論文 / 著書情報
Article / Book Information

題目(和文)	手に負えない問題の構造について
Title(English)	On the Structure of Intractable Complexity Classes
著者(和文)	渡辺治
Author(English)	O. Watanabe
出典(和文)	学位:工学博士, 学位授与機関:東京工業大学, 報告番号:乙第1688号, 授与年月日:1987年5月31日, 学位の種別:論文博士, 審査員:
Citation(English)	Degree:Doctor of Engineering, Conferring organization: Tokyo Institute of Technology, Report number:乙第1688号, Conferred date:1987/5/31, Degree Type:Thesis doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

On the Structure of Intractable Complexity Classes

BY

Osamu WATANABE

Department of Computer Science
Tokyo Institute of Technology
Ookayama, Meguro-ku
Tokyo 152, JAPAN

A dissertation
submitted for the degree of doctor of engineering
at Tokyo Institute of Technology
January, 1987

ACKNOWLEDGEMENTS

I am indebted to my advisers, Professor Takuya Katayama and Professor Kojiro Kobayashi, for their encouragement and support throughout my preparing this dissertation. Professor Kojiro Kobayashi also has helped me and given me invaluable advice since I started my research in theoretical computer science.

I would like to thank Professor Ronald Book who invited me to University of California, Santa Barbara. There I could have fruitful discussions with Professor Ronald Book, Professor Ker-I Ko, Dr. Pekka Orponen, Dr. David Russo, and Professor Uwe Schöning which contributed to this work. Professor Ronald Book and Professor Eric Allender read an earlier version of this manuscript carefully and gave me helpful comments.

I would like to thank staffs and colleagues in Department of Information Sciences and Department of Computer Science, Tokyo Institute of Technology for the good environment to study computer science and to prepare manuscripts.

ABSTRACT

This dissertation explores the meaning of several structural properties. Structural properties provide the method to investigate the "intractability" of problems. We estimate the difficulty of a problem by showing that it does (or does not) posses some structural property. During this decade, several structural properties have been introduced expecting that they express certain tractability/intractability concepts. However, for many structural properties, we have not understood their meaning mathematically in the context of computational complexity theory. This dissertation is devoted to the investigation of what these properties express and how they do it in computational complexity theory.

Structural properties have been introduced to attack the unsolved P $\neq$ NP question. They are expected to express certain (in)tractability notions and to show how and why NP-computation is more powerful than P-computation. Note that it is difficult to prove the latter type of intuitions since we do not even know whether P = NP. However we can examine the former type of intuitions by studying these structural properties on the class of problems which is *known* to be intractable. The class EXP (= $DTIME(2^{lin})$) is the smallest time complexity class such that we can prove so far that (a) it properly includes P and that (b) it contains the essential part of NP. Hence, in order to investigate the meaning of each structural property, we study how it works on EXP.

Four basic structural properties and the related concepts are studied: several polynomial time reducibilities and completeness notions; similarity of $\leq_m^p$-complete sets; polynomial time reducibility to a set of small density; and lowness and highness notions. We solve several conjectures concerning these structural properties on EXP. Although we state our results for EXP, many results and proofs which we develop are also applicable to other complexity classes. The possibility and a way to investigate other complexity classes are also discussed.

TABLE OF CONTENTS

1. Introduction

In the course of developing computational complexity theory we have established several concepts and techniques to measure the computational difficulty of a given problem. During this decade, several authors have developed and extensively studied different ways to show the "intractability" of problems which are called *structural methods*. In a structural method we consider a *structural property* which expresses (or seems to express) the computational difficulty: we estimate the difficulty of a problem or a class of problems by showing that they do (or do not) posses such a property. For example, the "NP-completeness" notion is a typical one. Since "NP-completeness" asserts being one of the hardest problems in NP, by showing "NP-completeness" for a given problem we prove that it is one of the hardest problems in NP.

Thus a structural property is a key in a structural method. For each structural property, we need to make its meaning precise and to understand how it shows "tractability/intractability". In order to attack the unsolved P $\neq$ NP question, we have introduced many structural properties. Many of them are imported from recursive function theory into complexity theory. Although such properties have meaning and express certain concepts in recursive function theory, we cannot unconditionally assume this meaning in complexity theory: they may not inherit their meaning properly. Furthermore there are some structural properties arising in complexity theory which have no correspondence in recursive function theory. For them our intuition on their meaning has no background in mathematical logic. Therefore it is necessary for us to reconsider or to investigate the phenomena that structural properties express and how they do this in the context of complexity theory.

This dissertation is devoted to the investigation of meaning of several structural properties and to understand what they express and how they express it in the context of polynomial time computations. To accomplish this investigation we will take one approach: we investigate each structural property by studying how it works on certain class of problems which is *known* to be "intractable". Before stating the contents of this dissertation we will give a quick review on basic notions and structural properties considered throughout this dissertation.

*   *   *

The Tractability Notion Based on Polynomial Time Computation

We will follow a standard framework in complexity theory (the reader should be familiar with the material covered with a textbook like [HU79]). We consider only *decision problems* (i.e., *recognition problem*) and formalize each "problem" by a corresponding set. That is, a problem is defined by the set of all strings that should be answered "Yes" in this problem. Throughout this dissertation, by a problem A we mean the corresponding set.

We use standard *deterministic/nondeterministic Turing machines* for the model of computation. In order to quantify the amount of resource needed to recognize a given set we use *complexity measures* (or *complexity functions*): in this dissertation we will mainly consider *time complexity* measured by the number of Turing machine steps.

For several reasons which have been discussed in many contexts (see, e.g., [Co71, GJ79, Ha78, HU79]), we adopt the "tractability" notion based on *deterministic polynomial time computability* (the word "deterministic" will be omitted unless necessary). That is, a set which is recognizable by some deterministic Turing machine within polynomial time should be considered "easy". In other words, we ignore the differences for which some polynomial time computation can compensate. Although we will consider weaker "tractability" notions later on, polynomial time computability is an essential one. We sometimes expand this way of thinking: we regard an amount bounded by some polynomial as a "tractable amount", which we refer as a "polynomial criterion".

The class of sets which are recognizable within polynomial time bound is called P. Also varying "time bound" or "computation style" we can define other classes: they are called *complexity classes* in general. In addition to the class P we will consider the following complexity classes:

NP = the class of sets decidable by nondeterministic Turing machines within polynomial time;

EXP = the class of sets decidable by deterministic Turing machines within $2^{O(n)}$ time; and

NEXP = the class of sets decidable by nondeterministic Turing machines within $2^{O(n)}$ time.

Note that it is provable that P $\subsetneq$ EXP [HU79]; thus, EXP is a provably intractable complexity class.

Polynomial Time Reducibility

The notion of "polynomial time reducibility" was introduced [Co71, Ka72] to analyze relative difficulty between two problems (i.e., sets). The following is a general explanation of polynomial time reducibility: for any two sets A and B, we say that A is *polynomial time reducible* to B if some deterministic algorithm recognizes A within polynomial time provided an oracle is available for deciding $x \in B$ for all x. This situation means that the intractability of A is at most that of B since A cannot be intractable unless B is intractable (i.e., $A \in$ P if $B \in$ P).

The notion of reducibility can be seen in recursive function theory [Ro66]. Indeed, several types of reducibility have been introduced from recursive function theory (see, e.g., [LLS76, Lo82]). Among them reducibility notions available to analyze the relative intractability in terms of polynomial time computations vary between $\leq^p_T$-reducibility to $\leq^p_m$-reducibility: that is, $\leq^p_T$-reducibility is the most general (the *weakest*) and $\leq^p_m$-reducibility is the most restrictive (the *strongest*); we use $\leq^p_r$ to denote any one of them. We refer reducibilities from $\leq^p_T$ to $\leq^p_m$ as *polynomial time reducibility* in general. Thus the above explanation for polynomial time reducibility is also for the most general one, i.e., $\leq^p_T$-reducibility. Restricting the way of using an oracle, we obtain stronger reducibilities; and $\leq^p_m$-reducibility is the most restrictive one. We can explain $\leq^p_m$-reducibility in terms of "transformation of a problem": A is $\leq^p_m$-reducible to B if some polynomial time computable function (called a $\leq^p_m$*-reduction*) maps each instance x for A to some instance y for B such that $x \in A$ iff $y \in B$ (one can see in this explanation some restrictive way of using information from an oracle; thus, $\leq^p_m$ is a special case of $\leq^p_T$).

The notion of "completeness" describes the idea of being most intractable in a given complexity class. For any complexity class $\mathcal{C}$, we say that a set A is $\leq^p_r$*-complete* in $\mathcal{C}$ if (i) A is in $\mathcal{C}$ and (ii) all sets in $\mathcal{C}$ are $\leq^p_r$-reducible to A. That is, A is one of the hardest set (to recognize) in $\mathcal{C}$ with respect to $\leq^p_r$-reducibility. Note that all $\leq^p_r$-complete sets in $\mathcal{C}$ are $\leq^p_r$-reducible to each other. Two sets are said to be $\equiv^p_r$*-equivalent* if they are $\leq^p_r$-reducible to each other. So all $\leq^p_r$-complete sets in $\mathcal{C}$ are $\equiv^p_r$-equivalent; in other words, the class of $\leq^p_r$-complete sets consists one $\equiv^p_r$-equivalence class, which is the most difficult $\equiv^p_r$-equivalence class in $\mathcal{C}$.

Density

Recall that a problem is defined by the set of all strings that should be answered "Yes" in this problem. Then *density* of a problem is the density of the corresponding set. Density is one of the basic and important structural properties. In particular the property of "sparseness" is important here. A set A is called *sparse* if there is a fixed polynomial such that for all n, the number of elements in $A^{\leq n}$ is bounded by this polynomial in n (where $A^{\leq n}$ denote elements in A of length less than or equal to n).

The "sparseness" is another (and weaker) notion of "tractability". Let A be any sparse set. For each $n > 0$, consider a table which lists up all elements in $A^{\leq n}$. Having this table we can recognize A for all inputs of length up to n. Since only polynomially many elements are in $A^{\leq n}$, the size of the table is polynomially bounded; thus using this table, we can solve A up to length n within polynomial time. With our "polynomial criterion" we regard this polynomial size table as feasible. That is, $A^{\leq n}$ is *tractably* solvable with a *feasible* size of table: A is "tractable" in this sense. Of course, the construction of this table may be hard and may need a huge amount of time; hence this idea of "tractability" is much weaker notion than polynomial time computability.

Following a similar argument, the property of "polynomial time reducibility to a sparse set" also gives the same notion of "tractability". Let A be any set which is $\leq_r^{\mathrm{P}}$-reducible to some sparse set B. That is, some polynomial time algorithm M recognizes A asking questions to an oracle for B. Note that the length of queries made by M is bounded by some polynomial $p(n)$. Thus with a table for $B^{\leq p(n)}$, which is still polynomial size, the above algorithm M can feasibly solve A for all inputs up to length n: this way of solving A is called a *table-lookup* algorithm.

Similarity of All $\leq_{\mathrm{m}}^{\mathrm{P}}$-Complete Sets

In 1971 Cook defined the idea of NP-completeness and showed that SAT was NP-complete; this being the first NP-complete set [Co71, Ka72] (in the following, by " $\mathcal{C}$ -complete set" we refer a $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete set in $\mathcal{C}$). Since then a huge number of NP-complete sets have been discovered (see [GJ79]). Although they have come from wide variety of problem areas, researchers began to notice that they are similar in some sense and share some properties. For example, we expect that all NP-complete sets have similar density. Indeed, Mahaney [Ma82] proved that no NP-complete sets are sparse if and only if P $\neq$ NP.

A similarity between NP-complete sets suggest that NP-computation (nondeterministic polynomial time bounded computation) is more powerful than P-computation (deterministic polynomial time bounded way of computations). Let A and B be an arbitrary pair of NP-complete sets. Then we have $A \equiv_{\mathrm{m}}^{\mathrm{P}} B$; thus there exist some $\leq_{\mathrm{m}}^{\mathrm{P}}$-reductions f from A to B and g from B to A. Suppose that there is not so much difference in computational power between P- and NP-computation. Then B can be structurally simple: the reduction f compensates for the lack of information in B so that we still have $x \in A \longleftrightarrow f(x) \in B$. Indeed, in such an extreme case as P = NP, B can be any set but the empty set or Σ^*. For example, assuming $A \in$ P, we have the following polynomial time computable function f which $\leq_{\mathrm{m}}^{\mathrm{P}}$-reduces A to a singleton $\{a_0\}$:

$$f(x) = \begin{cases} a_0 & \text{if } x \in A, \\ x & \text{otherwise,} \end{cases}$$

where a_0 is any string in A.

On the other hand, if NP-computation has a great advantage over P-computation, then A and B must have a similar structure since no polynomial time reduction has enough computational power to accomplish drastic change in the structure of A to that of B and vice versa. Notice that this argument holds not only for a pair of $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete sets in NP but also for a pair of sets in NP which are $\equiv_r^p$-equivalent. However, since NP-complete sets are regarded as the most intractable sets in NP, we expect to see in the similarity of NP-complete sets as much difference between P- and NP-computation as possible; thus, we consider NP-complete sets.

In general we can argue in a similar way for any other complexity class $\mathcal{C}$ such that P $\subseteq \mathcal{C}$. That is, for some kind of similarity we expect that similarity of all $\mathcal{C}$-complete sets characterizes the superiority of $\mathcal{C}$-computation to P-computation. For example "having a similar density" seems to be one of such similarity. On the other hand, "using the same alphabet" obviously has nothing to do with the difference between $\mathcal{C}$ - and P-computation. It is interesting to consider what types of similarity are available for this purpose.

Several notions are necessary in order to discuss "similarity". Recall that all $\mathcal{C}$-complete sets are $\equiv_{\mathrm{m}}^{\mathrm{p}}$-equivalent, that is, they all are reducible to each other by $\leq_{\mathrm{m}}^{\mathrm{p}}$-reductions. We state "similarity" in terms of the type of such $\leq_{\mathrm{m}}^{\mathrm{p}}$-reductions: for example, the assertion that two sets are $\leq_{\mathrm{m}}^{\mathrm{p}}$-reducible each other by *one-to-one* reductions states that they have a similar density. So, some notions concerning the type of functions are necessary here. A function f is called respectively *p-computable* and *length-increasing* if it is polynomial time computable by some deterministic Turing transducer and $|f(x)| > |x|$ for all x. A one-to-one function f is called *p-invertible* if f^{-1} is p-computable. And a function f is *one-way* if it is one-to-one, length-increasing, p-computable but not p-invertible. It is easy to show that if P = NP, no one-way functions exist. However the existence of one-way functions is widely believed, especially in the field of cryptography (see [Al85, Gs84, KLD85]).

We will use the following equivalence relations in order to express "similarity": for any sets A and B,

$A \equiv_{1,\mathrm{li}}^{\mathrm{p}} B$ if there exist one-to-one, length-increasing and p-computable reductions from A to B and B to A;

$A \equiv^{\mathrm{p}}_{\mathrm{inv}} B$ if there exist one-to-one, length-increasing, p-computable and p-invertible reductions from A to B and B to A;

$A \equiv^{\mathrm{p}}_{\mathrm{iso}} B$ if there exists a one-to-one, onto, p-computable and p-invertible reduction f from A to B.

The last relation, $\equiv^{\mathrm{p}}_{\mathrm{iso}}$, is called *p-isomorphism.* Berman and Hartmanis [BH77] proposed the notion "p-isomorphism" expecting that it appropriately states the basic similarity between all NP-complete sets. Surprisingly Berman and Hartmanis showed that indeed all of the *well-known* NP-complete sets (e.g., NP-complete sets in [GJ79]) are p-isomorphic; they conjectured that *all* NP-complete sets are p-isomorphic (this known as the Berman-Hartmanis conjecture).

Polynomial Lowness and Polynomial Highness

The concept of "lowness/highness" qualitatively measures the information content of a given set A: it shows how useful (the information encoded by) a set A may be. For any set A, we define P(A) to be the class of sets recognizable by some polynomial time algorithms with help of an oracle for A, i.e., $\mathrm{P}(A) = \{B : B \leq^{\mathrm{p}}_{\mathrm{T}} A\}$ (for any other complexity class $\mathcal{C}$, $\mathcal{C}(A)$ is defined similarly). We measure the "usefulness" of a given set by investigating the type of complexity class we can obtain when we use it as an oracle. For example, if $\mathrm{P}(A) = \mathrm{P}$ and $\mathrm{P}(B) = \mathrm{EXP}$, then B is more useful than A; thus B is considered to encode more information than A.

There may be many ways to measure the degree of usefulness according to this approach. Schöning offered one of such approach [Sc83]: he used the *polynomial time hierarchy* Σ^{p}_0, Σ^{p}_1, ... [St77] (where $\Sigma^{\mathrm{p}}_k = \underbrace{\mathrm{NP}(\mathrm{NP}(\cdots \mathrm{NP}}_{\mathrm{k}}(\mathrm{P}) \cdots))$, see Chapter 6 for the precise definition).

(*Note:* It follows from the definition that $\Sigma_0^{\mathrm{p}} \subseteq \Sigma_1^{\mathrm{p}} \subseteq \cdots$; furthermore it is conjectured that these inclusions are proper (the non-collapse conjecture).)

It follows from the definition that for every $k > 0$ and every set A in NP, we have that $\Sigma_k^{\mathrm{p}} \subseteq \Sigma_k^{\mathrm{p}}(A) \subseteq \Sigma_{k+1}^{\mathrm{p}}$. So, we define "polynomial lowness/highness" as follows: for any set A in NP and every $k \geq 0$, A is Σ_k^{p}-*low* if $\Sigma_k^{\mathrm{p}} = \Sigma_k^{\mathrm{p}}(A)$, and A is Σ_k^{p}-*high* if $\Sigma_{k+1}^{\mathrm{p}} = \Sigma_k^{\mathrm{p}}(A)$. Define $\mathrm{L}_k^{\mathrm{p}}$ and $\mathrm{H}_k^{\mathrm{p}}$ to be the class of Σ_k^{p}-low sets and Σ_k^{p}-high sets respectively. In general Σ_k^{p}-lowness (resp., Σ_k^{p}-highness) is called *polynomial lowness* (resp., *polynomial highness*).

From the definition, the class $\mathrm{L}_0^{\mathrm{p}}$ and $\mathrm{H}_0^{\mathrm{p}}$ is the class P and the class of $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete sets in NP, respectively. And it is easy to show that $\mathrm{L}_0^{\mathrm{p}} \subseteq \mathrm{L}_1^{\mathrm{p}}$ $\cdots$, and $\mathrm{H}_0^{\mathrm{p}} \subseteq \mathrm{H}_1^{\mathrm{p}}$ $\cdots$. Moreover we expect that $\{\mathrm{L}_k^{\mathrm{p}}\}_{k\geq 0}$ and $\{\mathrm{H}_k^{\mathrm{p}}\}_{k\geq 0}$ construct hierarchies of (in)tractability in NP between P and the class of $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete sets.

* * *

All the structural properties reviewed above and the concepts behind them have been introduced in the study of the P $\neq$ NP question. For almost two decades, we have been unable to develop a more clever algorithm for any of NP-complete problems than the one based on the exhaustive search. So, we feel that we have no deterministic algorithm for, say, SAT which is considerably faster than the trivial $2^{O(n)}$ time algorithm. However we have been unable to prove that P $\neq$ NP either; thus we try to investigate how and why NP differs from P, and this attempt has led us to develop several structural methods. Hence, as we have reviewed, these structural methods are expected to show how and why NP-computation is more powerful than P-computation. Unfortunately there often occur the cases where our conjectures concerning these structural properties are appear to be much more difficult than the P $\neq$

NP question.

For example, consider the problem of similarity between all NP-complete sets. As discussed above, we have an intuition that *some* similarity of NP-complete sets characterizes P $\subsetneq$ NP. The Berman-Hartmanis conjecture, i.e., the conjecture that all NP-complete sets are p-isomorphic, comes from the intuition that (i) P $\subsetneq$ NP and (ii) p-isomorphism is an appropriate similarity to exhibit such a difference. However this conjecture is at least as difficult as the P $\neq$ NP question: the Berman-Hartmanis conjecture certainly implies that P $\neq$ NP. Is there any way to investigate the soundness of the intuition (ii) without facing the big question such as P $\subsetneq$ NP? One approach is to investigate it not on the class NP but on some complexity class $\mathcal{C}$ for which we *know* (a) P $\subsetneq$ $\mathcal{C}$ and (b) NP $\subseteq$ $\mathcal{C}$. If the intuition (ii) is sound, then we can also expect the p-isomorphism conjecture for $\mathcal{C}$ since $\mathcal{C}$ is at least as intractable as NP. Furthermore we may have more chance to prove it since we *know* that P $\subsetneq$ $\mathcal{C}$. On the contrary if we disprove p-isomorphism on $\mathcal{C}$, we could suspect (ii), i.e., "p-isomorphism" may be a too strong similarity.

In general this approach offers a way to verify our intuition on structural properties. For any one of structural properties considered here, a conjecture, say X, from our intuitive understanding implies P $\subsetneq$ NP (or assumes P $\subsetneq$ NP). Thus the conjecture X is at least as difficult as the P $\neq$ NP question. On the other hand, for the conjecture X we can usually expect that it holds on any other complexity class $\mathcal{C}$ such that NP $\subseteq$ $\mathcal{C}$ if it should hold on NP. Thus if we prove (resp., disprove) it on a reasonably small class $\mathcal{C}$ such that NP $\subseteq$ $\mathcal{C}$, this fact is a good evidence for the soundness (resp., unsoundness) of the conjecture X. We need not solve the P $\neq$ NP question for proving the conjecture X. Furthermore, since we *know* P $\subsetneq$ $\mathcal{C}$, the proof may be tractable! We will take this approach in this dissertation. We will (mainly) consider the class EXP (= $DTIME(2^{lin})$) and study the above structural properties on EXP.

The class EXP is the smallest class such that we can prove so far that (a) it *properly* includes P and that (b) it contains all the *well-known* NP-complete sets. Note that what we can prove so far is not NP $\subseteq$ EXP but NP $\subseteq$ EXPOLY (= $DTIME(2^{poly})$). However we do not have even the candidate for EXP − NP or NP − EXP. Also it is easy to show that all the proofs shown here work for EXPOLY (in general, they work for any deterministic complexity classes which include EXP). Hence we will consider this small class EXP. Of course, there is a difference between NP and EXP; thus what is true on EXP may not on NP: we may use the property that is particular for EXP. So, in this dissertation we will also investigate the structural properties on other complexity classes besides the class EXP. The differences between NP and EXP are the following two points: (i) NP is a nondeterministic complexity class while EXP is a deterministic one and (ii) the time bound for NP is polynomial while that for EXP is exponential (this difference also causes the phenomenon that EXP is not closed under $\leq^{\mathrm{p}}_{\mathrm{m}}$-reductions but NP is). Thus, as other complexity classes, we consider (i) nondeterministic ones and (ii) complexity classes which have other time bounds than 2^{lin}. In the following we survey the contents of this dissertation and explain the main observations and results achieved.

* * *

Chapter 2 provides definitions and propositions used elsewhere in this dissertation.

In Chapter 3 we discuss the difference between several polynomial time reducibilities. Especially, we compare each notion of completeness w.r.t. these reducibilities.

There are several notions of "polynomial time reducibility": $\leq^{\mathrm{p}}_{\mathrm{T}}$-reducibility is the most general one and $\leq^{\mathrm{p}}_{\mathrm{m}}$-reducibility is the most restrictive one. Although every polynomial time reducibility notion states relative tractability within polynomial time variance, we think that its meaning differs by each reducibility and that a restrictive reducibility yields a tighter relation than the general one. Ladner, Lynch and Selman proved this intuition: they proved the following.

Theorem. [LLS76]

Let $\leq^{\mathrm{p}}_{r_1}$ and $\leq^{\mathrm{p}}_{r_2}$ be any pair of polynomial time reducibilities considered in several contexts (see Chapter 2 for their definitions). Then either it is clear from the definition that $\leq^{\mathrm{p}}_{r_1}$ implies $\leq^{\mathrm{p}}_{r_2}$, or there exist sets A and B in EXP such that $A \leq^{\mathrm{p}}_{r_1} B$ but $A \not\leq^{\mathrm{p}}_{r_2} B$. □

Then consider the meaning of completeness notions induced by polynomial time reducibilities. Here again we conjecture that the meaning of "completeness" differs by each reducibility. However the above theorem does not state such differences; and the differences between completeness notions have been left open in general [KM81, LLS76]. We solve almost all of them on EXP: we distinguish between completeness notions w.r.t. almost all typical reducibilities.

In Section 3.1 we compare completeness notions w.r.t. all reducibilities weaker than $\leq^{\mathrm{p}}_{\mathrm{tt}}$-reducibility. Here we introduce a structural property ("having $\mathcal{I}$-easy subset") which is a generalization of "non-p-immunity". Using this notion we can uniformly show the difference between several completeness notions on EXP. We also consider nondeterministic complexity classes. We obtain the difference between $\leq^{\mathrm{p}}_{\mathrm{m}}$-completeness and $\leq^{\mathrm{p}}_{\mathrm{2\text{-}c}}$-completeness (thus $\leq^{\mathrm{p}}_{\mathrm{btt}}$-completeness) on NEXP.

In Section 3.2 we distinguish $\leq_{tt}^{p}$-completeness from $\leq_{T}^{p}$-completeness on EXP. From the definition of these reducibilities, we know that all $\leq_{tt}^{p}$-reduction can use only non-adaptive query information whereas some $\leq_{T}^{p}$-reduction may use adaptive one. This difference appears in the compressibility of query strings. Using the concept of "uncompressible string" (i.e., "string of high Kolmogorov complexity"), we construct a $\leq_{T}^{p}$-complete set in EXP which is not $\leq_{tt}^{p}$-complete.

In Chapter 4 we discuss the similarity of EXP-complete sets. First, in Section 4.1, we investigate the problem of what is an appropriate similarity to characterize the superiority of $\mathcal{C}$-computation to P-computation.

Berman and Hartmanis [BH77] conjectured that all NP-complete sets are p-isomorphic. We investigate the validity of this conjecture by considering the same question on EXP: that is, the question of whether all $\leq_{m}^{p}$-complete sets in EXP are p-isomorphic. Then we obtain some similarity ($\equiv_{1,li}^{p}$-equivalence relation) between all $\leq_{m}^{p}$-complete sets in EXP, which almost solves the above question. However we see, at the same time, that "non-p-invertibility" of a $\leq_{m}^{p}$-reduction is an essential and crucial obstacle in showing the p-isomorphism result on EXP.

We prove that all $\leq_{m}^{p}$-complete sets in EXP are $\equiv_{1,li}^{p}$-equivalent. Note that if no one-way functions exist, $\equiv_{1,li}^{p}$ is the same as $\equiv_{inv}^{p}$ since every one-to-one and length-increasing $\leq_{m}^{p}$-reduction is always p-invertible. Also we have the following technical lemma.

Lemma. [BH77]

For any two sets A and B, if $A \equiv_{inv}^{p} B$ then A and B are p-isomorphic. That is, $\equiv_{inv}^{p}$ and $\equiv_{iso}^{p}$ define the same equivalence relation. □

Hence if we assume the non-existence of one-way functions, then p-isomorphism is the same as $\equiv^{p}_{1,li}$-equivalence relation; and our result also shows that all $\leq^{p}_{m}$-complete sets in EXP are p-isomorphic. On the contrary, assuming the existence of one-way functions, we have a candidate for non-p-isomorphic pairs. Consider any one-way function f and any standard $\leq^{p}_{m}$-complete set U in EXP. It is clear that $f(U)$ is also $\leq^{p}_{m}$-complete in EXP; furthermore we have that $U \equiv^{p}_{1,li} f(U)$. However we have been unable to prove that $U \leq^{p}_{inv} f(U)$. In fact, assuming the existence of a one-way function f, we conjecture that U and $f(U)$ are not p-isomorphic.

The same arguments also hold for any deterministic complexity classes which include EXP. Also we observe partial $\equiv^{p}_{1,li}$-equivalence relation on nondeterministic complexity classes such as NEXP. As a consequence of these investigations we propose the following conjecture.

Conjecture.

For each sufficiently difficult complexity class $\mathcal{C}$, we have

(1) all $\leq^{p}_{m}$-complete sets in $\mathcal{C}$ are $\equiv^{p}_{1,li}$-equivalent; but

(2) not all $\leq^{p}_{m}$-complete sets in $\mathcal{C}$ are p-isomorphic.

(*Note:* That is, the (extended) Berman-Hartmanis conjecture is false; and instead of the similarity by "p-isomorphism", we expect a little weaker similarity, i.e., $\equiv^{p}_{1,li}$-equivalence relation.) □

Since we propose the non-p-isomorphism conjecture, next, in Section 4.2, we investigate further the structure of the class of $\leq^{p}_{m}$-complete sets in EXP under the non-p-isomorphism conjecture. We discuss two topics here.

The first topic concerns the form of $\leq^{p}_{m}$-complete sets which are non-p-isomorphic to a standard $\leq^{p}_{m}$-complete set U. For any one-way function f, the pair U and $f(U)$ is a candidate for non-p-isomorphic pair. On the other hand, Joseph and Young showed another type of non-p-isomorphic candidates on NP (k-creative sets [JY86]). One of the problems of interest is whether we have the former simple type of non-p-isomorphic pair if the non-p-isomorphism conjecture is true at all. We can partially solve this problem on NP and EXP, and completely on EXPSPACE.

The second topic is motivated with the observation by Mahaney.

Theorem. [Ma81]

If not all $\leq_m^p$-complete sets in NP (EXP, etc.) are p-isomorphic, there are infinitely many mutually non-p-isomorphic $\leq_m^p$-complete sets in NP (EXP, etc.). □

How can one construct these infinitely many $\leq_m^p$-complete sets? We obtain some function f such that $\{f^i(U)\}_{i \geq 0}$ is the class of mutually non-p-isomorphic $\leq_m^p$-complete sets in NP (EXP, etc.) (some assumptions are necessary depending on each complexity class). That is, we exhibit a uniform and effective way to generate infinitely many non-p-isomorphic $\leq_m^p$-complete sets.

In Chapter 5 we discuss polynomial time reducibility to a set of small density, e.g., a sparse set.

The notion of "polynomial time reducibility to a sparse set" offers a weaker notion of tractability than "polynomial time computability (or recognizability)". We conjecture that if P $\subsetneq \mathcal{C}$, then sufficiently difficult sets in $\mathcal{C}$, e.g., complete sets in $\mathcal{C}$, are not "tractable" in this weak sense. Indeed Berman and Hartmanis showed that no $\leq_m^p$-complete sets in EXP are $\leq_m^p$-reducible to a sparse set: moreover we have the following theorem.

Theorem. [BH77, BS84]

Let A be any set to which some $\leq_m^p$-complete set in EXP is $\leq_m^p$-reducible (i.e., A is any $\leq_m^p$-hard set for EXP). Then A has a bi-exponential density. □

We extend it in Section 5.1. We prove that all $\leq_{c}^{p}$- (resp., $\leq_{d}^{p}$-, $\leq_{btt}^{p}$-) hard sets for EXP are of bi-exponential density. In other words, no $\leq_{c}^{p}$- (resp., $\leq_{d}^{p}$-, $\leq_{btt}^{p}$-) complete sets in EXP are $\leq_{c}^{p}$- (resp., $\leq_{d}^{p}$-, $\leq_{btt}^{p}$-) reducible to a set of small density: that is, no $\leq_{c}^{p}$- (resp., $\leq_{d}^{p}$-, $\leq_{btt}^{p}$-) complete sets are "tractable" even in the weak sense.

