

1400008559
còpia 1

**Self-reducible sets
of small density**

A. Lozano
J. Torán

Report LSI-89-27



Abstract

We study the complexity of sets that are at the same time self-reducible and sparse or m -reducible to sparse sets. We show that sets of this kind are low for the complexity classes Δ_2^P , Θ_2^P , NP or P, depending on the type of self-reducibility used and on certain restrictions imposed on the query mechanism of the self-reducibility machines. The proof of some of these results is based on graph theoretic properties that hold for the graphs induced by the self-reducibility structures.

Resum

S'estudia la complexitat dels conjunts auto-reduïbles que són, alhora, esparsos o m -reduïbles a esparsos. Aquests conjunts esdevenen baixos per a les classes de complexitat: Δ_2^P , Θ_2^P , NP o P, depenent del tipus d'auto-reduïbilitat i de certes restriccions imposades al mecanisme de preguntes de les màquines de l'auto-reduïbilitat. La demostració d'alguns d'aquests resultats es basa en propietats teòriques que satisfan els grafs induïts per les estructures d'auto-reduïbilitat.

Resumen

Se estudia la complejidad de los conjuntos auto-reducibles que son, además, esparsos o m -reducibles a esparsos. Estos conjuntos resultan ser bajos para las clases de complejidad: Δ_2^P , Θ_2^P , NP o P, en función del tipo de auto-reducibilidad y de ciertas restricciones impuestas al mecanismo de preguntas de las máquinas de la auto-reducibilidad. La demostración de algunos de estos resultados se basa en propiedades teóricas que cumplen los grafos inducidos por las estructuras de auto-reducibilidad.

Self-reducible sets of small density

(preliminary version, November 1989)

Antonio Lozano and Jacobo Torán
Department L.S.I.
U. Politècnica de Catalunya
Pau Gargallo 5
08028 Barcelona
Spain

Abstract

We study the complexity of sets that are at the same time self-reducible and sparse or m -reducible to sparse sets. We show that sets of this kind are low for the complexity classes Δ_2^P , Θ_2^P , NP or P, depending on the type of self-reducibility used and on certain restrictions imposed on the query mechanism of the self-reducibility machines. The proof of some of these results is based on graph theoretic properties that hold for the graphs induced by the self-reducibility structures.

1. Introduction

The fact that many important sets are self-reducible, i.e. have an internal structure that allows to reduce the decision problem for words in the set to the decision problems for smaller words, has been extensively studied, and this property has played a central role in the proof of important results; for example, in the proof of the theorem that states that there are no sparse NP-complete sets unless $P=NP$ [Ma 82], the existence of NP-complete self-reducible sets is crucial.

In this article we study the question of whether sets with small density can have enough internal structure to be non-trivially self-reducible, and if so, what is the complexity of such sets. We observe that there are sparse (and sparse related) self-reducible sets, and we show that depending on the self-reducibility type used, these sets are low for the complexity classes Δ_2^P , Θ_2^P , NP and P.

We work with two definitions of polynomial time self-reducibility. In the first place we consider in section 3 the definition from [Me,Pa 79], a generalization of the usual *length decreasing* self-reducibility. Basically, a set is length decreasing self-reducible if the decision problem for words in the set can be reduced to decision problems for words of smaller length. We study in section 4 the case of *word decreasing* self-reducibility [Ba 89]. In this case the self-reducing machine can query words that are just smaller than the input in lexicographical order, and therefore the self-reducibility structure can contain exponentially long decreasing chains.

There are sets that are at the same time sparse and self-reducible, for example, for every polynomial time computable set A an encoding of the set of prefixes of the minimum word in A for each length is sparse and word decreasing self-reducible. However, we show that this kind of sets cannot be too complex. It holds that every sparse self-reducible set (for both definitions) is low [Sc 83] for the class Δ_2^P , i.e., it does not help a Δ_2^P machine when used as oracle.

We are interested in the fact of whether the adaptive queries to NP from the algorithm giving the above upper bound could be made non adaptive, or in other words, whether the sparse self-reducible sets are low for Θ_2^P , the class of problems computed by a deterministic polynomial time machine asking non-adaptive queries to an NP oracle. It is not hard to see that all the sets that were previously known to be low for Δ_2^P (sparse sets in NP, almost polynomial time sets in NP, [Ko,Sc 85]) or low for Δ_3^P (co-sparse sets in NP [Ko,Sc 85], P-close sets in NP [Sc 85]) are also low for the respective classes Θ_2^P and Θ_3^P . The sparse self-reducible sets are the first example of sets

This research was partially supported by ESPRIT-II Basic Research Actions Program of the EC under contract No. 3075 (project ALCOM).

that are low for Δ_2^p and do not seem to be low for Θ_2^p . Moreover, in section 4 we give a sparse word decreasing self-reducible set and a relativization under which the set is not low for Θ_2^p , showing therefore that under relativization the above upper bound is optimal.

We study also the complexity of sparse self-reducible sets whose self-reducibility machine has a specific type of query mechanism: disjunctive queries, conjunctive queries or just one query, which we will respectively denote d -, c - and 1-self-reducible. These restrictions have been frequently used; for example, the set *SAT* is d -self-reducible, and all Near-Testable sets, [Go,Jo,Yo 87], are word decreasing 1-self-reducible.

For the case of length decreasing self-reducibility, using previous results it can be seen that sparse d -self-reducible sets are in NP and low for Θ_2^p , while c -self-reducible and 1-self-reducible sets are in P.

For the word decreasing self-reducibility case if only sparse sets are considered, all the results from section 3 trivially hold; but for this kind of sets the restrictions in the query mechanism from the self-reducibility machine do not make much sense since, as we will see, under these restrictions we can consider polynomially long decreasing chains in the self-reducibility structure to decide membership. The definition of word decreasing self-reducibility then loses its main characteristic. Therefore we will consider also sets that are many-one reducible to sparse sets, which on one side have some of the characteristics of sparse sets, and on the other allow to show non trivial results.

