Universal Relations*

Manindra Agrawal

Department of Computer Science and Engineering
Indian Institute of Technology
Kanpur - 208 016, INDIA
email : manindra@iitk.ernet.in

&

Somenath Biswas

Department of Computer Science and Engineering
Indian Institute of Technology
Kanpur - 208 016, INDIA
email : sb@iitk.ernet.in

*A preliminary version of this paper was presented at the Seventh Conference on Structure in Complexity Theory, Boston, 1992.

Running Head: Universal Relations

Name and mailing address of contact author:

Manindra Agrawal

Department of Computer Science and Engineering
Indian Institute of Technology
Kanpur 208 016, INDIA.

Abstract

Let R be a polynomial-time verifiable binary relation witnessing language A in NP. For any other polynomial-time verifiable binary relation Q witnessing some language B in NP, the notion of *solution-preserving* reduction of Q to R is defined. Informally, it is a polynomial-time reduction of B to A with the additional property that the set of witnesses (w.r.t. Q) of any instance can be quickly recovered from the set of witnesses (w.r.t. R) of the image of the instance under the reduction. Relation R is called a *universal* relation if there exists a solution-preserving reduction to R from every polynomial-time verifiable binary relation. Two properties on R are defined: *joinability* and *couplability*, and it is shown that R is a universal relation if and only if R has these two properties and a particular kind of instance. Therefore, if R has these two properties and the instance then A is NP-complete. The above characterization is used to obtain easy NP-completeness proofs for several well known natural problems, e.g., Hamiltonian cycle problem, Max Cut problem, Knapsack problem etc. It is also shown that the obvious relations for *k-creative* sets are universal. The notion of universal relations is generalized to *near-universal* relations and it is shown that the obvious witnessing relations for several problems in P and the Graph Isomorphism problem are not near-universal. Finally, the two properties joinability and couplability are related to paddability and d-self-reducibility respectively.

1 Introduction

Since the seminal work of Cook [Coo71] which showed that the satisfiability problem of the propositional logic is NP-complete, a large number of problems that arise naturally in many diverse fields, e.g., logic, graph theory, operations research, number theory, algebra etc., have been shown to be NP-complete (see [GJ78] for a number of such problems). Further, new problems are being continually added to the list of already known NP-complete problems [Joh]. One uses the term *natural* NP-*complete problems* for those NP-complete problems that arise naturally in some field or other, as opposed to those that which arise out of internal, theoretical considerations; an example of latter type being k-creative sets [JY85].

The usual way of proving a set L_1 in NP to be complete is by suitably choosing an already known NP-complete set, and then showing that this set reduces to L_1 via a polynomial-time many-one reduction. It can be observed that all such reductions are fairly similar in nature, once we abstract away the idiosyncrasies of the particular pair of sets involved in the reduction. the purpose of this paper is to formally capture what lies behind this similarity, and then to investigate into the structure of natural NP-complete sets in this light.

For our investigation, we use the observation that the reductions amongst natural NP-complete sets are *solution preserving* (A *solution* of an instance for a set is a witness of the fact that the instance belongs to the set, e.g., a satisfying assignment of a boolean formula for Satisfiability problem, a Hamiltonian cycle of a graph for Hamiltonian cycle problem), i.e., from any solution of an instance in the range of the reduction, one can easily extract a solution of the inverse of the instance. As the solutions of an instance depend on the polynomial-time relation chosen to witness the set, we work with relations instead of sets. We call a relation *universal* if the set witnessed by it is complete for NP under solution-preserving reductions (in the above sense).

It is usually the case, as we will see later in this paper, that the obvious witnessing relation of a natural NP-complete set is universal. What causes a relation to be universal? A satisfactory answer to this question will identify the structural ingredients in a set that account for its completeness. Towards this, we define two properties: *joinability* and *couplability*. Intuitively, *joinability* allows us to join any two instances so that the solutions of the resulting instance are concatenations of the solutions of the two instances, and *couplability* allows us to couple any two solution bits of an instance together so that they are always complement of each other. We show that a relation is universal if and only if it has these two properties along with an instance having a specific solution structure.

There are two important structural properties, viz., paddability, and disjunctive self-reducibility (d-self-reducibility, for short), which are known to be shared by all natural NP-complete sets. Many structural results about natural NP-complete sets make use of one, or both of these properties in an essential way. We show that joinability and couplability are related to these notions; a strong form of joinability in a relation guarantees paddability for the language witnessed by hte relation. Similarly, a strong form of couplability ensures that the witnessed language is d-self-reducible. These results can be taken as evidence of the naturalness of joinability and couplability.

Many of the standard witnessing relations for natural NP-complete sets can easily be seen to be both joinable and couplable along with having an instance with the specific solution structure, and therefore they are universal. Section 6 contains some examples. Thus, these properties capture at least some of the common structure of natural complete problems. They also give us a new method of proving NP-completeness which, at least in some cases, significantly simplifies the proof (see example 4 in section 6).

In fact, the notion of universal relations seems to capture more than just natural NP-complete sets as we show that, besides all paddable NP-complete sets, well known classes of *k-creative*

sets [JY85] have a universal witnessing relation.

Our definition of universal relations also allows us to show that certain relations are *not* universal. We exhibit relations for NP-complete sets that are not universal. We generalize our definition to *near-universal relations* and show that such non-universal relations are near-universal. Finally, we show that the standard witnessing relation for the Graph Isomorphism problem is not near-universal.

The paper is organized as follows. Section 2 gives notations to be used in the paper. In section 3, we give the definitions of solution-preserving reductions and the universal relation. In section 4, we give the definitions of the joinability and couplability properties and prove a characterization of universal relations. Section 5 contains some results that are useful in proving a given relation universal. Section 6 is devoted to providing natural examples of relations that are universal while section 7 gives natural examples of relations that are not universal. In this section we also define near-universal relations, the class of which strictly contains the class of universal relations. In section 8, we investigate the structural properties of universal relations and relate the properties of productability and couplability to paddability and d-self-reducibility respectively. Finally, section 9 contains some concluding remarks.

2 Preliminaries

2.1 Strings

All the strings are assumed to be over the alphabet $\Sigma = \{0, 1\}$ with $\Sigma_{=n}$ denoting the set of all strings of length n. For string x, $|x|$ denote the bit length of the string. For convenience, we sometimes define strings over alphabets with additional symbols, however, these can be coded into strings over Σ by coding the additional symbols using Σ. We shall also use the standard binary representation of a natural number n to code it into a string, and by abuse of notation, shall use n itself to denote the string.

We shall frequently require a function $\langle \cdot, \ldots, \cdot \rangle$ that takes n strings x_1, x_2, ..., x_n, and returns a string y such that each x_i can be easily recovered from y. This can be done by defining: $\langle x_1, x_2, \ldots, x_n \rangle = x_1 \# x_2 \# \cdots \# x_n$ where $\#$ is a new symbol (to code this string over Σ, we take 11 to represent 1, 01 to represent 0, and 00 to represent $\#$). One can easily verify that $|\langle x_1, \ldots, x_n \rangle| = 2 \cdot (|x_1| + \cdots + |x_n| + n - 1)$.

If x is a string over Σ of length n then we use $x[i]$ to denote the i^{th} bit of x, i.e., $x = x[1]x[2]\cdots x[n]$ with each $x[i] \in \Sigma$.

We shall frequently deal with functions that output a pair of strings and shall require to extract the two strings from the output. We write $(\pi_1 \circ f)(x)$ and $(\pi_2 \circ f)(x)$ to denote the first and second string respectively in the pair $f(x)$.

2.2 Projections

We use $\alpha, \beta, \gamma, \ldots$ etc. to denote *non-empty* finite sequences of *distinct positive integers*, and if α is the sequence $n_1, \ldots, n_k$, i.e., $\alpha = \{n_i\}_{i=1}^{k}$ then $\alpha[j]$ will denote n_j, the j^{th} element of the sequence. For a sequence α, $|\alpha|$ denotes the number of elements in α. For any positive integer m and a sequence α, $\alpha + m \stackrel{\text{def}}{=} \{m + \alpha[i]\}_{i=1}^{|\alpha|}$, i.e., the sequence obtained from α by adding m to each element of α. We denote by α, β the sequence which is obtained by *concatenating* β to the right of α. Such concatenation of α and β will be defined only if they have no elements in common. For sequences α and β, if the largest element of β is less than or equal to $|\alpha|$, then the sequence $\alpha \circ \beta$ is defined as follows: $\alpha \circ \beta \stackrel{\text{def}}{=} \{\alpha[\beta[i]]\}_{i=1}^{|\beta|}$, otherwise it is undefined. For example, if $\alpha = 3, 1, 5, 2, 7,$ and $\beta = 2, 1, 4,$ then $\alpha \circ \beta = 1, 3, 2.$

Let S and T be two sets containing strings of length $k+l$ and l respectively with $k, l > 0$ and α be a sequence of length l such that its largest element is not greater than $k+l$. Then we say that T *is the projection of* S *via* α, notationally, $proj_\alpha(S) = T$, if $T = \{t \mid (\exists s) s \in S \wedge t = s[n_1]s[n_2]\cdots s[n_l]\}$ where $\alpha = n_1, n_2, \ldots, n_l$. The following lemmas are obvious.

Lemma 2.1 *For every* α *and* S, *if* $proj_\alpha(S)$ *is defined, then* $proj_\alpha(S) = \emptyset \Leftrightarrow S = \emptyset$.

Lemma 2.2 $proj_{\alpha\circ\beta}(S) = proj_\beta(proj_\alpha(S))$.

We code any sequence $\alpha = \{n_i\}_{i=1}^k$ into a string as $\langle n_1, \ldots, n_k\rangle$.
Our projections are similar to the projections defined in database query languages.