Polynomial time reducibility to a sparse set means the existence of a feasible table-lookup algorithm. For any set A and for every $n > 0$, a feasible table-lookup algorithm solves $A^{\leq n}$ with help of some polynomial size table which is determined by each n. Thus, a table-lookup algorithm differs from our usual notion of computation: it is not a uniform algorithm. We call algorithms of this type as *pseudo algorithms*. Several authors have introduced polynomial time pseudo deterministic algorithms: a (1-)APT machine [MP79, OS86]; a p-close machine [Ye83, Sc86b]; and a polynomial size circuit [KL80]. Each of these pseudo algorithms is regarded as a special case of table-lookup algorithms. Also we define several types of table-lookup algorithms depending on the type of polynomial time reductions. In Section 5.2, we investigate the relation between these pseudo algorithms. As a consequence (and using the results in Section 5.1), we prove that no $\leq_{btt}^{p}$-hard sets have p-close machines nor (1-)APT machines.

In Chapter 6 we study the concepts of polynomial lowness (and highness). Schöning [Sc83] has introduced these notions expecting that they exhibit the degree of "usefulness" and thus the degree of "difficulty". More precisely, we expect that $\{L_{k}^{p}\}_{k\geq 0}$ and $\{H_{k}^{p}\}_{k\geq 0}$ construct hierarchies of (in)tractability in NP between P and the class of $\leq_{T}^{p}$-complete sets. However we do not know whether $\{L_{k}^{p}\}_{k\geq 0}$ (resp., $\{H_{k}^{p}\}_{k\geq 0}$) does not collapse. For example, consider L_{0}^{p} and L_{1}^{p}, and let A_{0} and A_{1} be any sets arbitrary chosen from L_{0}^{p} and L_{1}^{p} respectively. Then, from the definition, we have $P(A_{0}) = P$

and $\mathrm{NP}(A_1) = \mathrm{NP}$. That is, any oracle information of A_0 is P-computable while that of A_1 is NP-computable. Hence, we conjecture that some set in $\mathrm{L}_1^{\mathrm{p}}$ is *more* difficult than any sets in $\mathrm{L}_0^{\mathrm{p}}$. However we cannot prove it even assuming $\mathrm{P} \neq \mathrm{NP}$ (this problem is shown to be the same as the $\mathrm{P} \neq \mathrm{NP} \cap$ co-NP question [Sc83]). We investigate this conjecture on EXP: we consider the similar problem in the context of EXP-computation.

Recall that we use the polynomial time hierarchy to define the polynomial lowness and highness notions. In order to define lowness and highness similar to the polynomial ones, we use EXP hierarchy, E_0, E_1, $\cdots$ (where $\mathrm{E}_k = \underbrace{\mathrm{EXP}(\mathrm{EXP}(\cdots\mathrm{EXP}}_{k}(\mathrm{P})\cdots)))$. We define EXP lowness and highness, i.e., $\{\mathrm{L}_k^{\mathrm{e}}\}_{k\geq 0}$ and $\{\mathrm{H}_k^{\mathrm{e}}\}_{k\geq 0}$ in almost the same way to the polynomial ones. We have similar properties and conjectures concerning these notions. We investigate one of these conjectures: consider the conjecture that $\mathrm{L}_0^{\mathrm{e}} \subsetneq \mathrm{L}_1^{\mathrm{e}}$. From the definition, this essentially asks if $\mathrm{EXP}(A) = \mathrm{EXP}$ always implies $A \in \mathrm{P}$. Using the concept of Kolmogorov complexity here again, we prove this conjecture [BORW86].

In Chapter 7 we summarize important open problems related to this research. We also survey some new structural approaches. There are other kind of properties which we do not consider in this dissertation. Several interesting notions have been defined quite recently. Especially the notions of "generalized Kolmogorov complexity" [Ha83, Ko86] and "instance complexity" [KSOW86] seem to provide new structural methods. We give a short survey for these two subjects.

2. Preliminaries

This chapter provides definitions and propositions used elsewhere in this dissertation.

Strings and Languages

Let Σ denote the set $\{0, 1\}$, and let Σ^* denote the set of all words on Σ. By a *language* we mean a subset of Σ^*. For a string x, $|x|$ denotes the length of x, and for a set S, $\|S\|$ denotes the number of elements in S. For a set A, let $A^{\leq n}$, $A^{=n}$ and $A^{>n}$ denote $\{x \in A : |x| \leq n\}$, $\{x \in A : |x| = n\}$ and $\{x \in A : |x| > n\}$.

For any set A, we define the *census* of A to be the function $\text{cens}_A(n) = \|A^{\leq n}\|$. For any integer valued function $d(n)$, we say that a set A is of $d(n)$-*density* if $\text{cens}_A(n) = d(n)$ for all $n > 0$.

We assume a polynomial time computable pairing function from $\Sigma^* \times \Sigma^*$ to Σ^*. Let $\lambda xy.\langle x, y\rangle$ be such a function.

Computation Model

Our basic computation model is the standard deterministic / nondeterministic multi-tape Turing machines (the reader will find precise definitions in, e.g., [HU79]). A Turing machine may or may not be an oracle machine. All nondeterministic machines are acceptors; but a deterministic machine may be an acceptor or a transducer.

Remarks on Notations. We use the symbol M and N to denote a Turing machine acceptor and transducer respectively. We sometimes use notations such as M_A or N_μ to denote some specific machines. We use $M(x)$ (resp., $M^A(x)$) to denote the execution of M on input x (relative to A); and use $N(x)$ to denote the output of N on input x. For any machine M (and any set of strings A), define $L(M)$ (resp., $L(M, A)$) to be the set of strings *accepted* by

M (relative to A).

We consider classes of machines such as a class of all polynomial time bounded (in some way) Turing machine acceptors, or the class of all Turing machine transducers, and so on. In each case, we assume some standard way of enumerating all machines in each class. Furthermore we assume to such an enumeration the existence of a *universal Turing machine* which satisfies, for example, the following proposition [HU79].

Proposition 2.1. Let $\{M_i\}_{i>0}$ be a standard enumeration of oracle Turing machines. Then there exists a universal Turing machine M_U such that for every oracle set A, it satisfies the following:

for every $i > 0$ and every $x \in \Sigma^*$,

(1) $\langle i, x\rangle \in L(M_U, A) \leftrightarrow x \in L(M_i, A)$; and

(2) $M_U^A(\langle i, x\rangle)$ halts within $|i|\cdot t_i(x)\log t_i(x)$ steps, where t_i is a time bound for M_i.

(*Note:* We often use a more rough time bound, say $|i|\cdot t_i(x)^2$, for a time bound of $M_U^A(\langle i, x\rangle)$.) □

Complexity Classes

A *complexity class* is a class of languages accepted by the class of all Turing machines of some specific type. For every class of Turing machines, we can define the corresponding complexity class; and vice versa. That is, each complexity class $\mathcal{C}$ corresponds to some class of Turing machines; by a $\mathcal{C}$ *-machine* we mean one of such machines. $\mathcal{C}$ *-computation* refers to the computation by some $\mathcal{C}$ -machine. Note that for any complexity class $\mathcal{C}$, the set of $\mathcal{C}$ -machines must be definable, but it may or may not be recursively enumerable. For example, the class P is the class of languages accepted by deterministic machines which halts within some polynomial time: these machines are called *polynomial time deterministic machines*. Note that the set of all polynomial time deterministic machines is not recursively enumerable (Cf. the class of all polynomial time *bounded* deterministic machines is recursively enumerable).

We will consider the following complexity classes:

$DTIME(t(n)) = \{L : L$ is accepted by some deterministic Turing machine within $t(n)$ steps$\}$;

$NTIME(t(n)) = \{L : L$ is accepted by some nondeterministic Turing machine within $t(n)$ steps$\}$;

$DSPACE(s(n)) = \{L : L$ is accepted by some deterministic Turing machine within $s(n)$ space$\}$;

$\mathrm{P} = \cup\{DTIME(p(n)) : p$ is a polynomial$\}$;

$\mathrm{NP} = \cup\{NTIME(p(n)) : p$ is a polynomial$\}$;

$\mathrm{EXP} = \cup\{DTIME(2^{cn}) : c > 0\}$;

$\mathrm{NEXP} = \cup\{NTIME(2^{cn}) : c > 0\}$;

$\mathrm{EXPOLY} = \cup\{DTIME(2^{p(n)}) : p$ is a polynomial$\}$;

$\mathrm{PSPACE} = \cup\{DSPACE(p(n)) : p$ is a polynomial$\}$;

$\mathrm{EXPSPACE} = \cup\{DSPACE(2^{cn}) : c > 0\}$; and

$\mathrm{FP} = \{f : f$ is computable by some deterministic Turing machine transducer within some polynomial time$\}$

(i.e., a *polynomial time function* is a function in FP).

We will use the term *super-SUBEXP* to refer to any *deterministic* complexity class that includes the class $\cup\{DTIME(2^{cn^{\epsilon}}) : c > 0\}$ for some ϵ, $0 < \epsilon < 1$; and also use the term *super-PSPACE* to refer to any *deterministic* complexity class that includes the class $DSPACE(\tau(n))$ for some super polynomial τ (a function τ is *super polynomial* if for all polynomial p, $\tau(n) = \omega(p(n))$). Precisely speaking, since we assume that each complexity class has a *standard enumeration* of Turing machines which accept only and all languages in the class, the notions super-SUBEXP and super-PSPACE refer to only such classes.

Remarks on Notations. In computational complexity theory, we often use the abbreviations such as $O(f(n))$ to simplify statements. Among several definitions [BC86], we will use the following simple ones (in the following by "$\overset{\infty}{\exists}\, n$" and by "$\overset{\infty}{\forall}\, n$" we mean "for infinitely many n" and "for almost all n" respectively):

(1) $\Omega(f(n))$ denotes any function g such that $(\exists c > 0)(\overset{\infty}{\exists}\, n)[\, g(n) > c \cdot f(n)\,]$;

(2) $\omega(f(n))$ denotes any function g such that $(\exists c > 0)(\overset{\infty}{\forall}\, n)[\, g(n) > c \cdot f(n)\,]$;

(3) $O(f(n))$ denotes any function g such that $(\exists c > 0)(\overset{\infty}{\forall}\, n)[\, g(n) < c \cdot f(n)\,]$; and

(4) $o(f(n))$ denotes any function g such that $(\exists c > 0)(\overset{\infty}{\exists}\, n)[\, g(n) < c \cdot f(n)\,]$.

For any complexity class $\mathcal{C}$, consider any oracle machine that is made by attaching the oracle-query facility to some $\mathcal{C}$-machine. The computation by such an oracle machine is called $\mathcal{C}(\,)$-computation; and the class of languages accepted by $\mathcal{C}(\,)$-computation using oracle X is denoted by $\mathcal{C}(X)$. For any complexity class $\mathcal{C}$ and $\mathcal{D}$, we define $\mathcal{C}(\mathcal{D})$ to be the class $\cup\{\, \mathcal{C}(A) : A \in \mathcal{D}\}$.

Reducibility, Hardness and Completeness

A polynomial time reducibility between two sets shows a relative tractability between them. For any two sets A and B, A is *polynomial time reducible* to B ($A \leq_r^{\mathrm{p}} B$, where $\leq_r^{\mathrm{p}}$ stands for some reduction type) if there exists a polynomial time *reduction* from A to B. There are several types of reductions; they determine the type of reducibility. We first define three basic types of reductions.

Let A and B be any set of strings.

(1) A *polynomial time many-one reduction* ($\leq_{\mathrm{m}}^{\mathrm{p}}$-reduction) from A to B is a polynomial time computable function f such that $x \in A$ iff $f(x) \in B$;

(2) The ordered pair $\langle\langle a_1, \cdots, a_k\rangle, \alpha\rangle$ is a *truth-table condition* (tt-condition) of *norm* k if $\langle a_1, \cdots, a_k\rangle$ is a k-tuple, $k > 0$ of strings in Σ^* and α is a k-ary Boolean function presented in terms of a Boolean formula (we omit the detailed definition of a "Boolean formula", see [LLS76; Section 3]). Here the set $\{a_1, \cdots, a_k\}$ and and the Boolean formula is called the *associated set* and *evaluator* of the tt-condition respectively. A function f is a *truth-table function* (tt-function) if f is total and $f(x)$ is a tt-condition for all x in Σ^*. Then a *polynomial time truth-table reduction* ($\le^{\mathrm{p}}_{\mathrm{tt}}$-reduction) from A to B is a polynomial time computable tt-function f such that $x \in A$ iff $\alpha(C_B(a_1), \cdots, C_B(a_k)) =$ **true**, where $f(x) = \langle\langle a_1, \cdots, a_k\rangle, \alpha\rangle$ and C_B is the characteristic function of the set B; and

(3) A *polynomial time Turing reduction* ($\le^{\mathrm{p}}_{\mathrm{T}}$-reduction) from A to B is a polynomial time deterministic oracle machine M such that $A = L(M, B)$.

Remarks on Notations. For any oracle Turing machines, any oracle set A and every string x, let $Q(M, x, A)$ denote the set of queries made during the computation of $M^A(x)$. Also for any tt-function f and every string x, let $Ass(f, x)$ denote the associated set of the tt-condition $f(x)$.

We also classify $\le^{\mathrm{p}}_{\mathrm{tt}}$-reductions according to the following classification of truth-table functions (*tt-functions*).

(2-1) For any polynomial p, a *p(n)-tt-function* is a tt-function whose norm (i.e., the size of a truth-table) is bounded by p;

(2-2) A *btt-function* is a tt-function whose norm is bounded by some constant; and

(2-3) A *ctt-function* (resp., *dtt-function*) is a tt-function whose truth-table evaluator is always conjunctive (resp. disjunctive).

Then a $\le^{\mathrm{p}}_{p(n)\text{-tt}}$-, $\le^{\mathrm{p}}_{\mathrm{btt}}$-, $\le^{\mathrm{p}}_{\mathrm{c}}$- and $\le^{\mathrm{p}}_{\mathrm{d}}$-reduction is a $\le^{\mathrm{p}}_{\mathrm{tt}}$-reduction whose tt-function is a $p(n)$-tt-function, btt-function, ctt-function and dtt-function respectively. We sometimes combine these notations to denote combined reductions: e.g., a $\leq^{\mathrm{p}}_{\mathrm{1\text{-}c}}$-reduction is a polynomial time 1-bounded and conjunctive truth-table reduction.

For any reduction type $\leq^{\mathrm{p}}_{r}$, we say that A is $\leq^{\mathrm{p}}_{r}$- reducible to B if A is polynomial time reducible via some $\leq^{\mathrm{p}}_{r}$-reduction. Notice that among reducibilities defined above, $\leq^{\mathrm{p}}_{\mathrm{T}}$-reducibility is the most general; that adding restrictions on the way of oracle queries, we obtain the others; and that $\leq^{\mathrm{p}}_{\mathrm{m}}$-reducibility is the most restrictive.

There are another types of reductions which offer more general reducibility than $\leq^{\mathrm{p}}_{\mathrm{T}}$-reducibility (e.g., [Lo82, Se82, Se83]). However in our use of "reducibility", we need not consider more general one than $\leq^{\mathrm{p}}_{\mathrm{T}}$-reducibility. Also there are another types of reductions which offer more fine (restrictive) reducibility than $\leq^{\mathrm{p}}_{\mathrm{m}}$-reducibility (e.g., $\leq^{\mathrm{p}}_{\mathrm{1,li}}$-reductions). We use them not for comparing intractability but for exhibiting structural similarity; they will be defined and studied in Chapter 4.

Let $\mathcal{C}$ be any complexity classes. For any reduction type $\leq^{\mathrm{p}}_{r}$, a set C is *$\leq^{\mathrm{p}}_{r}$-hard for* $\mathcal{C}$ if every set in $\mathcal{C}$ is $\leq^{\mathrm{p}}_{r}$-reducible to C; and C is *$\leq^{\mathrm{p}}_{r}$-complete in* $\mathcal{C}$ if C is $\leq^{\mathrm{p}}_{r}$-hard and $C \in \mathcal{C}$.

For typical $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets, a *standard $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set* is often considered (e.g., [Ha78]). Let $\mathcal{C}$ be any complexity class $\mathcal{C}$ which has a standard enumeration $\{M_i\}_{i>0}$ of Turing machines which accept only and all languages in $\mathcal{C}$. Then we can define a standard $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set for any complexity class in the following way:

$U = \{\langle i, x, pad(i, x)\rangle : M_i \text{ accepts } x\}$,

where $pad(i, x)$ is some polynomial time computable padding function.

We will often use a standard complete set for EXP. The following is one example of standard complete sets for EXP: letting $\{M_i\}_{i>0}$ be a standard enumeration of deterministic Turing machines, define U by

$$U = \{\langle i, x, m, 0^{|i|+|m|}\rangle : M_i \text{ accepts } x \text{ within } 2^{|m|} \text{ steps}\}.$$

Then from our assumption concerning standard enumeration, it is easy to show that U is $\leq^{\mathrm{P}}_{\mathrm{m}}$-complete in EXP.

Generalized Kolmogorov Complexity

Hartmanis [Ha83] introduced a *generalized Kolmogorov complexity measure*. Informally, it measures how far and how fast a string can be compressed. More formally, we define the following sets. Consider an enumeration of all deterministic Turing machine transducers $\{N_i\}_{i>0}$. For every $i > 0$ and every string x, let $t_i(x)$ denote a running time of N_i on x. Then, for each pair g and t of functions, we define the following sets:

$$K_i[g(n), t(n)] = \{y : (\exists x)[\ |x| \leq g(|y|), N_i(x) = y, \text{ and } t_i(x) \leq t(|y|)\ ]\};$$
$$K[g(n), t(n)] = K_u[g(n), t(n)],$$

where u is the index of a universal Turing transducer.

Then from our assumption on a universal Turing machine, we have the following fact.

Proposition 2.2. [Ha83] Let i be any index and let g and t be time constructible functions. Then there exists $c > 0$ and $d > 0$ such that $K_i[g(n), t(n)] \subseteq K[g(n)+d, ct(n)\log t(n)+c]$. □

3. Comparison of Polynomial Time Reducibilities

In this chapter we investigate the difference between several polynomial time reducibilities. Especially, we compare each notion of completeness w.r.t. these reducibilities.

A polynomial time reducibility between two sets shows the relative tractability between them. Consider two sets A and B such that A is polynomial time reducible to B (e.g., $A \leq_{\mathrm{T}}^{\mathrm{p}} B$; see Chapter 2). Then the assertion $A \leq_{\mathrm{T}}^{\mathrm{p}} B$ states that some deterministic algorithm accepts A within polynomial time using B as an oracle (such an algorithm is called a *$\leq_{\mathrm{T}}^{\mathrm{p}}$-reduction* from A to B). That is, the computational complexity of A is at most that of B within polynomial time variance. Recall that a $\leq_{\mathrm{T}}^{\mathrm{p}}$-reduction is the most general one and that adding restrictions to the method of asking oracle queries and using oracle answers yields several types of polynomial time reductions; thus we have several polynomial time reducibilities depending on the type of reductions. Each reducibility seems to offer different notion of "relative tractability". In this chapter we will investigate such differences.

The basic types of reductions considered in this paper are *polynomial time many-one reduction* ($\leq_{\mathrm{m}}^{\mathrm{p}}$-reduction), *polynomial time truth-table reduction* ($\leq_{\mathrm{tt}}^{\mathrm{p}}$-reduction), and *polynomial time Turing reduction* ($\leq_{\mathrm{T}}^{\mathrm{p}}$-reduction). We also classify $\leq_{\mathrm{tt}}^{\mathrm{p}}$-reductions according to the classification of truth-table functions: they are $\leq_{p(n)\text{-tt}}^{\mathrm{p}}$-, $\leq_{\mathrm{btt}}^{\mathrm{p}}$-, $\leq_{\mathrm{c}}^{\mathrm{p}}$- and $\leq_{\mathrm{d}}^{\mathrm{p}}$- reductions and their possible combinations (see Chapter 2 for their definitions). In this chapter we will consider polynomial time reducibilities with respect to the above reduction types.

Let $\leq_{r_1}^{\mathrm{p}}$ and $\leq_{r_2}^{\mathrm{p}}$ be any two polynomial time reducibilities. We say that $\leq_{r_1}^{\mathrm{p}}$ is *stronger* than $\leq_{r_2}^{\mathrm{p}}$ (or $\leq_{r_2}^{\mathrm{p}}$ is *weaker* than $\leq_{r_1}^{\mathrm{p}}$) if for all A and B, we have that $A \leq_{r_1}^{\mathrm{p}} B$ implies $A \leq_{r_2}^{\mathrm{p}} B$. Thus, among the above, $\leq_{\mathrm{T}}^{\mathrm{p}}$-reducibility is the weakest and $\leq^{\mathrm{p}}_{\mathrm{m}}$-reducibility is the strongest. Although every polynomial time reducibility notion states relative tractability within polynomial time variance, we think that its meaning differs by each reducibility and that strong reducibility offers a tighter relation than a weak one. For example, suppose that $A_1 \leq^{\mathrm{p}}_{1\text{-tt}} B$ and that $A_2 \leq^{\mathrm{p}}_{\mathrm{tt}} B$ (note that $\leq^{\mathrm{p}}_{1\text{-tt}}$-reducibility is stronger than $\leq^{\mathrm{p}}_{\mathrm{tt}}$-reducibility). Then a $\leq^{\mathrm{p}}_{1\text{-tt}}$-reduction from A_1 to B accepts A_1 by using only one oracle query of B for each input whereas every $\leq^{\mathrm{p}}_{\mathrm{tt}}$-reduction from B to A_2 may need polynomially many oracle queries of B; in this case, we view the amount of information about B per string to be closer to that of A_1 than that of A_2. We conjecture this situation occurs. That is, we conjecture that $\leq^{\mathrm{p}}_{1\text{-tt}}$-reducibility offers a tighter relation than $\leq^{\mathrm{p}}_{\mathrm{tt}}$-reducibility. Ladner, Lynch and Selman confirmed this intuition: they established the differences between all the possibly different pair of reducibilities; moreover, the witness they exhibited are in EXP. Their results are summarized as follows.

Theorem 3.1. [Theorem 3.2 and 3.3 of LLS76]

Let p and q be any polynomials such that $p(n) > q(n) \geq 0$ for almost all n. Then, for each condition below, there exist sets A and B in EXP which satisfy that condition:

(1) $A \leq^{\mathrm{p}}_{p(n)\text{-}d} B$ (thus, $A \leq^{\mathrm{p}}_{p(n)\text{-tt}} B$) but $A \not\leq^{\mathrm{p}}_{q(n)\text{-}tt} B$;

(2) $A \leq^{\mathrm{p}}_{p(n)\text{-}c} B$ (thus, $A \leq^{\mathrm{p}}_{p(n)\text{-tt}} B$) but $A \not\leq^{\mathrm{p}}_{q(n)\text{-}tt} B$;

(3) $A \leq^{\mathrm{p}}_{1\text{-}tt} B$ but $A \not\leq^{\mathrm{p}}_{\mathrm{c}} B$;

(4) $A \leq^{\mathrm{p}}_{1\text{-}tt} B$ but $A \not\leq^{\mathrm{p}}_{\mathrm{d}} B$;

(5) $A \leq^{\mathrm{p}}_{2\text{-}d} B$ but $A \not\leq^{\mathrm{p}}_{\mathrm{c}} B$; and

(6) $A \leq^{\mathrm{p}}_{2\text{-}c} B$ but $A \not\leq^{\mathrm{p}}_{\mathrm{d}} B$. □

From the definitions, we have obvious relations between polynomial time reducibilities: e.g., $\leq^{\mathrm{p}}_{\mathrm{m}}$-reducibility is the same as $\leq^{\mathrm{p}}_{1\text{-c}}$-reducibility, e.t.c.

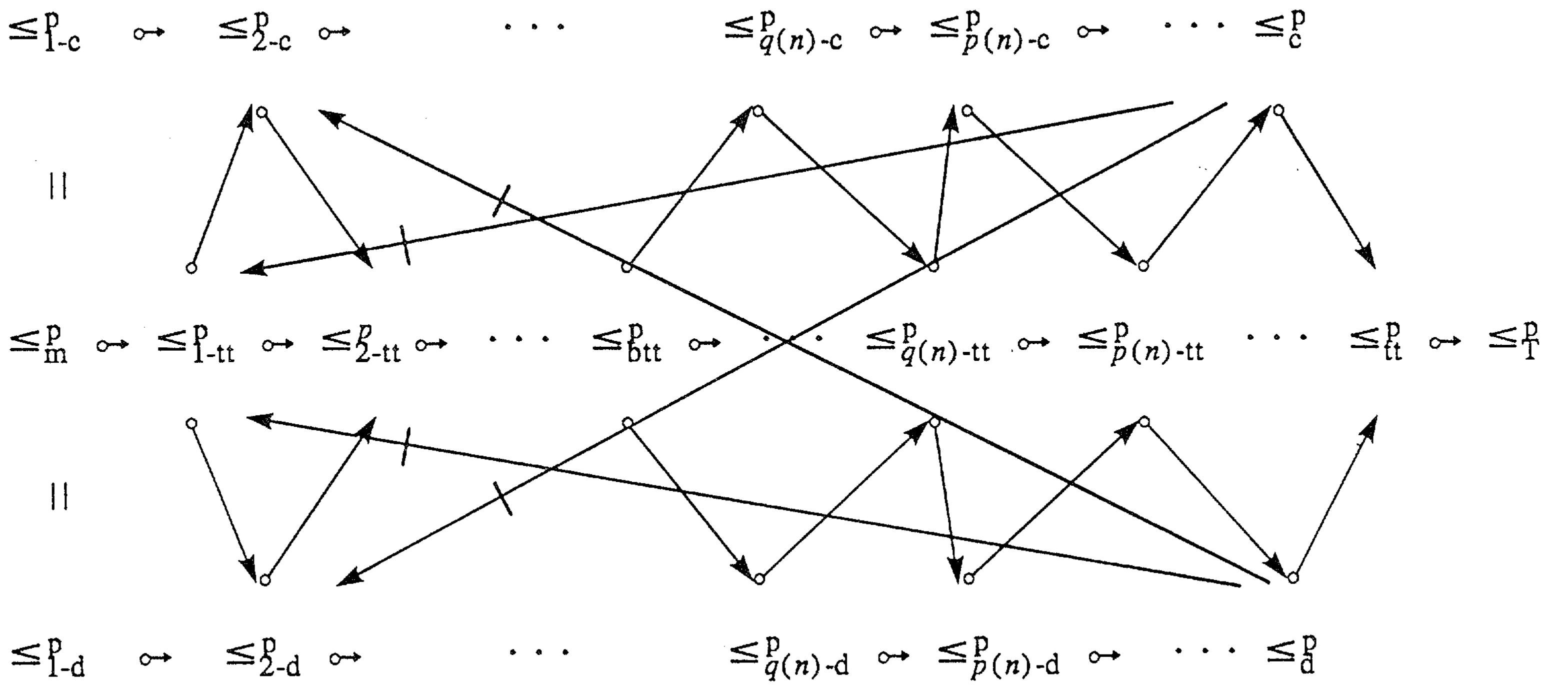
Therefore, Theorem 3.1 shows all the differences between any possible pair of polynomial time reducibilities considered here in EXP: we have the complete results of comparison on all the pair of polynomial time reducibilities in EXP (Figure 3.1).

What about the meaning of completeness notions induced by polynomial time reducibilities? For any $\leq_r^{\mathrm{p}}$-reducibility and any complexity class $\mathcal{C}$, a $\leq_r^{\mathrm{p}}$-complete set C in $\mathcal{C}$ is one of the most difficult sets in $\mathcal{C}$ with respect to $\leq_r^{\mathrm{p}}$-reducibility: all sets in $\mathcal{C}$ are $\leq_r^{\mathrm{p}}$-reducible to C; thus, they are at most as difficult as C. Here again, following the same argument as the above, we conjecture that the meaning of "completeness" differs by each reducibility. Let $\leq_{r_1}^{\mathrm{p}}$ and $\leq_{r_2}^{\mathrm{p}}$ be any two reducibilities such that $\leq_{r_1}^{\mathrm{p}}$ is stronger than $\leq_{r_2}^{\mathrm{p}}$, and let $\mathcal{C}$ be any complexity class such that $\mathrm{P} \subsetneq \mathcal{C}$. Then we have that all $\leq_{r_2}^{\mathrm{p}}$-complete sets in $\mathcal{C}$ are $\leq_{r_1}^{\mathrm{p}}$-complete. On the other hand, we conjecture that there exists a $\leq_{r_1}^{\mathrm{p}}$-complete set in $\mathcal{C}$ which is not $\leq_{r_2}^{\mathrm{p}}$-complete: that is, $\leq_{r_1}^{\mathrm{p}}$ offers narrower notion of completeness than $\leq_{r_2}^{\mathrm{p}}$. We will investigate the validity of this intuition.

Here notice that Theorem 3.1 does not show the differences in completeness notions. For example, Theorem 3.1 gives some sets A and B in EXP such that $A \leq_{\mathrm{T}}^{\mathrm{p}} B$ but $A \not\leq_{\mathrm{m}}^{\mathrm{p}} B$; thus B is not $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete in EXP. But for our purpose, i.e., in order to obtain a set which is $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete but not $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete, B need to be $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete in EXP. However it seems difficult to modify the proof of Theorem 3.1 so that B could be $\leq_{\mathrm{T}}^{\mathrm{p}}$-complete. In the following two sections we will prove these differences.

3.1. $\leq_{\mathrm{m}}^{\mathrm{p}}$-Completeness and $\leq_{\mathrm{tt}}^{\mathrm{p}}$-Completeness Notions

First we consider polynomial time many-one reducibility and all kinds of truth-table reducibilities. That is, we compare completeness notions w.r.t. all


$\leq^p_{1\text{-c}}$ ⊸ $\leq^p_{2\text{-c}}$ ⊸ . . . $\leq^p_{q(n)\text{-c}}$ ⊸ $\leq^p_{p(n)\text{-c}}$ ⊸ . . . $\leq^p_{c}$
||
$\leq^p_{m}$ ⊸ $\leq^p_{1\text{-tt}}$ ⊸ $\leq^p_{2\text{-tt}}$ ⊸ . . . $\leq^p_{btt}$ ⊸ . . . $\leq^p_{q(n)\text{-tt}}$ ⊸ $\leq^p_{p(n)\text{-tt}}$ . . . $\leq^p_{tt}$ ⊸ $\leq^p_{T}$
||
$\leq^p_{1\text{-d}}$ ⊸ $\leq^p_{2\text{-d}}$ ⊸ . . . $\leq^p_{q(n)\text{-d}}$ ⊸ $\leq^p_{p(n)\text{-d}}$ ⊸ . . . $\leq^p_{d}$

Where

$p(n) > q(n) > 0$ for almost all n;

$\alpha \nrightarrow \beta$: $A\ \alpha\ B$ does not imply $A\ \beta\ B$;

$\alpha \multimap \beta$: $A\ \alpha\ B$ implies $A\ \beta\ B$ but the converse is false; and

$\alpha = \beta$: α and β are the same, i.e., $A\ \alpha\ B$ implies $A\ \beta\ B$ *and* $A\ \beta\ B$ implies $A\ \alpha\ B$.