Restricting ourselves to sets that are many one reducible to sparse sets and word decreasing self-reducible, we show that for the disjunctive case, the sets are also NP and low for Θ_2^p . For the conjunctive and one-query case, the second definition of self-reducibility seems to behave in a different way than the former one; we give as upper bound for the complexity of these sets the class $NP \cap co-NP$. These last results are proved considering certain graph theoretic properties that hold in the graphs induced by the self-reducibility structures.

A summary of the results obtained in the paper can be seen in table 1, in the section of conclusions.

2. Basic definitions and examples

The notation used in this paper is the standard one, following the basic books in the area, as [Ba,Di,Ga 88].

Let us first define the types of self-reducibility that will be used.

Definition 1: [Ma,Pa 79], [Ko 83]. We will say that a partial order \prec on Σ^* is polynomially well-founded and length-related (abbr. polynomially related) if there is a polynomial p such that

- (i) For all $x, y \in \Sigma^*$, if $x \prec y$ then $|x| \leq p(|y|)$.
- (ii) It is decidable in polynomial time (in $|x| + |y|$) whether $x \prec y$.
- (iii) Every decreasing chain is bounded in length by p of the length of its maximum element.

Definition 2: [Ma,Pa 79], [Ko 83]. A language $A \subseteq \Sigma^*$ is polynomial time length decreasing self-reducible (ld-self-reducible) (resp. d -, c - or 1-ld-self-reducible) if there is a deterministic polynomial time oracle Turing machine M (resp. a Turing machine with a disjunctive or conjunctive query mechanism, or a machine which on every input asks just one query) and a polynomially related ordering \prec such that $A = L(M, A)$ and, on any input x , M queries the oracle only about words strictly smaller than x in the partial order. For a more formal definition we refer to [Ko 83].

The set of satisfiable boolean formulas *SAT* is an example of a disjunctive self-reducible set, while the set $\#SAT = \{\langle F, k \rangle : F \text{ is a boolean formula with at least } k \text{ satisfying assignments}\}$ is an example of a self-reducible set with a more complicated query mechanism [Ba,Bo,Sc 86].

We will also consider a type of self-reducibility based on the lexicographical order in Σ^* , where we allow exponentially long self-reducing chains to exist.

Definition 3: [Ba 89], A language $A \subseteq \Sigma^*$ is polynomial time word decreasing self-reducible (wd-self-reducible) if there is a polynomial time deterministic oracle Turing machine M such that

$A = L(M, A)$ and on any input x , M queries the oracle only about words of length less than $|x|$, or about words of length $|x|$ and smaller than x in lexicographical order. Disjunctive, conjunctive or 1-query word decreasing self-reducible sets (d -wd, c -wd, 1-wd-self-red.) are defined analogously.

If $M_1, M_2 \dots$ is an enumeration of deterministic Turing machines, an appropriate encoding of the set $A = \{(i, j, k, 0^n) : M_j \text{ starting at configuration } k \text{ reaches an accepting state using less than } i \text{ steps and less than } n \text{ tape cells}\}$ is an example of a PSPACE complete and wd-self-reducible set. Observe that A is disjunctive, as well as conjunctive and as 1-query wd-self-reducible.

Let us consider another interesting set, which is at the same time self-reducible and sparse.

Let $A \subseteq \Sigma^*$ be a set in P. Consider the sparse set consisting on the prefixes of the minimum word in A for each length,

$$\text{Min-Pref}(A) = \{0^n \# x : x \text{ is a prefix of the minimum word of length } n \text{ in } A\}$$

Proposition 4: There is an appropriate encoding of the set $\text{Min-Pref}(A)$ such that it is wd-self-reducible.

Proof: Consider the following self-reducibility structure, where by predecessor of a word we mean predecessor in lexicographical order.

if $|x| < n$ then

$$0^n \# x \in \text{Pref-Min}(A) \iff x0 \in \text{Pref-Min}(A) \text{ or } x1 \in \text{Pref-Min}(A)$$

if $|x| = n$ then

$$0^n \# x \in \text{Pref-Min}(A) \iff x \in A \text{ and the predecessor of every prefix of } x \text{ ending with a 1 is not in } \text{Pref-Min}(A)$$

In order to see whether a prefix x is in the set, we ask in the usual way if either $x0$ or $x1$ is a prefix, until we reach a word y of length n ; then we have to check that the word y is in A and that it is the minimum one of the same length in lexicographical order in the set. The second fact is true if and only if there is no word in A with a prefix that is smaller than the prefix of that length of y , and this holds exactly when none of the predecessor of a prefix from y ending with a 1 is a prefix of a word in A , i.e., when none of the predecessors of a prefix from y ending with a 1 belongs to $\text{Pref-Min}(A)$.

We can encode the set in such a way that always smaller words in lexicographical order are queried, for example writing a string of $n - |x|$ 1's at the end of a prefix of a word of length n , followed by a string of $\lceil \log n \rceil$ bits encoding the number $n - |x|$. Therefore the encoded set is wd-self-reducible. \square

Another concept that will be used in the paper is the one of lowness [Sc 83].

Definition 5: We will say that a set $A \subseteq \Sigma^*$ is low for a class C in the polynomial time hierarchy, if $C(A) \subseteq C$. C -low represents the class of all sets that are low for C .

In particular if a set is low for a class, then the set belongs to the class. Observe that the definition is a little different from the one in [Sc 83] since we do not restrict ourselves to low sets in NP. We will consider also low sets for Θ_2^P , the class of sets polynomial time truth-table reducible to sets in NP. The low sets for Θ_2^P are defined in the same way as the low sets for the classes in PH.

3. Length decreasing self-reducibility

We start giving an upper bound for the class of sparse ld-self-reducible sets.

Theorem 6: If $S \subseteq \Sigma^*$ is sparse and ld-self-reducible then S is low for the class Δ_2^P .

