Chapter 6

Hartmanis Conjectures

In this lecture, we discuss Berman-Hartmanis isomorphism conjecture.

6.1 Introduction

The Berman-Hartmanis conjecture, also known as the *isomorphism conjecture* is related to NP vs. P question. We start with a definition. Two languages A and B are said to be p-isomorphic (polynomial time isomorphic), if there is a polynomial computable, polynomial time invertible, 1-1 and onto reduction from A to B. Formally, we need a function $f : \Sigma^* \mapsto \Sigma^*$, such that i) f is 1-1 and onto, ii) $x \in A \iff f(x) \in B$, iii) f polynomial time computable, iv) f^{-1} is polynomial time computable. Berman and Hartmanis observed that all "known" NP-Complete problems are p-isomorphic to each other. Here, "known" refers to, for example, all problems in Garey and Johnson. We will get into to the issue of proving this claim, shortly. Based on this claim, they formulated the now-famous conjecture:

Berman-Hartmanis isomorphism conjecture: Any two NP-Complete sets are p-isomorphic to each other.

The most interesting aspect of the conjecture is that, if it is true then NP $\neq$ P. Because, if NP = P, then, even finite sets would be NP-Complete. But, a finite set cannot be isomorphic to a infinite set like SAT. The conjecture has been studied extensively in the past two decades. If the conjecture is true, then not even sparse sets can be NP-Complete, because SAT is an "exponentially" dense set and there cannot be a polynomial time isomorphism from a dense set to a sparse set. This raises an interesting issue of whether sparse sets can be NP-Complete. Mahaney's theorem (which we proved in an earlier lecture) shows that no sparse set can be NP-Complete, unless NP=P. The isomorphism conjecture remains open. Some recent evidence shows that it may be false. We will discuss this evidence at the end of the lecture.

We now return to the claim made by Berman and Hartmanis: all "known" NP-Complete problems are p-isomorphic to each other. The claim is based on their following theorem. Suppose there are polynomial time computable, polynomial time invertible, 1-1 and length increasing reductions from A to B and B to A. Then, A and B are p-isomorphic to each other. (We will state the theorem more formally later.) They observe that such reductions exist for all "known" NP-Complete problems. Proof of the above theorem is based on a theorem due to Cantor. We prove Cantor's theorem first, and then continue our discussion of the isomorphism conjecture.

6.2 A Theorem of Cantor

THEOREM 6.1 *If A and B are sets such that there exists a $1-1$ map from A to B and a $1-1$ map from B to A, then there exists a $1-1$ correspondence (i.e., a map that is both $1-1$ and onto) between A and B*

Proof.

Let f be a $1-1$ map from A to B, and g be a $1-1$ map from B to A.

If either f or g is also onto, then it is a $1-1$ correspondence. We will henceforth assume that neither f nor g is onto and prove that there still exists a $1-1$ correspondence.

Consider the following two sequences of subsets of A and B: $A_0 = A$, $B_0 = B$, and the rest of the A_i and B_i are given by:

$$A_i = g(B_{i-1}) \qquad B_i = f(A_{i-1})$$

These two sequences are shown pictorially in figure 6.1.

The first thing to note about this sequence is that $A_i \subset A_{i-1}$ (i.e., A_i is a proper subset of A_{i-1}) for all $i \in \mathbf{N}$, and the same is true for the B_i. To see this, observe first that it is clear from the definitions of A_i and B_i that $A_i \subseteq A_{i-1}$ and $B_i \subseteq B_{i-1}$. We will show (by induction on i) that successive sets are not equal.

Observe that $A_1 = g(B_0)$. Since g is not onto, there exists an element in A_1 that is not in A_0. Similarly, $B_1 \subset B_0$.

Now, assuming that $B_i \subset B_{i-1}$, let x be any element of $B_{i-1} - B_i$. Note first that $g(x) \in A_i$, since $A_i = g(B_{i-1})$. We claim that $g(x) \notin A_{i+1}$. If $g(x) \in A_{i+1}$, then $\exists y \in B_i$ such that $g(y) = g(x)$. Since $x \notin B_i$, this would imply that two distinct elements (x and y) of B are mapped to the same element of A, contradicting the stipulation that g is $1-1$. This completes the induction and proves that $A_i \subset A_{i-1}$ for all $i \in \mathbf{N}$. A similar argument can be made for the B_i.