2.3 Relations

For every set A in NP, by definition, there exists a polynomial-time verifiable binary relation R, $R \subseteq \Sigma^* \times \Sigma^*$, and a polynomial p such that

$$x \in A \Leftrightarrow (\exists s)[|s| \leq p(|x|) \wedge xRs].$$

We shall refer to the strings s such that xRs and $|s| \leq p(|x|)$, as *solutions* of x. Define the *solution set* of x, $sol_R(x) \stackrel{\text{def}}{=} \{s \mid xRs\}$ (for $x \notin A$, $sol_R(x) = \emptyset$). The relation R is said to be a relation *witnessing* A *to be in* NP, or simply, *witnessing* A. If one changes the polynomial p, the set witnessed by the relation R may no longer be A as the solutions may change. Thus, one needs a relation and a polynomial to specify a language in NP. To avoid specifying a polynomial every time we talk of a language witnessed by some relation, we shall consider relations whose witnessed set will be independent of the polynomial chosen.

Definition 2.3 A relation R is called an *admissible relation* if there is a polynomial-time computable function *sol-len*$_R$, *sol-len*$_R : \Sigma^* \mapsto 1^+$, such that for every x and s, if xRs then $|s| = |$*sol-len*$_R(x)|$. In other words, all the solutions for any string are of equal and polynomially-bounded length, and this length is easily computable.

Remark. For an admissible relation R, if for some string x, $sol_R(x) = \emptyset$, *sol-len*$_R(x)$ is still defined although its value has no meaning.

It is easy to see that every set in NP is witnessed by an admissible relation. For an admissible relation R, we define the set

$$L_R = \{x \mid (\exists s) xRs\}.$$

We shall restrict ourselves to admissible relations and from now on, whenever we refer to a relation, we assume it to be admissible. For any set in NP, there are infinitely many admissible relations witnessing it. However, only a few of these relations are immediately obvious from the definition of the set. We shall refer to these obvious relations as *naturally defined* relations for the set (it is difficult to formalize this notion).

3 The universal relation

For a natural NP-complete problem, it is usually the case that its instances consist of small 'atomic' units joined together in various ways, e.g., an instance of SAT has variables as atomic units, an instance of Hamiltonian cycle problem (HAM) has edges as atomic units. A solution of such an instance is usually a subset of these atomic units satisfying certain properties. Reductions amongst these problems (see e.g., [GJ78, GJS76]) map an atomic unit of the source

instance to one or more such units of the target instance in such a way that any atomic unit of the source instance is present in a solution if and only if the corresponding atomic units of the target instance are present in the corresponding solution. Thus, one can extract out the solution of the source instance given a solution of the target instance. We capture this *solution-preserving* property of the reductions in the following definition. As the notion of solution is dependent on the relation witnessing the set, we define these reductions in terms of relations.

Definition 3.1 Function f, $f : \Sigma^* \mapsto \Sigma^*$, is a *solution-preserving reduction* of relation Q to relation R if it is polynomial-time computable and satisfies the following conditions.

1. $f(x) = \langle z, \alpha \rangle$ where $x, z \in \Sigma^*$ and α is a sequence with $|\alpha| = \mathit{sol\text{-}len}_Q(x)$.
2. $\mathit{proj}_\alpha(\mathit{sol}_R(z)) = \mathit{sol}_Q(x)$.

The following proposition is easy to prove.

Proposition 3.2 *$L_Q \leq_m^p L_R$ via $(\pi_1 \circ f)$.*

Proof. For any string x, let $\alpha = (\pi_2 \circ f)(x)$. Then, $x \in L_Q$ iff $\mathit{sol}_Q(x) \neq \emptyset$ iff $\mathit{proj}_\alpha(\mathit{sol}_R((\pi_1 \circ f)(x))) \neq \emptyset$ (by definition) iff $\mathit{sol}_R((\pi_1 \circ f)(x)) \neq \emptyset$ (by Lemma 2.1) iff $(\pi_1 \circ f)(x) \in L_R$. ■

One can see that we allow a very strong type of extraction in the above definition: in polynomial-time a sequence of bits positions are computed and then a solution of the source instance is just the projection of the corresponding solution of the target instance on the computed bit positions. A weaker extraction scheme would be to give a polynomial-time function that maps solutions of the target instance to solutions of the source instance. We preferred the stronger one because such an extraction is indeed possible in case of natural NP-complete problems, thereby showing a very strong structural similarity amongst them.

Remark. Natural NP-complete sets have been related via projections in another way too. *Projection reductions* were defined in [Val82, IL95]: under such reductions, every bit of the output instance depends on at most one bit of the input instance. It has been observed (see, e.g., [PY86]) that natural NP-complete sets are reducible to each other via projection reductions.

Definition 3.3 Relation R is a *universal* relation if for every relation Q, there is a solution-preserving reduction of Q to R.

Proposition 3.4 *If R is a universal relation then L_R is* NP*-complete under polynomial-time honest reductions.*

Proof. That L_R is NP-complete follows immediately from the definition. Take any set $A \in$ NP and construct a relation Q for A such that for every x, $\mathit{sol\text{-}len}_Q(x) \geq |x|$. Now let f be a solution-preserving reduction of Q to R. It follows that $\mathit{sol\text{-}len}((\pi_1 \circ f)(x)) \geq |x|$ (since the sequence $(\pi_2 \circ f)(x)$ has $|x|$ distinct numbers) and therefore $|(\pi_1 \circ f)(x)| \geq p^{-1}(|x|)$ for some polynomial p. ■

Do universal relations exist? The answer is yes, we shall show later that witnessing relations for a number of NP-complete problems are universal. Here, we give one example: the witnessing relation for SAT.

Define R_{SAT} as: $xR_{SAT}s$ iff $|s| = n$ (n is the number of variables in x) and s codes a satisfying assignment of x with $s^i = 1$ iff variable v_i is true in the assignment.

Theorem 3.5 *Relation R_{SAT} is universal.*

Proof Sketch. Recall Cook's encoding of the computation string of any NDTM on an instance into a SAT formula [Coo71]. The formula had a variable corresponding to each bit position in the computation string plus some extra variables. Further, ignoring the assignment to extra variables, the set of satisfying assignments of the formula was *exactly* the set of accepting computation strings of the NDTM. Now consider any relation Q. One can construct an NDTM M that on any input x, guesses a string of size $\mathit{sol\text{-}len}_Q(x)$ and then verifies if it is a solution of x w.r.t. Q. One can also identify the bit positions in the computation string of M on which the guess bits are written. Then, the solution-preserving reduction of Q to R_{SAT} would reduce an instance x to the formula encoding the computation string of M on x. The corresponding sequence would have the variable numbers corresponding to the bit positions in the computation string of M that are the guess bits. The set $\mathit{sol}_Q(x)$ is just the projection of the solution set of the formula via the above sequence. ■

A variation of the Satisfiability problem is 3SAT, in which each clause is restricted to contain exactly three literals. This problem is also known to be NP-complete and one can easily show that the witnessing relation for 3SAT, R_{3SAT}, defined analogously, is also universal. This fact will be useful when we give an alternative definition of universal relations. To prove R_{3SAT} universal, we use the following lemma showing the closure of universal relations under solution-preserving reductions.

Lemma 3.6 *If R is a universal relation and there is a solution-preserving reduction of R to relation S then S is also universal.*

Proof. Let Q be any relation and f and g be the solution-preserving reductions of Q to R and R to S respectively. Define function h as: $h(x) = \langle(\pi_1 \circ g)((\pi_1 \circ f)(x)), \beta \circ \alpha\rangle$ where $\alpha = (\pi_2 \circ f)(x)$ and $\beta = (\pi_2 \circ g)((\pi_1 \circ f)(x))$. Now, $\mathit{proj}_\beta(\mathit{sol}_S((\pi_1 \circ h)(x))) = \mathit{sol}_R((\pi_1 \circ f)(x))$, and $\mathit{proj}_\alpha(\mathit{sol}_R((\pi_1 \circ f)(x))) = \mathit{sol}_Q(x)$. Therefore, by Lemma 2.2, $\mathit{proj}_{\beta\circ\alpha}(\mathit{sol}_S((\pi_1 \circ h)(x))) = \mathit{sol}_Q(x)$. So, h is a solution-preserving reduction of Q to S. ■

Corollary 3.7 *Relation R_{3SAT} is universal.*

Proof Sketch. The reduction of SAT to 3SAT maps a formula to another that contains some additional variables apart from all the variables of the source formula [GJ78]. Further, an assignment of variables satisfies the source formula if and only if it can be extended (by assigning values to the additional variables) to a satisfying assignment of the target formula. Therefore, one can construct a solution-preserving reduction of R_{SAT} to R_{3SAT} and then, from the above lemma, it follows that R_{3SAT} is universal. ■

Remark. Our notion of solution-preserving reductions is similar in spirit to that of *parsimonious* reductions [Sim77]. However, there are certain crucial differences. Our notion is a lot more restrictive in that it forces the solution sets of any instance and its image to be identical via a projection. At the same time, the actual number of solutions of the instance and its image need not be same as the 'masking' of certain bits by the projection may collapse two or more solutions in one. Thus, one can only say that the number of solutions of the image is at least as large as the number of solutions of the instance.

