On Polynomial-Time Relation Reducibility

Su Gao and Caleb Ziegler

Abstract We study the notion of polynomial-time relation reducibility among computable equivalence relations. We identify some benchmark equivalence relations and show that the reducibility hierarchy has a rich structure. Specifically, we embed the partial order of all polynomial-time computable sets into the polynomial-time relation reducibility hierarchy between two benchmark equivalence relations $E_A$ and $id$. In addition, we consider equivalence relations with finitely many non-trivial equivalence classes and those whose equivalence classes are all finite.

1 Introduction

It is well known that the relative complexity of binary relations can be studied by two different notions of reducibility. First, a binary relation $R$ with domain $X$ is a subset of $X \times X$. Thus reducibility between relations is a special case of reducibility between sets on product spaces. In this sense, if $R$ is a binary relation on $X$ and $S$ a binary relation on $Y$, then a reduction function $f$ from $R$ to $S$ is a function from $X \times X$ to $Y \times Y$ such that for all $x_1, x_2 \in X$, $\langle x_1, x_2 \rangle \in R \iff f(x_1, x_2) \in S$.

A stronger notion of reducibility can also be defined. For the abovementioned $R$ and $S$, a strong reduction function $f$ is a function from $X$ to $Y$ such that, for all $x_1, x_2 \in X$, $\langle x_1, x_2 \rangle \in R \iff \langle f(x_1), f(x_2) \rangle \in S$.

To distinguish between the two notions of reducibility, we call the latter notion the relation reducibility, since it only applies to relations.

The relation reducibility has been studied intensively in at least two different contexts. The best known example is the Borel reducibility between analytic quasi-orders and analytic equivalence relations on Polish spaces. Another instance is the computable reducibility between computably enumerable equivalence relations on
natural numbers. In this article, we study the relation reducibility for equivalence relations, where the domains are subsets of \( \Sigma^* \), which is the set of all finite sequences over a certain finite alphabet \( \Sigma \), and the strong reduction function is polynomial-time computable. We denote this reducibility notion by \( \leq_R \).

This reducibility was defined in [4] by Fortnow and Grochow and was called kernel reduction. The more recent preprint [5] adopted this terminology and studied NP-equivalence relations and the issue of completeness. The same notion was called strong equivalence reduction in [3], in which the authors considered this reducibility primarily restricted to classes of finite structures under the isomorphism relation (which they called strong isomorphism reduction). It was shown in [3] that the partial order \( \leq_K \), particularly restricted to isomorphism relations that are polynomial-time computable, has a rich structure. Specifically, they showed that the partial order on the countable atomless Boolean algebra is embeddable into the partial order \( \leq_R \) on isomorphism relations that are in P.

In this article, we obtain some more structural results for \( \leq_R \). We also focus on computable equivalence relations, even P-equivalence relations and NP-equivalence relations, although some of our results are general. Among P-equivalence relations, we consider some natural benchmark equivalence relations \( \text{id}, E_\lambda \) and \( E_\sigma \) (defined in Section 2). It turns out that the embeddability result from [3] is about isomorphism relations below \( E_\lambda \). We show in this article that the partial order \( \leq_R \) between \( \text{id} \) and \( E_\lambda \) also has a rich structure. Our main embeddability result is the following.

**Theorem 1.1** There is an assignment \( X \mapsto E_X \) from the collection of all subsets of natural numbers \( X \in \mathbb{P} \) to equivalence relations between \( E_\lambda \) and \( \text{id} \) such that, for any subsets of natural numbers \( X, Y \in \mathbb{P} \),

\[
X \subseteq Y \iff E_X \leq_R E_Y.
\]

Some reducibility or non-recudibility results can be obtained under assumptions on the relationship among complexity classes. We will identify several other benchmark equivalence relations and investigate their mutual reducibility with assumptions such as \( \text{P} = \text{NP} \) or \( \text{P} \neq \text{NP} \).

We also consider equivalence relations with finitely many non-trivial equivalence classes and those whose equivalence classes are all finite.

Some of our stated results are easy to prove and can be found in the literature (especially [3] and [5]), but we include them for the completeness and coherence of our exposition.

## 2 Preliminaries

Throughout the article, we fix a finite alphabet \( \Sigma \) with at least two elements 0 and 1. Let \( \Sigma^* \) be the set of all finite sequences over the alphabet \( \Sigma \). For \( x \in \Sigma^* \), we denote by \( |x| \) the length of \( x \). All equivalence relations we consider will have some subsets of \( \Sigma \) as their domains.

We use \( \mathbb{P} \) to denote the class of all subsets of \( \Sigma^* \) that are polynomial-time computable by a deterministic Turing machine, \( \text{NP} \) to denote the class of all subsets of \( \Sigma^* \) that are polynomial-time acceptable by a non-deterministic Turing machine, and \( \text{PF} \) to denote the class of all polynomial-time computable functions from \( \Sigma^* \) to \( \Sigma^* \).

In addition, \( \text{coNP} \) denotes the class of all subsets of \( \Sigma^* \) whose complement is in \( \text{NP} \), and \( \text{DP} \) denotes the class of all sets of the form \( L_1 \cap L_2 \) where \( L_1 \in \text{NP} \) and
$L_2 \in \text{coNP}$. Obviously, 
\[ P \subseteq \text{NP} \cap \text{coNP} \subseteq \text{NP} \cup \text{coNP} \subseteq \text{DP}. \]

We use $\mathbb{N}$ to denote the set of all natural numbers $\{0, 1, 2, \ldots\}$. Via a fixed coding that we will not specify, we view $\mathbb{N}$ as a subset of $\Sigma^*$. In particular, we will speak of polynomial-time computable functions from $\mathbb{N}$ to $\mathbb{N}$ and subsets of $\mathbb{N}$ that are in various complexity classes.

We define again the main notion of reducibility under investigation.

**Definition 2.1** For binary relations $R$ and $S$, we say that $R$ is **polynomial-time relation reducible** to $S$, denoted $R \leq_R S$, if there is a function $f : \text{dom}(R) \rightarrow \text{dom}(S)$ such that $f \in \text{PF}$ and for all $x, y \in \text{dom}(R)$, $(x, y) \in R \iff (f(x), f(y)) \in S$.

