On Random Reductions from Sparse Sets to Tally Sets

Uwe Schöning
Universität Ulm, Abt. Theoretische Informatik
Postfach 40 66, D-7900 Ulm
Germany

keywords: computational complexity, random reductions, sparse sets, tally sets, completeness

We show that every sparse set S can be many-one reduced to an appropriate tally set T by a polynomial-time, randomized reduction (see formal definitions below.) Since T is in NP if S is in NP, this result can be used to show that there is a tally set in NP being randomized many-one complete for all sparse sets in NP. This partially answers an open problem posed by Hartmanis and Yesha [6].

In [6] it is shown that there is a tally set in NP being polynomial-time *Turing* complete (actually, truth-table complete) for all sparse sets in NP, and the question was posed whether there is also a *many-one* complete set for all sparse sets in NP.

In [3] it was first shown that every sparse set is *conjunctively* reducible to an appropriate tally set. This is proved by a certain "low degree polynomial trick." Alternatively, in [8] the result is proved by a "chinese remainder trick." We use a modification of the latter proof method to present our randomized reduction. (A modification of the former method could be used just as well.)

A set $A \subseteq \{0,1\}^*$ is called *(polynomially) sparse* if there is a polynomial p

such that for each n, the cardinality of $A \cap \{0,1\}^n$ is upper bounded by $p(n)$. Any set $A \subseteq \{1\}^*$ over a one-letter alphabet is called *tally*. (Every tally set is clearly sparse.)

A set A *randomly many-one reduces* to a set B if there is a randomized, polynomial-time transducer algorithm M and a polynomial $q > 1$ such that for all x,

$$\begin{aligned} x \in A &\Rightarrow M(x) \in B \text{ with probability 1,} \\ x \notin A &\Rightarrow M(x) \notin B \text{ with probability at least } 1 - 1/q(|x|). \end{aligned}$$

Here $M(x)$ denotes the output of M on input x. This definition coincides essentially with Adleman and Manders's UR-reducibility [1], with the PR-reductions in [9] and the $\leq_m^{co-rp}$ reductions in [4]. In [9], a problem is shown to be NP-complete under PR-reductions which is not known to be NP-complete under polynomial-time Turing reductions.

We say that A *randomly many-one reduces* to B *with error polynomial* q if we want to specify the polynomial q in the above definition explicitly.

In the following, let $p_1, p_2, p_3, \ldots$ denote the sequence of prime numbers. For a string $x \in \{0,1\}^*$, $num(x)$ denotes the natural number whose binary representation is $1x$. We will make use of a polynomial-time computable pairing function $\langle \cdot, \cdot \rangle$ and its generalization to arbitrary n-tuples.

Theorem. *For every sparse set S there is a tally set T such that for every polynomial q, S randomly many-one reduces to T with error polynomial q.*

Additionally, if $S \in$ NP, then $T \in$ NP.

Proof. Let S be a sparse set, and let p be a polynomial such that $|S \cap \{0,1\}^n| \leq p(n)$ for all n. Define the tally set T as follows.

$$\begin{aligned} T = \{1^{\langle n,i,r \rangle} \mid\ & n \geq 0, \text{ and there is an } x \in S, \\ & |x| = n, \text{ such that } r = (num(x) \bmod p_i)\} \end{aligned}$$

The randomized reduction algorithm M works as follows.

input x ; $(*\ |x| = n\ *)$

guess randomly $i \in \{1, 2, \ldots, n \cdot p(n) \cdot q(n)\}$;

$r := num(x)$ **mod** p_i;

output $1^{\langle n,i,r \rangle}$;

By the prime number theorem (cf. [7]) the value of the i-th prime p_i is of order $O(i \cdot \ln i)$. Therefore, the length of the binary encoding of the $n \cdot p(n) \cdot q(n)$-th prime is $O(\ln n)$. That means, any inefficient, exponential-time primality test will be efficient enough—relative to the input length n—to guarantee that the procedure works in polynomial-time. (An interesting detail here is that the above procedure, for performing the random guess instruction, needs only *logarithmically* many random bits.)

Clearly, by definition of T, if $x \in S$, then every possible output $1^{\langle n,i,r \rangle}$ of M on input x will be a member of T.