4 A characterization of universal relations

From the results in the previous section, we know that a relation is universal if and only if there exists a solution-preserving reduction of R_{3SAT} to it. Our aim in this section is to identify some simple properties that allow the construction of such a reduction. Towards this, we first identify the properties of R_{3SAT} that allow the building of an arbitrary instance from a single clause:

> Suppose that one wants to add a clause c to some formula x. This clause may have some common variables with x. Of course, one can simply 'and' c to x to get the desired formula but we want it to be done in a way that changes the solution set of x nicely. Consider the following procedure: first 'and' a clause c' to x that does not have *any* common variables with it, then for each variable that is common between x and c, make it equivalent to one of the unique variables of the added clause by 'and'ing two 2-literal clauses of the form $v_1 \vee \bar{v}_2$, $\bar{v}_1 \vee v_2$ or $v_1 \vee v_2$, $\bar{v}_1 \vee \bar{v}_2$, depending on whether the common variable occurs positively or negatively in c.

The two operations defined above modify the solution sets in a fairly natural way. The first operation simply 'concatenates' the solution sets of x and c'. And in the second one, starting from the instance $x \wedge c'$, we 'couple' (make equivalent or complement of each other) a variable in x with a variable in c', thereby pruning the solution space of $x \wedge c'$ to make it identical to that of $x \wedge c$. Clearly, if the two operations can be carried out on some relation in polynomial-time then a solution-preserving reduction of R_{3SAT} can be easily constructed. Now we give the formal definitions of properties that allow these operations.

We first define the 'starting point' ,i.e., an instance that acts as a single clause.

Definition 4.1 Relation R has a *building block* if there is an element in L_R, $block_R$, and three positive integers bit_1, bit_2, and bit_3 such that

$$proj_{bit_1,bit_2,bit_3}(sol_R(block_R)) = \Sigma_{=3} - \{000\}.$$

In other words, the solution set of $block_R$ has at least one solution for each assignment of the three bit positions bit_1, bit_2 and bit_3 except for the assignment 000.

Now, the properties capturing the above operations.

Definition 4.2 Relation R is *joinable* if there exists a polynomial-time computable function $join_R$, $join_R : \Sigma^* \mapsto \Sigma^*$, satisfying the following conditions.

1. $join_R(\langle x_1, \ldots, x_n \rangle) = \langle z, \alpha \rangle$ where $x_1, \ldots, x_n, z \in \Sigma^*$ and $|\alpha| = \sum_{k=1}^{n} sol\text{-}len_R(x_k)$.
2. $proj_\alpha(sol_R(z)) = \{s_1 s_2 \cdots s_n \mid (\forall k \leq n) s_k \in sol_R(x_k)\}$.

Definition 4.3 Relation R is *couplable* if there exists a polynomial-time computable function cpl_R, $cpl_R : \Sigma^* \mapsto \Sigma^*$, satisfying the following conditions.

1. $cpl_R(x, \langle i_1, \ldots, i_n \rangle, \langle j_1, \ldots, j_n \rangle) = \langle z, \alpha \rangle$ where $x \in \Sigma^*$, $1 \leq i_1, \ldots, i_n, j_1, \ldots, j_n \leq sol\text{-}len_R(x)$, $i_m \neq j_m$ for each $1 \leq m \leq n$ and $|\alpha| = sol\text{-}len_R(x)$.
2. $proj_\alpha(sol_R(z)) = \{s \mid s \in sol_R(x) \wedge (\forall m \leq n) s[i_m] \neq s[j_m]\}$.

Functions $join_R$ and cpl_R can carry out the operations mentioned in the beginning of this section, albeit the joining and coupling is obtained only via a projection. This is a very useful (and necessary) generalization as we shall see later. As one may need several applications of the operations to construct an arbitrary instance from a single clause, we have defined the functions $join_R$ and cpl_R to take any number of inputs. The following theorem characterizes universal relations in terms of these properties.

Theorem 4.4 *Relation R is universal iff R is joinable, couplable and has a building block.*

Proof. ($\Leftarrow$) We shall exhibit a solution-preserving reduction of R_{3SAT} to R. By Lemma 3.6, this would imply that R is universal. Let x be an instance of 3SAT with n variables and m clauses. Let

$$join_R(\underbrace{\langle block_R, \ldots, block_R \rangle}_{2n+m \text{ times}}) = \langle y, \alpha \rangle.$$

By definition, we have,

$$proj_\alpha(sol_R(y)) = \{s_1 \cdots s_{2n+m} \mid (\forall i \leq 2n+m) s_i \in sol_R(block_R)\}.$$

We first define a sequence β that 'picks up' the bit bit_1 (as in Definition 4.1) of each of the first $2n$ copies of $sol_R(block_R)$ in $sol_R(y)$, and all the three bits bit_1, bit_2, and bit_3 of each of the last m copies of $sol_R(block_R)$ in $sol_R(y)$.

Let $\ell = sol\text{-}len_R(block_R)$. Let $\beta[i] = \alpha[\ell \cdot (i-1) + bit_1]$, for $1 \leq i \leq 2n$, and $\beta[2n + 3 \cdot (i - 2n - 1) + j] = \alpha[\ell \cdot (i - 2n - 1) + bit_j]$ for $2n < i \leq 2n + m$, $1 \leq j \leq 3$. In other words, $\beta = \alpha[bit_1], \alpha[\ell + bit_1], \ldots, \alpha[\ell \cdot (2n-1) + bit_1], \alpha[\ell \cdot 2n + bit_1], \alpha[\ell \cdot 2n + bit_2], \alpha[\ell \cdot 2n + bit_3], \ldots, \alpha[\ell \cdot (2n + m - 1) + bit_1], \alpha[\ell \cdot (2n + m - 1) + bit_2], \alpha[\ell \cdot (2n + m - 1) + bit_3]$. Then we have,

$$proj_\beta(sol_R(y)) = \{ss't_1t_2 \cdots t_m \mid s, s' \in \Sigma_{=n} \wedge (\forall i \leq m) t_i \in \Sigma_{=3} - \{000\}\}.$$

We identify the m t's in the above set with the solution sets of m three literal clauses that have no variable in common. And the strings s and s' shall be identified with the assignment to n variables and their completements. To be a valid assignment, s' must be the bitwise complement of s. This we achieve by using the couplability of R.

Let

$$cpl_R(y, \langle \beta[1], \beta[2], \ldots, \beta[n] \rangle, \langle \beta[n+1], \beta[n+2], \ldots, \beta[2n] \rangle) = \langle z, \gamma \rangle.$$

By the definition of couplability, we have

$$proj_{\gamma \circ \beta}(sol_R(z)) = \{s\bar{s}t_1t_2 \cdots t_m \mid s \in \Sigma^*, (\forall i \leq m) t_i \in \Sigma_{=3} - \{000\}\}.$$

Finally, for every clause of x, we couple the bits of the corresponding t in the above solution set with the bits of s or $\bar{s}$ corresponding to the literals occurring in the clause. More precisely, if the j^{th} clause of x is $x_{k_1} \vee \bar{x}_{k_2} \vee x_{k_3}$, then we shall couple the first bit of t_j with the ${k_1}^{th}$ bit of $\bar{s}$, the second bit of t_j with the ${k_2}^{th}$ bit of s, and the third bit of t_j with the ${k_3}^{th}$ bit of $\bar{s}$. This would prune the solution set of z such that every remaining solution would yield an assignment (via projection, of course) that satisfies the j^{th} clause of x.

Formally, let $\gamma' = \gamma \circ \beta$, and

$$cpl_R(z, \langle \gamma'[2n+1], \gamma'[2n+2], \ldots, \gamma'[2n+3m] \rangle, \langle i_1, i_2, \ldots, i_{3m} \rangle) = \langle w, \delta \rangle,$$

where for $1 \leq k \leq 3$ and $1 \leq j \leq m$, $i_{3(j-1)+k} = \gamma'[n+l]$ if in the k^{th} position of the j^{th} clause of x the l^{th} variable occurs *positively*, $\gamma'[l]$ if it occurs *negatively*.

As is clear from the above discussion, the first n bits of the solution set of w after projection via $\delta \circ \gamma'$ would always be a satisfying assignment of x and vice versa. Therefore, letting $\delta' = \delta[\gamma'[1]], \delta[\gamma'[2]], \ldots, \delta[\gamma'[n]]$, we get

$$proj_{\delta'}(sol_R(w)) = sol_R(x).$$

It is easy to see that that w and δ' can be computed in time bounded by a polynomial in $|x|$. Therefore $f \stackrel{\text{def}}{=} \lambda x. \langle w, \delta' \rangle$ is a solution-preserving reduction of R_{3SAT} to R.

($\Rightarrow$) Since R is universal, there is a solution-preserving reduction f of R_{3SAT} to R. Let g be the solution-preserving reduction of R to R_{3SAT}. We shall construct the functions $join_R$ and cpl_R for the relation R using f, g, $join_{R_{3SAT}}$ and $cpl_{R_{3SAT}}$.

To compute $join_R(\langle x_1, \ldots, x_n\rangle)$ do the following. Let $g(x_i) = \langle y_i, \alpha_i\rangle$ for $1 \leq i \leq n$. We have, $proj_{\alpha_i}(sol_{R_{3SAT}}(y_i)) = sol_R(x_i)$ for $1 \leq i \leq n$. Now, applying $join_{R_{3SAT}}$ on these y_i's and then function f on the output would give the required instance. To elaborate, let $Sum(i) = \sum_{j=1}^{i} sol\text{-}len_{R_{3SAT}}(y_j)$ and define sequence $\beta = \alpha_1, (Sum(1)+\alpha_2), \ldots, (Sum(n-1)+\alpha_n)$. Then, letting $join_{R_{3SAT}}(\langle y_1, \ldots, y_n\rangle) = \langle y, \gamma\rangle$, we get $proj_{\gamma\circ\beta}(sol_{R_{3SAT}}(y)) = \{s_1 \cdots s_n \mid (\forall i \leq n) s_i \in sol_R(x_i)\}$. Therefore,

$$join_R(\langle x_1, \ldots, x_n\rangle) = \langle (\pi_1 \circ f)(y), (\pi_2 \circ f)(y) \circ (\gamma \circ \beta)\rangle.$$

One can similarly construct the function cpl_R and the building block for the relation is given by $block_R \stackrel{\text{def}}{=} (\pi_1 \circ f)(block_{R_{3SAT}})$. ■

One may interpret these properties, in the case of the natural complete sets, in the following way: the building block guarantees the existence of an instance having a 'rich' solution structure. The joinability property allows one to join instances together and the couplability property allows one to link any two different components of the instance. It is worth noting at this point that the existence of the building block is necessary for proving the above theorem as there are sets in P that are both joinable and couplable. An example is 2SAT, the set of all satisfiable formulas with exactly two variables in each clause. This language is in P and the witnessing relation for this set is both joinable and couplable.