(a relation deduced from the others may be omitted)

Figure 3.1. The comparison of several reducibilities in EXP.

reducibilities weaker than $\leq_{tt}^{p}$-reducibility. Ko and Moore [KM81] proved the existence of a $\leq_{tt}^{p}$-complete set in EXP which is not $\leq_{m}^{p}$-complete. But their proof is quite complicated and they did not establish the differences between the other types of completeness. Here we take an indirect approach. First we establish for each polynomial time reducibility, some structural properties of complete sets in EXP with respect to this reducibility. Using these properties we can construct the desired complete sets based on simple diagonalizations. For example, in order to obtain a $\leq_{tt}^{p}$-complete set which is not $\leq_{m}^{p}$-complete, we first show some structural property of $\leq_{m}^{p}$-complete sets (i.e., all $\leq_{m}^{p}$-complete sets have FP-easy subsets); and then construct a $\leq_{tt}^{p}$-complete set that does not have such a property. This strategy is much easier than to construct $\leq_{tt}^{p}$-complete set which is not $\leq_{m}^{p}$-complete directly (this proof strategy is one of the good examples of using structural properties!).

As a structural property of complete sets in EXP, we consider a property of "having an infinite easy subset". The question of whether a set has an infinite "easy" subset has been considered in several contexts [Be76, KM81, Ro67]. If a set A has no infinite "easy" subset, every nontrivial part of A is not "easy": that is, A is almost everywhere complex. Such a set A is called an *immune* set. The notion of "easy" differs in each context. For example, consider the "polynomial time" case. Then our problem is whether a set has an infinite and polynomial time recognizable subset, i.e., an "easy" subset. If a set A has no infinite subset which is in P, it is called *P-immune*. In general, we have the following definition of "immunity".

Definition 3.1. For any complexity class $\mathcal{C}$, a set A is $\mathcal{C}$*-immune* if A contains no infinite subset in $\mathcal{C}$.

Consider the problem of showing that a given set A is not P-immune, that is, showing the existence of an infinite polynomially recognizable subset of A.

One of the ways to exhibit an infinite "easy" subset is to consider a length increasing function whose range is a subset of A (this idea was used first by Berman [Be76; Theorem 8]). In the case of non-P-immunity we need a polynomial time computable length increasing function f such that $f(\Sigma^*) \subseteq A$. Suppose that such a function f exists. Then the set $S = \{f(0^n) : n \geq 0\}$ is an infinite subset of A such that $S \in$ P. Thus, A has an infinite "easy" (i.e., polynomially recognizable) subset, so A is not P-immune. Berman [Be76] proved that all $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete sets in EXP have easy subsets like S. We show a similar property for every polynomial time reducibilities.

We extend the above construction of an "easy" subset in the following way. Let $\mathcal{F}$ be a class of functions. We say that a set A has an *$\mathcal{F}$-easy subset* if there exists a total function f in $\mathcal{F}$ such that for almost all x, $|f(x)| > |x|$ and $f(x) \in A$. The name "easy subset" comes from the following observation: let A have an $\mathcal{F}$-easy subset w.r.t. a function f in $\mathcal{F}$. Then there exists an integer n_0 such that the set $\{f(0^n) : n > n_0\}$ is an infinite subset of A that is easily recognizable relative to f. For any $\leq_r^{\mathrm{p}}$-reducibility, we will show that every $\leq_r^{\mathrm{p}}$-complete set in EXP has an $\mathcal{F}$-easy subset where the class $\mathcal{F}$, the class of functions, depends on the reduction type r.

We begin by defining several classes of functions. Note that each reduction is a special case of Turing reductions: i.e., we can make an oracle machine M which corresponds to each reduction. Consider a function g which maps every $x \in \Sigma^*$ to some query of M on input x (with an oracle A). We say such a function g is *generated by* a reduction M (and A). We consider the following classes of functions generated by reductions of each type:

(1) Fm $= \{g : g$ is generated by a polynomial time oracle machine which corresponds to some $\leq_{\mathrm{m}}^{\mathrm{p}}$-reduction$\}$;

(2) Fbtt $= \{g : g$ is generated by a polynomial time oracle machine

which corresponds to some $\leq^{\mathrm{p}}_{\mathrm{btt}}$-reduction};

(3) Fp-tt = {g : g is generated by a polynomial time oracle machine which corresponds to some $\leq^{\mathrm{p}}_{p(n)\text{-tt}}$-reduction};

(4) Ftt = {g : g is generated by a polynomial time oracle machine which corresponds to some $\leq^{\mathrm{p}}_{\mathrm{tt}}$-reduction}; and

(5) FT = {g : g is generated by a polynomial time oracle machine which corresponds to some $\leq^{\mathrm{p}}_{\mathrm{T}}$-reduction to some oracle set}.

(*Note:* Concerning *total* functions, we can use the class FP, i.e., the class of polynomial time computable functions, instead of using Fm: the class Fm is the class of total functions in FP. Thus we will use the term "FP-easy subset" instead of "Fm-easy subset").

We show that for any $\leq^{\mathrm{p}}_{r}$-reducibility, every $\leq^{\mathrm{p}}_{r}$-complete set in EXP has an $\mathcal{F}$-easy subset where $\mathcal{F}$ depends on the reduction type r. These results are corollaries of the following theorem.

Theorem 3.2. Let A be any set in EXP. Then there is a set $L_A \subseteq \mathrm{N}\times\Sigma^*$ in EXP which satisfies the following property: if $L_A \leq^{\mathrm{p}}_{\mathrm{T}} A$ by M_i, then for almost all x, there exists y in $Q(M_i, A, \langle i, x\rangle) \cap A$ such that $|y| > |x|$ (recall that $Q(M_i, A, \langle i, x\rangle)$ denotes the set of strings queried by M_i during the execution on input $\langle i, x\rangle$ relative to A).

Proof. Let $\{M_i\}_{i>0}$ be an enumeration of polynomial time bounded oracle machines, and let p_i denote a polynomial time bound of M_i.

Define the set L_A as follows:

$$\langle i, x\rangle \in L_A \iff |i|\cdot p_i(n)^2 \leq 2^n \textit{ and } \langle i, x\rangle \notin L(M_i, A^{\leq n}),$$
$$\text{where } n = |\langle i, x\rangle|.$$

Recall that for any M_i and any oracle X, the standard universal machine $M_U^X(\langle i, x\rangle)$ simulates $M_i^X(x)$ within $|i|\cdot p_i(|\langle i, x\rangle|)^2$ steps for all x (see Chapter 2 for "universal Turing machine"). Also since $A \in$ EXP, there exists an exponential time machine M_A which accepts A. Using the machines M_A and M_U, it is easy to check whether $\langle i, x\rangle \in L_A$ within $2^{O(n)}$ steps (note that we need to simulate $M_i^{A^{\leq n}}(\langle i, x\rangle)$, but not $M_i^A(\langle i, x\rangle)$). Thus, $L_A \in EXP$.

Let M_i be a polynomial time bounded oracle machine such that $L_A = L(M_i, A)$. Let x be a sufficiently long string so that $|i|\cdot p_i(n)^2 \leq 2^n$, where $n = |\langle i, x\rangle|$. Then there exists at least one element y in $Q(M_i, A, \langle i, x\rangle) \cap A$ such that $|y| > |x|$.

Suppose otherwise. That is, assume that the length of each element of $Q(M_i, A, \langle i, x\rangle) \cap A$ is less than or equal to $|x|$. Then $M_i^{A^{\leq n}}(\langle i, x\rangle) = M_i^A(\langle i, x\rangle)$, so $\langle i, x\rangle \in L(M_i, A^{\leq n})$ iff $\langle i, x\rangle \in L(M_i, A)$. Thus, $\langle i, x\rangle \in L_A$ if and only if $\langle i, x\rangle \notin L(M_i, A)$. This contradicts the fact that $L_A = L(M_i, A)$. □

Corollary 3.3. For any polynomial time reducibility $\leq_r^{\mathrm{P}}$, let A be any $\leq_r^{\mathrm{P}}$-complete set in EXP. Then A has an $\mathcal{F}$-easy subset, where $\mathcal{F}$ stands for the following classes depending on the reduction type $\leq_r^{\mathrm{P}}$ respectively: FT for $\leq_{\mathrm{T}}^{\mathrm{P}}$, Ftt for $\leq_{\mathrm{tt}}^{\mathrm{P}}$, F$p$-tt for $\leq_{p(n)\text{-tt}}^{\mathrm{P}}$, Fbtt for $\leq_{\mathrm{btt}}^{\mathrm{P}}$, FP for $\leq_{\mathrm{c}}^{\mathrm{P}}$ and $\leq_{\mathrm{m}}^{\mathrm{P}}$.

Proof. First we prove for $\leq_{\mathrm{T}}^{\mathrm{P}}$-reducibility; let A be any $\leq_{\mathrm{T}}^{\mathrm{P}}$-complete set in EXP. For this set A we consider the set L_A which is defined in Theorem 3.2.

Since $L_A \in$ EXP and A is $\leq_{\mathrm{T}}^{\mathrm{P}}$-hard for EXP, there exists a polynomial time oracle machine M_i such that $L_A = L(M_i, A)$. We can also assume that $Q(M_i, A, z) \neq \emptyset$ for every $z \in \Sigma^*$. Then it follows from Theorem 3.2 that for almost all x there exists y_x in $Q(M_i, A, \langle i, x\rangle) \cap A$ such that $|y_x| > |x|$.

Define g by

$$g(x) = \begin{cases} y_x & \text{if } y_x \text{ exists,} \\ \text{some fixed elements of } Q(M_i, A, \langle i, x\rangle) & \text{otherwise.} \end{cases}$$

(*Note:* In the above definition we do not explicitly state the way of choosing y_x from $\{y : y \in Q(M_i, A, \langle i, x\rangle) \cap A$ *and* $|y| > |x|\}$. We may use an arbitrary way even though it is non-recursive.)

Then g is total, and for almost all x, $|g(x)| > |x|$ and $g(x) \in A$. Note also that g is a function in FT since there is a polynomial time bounded oracle machine M such that $Q(M, A, x) = Q(M_i, A, \langle i, x\rangle)$ for all x: that is, g is generated by M and A. Therefore A has an FT-easy subset w.r.t. this function g.

Next consider the other type of reducibilities. For reducibilities such as $\leq^{\mathrm{p}}_{\mathrm{tt}}$, $\leq^{\mathrm{p}}_{p(n)\text{-tt}}$, $\leq^{\mathrm{p}}_{\mathrm{btt}}$ and $\leq^{\mathrm{p}}_{\mathrm{m}}$, the proof is similar to the above if we note that each type of polynomial time reductions is a special case of $\leq^{\mathrm{p}}_{\mathrm{T}}$-reductions. So, the details of the proof are left to the reader. In the following we consider the case of $\leq^{\mathrm{p}}_{\mathrm{c}}$-reducibility.

Let A be any $\leq^{\mathrm{p}}_{\mathrm{c}}$-complete set in EXP. Then we first show that A has an Fc-easy subset where the class Fc is defined as follows.

$$\mathrm{Fc} = \{g : (\exists f\text{: conjunctive tt-function in FP})(\forall x \in \Sigma^*)$$
$$[\, g(x) = \text{the (lexicographically) largest element of } \mathit{Ass}(f, x)\ ]\,\}.$$

For the set A, consider the set L_A which is constructed in Theorem 3.2. Also consider a polynomial time conjunctive tt-reduction f from L_A to A. Let M_i be a polynomial time bounded oracle machine which witnesses the reduction f. Then for almost all x, $Q(M_i, A^{\leq n}, \langle i, x\rangle) \cap A^{>n} \neq \emptyset$ where $n = |x|$. So $\langle i, x\rangle$ is not in $L(M_i, A^{\leq n})$ because the reduction f is conjunctive. Thus it follows from the definition of L_A that $\langle i, x\rangle \in L_A$. Hence, for almost all x, $Q(M_i, A, \langle i, x\rangle) \subseteq A$ since f is conjunctive. Moreover, for almost all x, there exists some y in $Q(M_i, A, \langle i, x\rangle)$ such that $|y| > |x|$. Therefore A has an Fc-easy subset w.r.t. the following function: $g(x)$ = the largest element of $Q(M_i, A, \langle i, x\rangle)$.

Notice here that the class Fc is the same as the class of total functions in FP. Therefore we conclude that every $\leq_{c}^{p}$-complete set has an FP-easy subset. □

At this point we have some useful properties of polynomial time complete sets in EXP. For example, all $\leq_{m}^{p}$-complete sets for EXP have FP-easy subsets. So, in order to construct a $\leq_{btt}^{p}$-complete set which is not $\leq_{m}^{p}$-complete, we need to obtain a $\leq_{btt}^{p}$-complete set which does not have any FP-easy subset; this can be done by simple diagonalization.

Theorem 3.4.

(1) There exists a $\leq_{2\text{-d}}^{p}$-complete (thus, $\leq_{2\text{-tt}}^{p}$-complete) set A in EXP which has no FP-easy subset. Thus, A is not $\leq_{m}^{p}$-complete (nor $\leq_{c}^{p}$-complete) in EXP;

(2) For any polynomial p and q such that $p(n) > q(n) \geq 0$ for almost all n, there exists a $\leq_{p(n)\text{-d}}^{p}$-complete (thus, $\leq_{p(n)\text{-tt}}^{p}$-complete) set B which has no F$q$tt-easy subset. Thus, B is not $\leq_{q(n)\text{-tt}}^{p}$-complete in EXP; and

(3) There exists a $\leq_{d}^{p}$-complete (thus, $\leq_{tt}^{p}$-complete) set C which has no Fbtt-subset. Thus, C is not $\leq_{btt}^{p}$-complete in EXP.

The proofs for (1), (2) and (3) are similar; so we will show the most difficult one, i.e., the proof of (3). We will describe the set C by a *stage construction* (see, e.g., [BS85, Ro67, Wa85]). In a stage construction, a set L is defined in stages. Let b be a function from **N** to **N**. At each stage $n \geq 0$, the value $b(n)$ and an initial segment $L_n = \{x \in L : b(n-1) < |x| \leq b(n)\}$ of L are defined in terms of algorithms on Σ^*. The set L is then defined by $\cup_{n\geq 0} L_n$. Note that if L can be defined by a stage construction in which, at each stage $n \geq 0$, the value $b(n)$ and the set L_n are defined, and the deterministic running time of an algorithm which computes $b(n)$ and L_n is bounded by $T(n)$, then $L \in DTIME(T(n)\cdot 2^{cn})$ for some $c > 0$.

Proof of Theorem 3.4 (3).

We use an enumeration of polynomial time bounded deterministic transducers; let $\{N_i\}_{i>0}$ denote it. We assume that for any total function f in FP there exist some polynomial p and infinitely many indices $i > 0$ such that N_i is a $p(n)$ time bounded transducer which computes f. Also we use a standard $\leq_m^p$-complete set in EXP; let $U \subseteq \Sigma^*$ be a standard $\leq_m^p$-complete set in EXP. What follows is an explanation of the idea of how to construct the desired set C.

For a string x and an integer i, $1 \leq i \leq |x|$, let $c(i, x)$ be $0^m\text{bin}(i)x$ where $\text{bin}(i)$ is the binary representation of i and $|0^m\text{bin}(i)x| = 2|x|$. We put at least one element of $U_x = \{c(i, x) : 1 \leq i \leq |x|\}$ into C for all and only x in U so that the set C can be $\leq_d^p$-complete in EXP. On the other hand, for any total function g in Fbtt such that $|g(z)| > |z|$ for almost all z, we need to put $g(x)$ into C^c for infinitely many x so that C can have no Fbtt-easy subset. For this purpose, we do not put any elements of $Y = \{y \in Ass(N_n, 0^{n'}) : |y| > n'\}$ into C_n at stage n, where $n' = b(n - 1)$ (recall that $Ass(N, x)$ denotes the associated set of tt-condition $N(x)$). These two requests do not conflict if $\|U_x\| > |Y|$ (or equivalently, if $b(n - 1)/2 \geq |Y|$). Thus the construction becomes as follows:

$C = \cup_{n\geq 0} C_n$, where

```
**stage** $n = 0$;
  $b(0) \leftarrow 0$; $C_0 \leftarrow \emptyset$;
**stage** $n > 0$;
  $n' \leftarrow b(n-1)$;
  $Y \leftarrow \{y \in Ass(N_n, 0^{n'}) : |y| > n'\}$;
  (if this computation needs more than $2^{n'}$ steps, then $Y \leftarrow$ **undef**)
```

```
    if $Y \neq$ undef and $0 < \|Y\| \leq b(n-1)/2$ then
      $b(n) \leftarrow$ the length of longest element of $Y$;
      $C_n \leftarrow \{c(i, x) : 1 \leq i \leq |x|, b(n-1) < 2|x| \leq b(n)$ and $x \in U\} - Y$;
      (we say that $N_n$ is diagonalized here)
    else
      $b(n) \leftarrow b(n-1) + 2$;
      $C_n \leftarrow \{c(0, x) : b(n-1) < 2|x| \leq b(n)$ and $x \in U\}$
    end-if
  end-construction.
```

It is clear that $C \in$ EXP. Suppose that C has an Fbtt-easy subset by means of a function g in Fbtt. Then there exists an integer $k > 0$, and a k-tt-function f in FP such that $g(x) \in Ass(f, x)$ for all x. Since $\|Ass(f, x)\| \leq k$, there exist infinitely many n such that $f = N_n$ and that N_n is diagonalized in the above construction. Thus there exist infinitely many n' such that $Y = \{y \in Ass(f, 0^{n'}) : |y| > n'\}$ has no element in C. So $g(0^{n'})$ is not in C or $|g(0^{n'})| \leq n'$ for such n'. Which contradicts the assumption that $g(x) \in C$ and $|g(x)| > |x|$ for almost all x.

From the construction of the set C, it is easy to show that the following function f is a $\leq^{\mathrm{p}}_{\mathrm{d}}$-reduction from U to C:

for every x, $f(x) = \langle\langle c(1, x), \cdots, c(m, x)\rangle, \alpha\rangle$
where $\alpha(b_1, \cdots, b_m) = b_1 \vee \cdots \vee b_m$, and $m = |x|$.

Therefore the set C is $\leq^{\mathrm{p}}_{\mathrm{c}}$-complete in EXP, which completes the proof. □

In order to show the differences between polynomial time truth-table completeness notions including conjunctive type truth-table reductions, we prove the following lemma.

Lemma 3.5.

Let A be a $\leq^{\mathrm{p}}_{\mathrm{tt}}$- (resp., $\leq^{\mathrm{p}}_{p(n)\text{-tt}}$-) complete set in EXP. Then A^{c} is also $\leq^{\mathrm{p}}_{\mathrm{tt}}$- (resp., $\leq^{\mathrm{p}}_{p(n)\text{-tt}}$-) complete in EXP. In particular, if A is $\leq^{\mathrm{p}}_{\mathrm{d}}$- (resp., $\leq^{\mathrm{p}}_{p(n)\text{-d}}$-)

complete, then A^c is $\le^p_c$- (resp., $\le^p_{p(n)\text{-c}}$-) complete, and vice versa.

Proof. Let U be a standard $\le^p_m$-complete set in EXP, and f be a polynomial time $\le^p_{tt}$-reduction from U to A. Then the following polynomial time computable tt-function g is a reduction from U^c to A^c:

for every $x \in \Sigma^*$, $g(x) = \langle\langle a_1, \cdots, a_k\rangle, \beta\rangle$,
where
(1) $\langle\langle a_1, \cdots, a_k\rangle, \alpha\rangle = f(x)$, and
(2) $\beta(b_1, \cdots, b_k) = \neg\alpha(\neg b_1, \cdots, \neg b_k)$
(by "$\neg R$" we mean "not R").

Note that U^c is $\le^p_m$-complete in EXP. Thus A^c is $\le^p_{tt}$-complete in EXP. Especially, g is conjunctive (resp., disjunctive) if f is disjunctive (resp., conjunctive). □

Then the following theorem follows from Theorem 3.4 and Lemma 3.5.

Corollary 3.6. Let p and q be any polynomials such that $p(n) > q(n) \ge 0$ for almost all n. Then, for each condition below, there exists a set in EXP which satisfies that condition:

(1) $\le^p_{p(n)\text{-}d}$-complete (thus, $\le^p_{p(n)\text{-tt}}$-complete) but not $\le^p_{q(n)\text{-}tt}$-hard;

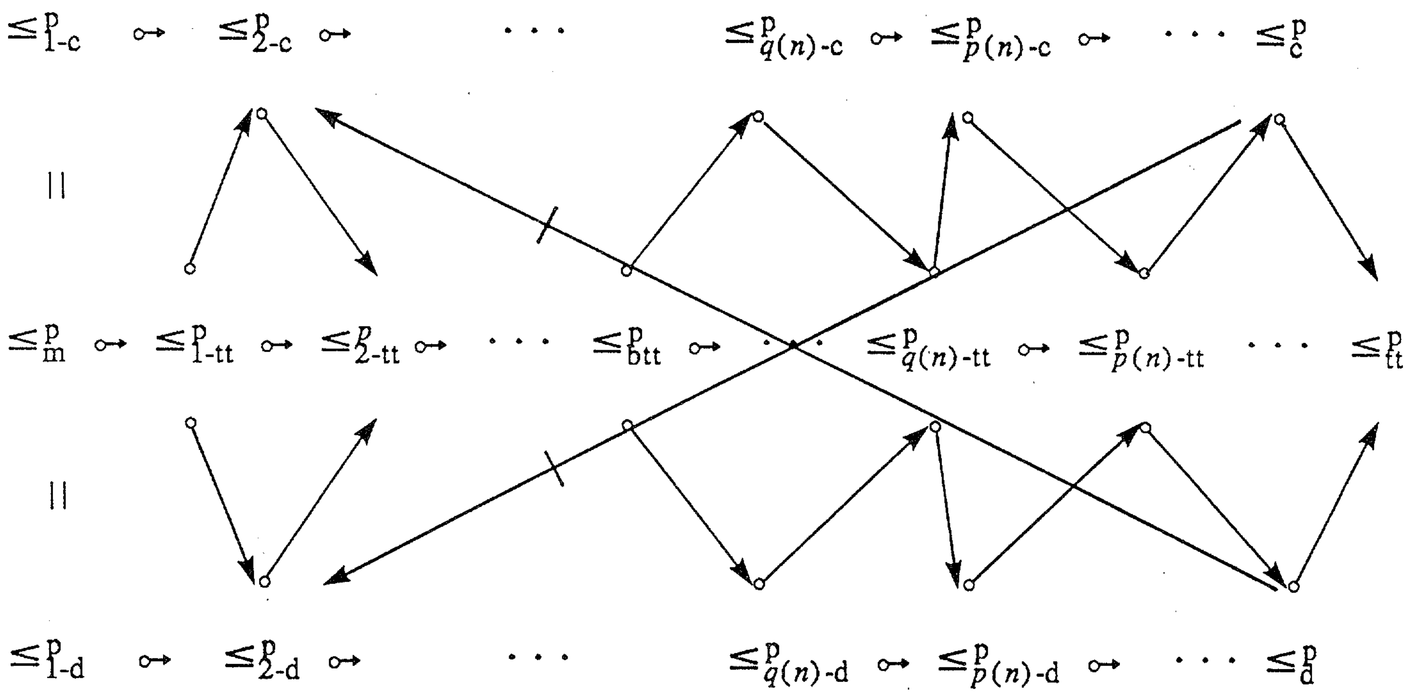
(2) $\le^p_{p(n)\text{-}c}$-complete (thus, $\le^p_{p(n)\text{-tt}}$-complete) but not $\le^p_{q(n)\text{-}tt}$-hard;

(3) $\le^p_{2\text{-}d}$-complete but not $\le^p_c$-hard; and

(4) $\le^p_{2\text{-}c}$-complete but not $\le^p_d$-hard.

Remark. Since $\le^p_m$-reducibility is equivalent to $\le^p_{1\text{-c}}$- and $\le^p_{1\text{-d}}$-reducibilities, $\le^p_m$-, $\le^p_{1\text{-c}}$-, $\le^p_{1\text{-d}}$-completeness notions are all equivalent. □

Hence, among all the possible pairs of reduction types considered here, only the following comparison remain open: for every $k \ge 1$, whether $\le^p_{k\text{-}tt}$-completeness implies $\le^p_{(k\text{-})c}$- (resp., $\le^p_{(k\text{-})d}$-) completeness (we conjecture not). That is, we have Figure 3.2 on comparison of polynomial time truth-table


$\leq^{\mathrm{p}}_{1\text{-c}}$ ⊶ $\leq^{\mathrm{p}}_{2\text{-c}}$ ⊶ . . . $\leq^{\mathrm{p}}_{q(n)\text{-c}}$ ⊶ $\leq^{\mathrm{p}}_{p(n)\text{-c}}$ ⊶ . . . $\leq^{\mathrm{p}}_{\mathrm{c}}$
||
$\leq^{\mathrm{p}}_{\mathrm{m}}$ ⊶ $\leq^{\mathrm{p}}_{1\text{-tt}}$ ⊶ $\leq^{\mathrm{p}}_{2\text{-tt}}$ ⊶ . . . $\leq^{\mathrm{p}}_{\mathrm{btt}}$ ⊶ . . . $\leq^{\mathrm{p}}_{q(n)\text{-tt}}$ ⊶ $\leq^{\mathrm{p}}_{p(n)\text{-tt}}$ . . . $\leq^{\mathrm{p}}_{\mathrm{tt}}$
||
$\leq^{\mathrm{p}}_{1\text{-d}}$ ⊶ $\leq^{\mathrm{p}}_{2\text{-d}}$ ⊶ . . . $\leq^{\mathrm{p}}_{q(n)\text{-d}}$ ⊶ $\leq^{\mathrm{p}}_{p(n)\text{-d}}$ ⊶ . . . $\leq^{\mathrm{p}}_{\mathrm{d}}$

Where

$p(n) > q(n) > 0$ for almost all n;

$\alpha \not\to \beta$: $A\ \alpha\ B$ does not imply $A\ \beta\ B$;

α ⊶ β : $A\ \alpha\ B$ implies $A\ \beta\ B$ but the converse is false; and

$\alpha = \beta$: α and β are the same, i.e., $A\ \alpha\ B$ implies $A\ \beta\ B$ *and* $A\ \beta\ B$ implies $A\ \alpha\ B$.

(a relation deduced from the others may be omitted)

Figure 3.2. The comparison of several completeness notions in EXP.

completeness notions.

Other Complexity Classes

All the theorems in the above discussion also holds for any super-SUBEXP and any super-PSPACE complexity classes which have a standard $\leq_{m}^{p}$-complete set. And the proof of Theorem 3.1 also works for nondeterministic time complexity classes such as NEXP. However, the proof techniques used in Theorem 3.2 and Corollary 3.3 are not applicable to nondeterministic time complexity classes. In the following we consider similar results for NEXP.

Note that for Theorem 3.4 the same proof works also for NEXP. Thus, concerning the differences between completeness notions, we have the following proposition as an immediate corollary of Theorem 3.4 (1).

Proposition 3.7. If all $\leq_{m}^{p}$-complete sets in NEXP have FP-easy subsets, then there exists a $\leq_{btt}^{p}$-complete set in NEXP which is not $\leq_{m}^{p}$-hard for NEXP.

Remark. We have several similar statements concerning the differences between completeness notions; but they are omitted. □

Unfortunately we have not obtained any proofs of differences from this proposition so far. Instead, using a similar proof strategy we show the difference between $\leq_{m}^{p}$-completeness and $\leq_{btt}^{p}$-completeness in NEXP.

According to an argument similar to the proof of Theorem 3.4, we have the following lemma.

Lemma 3.8. There exists a $\leq_{2\text{-}c}^{p}$-complete set D in NEXP whose complement does not include a range of any one-to-one and total functions in FP, i.e., for all one-to-one and total functions f in FP, $f(\Sigma^{*}) \not\subset D^{c}$.

Proof. The proof is similar to the one for Theorem 3.5 (1), so we omit it here (see also [Theorem 4.2 of Wa85]). □

The following lemma is also provable (see Corollary 4.10).

Lemma 3.9. For all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets L in NEXP, there exists a one-to-one function f in FP such that $f(\Sigma^*) \subset L^{\mathrm{c}}$. □

Therefore we show the difference between $\leq^{\mathrm{p}}_{\mathrm{m}}$-completeness and $\leq^{\mathrm{p}}_{\text{2-c}}$-completeness (thus, $\leq^{\mathrm{p}}_{\mathrm{btt}}$-completeness) on NEXP.

Theorem 3.10. There exists a $\leq^{\mathrm{p}}_{\text{2-c}}$-complete set in NEXP which is not $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete. □

3.2. $\leq^{\mathrm{p}}_{\mathrm{tt}}$-Completeness and $\leq^{\mathrm{p}}_{\mathrm{T}}$-Completeness Notions

We have compared the completeness notions which are weaker than $\leq^{\mathrm{p}}_{\mathrm{tt}}$-completeness. The remainder notion which we need to consider here is $\leq^{\mathrm{p}}_{\mathrm{T}}$-completeness. From now on, we will investigate the difference between $\leq^{\mathrm{p}}_{\mathrm{tt}}$- and $\leq^{\mathrm{p}}_{\mathrm{T}}$-completeness notions: i.e., we will construct a set A which is $\leq^{\mathrm{p}}_{\mathrm{T}}$-complete but not $\leq^{\mathrm{p}}_{\mathrm{tt}}$-complete in EXP. The outline of our idea is as follows.

Define a sequence $\{\mu_m\}_{m>0}$, where each $\mu_m \in \{0, 1\}^*$ and $|\mu_m| = m$, having the following property: (*) $\mu_m \in \mathrm{K}[n/2, 2^n]^{\mathrm{c}}$. That is, μ_m is not producible from any short description (i.e., less than $m/2$ bits) within 2^m steps (the reader may wish to review Chapter 2 for the concept of Kolmogorov complexity).

We will use this sequence to construct the set A. Consider the set $V = \{\mu_m x : x \in U$ and $m = \lceil \log |x| \rceil^2\}$, where U is a standard $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set in EXP. We know whether $x \in U$ by checking V to determine whether $\mu_m x \in V$. But it may be difficult to produce μ_m from x in general. So we introduce the set $E = \{\langle i, 0^n \rangle :$ the ith bit of μ_m is '1', where $m = \lceil \log n \rceil^2\}$, and define $A = V \cup E$ (here we can assume that $V \cap E = \emptyset$). Then it is easy to construct a polynomial time oracle machine which accepts U relative to A. That is, A is $\leq^{\mathrm{P}}_{\mathrm{T}}$-complete in EXP. Next consider $\leq^{\mathrm{P}}_{\mathrm{tt}}$-hardness of the set A. Note that $E \in DTIME(2^{c(\log n)^2})$ and $DTIME(2^{c(\log n)^2}) \subsetneq$ EXP. Thus, it is provable that E itself cannot be $\leq^{\mathrm{P}}_{\mathrm{tt}}$-hard in EXP. Also from the property (*) of $\{\mu_m\}_{m>0}$, all $\leq^{\mathrm{P}}_{\mathrm{tt}}$-reductions cannot query "$\mu_m x$" to V on some type of inputs, which proves that for such inputs, information from V cannot be used by $\leq^{\mathrm{P}}_{\mathrm{tt}}$-type (non additive type) reductions. Therefore A $(= V \cup E)$ is not $\leq^{\mathrm{P}}_{\mathrm{tt}}$-hard in EXP.

Now we proceed to the precise argument. First, we define the sequence $\{\mu_m\}_{m>0}$ and investigate its computational complexity.