Proof: Let M be the self-reducibility machine for S and \prec the polynomially related partial order from the self-reducibility. We will say that there is a path of length k between two words x, y if there is a sequence of words x_1, \dots, x_{k-1} satisfying $x \succ x_1 \succ \dots \succ x_{k-1} \succ y$. Observe that a path between two words x and y can be guessed existentially.

Let p be a polynomial bounding the ordering defining the self-reducibility structure. We will describe an algorithm which, given input 0^n , produces all the words in S of length n . The idea is to obtain first all the words of length $\leq p(n)$ whose inclusion in S can be decided without querying smaller words in S . With this set of words we can obtain the words in S that only query words that are already obtained or words that are not in S , and so on. The problem is that we can decide if a word is already obtained, but how can we decide if a word which has not been obtained at a certain stage does not belong to S or will be eventually included? To solve this problem we will explore the set of words by levels, depending on the length of a path from an element of length n to the word. First we will obtain all the words y whose inclusion in S can be decided without querying any other word, and for which there is a word x of length n and a path $x \succ x_1 \succ x_2 \succ \dots \succ x_{p(n)-1} \succ y$, of length exactly $p(n)$. With the list of obtained words of level $p(n)$ we can now get the words of the previous level (words with a path of length exactly $p(n) - 1$ from a word of length n), since if a word of level $p(n) - 1$ queries some word this word either belongs to level $p(n)$ and therefore it is in our list, or it does not belong to S .

The process of obtaining the words in S of a certain level (knowing the words from the levels underneath) is just a binary search of logarithmic many queries to an oracle in NP to obtain first the census of the words belonging to S in the level, and then polynomially many non adaptive queries to NP to obtain the new words and include them in our list. The size of the list remains always polynomial by the sparseness of S . The process has to be repeated a polynomial number of times, until we obtain the words of level 1 which are exactly the words of length n .

We give a more formal description of the algorithm; it calls procedure *obtain-words*, which, given the list of words of a certain level, computes the list of words of the previous one.

```

input  $0^n$ 
 $S := \emptyset$ ;  $level := p(n)$ 
repeat
  obtain-words( $S, level$ )
   $level := level - 1$ 
until  $level = 0$ 
output words in  $S$  of length  $n$ 

procedure obtain-words( $S, level$ )
  binary search for  $\max \{k : \langle S, level, k \rangle \in O_1\}$ , [census of  $S$  at  $level - 1$ ]
  for  $i:=1$  to  $k$  do
    binary search for  $\max \{l_i : \langle S, level, k, i, l_i \rangle \in O_2\}$  [length of the  $i$ -th word in  $S$  of  $level - 1$ ]
  endfor
  for  $i:=1$  to  $k$  do
     $w_i := \lambda$ ;  $j := 1$ 
    repeat
      if  $\langle S, level, k, i, j \rangle \in O_3$  then
         $w_i := w_i 1$ 
      else
         $w_i := w_i 0$ ; [bitwise obtain all the words of  $S$  at  $level - 1$ ]
      endif
       $j := j + 1$ 
    until  $j = l_i + 1$ 
  endfor
   $S := S \cup \{w_1, \dots, w_k\}$ 
endprocedure

```

We have to explain what the sets O_1 , O_2 and O_3 are, and prove that they are in NP. Since they are all defined similarly we only explain in detail set O_1 .

O_1 is the set used to obtain the number of new words in S in each level; it is defined as

$$\begin{aligned}
O_1 = & \{ \langle S, t, k \rangle : \text{there are at least } k \text{ words that are not in } S \\
& \text{which are accepted by the self-reducibility machine } M \text{ with oracle } S, \\
& \text{and there is a path of length } t \text{ from a word of length } n \text{ to each one of the words} \} \\
= & \{ \langle S, t, k \rangle : \exists y_1, \dots, y_k, (y_i \neq y_j) \text{ s.t. } y_i \notin S, \text{ and } y_i \in L(M, S), (i = 1, \dots, k) \\
& \text{and } \exists x_{1,1}, \dots, x_{1,t-1}, \dots, x_{k,1}, \dots, x_{k,t-1}, \text{ s.t. } |x_{i,1}| = n \\
& \text{and } x_{i,1} \succ x_{i,2} \dots \succ x_{i,t-1} \succ y_i (i = 1, \dots, k) \}
\end{aligned}$$

O_2 is used to obtain the length of the new words in S in each level.

$$O_2 = \{ \langle S, t, k, i, l \rangle : \langle S, t, k \rangle \in O_1, \text{ and the } i\text{-th new word has length } \geq l \}$$

With O_3 we can bitwise obtain the new words in S .

$$O_3 = \{ \langle S, t, k, i, j \rangle : \langle S, t, k \rangle \in O_1, \text{ and the } i\text{-th new word has a 1 in position } j \}$$

It is clear that the three sets are in NP. We can use as oracle the marked union of them. Our algorithm runs in polynomial time having access to an NP oracle. On input 0^n it prints all the words of length n in S . Set A is low for the class Δ_2^P since a Δ_2^P machine M having A as oracle can be simulated by another one that first prints all the words in A up to a certain length (using the algorithm given above), and then simulates M substituting queries to A by queries to the list of printed words. \square

We study now the complexity of sparse self-reducible sets considering certain restrictions in the query mechanism of the machine.

Observation 7: If S is sparse and disjunctive ld-self-reducible then S is in $\text{NP} \cap \Theta_2^P\text{-low}$.

Disjunctive ld-self-reducible sets are in NP [Ko 83]. Following the same technique as in [Ka 88], it can be seen that every sparse set in NP is low for the class Θ_2^P .

Observation 8: If S is sparse and conjunctive ld-self-reducible then S is in P.

This follows directly from the proof of the result stating that if SAT is many-one reducible to a co-sparse set then $\text{P}=\text{NP}$ [Fo 79].

Observation 9: If S is sparse and 1-ld-self-reducible then S is in P.

In fact every 1-ld-self-reducible set is in P, since in order to decide membership of an element one only needs to follow polynomially many steps in the unique chain of words produced by the self-reducibility machine. In next section, when we consider word decreasing self-reducibility, we will see that the restriction of querying just one word becomes non trivial.