If $R \leq_R S$ but $S \not\leq_R R$, then we denote $R <_R S$ and say that $R$ is **strictly polynomial-time relation reducible** to $S$. If $R \leq_S S$ and $S \leq_R R$, then we write $R \equiv_R S$ and say that $R$ and $S$ are **polynomial-time relation bi-reducible** with each other.

It is easy to see that $\leq_R$ is a quasi-order (i.e., reflexive and transitive) and $\equiv_R$ is an equivalence relation.

We will be primarily concerned with the structure of $\leq_R$ on the class of all computable equivalence relations. We introduce a few canonical examples of computable equivalence relations, which will serve as benchmarks for the hierarchy under $\leq_R$.

**Definition 2.2**

1. We denote by $\text{id}$ the **identity relation**: for $x, y \in \Sigma^*$, 
   \[ (x, y) \in \text{id} \iff x = y. \]

2. We denote by $E_\lambda$ the **equality of length relation**: for $x, y \in \Sigma^*$, 
   \[ (x, y) \in E_\lambda \iff |x| = |y|. \]

3. Let $l_n$ be the super-exponential sequence of natural numbers defined by $l_0 = 0$ and $l_{n+1} = 2^{l_n}$. For $x, y \in \Sigma^*$, define
   \[ (x, y) \in E_\sigma \iff \text{if } l_k \leq |x| < l_{k+1} \text{ and } l_m \leq |y| < l_{m+1}, \text{ then } k = m. \]

We will show below that $E_\sigma <_R E_\lambda <_R \text{id}$. Before doing this, we prove a general lemma that allows us to obtain non-reducibility results from counting. The concept involved is a generalization of the notion of potential reducibility in [3]. All of our non-reducibility results that do not require assumptions on relationships between complexity classes will be proved by the counting method used in the proof of the lemma.

**Lemma 2.3** Let $E, F$ be equivalence relations on $\Sigma^*$ and let $\#E(n), \#F(n)$ denote the number of equivalence classes containing elements of length at most $n$. If $\#E(n) \neq O(\#F(p(n)))$ for any polynomial $p(n)$, then $E \not\leq_R F$.

**Proof** Assume $E \leq_R F$ via a strong reduction function $f \in \text{PF}$. Let $p(n)$ be a polynomial bound for the running time of $f$, i.e., for any input $x \in \Sigma^*$ with $|x| = n$, it takes no more than $p(n)$ steps for the computation of $f(x)$ to halt. Assume without loss of generality that $p(n)$ is monotone increasing. It follows that if $|x| \leq n$, then $|f(x)| \leq p(n)$. Now fix any length $n$, let $\#E(n) = k$ and let $x_1, \ldots, x_k \in \Sigma^*$ be pairwise non-$E$-equivalent elements of length at most $n$. Then $f(x_1), \ldots, f(x_k)$ are pairwise non-$F$-equivalent since $f$ is a strong reduction function. We conclude that $\#F(p(n)) \geq k$, or $\#E(n) \leq \#F(p(n))$. \qed
Proposition 2.4  \( E_\sigma <_R E_\lambda <_R \text{id} \).

Proof  \( E_\lambda <_R \text{id} \) is easy to see: just define the strong reduction function \( f : \Sigma^* \to \Sigma^* \) by \( f(x) = 0^{|x|} \). To see that \( E_\sigma \leq_R E_\lambda \), define a strong reduction function \( f : \Sigma^* \to \Sigma^* \) by \( f(x) = 0^k \) where \( k \) is the largest natural number with \( l_k \leq |x| \).

To prove the non-reducibility, we apply Lemma 2.3. It is easy to see that the lemma applies since \( \#(n) \geq 2^n \), \( \#E_\lambda(n) = n \), and \( \#E_\sigma < \log n \). 

By replacing the super-exponential sequence \( l_n \) by “sparser” sequences, we can modify the definition of \( E_\sigma \) to obtain an infinite descending chain of equivalence relations below \( E_\sigma \). The proof of non-reducibility is built on the same counting argument as in the above proof. We state this result without giving the details.

Proposition 2.5  There is an infinite sequence \( E_n \) of equivalence relations such that \( E_{n+1} <_R E_n <_R E_\sigma \) for all \( n \).

Proposition 2.5 is not new since it has been shown in [3] that the partial order \( \leq_R \) below \( E_\lambda \) contains a copy of the countable atomless Boolean algebra. However, the equivalence relations constructed here are different from those in constructed in [3].

3 Structural results between \( E_\lambda \) and \( \text{id} \)

In this section, we show that the reducibility \( \leq_R \) between \( E_\lambda \) and \( \text{id} \) already has a rich structure. Among other things, we will demonstrate an infinite ascending chain and an infinite antichain between \( E_\lambda \) and \( \text{id} \). We will prove the main embeddability result mentioned in Section 1, which in particular implies that any finite partial order can be embedded into the relation \( <_R \) among equivalence relations between \( E_\lambda \) and \( \text{id} \). To achieve these, we construct a class of equivalence relations as follows.

Definition 3.1  
(a) We call a set \( A \subseteq \mathbb{N} \) good if \( A \) is infinite, \( A \in P \), and, letting \( \{a_n : n \geq 1\} \) enumerate the elements of \( A \) in strictly increasing order, the function \( a_n \mapsto n \) is polynomial-time computable.

(b) For a good set \( A \subseteq \mathbb{N} \), let \( \{a_n : n \geq 1\} \) enumerate all elements of \( A \) in strictly increasing order. Define \( L_A : \Sigma^* \to \mathbb{N} \) by

\[
L_A(x) = \begin{cases} 
0 & \text{if } |x| < a_1 \\
|y| & \text{if } n \geq 1 \text{ is such that } a_n \leq |x| < a_{n+1}.
\end{cases}
\]

Note that \( L_A \) is polynomial-time computable (see Lemma 3.2 below).