Define J to be $\mathrm{K}[n/2, 2^n]^{\mathrm{c}}$. Then we have the following facts:

Lemma 3.11.

(1) $J \cap \{0, 1\}^m \neq \varnothing$ for all $m > 0$.

(2) J is in EXP. □

For any integer $m > 0$, define μ_m to be the (lexicographically) smallest element of $J \cap \{0, 1\}^m$. Define $E \subseteq \mathbf{N} \times \Sigma^*$ by

$$\langle i, 0^n \rangle \in E \iff \text{the } i\text{th bit of } \mu_m \text{ is '1', where } m = \lceil \log n \rceil^2.$$

And define V by

$$V = \{\mu_m x : x \in U \text{ and } m = \lceil \log |x| \rceil^2\}.$$

Here we can assume, without loss of generality, that each element of E has even length, and that each element of V has odd length (for example, by putting a padding bit at the end). That is, we can assume that $V \cap E = \varnothing$. Finally, define $A = V \cup E$.

Then it is easy to show the following facts.

Lemma 3.12.

(1) $\{\mu_m : m > 0\}$ is in EXP.

(2) E is in $DTIME(2^{c(\log n)^2})$ for some $c > 0$. □

Therefore we have the following lemma:

Lemma 3.13. The set A is $\le^{\mathrm{p}}_{\mathrm{T}}$-complete in EXP.

Proof. It follows from Lemma 3.13 that $A \in$ EXP. The $\le^{\mathrm{p}}_{\mathrm{T}}$-hardness of A is straight forward and omitted here. □

In the following we show that A is not $\le^{\mathrm{p}}_{\mathrm{tt}}$-hard for EXP. Let $\{N_i\}_{i>0}$ denote an enumeration of polynomial time bounded transducers. Then for every $\le^{\mathrm{p}}_{\mathrm{tt}}$-reduction, there exists some N_i which achieves it. The following lemma states that such an N_i cannot query "$\mu_m w$" on some type of inputs.

Lemma 3.14. For all $\le^{\mathrm{p}}_{\mathrm{tt}}$-reduction N_i, we have the following property:

$$(\overset{\infty}{\forall} n)(\forall a \in Ass(N_i, \langle i, 0^n\rangle))$$
$$[\, a \text{ is not of the form } \mu_m w \text{ for some } m \ge \lceil \log n - 1\rceil^2 \text{ and } w \in \Sigma^* \,].$$

Proof. Suppose that there exists a $\le^{\mathrm{p}}_{\mathrm{tt}}$-reduction N_{i_0} that does not satisfy the above property: that is, for such an N_{i_0} and for infinitely many n,

$$(\exists a \in Ass(N_i, \langle i, 0^n\rangle))[\, a = \mu_m w \text{ for some } m \ge \lceil \log n - 1\rceil^2 \text{ and } w \in \Sigma^* \,].$$

Consider the following procedure:

```
procedure printμ($n$, $j$, $k$: integer);
    $\langle\langle a_1, \cdots, a_r\rangle, \alpha\rangle \leftarrow N_{i_0}(\langle i, 0^n\rangle)$;
    $\mu \leftarrow$ the first $k$ bits of $a_j$;
    output($\mu$)
end-procedure.
```

Let $p(n)$ be a polynomial time bound for N_{i_0}. Then for infinitely many n, printμ(n, j, k) outputs some μ_m, where $m \ge \lceil \log n - 1\rceil^2$, within $O(p(n))$ steps for some j, $1 \le j \le p(n)$, and some k, $\lceil \log n - 1\rceil^2 \le k \le p(n)$. For such integers j, k, n and m, we have

(1) $|n| + |j| + |k| \le |n| + 2|p(n)| = O(\log n) = O(m^{1/2})$; and

(2) $p(n) \leq p(2^{m^{1/2}}) = 2^{O(m^{1/2})}$.

Let N_μ be some deterministic transducer which achieves the computation of printμ. Then there exists some c_1 and c_2 such that for infinitely many m, we have $\mu_m \in \mathrm{K}_\mu[c_1 n^{1/2}, 2^{c_2 n^{1/2}}]$. Hence it follows from Proposition 2.2 that for infinitely many m, we have $\mu_m \in \mathrm{K}[n/2, 2^n]$. This contradicts our definition of the sequence $\{\mu_m\}_{m>0}$. □

Theorem 3.2 plays an important role in the comparison of several truth-table completeness notions. Here we use a similar lemma. The following result is stronger than the one we need; however, it illustrates the underlying ideas more clearly.

Lemma 3.15. Consider an enumeration $\{N_i\}_{i>0}$ of polynomial time bounded transducers. Let t_1 and t_2 be time constructible functions, and let X and Y be sets in $DTIME(t_1)$ and $DTIME(t_2)$ respectively. Also let τ be a super polynomial running time. Then there exists a set L_{XY} which satisfies the following conditions:

(a) $L_{XY} \in DTIME(t)$, where $t(n) = \tau(n) \cdot (1 + t_1(n) + t_2 \circ \tau(n))$; and

(b) for all polynomial time $\leq_{\mathrm{tt}}^{\mathrm{P}}$-reduction N_i and for almost all n, we have the following (**) for all x, $|x| = n$.

(**) $\langle i, x \rangle \in L_{XY} \leftrightarrow \alpha(b_1, \cdots, b_r) = \textbf{false}$,

where

(1) $N_i(\langle i, x \rangle) = \langle\langle a_1, \cdots, a_r \rangle, \alpha\rangle$, and

(2) for every j, $1 \leq j \leq$ r, $b_j = \textbf{true}$ iff $a_j \in X^{\leq n} \cup Y^{>n}$.

Remark. The statement (**) means that $\langle i, x \rangle$ is a witness to the falsify of $L_{XY} \leq_{\mathrm{tt}}^{\mathrm{P}} (X^{\leq n} \cup Y^{>n})$ by N_i.

Proof. First define $L'_{XY} \subseteq \mathrm{N} \times \Sigma^*$ as follows:

$\langle i, x \rangle \in L'_{XY} \iff \alpha(b_1, \cdots, b_r) = \textbf{false}$, where

(1) $N_i(\langle i, x\rangle) = \langle\langle a_1, \cdots, a_r\rangle, \alpha\rangle$, and

(2) for every j, $1 \le j \le r$, b_j = **true** iff $a_j \in X^{\le n} \cup Y^{>n}$.

Then it is clear that L'_{XY} satisfies the condition (b). Consider an acceptor for the set L'_{XY}. Let M_X (resp., M_Y) be $t_1(n)$ (resp., $t_2(n)$) time bounded machine which accepts X (resp., Y). Recall that the universal transducer N_U simulates every transducer N_i. Hence, using M_X, M_Y and N_U, we can construct a machine M' which accepts L'_{XY} and whose running time on $\langle i, x\rangle$ is bounded by $|i| \cdot p_i(|\langle i, x\rangle|)^2 + p_i(|\langle i, x\rangle|) \cdot (t_1(n) + t_2 \circ p_i(|\langle i, x\rangle|)$ steps, where $n = |x|$ and p_i is a polynomial time bound for N_i. Note that M' is not $t(n)$ time bounded.

Next consider the following set L_{XY}:

$\langle i, x\rangle \in L_{XY}$ <=> M' halts within $t(|\langle i, x\rangle|)$ steps *and* $\langle i, x\rangle \in L'_{XY}$.

Then L_{XY} satisfies the condition (a). Also it is easy to show that for all polynomial time tt-function N_i and for almost all x, $\langle i, x\rangle \in L'_{XY}$ if and only if $\langle i, x\rangle \in L_{XY}$. Thus, L_{XY} still satisfies the condition (b). Therefore L_{XY} is the desired set. □

Corollary 3.16. Let X be a set in EXP and Y be a set in $DTIME(2^{c(\log n)^2})$. Then there exists a set L_0 in EXP which satisfies the following: for all $\le^{\mathrm{p}}_{\mathrm{tt}}$-reduction N_i and for almost all n, we have

$\langle i, 0^n\rangle \in L_0 \leftrightarrow \alpha(b_1, \cdots, b_r)$ = **false**,

where

(1) $N_i(\langle i, 0^n\rangle) = \langle\langle a_1, \cdots, a_r\rangle, \alpha\rangle$, and

(2) for every j, $1 \le j \le r$, b_j = **true** iff $a_j \in X^{\le n} \cup Y^{>n}$.

That is, $\langle i, 0^n\rangle$ is a witness that L is not $\le^{\mathrm{p}}_{\mathrm{tt}}$-reducible to $X^{\le n} \cup Y^{>n}$ by the reduction N_i.

Proof. Let $t_1(n) = 2^{c_1 n}$, $t_2(n) = 2^{c_2(\log n)^2}$, and $\tau(n) = 2^{(\log n)^2}$ in Lemma 3.15. Then we obtain the desired set L_0 in EXP. □

Lemma 3.17. The set A is not $\le^{\mathrm{p}}_{\mathrm{tt}}$-hard for EXP.

Proof. Consider Corollary 3.16 where X is the set A and Y is the set E. Then we obtain the set L_0 in EXP. Suppose by the way of contradiction that A is $\le^{\mathrm{p}}_{\mathrm{tt}}$-hard for EXP. Then there exists a $\le^{\mathrm{p}}_{\mathrm{tt}}$-reduction N_{i_0} from L_0 to A.

Corollary 3.12 shows that for almost all n, $\langle i_0, 0^n\rangle$ is a witness that L_0 is not $\le^{\mathrm{p}}_{\mathrm{tt}}$-reducible to $A^{\le n} \cup E^{>n}$ by N_{i_0}. Note that every element of $V^{>n}$ is of the form $\mu_m x$ and $|\mu_m| > n$; thus $m \ge \lceil \log n - 1\rceil^2$ for sufficiently large n. But it follows from Lemma 3.14 that for almost all n, $Ass(N_{i_0}, \langle i_0, 0^n\rangle)$ does not have any element of the form $\mu_m x$ where $m \ge \lceil \log n - 1\rceil^2$. So, $Ass(N_{i_0}, \langle i_0, 0^n\rangle) \cap V^{>n} = \varnothing$. Hence, for sufficiently large n, we have

$\langle i_0, 0^n\rangle \in L_0 \leftrightarrow \alpha(b_1, \cdots, b_r) =$ **false**,

where

(1) $N_{i_0}(\langle i_0, 0^n\rangle) = \langle\langle a_1, \cdots, a_r\rangle, \alpha\rangle$; and

(2) for every j, $1 \le j \le r$, $b_j =$ **true** iff $a_j \in A^{\le n} \cup E^{>n}$.

$\leftrightarrow \alpha(b'_1, \cdots, b'_r) =$ **false**,

where

(1) $N_{i_0}(\langle i_0, 0^n\rangle) = \langle\langle a_1, \cdots, a_r\rangle, \alpha\rangle$; and

(2) for every j, $1 \le j \le r$, $b'_j =$ **true** iff $a_j \in A^{\le n} \cup E^{>n} \cup V^{>n}$.

Thus, $\langle i_0, 0^n\rangle$ is a witness that L_0 is not $\le^{\mathrm{p}}_{\mathrm{tt}}$-reducible to $A^{\le n} \cup E^{>n} \cup V^{>n}$ by N_{i_0}. Note that $A^{\le n} \cup E^{>n} \cup V^{>n} = A^{\le n} \cup A^{>n} = A$ so that we have a contradiction. □

Now we obtain the following theorem as an immediate consequence of Lemma 3.13 and 3.17.

Theorem 3.18. There exists a $\le^{\mathrm{p}}_{\mathrm{T}}$-complete set in EXP which is not $\le^{\mathrm{p}}_{\mathrm{tt}}$-hard for EXP. □

We have considered the differences between several completeness notions. Here we review these results from a different point of view.

As we have investigated, a notion of "polynomial time reducibility" offers a way of comparing intractability between given two sets; and each notion of reducibility offers different type of comparison (Theorem 3.1). Then consider the equivalence relations induced by these reducibilities. For example, consider any polynomial time reducibility $\leq_r^{\mathrm{p}}$ and any two sets A and B such that $A \equiv_r^{\mathrm{p}} B$ (recall that $A \equiv_r^{\mathrm{p}} B$ means both $A \leq_r^{\mathrm{p}} B$ and $B \leq_r^{\mathrm{p}} A$). Then the assertion $A \equiv_r^{\mathrm{p}} B$ states that A and B has a similar (degree of) intractability in some sense. Theorem 3.1 lets us conjecture that each type of reducibility offers different "similarity": e.g., $\equiv_{\mathrm{m}}^{\mathrm{p}}$-equivalence is closer relation than $\equiv_{\mathrm{T}}^{\mathrm{p}}$-equivalence. We investigate it in EXP.

In order to investigate the above problem, we introduce a new notion, *polynomial time degree*.

Definition 3.2.

(1) For any $\leq_r^{\mathrm{p}}$-reducibility, a class of all $\equiv_r^{\mathrm{p}}$-equivalent sets is $\equiv_r^{\mathrm{p}}$*-degree*. That is, one $\equiv_r^{\mathrm{p}}$-degree is a class of sets which has the same *degree* of difficulty w.r.t. $\leq_r^{\mathrm{p}}$-reducibility.

(2) For any two $\equiv_r^{\mathrm{p}}$-degrees $\mathcal{D}_1$ and $\mathcal{D}_2$, $\mathcal{D}_1$ is *higher* than $\mathcal{D}_2$ if all sets in $\mathcal{D}_2$ is $\leq_r^{\mathrm{p}}$-reducible to some sets in $\mathcal{D}_1$.
(*Note:* For any complexity class $\mathcal{C}$, the class of $\leq_r^{\mathrm{p}}$-complete sets in $\mathcal{C}$ is the highest $\leq_r^{\mathrm{p}}$-degree in $\mathcal{C}$.)

Remark. Following [Ro67, MY85] we will use the term "$\equiv_r^{\mathrm{p}}$*-type*" instead of "$\equiv_r^{\mathrm{p}}$-degree" for any $\leq_r^{\mathrm{p}}$-reducibility which is stronger than $\leq_{\mathrm{m}}^{\mathrm{p}}$-reducibility (such an $\leq_r^{\mathrm{p}}$-reducibility will be used in the next chapter).

In general the problem of interest is how polynomial time degrees divide the class EXP. And the problem we proposed above is whether each polynomial time reducibility offers a different way of dividing EXP; especially whether a stronger polynomial time reducibility gives smaller equivalence classes (i.e., polynomial time degrees). Note here that the class P consists of one $\equiv_r^p$-degree for all $\leq_r^p$-reducibilities; thus, the lowest polynomial time degrees are the same. On the other hand, since the class of $\leq_r^p$-complete sets in EXP is the highest $\equiv_r^p$-degree in EXP, our comparisons on completeness notions show that the highest polynomial time degree differs depending on reduction types. For example, we have the following picture concerning the structure of EXP.

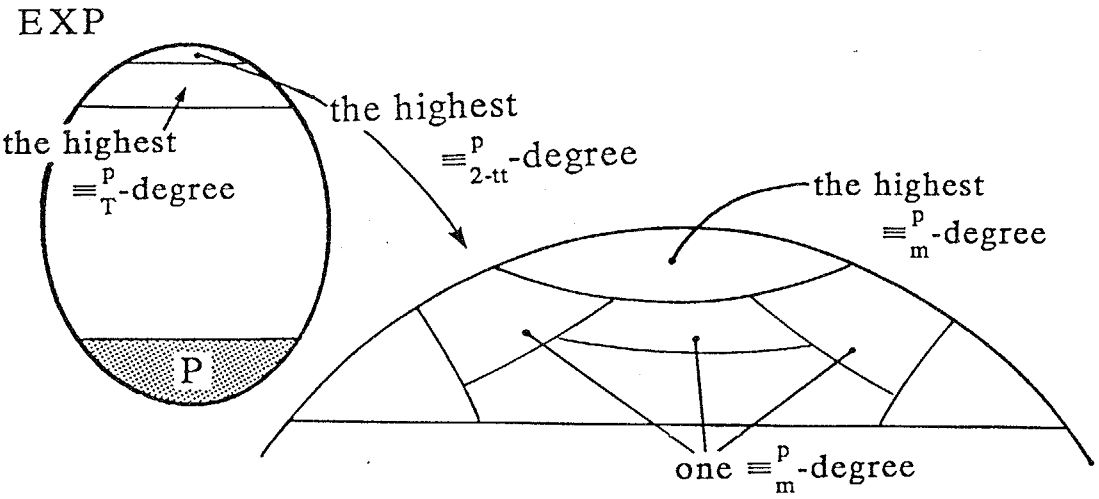

EXP
the highest $\equiv_T^p$-degree
P
the highest $\equiv_{2\text{-tt}}^p$-degree
the highest $\equiv_m^p$-degree
one $\equiv_m^p$-degree

Figure 3.3. The structure of EXP w.r.t. polynomial time degrees.

4. P-Isomorphism of $\leq_m^p$-Complete Sets

In the early 1970's [Co71, Ka72], Cook found SAT, the first NP-complete set (i.e., $\leq_m^p$-complete set in NP). Since then, a huge number of NP-complete sets have been discovered (e.g., see [GJ79]). They have come from wide variety of problem areas: some concern theorem proving; some concern a property on graphs. Also the difficulty in proving NP-completeness varies: for some problems, NP-completeness is clear; but for others, the proof is quite difficult. In spite of these differences, it became apparent that these complete sets have some similarity: they are complicated enough to encode each other; they have simple encoding functions called "(weak) padding functions". For many complete sets, we can easily see such encoding functions once we have obtained the NP-completeness proof.

A similarity between NP-complete sets suggests that NP-computation is more powerful than P-computation. Let A be an arbitrary $\leq_m^p$-complete set in NP. Then SAT $\leq_m^p A$ by some polynomial time reduction f. Suppose that there is not so much difference in computational power between P- and NP-computation. Then A can be structurally simple: the reduction f compensates for the lack of information in A so that we still have $x \in \text{SAT} \leftrightarrow f(x) \in A$. Indeed, in such an extreme case as P = NP, A can be any set but the empty set or Σ^*. On the other hand, if NP-computation has a great advantage over P-computation, A must have a similar structure as SAT since no polynomial time reduction have enough computational power to accomplish drastic changes in the structure when mapping between SAT and A.

The same arguments also hold for any other complexity class $\mathcal{C}$, $\mathcal{C} \supseteq$ P, and polynomial time reducibility $\leq_r^p$. That is, for some kind of structural property, we can expect that

(*) the similarity of this property over all $\leq_r^p$-complete sets in $\mathcal{C}$ characterizes the superiority of $\mathcal{C}$-computation to P-computation.

For example, consider the cardinality of $\leq_{btt}^p$-complete sets. Then it is easy to show that P $\subsetneq \mathcal{C}$ if and only if all $\leq_{btt}^p$-complete sets in $\mathcal{C}$ are infinite. (*Note:* more detailed arguments about cardinality will be made in the next chapter.)

We consider "complete" sets since they are regarded as the most "intractable" sets in $\mathcal{C}$. In particular, $\leq_m^p$-reducibility offers the strongest sense of intractability; thus, it will be considered. Note that (*) is not true for any type of properties: for example, such a similarity as "using the same alphabet" obviously has nothing to do with the difference between P and $\mathcal{C}$. It is interesting to consider what types of similarity are available to exhibit the difference between P- and $\mathcal{C}$-computations.

Berman and Hartmanis [BH77] introduced "p-isomorphism" in order to investigate a structural similarity between $\leq_m^p$-complete sets in NP. They also developed tools to show that indeed all of the well-known $\leq_m^p$-complete sets (e.g., NP-complete sets in [GJ79]) are p-isomorphic. Hence, they conjectured that all $\leq_m^p$-complete sets in NP are p-isomorphic (the Berman-Hartmanis conjecture [BH77]). However we are not sure about the conjecture from the following two points: (i) we do not know whether NP-computation has an advantage over P-computation (thus P $\neq$ NP); and (ii) we do not know whether p-isomorphism is appropriate to characterize the superiority of NP-computation to P-computation. We will study the second point by investigating the structure of EXP.

In the Berman-Hartmanis conjecture, we are expecting that (a) NP-computation is much stronger than P-computation; and (b) the assertion (i) implies the p-isomorphism of all $\leq_m^p$-complete sets for NP. Then it is reasonable to expect the p-isomorphism of all $\leq_m^p$-complete sets for any other complexity class that is more intractable than NP. That is, the Berman-Hartmanis conjecture leads to the following conjecture: for any complexity class $\mathcal{C}$, if $\mathcal{C}$ is sufficiently intractable, then all $\leq_m^p$-complete sets in $\mathcal{C}$ are p-isomorphic (the extended Berman-Hartmanis conjecture). For example, we can expect that all $\leq_m^p$-complete sets in EXP are p-isomorphic since it is provable that P $\neq$ EXP and EXP contains several NP-complete sets (e.g., SAT). Hence, we can estimate the validity of the (extended) Berman-Hartmanis conjecture by studying the structure of EXP.

Several notions are necessary in order to discuss "similarity". In Chapter 3, we considered polynomial time reducibilities which are weaker than $\leq_m^p$-reducibility: $\leq_m^p$-reducibility offers the finest relation among them. In this chapter, since we are interested in the structure of the class of $\leq_m^p$-complete sets, we need yet finer relation than $\leq_m^p$-reducibility. Recall that all $\leq_m^p$-complete sets in $\mathcal{C}$ are $\equiv_m^p$-equivalent, that is, they are all reducible to each other by $\leq_m^p$-reductions. Hence, we state "similarity" in terms of the type of such $\leq_m^p$-reductions. So, notions concerning the type of $\leq_m^p$-reductions (i.e., functions) are needed here. We define the following classes of functions:

FP = $\{f : f$ is a polynomial time computable partial function$\}$;
LIFP = $\{f \in$ FP $: f$ is length increasing function$\}$;
1-LIFP = $\{f \in$ FP $: f$ is one-to-one and length increasing function$\}$; and
INVFP = $\{f \in$ 1-LIFP $: f^{-1}$ is in FP (i.e., f is polynomial time invertible)$\}$.

Define finer relations than $\equiv_m^p$ as follows.

Definition 4.1. For any sets A and B, A is $\leq_1^p$-, $\leq_{li}^p$-, $\leq_{1,li}^p$- and $\leq_{inv}^p$-*reducible* to B if it is $\leq_m^p$-reducible to B by a reduction f where f is one-to-one, f is in LIFP, f is in 1-LIFP, and f is in INVFP respectively. Also A and B are $\equiv_m^p$-, $\equiv_1^p$-, $\equiv_{li}^p$-, $\equiv_{1,li}^p$- and $\equiv_{inv}^p$-*equivalent* if they are $\leq_m^p$-, $\leq_1^p$-, $\leq_{li}^p$-, $\leq_{1,li}^p$- and

$\leq^{\mathrm{p}}_{\mathrm{inv}}$-reducible to each other.

In section 4.1, we will investigate how far we can prove the similarity of all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in EXP and in other complexity classes. From the results and observations in this section, we will finally propose the following conjecture on the structural similarity of $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets.

Conjecture. For all complexity class $\mathcal{C}$ that is sufficiently difficult, we have

(1) all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in $\mathcal{C}$ are $\equiv^{\mathrm{p}}_{1,\mathrm{li}}$-equivalent; but

(2) not all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in $\mathcal{C}$ are p-isomorphic.

(*Note:* That is, the extended Berman-Hartmanis conjecture is false; and instead of similarity by "p-isomorphism", we expect a little weaker similarity, i.e., $\equiv^{\mathrm{p}}_{1,\mathrm{li}}$-equivalence relation.) □

Since we propose the non-p-isomorphism conjecture in section 4.1, we will investigate further the structure of the class of $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in EXP under the non-p-isomorphism conjecture in section 4.2.

4.1. One-to-One Length-Increasing Equivalence Relation

In the following we will investigate the similarity of all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in EXP. To begin we review the approach taken by Berman and Hartmanis [BH77] for showing that *all known (at that time)* $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in NP are p-isomorphic. In order to show p-isomorphism between given two sets, Berman and Hartmanis introduced one important notion and proved two key lemmas. First lemma is the following.

Lemma 4.1. [BH77] Let A and B be any sets such that $A \equiv^{\mathrm{p}}_{\mathrm{inv}} B$. Then $A \equiv^{\mathrm{p}}_{\mathrm{iso}} B$. □

Hence, in order to show $A \equiv^{\mathrm{p}}_{\mathrm{iso}} B$ for given two sets A and B, it suffices to show that $A \equiv^{\mathrm{p}}_{\mathrm{inv}} B$.

Next they pointed out some structural property which can be seen in many $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in NP. This property is called *polynomial time paddability*.

Definition 4.2. [BH77] A set A is *p-paddable* (resp. *weakly p-paddable*) if $A \times \Sigma^* \leq^{\mathrm{p}}_{\mathrm{inv}} A$ (resp., $A \times \Sigma^* \leq^{\mathrm{p}}_{1,\mathrm{li}} A$).

Remark. In [BH77] only the notion of "paddability" was introduced. The notion of "weak paddability" has been considered in several contexts later (e.g., [JY85, OS86, Wa85]).

The (weak) p-paddability is a structural property which plays an important role in showing $\equiv^{\mathrm{p}}_{\mathrm{inv}}$- (resp., $\equiv^{\mathrm{p}}_{1,\mathrm{li}}$-) equivalence relation. The next lemma states it.

Lemma 4.2. [BH77] Let $\mathcal{C}$ be any complexity class which includes P and contains a standard $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set U. Then we have,

(1) U is p-paddable and $A \leq^{\mathrm{p}}_{\mathrm{inv}} U$ for all set A in $\mathcal{C}$; and

(2) for any $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set A in $\mathcal{C}$, A is (weakly) p-paddable iff $U \leq^{\mathrm{p}}_{\mathrm{inv}} A$ (resp., $U \leq^{\mathrm{p}}_{1,\mathrm{li}} A$).

Hence, if all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in $\mathcal{C}$ are (weakly) p-paddable, then they are all $\equiv^{\mathrm{p}}_{\mathrm{inv}}$-equivalent (resp., $\equiv^{\mathrm{p}}_{1,\mathrm{li}}$-equivalent).

Remark. We can extend the lemma in [BH77] in order to state the fact about weak paddability and to consider other complexity classes than NP. However this extension is straight forward; so its proof is omitted here. □

By showing p-paddability of all known (at that time) $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in NP, Berman and Hartmanis proved that they are all p-isomorphic. This approach is also important here. Indeed using Lemma 4.2 we prove the following similarity between *all* $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in EXP.

Theorem 4.3. All $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete sets in EXP are $\equiv_{1,\mathrm{li}}^{\mathrm{p}}$-equivalent.

Proof. From the above discussions it suffices to prove the following lemma. (*Note:* The same result was independently shown by Berman in his Ph.D Thesis [Be77]. We will state a simpler proof here.) □

Lemma 4.4. Let A be any $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete set in EXP. Then A is weakly p-paddable.

Proof. Berman and Hartmanis [BH77] constructed a set in EXP whose reduction to any other set must be one-to-one almost everywhere. Here we extend this idea to obtain a one-to-one and length-increasing $\leq_{\mathrm{m}}^{\mathrm{p}}$-reduction from $A \times \Sigma^*$ to A. First we define a set $L_A \subseteq \mathbf{N} \times \Sigma^* \times \Sigma^*$ by giving a description of a machine M which accepts L_A. Let $\{N_i\}_{i>0}$ be an enumeration of polynomial time bounded transducers, and let p_i denote a polynomial time bound for N_i. On input z, $z = \langle i, x, y\rangle$, M operates as follows (let n denote $|z|$ in the following):

```
 begin
   if $|i| \cdot p_i(n)^2 > 2^n$ then reject;
(1) $u \leftarrow N_i(\langle i, x, y\rangle)$;
(2) if $|u| \leq n$ then accept iff $u \notin A$;
(3) if $(\exists \langle x', y'\rangle < \langle x, y\rangle)[\, u = N_i(\langle i, x', y'\rangle)\, ]$ then
        $\langle x', y'\rangle \leftarrow$ the smallest one among such pairs;
        $n' \leftarrow |\langle i, x', y'\rangle|$
        accept iff $|i| \cdot p_i(n')^2 > 2^{n'} \vee x' \notin A$
        (i.e., $M$ rejects $\langle i, x', y'\rangle$)
    end-if
(4) otherwise accept iff $x \in A$
 end.
```

Recall that the universal transducer simulates $N_i(z)$ within $|i| \cdot p_i(n)^2$ steps; thus statement (1) can be done within $2^{O(n)}$ steps. Also note that the number of different $\langle x', y' \rangle$ which is less than $\langle x, y \rangle$ (by lexicographic order) is bounded by $2^{O(n)}$; so, statement (3) can be done within $2^{O(n)}$ steps. Since A is in EXP, some machine M_A accepts A within $2^{O(n)}$ steps; thus using M_A, statements (2) and (4) can be executed within $2^{O(n)}$ steps. Hence M on z halts within $2^{O(n)}$ steps, which proves that L_A is in EXP.

Since $L_A \in$ EXP and A is $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete in EXP, there exists a $\leq^{\mathrm{p}}_{\mathrm{m}}$-reduction from L_A to A. Let N_{i_0} achieve this reduction. We show that N_{i_0} is one-to-one and length-increasing almost everywhere on $\{i_0\} \times \Sigma^* \times \Sigma^*$.

Consider any input $z_0 = \langle i_0, x, y \rangle$ which is sufficiently long such that $|i| \cdot p_{i_0}(n)^2 \leq 2^n$, where $n = |z_0|$. The length-increasing property comes from statement (2). That is, $N_{i_0}(z_0)$ is longer than z_0. Suppose otherwise; then if-condition at statement (2) holds; thus $z_0 \in L_A$ if and only if $N_{i_0}(z_0)\ (= u) \notin A$, which contradicts that N_{i_0} is a $\leq^{\mathrm{p}}_{\mathrm{m}}$-reduction from L_A to A. Next the one-to-one property comes from statement (4). That is, for all $z'_0 = \langle i_0, x', y' \rangle$ such that $\langle x' ,y' \rangle < \langle x, y \rangle$, $N_{i_0}(z'_0) \neq N_{i_0}(z_0)$. Suppose otherwise, and let $\langle x', y' \rangle$ be the smallest pair of strings such that $N_{i_0}(z'_0) = u\ (= N_{i_0}(z_0))$. Then we have

$$
\begin{aligned}
z \in L_A &\longleftrightarrow M \text{ rejects } z'_0 \\
&\longleftrightarrow z'_0 \notin L_A \\
&\longleftrightarrow N_{i_0}(z'_0) \notin A && \text{(since } N_{i_0} \text{ is a reduction from } L_A \text{ to } A\text{)} \\
&\longleftrightarrow N_{i_0}(z_0) \notin A && \text{(since } N_{i_0}(z_0) = N_{i_0}(z'_0)\text{)} \\
&\longleftrightarrow z_0 \notin L_A && \text{(since } N_{i_0} \text{ is a reduction from } L_A \text{ to } A\text{)}
\end{aligned}
$$

This is a contradiction. Therefore if $\langle x, y \rangle$ is sufficiently long, then function N_{i_0} is one-to-one and length-increasing on $\langle i_0, x, y \rangle$.

Finally define f by

for every x and y in Σ^*, $f(\langle x, y \rangle) = N_{i_0}(\langle i_0, x, y0^m \rangle)$, where m is a sufficiently large constant.