4. Word decreasing self-reducibility

In this section we deal with polynomial time word decreasing self-reducible sets (Definition 3). The self-reducibility machine for these sets can query words that just need to be lexicographically smaller than the input, and there can be exponentially long decreasing chains. If we study wd-self-reducible sparse sets, most of the restrictions in the query mechanism considered in the previous section do not make much sense. One can see that in order to decide membership in a sparse disjunctive, conjunctive or 1-query wd-self-reducible set, the decreasing chains induced by the self-reducibility that have to be considered can have only polynomial length, and therefore the main characteristic of the new definition of self-reducibility is spoiled. Therefore we will consider in this section the class of sets many-one reducible to sparse sets, which have the flavour of the sparse sets but permit more interesting results. Let us first start giving an upper bound for the general case.

wd-self-reducible sets that are m -reducible to sparse sets are low for Δ_2^p , as in the length decreasing self-reducibility case. Now in the proof of the result we cannot use the condition that decreasing chains in the partial order are polynomially long to divide the words in the self-reducible set into levels, but we can make use of the lexicographical order to obtain these words stage by stage.

Theorem 10: If A is wd-self-reducible and there is a sparse set S such that A is polynomial time m -reducible to S then A is low for the class Δ_2^p .

Proof: Let M be the self-reducibility machine for A , and let f the function reducing A to S , being the computation time of f bounded by a polynomial p . We describe an algorithm which, on input 0^n , produces $f(A_{\leq n})$, i.e. the list of words in S that are images of elements of length $\leq n$ in A .

At the beginning, the algorithm executes binary search in the oracle to obtain the image of the smallest word x_1 that is accepted by the self-reducibility machine querying the empty set, and includes the word $s_1 = f(x_1)$ in a list of obtained images S' . Then, it obtains the smallest word x_2 whose image is not in S' and is accepted by machine M . This last point can be checked using a deterministic polynomial time machine M' which on input $\langle S', x \rangle$ decides if x is accepted by the self-reducibility machine M in the following way: M' simulates M on input x and each time M produces a query y to A ($y < x$), M' computes $f(y)$ and answers the query positively iff $f(y) \in S'$.

Since x_2 is the minimum word greater than x_1 such that $f(x_2)$ is not in S' , no word between x_1 and x_2 can be in A and have a new image that is not in S' , it follows that M on input x_2 can only query words which either are not in A or whose image is in S' . Therefore M' simulates M correctly.

The minimum word in A whose image is not in S' can be obtained doing binary search in the oracle

$$Oracle = \{ \langle S', y, 0^n \rangle : |y| \leq n \text{ and } \exists x x \leq y \text{ and } f(x) \notin S' \text{ and } \langle S', x \rangle \in L(M') \}$$

which is clearly in NP.

The algorithm repeats the described process until the images of all the words in A of length $\leq n$ are obtained. Since there is only a polynomial number of them, the algorithm runs in polynomial time.

A is then low for the class Δ_2^p since any Δ_2^p algorithm using set A as oracle can be simulated by an algorithm that uses the above procedure to obtain the set S' of images of A , and then substitutes queries to A by queries to S' . \square

As in the case of length decreasing self-reducibility from the previous section, it does not seem possible to compute the self-reducible set doing only non adaptive queries to an NP oracle since at each stage we need the previously obtained words to obtain the new ones. We exhibit a relativization showing that there are sparse wd-self-reducible sets that are not low for the class Θ_2^p . The definition of self-reducibility relative to an oracle B is the same as in the non relativized case except that the self-reducibility machine for a set A can either query A for words smaller than the input or query oracle B with no restriction. Examples of relativizations showing that certain sets are optimally classified in low classes have been previously obtained in [Al,He 89].

Theorem 11: There is an oracle B and a sparse set S such that S is wd-self-reducible but is not low for the class Θ_2^p , relative to B .

Proof: We will make use of the following fact, whose proof can be seen in the appendix.

Fact: There is a set B relative to which the test language $L_B = \{0^n : B = \emptyset \text{ or the minimum word of length } n \text{ in } B \text{ is even} \}$ is in Δ_2^p but not in Θ_2^p .

Let B be the set from the above fact, and consider the language

$$Pref-Min(B) = \{0^n \# x : x \text{ is a prefix of the minimum word of length } n \text{ in } B\}$$

It is clear that $Pref-Min(B)$ is sparse and, by the same argument as in the example from section 2, an appropriate encoding of $Pref-Min(B)$ is wd-self-reducible relative to B .

Observe that the language L_B is in $P(\text{Pref-Min}(B))$ since in order to see whether a certain word 0^n is in L_B , we can query the set $\text{Pref-Min}(B)$ to obtain the minimum word of length n in B , and then check if the word is even or odd. Therefore $\text{Pref-Min}(B)$ cannot be low for Θ_2^P since this would imply that L_B is low for Θ_2^P (the low classes are closed under polynomial T -reducibility), contradicting the fact that L_B is not in Θ_2^P relative to B . \square

In the previous section we have seen that, for the case of ld-self-reducibility, the concept of path between words in the partial order is a useful one. Since these paths always have polynomial length, polynomial time machines can go through them and reach words that do not need to query smaller words.

In the case of wd-self-reducibility, the decreasing chains can be exponentially long and therefore we need some new idea to be able to go through them in polynomial time. We will take advantage of the fact that the self-reducible set is reducible to a sparse set. This fact allows to find some shortcuts in the decreasing chains by going through the sparse set, obtaining polynomial length paths, as will be shown in the following lemma.

Let us first define the concept of PB-graph, (path-bipartite graph) which is just a bipartite graph in which the nodes of one side are all connected in a path and are connected with exactly one node of the other side. Later we will consider the case in which the nodes of one side of the graph are the words of the self-reducible set, and the ones in the other side are the words from the sparse set.