(c) Let \( A \subseteq \mathbb{N} \) be good and let \( \{a_n : n \geq 1\} \) enumerate all elements of \( A \) in strictly increasing order. Let \( \varphi : \mathbb{N} \to \mathbb{N} \) be a polynomial-time computable function with \( \varphi(n) \leq a_n \) for all \( n \geq 1 \). Define an equivalence relation \( E_{\varphi,A} \) by

\[
\langle x, y \rangle \in E_{\varphi,A} \iff L_A(x) = L_A(y) \text{ and, if } L_A(x) > 0, x \text{ and } y \text{ agree in the first } \varphi(L_A(x)) \text{ bits.}
\]

Lemma 3.2  If \( A \subseteq \mathbb{N} \) is good, then \( L_A \) is polynomial-time computable.

Proof  Let \( A \) be good and let \( \{a_n : n \geq 1\} \) enumerate all elements of \( A \) in strictly increasing order. Then there is a polynomial-time computable function \( \varphi \) recognising \( A \), i.e. \( \varphi(x) = 1 \) if \( x \in A \) and \( \varphi(x) = 0 \) otherwise, and a polynomial-time computable function \( \psi : a_n \mapsto n \). Note that the values that \( \psi \) takes on elements not in \( A \) are not relevant. We compute \( L_A \) as follows. Given a string \( x \), we compute \( |x| \). We then run \( \varphi(m) \) for \( m = |x|, |x| - 1, \ldots, 0 \) until we find the largest \( m \) with \( \varphi(m) = 1 \). This
process requires at most $|x|$ applications of $\varphi$ on strings of length at most $|x|$, so can be done in time that is polynomial in $|x|$. If $\varphi(m) = 1$, then $m = a_n$ for some $n \geq 1$.

Since we run this computation in decreasing order, we see that if $m$ is the largest with \(\varphi(m) = 1\), then $m = a_n$ where $|x| < a_{n+1}$.

So suppose $m$ is the result of this computation. Then, $L_A(x) = \varphi(m)$ for this $m$. If this computation does not return any $m$ with $\varphi(m) = 1$, then $|x| < a_1$, and so in this case, $L_A(x) = 0$.

It follows easily from the definition that $E_{\varphi,A} \leq_R \text{id}$.

As the following theorem demonstrates, the relative complexity of $E_{\varphi,A}$ is closely related to the growth rates of the functions $\varphi$ and $n \mapsto a_n$, which enumerates elements of the set $A$.

**Theorem 3.3** Let $A \subseteq \mathbb{N}$ be good and let $\{a_n : n \geq 1\}$ enumerate all elements of $A$ in strictly increasing order. Let $\varphi$ and $\psi$ be polynomial-time computable functions from $\mathbb{N}$ to $\mathbb{N}$ with $\varphi(n), \psi(n) \leq a_n$ for all $n \geq 1$. Then the following hold:

(i) If $\varphi(n) \leq \psi(n)$ for all $n$, then $E_{\varphi,A} \leq_R E_{\psi,A}$.

(ii) If $\varphi$ is increasing, $\varphi(n) = \Omega(\log a_n)$ and for any polynomial $p(n)$,

$$
\psi(n) \neq O(\varphi(p(a_n))),
$$

then $E_{\psi,A} \leq_R E_{\varphi,A}$.

(iii) Let $B \subseteq \mathbb{N}$ be good and let $\{b_n : n \geq 1\}$ enumerate all elements of $B$ in strictly increasing order. If $a_n \leq b_n$ for all $n \geq 1$, then $E_{\varphi,B} \leq_R E_{\varphi,A}$.

**Proof** For (i), suppose $\varphi(n) \leq \psi(n)$ for all $n$. We define a strong reduction function $f : \Sigma^* \rightarrow \Sigma^*$ from $E_{\varphi,A}$ to $E_{\psi,A}$. Let $x \in \Sigma^*$. If $|x| < a_1$ then $L_A(x) = 0$ and we define $f(x)$ to be the empty sequence. Otherwise, $L_A(x) > 0$ and $\varphi(L_A(x)) \leq a_{L_A(x)} \leq |x|$. In this case, define $f(x)$ to be the sequence of the same length as $x$ which agrees with $x$ on the first $\varphi(L_A(x))$ bits and is 0 for all remaining bits. We check that $f$ works. If $x, y \in \Sigma^*$ are $E_{\varphi,A}$-equivalent, then we actually have $f(x) = f(y)$ and therefore $f(x)$ and $f(y)$ are $E_{\psi,A}$-equivalent. If $x$ and $y$ are not $E_{\varphi,A}$-equivalent, then either $L_A(x) \neq L_A(y)$ or else $L_A(x) = L_A(y) > 0$ but $x$ and $y$ disagree in the first $\varphi(L_A(x))$ bits. If $L_A(x) \neq L_A(y)$, then $L_A(f(x)) \neq L_A(f(y))$ and so $f(x)$ and $f(y)$ are not $E_{\psi,A}$-equivalent. If $L_A(x) = L_A(y) > 0$ and $x$ and $y$ disagree in the first $\varphi(L_A(x))$ bits, then $f(x)$ and $f(y)$ also disagree in the first $\varphi(L_A(f(x))) = \varphi(L_A(x))$ bits, which implies that $f(x)$ and $f(y)$ are not $E_{\psi,A}$-equivalent.

For (ii), we give a direct proof instead of applying Lemma 2.3; this turns out to be notationally simpler but we use the same counting technique. Assume $E_{\psi,A} \leq_R E_{\varphi,A}$ and let $f$ be a strong reduction function from $E_{\psi,A}$ to $E_{\varphi,A}$. Let $n \geq 1$ be arbitrarily fixed. Let $N_\varphi(n)$ be the number of $E_{\varphi,A}$-equivalence classes which consist of strings $x$ with $L_A(x) = n$. It is easily seen by direct counting that $N_\varphi(n) = 2^{|\varphi(n)|}$. Similarly, define $N_\psi(n) = 2^{|\psi(n)|}$. Now $f$ is a one-one function on the equivalence classes of $E_{\psi,A}$, thus the $N_\varphi(n)$ many distinct $E_{\varphi,A}$-equivalence classes are mapped via $f$ to $N_\psi(n)$ many distinct $E_{\psi,A}$-equivalence classes. Let $S(n)$ be the set of all $x \in \Sigma^*$ with $|x| = a_n$. Then for each $x \in S(n)$, $L_A(x) = n$, and for any $y \in \Sigma^*$ with $L_A(y) = n$, there is $x \in S(n)$ which is $E_{\varphi,A}$-equivalent to $y$ (since $\psi(n) \leq a_n$). In other words, each of the $N_\varphi(n)$ many distinct $E_{\varphi,A}$-equivalence classes contains an element of $S(n)$.