Then the above argument proves that f is a one-to-one and length-increasing function in FP. Also it is clear that the execution of M always proceeds to statement (4) on input $\langle i_0, x, y0^m \rangle$ for all x and y; so for every x and y we have

$$x \in A \leftrightarrow \langle i_0, x, y0^m \rangle \in L \leftrightarrow N_{i_0}(\langle i_0, x, y0^m \rangle) \in A \leftrightarrow f(\langle x, y \rangle) \in A.$$

That is, f is a $\equiv^{\mathrm{p}}_{1,\mathrm{li}}$-reduction from $A \times \Sigma^*$ to A. □

Here consider how far we have proved the similarity of all $\le^{\mathrm{p}}_{\mathrm{m}}$-complete sets in EXP by Theorem 4.3. What we proved in Theorem 4.3 is $\equiv^{\mathrm{p}}_{1,\mathrm{li}}$-equivalence relation of all $\le^{\mathrm{p}}_{\mathrm{m}}$-complete sets in EXP. In order to prove the (extended) Berman-Hartmanis conjecture on EXP, what else we need to show? Note that p-isomorphism is the same as $\equiv^{\mathrm{p}}_{\mathrm{inv}}$-equivalence relation (Lemma 4.1); thus the difference between $\equiv^{\mathrm{p}}_{1,\mathrm{li}}$ and $\equiv^{\mathrm{p}}_{\mathrm{iso}}$ is the same as the one between $\equiv^{\mathrm{p}}_{1,\mathrm{li}}$ and $\equiv^{\mathrm{p}}_{\mathrm{inv}}$. So compare $\equiv^{\mathrm{p}}_{1,\mathrm{li}}$ and $\equiv^{\mathrm{p}}_{\mathrm{inv}}$ (or $\le^{\mathrm{p}}_{1,\mathrm{li}}$ and $\le^{\mathrm{p}}_{\mathrm{inv}}$); we will see that the difference between these two notions is only the p-invertibility of reductions.

Let us investigate this difference more precisely. Let A and B be any sets. Then $A \le^{\mathrm{p}}_{1,\mathrm{li}} B$ if there exists a $\le^{\mathrm{p}}_{1,\mathrm{li}}$-reduction f from A to B: namely, f is a one-to-one, length-increasing and p-computable reduction. On the other hand, $A \le^{\mathrm{p}}_{\mathrm{inv}} B$ if $A \le^{\mathrm{p}}_{1,\mathrm{li}} B$ and some $\le^{\mathrm{p}}_{1,\mathrm{li}}$-reduction from A to B is *p-invertible*. Hence the p-invertibility of one-to-one and length-increasing reductions in FP is important here; in general, this is the problem of whether all one-to-one and length-increasing functions in FP are p-invertible.

Recently several authors have discussed the problem of p-invertibility of functions in FP (see [Al85]). Among them the p-invertibility of one-to-one and length-increasing functions in FP has been often studied [GS84, Ko85, Va76]. For a given one-to-one and length-increasing function f in FP, the function f^{-1} is definable and moreover exponential time computable. However it is not known whether f^{-1} is p-computable in general. If f^{-1} is not p-computable, we call it a *one-way function*.

Definition 4.3. A *one-way function* is a one-to-one, length-increasing and polynomial time computable function whose inverse cannot be computed within polynomial time. That is, a one-way function is a function in INVFP − 1-LIFP.

Remark. In some contexts one-way functions are not necessarily one-to-one and length-increasing (e.g., [Al85]); those "one-way" functions are different from the above. In other contexts (e.g., [JY85]) one-way functions are defined using the notion of "polynomially honest" instead of "length-increasing"; but they are essentially equivalent to the above in our discussions.

Note again the difference between $\equiv^{\mathrm{p}}_{1,\mathrm{li}}$ and $\equiv^{\mathrm{p}}_{\mathrm{inv}}$ is the p-invertibility of one-to-one and length-increasing reductions. Thus, the existence of one-way functions relates deeply to the question of how far we could prove the similarity of $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in EXP by Theorem 4.3. Indeed, we have the following immediate corollary of Theorem 4.3.

Corollary 4.5. If no one-way functions exist, all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in EXP are p-isomorphic. □

However the assumption of this corollary (i.e., non existence of one-way functions) seems too strong. Although we do not know whether there exist one-way functions, one-way functions have been conjectured in several contexts [Ko85, Va76, Wa86b]: especially many results in recent research in cryptography assume the existence of some type of one-way functions (e.g., [GS84, Ya82]). That is, the hypothesis of Corollary 4.5 contradicts the following conjecture which is widely believed.

Conjecture 4.1. (One-Way Function Conjecture)

There exist one-way functions. □

Assume the existence of one-way functions. Then we have a candidate for non-p-isomorphic pairs. Consider any one-way function f. It is clear that $f(\mathrm{SAT})$ and $f(U)$ are respectively $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete in NP and in EXP; furthermore we have that $\mathrm{SAT} \equiv^{\mathrm{p}}_{1,\mathrm{li}} f(\mathrm{SAT})$ and $U \equiv^{\mathrm{p}}_{1,\mathrm{li}} f(U)$. However we have been unable to prove that $\mathrm{SAT} \leq^{\mathrm{p}}_{\mathrm{inv}} f(\mathrm{SAT})$ nor that $U \leq^{\mathrm{p}}_{\mathrm{inv}} f(U)$; and it seems difficult to prove these relationships. That is, assuming the existence of a one-way function f, we conjecture that SAT and $f(\mathrm{SAT})$ (resp., U and $f(U)$) are not p-isomorphic. Hence we propose the following conjecture in opposition to the Berman-Hartmanis conjecture.

Conjecture 4.2. (Non-P-Isomorphism Conjecture [JY85, Wa85])

(NP) Not all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in NP are p-isomorphic; and
(i.e., the Berman-Hartmanis conjecture is false)

(EXP) Not all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in EXP are p-isomorphic.
(i.e., the extended Berman-Hartmanis conjecture is false on EXP)

Remark. Note that this conjecture consists of two conjectures: (i) there exist one-way functions; and (ii) if one-way functions exist, then not all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets are p-isomorphic. □

By consecutive works [KLD85, KMR86] initiated by Long, we have obtained several results concerning this conjecture.

The class of $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in EXP is that of the most "intractable" sets in EXP in the sense of $\leq^{\mathrm{p}}_{\mathrm{m}}$-reducibility: it is the highest $\equiv^{\mathrm{p}}_{\mathrm{m}}$-degree in EXP. That is, we are considering the structure of the highest $\equiv^{\mathrm{p}}_{\mathrm{m}}$-degree. Although we have not obtained the answer to the above conjecture on the highest $\equiv^{\mathrm{p}}_{\mathrm{m}}$-degree in EXP, we have the answer on other $\equiv^{\mathrm{p}}_{\mathrm{m}}$-degrees which are relatively high in EXP.

There are weaker types of reducibility each of which also defines complete sets (see Chapter 3). Here we consider $\leq^p_{2\text{-tt}}$-reducibility. Obviously the highest $\equiv^p_{2\text{-tt}}$-degree contains the highest $\equiv^p_m$-degree. Moreover, we saw in Chapter 3 that the highest $\equiv^p_{2\text{-tt}}$-degree consists of more than one $\equiv^p_m$-degrees (Corollary 3.6).

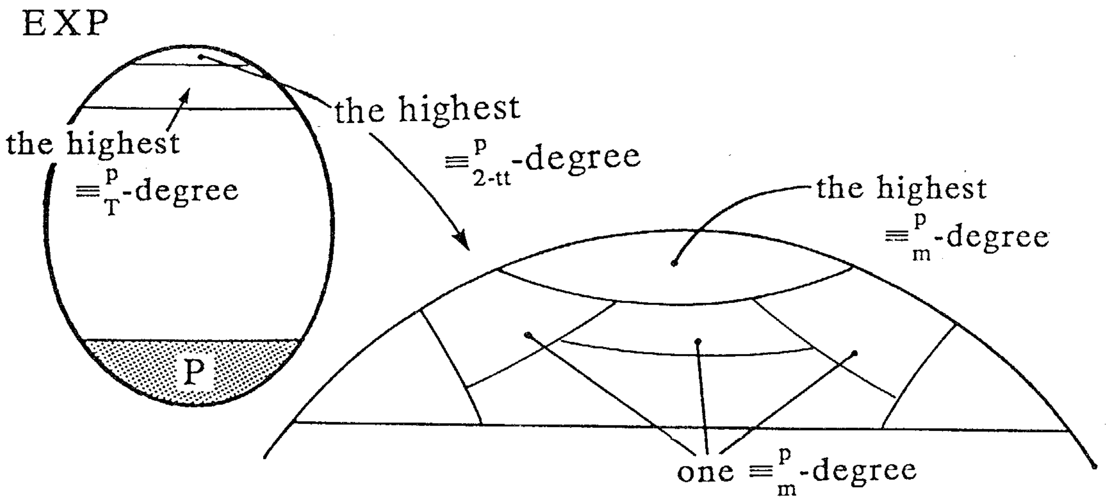


Figure 4.1. The structure of the highest $\equiv^p_{2\text{-tt}}$-degree in EXP.

Conjecture 4.2 concerns the problem of how similar sets of the highest $\equiv^p_m$-degree are. Consider the same problem to $\equiv^p_m$-degrees in the highest $\equiv^p_{2\text{-tt}}$-degree: i.e., relatively high $\equiv^p_m$-degrees in EXP. Then the following theorem gives us three kinds of $\equiv^p_m$-degrees in the highest $\equiv^p_{2\text{-tt}}$-degree: (1) a $\equiv^p_m$-degree such that all sets in it are p-isomorphic (i.e., the extended Berman-Hartmanis conjecture is true); (2) a $\equiv^p_m$-degree such that all sets in it are $\equiv^p_{1,li}$-equivalent but not all sets are p-isomorphic (i.e., the non-p-isomorphic conjecture is true); and (3) a $\equiv^p_m$-degree such that not all sets in it are $\equiv^p_{1,li}$-equivalent (i.e., even Theorem 4.3 does not hold).

Theorem 4.6.

(1) [KMR86] There exists a $\equiv_{m}^{p}$-degree in the class of $\leq_{2\text{-tt}}^{p}$-complete sets that consists of a single $\equiv_{iso}^{p}$-type;

(2) [KLD86] Assume that one-way functions exist. Then there exists a $\equiv_{m}^{p}$-degree in the class of $\leq_{2\text{-tt}}^{p}$-complete sets that consists of a single $\equiv_{1,li}^{p}$-type but more than one $\equiv_{iso}^{p}$-type; and

(3) [Wa85] There exists a $\equiv_{m}^{p}$-degree in the class of $\leq_{2\text{-tt}}^{p}$-complete sets that consists of more than one $\equiv_{1,li}^{p}$-type.

Remark. At (2), we need to assume the one-way function conjecture: recall that for any A and B, $A \equiv_{1,li}^{p} B$ implies $A \equiv_{inv}^{p} B$ (thus, $A \equiv_{iso}^{p} B$) unless one-way functions exist. □

Other Complexity Classes

Theorem 4.3 also holds for any other deterministic complexity class that includes EXP: more precisely, we have the same result for all super-SUBEXP classes and all super-PSPACE classes that have $\leq_{m}^{p}$-complete sets. This suggests the following conjecture: if a complexity class $\mathcal{C}$ is a sufficiently intractable complexity class, then all $\leq_{m}^{p}$-complete sets in $\mathcal{C}$ are $\equiv_{1,li}^{p}$-equivalent. That is, the $\equiv_{1,li}^{p}$-equivalent similarity between all $\leq_{m}^{p}$-complete sets in $\mathcal{C}$ characterizes the superiority of $\mathcal{C}$-computation to P-computation. Note that we have not proved this conjecture since we have not obtained any $\equiv_{1,li}^{p}$-equivalence results for nondeterministic time complexity classes.

For an example of a typical nondeterministic complexity class, we will consider the class NEXP. Let A and B be arbitrary $\leq_{m}^{p}$-complete sets in NEXP; thus, they are $\equiv_{m}^{p}$-equivalent. Although the following development is not sufficient to prove that they are $\equiv_{1,li}^{p}$-equivalent, it does give some evidence for this.

To simplify our discussions, assume that both A and B are in $NTIME(2^n)$: let M_A and M_B be respectively a nondeterministic acceptor which accepts A and B within 2^n steps. Since $A \leq^{\mathrm{p}}_{\mathrm{m}} B$, there exists a $\leq^{\mathrm{p}}_{\mathrm{m}}$-reduction f from A to B. The reduction f may not be in 1-LIFP, and the following cases may occur: (i) $|f(x_0)| < |x_0|$ for some x_0; and (ii) $f(x_1) = f(x_2)$ for some $x_1 < x_2$. In these two cases, it is sometimes possible to reduce the computational difficulty of A to a large extent.

Consider the case (i). Also assume that M_A on x_0 needs 2^n steps. Then we can decide "$x_0 \in A$?" by executing M_B on $f(x_0)$ which is *exponentially faster* than M_A on x_0 providing that x_0 is sufficiently long. That is, the reduction f exponentially decreases the difficulty of deciding "$x_0 \in A$?". Next consider the case (ii). Then the problem of "$x_2 \in A$?" is indirectly reduced to that of "$x_1 \in A$?": we can determine whether $x_2 \in A$ by computing $x_1 = f^{-1} \circ f(x_2)$ and executing M_A on x_1. Here again f can largely decrease the difficulty of A if the problem "$x_1 \in A$?" is much easier than "$x_2 \in A$?".

On the other hand, from our intuition that P-computation is much weaker than NEXP-computation, it should hardly be the case that f largely reduces the difficulty of A. Thus, the above two cases may not occur so often. The following two theorems prove this idea.

(*Note:* The following two theorems have been proved by several authors and appeared (in different ways) in several contexts [Al83, HLY86, JY85].)

Theorem 4.7. Let A and B be any $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in NEXP. Then there exists a $\leq^{\mathrm{p}}_{\mathrm{m}}$-reduction f from A to B such that

(1) for infinitely many x in A, $|f(x)| > |x|$; and

(2) for infinitely many x in A^{c}, $|f(x)| > |x|$.

Remark. We also have the similar $\leq^{\mathrm{p}}_{\mathrm{m}}$-reduction from B to A: that is,

$A \equiv_{\mathrm{li}}^{\mathrm{p}} B$ for infinitely many x.

Proof. By way of showing contradiction, suppose the otherwise. That is, there exists some $\leq_{\mathrm{m}}^{\mathrm{p}}$-reduction f from A to B such that at least one of the above conditions does not hold. Here we assume that the condition (i) does not hold for f (we can prove in almost the same way for the case that the condition (ii) does not hold): i.e., $|f(x)| \leq |x|$ for all x, $|x| \geq n_0$, where n_0 is some constant. Then it is easy to show that the following procedure accepts A within polynomial time.

```
begin
  input($x$);
  $u \leftarrow f(x)$;
  while $|u| < |f(x)|$ do $u \leftarrow f(u)$;
  if $|u| > n_0$ then reject
                 else accept iff $u \in B$
end.
```

Hence we have $A \in \mathrm{P}$. A contradiction. □

Theorem 4.8. Let A and B be any $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete sets in NEXP. Then there exists a $\leq_{\mathrm{m}}^{\mathrm{p}}$-reduction f from A to B such that f is one-to-one on A^{c}: i.e., for all x in A^{c}, there is no x' such that $f(x) = f(x')$. Moreover, if $A \equiv_{\mathrm{li}}^{\mathrm{p}} B$ then f is made so that it also satisfies the length-increasing property.

Remark. We also have a similar $\leq_{\mathrm{m}}^{\mathrm{p}}$-reduction from B to A: that is, $A \equiv_{1}^{\mathrm{p}} B$ on A^{c} and B^{c}.

Proof. The proof is similar to the one for Theorem 4.3 and Lemma 4.4. It suffices to show that for any $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete set A in NEXP, there exists a $\leq_{\mathrm{m}}^{\mathrm{p}}$-reduction f from $A \times \Sigma^*$ to A such that f is one-to-one on $A^{\mathrm{c}} \times \Sigma^*$.

Define the set $L_A \subseteq \mathrm{N} \times \Sigma^* \times \Sigma^*$ in the following way (let $\{N_i\}_{i>0}$ be an enumeration of polynomial time bounded deterministic transducers and let p_i

be a polynomial time bound for N_i):

$$\begin{aligned} &\langle i, x, y\rangle \in L_A \\ &\quad <=> \ |i| \cdot p_i(n)^2 \leq 2^n \ \textit{and} \\ &\qquad ((\exists \langle x', y'\rangle < \langle x, y\rangle)[N_i(\langle i, x', y'\rangle) = N_i(\langle i, x, y\rangle)] \vee x \in A), \\ &\qquad \text{where } n = |\langle i, x, y\rangle|. \end{aligned}$$

Following a similar argument in the proof of Lemma 4.4, we can easily prove that $L_A \in$ EXP $\subseteq$ NEXP. Thus, there exists some reduction N_{i_0} from L_A to A since A is $\leq^{\mathrm{p}}_{\mathrm{m}}$-hard for NEXP. Then it is also easy to show that N_{i_0} is one-to-one almost everywhere on $\{i_0\} \times A^c \times \Sigma^*$: namely that, for almost all $x, x' \in A^c$ and $y, y' \in \Sigma^*$, $N_{i_0}(\langle i_0, x', y'\rangle) \neq N_{i_0}(\langle i_0, x, y\rangle)$. The rest of the proof is similar to Lemma 4.4 and is left to the reader. □

Although the above theorems are not sufficient to show $A \equiv^{\mathrm{p}}_{1,\mathrm{li}} B$, they are good evidence for it. These arguments lead us to the following conjecture.

Conjecture 4.3. (Conjecture on the Similarity of $\leq^{\mathrm{p}}_{\mathrm{m}}$-Complete Sets)

For any complexity class $\mathcal{C}$, if $\mathcal{C}$ is *sufficiently difficult* then all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in $\mathcal{C}$ are $\equiv^{\mathrm{p}}_{1,\mathrm{li}}$-equivalent. □

If this conjecture is true, then the next important questions are (i) how difficult should $\mathcal{C}$ be when we say "sufficiently difficult"; and (ii) whether the class NP posses this difficulty. On the other hand, there seems no reason that all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in NEXP are $\equiv^{\mathrm{p}}_{\mathrm{inv}}$-equivalent, i.e., p-isomorphic: the invertibility of $\leq^{\mathrm{p}}_{\mathrm{m}}$-reduction seems nothing to do with the superiority of NEXP-computation to P-computation. Moreover, according to the arguments on EXP, we also have a candidate for non-p-isomorphic pairs of $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in NEXP: for any one-way function f and any $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set C, we have that $f(C)$ is $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete in NEXP and that $C \equiv^{\mathrm{p}}_{1,\mathrm{li}} f(C)$, but we have not been able to prove $C \equiv^{\mathrm{p}}_{\mathrm{inv}} f(C)$. This leads us the extended non-p-isomorphic conjecture (the extension of Conjecture 4.2 for any complexity classes). That is, we conjecture that for any complexity class $\mathcal{C}$, not all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in $\mathcal{C}$ are p-isomorphic.

It is worth mentioning that Theorem 4.7 and 4.8 also give some structural properties of $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in nondeterministic complexity classes though they are not main issue of this chapter. That is, we have the following corollary from each theorem.

Corollary 4.9. [HLY86] Let C be any $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set in NEXP. Then neither C nor C^{c} is NP-immune. □

Corollary 4.10. For any $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set C in NEXP, there exists a one-to-one function f in FP such that $f(\Sigma^*) \subseteq C^{\mathrm{c}}$. □

Especially, Corollary 4.10 was used in the previous chapter to show the difference between $\leq^{\mathrm{p}}_{\mathrm{m}}$-completeness and $\leq^{p}_{\mathrm{btt}}$-completeness in NEXP (Theorem 3.10 of section 3.1).

4.2 The Structure of the Class of $\leq^{\mathrm{p}}_{\mathrm{m}}$-Complete Sets

In the previous section, we conjectured that not all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in EXP are p-isomorphic (the non-p-isomorphic conjecture; Conjecture 4.2). That is, the highest $\equiv^{\mathrm{p}}_{\mathrm{m}}$-degree of EXP does not consist of a single $\equiv^{\mathrm{p}}_{\mathrm{iso}}$-type. Here, assuming the non-p-isomorphism conjecture, we investigate further the structure of the highest $\equiv^{\mathrm{p}}_{\mathrm{m}}$-degree.

Let U denote a standard $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set in EXP. If the non-p-isomorphic conjecture is true, there exists a $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set in EXP which is not p-isomorphic to U. As a first topic of this section, we investigate the type of $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets which is not p-isomorphic to U.

Recall that for any one-way function f, the pair U and $f(U)$ is a candidate for non-p-isomorphic pairs of $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets. One of the problems of interest is whether we have this type of non-p-isomorphic pair if the non-p-isomorphic conjecture is true at all. Let X be an arbitrary $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set in EXP which is not p-isomorphic to U. Recall that all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in EXP are $\equiv^{\mathrm{p}}_{1,\mathrm{li}}$-equivalent (Theorem 4.3); thus, $U \leq^{\mathrm{p}}_{1,\mathrm{li}} X$. Also recall that every $\leq^{\mathrm{p}}_{1,\mathrm{li}}$-reduction f from U to X must be a one-way function (Lemma 4.2 and the related discussions). Hence the pair U and $f(U)$ seems to be a non-p-isomorphic pair of $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets. Which suggests that if the non-p-isomorphic conjecture holds, then $U \not\equiv^{\mathrm{p}}_{\mathrm{iso}} f(U)$ for some one-way function f. We can confirm this intuition in higher deterministic complexity classes.

Theorem 4.11. Let U be a standard $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set in EXPSPACE (i.e., $DSPACE(2^{poly})$). Then if not all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in EXPSPACE are p-isomorphic, there exists a one-way function f such that $U \not\equiv^{\mathrm{p}}_{\mathrm{iso}} f(U)$.

Proof. If not all $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete sets in EXPSPACE are p-isomorphic, then there exists a set X such that $U \not\equiv^{\mathrm{p}}_{\mathrm{iso}} X$; so, $U \not\leq^{\mathrm{p}}_{\mathrm{inv}} X$ from Lemma 4.2. On the other hand, we have $U \equiv^{\mathrm{p}}_{1,\mathrm{li}} X$ from Theorem 4.3; thus, $U \leq^{\mathrm{p}}_{1,\mathrm{li}} X$ by some function f in 1-LIFP. Furthermore, f is not p-invertible: i.e., f is a one-way function. We will show that $U \not\equiv^{\mathrm{p}}_{\mathrm{iso}} f(U)$.

Suppose by the way of contradiction that $U \equiv^{\mathrm{p}}_{\mathrm{iso}} f(U)$; then $U \leq^{\mathrm{p}}_{\mathrm{inv}} f(U)$. Consider the following set $L_f \subseteq \mathbf{N} \times \Sigma^*$ (let $\{N_i\}_{i>0}$ be an enumeration of polynomial time bounded deterministic transducers and let p_i be a polynomial time bound for N_i):

$$\langle i, x\rangle \in L_f <=> |i|\cdot p_i(|\langle i, x\rangle|)^2 \leq 2^n \textit{ and } [\, N_i(\langle i, x\rangle) \notin f(\Sigma^*) \vee x \in U \,].$$

Then it is easy to show that $L_f \in$ EXPSPACE. Thus, it follows from Lemma 4.2 we have $L_f \leq^{\mathrm{p}}_{\mathrm{inv}} U$; so, $L_f \leq^{\mathrm{p}}_{\mathrm{inv}} f(U)$. Let $N_{i_0} \in$ INVFP be a reduction which witnesses $L_f \le^{\mathrm{p}}_{\mathrm{inv}} f(U)$.

Let n_0 be an integer such that $|i_0| \cdot p_{i_0}(n)^2 \le 2^n$ for all $n \ge n_0$. Consider any input $z_0 = \langle i_0, x\rangle$ such that $|z_0| \ge n_0$. Then we have $N_{i_0}(z_0) \in f(\Sigma^*)$. Suppose otherwise; then $z_0 \in L_f$ (from the definition of L_f) and $N_{i_0}(z_0) \notin f(U)$ (since $f(U) \subseteq f(\Sigma^*)$); which contradicts that N_{i_0} is a $\le^{\mathrm{p}}_{\mathrm{m}}$-reduction from L_f to $f(U)$. Thus, it follows from the definition of L_f that $\langle i_0, x\rangle \in L_f$ iff $x \in U$. Hence, letting $g_1 = \lambda x.N_{i_0}(\langle i_0, x\rangle)$, we have (i) $U \le^{\mathrm{p}}_{\mathrm{inv}} f(U)$ by g_1 and (ii) $g_1(x) \in f(\Sigma^*)$ for all x, $|x| \ge n_0$.

Recall that there exists a function *pad* which witnesses $U \times \Sigma^* \le^{\mathrm{p}}_{\mathrm{inv}} U$ since U is paddable. So, letting $g_2 = \lambda x.g_1 \circ pad(\langle x, 0^{n_0}\rangle)$, we have (i) $U \le^{\mathrm{p}}_{\mathrm{inv}} f(U)$ by g_2 and (ii) $g_2(x) \in f(\Sigma^*)$ for all x. Thus, for every $x \in \Sigma^*$, we have

$$\begin{aligned}
x \in U &\rightarrow g_2(x) \in f(U) \subseteq X; \text{ and} \\
x \notin U &\rightarrow g_2(x) \notin f(U) \textit{ and } g_2(x) \in f(\Sigma^*) \\
&\rightarrow g_2(x) \in f(\Sigma^* - U) \qquad (\text{since } f(\Sigma^*) - f(U) = f(\Sigma^* - U)) \\
&\rightarrow g_2(x) \in X^{\mathrm{c}} \qquad (\text{since } f(\Sigma^* - U) \subseteq X^{\mathrm{c}}).
\end{aligned}$$

Therefore, we have that $x \in U$ iff $g_2(x) \in X$: that is, g_2 is a $\le^{\mathrm{p}}_{\mathrm{inv}}$-reduction from U to X. A contradiction. □

Concerning the class EXP, we have only the following partial result.

Theorem 4.12. Consider a standard $\le^{\mathrm{p}}_{\mathrm{m}}$-complete set U in EXP. Let X be any $\le^{\mathrm{p}}_{\mathrm{m}}$-complete set in EXP such that

(1) $X \not\equiv^{\mathrm{p}}_{\mathrm{iso}} U$; and

(2) $U \le^{\mathrm{p}}_{1,\mathrm{li}} X$ by some function f such that $f(\Sigma^*) \in \mathrm{P}$.

Then we have $U \not\equiv^{\mathrm{p}}_{\mathrm{iso}} f(U)$. That is, f produces from U a $\le^{\mathrm{p}}_{\mathrm{m}}$-complete set which is not p-isomorphic to U.

Remark. Note that every $\le^{\mathrm{p}}_{\mathrm{m}}$-complete set in EXP is $\equiv^{\mathrm{p}}_{1,\mathrm{li}}$-equivalent to U;

thus we have $U \leq^{\mathrm{p}}_{1,\mathrm{li}} X$ by some reduction f. The condition (2) requires that $f(\Sigma^*) \in \mathrm{P}$ for such reductions.

Proof. The proof is similar to the one for Theorem 4.11; so, it is omitted. □

Next, consider the problem of how the highest $\leq^{\mathrm{p}}_{\mathrm{m}}$-degree of EXP is divided by $\equiv^{\mathrm{p}}_{\mathrm{iso}}$-types, i.e., p-isomorphism classes. We conjecture that the highest $\equiv^{\mathrm{p}}_{\mathrm{m}}$-degree of EXP does not consist of a single $\equiv^{\mathrm{p}}_{\mathrm{iso}}$-type (the non-p-isomorphic conjecture; Conjecture 4.2). Then consider the question of how many $\equiv^{\mathrm{p}}_{\mathrm{iso}}$-types in such $\leq^{\mathrm{p}}_{\mathrm{m}}$-degree. Mahaney [Ma81] answered this question.

Theorem 4.13. [Ma81] If the highest $\equiv^{\mathrm{p}}_{\mathrm{m}}$-degree in NP does not consist of a single $\equiv^{\mathrm{p}}_{\mathrm{iso}}$-type, it contains infinitely many $\equiv^{\mathrm{p}}_{\mathrm{iso}}$-types.

Remark. This theorem also holds for any other complexity class which contains P. □

Mahaney [Ma81] also considered the order relations of these $\equiv^{\mathrm{p}}_{\mathrm{iso}}$-types where the order of two $\equiv^{\mathrm{p}}_{\mathrm{iso}}$-types is defined by $\leq^{\mathrm{p}}_{\mathrm{inv}}$-reducibility: for any two $\equiv^{\mathrm{p}}_{\mathrm{iso}}$-types $\mathcal{D}_1$ and $\mathcal{D}_2$, we say that $\mathcal{D}_1 \lesssim \mathcal{D}_2$ if any set in $\mathcal{D}_1$ is $\leq^{\mathrm{p}}_{\mathrm{inv}}$-reducible to some set in $\mathcal{D}_2$. Recently, Mahaney and Young [MY85] considerably extended the Mahaney's work in this direction.

Theorem 4.14. [MY85] Any $\equiv^{\mathrm{p}}_{\mathrm{m}}$-degree which does not consist of a single $\equiv^{\mathrm{p}}_{\mathrm{iso}}$-type contains infinitely many $\equiv^{\mathrm{p}}_{\mathrm{iso}}$-types densely ordered under the $\lesssim$-ordering. Also it contains infinitely many $\equiv^{\mathrm{p}}_{\mathrm{iso}}$-types which are incomparable under the $\lesssim$-ordering. Thus, any $\equiv^{\mathrm{p}}_{\mathrm{m}}$-degree which fails to consist of a single $\equiv^{\mathrm{p}}_{\mathrm{iso}}$-type contains every possible countable linear ordering under the $\lesssim$-ordering.

Remark. Note that the above theorem extends Theorem 4.13 also in the sense that it states the structure of all $\equiv^{\mathrm{p}}_{\mathrm{m}}$-degrees besides the highest one. □

Here we consider another extension of Theorem 4.13. In the first topic, we investigate the problem of whether we have a non-p-isomorphic pair such as U and $f(U)$ if non-p-isomorphic pairs indeed exist (where U is a standard $\leq_m^p$-complete set). That is, we considered whether some one-way function f produce a $\leq_m^p$-complete set from U which is non-p-isomorphic to U. Next, let us consider a similar problem for Theorem 4.13. Namely, the following problem: assuming the non-p-isomorphic conjecture, is there any one-way function g that generates infinitely many $\leq_m^p$-complete sets from U which are not p-isomorphic to each other? We have the following partial answer to this problem in EXP.

Theorem 4.15. Let U be a standard $\leq_m^p$-complete set in EXP. If there exists a one-way function f such that $U \not\equiv_{iso}^p f(U)$, then there exists a one-way function g such that $g^i(U) \not\equiv_{iso}^p g^j(U)$ for all $i > j \geq 0$.

Remark. Note that $g^i(U)$ is $\leq_m^p$-complete in EXP. That is, g produces from U infinitely many $\equiv_{iso}^p$-types in the highest $\equiv_m^p$-degree of EXP. □

We prove this theorem following the idea that is used to prove Theorem 4.11: we use a delayed diagonalization technique [La75] to construct infinite pairwise non-p-isomorphic sets. Since the construction in this proof is rather complicated, we start by explaining a simple example.