5 Some results useful in application

We now would like to see if the naturally defined witnessing relations for natural NP-complete sets are universal. But before proceeding to do that, we prove some results that would simplify our task to a great extent. This section is devoted to such results.

Functions $join_R$ and cpl_R are defined to take any number of inputs and thus showing the existence of such functions may be a bit cumbersome. One can simplify this when the length of the output does not increase too rapidly.

Define

$$\begin{aligned} bprod_R(x, y) &= join_R(\langle x, y\rangle) \\ bcpl_R(x, i, j) &= cpl_R(x, \langle i\rangle, \langle j\rangle) \end{aligned}$$

Lemma 5.1 (*i*) *Suppose function $bprod_R$ exists for relation R satisfying*

$$(\forall x)(\forall y)[|(\pi_1 \circ bprod_R)(x, y)| \leq c.(|x| + |y|)]$$

for some constant c, then R is joinable.

(*ii*) *Suppose function $bcpl_R$ exists for relation R satisfying*

$$(\forall x)(\forall i)(\forall j)[|(\pi_1 \circ bcpl_R)(x, i, j)| \leq p(p^{-1}(|x|) + c)]$$

for some polynomial p and constant c, then R is couplable.

Proof. We give a recursive definition of the $join_R$ and cpl_R functions.

(*i*) Let n be an exact power of 2 and $m = n/2$ (the case when n is not an exact power of 2 can be taken care of by adding some trivial instances in the language). Let

$$\begin{aligned} \langle y_1, \alpha \rangle &= join_R(\langle x_1, \ldots, x_m \rangle), \\ \langle y_2, \beta \rangle &= join_R(\langle x_{m+1}, \ldots, x_n \rangle), \\ \ell &= \sum_{k=1}^{m} sol\text{-}len_R(x_k), and \\ \gamma &= \alpha, \ell + \beta. \end{aligned}$$

Define,

$$join_R(\langle x_1, \ldots, x_n \rangle) = \begin{cases} \langle x_1, \{i\}_{i=1}^{\ell} \rangle & \text{if } n = 1, \\ bprod_R(x_1, x_2) & \text{if } n = 2, \\ \langle (\pi_1 \circ bprod_R)(y_1, y_2), (\pi_2 \circ bprod_R)(y_1, y_2) \circ \gamma \rangle & \text{if } n > 2. \end{cases}$$

The above function will be computable in polynomial-time since it is easy to show, by induction, that $|join_R(\langle x_1, \ldots, x_n \rangle)| \leq c^{\log n} \cdot (|x_1| + \cdots + |x_n|)$.

(*ii*) Denoting

$$\begin{aligned} \langle y, \alpha \rangle &= cpl_R(x, \langle i_1, \ldots, i_{n-1} \rangle, \langle j_1, \ldots, j_{n-1} \rangle), \\ i &= \alpha[i_n], \\ j &= \alpha[j_n], \end{aligned}$$

define,

$$cpl_R(x, \langle i_1, \ldots, i_n \rangle, \langle j_1, \ldots, j_n \rangle) = \begin{cases} bcpl_R(x, i, j) & \text{if } n = 1, \\ \langle (\pi_1 \circ bcpl_R)(y, i, j), (\pi_2 \circ bcpl_R)(y, i, j) \circ \alpha \rangle & \text{if } n > 1. \end{cases}$$

It is easily verifiable, by induction, that $|cpl_R(x, \langle i_1, \ldots, i_n \rangle, \langle j_1, \ldots, j_n \rangle)| \leq p(|x| + (n - 1) * c)$ and therefore, cpl_R is computable in polynomial time. ■

Further, it is often easier to define these functions over an infinite subset S of Σ^*. We can extend them to be over Σ^* using the following theorem.

Definition 5.2 Let $S \subseteq \Sigma^*$. Relation R is *S-universal* if its building block belongs to S and the two functions $join_R$ and cpl_R are defined over S with their range also contained in S.

Theorem 5.3 *Relation R is universal iff it is S-universal for some S.*

Proof. The forward implication is obvious. We prove the reverse implication. Let R be S-universal. As the building block is in S and the functions $join_R$ and cpl_R take instances in S to instances in S itself, the function f, as defined in the proof of Theorem 4.4, can be constructed for the relation R. Since f is a solution-preserving reduction from R_{3SAT}, R would be universal. ■

Corollary 5.4 *If for some set S, relation R has a building block in S, has functions $bprod_R$ and $bcpl_R$ defined over S with their range contained in S, and the two functions satisfy Lemma 5.1, then R is universal.*

Proof. The general input versions of these functions will also be defined over S with their range in S as they are computed by a repeated application of the two input versions. The corollary follows from the above theorem. ■

6 Natural examples of universal relations

It will be shown in section 8 that every paddable NP-complete set has a universal relation. And since natural NP-complete sets are paddable (see [BH77]) it follows that all of them have a universal relation. The aim of this section is to show that even the naturally defined relations of many natural complete sets are universal.

In the examples below, we only give the instance output of the $bprod_R$ and $bcpl_R$ functions, the corresponding sequences will be obvious. Functions $join_R$ and cpl_R can be seen to exist by verifying that Lemma 5.1 holds trivially for the functions that we define.

For graph problems below, we use the following encoding of graphs into strings: Let graph $G = (V, E)$ with $|V| = n$. Assume that the vertices in V are numbered from 0 to $n-1$. We represent graph G as a string of length n^2 such that bit $\langle i, j\rangle$, for $0 \leq i, j < n$, is 'on' iff edge $(i, j) \in E$. From now on, when we refer to a string as graph, we shall mean the graph represented by the string. Similarly, reference to a bit as edge will mean the edge encoded by the bit.

Example 1: Directed Hamiltonian Cycle Problem. A Hamiltonian cycle of a graph is a cycle that passes through all the vertices. Let

$$\text{HAM} \stackrel{\text{def}}{=} \{x \mid x \text{ is a directed graph containing a Hamiltonian cycle}\}.$$

The relation witnessing HAM, R_{HAM} is: $xR_{HAM}s$ iff s is a subgraph of x and is a Hamiltonian cycle. R_{HAM} is admissible with $\textit{sol-len}_{R_{HAM}}(x) = |x|$.

Theorem 6.1 *R_{HAM} is universal.*

Proof. We show that R_{HAM} is S-universal, where S is the set of graphs that have the edge $\langle 0, 1\rangle$ *present in every solution* of the graph.

Define $(\pi_1 \circ bprod_{R_{HAM}})(x, y) = z$ where z is obtained as follows: delete edge $\langle 0, 1\rangle$ from both x and y, introduce edges from the vertex number 0 of x to the vertex number 1 of y and from the vertex number 0 of y to the vertex number 1 of x and finally renumber the vertices of x and y in the following way: let V be the number of vertices in x, then the vertex number 1 of x is assigned the number $V + 1$, the rest of the vertices of x retaining their number. For y, the vertex number 1 retains its number while the rest of them will be renumbered by adding V to their numbers. The two new edges will be present in every solution of z, thus, in particular, the edge $\langle 0, 1\rangle$ will be present in all the solutions of z.

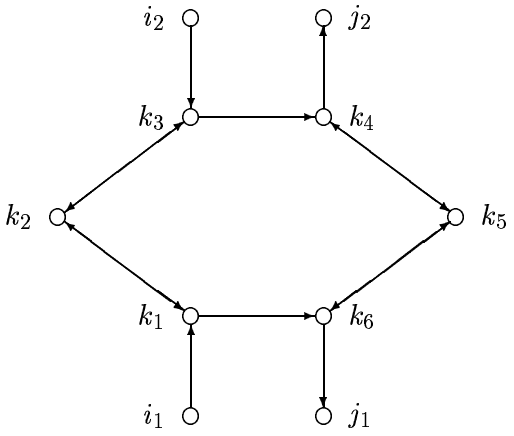

Figure 1: The modified part of the graph $(\pi_1 \circ cpl_{R_{HAM}})(x, \langle i_1, j_1\rangle, \langle i_2, j_2\rangle)$

Define $(\pi_1 \circ bcpl_{R_{HAM}})(x, e_1, e_2) = z$ where z is obtained as follows: Let $e_1 = \langle i_1, j_1 \rangle$ and $e_2 = \langle i_2, j_2 \rangle$. z has six more vertices k_1, k_2, k_3, k_4, k_5 and k_6 with the extra edges $\langle i_1, k_1 \rangle$, $\langle k_1, k_2 \rangle$, $\langle k_2, k_3 \rangle$, $\langle k_3, k_4 \rangle$, $\langle k_4, k_5 \rangle$, $\langle k_5, k_6 \rangle$, $\langle k_6, j_1 \rangle$, $\langle i_2, k_3 \rangle$, $\langle k_3, k_2 \rangle$, $\langle k_2, k_1 \rangle$, $\langle k_1, k_6 \rangle$, $\langle k_6, k_5 \rangle$, $\langle k_5, k_4 \rangle$ and $\langle k_4, j_2 \rangle$. Edges e_1 and e_2 are deleted from z (see Figure 1).