Definition 12: An undirected graph $G = (V, E)$ with $V = \{u_1, \dots, u_m, v_1, \dots, v_n\}$ is said to be a PB-graph if there exists an application $f: \{1, \dots, m\} \rightarrow \{1, \dots, n\}$ such that the set of edges of G is defined by

$$E = \{(u_i, u_{i+1}) : 1 \leq i < m\} \cup \{(u_i, v_j) : f(i) = j\}$$

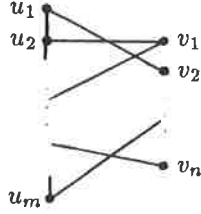


Fig. 1, a PB-graph.

Lemma 13: For every PB-graph $G = (V, E)$ with $V = \{u_1, \dots, u_m, v_1, \dots, v_n\}$ there exists a path from node u_1 to node u_m of length at most $3n - 1$ (independently of m).

Proof: Let $G = (V, E)$ be a PB-graph. We prove the lemma by induction on n .

For the induction base, if $n = 1$, we have the path $u_1 \rightarrow v_1 \rightarrow u_m$, of length 2. For the induction step, let i be the maximum value for which $f(u_i) = f(u_1)$. If we delete in G the nodes u_1 to u_i and the node $f(u_1)$, it holds, by induction hypothesis, that there is a path c from u_{i+1} to u_m of length at most $3(n - 1) - 1$. Therefore, we have a new path from u_1 to u_m that starts with $u_1 \rightarrow f(u_1) \rightarrow u_i \rightarrow u_{i+1}$ and then connects with c , resulting in a new path of total length at most $3n - 1$. \square

In order to use the above lemma in the paths induced by the self-reducibility structures, we introduce the following definition.

Definition 14: Let A and B be two sets in Σ^* , being A self-reducible, and $A \leq_m B$ via the polynomial time function f . We will write $x_1 \rightarrow x_2$ if the self-reducibility machine on input x_1 queries x_2 . We will say that a sequence of elements x_1, x_2, \dots, x_k is a valid path from x_1 to x_k with respect to A, B if for all $i, 1 < i \leq k$ it holds

- (i) $x_{i-1} \rightarrow x_i$ or
- (ii) $x_{i-1} \in A \wedge f(x_{i-1}) = x_i$, or
- (iii) $x_{i-1} \in B \wedge f(x_i) = x_{i-1}$

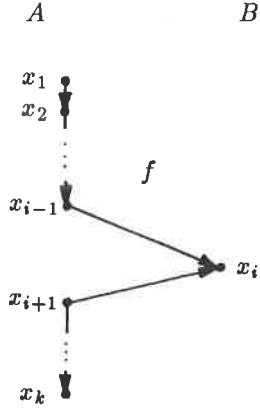


Fig. 2, a valid path.

Observe that for the case of disjunctive, conjunctive and 1-query wd-self-reducibility, one can decide which are the words queried by the self-reducibility machine on a certain input. Instead of considering the partial order over Σ^* from the self-reducibility, we will consider from now on the directed acyclic graph that results from generating the self-reduction queries starting at a certain string x . We will call such graph the graph induced by M on input x . The nodes of this graph that have no descendants are the words that M accepts or rejects directly, without making any query. Such words will be called base elements.

A valid path is therefore a sequence of elements where every two consecutive ones are connected in the graph induced by the self-reducibility machine of A or are related by the reducing function f . The following observation indicates that some properties about polynomially long paths (used in the case of length decreasing self-reducibility) are preserved when we deal with valid paths. Part (c) of the observation uses the concept of *number of sign alternations*. This applies to the case of 1-self-reducible sets, where the induced graph generated from a string x is just a path. If A is 1-self-reducible, $x \in \Sigma^*$, and c is a path generated at x , the number of sign alternations in c , $sign-alt(c)$ is the number of elements in path c that belong to A if and only if its successor in the path does not belong to the set.

Observation 15:

- (a) If A is disjunctive wd-self-reducible, $A \leq_m B$ and there exists a valid path from x to y w.r.t. A, B , then if y is in A , x is in A .
- (b) If A is conjunctive wd-self-reducible, $A \leq_m B$ and there exists a valid path from x to y w.r.t. A, B , then if y is not in A , x is not in A .
- (c) If A is 1-wd-self-reducible, $A \leq_m B$ and there exists a valid path c from x to y w.r.t. A, B , then x is in A if and only if it holds ($y \in A \Leftrightarrow sign-alt(c)$ is even).

We can now proceed to prove the upper bounds for word decreasing self-reducible sets that are m -reducible to sparse sets.

Theorem 16: If A is disjunctive wd-self-reducible and there is a sparse set S such that A is m -reducible to S then A is in $NP \cap \Theta_2^p$ -low.

Proof: The proof is based on part (a) of the above observation. Let M be the self-reducibility machine for A and let S be the sparse set such that $A \leq_m S$, via function f .

Consider the graph induced by M on input x , and the NP machine M' defined by the following algorithm:

```

input  $x$ 
guess  $y$ 
guess  $c$ 
if  $y$  is a base element in  $A$ 
and  $c$  is a valid path from  $x$  to  $y$  w.r.t.  $A, S$  then
  ACCEPT
else
  REJECT
endif

```

If x is accepted by M' then there is a valid path from x to a word y in A ; by observation (a), x is in A .

On the other hand, if x is in A then, since the self-reduction is disjunctive, there is a path in the induced graph from x to a base element y that belongs to A . All the elements along the path that connects x with y are in A . Thus, the images by f of these elements are in S .