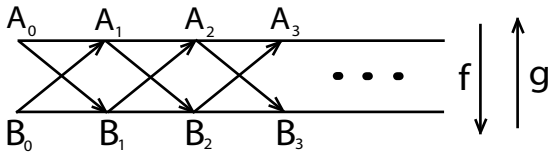

Figure 6.1: A pictorial representation of the sequences created by applying the functions $f(x)$ and $g(x)$. The A_i's are listed by cardinality with the largest on the left. The same is true of the B_i's.

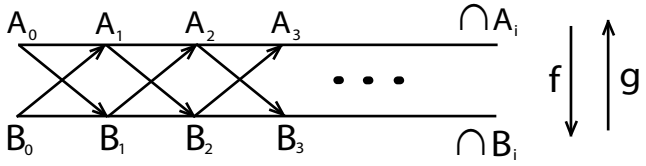

Figure 6.2: The pictorial description from 6.1 modified to include the sets $\bigcap_{i=0}^{\infty} A_i$ and $\bigcap_{i=0}^{\infty} B_i$. Again, the largest sets are on the left of the image while the smallest (i.e. $\bigcap_{i=0}^{\infty} A_i$ and $\bigcap_{i=0}^{\infty} B_i$) are on the right.

Now, with these facts in hand, we note that the set A can be decomposed as follows:

$$A = (A_0 - A_1) \cup (A_1 - A_2) \cup \cdots \cup (A_0 \cap A_1 \cap A_2 \cap \cdots)$$

$$= \left(\bigcup_{i=0}^{\infty} (A_i - A_{i+1}) \right) \cup \left(\bigcap_{i=0}^{\infty} A_i \right)$$

Similarly,

$$B = \left(\bigcup_{i=0}^{\infty} (B_i - B_{i+1}) \right) \cup \left(\bigcap_{i=0}^{\infty} B_i \right)$$

Thus, for any element $x \in A$, either $x \in (A_i - A_{i+1})$ for some (unique) i, or $x \in A_i$ for all i (see figure 6.2). We'll define a function $F : A \to B$, which will be a $1-1$ correspondence, as follows: if $x \in \bigcap_{i=0}^{\infty} A_i$, then $F(x) = f(x)$. Otherwise, $x \in (A_i - A_{i+1})$ for some i. In this case,

$$F(x) = \begin{cases} f(x) & \text{if } i \text{ is even} \\ g^{-1}(x) & \text{if } i \text{ is odd} \end{cases}$$

That is, an element x of A is mapped to B by either f or g^{-1}, depending on which subset(s) of A it falls into. Note that for all $i > 0$, A_i contains only elements that are in the image of g, so $g^{-1}(x)$ is well-defined in all cases where we use it. It remains for us to show that F is $1-1$ and onto.

We start with $1-1$. We first consider an element $x \in \bigcap_{i=0}^{\infty} A_i$ and argue that no other element of A maps to $F(x)$ under F. Here we have $F(x) = f(x)$. Moreover, note that $f(x) \in \bigcap_{i=0}^{\infty} B_i$. Consider any element $x' \in A$ with $x' \neq x$. If $x' \in \bigcap_{i=0}^{\infty} A_i$, $F(x') = f(x')$ and as f is 1-1, $F(x) \neq F(x')$. If $x' \notin \bigcap_{i=0}^{\infty} A_i$, then x' is mapped to some element in $B_i - B_{i-1}$, for some i. Hence, such an x' cannot be mapped to an element in $\bigcap_{i=0}^{\infty} B_i$. So, $F(x) \neq F(x')$. Next consider an element $x \in (A_i - A_{i+1})$ for some i. We have two cases: if i is even, then $f(x) \in (B_{i+1} - B_{i+2})$ – i.e., the largest j for which $f(x) \in B_j$ is odd; if i is odd, then $g^{-1}(x) \in B_{i-1} - B_i$ – i.e., the largest j for which $g^{-1}(x) \in B_j$ is even. Thus, an element of A mapped by f never collides with an element mapped by g^{-1}. Since f and g are both $1-1$, elements mapped by f never collide with each other, and likewise for g^{-1}. Thus, F is $1-1$.