To make our statement simple, we introduce several notations. Recall that INVFP is a class of one-to-one, length-increasing, p-computable and p-invertible functions, and that $A \leq_{inv}^p B$ means that A is $\leq_m^p$-reducible to B by a reduction in INVFP. Here we will consider a weaker relation: for any sets A and B, we say that $A \leq_{winv}^p B$ if A is $\leq_m^p$-reducible to B by a one-to-one, p-computable and p-invertible reduction (i.e., the length-increasing property is not necessary). Note that if A is p-paddable, then $A \leq_{inv}^p B$ if and only if A

$\leq^{\mathrm{p}}_{\mathrm{winv}}$ B. Also we use an abbreviation $\leq^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}}$: for any two sets A and B, $A \leq^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} B$ if there exists $n \geq 0$ such that $A^{\geq n} \leq^{\mathrm{p}}_{\mathrm{winv}} B^{\geq n}$.

Suppose that there exists two sets A and B such that $A \not\leq^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} B$, and consider the problem of constructing infinitely many sets A_1, A_2, $\cdots$ such that $A_h \not\leq^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} A_k$ for all $k > h > 0$. To accomplish this we consider the following stronger conditions and define sets which satisfy them:

(C1) for all $k \geq 0$, $A_k \not\leq^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} B$; and

(C2) for all $k > h \geq 0$, $A_h \not\leq^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} A_k$

(A_0 is used to denote the set A).

We define the desired sets inductively from A_1. Let $k > 0$ be arbitrarily fixed, and consider the construction of the set A_k. Here we assume, as an induction hypothesis, that we have already had the sets A_0, A_1, $\cdots$, A_{k-1}. Then the set A_k is constructed by a delayed diagonalization. What follows is an explanation of the outline of the construction. Let $\{N_i\}_{i>0}$ be a standard enumeration of polynomial time transducers (we assume that every polynomial time function appears infinitely many times in this enumeration). We define A_k to be A_{k-1} and to be B in turn so that it satisfies (1) $A_k \not\leq^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} B$; and (2) for all $k > h \geq 0$, $A_h \not\leq^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} A_k$. That is, at some stages of the construction where we want to make A_k falsify $A_k \leq^{\mathrm{p}}_{\mathrm{winv}} B$ by N_i, A_k is defined to be A_{k-1} until a witness for the falsity of $A_k \leq^{\mathrm{p}}_{\mathrm{winv}} B$ by N_i is found (such a witness will be eventually found since $A_{k-1} \not\leq^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} B$); and at some stages where A_k is expected to falsify $A_h \leq^{\mathrm{p}}_{\mathrm{winv}} A_k$ by N_i, it is defined to be B until a witness for this fact is found (such a witness will be eventually found since $A_h \not\leq^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} B$). The former and the latter witness is respectively called a witness of type (1) and type (2).

Let us state this construction more precisely. The above two types of witnesses are defined formally as follows:

(type 1: a witness for the falsity of $A_k \leq^{\mathrm{p}}_{\mathrm{winv}} B$ by N_i)

for every $\langle i, j \rangle$, $0 < i, j$, a string x is a *witness of type (1)* (w.r.t. $\langle i, j \rangle$) if one of the following conditions holds:

(i) $(\exists y < x)[\, N_i(y) = N_i(x) \,]$,

(ii) $N_j \circ N_i(x) \neq x$, or

(iii) $(x \in A_k \wedge N_i(x) \notin B) \vee (x \notin A_k \wedge N_i(x) \in B)$.

(type 2: a witness for the falsity of $A_h \leq^{\mathrm{p}}_{\mathrm{winv}} A_k$ by N_i)

for every $\langle i, j, h \rangle$, $0 < i, j$ and $0 \leq h < k$, a string x is a *witness of type (2)* (w.r.t. $\langle i, j, h \rangle$) if one of the following conditions holds:

(i) $(\exists y < x)[\, N_i(y) = N_i(x) \,]$,

(ii) $N_j \circ N_i(x) \neq x$, or

(iii) $(x \in A_h \wedge N_i(x) \notin A_k) \vee (x \notin A_h \wedge N_i(x) \in A_k)$.

Then A_k is defined by the following stage construction (recall that $L^{=n}$ denotes the set of elements of L of length n):

```
stage $n = 0$;
  $A_k \leftarrow \emptyset$;
  $\langle i, j \rangle \leftarrow \langle 1, 1 \rangle$; $status \leftarrow 1$;
stage $n > 0$;
  if $status = 1$ then   (i.e., a state waiting for a witness of type (1))
    $A_k \leftarrow A_k \cup A_{k-1}^{=n}$;
 (*) search for a witness of type (1) for all $x$, $|x| \leq loglog\ n$;
    if a witness of type (1) is found then
      $h \leftarrow 0$; $status \leftarrow 2$
    end-if
  else                   (i.e., a state waiting for a witness of type (2))
    $A_k \leftarrow A_k \cup B^{=n}$;
    (*) search for a witness of type (2) for all $x$, $|x| \le loglog\ n$;
        if a witness of type (2) is found then
           if $h \ge k - 1$ then
              $\langle i, j \rangle \leftarrow next(\langle i, j \rangle)$;   (i.e., $\langle i, j \rangle$ becomes the next pair)
              $status \leftarrow 1$
           else
              $h \leftarrow h + 1$; $status \leftarrow 2$
           end-if
        end-if
     end-if
  end-construction.
```

(*Note:* The function "*loglog n*" used at step (*) is not essential: it can be any function $l(n)$ which satisfies (i) $l(n)$ is nondecreasing and increases infinitely many times; and (ii) the time needed for step (*) is bounded by some polynomial in n.)

Consider any integers i, j and h such that $0 < i, j$ and $0 \le h < k$. Note that we assumed that $A_h \not\le^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} B$ for all h, $0 \le h < k$; thus, a witness of type (1) w.r.t. i and j (resp., type (2) w.r.t. i, j and h) is eventually found in this construction. Such a witness ensures the falsity of $A_k \le^{\mathrm{p}}_{\mathrm{winv}} B$ by N_i (resp., $A_h \le^{\mathrm{p}}_{\mathrm{winv}} A_k$ by N_i). We also assumed that every polynomial time function appears infinitely many times in the enumeration $\{N_i\}_{i>0}$. Hence, we have both $A_k \not\le^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} B$ and $A_h \not\le^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} A_k$: namely that, A_k satisfies the conditions (C1) and (C2).

Now consider the proof of Theorem 4.15. Here we assume that $U \not\equiv^{\mathrm{p}}_{\mathrm{iso}} f(U)$; thus, $U \not\le^{\mathrm{p}}_{\mathrm{inv}} f(U)$ since $f(U) \le^{\mathrm{p}}_{\mathrm{inv}} U$ (see Lemma 4.1). Hence, we have $U \not\le^{\mathrm{p}}_{\mathrm{winv}} f(U)$ since U is p-paddable. Besides this relation, we need, in the following proof, a rather technical assumption on f. This assumption is stated precisely as "$f(\Sigma^*)^c$ has an WINVFP-easy subset", where WINVFP denotes the class of one-to-one, p-computable and p-invertible functions (see Chapter

3.1 for the definition of "easy subset"); but we will use a simpler (but stronger) assumption. The following lemma shows that we can assume it on EXP.

Lemma 4.16. Let f be a one-way function such that $U \not\leq^{\mathrm{p}}_{\mathrm{winv}} f(U)$, where U denotes a standard $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set in EXP. Then there exists a one-way function f' such that $U \not\leq^{\mathrm{p}}_{\mathrm{winv}} f'(U)$ and $f'(\Sigma^*) \subseteq 0\Sigma^*$ (so, $f'(\Sigma^*)^{\mathrm{c}}$ contains $1\Sigma^*$; thus it has an WINVFP-easy subset).

Proof. We show that $f' = \lambda x.0f(x)$ satisfies the theorem. The outline of the proof is similar to the one for Theorem 4.11.

Suppose by the way of contradiction that $U \leq^{\mathrm{p}}_{\mathrm{winv}} f'(U)$. Define the set $L_0 \subseteq \mathbf{N} \times \Sigma^*$ by

$$\langle i, x\rangle \in L_0 \iff |i| \cdot p_i(|\langle i, x\rangle|)^2 \leq 2^n \textit{ and } [\, N_i(\langle i, x\rangle) \notin 0\Sigma^* \vee x \in U \,].$$

Since we assumed that $U \leq^{\mathrm{p}}_{\mathrm{winv}} f'(U)$, there exists a reduction N_{i_0} in WINVFP from U to $f'(U)$. Then following the same argument as Theorem 4.11, we obtain, from N_{i_0}, a reduction g' such that (i) $U \leq^{\mathrm{p}}_{\mathrm{winv}} 0f(U)$ by g' and (ii) $g'(\Sigma^*) \subseteq 0\Sigma^*$. So, define g by

for every $x \in \Sigma^*$, $g(x) = y$, where $g'(x) = 0y$.

Then we have $U \leq^{\mathrm{p}}_{\mathrm{winv}} f(U)$ by g. A contradiction. □

Hence, we can assume without loss of generality that $f(U) \subseteq 0\Sigma^*$.

Letting $A = U$ and $B = f(U)$, the above construction yields infinitely many sets which satisfy the conditions (C1) and (C2). However we need to consider further requirements so that each A_k can be generated by some one-to-one polynomial time function g (i.e., $A_k = g^k(U)$). That is, the following two additional conditions are necessary (we use a, b, c and d to denote a string 00, 01, 10 and 11 respectively):

(C3) for all $k \geq 0$, $A_k \subseteq a^k b\Sigma^* \cup a^k c\Sigma^*$

(i.e., each A_k must be constructed in the different domain); and

(C4) for all $k \geq 0$ and $x \in U$, A_k contains only and at least one of $a^k bx$ and $a^k cf(x)$.

Letting $A_0 = bU$ and $B = f(U)$, we define sets $\{A_k\}_{k>0}$ which satisfy the conditions (C1) – (C4). We construct them inductively from A_1.
(*Note:* We need to assume $bU \not\leq^{\mathrm{p}}_{\mathrm{winv\,(a.e.)}} f(U)$ as an induction base for (C1), which immediately follows from the fact that $U \leq^{\mathrm{p}}_{\mathrm{winv}} bU$ and the assumption that $U \not\leq^{\mathrm{p}}_{\mathrm{winv}} f(U)$.)
For arbitrarily fixed $k > 0$, consider the construction of A_k. The set A_k is defined as follows:

```
stage $n = 0$;
  $A_k \leftarrow \emptyset$;
  $\langle i, j\rangle \leftarrow \langle 1, 1\rangle$; $status \leftarrow 1$;
stage $n > 0$;
  if $status = 1$ then   (i.e., a state waiting for a witness of type (1))
    $A_k \leftarrow A_k \cup a\{a^{k-1}bx \in A_{k-1} : |f(x)| = n\}$
            $\cup\ a\{a^{k-1}cy \in A_{k-1} : |y| = n\}$;
    (i.e., $A_k$ becomes almost $aA_{k-1}$ if $status = 1$ forever)
 (*) search for a witness of type (1) for all $x$, $|x| \leq loglog\ n$;
    if a witness of type (1) is found then
      $h \leftarrow 0$; $status \leftarrow 2$
    end-if
  else                  (i.e., a state waiting for a witness of type (2))
    $A_k \leftarrow A_k \cup a^k cB^{=n}$
    (i.e., $A_k$ becomes almost $a^k cB$ if $status = 2$ forever)
 (*) search for a witness of type (2) for all $x$, $|x| \leq loglog\ n$;
    if a witness of type (2) is found then
      if $h \geq k - 1$ then
```

```
        $\langle i, j\rangle \leftarrow next(\langle i, j\rangle)$; (i.e., $\langle i, j\rangle$ becomes the next pair)
        $status \leftarrow 1$
      else
        $h \leftarrow h + 1$; $status \leftarrow 2$
      end-if
    end-if
  end-if
end-construction.
```

Then it follows from the construction that A_k satisfies the conditions (C3) and (C4). From the induction hypothesis and Claim 1 (see below), we have that $aA_{k-1} \not\leq^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} B$ and that $A_h \not\leq^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} a^k cB$ for all h, $0 \leq h < k$. Thus, for every i, j and h, $0 < i, j$ and $0 \leq h < k$, a witness of type (1) (resp., type (2)) will be eventually found during the construction; hence, A_k satisfies the conditions (C1) and (C2) (see the discussion for the previous example). Therefore, sets $\{A_k\}_{k>0}$ satisfy the conditions (C1) − (C4).

Claim 1. Suppose that $A \not\leq^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} B$. Then for all $u, v \in \Sigma^*$, we have $uA \not\leq^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} vB$.

Remark. The assumption discussed in Lemma 4.16 is needed only to prove this claim.

Proof. Suppose by the way of contradiction that $uA \leq^{\mathrm{p}}_{\mathrm{winv}} vB$ by g. Define g' by

$$g'(x) = \begin{cases} y & \text{if } g(ux) = vy \text{ for some } y \in 0\Sigma^*, \text{ and} \\ 1x & \text{otherwise.} \end{cases}$$

Then it is easy to show that g' is a one-to-one, p-computable and p-invertible function.

Let x be any string such that $g(ux) \notin v0\Sigma^*$. Recall that we assumed B (= $f(U)$) $\subseteq 0\Sigma^*$; so, $g(ux) \notin v0\Sigma^*$ (i.e., $g(ux) \notin vB$) implies that $x \notin A$ since

$g(ux) \in vB$ iff $ux \in uA$. Hence, we have both $x \notin A$ and $g'(x)$ $(= 1x) \notin B$. On the other hand, it is clear that $x \in A$ iff $g'(x) \in B$ for any string x such that $g(ux) \in v0\Sigma^*$. Therefore, g' is a reduction from A to B; namely, $A \leq^{\mathrm{p}}_{\mathrm{winv}} B$. A contradiction. □

The following claim shows that each A_k is generated by some one-to-one, length-increasing and polynomial time function.

Claim 2. There exists a one-to-one, length-increasing and polynomial time function g_1 such that for every $k > 0$, $A_k = g_1^k(A_0)$.

Proof. For every k and $n > 0$, define $up(k, n)$ to be false (resp., true) if *status* $= 1$ (resp., $= 2$) at stage n during the construction of A_k (we define $up(k, 0)$ and $up(0, n)$ to be true for all k and n). Then it is easy to show that up is polynomial time computable with respect to k and n. The desired function g_1 is defined as follows:

for every $k > 0$ and $x \in \Sigma^*$ (let n and m denote $|x|$ and $|f(n)|$),

$$g_1(x) = x \qquad \text{if } x \notin a^*\{b, c, d\}\Sigma^*;$$

$$g_1(a^{k-1}bx) = \begin{cases} a^k bx & \text{if } up(k-1, m) \vee \neg up(k, m), \\ a^k cf(x) & \text{if } \neg up(k-1, m) \wedge up(k, m); \end{cases}$$

$$g_1(a^{k-1}cx) = \begin{cases} a^k cx & \text{if } up(k-1, n) \vee \neg up(k, n), \\ a^k d0x & \text{if } \neg up(k-1, n) \wedge up(k, n); \text{ and} \end{cases}$$

$$g_1(a^{k-1}dx) = a^k d1x.$$

Then it is easy to show that g_1 is a one-to-one, length-increasing and polynomial time function. Also we have $A_k = g_1(A_{k-1})$ for every $k > 0$; thus, $A_k = g_1^k(A_0)$. Since these proofs are tedious but easy, we left them to the interested readers. □

As a consequence of the above discussions, we have the following lemma.

Lemma 4.17. Suppose that there exists a one-way function f such that $U \not\equiv^{\mathrm{p}}_{\mathrm{iso}} f(U)$, where U is a standard $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set in EXP. Then there exist strings a and b, and a one-to-one, length-increasing and polynomial time function g_1 such that

(1) $g_1^i(bU) \subseteq a\Sigma^*$ for all $i > 0$; and

(2) $g_1^j(bU) \not\equiv^{\mathrm{p}}_{\mathrm{winv\ (a.e.)}} g_1^i(bU)$ for all $i > j \geq 0$. □

Proof of Theorem 4.15.

Let g_1 be the function defined in the above. Note that $A \leq^{\mathrm{p}}_{\mathrm{winv}} B$ if $A \equiv^{\mathrm{p}}_{\mathrm{iso}} B$; thus, we have that $g_1^j(bU) \not\equiv^{\mathrm{p}}_{\mathrm{iso}} g_1^i(bU)$ for all $i > j > 0$. Also it follows from $U \equiv^{\mathrm{p}}_{\mathrm{iso}} bU$ that $U \not\equiv^{\mathrm{p}}_{\mathrm{iso}} g_1^i(bU)$ for all $i > 0$. Hence, it suffice to prove that there exists a one-to-one, length-increasing and polynomial time function g such that $g^k(U) = g_1^k(bU)$ for all $k > 0$. We can assume without loss of generality that $U \cap a\Sigma^* = \emptyset$. Furthermore we have $g_1^i(bU) \subseteq a\Sigma^*$ for all $i > 0$. Therefore, the following function satisfies our purpose:

for every $x \in \Sigma^*$,

$$g(x) = \begin{cases} g_1(x) & \text{if } x \in a\Sigma^*, \\ g_1(bx) & \text{otherwise.} \end{cases}$$

□

Recall that we have Theorem 4.11 for higher complexity classes. Thus, we can completely solve this problem for higher complexity classes such as EXPSPACE.

Theorem 4.18. If the non-p-isomorphic conjecture is true on EXPSPACE, then there exists a one-way function g such that $g^i(U) \not\equiv^{\mathrm{p}}_{\mathrm{iso}} g^j(U)$ for all $i > j \geq 0$, where U is a standard $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set in EXPSPACE. □

Other Complexity Classes

Theorem 4.12 and 4.15 (resp., Theorem 4.11 and 4.18) hold for all complexity classes that contain EXP (resp., EXPSPACE).

The proof of Theorem 4.12 is not applicable to complexity classes such as NP. However, a different proof gives us a similar result.

Theorem 4.19. [Wa86c] Consider any p-paddable $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set, say SAT. Let X be any $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set in NP such that

(1) $X \not\equiv^{\mathrm{p}}_{\mathrm{iso}}$ SAT; and

(2) SAT $\leq^{\mathrm{p}}_{1,\mathrm{li}} X$ for some function f such that $f(\Sigma^*) \in$ P.

Then we have

(3) both $f(\mathrm{SAT})$ and $f(\mathrm{SAT}^c)^c$ are $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete in NP; and

(4) either SAT $\not\equiv^{\mathrm{p}}_{\mathrm{iso}} f(\mathrm{SAT})$ or SAT $\not\equiv^{\mathrm{p}}_{\mathrm{iso}} f(\mathrm{SAT}^c)^c$. □

In order to prove Theorem 4.15 for the class NP, we need to assume the technical assumption that was proved on EXP in Lemma 4.16. So, the corresponding theorem for NP becomes as follows:

Theorem 4.20. Let U be a standard $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set in EXP. If there exists a one-way function f such that (i) $U \not\equiv^{\mathrm{p}}_{\mathrm{iso}} f(U)$, and (ii) $f(\Sigma^*)^c$ has a WINVFP-easy subset, then there exists a one-way function g such that $g^i(U) \not\equiv^{\mathrm{p}}_{\mathrm{iso}} g^j(U)$ for all $i > j \geq 0$. □

5. Polynomial Time Reducibility to a Set of Small Density

In order to analyze a complexity class $\mathcal{C}$, we have investigated properties of sets *in* $\mathcal{C}$. For example, we have observed several properties of complete sets in EXP in order to show how EXP is difficult than P: note that a complete set in $\mathcal{C}$ is a hard set which *is in* $\mathcal{C}$. Here we will investigate a property (i.e., density) of sets to which sets in EXP are polynomial time reducible; those sets are not necessarily in EXP. This approach yields another type of "tractability/intractability" notions. We show the intractability of EXP from this different point of view.

Let A be any set. Recall that $\mathrm{cens}_A(n)$ denotes the census function of A. We say that A is *sparse* if $\mathrm{cens}_A(n) = O(p(n))$ for some polynomial p. We say that A is *exponentially dense* if $\mathrm{cens}_A(n) = \omega(2^{n^{\epsilon}})$ for some constant ϵ, $0 < \epsilon$; and A is *bi-exponentially dense* if both A and A^{c} are exponentially dense.

Consider any set $A \notin$ P. Then A is intractable in the sense that the time complexity of any machine that accepts A is more than polynomial: here, time complexity is used for measuring tractability. On the other hand, density of sets to which A is polynomial time reducible is another measure of tractability. For example, suppose that A is $\le_r^{\mathrm{p}}$-reducible to some sparse set X, where $\le_r^{\mathrm{p}}$ is any polynomial time reducibility. Then we have the following polynomial time *table-lookup algorithm* for A [Ma86].

> Let M be an oracle machine which achieve $\le_r^{\mathrm{p}}$-reduction from A to X. Note that M solves A up to size n if the information of $X^{\le p(n)}$ is given, where p is some polynomial; and that a polynomial size table is enough to express this information since X is sparse. Although the computation of the table may need huge amount of time, the table itself is not so large. Hence, having the table after the precomputation, we can solve A within polynomial time for all instances up to size n: the computation of M and searching the table

can be accomplished within polynomial time.

So A has a feasible algorithm in a weak sense even though it is not in P. Note that the density of X determines the size of the table; hence it determines the feasibility of table-lookup algorithms. In this chapter we use the density of such sets as X for measuring tractability of A.

For any $\leq_r^{\mathrm{P}}$-reducibility and any complexity class $\mathcal{C}$, a $\leq_r^{\mathrm{P}}$-hard set for $\mathcal{C}$ is a set to which all sets in $\mathcal{C}$ are $\leq_r^{\mathrm{P}}$-reducible. Hence, analyzing density of $\leq_r^{\mathrm{P}}$-hard sets for $\mathcal{C}$, we can investigate the tractability of all sets in $\mathcal{C}$. In section 5.1, we will study density of $\leq_r^{\mathrm{P}}$-hard sets for EXP.

In section 5.2, we will discuss *polynomial time pseudo algorithms*: i.e., approximation algorithms and polynomial size circuits. Several authors have introduced several types of pseudo algorithms and considered the notion of "having polynomial time pseudo algorithms": e.g., APT [MP79], 1-APT [OS86], p-closeness to P [Ye83, Sc86b] and P/poly [KL80]. They considered these notions in order to define weaker notions of tractability than the polynomial time computability. Each of these pseudo algorithms is regarded as a special case of table-lookup algorithms considered above. We show that the existence of polynomial time pseudo algorithms relates deeply to polynomial time reducibilities to sparse sets.

5.1. Density of Hard Sets

We first consider the most general case: that is, consider hard sets with respect to $\leq_{\mathrm{T}}^{\mathrm{P}}$-reductions. We have the following partial result on the density of $\leq_{\mathrm{T}}^{\mathrm{P}}$-hard sets.

Theorem 5.1. Let $t(n)$ be any nondecreasing function which is exponential time computable. And let A be any $\leq_{\mathrm{T}}^{\mathrm{P}}$-hard set for EXP such that $A \in DTIME(2^{t(n)})$. Then $\mathrm{cens}_A(n) \geq (p \circ t)^*(n) + c$ for some polynomial p

and constant c, where for any function f, we define f^* to be the function that maps every integer $n > 0$ to the largest k such that $\underbrace{f \circ f \cdots f}_{k}(1) < n$.

Proof. We use essentially the same technique used in the proof of Theorem 3.2 where we showed the length-increasing property of $\leq_{\mathrm{T}}^{\mathrm{P}}$-reduction from the set L_A to A.

Consider an enumeration $\{M_i\}_{i>0}$ of polynomial time bounded deterministic oracle machines and let p_i be a polynomial time bound for M_i. Define $L_A \subseteq \mathbf{N} \times \Sigma^*$ as follows:

$$\langle i, x\rangle \in L_A \iff |i| \cdot p_i(n)^2 \leq 2^n \textit{ and } \langle i, x\rangle \notin L(M_i, A^{\leq m}),$$
$$\text{where } n = |\langle i, x\rangle| \text{ and } m = t^{-1}(n).$$

Then L_A is in EXP. Thus, there exists some $\leq_{\mathrm{T}}^{\mathrm{P}}$-reduction from L_A to A since A is EXP-hard set. Let M_{i_0} be an oracle machine which achieves this reduction. Following an argument similar to the proof of Theorem 3.2, we can prove some sort of length-increasing property for M_{i_0}. That is, for almost all $x \in \Sigma^*$, M_{i_0} on $\langle i_0, x\rangle$ queries some y in A whose length is larger than $t^{-1}(|\langle i_0, x\rangle|)$ (the proof of this fact is almost the same as that of Theorem 3.2, and is omitted here). Thus, letting p be a polynomial time bound for M_{i_0}, we have

$(\overset{\infty}{\exists} x \in \Sigma^*)[\, M_{i_0}$ queries some y in A such that $t^{-1}(|\langle i_0, x\rangle|) \leq |y| \leq p(|\langle i_0, x\rangle|)\ ]$.

So,

$$(\exists c_0)(\forall n > c_0)(\exists y \in A)[\, t^{-1}(n) \leq |y| \leq p(n)\ ].$$

Hence,

$$(\exists c_1)(\forall n > c_1)(\exists y \in A)[\, n \leq |y| \leq p \circ t(n)\ ].$$

That is, A contains at least one element of length between c_1 and $p \circ t(c_1)$, and one of length between $p \circ t(c_1)$ and $(p \circ t) \circ (p \circ t)(c_1)$, and so on. Hence, $\|A^{\leq n}\|$

$\geq k$ for almost all n, where k is the largest integer such that $(p \circ t)^k(c_1) \leq n$. Since c_1 is a constant, there exists some constant c such that $\mathrm{cens}_A(n) \geq (p \circ t)^*(n) + c$ for all n. □

As an immediate consequence of this theorem, we can prove that all $\leq_T^p$-complete sets for EXP have $\omega(\log\log n)$-density. Also a little stronger result is provable according to the same line.

Corollary 5.2. Let A be any $\leq_T^p$-hard set for EXP that is in EXPOLY (i.e., $DTIME(2^{\mathrm{poly}})$). Then $\mathrm{cens}_A(n) = \omega(\log\log n)$. □

Although we have only the above partial result concerning the most general reducibility, i.e., $\leq_T^p$-reducibility, we can prove much stronger results for more restricted type of reducibilities. Berman and Hartmanis [BH77] showed that all $\leq_m^p$-hard sets for EXP have high density; and Balcâzar and Schöning [BS84] showed that they are all bi-exponentially dense. Here we will obtain more general ones.

First consider hard sets for EXP with respect to $\leq_c^p$- and $\leq_d^p$-reducibilities.

Theorem 5.3.

(1) For all $\leq_c^p$-hard sets A for EXP, there exists some constant ϵ, $0 < \epsilon$, $\mathrm{cens}_A(n) = \omega(2^{n^\epsilon})$; and

(2) For all $\leq_d^p$-hard sets A for EXP, there exists some constant ϵ, $0 < \epsilon$, $\mathrm{cens}_A(n) = \omega(2^{n^\epsilon})$.

Thus, every $\leq_c^p$- (resp., $\leq_d^p$-) hard set for EXP is bi-exponentially dense.

Proof. (1) Let $\{N_i\}_{i>0}$ be an enumeration of polynomial time bounded deterministic transducers. We will define the set C to be $\cup_{i>0} C_i$, where for all $i > 0$, the set $C_i \subseteq \{i\} \times \Sigma^*$ will be described by a variation of the stage construction method (see, the proof of Theorem 3.5 for the "stage construction").

Let i be any integer and fixed. Then the set C_i is defined in the following way:

```
global var $X$; $C_i$;
stage $x = \lambda$; (we use $\lambda$ to denote null string)
  $X \leftarrow \emptyset$;
  $C_i \leftarrow \emptyset$; (i.e., $\lambda \notin C_i$ is determined)
stage $x \in \Sigma^+$;
  compute $N_i(\langle i, x\rangle)$ and let $\{a_1, \cdots, a_k\}$ be the associated set of $N_i(\langle i, x\rangle)$;
  (if either the computation of $N_i(\langle i, x\rangle)$ needs more than $2^n$ steps or $N_i(\langle i, x\rangle)$
   is not a tt-condition, then go to the next stage)
  if $\{a_1, \cdots, a_k\} \not\subset X$ then
  (Case 1)
    $C_i \leftarrow C_i \cup \{\langle i, x\rangle\}$; (i.e., $\langle i, x\rangle \in C_i$ is determined)
    $X \leftarrow X \cup \{a_1, \cdots, a_k\}$
  else
  (Case 2)
    $C_i \leftarrow C_i$ (i.e., $\langle i, x\rangle \notin C_i$ is determined)
  end-if
end-construction.
```

In the above stage construction, C_i is defined by the lexicographic order from stage λ: at each stage $x \in \Sigma^*$, global variables are altered and $\langle i, x\rangle$ is determined whether it is in C_i; they are achieved by some deterministic algorithm within exponential time. Finally define the desired set C to be $\cup_{i>0} C_i$. Then it is easy to show that $C \in$ EXP.

Let A be any $\leq_c^P$-hard set for EXP. Then there exists a $\leq_c^P$-reduction from C to A. Let N_{i_0} achieves this reduction; and consider the stage construction of C_{i_0}. Then $\{a_1, \cdots, a_k\} \subseteq A$ if (Case 1) is chosen during the stages. Also note that X is the set of all strings appeared as elements of associated sets in the previous stages. Thus, if (Case 1) occurs, all elements of $\{a_1, \cdots, a_k\}$ are in A and at least one a_j did not appear in the previous stages.

Next consider how often (Case 1) occurs in the stage construction of C_{i_0}. Note that $X \subseteq A$ (we can prove this fact by induction on stage x). Thus, $\{a_1, \cdots, a_k\} \not\subseteq X$ for every stage x such that x is sufficiently long. Suppose otherwise: that is, $\{a_1, \cdots, a_k\} \subseteq X$ at some stage x. Then (Case 2) is chosen; thus, $\langle i_0, x\rangle$ is not in C_{i_0}, so nor in C. Hence, $\{a_1, \cdots, a_k\} \cap A^c \neq \varnothing$ since N_{i_0} is a $\leq_c^p$-reduction from C to A. Note that $\{a_1, \cdots, a_k\} \subseteq X$ from the assumption; hence, $X \cap A^c \neq \varnothing$, which contradicts the fact that $X \subseteq A$. Therefore if x is sufficiently long, (Case 2) never occurs at stage x: that is, (Case 1) occurs at almost all stages. Define $r_1(n)$ to be the number of stages x, $|x| \leq n$, such that (Case 1) is chosen. Then there exists some constant $c > 0$ such that $r_1(n) \geq 2^{cn}$ for all $n > 0$.

It follows from the above discussions that at least 2^{cn} different $a \in A$ appears during the all stages x, $|x| \leq n$. Here note that for every such a and the corresponding stage x, we have $|a| \leq p(|\langle i_0,x\rangle|)$, where p is a polynomial time bound for N_{i_0}. Hence, for some constant c', we have that $\|A^{p(n+c')}\| \geq 2^{cn}$ for all n. This proves that A is of exponential density.