Consider now the graph whose vertices are the elements of the path from x to y , say x_1, \dots, x_k , plus the images of these elements by f : $f(x_1), \dots, f(x_k)$, and its edges are $\{(x_i, x_{i+1}) : 1 \leq i \leq k\} \cup \{(x_i, f(x_i)) : 1 \leq i \leq k\}$. This forms a PB-graph, and by lemma 13, there is a path from x to y whose length only depends on the number of different images of the elements in the path, which is polynomial in $|x|$. Moreover, this path is a valid path from x to y , and therefore M' on input x can guess the path and accept x . We have shown that A is in NP. Again, using the same techniques as in [Ka 87] it can be seen that every NP set which is m -reducible to a sparse set is low for Θ_2^p , and the theorem follows. \square

The upper bound $\text{NP} \cap \Theta_2^p$ -low seems the best one for the case of disjunctive wd-self-reducible sets, since there are relativizations showing that sets of this kind cannot be in co-NP. From the results in [Ba,Gi,So 75] we know that there is a sparse set B such that the language $L_B = \{0^n : B_n \neq \emptyset\}$ is not in co-NP relative to B . The set of prefixes of B is disjunctive self-reducible (even length decreasing) and sparse. Also, it is not in co-NP relative to B since in order to decide $0^n \in L_B$ it suffices to decide whether the empty word is in the set of prefixes of words of length n .

Next we will consider conjunctive wd-self-reducible sets. We show that the sets that are at the same time m -reducible to sparse sets are in $\text{NP} \cap \text{co-NP}$ and therefore low for NP, [Sc 83].

Theorem 17: If A is conjunctive wd-self-reducible and there is a sparse set S such that A is m -reducible to S then A is in NP.

Proof: Let M be the self-reducibility machine for A and let S be the sparse set such that $A \leq_m S$, via function f .

Let N be an NP machine that, on input $\langle x, y \rangle$, simulates machine M on input x , and when M makes query q to the oracle, N checks whether $f(q)$ is in the set encoded by y .

The algorithm is similar to the one in theorem 10, with the difference that now we do not search the minimum word, but we guess at each step some word that is accepted by M using S' as oracle, and whose image is not in the S' , being S' the sparse set constructed so far. Consider machine M' described by the following program:

```

input  $x$ 
 $S' := \emptyset; y := 0$ 
while  $y < x$  do
  guess  $y'$  ( $y' \geq y$ )
  if  $f(y') \notin S' \wedge N(\langle y', S' \rangle)$  accepts then
     $S' := S' \cup \{f(y')\}$ 
  else
    REJECT
  endif
   $y := y'$ 
endwhile

```

```

if  $f(x) \in S'$  then
  ACCEPT
else
  REJECT
endif

```

Using the fact that the self-reduction is conjunctive it is not difficult to see that set S' constructed by M' is a subset of S , i.e., M' does not make mistakes deciding whether a word is in A . Therefore, if M' accepts x then x is in A .

On the other hand, if $x \in A$, M' only needs to guess, at each step of the loop, the minimum word y such that $f(y) \notin S'$ and $N(\langle y, S' \rangle)$ accepts. Since at most a polynomial number of elements in S are needed to decide x , and each time the machine goes through the loop a new element from S is added to the list, M' will accept x in polynomial time. \square

Theorem 18: If A is conjunctive wd-self-reducible and there is a sparse set S such that A is m -reducible to S then A is in co-NP.

Proof: Let M be the self-reducibility machine for A and let S be the sparse set such that $A \leq_m S$, via function f . Let p and q be the polynomials that bound, respectively, the density of S and the size of f .

We give an NP algorithm accepting \overline{A} . Given an input string x , it holds that $x \in \overline{A}$ iff there is a path in the graph induced by M on input x to a base element in \overline{A} . There are two possibilities: either there are at most $p(q(|x|))$ different values that are images via f of elements in the induced graph starting at x , or there are more than $p(q(|x|))$ such values.

In the first case, applying lemma 13, $x \in \overline{A}$ iff there is a polynomially long valid path from x to an element y in the base that is not in A . If this is the case, the path can be guessed by our algorithm, and it can accept x .

In the second case, (more than $p(q(|x|))$ descendants from x with different images) we can already conclude that $x \notin A$, since we are dealing with conjunctive self-reducibility and there must be a word w queried in the induced graph starting at x whose image is not in S (implying that $w \notin A$ and therefore $x \notin A$). To check that we actually are in the second case the algorithm can guess $p(q(|x|)) + 1$ elements with different image checking that all of them are descendants of x . For this last condition, the algorithm uses lemma 13 to obtain a valid path (of length at most $p(q(|x|))$) from x to each one of the guessed elements.

The algorithm is given in more detail in the following program, simulating a nondeterministic machine M' ; at the beginning, M' guesses a boolean variable b , and depending on the value of b it performs the algorithm for one of the two possible cases explained above.

```

input  $x$ 
 $n := q(|x|)$ 
guess  $b \in \{0, 1\}$ 
if  $b = 0$  then
  guess  $y$ 
  guess  $c$ 
  if  $c$  is a valid path from  $x$  to  $y$  and  $y$  is an element
  in the base and  $y \notin A$  then
    ACCEPT
  else
    REJECT
endif

```

```

else
  guess  $x_1, \dots, x_{p(q(n))+1}$ 
  guess  $c_1, \dots, c_{p(q(n))+1}$ 
  if for all  $i, j, 1 \leq i, j \leq p(n), f(x_i) \neq f(x_j)$ 
    and for all  $i, 1 \leq i \leq p(n), c_i$  is a valid path from  $x$  to  $x_i$  then
      ACCEPT
    else
      REJECT
  endif
endif

```

□

Finally we consider the class of 1-wd-self-reducible sets. It is not hard to see that this class contains the class of Near-Testable sets [Go,Jo,Yo 88]. Therefore from our next theorem it follows that every near testable set which is m -reducible to a sparse set is in the class $\text{NP} \cap \text{co-NP}$.

Theorem 19: If A is 1-wd-self-reducible and there is a sparse set S such that A is m -reducible to S then A is in $\text{NP} \cap \text{co-NP}$.

Proof: As before, let M be the self-reducibility machine for A and let S be the sparse set such that $A \leq_m S$, via function f . Let p and q be the polynomials that bound, respectively, the density of S and the size of f .

Given an element x of length n , consider the path formed by the queries succesively made by M on input x . This path (which is the induced graph) starts at x and ends at a base element y .

