right as $F_{h,k}$. Note that the leaves are $F_{0,1}, \ldots, F_{0,2^r}$. For depth $0 < h \leq r$, if a_i occurs in $F_{h,j}$, with odd j, associate $y_{h,i}$ with it, and if a_i occurs in $F_{h,j}$ with even j, associate $1/y_{h,i}$ with it. Notice that the root, $F_{r,1}$ does not have any variable associated with it. Starting from the children of the root, apply the following rule in moving down the tree. Each vertex sends all of its associated expressions in y variables to its left child, and all of its associated expressions in reciprocals of y variables to its right child. Now we can redefine the leaf expression $F_{0,1}$ by concatenating each occurrence of a_i with all of the associated $y_{h,i}$ and $1/y_{h,i}$. In fact this leftmost leaf will not have any reciprocals associated with it. For $F_{0,j}$, $j > 1$ we do the same thing, except that we set $a_i = \lambda$, the empty string.

We now regard all of the variables, including the a_i as mutually commutative. Define F as

$$F = \int_\gamma F_{0,1} \cdots F_{0,q} \cdot \prod_{h=0,r-1} \prod_{1 \leq i \leq q} \frac{dy_{h,i}}{y_{h,i}} .$$

The pattern of redefinition of the $F_{0,i}$ guarantees that we recover the Hadamard products $F_{h,k}$ for $h > 0$ in evaluating F so that $F = F_{r,1}$.

We now have that F, up to an integer power of $\frac{1}{2\pi\sqrt{-1}}$, of degree polynomial in n is a sum of the admissible monomials of (X, B, A, C). It is possible that admissible monomials may have coefficients greater than 1 because of nondeterminism in the regular expressions E_i. In any case, the monomial of highest total degree occurring in F is just the admissible monomial of highest total degree. Factor out each occurrence of $1/y_{h,i}$ for $0 < h \leq r - 1$ and $1 \leq i \leq q$. Let $m_{h,i}$ be the degree of $1/y_{h,i}$ that results. Note that the extra factor $dy_{h,i}/y_{h,i}$ will make a contribution here. If $1/y_{h,i}$ does not occur, $m_{h,i} = 0$. By the residue theorem, up to an integer power of $\frac{1}{2\pi\sqrt{-1}}$,

$$F = \prod_{h=1}^{r-1} \prod_{k=1}^{2^{r-h}} D_{h,k}^{m_{h,k}-1} F_{0,1} \cdots F_{0,q} ,$$

where $D_{h,k}^{m_{h,i}-1}$ is again formal $m_{h,k} - 1$-fold differentiation w.r.t. $y_{h,k}$, followed by setting $y_{h,k} = 0$. We use the convention that application of $D_{h,i}^{-1}$ is multiplication by 0. What results is just an instance of $\mathcal{F}$. □

References

[1] H. Furstenberg. Algebraic functions over finite fields. *J. of Algebra*, 7:271–277, 1967.

[2] Michael R. Garey and David S. Johnson. *Computers and Intractability.* W.H.Freeman and Company, New York, 1979.

[3] M. Hall. *Combinatorial Theory.* Wiley Interscience, 1967.

[4] W. Kuich and A. Salomaa. *Semirings, Automata, Languages.* Springer-Verlag, 1986.

[5] B. Litow. On a simple subexponential algorithm for Hamiltonian circuit. Technical report, James Cook Uni., 1998. TR98-04.

[6] V. Rödl. On a packing and covering problem. *European J. Combinatorics*, 5:69–78, 1985.

[7] A. Salomaa. *Jewels of Formal Language Theory.* Computer Sci. Press, 1981.

[8] M.P. Schützenberger. On a theorem of Jungen. *Proc. Am. Math. Soc.*, 13:885–890, 1962.

(Received 27/7/99)