Edge $\langle i_1, k_1 \rangle$ corresponds to e_1 and $\langle i_2, k_3 \rangle$ to e_2. It is easy to see that the new vertices can be covered only by either choosing edges $\langle i_1, k_1 \rangle$, $\langle k_1, k_2 \rangle$, $\langle k_2, k_3 \rangle$, $\langle k_3, k_4 \rangle$, $\langle k_4, k_5 \rangle$, $\langle k_5, k_6 \rangle$, $\langle k_6, j_1 \rangle$ or $\langle i_2, k_3 \rangle$, $\langle k_3, k_2 \rangle$, $\langle k_2, k_1 \rangle$, $\langle k_1, k_6 \rangle$, $\langle k_6, k_5 \rangle$, $\langle k_5, k_4 \rangle$, $\langle k_4, j_2 \rangle$. The rest of the solution remains the same. A suitable renumbering of the vertices can be easily worked out to ensure that the edge $\langle 0, 1 \rangle$ is present in every solution of z.

Define $block_{R_{HAM}}$ as in the Figure 2. It can be easily seen to have the required solution set.

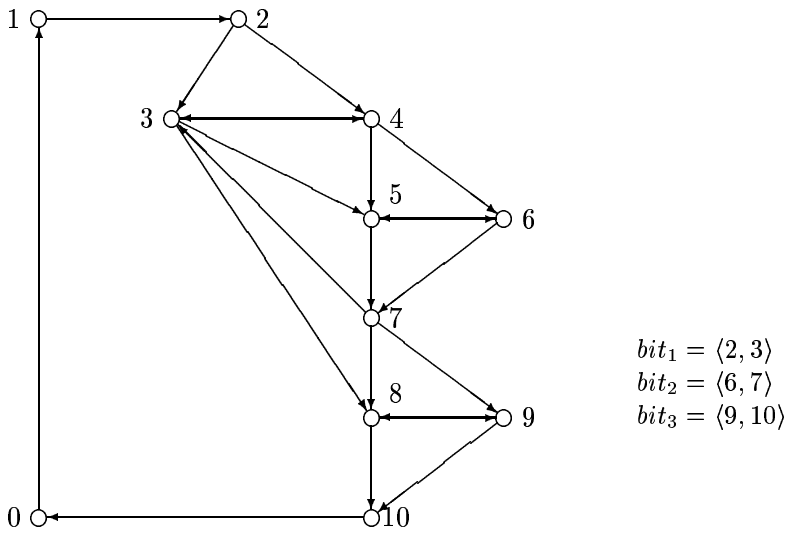

Figure 2: The instance $block_{R_{HAM}}$

■

Example 2: Independent Set Problem. An r-independent set of a graph is a set of r vertices that do not have any edge between them. Let

$$\text{IND} \stackrel{\text{def}}{=} \{\langle x, r \rangle \mid \text{ undirected graph } x \text{ has an } r\text{-independent set}\}.$$

The relation R_{IND} is: $\langle x, r \rangle R_{IND} s$ iff $|s| = n$, where n is the number of vertices in x and s has exactly r bits 'on' corresponding to the vertices in r-independent set. R_{IND} is admissible with $sol\text{-}len_{R_{IND}}(\langle x, r \rangle) = n$.

Theorem 6.2 *R_{IND} is universal.*

Proof. We prove this by showing that R_{IND} is S-universal where S is the set of instances $\langle x, r \rangle$ that have *at most* an r-independent set.

Define $(\pi_1 \circ bprod_{R_{IND}})(\langle x, r_1 \rangle, \langle y, r_2 \rangle) = \langle z, r_1 + r_2 \rangle$, graph z is obtained by putting x and y together and renumbering all the vertices of y to make them different from those of x. Since x and y can have at most r_1-independent set and r_2-independent set respectively, z will have at most $r_1 + r_2$-independent set, and thus z also belongs to S. Also, any solution of z is a concatenation of solutions of x and y.

Define $(\pi_1 \circ bcpl_{R_{IND}})(\langle x, r\rangle, i, j) = \langle z, r+1\rangle$. Graph z is obtained as follows: Add two new vertices k_1, k_2 and the following edges to x: between k_1 and k_2, between k_1 and i, between k_2 and j, between k_1 and every vertex adjacent to j, between k_2 and every vertex adjacent to i. Since x has at most an r-independent set, any solution of z must have exactly one of the vertices k_1 or k_2 in the independent set. Vertex k_1 will be present in the independent set for exactly those solutions of x in which j is present and i is not while vertex k_2 will be present in the independent set for exactly those solutions of x in which i is present and j is not.

Define instance $block_{R_{IND}} = \langle G, 3\rangle$ where G is the graph given in Figure 3. It can easily be seen to satisfy the required properties.

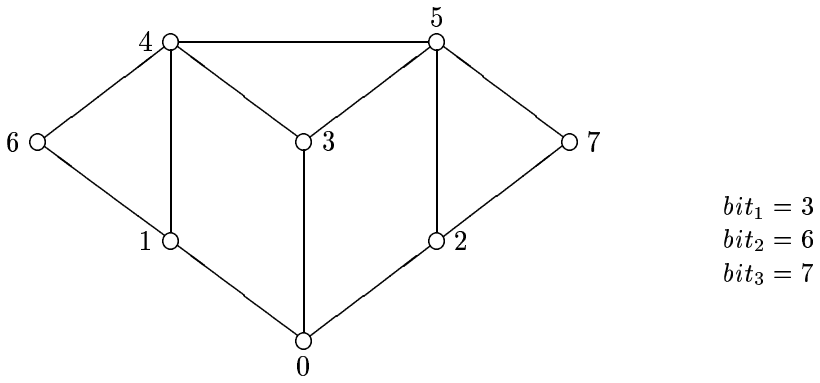

Figure 3: Graph for the instance $block_{R_{IND}}$

■

Example 3: Knapsack Problem. This problem has $r+1$ numbers as input and the solution is a subset of the first r numbers such that the sum of numbers in the subset is equal to the $(r+1)^{th}$ number. Let

$$KNP \stackrel{\text{def}}{=} \{\langle n_1, n_2, \ldots, n_r, m\rangle \mid \text{ there is a subset of the set } \{n_1, \ldots, n_r\} \text{ having sum } m\}$$

Define relation R_{KNP} as: $\langle n_1, \ldots, n_r, m\rangle R_{KNP} s$ iff $|s| = r$ and the set of numbers $\{n_i \mid s^i = 1\}$ has sum m.

Theorem 6.3 *R_{KNP} is universal.*

Proof. Define $(\pi_1 \circ bprod_{R_{KNP}})(x, y) = z$, where $x = \langle n_1, \ldots, n_r, m_1\rangle$, $y = \langle l_1, \ldots, l_s, m_2\rangle$ and $z = \langle n_1, \ldots, n_r, l_1 * nsum, \ldots, l_s * nsum, m_1 + m_2 * nsum\rangle$ where $nsum = 1 + \sum_{i=1}^{r} n_i$.

Define $(\pi_1 \circ bcpl_{R_{KNP}})(x, i, j) = z$, where $x = \langle n_1, \ldots, n_r, m\rangle$ and $z = \langle l_1, \ldots, l_r, m'\rangle$. For each k, $k \neq i$ or j, $l_k = n_k$, $l_i = n_i + nsum$, $l_j = n_j + nsum$ and $m' = m + nsum$, $nsum$ is defined as before.

Define $block_{R_{KNP}} = \langle 1, 1, 1, 1, 1, 3\rangle$ with $bit_1 = 1$, $bit_2 = 2$, and $bit_3 = 3$. ■

Example 4: Simple Max Cut Problem. An r-cut of an undirected graph is a partition of its vertex set in two subsets such that there are r edges between them. Let

$$\text{SMC} \stackrel{\text{def}}{=} \{\langle x, r\rangle \mid x = (V, E) \text{ is an undirected graph having at least an } r\text{-cut}\}$$

Define relation R_{SMC} as: $\langle x, r\rangle R_{SMC} s$ iff s is a subset of the vertices of x giving at least an r-cut and s *contains vertex number* 0. If this condition is not present then the relation is not universal (see Example 5, section 7).

Theorem 6.4 *R_{SMC} is universal.*

Proof. We show that R_{SMC} is S-universal with S being the set of instances $\langle x, r\rangle$ that have *at most* an r-cut.

Define $(\pi_1 \circ bprod_{R_{SMC}})(\langle x, r_1\rangle, \langle y, r_2\rangle) = \langle z, r_1 + r_2\rangle$, where z is the graph obtained by *collapsing* the vertices numbered 0 in x and y into a single vertex also numbered 0 in z. The rest of the vertices remain as they are, except for a renumbering.

Define $(\pi_1 \circ bcpl_{R_{SMC}})(\langle x, r\rangle, i, j) = \langle z, r + 3\rangle$, where graph z is obtained by adding two new vertices k_1 and k_2 to x and joining i to k_1, k_1 to k_2, and k_2 to j. This ensures that exactly one of i and j must be chosen. The building block for R_{SMC} is $\langle G, 6\rangle$ where G is the graph given in Figure 4.