Let $p(n)$ be a polynomial bound for the running time of $f$, i.e., the computation of $f$ on an input string of length $n$ takes no more than $p(n)$ many steps to halt. Then we must have $|f(x)| \leq p(|x|)$ for any $x \in \Sigma^*$. Without loss of generality, assume that
\( p(n) \geq n \) and is monotone increasing. Considering the \( E_{\varphi,A} \)-equivalence classes of the elements of \( \{ f(x) : x \in S(n) \} \), we obtain

\[
N_{\varphi}(n) \leq \sum_{k \leq p(a_n)} N_{\varphi}(k) \leq p(a_n) N_{\varphi}(p(a_n)).
\]

The second inequality follows from the hypothesis that \( \varphi \) is increasing.

Taking logarithm on all sides of the inequality, we yield

\[
\psi(n) \leq \varphi(p(a_n)) + O(\log a_n).
\]

Since \( \varphi(n) = \Omega(\log a_n) \), we have \( \psi(n) = O(\varphi(p(a_n))) \), a contradiction.

For (iii), suppose \( a_n \leq b_n \) for all \( n \geq 1 \). Note that for any \( x \in \Sigma^* \), \( L_B(x) \leq L_A(x) \) and, if \( L_B(x) > 0 \), then \( a_{L_B(x)} \leq a_{L_A(x)} \leq |x| \). Define

\[
f(x) = \begin{cases} 
\text{the empty sequence} & \text{if } L_B(x) = 0 \\
x | a_{L_B(x)} & \text{otherwise.}
\end{cases}
\]

Then \( f \) is polynomial-time computable. In fact, given \( x \), the computations of \( L_A(x) \) and \( L_B(x) \) take time that is polynomial in \( |x| \); then we obtain \( x | a_{L_B(x)} \) by checking on all the candidates, which are successive truncations of \( x \). (Note that we do not need to assume that the function \( n \mapsto a_n \) is polynomial-time computable.)

We verify that \( f \) is a strong reduction. First, suppose \( x \) and \( y \) are \( E_{\varphi,B} \)-equivalent. Then we have \( L_B(x) = L_B(y) \). If they are zero, then \( f(x) = f(y) \) and is the empty sequence, and in particular \( f(x) \) and \( f(y) \) are \( E_{\varphi,A} \)-equivalent. Suppose \( L_B(x) = L_B(y) > 0 \). Then \( x \) and \( y \) must agree on the first \( \varphi(L_B(x)) \) bits. By definition, \( f(x) \) and \( f(y) \) are truncations to the length \( a_{L_B(x)} \) of \( x \) and \( y \), respectively. We have \( L_A(f(x)) = L_B(x) \) and \( L_A(f(y)) = L_B(y) \). Therefore, \( L_A(f(x)) = L_A(f(y)) \) and we have that \( f(x) \) and \( f(y) \) still agree on the first \( \varphi(L_A(f(x))) = \varphi(L_B(x)) \) bits, so \( f(x) \) and \( f(y) \) are \( E_{\varphi,A} \)-equivalent.

For the other direction, suppose \( x \) and \( y \) are not \( E_{\varphi,A} \)-equivalent. Then either \( L_B(x) \neq L_B(y) \) or else \( L_B(x) = L_B(y) > 0 \) but \( x \) and \( y \) do not agree on the first \( \varphi(L_B(x)) \) bits. Since \( L_A(f(x)) = L_B(x) \) and \( L_A(f(y)) = L_B(y) \), we have \( L_A(f(x)) \neq L_A(f(y)) \) if \( L_B(x) \neq L_B(y) \). If \( L_B(x) = L_B(y) \) but \( x \) and \( y \) do not agree on the first \( \varphi(L_B(x)) \) bits, then \( L_A(f(x)) = L_A(f(y)) > 0 \) but \( f(x) \) and \( f(y) \) do not agree on the first \( \varphi(L_A(f(x))) \) bits, since \( f(x) \) and \( f(y) \) are just truncations of \( x \) and \( y \) to the length \( a_{L_B(x)} \), and \( \varphi(L_A(f(x))) \leq \varphi(L_B(x)) \leq a_{L_B(x)} \). In either case, we see that \( f(x) \) and \( f(y) \) are not \( E_{\varphi,A} \)-equivalent. This shows that \( f \) is a strong reduction.

In the remainder of this section we give some applications of Theorem 3.3. In fact, for these applications we will fix a good set \( A = \mathbb{N} \setminus \{0\} \). Thus \( a_n = n \) in the enumeration of elements of \( A \), and we have \( L_A(x) = |x| \). For notational simplicity, let \( E_{\varphi} \) denote \( E_{\varphi,A} \) for this particular \( A \).

In summary, let \( \varphi : \mathbb{N} \to \mathbb{N} \) be an increasing, polynomial-time computable function with \( \varphi(n) \leq n \) for all \( n \in \mathbb{N} \), and \( E_{\varphi} \) be the equivalence relation defined by

\[
(x, y) \in E_{\varphi} \iff |x| = |y| \text{ and } x \text{ and } y \text{ agree in the first } \varphi(|x|) \text{ bits}.
\]

Note that \( E_\lambda \leq_R E_{\varphi} \leq_R \text{id} \). The first reduction is witnessed by the same strong reduction function \( f : \Sigma^* \to \Sigma^* \), where \( f(x) = 0^{|x|} \), that we used in the proof of Proposition 2.5.

We are now ready to construct an infinite ascending chain and an infinite antichain between \( E_\lambda \) and \( \text{id} \).
Corollary 3.4 There is an infinite sequence of functions $\varphi_m \in \mathbb{P}$ such that $E_{\varphi_m} \not< R E_{\varphi_m+1}$ for all $m$.