It may be obvious at this point that F is also onto, but we will prove it for good measure. Let y be an element of B. Suppose $y \in \bigcap_{i=0}^{\infty} B_i$. Clearly, $y \in B_1$ and hence, for some $x \in A$, $f(x) = y$. For any i, notice that $y \in B_i = f(A_{i-1})$ and f is 1-1 and hence, $x \in A_{i-1}$. Thus, $x \in \bigcap_{i=0}^{\infty} A_i$, and hence, $F(x) = f(x) = y$. Suppose, $y \in B_i - B_{i+1}$ for some i, then we again look at two cases: if i is even, then y is mapped to by $g(y)$; if i is odd, then i is mapped to by $f^{-1}(y)$, which is guaranteed to exist, since every odd B_i is the image under f of some A_j. Thus, F is $1-1$ and onto.

6.3 Myhill's Theorem

One interesting theorem that can be proven using a similar proof is Myhill's theorem.

THEOREM 6.2 *Every* RE*-complete set has a 1-1 onto recursive, invertible map to every other* RE*-complete set.*

This theorem shows that essentially, there is only "one" RE-complete set. In other words, any two undecidable, r.e. sets are just computable renaming of each other. We will not prove the theorem in this class.

6.4 The Berman-Hartmanis Conjecture

Berman and Hartmanis conjecture that all NP -Complete problems are p-isomorphic to each other. The conjecture is based on the following theorem.

THEOREM 6.3 *Let L_1 and L_2 be two sets such that there are functions $f : \Sigma^* \mapsto \Sigma^*$ and $g : \Sigma^* \mapsto \Sigma^*$ that satisfy*

1. *f and g are polynomial time computable.*

2. *f and g are 1-1.*

3. *f and g are length increasing. That is, for any x, $|f(x)| > |x|$ and $|g(x)| > |x|$.*

4. *f is a reduction from L_1 to L_2 and g is a reduction from L_2 to L_1.*

5. *f and g are polynomial time invertible. That is, there is a polynomial time algorithm that given x either outputs the (unique) y such that $f(y) = x$ or says no such y exists. Similarly, such an algorithm exists to compute g^{-1}.*

Then, L_1 and L_2 are p-isomorphic. That is, there is a polynomial time computable, polynomial time invertible reduction F from L_1 to L_2 which is 1-1 and onto.

Proof. We follow the proof of Cantor's theorem. Let $A_0 = B_0 = \Sigma^*$. Then, as in Cantor's proof, define

$$A_i = g(B_{i-1}) \text{ and } B_i = f(A_{i-1})$$

First of all notice that, a string x of length n cannot be in A_{n+1}, because both f and g are (strictly) length increasing. Thus, $\bigcap_{i=0}^{\infty} A_i$ and $\bigcap_{i=0}^{\infty} B_i$ are empty. Let x be any string and i be the (unique) integer such that $x \in A_i - A_{i+1}$. Because, $\bigcap_{i=0}^{\infty} A_i$ is empty, i is well-defined. Now let,

$$F(x) = \begin{cases} f(x) & \text{if } i \text{ is even} \\ g^{-1}(x) & \text{if } i \text{ is odd} \end{cases}$$

That completes the definition of F. We need to argue that F satisfies the required properties:

1. *F is 1-1 and onto:* The proof is same as Cantor's proof.

2. *F is a reduction from L_1 to L_2 :* For any x, $F(x)$ is either $f(x)$ or $g^{-1}(x)$. The claim follows from the fact that f is a reduction from L_1 to L_2 and g is a reduction from L_2 to L_1.

3. *F is polynomial time computable :* Given an x, all we need to do is, determine whether we are in the even or odd case. After that, we need to either compute $f(x)$ or $g^{-1}(x)$ and both these can be done in polynomial time. Notice that $x \in A_1$ iff $g^{-1}(x)$ exists. Similarly, $x \in A_2$ iff $f^{-1}(g^{-1}(x))$ exists and $x \in A_3$ iff $g^{-1}(f^{-1}(g^{-1}(x)))$ exists. In general, to determine whether $x \in A_k$, we need simply apply g^{-1} and f^{-1} alternatively k times and check whether that element exists or not. Now our goal is to find the unique d such that $x \in A_d - A_{d+1}$. We already noted that $x \notin A_n$, where $n = |x|$. Hence, $d \leq n$. Recall our assumption that f^{-1} and g^{-1} are polynomial time computable. So, the idea is to start with x and apply g^{-1} and f^{-1} alternatively: compute

   $$x_0 = x, x_1 = g^{-1}(x), x_2 = f^{-1}(x_1), x_3 = g^{-1}(x_2), x_4 = f^{-1}(x) \ldots$$

   Notice this, the strings in this sequence get shorter as we move on (i.e. $|x_{i+1}| < |x_i|$), because f and g are length increasing. After some steps, we will have to stop, becuase

either f^{-1} (or g^{-1}) does not exist or we end up with a string of length 1. Suppose we computed till x_i succesfully and x_{i+1} does not exist. Then, $x \in A_i$ but $x \notin A_{i+1}$, (i.e. $d = i$). As f^{-1} and g^{-1} are computable each step of this process can be carried out in polynomial time. As $d \leq n$, we will need at most n steps.