In the path, there are either at most $2p(q(n))$ elements with different image values via f , or there are more than this number. In the first case lemma 13 shows that there is a valid path of polynomial length from x to y . Thus, knowing the membership of y in A and the parity of the number of sign alternations in the path we can derive the membership of x in A , following part (c) of observation 15.

In the second case, when there are more than $2p(q(n))$ different images, consider the $2p(q(n))+1$ elements in the path which are closer to x and have different images. Among these there must at least $p(q(n))+1$ of them with the same parity in the number of sign alternations in their valid paths from x (i.e., all of them have paths with an odd number of alternations, or with an even number of them). Since they all have the same parity of sign alternations with respect to x , either they are all in A or none of them is in A . But we know that at least one, say z , is not in A (because in S there can be only $p(q(n))$ different images for words in our path). This implies that every one of the $p(q(n))+1$ elements that has the same parity of sign alternation as z does not belong to A .

To decide x , our algorithm has to find $p(q(|x|))+1$ elements for which there is a valid path from x with the same parity of sign alternations for all of them; these elements are not in A , and the algorithm accepts iff the number of alternations in the path from x to any of them is odd.

The algorithm deciding A is given below. As in the previous one it first guesses a variable b , deciding which one of the cases it simulates.

```

input  $x$ 
 $n := q(|x|)$ 
guess  $b \in \{0, 1\}$ 
if  $b = 0$  then
  guess  $y$ 
  guess  $c$ 
  if  $c$  is a valid path from  $x$  to  $y$  and  $y$  is an
  element in the base not in  $A$  then
    if ( $\text{sign-alt}(c)$  is even and  $y \in A$ ) or ( $\text{sign-alt}(c)$  is odd and  $y \notin A$ ) then
      ACCEPT

```

```

else
  REJECT
endif
else
  guess  $x_1, \dots, x_{p(n)+1}$ 
  guess  $c_1, \dots, c_{p(n)+1}$ 
  if for all  $i, j, 1 \leq i, j \leq p(n), f(x_i) \neq f(x_j)$ 
  and for all  $i, 1 \leq i \leq p(n), c_i$  is a valid path from  $x$  to  $x_i$ 
  and for all  $i, 1 \leq i \leq p(n), \text{sign-alt}(c_i)$  is odd then
    ACCEPT
  else
    REJECT
  endif
endif

```

Observe that the nondeterministic algorithm for \overline{A} is completely symmetrical, and to obtain it we only have to interchange the words “odd” and “even”. Therefore A is in the class $\text{NP} \cap \text{co-NP}$. \square

5. Conclusions and open problems

We have studied the complexity of sets that are sparse (or many-one reducible to sparse sets) and self-reducible. We have considered two types of self-reducibility, length decreasing and word decreasing, and also different restrictions in the query mechanism of the self-reducibility machine: disjunctive, conjunctive and one-query. The results obtained can be seen in the following table:

Restriction	sparse sets		sets m-reducible to sparse sets	
	Length decreasing self-red.	Word decreasing self-red.	Length decreasing self-red.	Word decreasing self-red.
	Low in the class	Low in the class	Low in the class	Low in the class
None	Δ_2^P	Δ_2^P	Δ_2^P	Δ_2^P
Disjunctive	Θ_2^P	Θ_2^P	Θ_2^P	Θ_2^P
Conjunctive	P	P	P	NP
1-query	P	P	P	NP

Table 1: Complexity of self-reducible sets.

Sparse self-reducible sets are the first class of sets that are low for the class Δ_2^P and do not seem to be low for Θ_2^P since, as shown in section 4, there is relativization under which there exist wd-self-reducible sparse sets which are not low for Θ_2^P .

As shown in the table, for the case of sets reducible to sparse sets which are conjunctive or 1-wd-self-reducible, the best upper bound we could find for their complexity is $\text{NP} \cap \text{co-NP}$. It is an open question whether this bound is optimal (even under relativizations). In fact, Ogiwara and Watanabe (personal communication) have shown that if a certain type of conjunctive wd-self-reducible set is many-one reducible to a sparse set, then it must be in P.

Acknowledgment

The authors would like to thank professor J.L. Balcazar for many helpful discussions on the topics of the paper.

References:

- [Al,He 89] E. Allender and L. Hemachandra: Lower bounds for the low hierarchy. *Proc. 16th ICALP* (1989) 31–46.
- [Ba,Gi,So 75] T.P. Baker, J. Gill, and R.M. Solovay: Relativizations of the $P=?NP$ question. *SIAM J. Comput.* 4 (1975), 431–442.
- [Ba 89] J.L. Balcázar: Self-reducibility. To appear in *Journal of Comp. and Syst. Sci.*
- [Ba,Di,Ga 88] J.L. Balcázar, J. Diaz, and J. Gabarró: “Structural Complexity” (vol. I). Springer-Verlag (1988).
- [Ba,Bo,Sc 86] J.L. Balcázar, R.V. Book, and U. Schöning: The polynomial-time hierarchy and sparse oracles. *Journal ACM* 33 (1986), 603–617.
- [Bu,Ha 88] S. Buss and L. Hay: On truth-table reducibility to SAT and the difference hierarchy over NP. *Proc. 3rd Structure in Complexity Conference*, (1988), 224–233.
- [Fo 79] S. Fortune: A note on sparse complete sets. *SIAM Journal on Computing* 8 (1979), 431–433.
- [Go,Jo,Yo 87] J. Goldsmith, D. Joseph and P. Young: Self-reducible, p -selective, near-testable and p -cheatable sets: the effect of internal structure on the complexity of a set. *Proc. 2nd Structure in Complexity Conference*, (1987), 50–59.
- [Ka 87] J. Kadin: $P^{NP[\log n]}$ and sparse Turing-complete sets for NP. *Proc. 2nd Structure in Complexity Conference*, (1987) 33–41.
- [Ko 83] K. Ko: On self-reducibility and weak p -selectivity. *Journal of Comp. and Syst. Sci.* 26 (1983), 209–221.
- [Ko,Sc 85] K. Ko and U. Schöning: On circuit-size complexity and the low hierarchy in NP. *SIAM J. Comput.* 14 (1985), 41–51.
- [Lo,To 89] A. Lozano and J. Torán: Relativized and positive separations of Δ_2^P and Θ_2^P . U. Politècnica de Catalunya, Tech. Report L.S.I-89-30, (1989).
- [Ma 82] S. Mahaney: Sparse complete sets for NP: solution of a conjecture by Berman and Hartmanis. *Journal Comput. Syst. Sci.* 25 (1982), 130–142.
- [Me,Pa 79] A. Meyer and M. Paterson: With what frequency are apparently intractable problems difficult?. M.I.T. Tech. Report TM-126, (1989).
- [Sc 83] U. Schöning: A low and a high hierarchy within NP. *Journal Comput. Syst. Sci.* 27 (1983), 14–28.
- [Sc 85] U. Schöning: “Complexity and structure”. *Lecture Notes in Computer Science* 211 Springer (1985).