Proof For each $m \geq 1$, define $\varphi_m(n) = \min\{n, (\log n)^m\}$. Then each $\varphi_m$ is polynomial-time computable and $\log n \leq \varphi_m(n) \leq n$ for all $n$. Since $\varphi_m(n) \leq \varphi_{m+1}(n)$ for all $n$, we have $E_{\varphi_m} \leq R E_{\varphi_{m+1}}$ by Theorem 3.3 (i). On the other hand, for any polynomial $p(n)$, $\varphi_m(p(n)) = \Theta((\log n)^m)$. Since $(\log n)^{m+1} \neq O((\log n)^m)$, we have that $\varphi_{m+1}(n) \neq O(\varphi(p(n)))$. Thus $E_{\varphi_{m+1}} \not\leq R E_{\varphi_m}$ by Theorem 3.3 (ii). We thus have $E_{\varphi_m} < R E_{\varphi_{m+1}}$ for all $m$.

Again, it is easily seen that the construction in the above proof can be modified to obtain longer chains. For instance, if we use functions such as $\log k$, etc., one can obtain more infinite ascending sequences in between the ones obtained in the above proof. As a consequence, one can embed the linear order on the ordinal $\omega^k$ for any finite $k$ into the relation $< R$ between $E_{\lambda}$ and id.

Corollary 3.5 There is an infinite sequence of functions $\varphi_m \in \mathbb{P}$ such that $E_{\varphi_m} \not\leq R E_{\varphi_{m'}}$ for any $m \neq m'$.

Proof Let $(\cdot, \cdot) : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ be a polynomial-time computable pairing function such that the decoding functions $(\cdot)_1$ and $(\cdot)_2$ are both polynomial-time computable. For instance, the usual Cantor diagonalization function is itself a polynomial in both variables and therefore is such a function.

Define an increasing sequence of natural numbers $N_k$ by induction.

$$
N_0 = 1,
N_{k+1} = 2^{1+(\log N_k)^2}.
$$

We are now ready to define $\varphi_m$ for any $m \in \mathbb{N}$ as follows:

$$
\varphi_m(n) = \begin{cases} 
\min\{n, (\log n)^2\} & \text{if } N_k \leq n < N_{k+1} \text{ and } (k)_1 = m, \\
\min\{n, (\log N_k)^2\} & \text{if } N_k \leq n < N_{k+1} \text{ and } (k)_1 \neq m.
\end{cases}
$$

Each $\varphi_m$ is polynomial-time computable. It is easy to see that each $\varphi_m$ is increasing. To see that $\varphi_m(n) = \Omega(\log n)$, it is enough to check that, as functions in $k$, $\log N_{k+1} = O(\varphi_m(N_k))$. For this, just notice that, asymptotically, $\varphi_m(N_k) = (\log N_k)^2$ whereas $\log N_{k+1} = 1 + (\log N_k)^2$.

We claim that, if $m \neq m'$, then $\varphi_m(n) \neq O(\varphi_{m'}(n))$. This would complete the proof by Theorem 3.3 (ii).

Suppose not and assume $\varphi_m(n) = O(\varphi_{m'}(n))$. Then for some constant $C > 0$ and $N$, $\varphi_m(n) \leq C \varphi_{m'}(n)$ for all $n \geq N$. Suppose $N$ is large enough that $(\log n)^2 \leq n$ for all $n \geq N$. It follows in particular that for all $k$ such that $N_k \geq N$,

$$
\varphi_m(N_k+1-1) \leq C \varphi_{m'}(N_k+1-1).
$$

If in addition $(k)_1 = m$, then by our definition of $\varphi_m$ we have

$$
\varphi_m(N_k+1-1) = (\log(N_k+1-1))^2 \leq C(\log N_k)^2.
$$

This implies that, as functions in $k$, $\log(N_k+1-1) = O(\log N_k)$. This is clearly impossible since there are infinitely many $k$ with $(k)_1 = m$ and because

$$
\log(N_k+1-1) = \Theta(\log N_k+1) = \Theta(1 + (\log N_k)^2) = \Theta((\log N_k)^2) \neq O(\log N_k).
$$
S. Gao and C. Ziegler

Note that the equivalence relations constructed in the above proofs are all between $E_\lambda$ and the second equivalence relation constructed in the proof of Corollary 3.4. Again the technique can be easily adapted for other pairs of equivalence relations obtained from the proof of Corollary 3.4.

The functions we constructed in the proof of Corollary 3.5 have a stronger property that we now explore.

**Corollary 3.6** There is an assignment $X \mapsto E_X$ from the collection of all subsets of natural numbers $X \in \mathcal{P}$ to equivalence relations between $E_\lambda$ and $\text{id}$ such that, for any subsets of natural numbers $X, Y \in \mathcal{P}$,

$$X \subseteq Y \iff E_X \leq_R E_Y.$$  

**Proof** Let $\varphi_m$ be the functions constructed in the proof of Corollary 3.5. For each polynomial-time computable $X \subseteq \mathbb{N}$, define

$$\varphi_X(n) = \max\{\varphi_m(n) : m \in X\}$$

and $E_X = E_{\varphi_X}$. Each $\varphi_X$ is polynomial-time computable because

$$\varphi_X(n) = \begin{cases} \min\{n, (\log n)^2\} & \text{if } N_k \leq n < N_{k+1} \text{ and } (k)_1 \in X, \\ \min\{n, (\log N_k)^2\} & \text{if } N_k \leq n < N_{k+1} \text{ and } (k)_1 \notin X. \end{cases}$$

Now if $X \subseteq Y \subseteq \mathbb{N}$ and $X, Y \in \mathcal{P}$, then $\varphi_X(n) \leq \varphi_Y(n)$ for all $n$, and therefore $E_X \leq_R E_Y$ by Theorem 3.3 (i). On the other hand, if $X \not\subseteq Y$, and let $m \in X \setminus Y$, then $E_{\varphi_m} \not\leq_R E_Y$ by the same proof of non-reducibility as in that of Corollary 3.5. □

**Corollary 3.7** Any finite partial order can be embedded into the relation $<_R$ among equivalence relations between $E_\lambda$ and $\text{id}$.