(2) Although we will define the set D in the way similar to the above, the arguments are a little more complicated. For each $i > 0$, define D_i as follows:

```
global var $W$; $Y$; $D_i$;
stage $x = \lambda$;
  $W \leftarrow \varnothing$; $Y \leftarrow \varnothing$;
  $D_i \leftarrow \varnothing$; (i.e., $\lambda \notin D_i$ is determined)
stage $x \in \Sigma^+$;
```

```
    compute $N_i(\langle i, x\rangle)$ and let $\{a_1, \cdots, a_k\}$ be the associated set of $N_i(\langle i, x\rangle)$;
    (if either the computation of $N_i(\langle i, x\rangle)$ needs more than $2^n$ steps or $N_i(\langle i, x\rangle)$
     is not a tt-condition, then go to the next stage)
    if $\{a_1, \cdots, a_k\} \cap W = \varnothing$ then
    (Case 1)
      $D_i \leftarrow D_i \cup \{\langle i, x\rangle\}$; (i.e., $\langle i, x\rangle \in D_i$ is determined)
      $W \leftarrow W \cup \{a_1, \cdots, a_k\}$
    else
    (Case 2)
      $D_i \leftarrow D_i$; (i.e., $\langle i, x\rangle \notin D_i$ is determined)
      $W \leftarrow W - \{a_1, \cdots, a_k\}$; $Y \leftarrow Y \cup \{a_1, \cdots, a_k\}$
    end-if
  end-construction.
```

Define D to be $\cup_{i>0} D_i$.

Let A be any $\leq_{\mathrm{d}}^{\mathrm{p}}$-hard set for EXP. Since $D \in$ EXP, there exists a $\leq_{\mathrm{d}}^{\mathrm{p}}$-reduction, say N_{i_0}, from D to A. So, consider the stage construction of D_{i_0}. Then almost in the same way as (1), we can prove that if (Case 1) occurs, then there exists some a_j in $\{a_1, \cdots, a_k\}$ which did not appear in the previous stages (here we use the fact that $W \cup Y$ is the set of all strings that appeared before and that $Y \subseteq A^{\mathrm{c}}$).

Then the next problem is how often (Case 1) occurs. Let n be sufficiently large integer and fixed. Define $r_1(n)$ (resp., $r_2(n)$) to be the number of stages x, $|x| \leq n$, in the construction of D_{i_0} such that (Case 1) (resp., (Case 2)) occurs. Also define $w(x)$ and $w'(x)$ to be the number of elements added to W and removed from W at stage x respectively. According to the argument similar to (1), we can prove that W is not empty for almost all stages. Thus, we have $\Sigma_{|x|\leq n} w'(x) \leq \Sigma_{|x|\leq n} w(x)$. So, $r_2(n) \leq \Sigma_{|x|\leq n} w'(x) \leq \Sigma_{|x|\leq n} w(x) \leq r_1(n)\cdot p(n)$, where p is a polynomial time bound for N_{i_0}. Note that $r_1(n) + r_2(n)$, the number of the all stages x, $|x| \leq n$, is more than 2^n (since $|\Sigma| \geq 2$). Hence, we have $r_1(n) \geq 2^n / (1 + p(n))$. Therefore there exists some constant $c > 0$ such that $r_1(n) > 2^{cn}$ for some constant all $n > 0$. The rest of the proof is the same as (1). □

Corollary 5.4. [BH77, BS84] All $\leq^{\mathrm{p}}_{\mathrm{m}}$-hard sets for EXP are bi-exponentially dense. □

Next we consider another extension of Corollary 5.4: consider the density of $\leq^{\mathrm{p}}_{\mathrm{btt}}$-hard sets

Theorem 5.5. All $\leq^{\mathrm{p}}_{\mathrm{btt}}$-hard sets for EXP are bi-exponentially dense.

Proof. We will define the set E in the same style as the above proofs: that is, the set E is defined to be $\cup_{i>0} E_i$; and, for every $i > 0$, define the set $E_i \subseteq \{i\} \times \Sigma^*$ by a variation of the stage construction method.

Here we use several notations. Let $\{N_i\}_{i>0}$ denote an enumeration of polynomial time bounded transducers. For a set A, recall C_A denote the characteristic function for A: i.e., for every x, $C_A(x) =$ **true** iff $x \in A$. To simplify our notations, we use 0 and 1 to denote **false** and **true** respectively. A *vector of length n* is the ordered sequence of n elements. We use $\langle c_1 \cdots c_u \rangle$ to denote the vector of length u whose elements are $c_1, \cdots, c_u$. A *0-1 vector* is a vector whose elements are 0 or 1, and a *characteristic vector* of $\langle a_1 \cdots a_u \rangle$ is the 0-1 vector $\langle c_1 \cdots c_u \rangle$ such that $c_i = C_A(a_i)$ for all i, $1 \leq i \leq u$. Then, for every $i > 0$, the set E_i is defined as follows.

```
global var TOUCHED; LOOKING; $S_{\langle a_1 \cdots a_u \rangle}$; $B_{\langle a_1 \cdots a_u \rangle}$; $E_i$;
($S_{\langle a_1 \cdots a_u \rangle}$ and $B\langle a_1 \cdots a_u \rangle$ are defined for all $u \geq 0$ and all vector $\langle a_1 \cdots a_u \rangle$)
stage $x = \lambda$;
  TOUCHED ← $\varnothing$; LOOKING ← $\varnothing$;
  $E_i$ ← $\varnothing$  (i.e., $\lambda \notin E_i$ is determined)
stage $x \in \Sigma^+$;
```

```
(1) compute $N_i(\langle i, x\rangle)$ and let $\{e_1, \cdots, e_k\}$ be the associated set of $N_i(\langle i, x\rangle)$;
    (if either the computation of $N_i(\langle i, x\rangle)$ needs more than $2^n$ steps or $N_i(\langle i, x\rangle)$
     is not a tt-condition, then go to the next stage)
(2) $\{a_1, \cdots, a_u\} \leftarrow \{e_1, \cdots, e_k\} \cap TOUCHED$;
    $\{b_1, \cdots, b_v\} \leftarrow \{e_1, \cdots, e_k\} \cap TOUCHED^c$;
    (we assume that $a_1 < \cdots < a_u$ by lexicographic order; and let $\alpha'$ be a
     Boolean function such that $\alpha'(a_1, \cdots, a_u, b_1, \cdots, b_v) = \alpha(e_1, \cdots, e_k)$)
    if $\langle a_1 \cdots a_u\rangle \notin LOOKING$ then
      $LOOKING \leftarrow LOOKING \cup \{\langle a_1 \cdots a_u\rangle\}$
      $S_{\langle a_1 \cdots a_u\rangle} \leftarrow$ the set of all 0-1 vector of length $u$;
      $B_{\langle a_1 \cdots a_u\rangle} \leftarrow \varnothing$
    end-if;
(3) $B_{\langle a_1 \cdots a_u\rangle} \leftarrow B_{\langle a_1 \cdots a_u\rangle} \cup \{b_1, \cdots, b_v\}$;
(4) $S_0 \leftarrow S_{\langle a_1 \cdots a_u\rangle} \cap \{\langle c_1 \cdots c_u\rangle : \alpha'(c_1, \cdots, c_u, 0, \cdots, 0) = 0\}$;
    $S_1 \leftarrow S_{\langle a_1 \cdots a_u\rangle} \cap \{\langle c_1 \cdots c_u\rangle : \alpha'(c_1, \cdots, c_u, 0, \cdots, 0) = 1\}$;
    if $\|S_0\| \leq \|S_1\|$ then (Case 1)
                  $S_{\langle a_1 \cdots a_u\rangle} \leftarrow S_0$;
                  $E_i \leftarrow E_i$  (i.e., $\langle i, x\rangle \notin E_i$ is determined)
            else (Case 2)
                  $S_{\langle a_1 \cdots a_u\rangle} \leftarrow S_1$;
                  $E_i \leftarrow E_i \cup \langle i, x\rangle$  (i.e., $\langle i, x\rangle \in E_i$ is determined)
    end-if;
(5) if (*) $S_{\langle a_1 \cdots a_u\rangle} = \varnothing$ (this includes the case that $u = 0$) then
      $TOUCHED \leftarrow TOUCHED \cup B_{\langle a_1 \cdots a_u\rangle}$;
      $LOOKING \leftarrow LOOKING - \{\langle a_1 \cdots a_u\rangle\}$
    end-if
end-construction.
```

Then there exists a $\leq^{\mathrm{p}}_{\mathrm{btt}}$-reduction from E to A since $E \in EXP$; so, let N_{i_0} achieve this reduction. Since N_{i_0} is a $\leq^{\mathrm{p}}_{\mathrm{btt}}$-reduction, the norm of the truth-table $N_{i_0}(x)$ is always bounded by some constant; let m denote this constant.

Consider the stage construction of E_{i_0}. In the following, we will prove that if the condition (*) of step (5) holds at some stage, then we can ensure that some $b \in B_{\langle a_1 \cdots a_u \rangle}$ belongs to A. Thus, we call the situation that the condition (*) holds as an *ensuring state* and a stage where ensuring state occurs as an *ensuring stage*. Moreover we will show that such a string b has never been ensured in the previous stages: that is, at each ensuring stage, we obtain a new element of A.

First we state the meaning of global variables. Let x be arbitrary any stage and fixed, and let T, S, and B be respectively the content of *TOUCH*, $S_{\langle a_1 \cdots a_u \rangle}$ and $B_{\langle a_1 \cdots a_u \rangle}$ just before step (5) at stage x. Then we claim as follows.

Claim 1.

(1) No strings in T^c have appeared in $B_{\langle a_1 \cdots a_u \rangle}$ in the previous ensuring stage;

(2) $B \subseteq T^c$; and

(3) Either (i) some $\langle c_1 \cdots c_u \rangle \in S$ represents the characteristic vector of $\langle a_1 \cdots a_u \rangle$, or (ii) there exists some $b \in B$ which is in A.

Proof. (1) and (2) are easily proved by induction; thus we omit their proofs.

(3) We also prove it by induction. Suppose that either (i) or (ii) holds at all the stages before x. Thus, just before step (3), we have either (i') some $\langle c_1 \cdots c_u \rangle \in S_{\langle a_1 \cdots a_u \rangle}$ represents the characteristic vector of $\langle a_1 \cdots a_u \rangle$, or (ii') there exists some $b \in B_{\langle a_1 \cdots a_u \rangle}$ which is in A. Suppose that (Case 1) is chosen at step (4). Note that $\langle i_0, x \rangle \notin E_{i_0}$; thus the characteristic vector of $\langle a_1, \cdots, a_u, b_1, \cdots, b_v \rangle$ is not $\langle c_1, \cdots, c_u, 0, \cdots, 0 \rangle$ of any $\langle c_1 \cdots c_u \rangle$ in S_1 since $\alpha'(c_1, \cdots, c_u, 0, \cdots, 0) = 1$ for all $\langle c_1 \cdots c_u \rangle$ in S_1. There-

fore, either (i) the characteristic vector for $\langle a_1 \cdots a_u \rangle$ is $\langle c_1 \cdots c_u \rangle$ for some $\langle c_1 \cdots c_u \rangle$ in S_0 (= S), or (ii) some $b_j \in B$ belongs to A. The similar argument holds for (Case 2). This completes the proof. □

Then it follows from Claim 1 (3) that if the condition (*) holds at step (5), i.e., $S = \emptyset$, then some $b \in \{b_1, \cdots, b_v\}$ ($\subseteq B_{\langle a_1 \cdots a_u \rangle} = B$) belongs to A. Also it follows from Claim 1 (1) and (2) that such a string b has never been considered in the previous ensuring states.

Next consider how often an ensuring state occurs during the stage construction of E_{i_0}. Here notice that ensuring state occurs at least once since *TOUCHED* is initially empty and $u = 0$ when *TOUCHED* $= \emptyset$. First investigate how many times the same vector appears after one ensuring stage until the next one. Then we have the following claim.

Claim 2. Let $\langle a_1 \cdots a_u \rangle$ be any vector. Then $\langle a_1 \cdots a_u \rangle$ does not appear more than $u + 1$ stages between two consecutive ensuring stages.

Proof. We investigate the number of elements in $S_{\langle a_1 \cdots a_u \rangle}$; let s denote $\|S_{\langle a_1 \cdots a_u \rangle}\|$. Initially s is 2^u: if the vector $\langle a_1 \cdots a_u \rangle$ appears for the first time after one ensuring stage (i.e., $\langle a_1 \cdots a_u \rangle \notin$ *LOOKING*), then $S_{\langle a_1 \cdots a_u \rangle}$ is the set of all 0-1 vectors of length u. Note that every time when $\langle a_1 \cdots a_u \rangle$ appears, s decreases at least half of it at step (4). Thus, $\langle a_1 \cdots a_u \rangle$ cannot appear more than $u + 1$ times before the next ensuring stage. □

For every $j > 0$, let t_j be the number of elements in *TOUCHED* just after the jth ensuring stage.

Claim 3. For all $j > 0$, $t_j \leq j \cdot m^2$.

Proof. Let $j > 0$ be any fixed integer, and consider the jth ensuring stage. We show that $\|B_{\langle a_1 \cdots a_u \rangle}\| \leq m^2$ just before step (5) of this stage. Note that

$B_{\langle a_1 \cdots a_u \rangle}$ increases its elements at only the stage where $\langle a_1 \cdots a_u \rangle$ appears, and that it increases at most $v \leq m - u$ elements at one stage. Also note that $B_{\langle a_1 \cdots a_u \rangle}$ becomes empty at step (2) when $\langle a_1 \cdots a_u \rangle$ appears for the first time after $j - 1$th ensuring stage, and that $\langle a_1 \cdots a_u \rangle$ appears at most $u + 1$ stages between $j - 1$th and jth ensuring stages (see Claim 2). Hence, $\|B_{\langle a_1 \cdots a_u \rangle}\| \leq (u + 1)(m - u) \leq m^2$.

Since *TOUCHED* increases its elements only at step (5) of an ensuring stage, we have $t_j = t_{j-1} + \|B_{\langle a_1 \cdots a_u \rangle}\|$. Note that *TOUCHED* is set empty initially. Therefore $t_j \leq j \cdot m^2$. □

Next investigate the number of stages between the consecutive two ensuring stages. For any $j > 0$, let r_j be the number of stages between the $j - 1$th ensuring stage and the jth one. Then $r_1 \leq c_1$ for some constant c_1. Note that the same $\langle a_1 \cdots a_u \rangle$ appears at most $u + 1$ times and that there are $\mathrm{C}(t_{j-1}; u)$ different vectors of length u (we use $\mathrm{C}(p; n)$ to denote pth coefficient of $(1 + x)^n$). Also note that u is bounded by m. Thus, we have

$$r_j \leq 1 \cdot \mathrm{C}(t_{j-1}; 0) + \cdots + (m + 1) \cdot C(t_{j-1}; m) \leq t_{j-1}^m .$$

Hence, we claim as follows.

Claim 4. For all $j > 1$, $r_j \leq t_{j-1}^m$. □

Now we can estimate the number of ensuring stages among all the stages x, $\|x\| \leq n$, during the construction of E_{i_0}. Let n be any integer and fixed. And define X and j_0 to be respectively $\Sigma^{\leq n}$ and the number of ensuring stages among all the stages in X. Note that there are at least $2^{n+1} - 1$ elements in X. Thus it follows from the meaning of r_j that

$$(r_0 + 1) + (r_1 + 1) + \cdots + (r_{j_0+1} + 1) > 2^{n+1} - 1.$$

From Claim 3 and 4, we have

$$(c_0 + 1) + ((m^2)^m + 1) + \cdots + (((j_0 + 1)m^2)^m + 1) \geq 2^{n+1}.$$

Thus,

$$c_0 + j_0\cdot((j_0 + 1)m^2)^m \geq 2^{n+1}.$$

Note that m is a constant. Therefore, we have $j_0 \geq 2^{cn}$ for some constant $c > 0$: that is, the number of ensuring stages x, $\|x\| \leq n$, is more than 2^{cn}. Recall that we obtain some new element b in A at each ensuring stage (see Claim 1). Thus, according to the same arguments as the proof of Theorem 5.3, we conclude that A has an exponential density. Notice here that the complement of $\leq^{\mathrm{p}}_{\mathrm{btt}}$-hard set is also $\leq^{\mathrm{p}}_{\mathrm{btt}}$-hard. Thus, following the same proof, we can also show that A^{c} is of exponential density. Therefore A is of bi-exponential density.

Remark. Corollary 5.4 is also the corollary of this theorem. □

Note that Theorem 5.3, 5.4 and 5.5 are optimal: the following proposition shows this fact.

Proposition 5.6. For every ϵ, $0 < \epsilon < 1$, there exists a $\leq^{\mathrm{p}}_{\mathrm{m}}$-hard set A for EXP such that $\mathrm{cens}_A(n) = O(2^{n^\epsilon})$.

Proof. Let U be the standard $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set in EXP defined in Chapter 2. For every $k > 0$, consider the following set:

$$U_k = \{x0^{|x|^k} : x \in U\}.$$

Then the desired set A is U_k for some $k > 0$.

Other Complexity Classes

The results in this section also holds for any super-SUBEXP classes and any super-PSPACE classes. Also consider any complexity class $\mathcal{C}$ such that $\mathcal{C}$ contains some super-SUBEXP or super-PSPACE complexity class $\mathcal{C}_0$ for which $\leq^{\mathrm{p}}_r$-hard sets exist (where $\leq^{\mathrm{p}}_r$ is respectively $\leq^{\mathrm{p}}_{\mathrm{T}}$, $\leq^{\mathrm{p}}_{\mathrm{c}}$, $\leq^{\mathrm{p}}_{\mathrm{d}}$ or $\leq^{\mathrm{p}}_{\mathrm{btt}}$). Then it is clear that we have the above Theorems since all $\leq^{\mathrm{p}}_r$-hard sets for $\mathcal{C}$ are also $\leq^{\mathrm{p}}_r$-hard for $\mathcal{C}_0$. Hence, we have the above results also for nondeterministic time complexity classes such as NEXP.

Although we considered the density of "hard" sets, we can also discuss similar topics even for complexity classes which have no "hard" sets. That is, we can rewrite the above theorems to equivalent ones which do not use the concept of "hard set". Here we introduce some new notations.

Let $\leq_r^{\mathrm{p}}$ be any polynomial time reducibility and $d(n)$ be any integer valued function. We consider the class of languages that are $\leq_r^{\mathrm{p}}$-reducible to sets of $O(d(n))$-density, which is denoted as $\leq_r^{\mathrm{p}}$-$d(n)$ and defined as follows:

$$\leq_r^{\mathrm{p}}\text{-}d(n) = \{\, L : \text{there exists } A \text{ such that} \quad \text{(i) } L \leq_r^{\mathrm{p}} A \ \textit{and} \ \text{(ii) } \mathrm{cens}_A(n) = O(d(n)) \,\}.$$

Also we use $\leq_r^{\mathrm{p}}$-*poly* to denote the class $\cup\, \{\leq_r^{\mathrm{p}}\text{-}p(n) : p \text{ is a polynomial}\}$, i.e., the class of languages which are $\leq_r^{\mathrm{p}}$-reducible to some sparse set.

Then we have the following statements that have the same meaning as the above corresponding theorems.

Theorem 5.7. Let $d(n)$ be any integer valued function such that $d(n) = o(2^{n^\epsilon})$ for all ϵ, $0 < \epsilon$.

(1) There exists a set in EXP which is not $\leq_{\mathrm{T}}^{\mathrm{p}}$-reducible to any set A such that $\mathrm{cens}_A(n) = \omega(\log\log n)$ and $A \in$ EXPOLY;

(2) There exists a set in EXP which is not in $\leq_{\mathrm{c}}^{\mathrm{p}}$-$d(n)$ (resp., $\leq_{\mathrm{d}}^{\mathrm{p}}$-$d(n)$).

(3) There exists a set in EXP which is not $\leq_{\mathrm{btt}}^{\mathrm{p}}$-$d(n)$. □

According to the same argument as above, it is clear that we have the same results for all complexity class $\mathcal{C}$ which includes some super-SUBEXP or super-PSPACE complexity class.

How about complexity classes which have polynomial time bounds: for example, the classes NP, BPP, UP, etc. We have the following results for these classes.

Theorem 5.8.

(1) [Ma82] If P $\neq$ NP, then there exists a set in NP which is not $\leq^{\mathrm{p}}_{\mathrm{m}}$-reducible to any sparse set in NP; and

(2) [Fo79] If P $\neq$ NP, then there exists a set in co-NP which is not in $\leq^{\mathrm{p}}_{\mathrm{m}}$-poly. □

Theorem 5.9. [Ad78, Sc86a] BPP $\subseteq$ $\leq^{\mathrm{p}}_{\mathrm{T}}$-poly. □

Theorem 5.10. [Wa86b] If P $\neq$ UP, then there exists a set in UP which is not in $\leq^{\mathrm{p}}_{\mathrm{m}}$-poly. □

5.2. Non-Existence of Polynomial Time Pseudo Algorithms

One of the applications of our observations in section 5.1 is to show (non) existence of feasible pseudo algorithms.

Even a set which has no polynomial time deterministic algorithms may have an another type of feasible algorithm. We consider the following polynomial time *pseudo algorithms*:

Let A be any recursive subset of $\{0, 1\}^*$.

(1) [MP79] APT machine (almost polynomial time machine):

A deterministic machine M is *APT machine* for A if it satisfies

(i) M is polynomial time bounded, and for every x, $M(x)$ always outputs "accept", "reject" or "?";

(ii) $\{x : M(x) =$ "accept" $\} \subseteq A$ *and* $\{x : M(x) =$ "reject" $\} \subseteq A^c$; and

(iii) $\{x : M(x) =$ "?" $\}$ is a sparse set.

(2) [OS86] 1-APT machine:

A deterministic machine M is 1-*APT machine* for A if it satisfies

(i) M is polynomial time bounded, and for every x, $M(x)$ always outputs "accept", "reject" or "?";

(ii) $\{x : M(x) =$ "accept" $\} \subseteq A$ *and* $\{x : M(x) =$ "reject" $\} \subseteq A^c$; and

(iii) $\{x : M(x) =$ "?" $\} \cap A$ is a sparse set.

(3) [Ye83, Sc86b] p-close machine:

A polynomial time bounded deterministic machine M is *p-close machine* for A if $L(M) \Delta A$ is sparse, where $X \Delta Y$ denotes the symmetric difference between X and Y.

(4) [KL80] polynomial size circuit:

A set of logical circuits $\{C_n\}_{n>0}$ is a *polynomial size circuit* for A if it satisfies

(i) for every $n > 0$, C_n is a n-input and 1-output Boolean circuit whose size is polynomially bounded (see, e.g., [Sc86a] for a precise definition of a Boolean circuit and its size); and

(ii) for all $n > 0$ and every $x \in \{0, 1\}^*$, $x \in A \leftrightarrow C_n(x) = 1$.

(*Note:* (1-)APT machines are defined in slightly different ways in the original definitions [MP79, OS86]. However the above gives the same definition provided we consider recursive sets.)

In the following we define the class of languages having the above pseudo algorithms.

Definition 5.1.

(1) APT $= \{L$: there exists an APT machine for $L\}$;

(2) 1-APT $= \{L$: there exists a 1-APT machine for $L\}$; and

(3) p-close-P $= \{L$: there exists an p-close machine for $L\}$.

Remark. Yesha [Ye83] originally defined the notion "closeness" in more general way than the one in the definition of "p-close-P". In general, "closeness" is defined as follows: let $\mathcal{C}$ and $d(n)$ be any complexity class and any function respectively; for any set A, A is *$d(n)$-close* to $\mathcal{C}$ if there exists a set

B in $\mathcal{C}$ such that $A \Delta B$ has less than $d(n)$-density. Here we also use this general concept and define the class $d(n)$-close-P to be the class of languages that are $O(d(n))$-close to P. □

Karp and Lipton [KL80] introduced some type of non uniform complexity classes. Among them, the class P/poly characterizes the sets having polynomial size circuits.

Definition 5.2. The class P/poly is the class of languages A for which there exists a set $B \in$ P and a polynomial p such that

$$(\forall n)(\exists y{:}|y| \leq p(n))[\, A^{=n} = \{x \in \Sigma^{=n} : \langle x, y\rangle \in B\,\}\,].$$

The following characterization is due to Pippenger.

Proposition 5.11. [Pi79] For any set $A \subseteq \{0, 1\}^*$, A has a polynomial size circuit if and only if A is in P/poly.

Remark. Recall that our finite alphabet Σ is the set $\{0, 1\}$ (Chapter 2): that is, all sets considered here are subsets of $\{0, 1\}^*$. Hence, we regard P/poly as the class of languages having polynomial size circuits. □

It is easy to show the following relations between the properties of "having polynomial time pseudo algorithms" and the polynomial time reducibilities to sparse sets.

Proposition 5.12.

(1) APT $\subsetneq$ 1-APT; and

(2) [MP79, OS86] 1-APT $\subseteq \leq_{\mathrm{m}}^{\mathrm{P}}$-poly. □

Proposition 5.13. [Sc86b]. Let $d(n)$ be any integer valued function such that $d(n) = O(2^{n^{\epsilon}})$ for some constant ϵ, $0 < \epsilon < 1$. Then $\leq_{\mathrm{m}}^{\mathrm{P}}$-$d(n)$ $\subsetneq$ $d(n)$-close-P $\subsetneq$ $\leq_{\text{1-tt}}^{p}$-$d(n)$. Thus, we have that $\leq_{\mathrm{m}}^{\mathrm{P}}$-poly $\subsetneq$ p-close-P $\subsetneq$ $\leq_{\text{1-tt}}^{p}$-poly. □

Theorem 5.14.

(1) [BK86] For all integer valued function $d(n)$, we have

$$\leq^{p}_{1\text{-tt}}\text{-}d(n) \subseteq \leq^{p}_{2\text{-tt}}\text{-}d(n) \subseteq \cdots \leq^{p}_{tt}\text{-}d(n).$$

Especially we have

$$\leq^{p}_{1\text{-tt}}\text{-poly} \subsetneqq \leq^{p}_{2\text{-tt}}\text{-poly} \subsetneqq \cdots \leq^{p}_{tt}\text{-poly}.$$

(2) [BK86] $\leq^{p}_{tt}\text{-poly} = \leq^{p}_{T}\text{-poly}$; and

(3) [Sc84] $\leq^{p}_{T}\text{-poly} = \text{P/poly}$. □

Therefore we have the following relations.

$$\begin{tikzcd}
\text{APT} \arrow[r, phantom, "\subsetneqq"] & \text{1-APT} \arrow[dr, phantom, sloped, "\not\subseteq"] & & \text{p-close-P} & & & \text{P/poly} \arrow[d, phantom, sloped, "="] \\
& & \leq^{p}_{m}\text{-poly} \arrow[ur, phantom, sloped, "\not\subseteq"] & & \leq^{p}_{1\text{-tt}}\text{-poly} \arrow[r, phantom, "\subsetneqq"] & \leq^{p}_{tt}\text{-poly} \arrow[r, phantom, "="] & \leq^{p}_{T}\text{-poly}
\end{tikzcd}$$

Figure 5.1. The comparison of several feasible pseudo algorithms.

From the results in section 5.1 and the above observations, we can show non existence results of pseudo polynomial time algorithms.

Theorem 5.15. Let $d(n)$ be any function such that $d(n) = o(2^{n^{\epsilon}})$ for all ϵ, $0 < \epsilon$. Then no $\leq^{p}_{btt}$-hard sets for EXP are in $d(n)$-close-P. Thus no $\leq^{p}_{btt}$-hard sets have p-close machines, nor (1-)APT machines. □

Theorem 5.16.

(1) If P $\neq$ NP then no $\leq^{p}_{m}$-hard sets for NP (nor co-NP) are (1-)APT; and

(2) If P $\neq$ UP then there exists a set in UP which is not $\leq^{p}_{m}$-reducible to any (1-)APT set. □

6. Polynomial Lowness and EXP Lowness

How does one measure the complexity of a given set? We have considered (i) computational complexity for accepting it; (ii) similarity to some well-known difficult set such as SAT; and (iii) reducibility to a set of small density. In this chapter, we consider another method: the method of estimating how useful a given set is when it is used in some computations.

The concept "usefulness" seems to offer new measure of the amount of information in a set. For example, consider P( )-computation (recall that P( )-computation is polynomial time deterministic oracle computation; and P(X) denotes the class of languages accepted by some P( )-computation using oracle X). If P(A) = P but P(B) = EXP, then B is more useful in P( )-computation than A; thus B is considered to encode more information than A.

Among many ways in using a set in a computation, we consider a specific one: we investigate usefulness of a set as an *oracle*. That is, we consider how helpful a given set is in $\mathcal{C}$( )-computation for some complexity class $\mathcal{C}$. The concept "lowness/highness" states this type of usefulness. Schöning [Sc83] first introduced "lowness/highness" into the context of NP-computation: they are called *polynomial lowness* and *polynomial highness*. Schöning and his coauthors have extensively studied these notions [BBS86, KS85, Sc86a, Sc87]; and it seems that the notions of polynomial lowness/highness yield some hierarchies showing the degree of intractability in NP. Although we have some evidence for our intuition in recursive function theory, we can not prove it so far. We consider a similar problem in the context of EXP-computation. Through this investigation, we obtain additional evidence for our intuition.

We first introduce the concepts of polynomial lowness and highness, and review the previous researches and conjectures.

The polynomial lowness and highness are defined in terms of *polynomial time hierarchy*, which is defined as follows.

Definition 6.1. [St77]

(1) For every integer $k \geq 0$, we define Σ_k^{p} as follows:

$$\Sigma_0^{p} = \mathrm{P} \text{ and } \Sigma_{k+1}^{\mathrm{p}} = \mathrm{NP}(\Sigma_k^{\mathrm{p}}).$$

That is, $\Sigma_k^{\mathrm{p}} = \underbrace{\mathrm{NP}(\mathrm{NP}(\cdots \mathrm{NP}(\mathrm{P})}_{\mathrm{k}} \cdots))$.

(2) The *polynomial time hierarchy* is the structure $\{\Sigma_k^{\mathrm{p}}\}_{k\geq 0}$. Let PH denote the class $\cup\{\Sigma_k^{\mathrm{p}} : k \geq 0\}$.

(3) For any set A, we define the polynomial time hierarchy relative to A, $\{\Sigma_k^{\mathrm{p}}(A)\}_{k\geq 0}$, in the same way as we do the relativized class $\mathrm{P}(A)$ from P.

Remark. Here we use alternating Turing machine computation model in order to define, e.g., $\Sigma_k^{\mathrm{p}}(A)$ in the same way as $\mathrm{P}(A)$: Σ_k^{p}-computation is one type of alternating computation. Precisely speaking, the usual definition of the relativized polynomial time hierarchy is slightly different from ours [St77, Wr77]. However we can easily show that they give the same definition.

It is easy to show that the class $\{\Sigma_k^{\mathrm{p}}\}_{k\geq 0}$ forms a hierarchy: that is, $\Sigma_k^{\mathrm{p}} \supseteq \Sigma_j^{\mathrm{p}}$ for all $k > j \geq 0$. Furthermore, we conjecture that they consist a real hierarchy (i.e., the non-collapse conjecture on the polynomial time hierarchy):

Conjecture 6.1. $\Sigma_0^{\mathrm{p}} \subsetneq \Sigma_1^{\mathrm{p}} \cdots \subsetneq \mathrm{PH}$. □

The concepts of *polynomial lowness* and *polynomial highness* are defined as follows.

Definition 6.2. [Sc83]

(1) For any set A in NP and every $k \geq 0$, A is Σ_k^{p}*-low* if $\Sigma_k^{\mathrm{p}} = \Sigma_k^{\mathrm{p}}(A)$, and A is Σ_k^{p}*-high* if $\Sigma_{k+1}^{\mathrm{p}} = \Sigma_k^{\mathrm{p}}(A)$.