Appendix:

In this section we obtain a relativized separation for the classes Δ_2^P and Θ_2^P . Although a separation satisfying this condition was first obtained in [Bu,Ha 88], we include one that shows that for a certain oracle A the test language

$$L_A = \{0^n : A_n = \emptyset \text{ or the minimum word of length } n \text{ in } A \text{ is even}\}$$

is not in Θ_2^P relative to A , since this particular result is needed in the proof of theorem 11.

Theorem: There is a recursive set A such that the set

$$L_A = \{0^n : A_n = \emptyset \text{ or the minimum word of length } n \text{ in } A \text{ is even}\}$$

is not in Θ_2^P relative to A .

Proof: Let $M_1, M_2, \dots, N_1, N_2, \dots$, be enumerations of the deterministic and nondeterministic Turing machines respectively, and suppose that a machine i has running time bounded by polynomial p_i .

It is well known that for every set B , the set

$$K(B) = \{\langle i, x, 0^n \rangle : N_i \text{ with oracle } B \text{ accepts } x \text{ in less than } n \text{ steps}\}$$

is complete for the class NP relative to B . We will construct a set A in such a way that L_A will not be accepted by any deterministic polynomial time machine making parallel queries to $K(A)$. We can consider that deterministic machines always make queries of type $\langle i, x, 0^n \rangle$ to the oracle.

A will be constructed in stages. At stage k we will add to A words of length n_k , diagonalizing away from the language recognized by M_k . We will also keep a list A' of words that will never be included in A . M_k with input 0^n produces a list of parallel queries to $K(A)$; for every query $q = \langle j, y, 0^t \rangle$ to $K(A)$ answered positively, we will include in A' the list of words that are not in A and are queried in a certain accepting path of M_j on input y .

We will say that machine M_k behaves correctly with input 0^{n_k} and oracle $K(A)$ if it accepts (rejects), and the minimum word of length n_k in A is even (odd). If this is the case, we add a new odd (even) word w to A of length n_k and smaller than the existing ones, in such a way that it does not affect the positive answers of the list of parallel queries to $K(A)$ (i.e. $w \notin A'$). Machine M_k , with the new oracle, either behaves incorrectly or has one more word in its list of parallel queries answered positively. Since M_k produces a list of polynomially many parallel queries, repeating the above procedure polynomially many times we can diagonalize away from M_k .

stage 0

$$\begin{aligned} A(0) &:= \emptyset \\ n_0 &:= 0 \end{aligned}$$

endstage

stage k

Let n_k be the smallest integer such that

- (i) $n_k > p_{k-1}(n_{k-1})$ and
- (ii) $2(p_k^2(n_k) + p_k(n_k)) < 2^{n_k}$

compute the list of parallel queries q_1, \dots, q_r

(with $q_i = \langle j_i, y_i, 0^{t_i} \rangle$ for $i = 1, \dots, r$) made by M_k on input 0^{n_k}

$$A' := \emptyset$$

$$A(k) := A(k-1)$$

$$d := 1; u_d := 1^{n_k}$$


```

repeat
   $A(k) := A(k) \cup \{u_d\}$ 
  for  $i := 1$  to  $r$  do
    if  $N_i$  with oracle  $A(k)$  accepts  $j_i$  in less than  $t_i$  steps then
       $A' := A' \cup \{ \text{words queried in the minimum accepting} \\ \text{path for } j_i \text{ that are not in } A(k) \}$ 
    endif
  endfor
   $d := d + 1$ 
   $u_d := \max\{x : |x| = n_k \text{ and } (x \text{ is even} \Leftrightarrow u_{d-1} \text{ is odd}) \text{ and } x \notin A' \text{ and } x < u_{d-1}\}$ 
until  $0^{n_k} \in L(M_k, K(A(k))) \Leftrightarrow 0^{n_k} \in L(M_k, K(A(k) \cup \{u_d\}))$ 
if  $M_k(K(A(k)))$  behaves correctly then
   $A(k) := A(k) \cup \{u_d\}$ 
endif
endstage

```

Whenever a new word is added to $A(k)$, the positive queries to $K(A(k))$ remain positive since for each one of them we reserve an accepting path that is not touched in the construction. Therefore, each time we go through the loop, we either force machine M_k to answer incorrectly, or we answer positively one more query to $K(A(k))$. Since there are only r queries to $K(A(k))$, and r is bounded by p_k , we only have to go p_k times through the loop. The selected word u_d included in $A(k)$ in each iteration always exists, since in A' there are at the most $p_k^2(n_k)$ reserved words, and before including u_d we could have included at the most $p_k(n_k)$ other words of length n_k in A . \square

It is interesting to observe that in the above construction the number of words in A can grow faster than any polynomial, i.e. set A is not sparse. In [Lo, To 89] a positive relativization of Δ_2^P and Θ_2^P is given that shows that these classes can be separated with a sparse oracle if and only if they are different in the absolute case.