**Proof** Corollary 3.6 obviously implies that any finite Boolean algebra is embeddable into the $<_R$ relation among equivalence relations between $E_\lambda$ and $\text{id}$. This corollary follows since any finite partial order can be embedded into a finite Boolean algebra. □

### 4 More benchmarks and non-reducibility results

An important class of computable equivalence relations is that of isomorphism for finite structures. For each kind of finite structures, such as graphs, groups, linear orders, etc., we fix a canonical coding of the structures by sequences in $\Sigma^*$. Then the isomorphism relation corresponds to an equivalence relation with a computable subset of $\Sigma^*$ as its domain. We adopt the following notation.

**Definition 4.1**

1. We denote by $\text{GI}$ the graph isomorphism relation: if $x,y \in \Sigma^*$ encode finite graphs, then $(x,y) \in \text{GI}$ if there is an isomorphism between the graphs coded respectively by $x$ and $y$.
2. Similarly, we denote by $\text{GROUP}$ the isomorphism for finite groups and $\text{BOOL}$ the isomorphism for finite Boolean algebras.
3. We denote by $\text{CLIQ}$ the clique relation: if $x,y \in \Sigma^*$ encode finite graphs, then $(x,y) \in \text{CLIQ}$ if there is an isomorphism between the graphs coded respectively by $x$ and $y$.

All isomorphism relations for finite structures are NP-equivalence relations. Thus $\text{GI}, \text{GROUP}$ and $\text{BOOL}$ are in NP. $\text{CLIQ}$, on the other hand, is DP-complete as a set [1].
Proposition 4.2  
The following are true:

(i) If $E$ is the isomorphism relation for finite structures of any finite language, then $E \leq_R \text{GI}$.

(ii) If $E$ is the isomorphism relation for any of the following classes of finite structures, then $E \equiv_R \text{id}$:
   (iia) finite trees;
   (iib) finite planar graphs;
   (iic) finite linear orderings with a unary relation.

(iii) If $E$ is the isomorphism relation for any of the following classes of finite structures, then $E \equiv_R \text{E}_\lambda$:
   (iiia) finite sets;
   (iiib) finite fields;
   (iiic) finite abelian groups;
   (iid) finite cyclic groups;
   (iie) finite linear orderings;
   (iii) finite linear orderings with a distinguished point.

(iv) If $E$ is the isomorphism relation for any of the following classes of finite structures, then $E \equiv_R \text{GI}$:
   (iva) finite regular graphs;
   (ivb) finite bipartite graphs.

See [3] for (i), (iic) and (iii). The proof of (iia) is folklore and implicit in any proof that finite tree isomorphism is in P. A proof of (iib) is implicit in the proof of the theorem of Hopcroft and Tarjan [6] that the isomorphism for finite planar graphs is in P. The proof of (iva) can be extracted from Miller [7]. We give a proof of (ivb) below.

Proof of Proposition 4.2 (ivb)  
Let $G = (V,E)$ be an arbitrary finite graph. We define a bipartite graph $G^* = (V',E^*)$. The mapping $G \mapsto G^*$ will be a strong reduction from GI to the isomorphism of finite bipartite graphs. Let $V^* = V \cup E$ and $E^* = \{ (v,e), (e,v) : v \in V, e \in E, \text{ and } v \text{ is a vertex of } e \}$. $G^*$ is bipartite because $\{V,E\}$ is a partition of its vertices such that every edge in $G^*$ is between a vertex in $V$ and a vertex in $E$. Suppose $G_1$ and $G_2$ are two finite graphs. It is clear that, if $G_1 \cong G_2$, then $G_1^* \cong G_2^*$. In fact, any isomorphism between $G_1$ and $G_2$ induces an isomorphism of $G_1^*$ and $G_2^*$ by the definition of $G^*$. For the converse, first note that $G$ is connected iff $G^*$ is connected. Without loss of generality, we may assume both $G_1$ and $G_2$ are connected. Suppose $\pi$ is an isomorphism from $G_1^*$ to $G_2^*$. Then $\pi$ must map $V_1$ to either $V_2$ or $E_2$, since the two graphs are bipartite. In the first case, it must happen that $\pi(V_1) = V_2$ and $\pi(E_1) = E_2$, and therefore $\pi$ induces an isomorphism between $G_1$ and $G_2$. In the second case, $\pi(V_1) = E_2$ and $\pi(E_1) = V_2$. Note that each vertex in $E_1$ in $G_1^*$ has degree 2, thus it must happen that each vertex in $V_2$ in $G_2^*$ has degree 2, and thus each vertex in $V_1$ in $G_1^*$ has degree 2. This means that all vertices in the two graphs have degree 2, and $G_1^*$ must be a simple cycle. The same holds for $G_2^*$. These imply that both $G_1$ and $G_2$ are simple cycles, and they are of the same length, thus $G_1$ and $G_2$ are isomorphic.  

It was shown in [3] (Corollary 18) that \[ \text{BOOL} \prec_R \text{E}_\lambda \prec_R \text{GROUP} \prec_R \text{GI} \]
id \not\leq_R \text{GROUP}.

The proofs of the non-reducibility directions are all based on the counting argument used to prove Lemma 2.3. The same argument gives also

$$E_\sigma <_R \text{BOOL}$$

since $\#\text{BOOL}(n) = \Theta(\log n)$ and $\#E_\sigma < \log \log n$.

Note that

$$E_\lambda \leq_R \text{CLIQ};$$

this is witnessed by the strong reduction function $x \mapsto f(x)$, where $f(x)$ is the code for the complete graph with $|x|$ vertices. Note that $\#\text{CLIQ}(n) = O(n)$. Thus by Lemma 2.3, we have

$$id \not\leq_R \text{CLIQ}.$$ 

Since $id \leq_R \text{GI}$, it follows that

$$\text{GI} \not\leq_R \text{CLIQ}.$$ 

We summarize the reducibility and non-reducibility results for the seven benchmark equivalence relations in a diagram on the following page. In the diagram, a solid arrow represents reducibility $<_R$, an arrow with a cross represents non-reducibility $\not\leq_R$, and a dotted line indicates that the opposite direction is open.