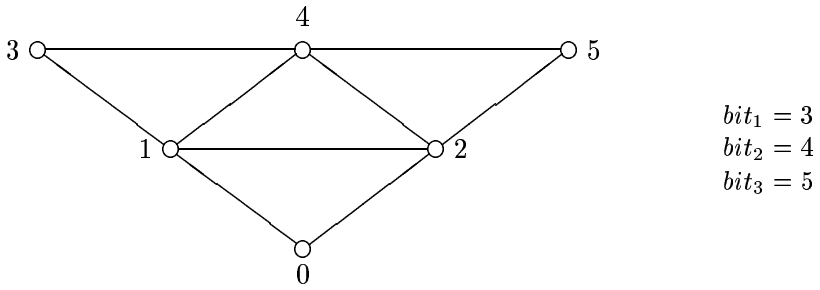

Figure 4: Graph for the instance $block_{R_{SMC}}$

■

It has been our experience, and is evident from the above examples as well, that by choosing the set S judiciously, the required properties are easily obtainable for a relation R witnessing a natural NP-complete problem. Therefore, it appears that proving NP-completeness for many problems is easier this way than the standard one of choosing a known NP-complete problem and constructing a reduction from it. An example is the Simple Max Cut problem, which was proved to be NP-complete through a complicated reduction from MAX SAT2 (see [GJS76]). Here the same problem has an exceedingly simple proof of completeness.

Remark. At a first glance, obtaining a building block may appear to be a difficult exercise. However, our experience is that the solution set structure required in a building block serves as a very useful guide towards its discovery. In any case, if R is universal, it is possible to mechanically obtain a building block by enumerating members of L_R and examining their solution sets (one need not consider large instance sizes for this as usually a building block is of small size). Therefore, obtaining a building block should not be considered a difficult step in our proposed method of proving NP-completeness.

7 Examples of non-universal relations

We can exploit the strong extraction scheme that we have for solution-preserving reductions to show certain relations non-universal. As all universal relations witness NP-complete sets, one

can hope to show that the witnessing relations of languages believed to be non-complete are not universal. The following lemma will be very useful in this context.

Lemma 7.1 *Let R be a universal relation and W be a set of equal length strings. Then there is an instance x and a sequence α such that $proj_\alpha(sol_R(x)) = W$.*

Proof Sketch. One can easily construct a SAT instance y such that $sol_{R_{SAT}}(y) = W$. Now, by the definition of universal relations, it follows that there is an instance x and a sequence α satisfying the required property. ■

Do there exist non-universal relations? The answer is yes. In fact, one can very simply construct non-universal relations for every set in NP. Take any relation R witnessing some set A. Define a new relation R' as follows: $xR'w$ iff either xRw or $xR\bar{w}$ where string $\bar{w}$ is the bitwise complement of w. There is no instance such that $\{1\}$ is the projection of the solution set of the instance under R' via some sequence and therefore, by Lemma 7.1, R' is not universal. So, one cannot claim that non-universality of a relation implies that the witnessed problem is not complete. In the case of natural problems, one may still claim, somewhat informally, that the non-universality of their *naturally defined* witnessing relation(s) implies the non-completeness of the problem. But this claim also fails. To see this consider the following relation for Simple Max Cut problem (defined in section 6): $\langle x, r\rangle R'_{SMC} s$ iff s is a subset of vertices of x giving at least an r-cut. Note that if s is an r-cut of $\langle x, r\rangle$, so will be $\bar{s}$ and therefore, R'_{SMC} is not universal.

The reason for non-universality of all the above relations is that they have redundant solutions: if s is a solution then $\bar{s}$ is the same solution in a different encoding. We would like to remove this redundancy and then see if the modified relation is universal or not. Hopefully, such an approach would allow us to differentiate between natural complete and non-complete problems. So, we first define the *refinement* of a relation in which the duplicate solutions are 'compressed' into a single one.

Definition 7.2 Relation $\tilde{R}$ is a *refinement* of the relation R if there is a polynomial-time computable many-one function h, $h : \Sigma^* \mapsto \Sigma^*$, such that for every x and s, function h satisfies the following conditions.

1. For every s, $h^{-1}(s)$ is either $\emptyset$ or contains s.
2. $x\tilde{R}h(s) \Longleftrightarrow xRs$.

Function h acts as a compressor of the solutions. The conditions on h ensure that it compresses solutions into solutions. An example of the refinement of a relation is given by R_{SMC} defined in section 6. This relation is a refinement of the relation R'_{SMC} defined above with $h(0s) = h(1s) = 1s$. Now we define a more general notion of universal relations.

Definition 7.3 Relation R is *near-universal* if there is a refinement $\tilde{R}$ of R such that $\tilde{R}$ is universal.

Every universal relation is trivially near-universal, as any relation is a refinement of itself. The following proposition immediately follows.

Proposition 7.4 *Set A has a near-universal relation iff it has a universal relation.*

Relation R'_{SMC} is an example of a non-trivial near-universal relation. For most of the natural problems, it turns out that their naturally defined relations do not have any non-trivial refinement as for every solution string of an instance they have another instance having *only that string as the solution.* This forces the function h to be one-one. For natural languages in P, it is very easy to show that their witnessing relations are not near-universal. For example, one can show that 2SAT does not have a building block, Horn Clause Satisfiability problem does not have a solution set that is projected to $\{10, 01\}$ via some sequence etc. We now consider the Graph Isomorphism problem which is neither known to be complete nor known to be in P. Further, it is believed to be non-complete [Sch88]. We give further evidence of this by showing that its naturally defined witnessing relation is not near-universal.

Example 5: Graph Isomorphism. The problem is to find out if the given two graphs are isomorphic. Let

$$\text{GISO} \stackrel{\text{def}}{=} \{\langle G, H\rangle \mid \text{ graphs } G \text{ and } H \text{ are isomorphic}\}.$$

The relation R_{GISO} is: $\langle G, H\rangle R_{GISO} s$ iff s encodes an isomorphism of G and H in the following way—$s[(j-1)\cdot n + k] = 1$ iff vertex j of G is mapped to vertex k of H, where n is the number of vertices in G. Clearly $sol\text{-}len_{GISO}(\langle G, H\rangle) = n \cdot m$, where m is the number of vertices in H (if an isomorphism exists between G and H, then $n = m$).

Theorem 7.5 *R_{GISO} is not near-universal.*

Proof. We first note that R_{GISO} does not have any non-trivial refinement, as for every isomorphism f, two graphs can be constructed with the only isomorphism between them being f. Now, let $W = \{11, 10, 01\}$ and assume that there is an instance $z_0 = \langle G_0, H_0\rangle$ and a sequence α such that $proj_\alpha(sol_{R_{GISO}}(z_0)) = W$. Let $\alpha = (i_1 - 1)\cdot n + j_1, (i_2 - 1)\cdot n + j_2$, where n is the number of vertices in G_0.

For any instance $z = \langle G, H\rangle$, let $\mathcal{F}(z, i, j)$ be the set of all isomorphic mappings between graphs G and H that map vertex i of G to vertex j of H. Define, $V(z, i, j, k) = \{l \mid f \in \mathcal{F}(z, i, j) \wedge f(k) = l\}$. $V(z, i, j, k)$ is the set of vertices of H to which the vertex k of G is mapped to under the different isomorphisms in $\mathcal{F}(x, i, j)$.

Claim: For any z, i, j_1, j_2 and k: if the sets $\mathcal{F}(z, i, j_1)$ and $\mathcal{F}(z, i, j_2)$ are non-empty then $|V(z, i, j_1, k)| = |V(z, i, j_2, k)|$.

Proof. Fix mappings f_1 and f_2 from $\mathcal{F}(z, i, j_1)$ and $\mathcal{F}(z, i, j_2)$ respectively (since the sets are non-empty, such mappings exist). Define $g = f_2 \circ f_1^{-1}$. g is an automorphism of graph H with $g(j_1) = j_2$. Similarly, g^{-1} $(= f_1 \circ f_2^{-1})$ is also an automorphism of the graph H with $g^{-1}(j_2) = j_1$.

It follows that for any $h \in \mathcal{F}(z, i, j_1)$, $g \circ h \in \mathcal{F}(z, i, j_2)$ and for any $h \in \mathcal{F}(z, i, j_2)$, $g^{-1} \circ h \in \mathcal{F}(z, i, j_1)$. Now, since g and g^{-1} are automorphisms, we have, $\mathcal{F}(z, i, j_2) = g(\mathcal{F}(z, i, j_1))$ and $\mathcal{F}(z, i, j_1) = g^{-1}(\mathcal{F}(z, i, j_2))$. Therefore, $g(V(z, i, j_1, k)) = \{(g \circ f)(k) \mid f \in F(z, i, j_1)\} = \{f'(k) \mid f' \in F(z, i, j_2)\} = V(z, i, j_2, k)$. Since g is one-one, $|V(z, i, j_1, k)| = |V(z, i, j_2, k)|$. □

We have $proj_\alpha(sol_{R_{GISO}}(z_0)) = \{11, 01, 10\}$ with $\alpha = (i_1 - 1)\cdot n + j_1, (i_2 - 1)\cdot n + j_2$. So, there is an isomorphic mapping between G_0 and H_0 in which vertex i_1 of G_0 is *not* mapped to vertex j_1 of H_0. Suppose it is mapped to vertex j_3 of H_0. So, in any mapping of $F(z_0, i_1, j_3)$, vertex i_2 of G_0 *must* be mapped to vertex j_2 of H_0. Thus, $V(z, i_1, j_3, i_2) = \{j_2\}$. Consider the case when vertex i_1 is mapped to vertex j_1. There will exist two different mappings in $F(z, i_1, j_1)$ such that in one of them vertex i_2 is mapped to vertex j_2 and in the other vertex i_2 is not mapped to vertex j_2. Thus $|V(z, i_1, j_1, i_2)| \geq 2$. This contradicts the above claim. Therefore, by Lemma 7.1, R_{GISO} is not universal. ■

8 Structural properties of universal relations

In this section, we look at the universal relations, and related properties, 'structurally'. In particular, we show that the properties of joinability and couplability are closely related to *paddability* ([BH77]) and *d-self-reducibility* ([Sel88]) respectively. We also show that all the known NP-complete sets appear to have a universal relation, thus indicating that all NP-complete sets may have a universal relation. Finally we define a new subclass of NP-complete sets using universal relations.