Using this theorem, Berman and Hartmanis observed that any two "known" NP-Complete sets are p-isomorphic to each other. Because, for any "known" NP-Complete problem, the NP-Completeness reduction can easily be modified to make it 1-1, length increasing and polynomial time invertibe! As examples, we will consider SAT and Hamiltonican circuit.

Let L be any lnaguage in NP and f be the reduction from L to SAT. We will construct a new reduction f' from L to SAT which is length increasing and 1-1. Let $x = a_1 a_2 \ldots a_n \in \{0,1\}^*$ be the input string of length n. Reduction f' first computes $\varphi = f(x)$. Then construct a new formula φ'. φ' uses all the boolean variables of φ and n new variables $z_1, z_2, \ldots, z_n$. New formula $\varphi' = \varphi \wedge (l_1 \vee l_2 \vee \ldots \vee l_n)$, where the lieteral l_i is z_i, if $a_i = 1$ and $\overline{z_i}$ if $a_i = 0$. It is clear that φ is satisfiable iff φ' is satisfiable and hence, f' is a reduction from L to SAT. Moreover, $|\varphi'| > |x|$. Finally, given φ', we can easily extract the string x. Thus, f' is polynomial time invertible.

A similar trick can be used in case of Hamiltonican circuit (HC). Let L be any language in NP and f be a reduction from L to HC. We construct a new reduction f'. Given input $x = a_1 a_2 \ldots a_n$, f' first computes the graph $G = f(x)$. f' will output a new graph G'. G' is obtained by first adding $3n+1$ new vertices $s_1, s_2, \ldots, s_{n+1}$, $u_1, u_2, \ldots, u_n$, and $v_1, v_2, \ldots, v_n$ to G. For each $1 \leq i \leq n$, if $a_i = 0$ then add the edges $s_i \rightarrow u_i \rightarrow v_i \rightarrow s_{i+1}$; if $a_i = 0$ then add the edges $s_i \rightarrow v_i \rightarrow u_i \rightarrow s_{i+1}$. Thus, we have created some "chain" of vertices, that depend on input x. Finally, take any edge (s,t) in G and "replace" it with the chain constructed from x. Namely, remove the egde (s,t) and add edges (x, s_1) and (s_{n+1}, t). It is clear that G has a Hamiltonican circuit iff G' has one. Moreover, $|G'| > |x|$ and given G', we can extract x easily.

For all "known" NP-Complete problems, say those listed in Garey and Johnson, we can do such simple tricks to make the reduction length increasing and 1-1. Then, the above theorem implies that all known NP-Complete problems are p-isomorphic to each other. Based on this evidence, Berman and Hartmanis conjectured that all NP-Complete problems are p-isomorphic to each other. But, now a days, it is widely believed to be fasle. The Joseph-Young conjecture gives some evidence for this belief.

6.5 Joseph-Young Conjecture

The Joseph and Young conjecture states that there is an NP-Complete languages A such that there is no polynomial time invertible reduction from SAT to A. Such a language would fail to satisfy the requirements of Theorem 6.3. The language A is constructed based on the assumption that one-way functions exist. Loosely speaking, a function is one-way, if it

is computable in polynomial time, but not invertible in polynomial time. It is not known whether one-way functions exist (existence of one-way functions imply NP $\neq$ P). But, there are some candidate functions believed to be one-way. One example is multiplication. Given two numbers a and b, it is easy to compute their product ab. But the inverse of multiplication, namely, factoring is believed to be hard. Suppose we have a one-way function f. It is easy to show that for any 1-1 function g, the language g(SAT) is NP-Complete. In particular, $f(SAT)$ is NP-Complete. Now, the conjecture is that there is no invertible reduction from SAT to $f(SAT)$.