The following propositions clarify the reducibility from \text{CLIQ} with other equivalence relations in the diagram.

**Proposition 4.3** The following are equivalent:

(i) $P = \text{NP}$;
(ii) $\text{CLIQ} \leq_R E_\lambda$;
(iii) $\text{CLIQ} \equiv_R E_\lambda$.

**Proof** Obviously (ii) and (iii) are equivalent. To see (i) $\Rightarrow$ (ii), assume $P = \text{NP}$. Then there is a polynomial-time algorithm to determine, given a finite graph $G$ and a number $k$, whether $G$ contains a clique of size $k$. Let $G$ have $n$ vertices. Then by setting $k = 2, \ldots, n$, the maximal $k$ such that $G$ contains a clique of size $k$ can be computed in polynomial time. If we run this decision algorithm and output a sequence of length of this maximal $k$, this gives a strong reduction function in $P$ from $\text{CLIQ}$ to $E_\lambda$. For (ii) $\Rightarrow$ (i), note that $\text{NP} \subseteq \text{DP}$ and that $\text{CLIQ}$ is $\text{DP}$-complete as a set. If $\text{CLIQ} \leq_R E_\lambda$, then $\text{CLIQ} \in \text{P}$ and so is every $\text{DP}$ set.

Thus if $P = \text{NP}$, then $\text{CLIQ}$ is of the same complexity as $E_\lambda$ and not a separate benchmark.
Proposition 4.4  If \( P \neq NP \) then \( CLIQ \not\leq_R id \).

Proof  This is because if \( CLIQ \leq_R id \), then \( CLIQ \in P \).

In other words, if \( P \neq NP \), then \( CLIQ \) is incomparable with \( id \) and \( E_\lambda <_R CLIQ \).

Proposition 4.5  If \( NP \neq coNP \), then \( CLIQ \not\leq_R GI \), and in particular \( CLIQ \not\leq_R id \) and \( CLIQ \not\leq_R GROUP \).

Proof  If \( CLIQ \leq_R GI \), then \( CLIQ \in NP \). Since \( coNP \subseteq DP \), it would follow that \( NP = coNP = DP \).

In other words, if \( NP \neq coNP \), then \( CLIQ \) is incomparable with either \( GI \) or \( id \), and \( E_\lambda <_R CLIQ \).

The following lemma was essentially proved in [2] (c.f. also Lemma 32 of [3]).

Lemma 4.6 (Blass–Gurevich)  If \( P = NP \) then \( E \leq_R id \) for every \( E \in NP \).

Thus, if \( P = NP \), then \( GI \equiv_R id \) and therefore \( GROUP <_R id \).
5 Finitary equivalence relations

In this section, we note that there is a canonical initial segment of the $\leq_R$ hierarchy for P equivalence relations. We also show that if $P \neq NP$, then there are NP equivalence relations strictly above id. These results are obtained by considering a special class of equivalence relations.

**Definition 5.1** An equivalence relation $E$ is called finitary if $E$ has only finitely many non-trivial equivalence classes, i.e., all but finitely many $E$-equivalence classes are singletons.

The following proposition is easy to see. We state it without proof.

**Proposition 5.2** Let $E$ be a finitary equivalence relation on $\Sigma^*$. Then the following are equivalent:

(i) $E \in P$;
(ii) Each $E$-equivalence class is in $P$;
(iii) $E \leq_R id$.

We will consider two subclasses of finitary equivalence relations. The first consists of those with only finitely many equivalence classes. The canonical examples of such equivalence relations are the congruence relations on natural numbers. For each positive $n \in N$, denote by $\equiv_n$ the congruence relation mod $n$, i.e.,

$$x \equiv_n y \iff x \equiv y \mod n$$

for $x, y \in N$. Then for each $n$, $(\equiv_n) <_R (\equiv_{n+1}) <_R id$. Moreover, if $E \leq_R (\equiv_n)$ for any $n \in N$, then $E$ is a P-equivalence relation with at most $n$ equivalence classes.

We again state the following easy proposition without proof.

**Proposition 5.3** Let $E$ be an equivalence relation on $\Sigma^*$ with $n$ equivalence classes. Then the following are equivalent:

(i) $E \in P$;
(ii) There is an equivalence relation $F \in P$ with infinitely many equivalence classes such that $E \leq_R F$;
(iii) For any equivalence relation $F$ with infinitely many equivalence classes, $E \leq_R F$;
(iv) $E \equiv_R (\equiv_n)$.

Thus the P equivalence relations with finitely many equivalence classes form an infinite ascending chain that is an initial segment of the $\leq_R$ hierarchy.

Next, we consider equivalence relations induced by a single set.

**Definition 5.4** For any subset $S$ of $\Sigma^*$, we define an equivalence relation $R_S$ on $\Sigma^*$ by

$$\langle x, y \rangle \in R_S \iff \text{either } x = y \text{ or both } x \in S \text{ and } y \in S.$$ 

Equivalence relations of the form $R_S$ are of course finitary. About their mutual reducibility, we have the following observation.

**Lemma 5.5** Let $S, T \subseteq \Sigma^*$. If $R_S \leq_R R_T$, then either $R_S \leq_R id$ or $S$ is polynomial-time reducible to $T$ as sets.
On Polynomial-Time Relation Reducibility

**Proof** Suppose $R_S \leq_R R_T$ via a strong reduction function $f \in PF$. Then for any $x, y \in S$, $(f(x), f(y)) \in R_T$. We have two cases. Case 1: for any $x \in S$, $f(x) \not\in T$. In this case, $f(x) = f(y)$ for any $x, y \in S$. Then, $f$ witnesses that $R_S \leq_R id$. Case 2: for any $x \in S$, $f(x) \in T$. Note that for any $x \not\in S$, $f(x) \not\in T$. Thus, $f$ is a reduction function from $S$ to $T$.