8.1 Joinability

For a joinable relation R, we have, $join_R(x, y) \in L_R$ iff $x \in L_R \wedge y \in L_R$. Thus, one can 'pad' any instance using this function. Paddability was defined in [BH77]:

Definition 8.1 Set A is *paddable* if there exist two polynomial-time functions $p : \Sigma^* \times \Sigma^* \mapsto \Sigma^*$, and $p' : \Sigma^* \mapsto \Sigma^*$ such that for all $x, y \in \Sigma^*$: $p(x, y) \in A$ iff $x \in A$ and $p'(p(x, y)) = y$.

Say that function $join_R$ is *p-invertible* if there exists a polynomial-time function f such that for every y, $y = \langle x_1, x_2, \ldots, x_n \rangle$, $f((\pi_1 \circ join_R)(y)) = y$. The following theorem relates joinability to paddability.

Theorem 8.2 *Let R be a joinable relation with function* join_R *being p-invertible. Then L_R is paddable.*

Proof. Let x_0 and x_1 be two different instances of A with x_0, $x_1 \in L_R$ (the case when $L_R = \emptyset$ can be taken care of easily). Define function $p(x, y) = (\pi_1 \circ join_R)(\langle x, x_{y[1]}, x_{y[2]}, \ldots, x_{y[|y|]} \rangle)$. Function p' can also be computed in polynomial-time using the inverse of $join_R$. Moreover, $x \in L_R$ iff $pad(x, y) \in L_R$ for any y. Therefore, L_R is paddable. ■

8.2 Couplability

The couplability property allows one to restrict the solution space of an instance. Therefore, we can use it to restrict the solution space in such a way that the resulting instance can only have a few solutions, if at all. This property is similar to d-self-reducibility [Ko83]:

Definition 8.3 [Ko83] An irreflexive partial order $\sqsubset_d$ on Σ^* is *polynomially related* if there is a polynomial p such that

1. $x \sqsubset_d y$ implies $|x| \leq p(|y|)$,
2. $x \sqsubset_d y$ is decidable in time polynomial in $|x| + |y|$, and
3. $x_1 \sqsubset_d x_2 \sqsubset_d \cdots \sqsubset_d x_k$ implies $k \leq p(|x_k|)$.

A set L is *disjunctively self-reducible* if there is a polynomial-time oracle DTM M such that $L = L(M, L)$, and on input x, M generates queries $y_1, y_2, \cdots, y_m$ $(m \geq 0)$ and accepts x iff for some i, $1 \leq i \leq m$, $y_i \in L$, where $y_i \sqsubset_d x$ for each i.

The following theorem shows how couplability can be used to obtain d-self-reducibility for the set. Define p-invertibility for the function cpl_R in the same way as for $join_R$.

Theorem 8.4 *Let R be a couplable relation with function* cpl_R *being p-invertible. Then L_R is d-self-reducible.*

Proof. We begin by defining a polynomially related irreflexive partial order on the strings that the self-reducing TM for A will make use of.

For any y, say that y is *properly invertible* if

1. $cpl_R^{-1}(y)$ is defined, and
2. let $y = (\pi_1 \circ cpl_R)(x, \langle i_1, \ldots, i_m\rangle, \langle j_1, \ldots, j_m\rangle)$, then the following conditions hold—

   (i) $m \leq sol\text{-}len_R(x) - 1$,

   (ii) $i_1 = 1 \neq j_1$, and $\{i_1, i_2, \ldots, i_m\} \subseteq \{1, j_1\}$,

   (iii) $\{j_1, j_2, \ldots, j_m\} = \{2, 3, \ldots, m+1\}$, and $j_2 < j_3 < \cdots < j_m$.

Define the relation $<$ as: $y < z$ iff y is properly invertible and one of the following two cases hold:

Case 1 : z is properly invertible with
1. $z = (\pi_1 \circ cpl_R)(x, \langle i_1, \ldots, i_m\rangle, \langle j_1, \ldots, j_m\rangle)$,
2. $m < sol\text{-}len_R(x) - 1$, and
3. $y = (\pi_1 \circ cpl_R)(x, \langle i_1, \ldots, i_{m+1}\rangle, \langle j_1, \ldots, j_{m+1}\rangle)$.

Case 2 : z is not properly invertible, and $y = (\pi_1 \circ cpl_R)(z, \langle i_1\rangle, \langle j_1\rangle)$.

Define the partial ordering $<^*$ as the transitive closure of $<$. To see that $<^*$ is polynomially related, we note that firstly, the length of any $<^*$-decreasing chain starting from any string z is less than $sol\text{-}len_R(z)$ if z is not properly invertible, and is less than $sol\text{-}len_R(x)$ where $x = \pi_1(cpl_R^{-1}(z))$ otherwise. In either case, this length is bounded by a polynomial in $|z|$. Secondly, the size of every element in this chain is also bounded by a polynomial in $|z|$. And finally, it is polynomial time decidable whether $y <^* z$ for any y and z.

The relf-reducing DTM that we define for A uses this ordering on strings. We first give an informal description of the TM. On any input z, the TM checks if z is properly invertible. If it is not, then—except for 1^n and 0^n ($n = sol\text{-}len_R(z)$) which can be checked separately—in any solution of z the first bit of the solution must be the complement of some other bit. So, the TM couples each of the last $n-1$ bits with the first bit one by one, and accepts iff any of the $n-1$ instances thus generated belongs to A. Note that all these instances will be below z in the partial order $<^*$.

On the other hand, if z *is* properly invertible, then it is the image of some instance x after a number of x's solution bits have been coupled together. If there are still some solution bits of x left that are not coupled, the TM picks one such bit and couples it with the first bit and its complement (to which the first bit has been coupled earlier) one by one. It accepts iff any of the two instances thus generated belongs to A. Finally, if all the bits of x have been coupled, then there are only two possible solutions for x left corresponding to the two assignments to the first bit (the rest of the bits get fixed by this). The TM can thus test this in polynomial-time and accept z if any of the two assignments is a solution of x.

Now, we give the formal description of the d-self-reducing TM. The TM, on input z, works as follows:

1. if z is not properly invertible then, letting $n = sol\text{-}len_R(z)$, accept if 0^n or 1^n is a solution of z. Otherwise, compute the set

   $$Q = \{(\pi_1 \circ cpl_R)(z, \langle 1\rangle, \langle 2\rangle), (\pi_1 \circ cpl_R)(z, \langle 1\rangle, \langle 3\rangle), \ldots, (\pi_1 \circ cpl_R)(z, \langle 1\rangle, \langle n\rangle)\},$$

   and accept iff $Q \cap A \neq \emptyset$.

2. if z is properly invertible with

   (a) $z = (\pi_1 \circ cpl_R)(x, \langle i_1, \ldots, i_m \rangle, \langle j_1, \ldots, j_m \rangle)$, and

   (b) $m < sol\text{-}len_R(x) - 1$,

   then choose the *smallest* number j not in the set $\{1, j_1, \ldots, j_m\}$ and compute the set

$$Q = \{(\pi_1 \circ cpl_R)(x, \langle i_1, \ldots, i_m, 1 \rangle, \langle j_1, \ldots, j_m, j \rangle), \\ (\pi_1 \circ cpl_R)(x, \langle i_1, \ldots, i_m, j_1 \rangle, \langle j_1, \ldots, j_m, j \rangle)\}.$$

   Accept iff $Q \cap A \neq \emptyset$.

3. if z is properly invertible with

   (a) $z = (\pi_1 \circ cpl_R)(x, \langle i_1, \ldots, i_m \rangle, \langle j_1, \ldots, j_m \rangle)$, and

   (b) $m = sol\text{-}len_R(x) - 1$,

   then construct the string s, $|s| = sol\text{-}len_R(x)$, such that $s[j] = 1$ if $j = 1$ or $(\exists k) i_k = j_1$ and $j_k = j$; $s[j] = 0$ if $(\exists k) i_k = 1$ and $j_k = j$. Accept iff s or $\bar{s}$ is a solution of x.

As explained above, this TM accepts L_R. ■

8.3 Universal relations for non-natural sets

We have seen that if a set has a universal relation then it is NP-complete. Can we say that *every* NP-complete set has a universal relation? An affirmative answer will obviously imply P$\neq$ NP as finite sets can not have a universal relation (this follows from Proposition 3.4) and so it is not an easy question to answer. However, all sets in the p-isomorphism degree of SAT can be easily shown to have a universal relation.

Proposition 8.5 *Let A be p-isomorphic to* SAT. *Then there is a universal relation R witnessing A.*

Proof. Let f be the polynomial-time isomorphism such that $x \in A$ iff $f(x) \in$ SAT. Define relation R as: xRw iff $f(x) R_{SAT} w$. It follows that f^{-1} (coupled with the identity projection) is a solution-preserving reduction of R_{SAT} to R. Therefore, R is universal (by Lemma 3.6). ■

What can we say about NP-complete sets that are not p-isomorphic to SAT? The existence of such sets was conjectured in [JY85] and is widely believed. In [JY85], a subclass of NP-complete sets, called the *k-creative sets*, is defined using one-one, honest, polynomial-time computable functions. We show that even these sets have universal relations. Let (as in [JY85])

$$K_f^k \stackrel{\text{def}}{=} \{f(i) \mid M_i \text{ accepts } f(i) \text{ within } |i|.|f(i)|^k + |i| \text{ steps}\}$$

where f is polynomial-time computable, one-one, honest and $k > 0$.