On the other hand, if $x \notin S$, then there is the possibility that the output string $1^{\langle n,i,r \rangle}$ is in T because for some $y \in S$ with $|y| = n$, $(num(y)$ **mod** $p_i) = r$. We estimate now the probability that this happens. By the Chinese remainder theorem, if $a \neq b$ and $p_1, p_2, \ldots, p_m$ are different primes satisfying $\prod_{i=1}^{m} p_i \geq \max(a, b)$, then the sequences of remainders $(r_1, \ldots, r_m)$ and $(r'_1, \ldots, r'_m)$ obtained by taking $r_i = a$ **mod** p_i and $r'_i = b$ **mod** p_i are different (in at least one component of the sequence.) We apply this to our scenario here: $a = num(x)$ and $b = num(y)$ where both numbers are bounded by 2^{n+1}. Further, $\prod_{i=1}^{m} p_i \geq 2^{n+1}$ is garanteed if $m = n + 1$ since $p_i \geq 2$ for all i. Now consider all potential remainders $r_1, r_2, \ldots, r_{np(n)q(n)}$ that might be calculated by the randomized procedure on input x. At least one of the first $n + 1$ will be different for an arbitrary input $y \neq x$. Taking out this different one, but adding the $(n + 2)$-nd remainder, there will be again at least one difference in the remainder sequences, and so on. Altogether, we are sure to find at least $np(n)q(n) - n$ different points in the remainder sequences for x and y where they differ, and at most n remainders are equal.

Therefore, for fixed $y \neq x$, the probability for hitting an equal element in the remainder sequence is at most $\frac{1}{p(n)q(n)}$, and the probability for hitting with the output string $1^{\langle n,i,r \rangle}$ an element of T (because of some $y \in S$, $|y| = n$) is at most $|S \cap \{0,1\}^n| \cdot \frac{1}{p(n)q(n)} \leq \frac{1}{q(n)}$.

The set T is in NP if S is in NP. This can be seen as follows. On input $1^{\langle n,i,r\rangle}$, guess some x of length n and verify that $x \in S$. Then, the i-th prime has to be found and $r = (num(x) \textbf{ mod } p_i)$ needs to be verified. □

In some sense, in the above proof, the information contained in the sparse set S is distributed in logarithmically smaller pieces in the tally set, and this is done in a certain redundant, error-correcting way. Encoding a sparse NP set in a tally NP set such that the original information can be recovered is the key technique of Hartmanis's result [5] that $\mathsf{DTIME}(2^{O(n)}) \neq \mathsf{NTIME}(2^{O(n)})$ if and only if there exist sparse sets in $\mathsf{NP} - \mathsf{P}$. The above technique can also be used to prove this.

With very similar techniques as in [5] Hartmanis and Yesha [6] show that there is a tally set in NP being complete for all sparse sets in NP. The completeness notion here is understood for *Turing* reducibility. (Actually, any Turing reduction to a tally set is already a truth-table reduction.) The authors ask whether there is also a *many-one* complete set for all sparse sets in NP, but in the same paper they show that there are relativizations which do not allow such sets. Therefore, we think the following (relativizable) result is of interest.

Corollary. *There is a tally set in* NP *which is complete for all sparse sets in* NP *under randomized many-one reductions.*

Proof. There is a set A that is complete for $\mathsf{NTIME}(2^{O(n)})$ under linear-time many-one reductions. Consider its tally version

$$TALLY(A) = \{1^{num(x)} \mid x \in A\}.$$

$TALLY(A)$ is a member of NP (cf. [2]) and is a tally set. Let $S \in \mathsf{NP}$ be a sparse set. By the last theorem there is a tally set $T \in \mathsf{NP}$ such that S randomly many-one reduces to T. Consider the binary encoded version of T,

$$BIN(T) = \{x \in \{0,1\}^* \mid 1^{num(x)} \in T\}.$$

$BIN(T)$ is a member of $\mathsf{NTIME}(2^{O(n)})$, and therefore linear-time many-one reducible to A. This implies the existence of a polynomial-time many-one reduc-

tion from T to $TALLY(A)$. Combining the reductions, we obtain a randomized many-one reduction from S to $TALLY(A)$. □

References

[1] L. Adleman, K. Manders. Reductions that lie. *Proceedings of the 20th Annual Conference on Foundations of Computer Science*, IEEE, 1979, 397–410.

[2] R.V. Book. Tally languages and complexity classes. *Information and Control* 26 (1974), 186–193.

[3] H. Buhrman, L. Longpré, E. Spaan. Sparse reduces conjunctively to tally. Northeastern University, College of Computer Science, Technical Report NU-CCS-92-8, 1992.

[4] R. Chang, J. Kadin, P. Rohatgi. Connections between the complexity of unique satisfiability and the threshold behavior of randomized reductions. *Proceedings of the 6th Annual Conference on Structure in Complexity Theory*, IEEEE, 1991, 255–266.

[5] J. Hartmanis. On sparse sets in NP – P. *Information Processing Letters* 16 (1983), 55–60.

[6] J. Hartmanis, Y. Yesha. Computation times of NP sets of different densities. *Theoretical Computer Science* 34 (1984), 17–32.

[7] I. Niven, H.S. Zuckerman. *An Introduction to the Theory of Numbers.* Wiley, New York, 1960.

[8] S. Saluja. Relativized limitations of the left set technique and closure classes of sparse sets. Department of Computer Science, Tata Institute of Fundamental Research, Bombay, India; Technical Report, 1992.

[9] U. Vazirani, V. Vazirani. A natural encoding scheme proved probabilistic polynomial complete. *Theoretical Computer Science* 24 (1983), 291–300.