(2) For every $k \geq 0$, L_k^P (resp., H_k^P) is the class of Σ_k^P-low sets (resp., Σ_k^P-high sets). The *polynomial low hierarchy* (resp., *polynomial high hierarchy*) refers the structure $\{\mathrm{L}_k^\mathrm{P}\}_{k\geq 0}$ (resp., $\{\mathrm{H}_k^\mathrm{P}\}_{k\geq 0}$).

The above notions of lowness (resp., highness) are called in general *polynomial lowness* (resp., *polynomial highness*). It is easy to see in the definition that the lowness/highness degrees somehow state the amount of information in a given set: i.e., they measure the usefulness of a set as an oracle in some alternating polynomial time computation. So, we discuss the amount of information in a set in terms of lowness/highness.

Consider any sets A and B in L_k^P and in $\mathrm{L}_{k+1}^\mathrm{P}$ respectively. Then it follows from the definition that any oracle information of A (used in $\Sigma_k^\mathrm{P}()$-computation) is Σ_k^P-computable whereas that of B (used in $\Sigma_{k+1}^\mathrm{P}()$-computation) may not. Thus, it seems that some sets in $\mathrm{L}_{k+1}^\mathrm{P}$ contains more oracle information than any set in L_k^P: that is, $\mathrm{L}_{k+1}^\mathrm{P}$ is more difficult than L_k^P. Hence, it is natural to conjecture that the class $\{\mathrm{L}_k^\mathrm{P}\}_{k\geq 0}$ (resp., $\{\mathrm{H}_k^\mathrm{P}\}_{k\geq 0}$) forms a hierarchy of difficulty in NP.

Let us consider this conjecture a little more and look at the structure of NP from this point of view. First, it is not difficult to show that the class $\{\mathrm{L}_k^\mathrm{P}\}_{k>0}$ (resp., $\{\mathrm{H}_k^\mathrm{P}\}_{k>0}$) forms a hierarchy: i.e., $\mathrm{L}_k^\mathrm{P} \supseteq \mathrm{L}_j^\mathrm{P}$ for all $k > j \geq 0$.

Proposition 6.1. [Sc83]

(1) $\mathrm{P} = \mathrm{L}_0^\mathrm{P} \subseteq \mathrm{L}_1^\mathrm{P} \subseteq \cdots$; and

(2) $\{\leq_\mathrm{T}^\mathrm{P}\text{-complete sets in NP}\} = \mathrm{H}_0^\mathrm{P} \subseteq \mathrm{H}_1^\mathrm{P} \subseteq \cdots$. □

Thus, the class $\{\mathrm{L}_k^\mathrm{P}\}_{k\geq 0}$ and $\{\mathrm{H}_k^\mathrm{P}\}_{k\geq 0}$ respectively form hierarchies between P and the class of $\leq_\mathrm{T}^\mathrm{P}$-complete sets. Also assuming that the polynomial time hierarchy does not collapse, we can prove that these two hierarchies are disjoint.

Proposition 6.2. [Sc83] For every $k \geq 0$, $L_k^p \cap H_k^p \neq \emptyset$ if and only if the polynomial time hierarchy collapses to Σ_k^p, i.e., $\Sigma_k^p = \Sigma_{k+1}^p = \cdots = \mathrm{PH}$. □

Thus, we conjecture the following structure in NP.

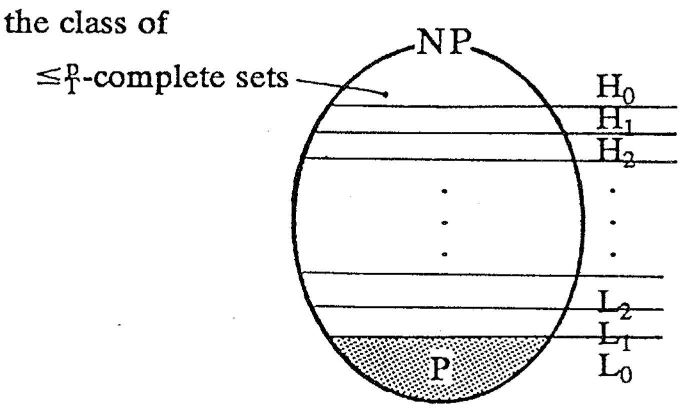


Figure 6.1. The high/low hierarchy in NP.

Namely, the polynomial time low and high hierarchies seem to offer another measure for analyzing complexity below $\leq_T^p$-complete sets, for which we have not investigated yet.

Here we should mention that most of our intuitions about the polynomial low and high hierarchies have not been proved yet. Especially, we do not know whether the polynomial low and high hierarchies are real hierarchies: i.e., they do not collapse. It is clear that if the polynomial time hierarchy collapses to Σ_k^p then the polynomial low (resp., high) hierarchy also collapse to L_k^p (resp., H_k^p): what follows is its brief proof. Assume that $\Sigma_k^p = \Sigma_{k+1}^p$ and consider any set L in L_{k+1}^p (we will show that L is in L_k^p). Then it follows from the definition of L_{k+1}^p that all sets in $\Sigma_{k+1}^p(L)$ are Σ_{k+1}^p-computable, where $\Sigma_{k+1}^p = \Sigma_k^p$ from the assumption; thus, they are also Σ_k^p-computable. Hence, if $\Sigma_k^p = \Sigma_{k+1}^p$ then $L_k^p = L_{k+1}^p$ (and the same argument holds for the high hierarchy). Note that we cannot prove the converse so far; so, we do not know whether the non-collapse conjecture on the polynomial time hierarchy implies the non-collapsing low/high hierarchies.

In the following, we consider the first step of such non-collapsing conjecture: that is, investigate whether $L_0^p \subsetneq L_1^p$. Then we have the following results.

Theorem 6.3.

(1) [Sc83] L_1^p = NP $\cap$ co-NP; thus $L_0^p \neq L_1^p$ if and only if P $\neq$ NP $\cap$ co-NP; and

(2) [KS85] If EXP $\neq$ NEXP, then $L_0^p \neq L_1^p$ □

Here we extend polynomial lowness and highness to the exponential case and consider low and high hierarchies in EXP. There are several ways to extend polynomial lowness (highness) to the exponential case. The "exponential lowness" considered in [BORW86] is one such approach. Although "exponential lowness" gives another characterization of the polynomial low hierarchy, it is not a counterpart of the polynomial low hierarchy in EXP. Thus, we take another approach: we replace NP by EXP and define low and high hierarchies in a similar way.

In order to define low and high hierarchies, we need to introduce a hierarchy like the polynomial time hierarchy. Note that EXP $\subsetneq$ EXP(EXP) $\subsetneq$ $\cdots$; so, this property of EXP yields us the real hierarchy (*Cf.* the exponential time hierarchy [He84, BORW86]). We defined the *EXP hierarchy* as follows:

Definition 6.3.

(1) For every $k \geq 0$, we define E_k as follows:

$$E_0 = P \text{ and } E_{k+1} = \text{EXP}(E_k).$$

That is, $E_k = \underbrace{\mathrm{EXP}(\mathrm{EXP}(\cdots \mathrm{EXP}}_{k}(P) \cdots))$.

(2) The *EXP hierarchy* is the structure $\{E_k\}_{k\geq 0}$.

Then "lowness/highness" in EXP is defined as follows:

Definition 6.4.

(1) For any set A in EXP and every $k \geq 0$, A is E_k*-low* if $E_k = E_k(A)$, and A is E_k*-high* if $E_{k+1} = E_k(A)$.

(2) For every $k \geq 0$, L_k^e (resp., H_k^e) is the class of E_k-low sets (resp., E_k-high sets). The *EXP low hierarchy* (resp., *EXP high hierarchy*) refers the structure $\{L_k^e\}_{k\geq 0}$ (resp., $\{H_k^e\}_{k\geq 0}$).

We call these lowness (resp., highness) notions as *EXP lowness* (resp., *EXP highness*) in general.

Now we investigate the same problems in the context of EXP-computation. Then we have the following facts which correspond Proposition 6.1 and 6.2.

Proposition 6.4.

(1) $P = L_0^e \subseteq L_1^e \subseteq \cdots$;

(2) $\{\leq_T^P$-complete sets in EXP$\} = H_0^e \subseteq H_1^e \subseteq \cdots$; and

(3) for every $k \geq 0$, $L_k^e \cap H_k^e = \varnothing$ (note that it is provable that the EXP-hierarchy does not collapse: *Cf.* Proposition 6.2). □

Now consider the problem of whether the low hierarchy does not collapse in EXP. In particular, we focus on the problem of whether $L_0^e \subsetneq L_1^e$. Then we have the following separation result, which gives some evidence for our intuition concerning the structure of the polynomial low hierarchy.

Theorem 6.5. $L_0^e \subsetneq L_1^e$.

Recall that $\mathrm{L}_0^e = \mathrm{P}$ and that L_1^e is the class of sets X in EXP such that $\mathrm{EXP}(X) = \mathrm{EXP}$ (In the following a set X such that $\mathrm{EXP}(X) = \mathrm{EXP}$ is called *EXP-low*). Thus the goal of our proof is to construct a set A such that A is EXP-low and is in EXP $-$ P.

We define the set A as a set of complex (noncompressible) strings: that is, we consider a set of high Kolmogorov complexity strings (the reader may wish to review Chapter 2 for notation on Kolmogorov complexity). Let $\epsilon > 0$ be arbitrarily fixed, and consider the set $\mathrm{K}[n/2, 2^{(1+\epsilon)n}]$. For brevity, we let K denote $\mathrm{K}[n/2, 2^{(1+\epsilon)n}]$ and $\overline{K}$ denote the complement of K. A simple counting argument is enough to show that for every $n > 0$, there exists at least one string in $\overline{K}$ of length n; there are not enough strings in Σ^* of length $n/2$ to map onto all the strings of length n.

Intuitively, $\overline{K}$ is a good candidate for a EXP-low set in EXP $-$ P. What follows is an intuitive explanation of our idea. Consider an exponential time oracle computation which uses $\overline{K}$ as an oracle. Note that $\overline{K} \in$ EXP. Thus, if the lengths of queries made by an oracle machine M are small, e.g., linearly bounded with respect to the length of input, then it can be simulated deterministically by an exponential time machine that makes no oracle queries. On the other hand, if M makes a long query with respect to the length of the input, then this query may be computable from a short description in exponential time and may be in K (i.e., not in $\overline{K}$). Thus, the answer to this query may be "no" (i.e., we do not need the oracle here). The next lemma establishes this type of property for the set $\overline{K}$.

Lemma 6.6. Let $\{N_i\}_{i>0}$ be a standard enumeration of deterministic transducers and let f_i denote the function computed by N_i. Consider any exponential time transducer N_i. Then there exists a d_i such that for almost all x, $|f_i(x)| > d_i \cdot |x|$ implies that $f_i(x) \in K$.

Proof. Let c be a constant such that N_i runs in time 2^{cn}. Let $d > 1$ be arbitrary. For any x such that $|f_i(x)| > d|x|$, we have $f_i(x) \in \mathrm{K}_i[n/d, 2^{c(n/d)}]$. From Proposition 2.2 we see that $\mathrm{K}_i[n/d, 2^{c(n/d)}] \subseteq \mathrm{K}[n/d+d', c'(n/d)\cdot 2^{c(n/d)}]$, where d' and c' are constants determined by N_i.

Choose a constant d_i such that $n/d_i + d' \leq n/2$ and $c'(n/d_i)\cdot 2^{c(n/d_i)} < 2^n$ for almost all n. Then we have, for almost all n,

$$\mathrm{K}[n/d+d', c'(n/d)\cdot 2^{c(n/d)}] \subseteq \mathrm{K}[n/2, 2^n] \subseteq \mathrm{K}[n/2, 2^{(1+\epsilon)n}].$$

Therefore, for almost all x such that $|f_i(x)| > d_i\cdot|x|$, we have $f_i(x) \in \mathrm{K}[n/2, 2^{(1+\epsilon)n}]$. □

However, we cannot prove the EXP-lowness of $\overline{K}$ in this way. The machine M may use some information obtained by answers to the previous stages. So the query may not result from a short description in exponential time even if it is long. But if the oracle is very sparse, we can compress the information about previous queries.

Thus define the desired set A to be any infinite subset of $\overline{K}$ in EXP with the property that for each $n > 0$, A contains at most one element of length between n and 2^n. For example, we can define A by

$$A = \{\, x : x \text{ is the smallest element (in lexicographic order) of } \overline{K} \text{ of length } 2^{\left.2^{\cdot^{\cdot^{2}}}\right\}m}, \text{ for some } m > 0 \,\}.$$

Then it is easy to show that this choice for A is in EXP. Moreover the following lemmas show that A is not in P.

Lemma 6.7. Let t be any time-constructible function. Let L be any infinite set accepted by a deterministic machine that runs in time t. Then there exists a deterministic transducer N_{i_0} such that $L \cap \mathrm{K}_{i_0}[\lceil \log n \rceil, 2^n\cdot t(n)]$ is infinite.

Proof. Let M be any deterministic machine that runs in time t and recognizes L. Then consider a transducer N_{i_0} which computes the function f_{i_0} defined as

$$f_{i_0}(\text{bin}(n)) = \begin{cases} \text{the smallest (in lexicographic order) } y \in L \text{ such that } |y| = n \\ \qquad\qquad\qquad \text{if such a } y \text{ exists,} \\ 0 \qquad\qquad\quad \text{if no such } y \text{ exists,} \end{cases}$$

where bin(n) denotes the binary representation of n on Σ^*.

It is easy to implement N_{i_0} so that it runs in time $2^n \cdot t(n)$. Thus, we have range(f_{i_0}) $= \mathrm{K}_{i_0}[\lceil \log n \rceil, 2^n \cdot t(n)] \subseteq L$. Since L is infinite, it is clear that range(f_{i_0}) is infinite. □

Lemma 6.8. If L is any infinite subset of $\overline{K}$, then L is not in P. Hence, the set A defined above is not in P.

Proof. Assume to the contrary that there is an infinite $L \in \mathrm{P}$ so that $L \subseteq \overline{K}$. By Lemma 6.7, for some polynomial p there exists a deterministic transducer N_{i_0} such that $L \cap \mathrm{K}_{i_0}[\lceil \log$ *nRCEIL*, $2^n \cdot p(n)]$ is infinite. But from Proposition 2.2 we have

$$\mathrm{K}_{i_0}[\lceil \log n \rceil, 2^n \cdot p(n)] \subseteq \mathrm{K}[\log n + d, (cn2^n \cdot p(n) \cdot \log p(n)) + c] \subseteq K$$

for almost all n. Therefore, $\overline{K} \cap K \neq \emptyset$, a contradiction. □

Now we show that A is EXP-low.

Lemma 6.9. If L is in EXP(A), then L is in EXP. Hence, EXP(A) = EXP.

Proof. Let M be a deterministic exponential time oracle machine that witnesses $L \in$ EXP(A), that is, $L = L(M, A)$. We must prove that L is in EXP.

Claim. There exists a constant $d > 0$ such that for almost all x, if y is a query made by M during its computation on x, then $|y| > d|x|$ implies y is not in A.

Proof. Let the running time of M be 2^n. Suppose that in M's computation on x relative to A, the oracle is queried about string y. Let $n = |x|$ and let

$q_1, \cdots, q_m$ ($m \leq 2^{cn}$) be the strings previously queried during this computation. We first show that with input x and some additional information, all of which requires less than $n + 2(2n + 2cn)$ bits to encode, this computation can be simulated up to this point without making any oracle calls.

Let $\{\alpha_1, \cdots, \alpha_k\}$ be the set of strings in A of length at most n. Notice that by the definition of A, $|\alpha_1| + \cdots + |\alpha_k| \leq 2n$. Hence, by using $2n$ bits of additional information, we can simulate an oracle call to A if the length of the string is at most n.

Note that $|q_i| \leq 2^{cn}$ for all i, $1 \leq i \leq m$. Furthermore, there is at most one element in A with length greater than n but no greater than 2^n. Since $2^{cn} < 2^{2^n}$ for almost all n, there exist at most two different q_j is in A and have lengths that are greater than n. Thus, if some q_j is in A but $|q_j|$, then we can indicate that fact by knowing its index j which costs at most cn bits of additional information for each q_j. This means that we can simulate the oracle calls to A even if the length of the string is greater than n, just so long as we have $2cn$ bits of additional information.

We can conclude that it is possible to simulate M's computation on x relative to A by making no oracle calls if we have enough additional information: at most $(2n + 2cn)$ bits of such information are required. This information can be added to the input string x by duplicating each bit; hence, at most $n + 2(2n + 2cn)$ bits are needed. Therefore, one can construct a deterministic exponential time transducer N_{i_0} which produces each query y from some z where z has less than $(5 + 4c)n$ bits. Now it follows from Lemma 6.6 that there exists d_{i_0} such that for almost all y, $|y| > d_{i_0} \cdot |z|$ implies $y \in K$ so that y is not in A. Letting d be $d_{i_0}(5 + 4c)$, we have $|y| > d|x| = d_{i_0}(5 + 4c) \cdot |x| \geq d_{i_0} \cdot |z|$ implies y is not in A, for almost all y. This concludes the proof of the Claim.

(*Note:* The technique used in this proof is similar to one used by Goldberg and Sipser [GS85].) □

To continue with the proof of the lemma, let M_A be a deterministic exponential time acceptor which recognizes A. Then there is a deterministic exponential time acceptor M_L which recognizes L without making any oracle queries that operates as follows:

```
begin
  input $x$;
  simulate $M$ on $x$ where
    if $M$ queries the oracle about a string $y$ and $|y| > d \cdot |x|$ then
      answer "NO" to the query;
    if $M$ queries the oracle about a string $y$ and $|y| \leq d \cdot |x|$ then
      begin
        simulate $M_A$ on $y$ to determine whether $y \in A$;
        if $y \in A$ then answer "YES" else "NO"
      end
end.
```

It is clear that M_L recognizes L and operates in exponential time. Thus, L is in EXP. □

Then our goal, i.e., the proof of Theorem 6.5, is obtained from Lemma 6.8 and 6.9.

7. Concluding Remarks

We have studied the meaning of several structural properties. For each basic structural properties, we have investigated how it expresses the difficulty of a problem (or a class of problems) which is known to be intractable. In this process, we gained a fair amount of insight into the nature of these properties. In this concluding section, we discuss the remaining open problems in our approach. Also we propose two new structural properties which seem to open up new research area in structural complexity theory.

*　　　　　*　　　　　*

Through our investigation of structural properties on *provably* intractable complexity classes (mainly on EXP), we could make their meaning more clear and obtain many witnesses to our intuition concerning these properties. Since the original motivation of introducing these structural properties is to show how (and why) difficult the class NP is, the results of our study are expected to use for this original purpose. We should notice, however, that the class EXP certainly differs from NP though EXP and NP seem to contain almost the same sets (i.e., NP $\neq$ EXP is provable!). Hence, our results on EXP may not hold on NP, or at least the different proofs are necessary. Then, what is to be considered in order to prove similar results on NP? Here we consider this problem a little more.

The differences between NP and EXP are the following two points: (i) NP is a nondeterministic complexity class while EXP is a deterministic one and (ii) the time bound for NP is polynomial while that for EXP is superpolynomial. These are essential differences to apply our results to NP since in our proofs for EXP, we often used the "deterministic" or "superpolynomial" features of EXP. Regarding the difference (i), we sometimes investigated nondeterministic complexity classes such as NEXP and obtained results similar to the ones for EXP. In general, however, the proof for nondeterministic complexity classes is technically much more difficult than that for deterministic ones. This is because a diagonalization which can be used on nondeterministic classes (i.e., "translational diagonalization" [SFM78]) is weaker than an ordinary one. Thus, we can obtain only partial answers in many cases. The difference (ii) is more serious. Almost all of our proofs shown in this dissertation use the "super-polynomial" feature. And for the theorem whose proof essentially needs the super-polynomialness, we have, so far, no ways to prove it on NP. We need a new proof technique to solve this type of problems.

*   *   *

In this dissertation, we studied basic structural properties. There are several other properties which express important concepts. Among them we review two new structural properties which may open up new research area in computational complexity theory.

The first is the *randomness* of successful guessing sequences in nondeterministic computation. We conjecture that P $\neq$ NP. One intuitive reasoning is as follows: consider some $\leq^{\mathrm{p}}_{\mathrm{m}}$-complete set, say SAT, and any nondeterministic acceptor *M* for SAT; then SAT $\notin$ P because and only because right guessing sequences of *M* on each input are so random that we need to search exhaustively to find an accepting path. That is, the "randomness" of accepting paths characterizes the difficulty of NP-computation. In order to estimate "randomness", we can use the concept of generalized Kolmogorov complexity [Ha83, Ko86]. This type of investigation is interesting in the sense that it directly indicates the advantage of nondeterministic computation to deterministic one. In other words, we cannot discuss this property on any deterministic complexity classes however intractable they are.

The next structural property concerns the difficulty of an instance. Schöning and Orponen [Or86] introduced a new framework, *instance complexity,* in order to measure the difficulty of a particular instance with respect to a given decision problem. In computational complexity theory the difficulty of an instance is usually ignored since for any instance and any program, obvious modification of the program yields a program which runs fast on this particular instance. Namely, we cannot measure the difficulty of a given instance in a usual way. Instance complexity considers the size of a program addition to its running time. Intuitively, an instance x is considered to be *hard* for a problem A if the size of every program that solves A and runs "fast" on x is more than that of x itself (which means that every such program needs to look up (a description of) x in a table in order to run "fast" on x). In the recent study [KOSW86] we show that all problems not in P have infinitely many polynomially hard instances.

References

[Ad78] L. Adleman, Two theorems on random polynomial-time, in "Proc. 19th Ann. Symposium on Foundations of Computer Science", Institute of Electrical and Electronics Engineers, New York (1978) 75-83.

[Al84] E. Allender, personal communication (1984).

[Al85] Eric Allender, "Invertible Functions", Ph.D. Dissertation, Georgia Institute of Technology (1985).

[BBS86] J. Balcázar, R. Book and U. Schöning, Sparse sets, lowness and highness, SIAM J. Computing 15 (1986) 739-747.

[BC86] J. Balcázar and J. Cabbaró, Some comments about notations of orders of magnitude, Bulletin of the EATCS 30 (1986) 34-42.

[BS85] J. Balcázar and U. Schöning, Bi-immune sets for complexity classes, Math. Systems Theory 18 (1985) 1-10.

[Be76] L. Berman, On the structure of complete sets, in "Proc. 17th Ann. Symposium on Foundations of Computer Science", Institute of Electrical and Electronics Engineers, New York (1976) 76-80.

[Be77] L. Berman, "Polynomial Reducibilities and Complete Sets", Ph.D. Dissertation, Cornell University (1977).

[BH77] L. Berman and J. Hartmanis, On isomorphisms and density of NP and other complete sets, SIAM J. Computing 1 (1977) 305-322.

[BK86] R. Book and K. Ko, On sets reducible to sparse sets, manuscript (1986)

[BORW86]R. Book, D. Russo, P. Orponen and O. Watanabe, On exponential lowness, in "Proc. 13th International Colloquium on Automata, Languages and Programming", Lecture Notes in Computer Science

226, Springer-Verlag, Berlin (1986) 40-49.

[Co71] S. Cook, The complexity of theorem-proving procedures, in "Proc. 3rd Ann. ACM Symp. on Theory of Computing", Association for Computing Machinery, New York (1971) 151-158.

[Fo79] S. Fortune, A note on sparse complete sets, SIAM J. on Computing 8 (1979) 431-433.

[GJ79] M. Garey and D. Johnson, "Computers and Intractability, A Guide to the Theory of NP-Completeness", Freeman, San Francisco (1979).

[GS84] J. Grollmann and A. Selman, Complexity measures for public-key cryptosystems, in "Proc. 25th Ann. Symposium on Foundations of Computer Science", Institute of Electrical and Electronics Engineers, New York (1984) 495-503.

[Ha78] J. Hartmanis, "Feasible Computations and Provable Complexity Properties", SIAM, Philadelphia (1978).

[Ha83] J. Hartmanis, Generalized Kolmogorov complexity and the structure of feasible computations, in "Proc. 24th Ann. Symposium on Foundations of Computer Science", Institute of Electrical and Electronics Engineers, New York (1984) 439-445.

[HLY86] J. Hartmanis, M. Li and Y. Yesha, Containment, separation, complete sets, and immunity of complexity classes, in "Proc. 13th International Colloquium on Automata, Languages and Programming", Lecture Notes in Computer Science 226, Springer-Verlag, Berlin (1986) 136-145.

[He84] H. Heller, On relativized polynomial and exponential computations, SIAM J. Computing 13 (1984) 717-725.

[HU79] J. Hopcroft and J. Ullman, "Introduction to Automata Theory, Language, and Computation", Addison-Wesley, Reading (1979).

[JY86] D. Joseph and P. Young, Some remarks on witness functions for nonpolynomial and noncomplete sets in NP, Theoretical Computer Science 39 (1985) 225-237.

[Ka72] R. Karp, Reducibility among combinatorial problems, in R. Miller and J. Thatcher (eds.), "Complexity of Computer Computations", Plenum Press, New York (1972) 85-103.

[KL80] R. Karp and R. Lipton, Some connections between non-uniform and uniform complexity classes, in "Proc. 12th Ann. ACM Symp. on Theory of Computing", Association for Computing Machinery, New York (1980) 302-309.

[Ko85] K. Ko, On some natural complete operators, Theoretical Computer Science 37 (1985) 1-30.

[Ko86] K. Ko, On the notion of infinite pseudorandom sequence, to appear in Theoretical Computer Science.

[Kob84] K. Kobayashi, "Keisan Kanousei Nyuumon" (in Japanese), Kindai-Kagaku-Sha, Tokyo (1984).

[KLD85] K. Ko, T. Long and D. Du, A note on one-way functions and polynomial time isomorphisms, in "Proc. 18th Ann. ACM Symp. on Theory of Computing", Association for Computing Machinery, New York (1986) 295-303.

[KM81] K. Ko and D. Moore, Completeness, approximation and density, SIAM J. Computing 10 (1981) 787-796.

[KOSW86]K. Ko, P. Orponen, U. Schöning and O. Watanabe, What is a hard instance of a computational problem?, in "Proc. 1st Structure in Complexity Theory Conference", Lecture Notes in Computer Science 223, Springer-Verlag, Berlin (1986) 197-217.

[KS85] K. Ko and U. Schöning, On circuit-size complexity and the low hierarchy in NP, SIAM J. Computing 14 (1985) 41-51.

[KSOW86]K. Ko, U. Schöning, P. Orponen and O. Watanabe, What is a hard instance of a computational problem?, in "Proc. 1st Structure in Complexity Theory Conference", Lecture Notes in Computer Science 223, Springer-Verlag, Berlin (1986) 197-217.

[KT83] K. Kobayashi and M. Takahashi, "Ohtomaton no Riron" (in Japanese), Kyoritsu-Shuppan, Tokyo (1983).

[KMR86] S. Kurtz, S. Mahaney and J. Royer, Collapsing degrees, in "Proc. 26th Ann. Symposium on Foundations of Computer Science", Institute of Electrical and Electronics Engineers, New York (1986) 380-389.

[La75] R. Ladner, On the structure of polynomial time reducibility, J. Association for Computing Machinery 22 (1975) 155-171.

[LLS76] R. Ladner, N. Lynch and A. Selman, A comparison of polynomial time reducibilities, Theoretical Computer Science 1 (1975) 103-123.

[Lo82] T. Long, Strong nondeterministic polynomial-time reducibilities, Theoretical Computer Science 21 (1982) 1-25.

[Ma81] S. Mahaney, On the number of p-isomorphism classes of NP-complete sets, in "Proc. 22nd Ann. Symposium on Foundations of Computer Science", Institute of Electrical and Electronics Engineers, New York (1981) 130-143.

[Ma82] S. Mahaney, Sparse complete sets for NP: solution of a conjecture of Berman and Hartmanis, J. Computer and System Sciences 25

(1982) 130-143.

[Ma86] S. Mahaney, Sparse and reducibility, in R. Book ed., "Studies in Complexity Theory", Pitman, London (1986) 63-118.

[MP79] A.R. Meyer and M.S. Paterson, With what frequency are apparently intractable problems difficult?, Technical Report, Massachusetts Institute of Technology, Cambridge TM-126 (1979).

[Or86] P. Orponen, "The Structure of Polynomial Complexity Cores", Ph.D. Dissertation, University of Helsinki (1986).

[OS86] P. Orponen and U. Schöning, The density of polynomial complexity cores for intractable sets, Information and Control 70 (1986) 54-68.

[Pi79] N. Pippenger, On simultaneous resource bounds, in "Proc. 19th Ann. Symposium on Foundations of Computer Science", Institute of Electrical and Electronics Engineers, New York (1979) 307-311.

[Ro67] H. Rogers, Jr., "Theory of Recursive Functions and Effective Computability", McGraw-Hill, New York (1967).

[Sc83] U. Schöning, A low and high hierarchy within NP, J. Computer and System Sciences 27 (1983) 14-28.

[Sc84] U. Schöning, On small generators, Theoretical Computer Science 34 (1984) 337-341.

[Sc86a] U. Schöning, "Complexity and Structure", Lecture Notes in Computer Science 211, Springer-Verlag, Berlin (1985).

[Sc86b] U. Schöning, Complete sets and closeness to complexity classes, Math. Systems Theory 19 (1986) 29-41.

[Sc87] U. Schöning, Graph isomorphism in the low hierarchy, in "Proc. 4th Symp. on Theoretical Aspects of Computer Science", Lecture

Notes in Computer Science, Springer-Verlag, Berlin (1987) to appear.

[SFM78] J. Seiferas, M. Fisher and A. Meyer, Separating nondeterministic time complexity classes, J. Association of Computing Machinery 25 (1978) 146-167.

[Se82] A. Selman, Reductions on NP and p-selective sets, Theoretical Computer Science 19 (1982) 287-304.

[Se83] A. Selman, More than you ever wanted to know about polynomial time reducibilities, in "Proc. Conference on Computational Complexity Theory at Univ. of California, Santa Barbara (1983)".

[St77] L. Stockmeyer, The polynomial time hierarchy, Theoretical Computer Science 3 (1977) 1-22.

[Va76] L. Valiant, Relative complexity of checking and evaluating, Information Processing Letters 5 (1) (1976) 20-23.

[Wa85a] O. Watanabe, On one-one polynomial time equivalence relations, Theoretical Computer Science 38 (1985) 157-165.

[Wa85b] O. Watanabe, Some properties of polynomial time truth-table complete sets, Research Report of Dept. of Information Sciences, Tokyo Institute of Technology, Tokyo C-66 (1985).

[Wa86a] O. Watanabe, A comparison of polynomial time completeness notions, Theoretical Computer Science, to appear.

[Wa86b] O. Watanabe, On hard one-way functions, Research Report of Dept. of Information Sciences, Tokyo Institute of Technology, Tokyo C-70 (1985); revised in 1986.

[Wa86c] O. Watanabe, Some observations of k-creative sets, Research Report of Dept. of Information Sciences, Tokyo Institute of

Technology, Tokyo C-79 (1986);

[Wr77] C. Wrathall, Complete sets and the polynomial-time hierarchy, Theoretical Computer Science (1976) 23-33.

[Ya82] A. Yao, Theory and applications of trapdoor functions, in "Proc. 23rd Ann. Symposium on Foundations of Computer Science", Institute of Electrical and Electronics Engineers, New York (1982) 80-91.

[Ye83] Y. Yesha, On certain polynomial-time truth-table reductions to sparse sets, SIAM J. Computing 12 (1983) 580-587.