It is believed that for many one-way functions f (a one-way function is a one-one, honest, polynomial-time function that is not p-invertible; such functions exist if P $\neq$ UP [Ko85, GS84]), these sets are not p-isomorphic to SAT. For these sets, the most obvious admissible relations witnessing them is the following: $xR_{k,f}i\#w$ iff $f(i) = x$, $|i\#w| = p(|x|)$ and w is an accepting computation of M_i on x for some fixed polynomial p.

Theorem 8.6 *Relation $R_{k,f}$ is universal.*

Proof. Let $y = \langle x_1, \ldots, x_n \rangle$. Define, $join_{R_{k,f}}(y) = \langle f(g(y)), \alpha \rangle$, where TM $M_{g(y)}$, on input z, rejects if $|z| < |y|$, otherwise guesses the string $s = i_1 \# w_1 \# i_2 \# w_2 \# \cdots \# i_n \# w_n$ and accepts iff for every r less than or equal to n, $x_r R_{k,f} i_r \# w_r$. By suitably padding g, we can ensure that $M_{g(y)}$ halts within $|g(y)|$ steps and $|f(g(y))| \geq |y|$, and therefore, $f(g(y)) \in K_k^f$ iff $M_{g(y)}$ accepts $f(g(y))$. One can also ensure that the above guess string is written in some fixed bit positions in the accepting computation of $M_{g(y)}$. This enables one to compute α properly.

Define,

$$cpl_{R_{k,f}}(x, \langle i_1, \ldots, i_n \rangle, \langle j_1, \ldots, j_n \rangle) = \langle f(h(x, \langle i_1, \ldots, i_n \rangle, \langle j_1, \ldots, j_n \rangle)), \beta \rangle,$$

where TM $M_{h(x,\langle i_1,\ldots,i_n\rangle,\langle j_1,\ldots,j_n\rangle)}$ on input z, rejects if $|z| < |x|$, otherwise guesses the string $s = i \# w$ and accepts iff $x R_{k,f} i \# w$ and for every r less than or equal to n, i_r^{th} and j_r^{th} bits of the string $i \# w$ are different. Other properties can be ensured by making h satisfy the same conditions as g above.

The instance $block_{R_{k,f}}$ is trivial: $block_{R_{k,f}} = f(i_0)$, where M_{i_0}, on input z, guesses a string of length three and accepts iff the guessed string is not 000. ∎

Thus, universal relations capture more than just the structure of natural NP-complete sets. In fact, one can construct an entirely new subclass of NP-complete sets, using one-one and size-increasing functions, such that every set in the subclass has universal relation.

Let f be any one-one and size-increasing polynomial-time computable function. Define relation R_f as follows.

zR_fw iff $|w| = |z|^2$ and one of the following three conditions hold—

1. $z = 00$ and $w \in \{0001, 0010, 0011, 0100, 0101, 0110, 0111\}$.
2. $w = x_1\$w_1\$x_2\$w_2\$\cdots\$x_n\$w_n\#^r$ (# and \$ are symbols not in $\{0,1\}$), $f(\langle 1, x_1, x_2, \ldots, x_n \rangle) = z$ for some $r > 0, n > 1$ and $(\forall i \leq n) x_i R_f w_i$.
3. $w = x\$i_1\$\cdots\$i_n\$j_1\$\cdots\$j_n\$w'\#^r$ for some $r > 0, n > 0$, $f(\langle 0, x, i_1, \ldots, i_n, j_1, \ldots, j_n \rangle) = z$, xR_fw' and for every k, $1 \leq k \leq n$, the i_k^{th} and the j_k^{th} bits of w' are different.

Theorem 8.7 *Relation R_f is universal.*

Proof Sketch. Since f is one-one, at most one of the above three conditions will hold for any z and f is size-increasing, R_f can be computed in polynomial-time. Define

$$\begin{aligned} join_{R_f}(\langle x_1, \ldots, x_n \rangle) &= \langle f(\langle 1, x_1, \ldots, x_n \rangle), \alpha \rangle, \\ cpl_{R_f}(x, \langle i_1, \ldots, i_n \rangle, \langle j_1, \ldots, j_n \rangle) &= \langle f(\langle 0, x, i_1, \ldots, i_n, j_1, \ldots, j_n \rangle), \beta \rangle. \end{aligned}$$

The sequences α and β are obvious. ∎

Now define the set U_f as—

$$U_f \stackrel{\text{def}}{=} \{z \mid (\exists w) z R_f w\}$$

Corollary 8.8 *For every one-one and size-increasing polynomial-time function f, the set U_f is* NP*-complete.*

Proof. Follows from the above theorem and Proposition 3.4. ∎

9 Concluding Remarks

In this paper, we have defined the notion of universal relations, and have given a new method for proving a set to be NP-complete based on this notion. Our principal aim was to capture the common structure of natural NP-complete problems. If, for every natural NP-complete problem there is a naturally defined universal relation witnessing it, then our aim would be fulfilled. There is some evidence that it is indeed true. Firstly, the intuitive evidence: the two properties joinability and couplability are very natural ones. Secondly, the empirical evidence: though we do not claim to have verified the existence of universal relations for a large fraction of known NP-complete problems, we have shown that very different kind of complete problems have universal witnessing relations. In this paper, we have given five examples of these: one problem from logic (Satisfiability), one from number theory (Knapsack), three from graph theory with first an edge-deletion problem (Hamiltonian cycle), second an edge-addition problem (Independent set) and third a node-deletion problem (Max cut). So, this property is, at the very least, not restricted to certain kind of NP-complete sets. However, we would like the following stronger question to be answered, as a positive answer to it allows us to separate natural complete problems from non-complete ones.

Question 1 *Are naturally defined relations for every natural complete problem near-universal?*

Note that this question is imprecise as the notion of naturally defined relations for languages is not formalized. So, a positive answer to the question is difficult to obtain (it is difficult for other reasons too: it would imply P $\neq$ NP!). However, a negative answer can be given by exhibiting a natural complete language with a reasonable relation which is not near-universal.

Remark. Note that it is possible to construct witnessing relations for natural NP-complete problems that are not even near-universal. However, these relations are not naturally defined.

The notion of universal relations goes beyond the natural sets. As is shown in section 8, even non-natural NP-complete sets have naturally defined universal relations. We believe that this notion may provide an interesting weakening of the notion of p-isomorphism. If two sets are both shown to have universal relations then this immediately implies that the two sets have many common structural properties. Also, for certain sets it is easier to show that they have universal relations than showing them to be p-isomorphic to SAT. As an example, we recall the case of k-creative sets. It is even conceivable that all NP-complete sets may have universal relations without their being in the same isomorphic degree. So, our next question is:

Question 2 *Do all* NP*-complete sets have universal relations?*

Again, a positive answer to this question implies P $\neq$ NP. However, answering this question either way is hard even under the assumption P $\neq$ NP. The reason is that there is an oracle A, for which $\mathrm{P}^A \neq \mathrm{NP}^A$ and there are dishonest NP^A-complete sets [HH91] and therefore in this world there will be complete sets without universal relations by Proposition 3.4 (the notion of universal relations easily relativizes). Also, there is an oracle A for which all NP^A-complete sets are p^A-isomorphic [FFK92] and therefore all complete sets will have universal relations in this world by Proposition 8.5.

References

[BH77] L. Berman and J. Hartmanis. On isomorphism and density of NP and other complete sets. *SIAM Journal on Computing*, 1:305–322, 1977.

[Coo71] S. Cook. The complexity of theorem proving procedures. In *Proceedings of STOC*, pages 151–158, 1971.

[FFK92] S. Fenner, L. Fortnow, and S. Kurtz. The isomorphism conjecture holds relative to an oracle. In *Proceedings of FOCS*, pages 30–39, 1992. To appear in SIAM J. Comput.

[GJ78] M. Garey and D. Johnson. *Computers and Intractability : A Guide to the Theory of NP-completeness*. Freeman, San Francisco, 1978.

[GJS76] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[GS84] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. In *Proceedings of FOCS*, pages 495–503, 1984.

[HH91] J. Hartmanis and L. Hemchandra. One-way functions and the non-isomorphism of NP-complete sets. *Theoretical Computer Science*, 81(1):155–163, 1991.

[IL95] N. Immerman and S. Landau. The complexity of iterated multiplication. *Information and Computation*, 116:103–116, 1995.

[Joh] D. Johnson. Ongoing column on NP-complete sets. Journal of Algorithms.

[JY85] D. Joseph and P. Young. Some remarks on witness functions for nonpolynomial and noncomplete sets in NP. *Theoretical Computer Science*, 39:225–237, 1985.

[Ko83] K. Ko. On self-reducbility and weak p-selectivity. *Journal of Computer and System Sciences*, 26:209–221, 1983.

[Ko85] K. Ko. On some natural complete operators. *Theoretical Computer Science*, 37:1–30, 1985.

[PY86] C. H. Papadimitriou and M. Yannakakis. A note on succinct representation of graphs. *Information and Computation*, 71:151–158, 1986.

[Sch88] U. Schöning. Graph isomorphism is in the low hierarchy. *Journal of Computer and System Sciences*, 37:312–323, 1988.

[Sel88] A. Selman. Natural self-reducible sets. *SIAM Journal on Computing*, 17:989–996, 1988.

[Sim77] J. Simon. On the difference between the one and the many. In *Proceedings of the International Colloquium on Automata, Languages and Programming*, pages 480–491. Springer LNCS 52, 1977.

[Val82] L. G. Valiant. Reducibility by algebraic projections. *L'Enseignement mathématique*, 28, 3-4:253–268, 1982.