Note that $id$ is itself of the form $R_S$, with $S = \emptyset$. Also, note that there is a co-infinite $S \subseteq \Sigma^*$ with $R_S \not\leq_R id$. An example is

$$S = \{x \in \Sigma^* : x(i) \neq 0 \text{ for some } i < |x|\}.$$ 

It is not hard to see that $R_S \equiv_R E^*_\lambda^\ast$.

**Proposition 5.6** Let $S \subseteq \Sigma^*$ be nonempty. Then the following hold:

(i) $S \in NP$ iff $R_S \in NP$.

(ii) If $S$ is NP-hard then $R_S$ is NP-hard as a set.

(iii) If $R_T \leq_R R_S$ for all $T \subseteq \Sigma^*$ with $T \in NP$, then $S$ is NP-hard.

(iv) If $P \neq NP$ and $S$ is NP-hard, then $R_S \not\leq_R id$.

**Proof** For (i), the implication ($\Rightarrow$) follows from the definition. For ($\Leftarrow$) of (i) and for (ii), fix an element $a \in S$. Then for any $x \in \Sigma^*$, $x \in S$ iff $(x, a) \in R_S$. The function $x \mapsto (x, a)$ is a polynomial-time computable function reducing $S$ to $R_S$ as sets.

Now (iv) follows immediately from (ii). To prove (iii), assume $R_T \leq_R R_S$ for all NP subseteq $T \subseteq \Sigma^*$. In particular, $id \leq_R R_S$, and it follows that $S$ is co-infinite. We argue for (iii) in two cases. If $P = NP$ then any nonempty, proper subset of $\Sigma^*$ is NP-hard. Suppose $P \neq NP$. Let $T$ be NP-hard. Then $R_T \not\leq_R id$ by (iv). By Lemma 5.5 we must have a polynomial-time reduction from $T$ to $S$, and hence $S$ is NP-hard.

Now it is easy to define NP-complete $S \subseteq \Sigma^*$ for which $id \not\leq_R R_S$. If $P \neq NP$ then $id \leq_R R_S$ for such an $S$.

The following notion is in some sense a dual to the notion of finitary equivalence relations.

**Definition 5.7** An equivalence relation $E$ is called finite if every $E$-equivalence class is finite.

The equivalence relations $E_{\varphi, A}$ we defined earlier in this paper and the natural isomorphism relations for finite structures are all finite equivalence relations. In particular $GI$ is finite.

The following proposition shows that the two notions are indeed orthogonal in terms of reducibility.

**Proposition 5.8** Let $E$ and $F$ be equivalence relations on $\Sigma^*$. Suppose $E$ is finitary and $F$ is finite. Then the following hold:

(a) If $E \leq_R F$, then $E \leq_R id$, and in particular $E \in P$.

(b) If $F \leq_R E$, then $F \leq_R id$, and in particular $F \in P$.

**Proof** To prove (a), we only need to verify that every $E$-equivalence class is in $P$ in view of Proposition 5.2. But any strong reduction function $f \in PF$ witnesses the reduction of any $E$-equivalence class to an $F$-equivalence class, and the latter, being finite, is in $P$.

For (b), assume that $f \in PF$ is a strong reduction function from $F$ to $E$. Let $X = \{x \in \Sigma^* : \text{ the $E$-equivalence class of } x \text{ is trivial}\}$. Then $f^{-1}(X)$ consists of only
We define a class of finite equivalence relations induced from a single set.

**Definition 5.9** For any subset $S$ of $\Sigma^*$, we define an equivalence relation $D_S$ on $\Sigma^*$ by

$$(x, y) \in D_S \iff \text{either } x = y \text{ or } x \upharpoonright (|x| - 1) = y \upharpoonright (|y| - 1) \in S.$$ 

$D_S$ is a finite equivalence relation since $\Sigma$ is finite. Note that $x \mapsto x \downarrow 0$ is a strong reduction function from $id$ to $D_S$. Thus $id \leq_R D_S$ for all $S \subseteq \Sigma^*$.

**Proposition 5.10** Let $S \subseteq \Sigma^*$. The following are equivalent:

(i) $D_S \in \mathbb{P}$;

(ii) $S \in \mathbb{P}$;

(iii) $D_S \leq_R id$;

(iv) $D_S \equiv_R id$.

**Proof** The implications (ii)$\Rightarrow$(i)$\Leftrightarrow$(iii) are obvious, and (iii)$\Leftrightarrow$(iv) follows from the above remark. It suffices to show (i)$\Rightarrow$(ii). Then for any $x \in \Sigma^*$, $x \in S$ iff $(x \downarrow 0, x \downarrow 1) \in D_S$. Thus if $D_S \in \mathbb{P}$ then $S \in \mathbb{P}$.

The above proof also gives the following proposition as an immediate corollary, which we state without proof.

**Proposition 5.11** Let $S \subseteq \Sigma^*$. Then the following hold:

(i) $S \in \mathbb{NP}$ iff $D_S \in \mathbb{NP}$.

(ii) If $S$ is NP-hard then $D_S$ is NP-hard as a set.

(iii) If $P \neq \mathbb{NP}$ and $S$ is NP-hard, then $id <_R D_S$.

Thus if $P \neq \mathbb{NP}$ and $S$ is NP-complete, then both $R_S$ and $D_S$ are NP-equivalence relations, but they are incomparable in the $\leq_R$ partial order.

### 6 Open Problems

Results from [3] and this paper show that it is possible to prove reducibility and non-reducibility results about $\leq_R$ without assumptions on the relationship between complexity classes. Let us call such results *absolute*. The following open problems might still have absolute answers.

**Problem 6.1** Does $E \leq_R id$ for all P-equivalence relations $E$?

**Problem 6.2** Does $E \leq_R GI$ for all finite NP equivalence relations $E$?

### References


**Acknowledgments**

The first author acknowledges the US NSF grants DMS-1201290 for the support of his research. The second author acknowledges the US NSF grant DMS-0943870 for the support of his research. We would like to thank Yijia Chen for useful discussions on the topic of this paper.

Gao
Department of Mathematics
University of North Texas
1155 Union Circle #311430
Denton, Texas 76203 USA
sgao@unt.edu
http://www.math.unt.edu/~sgao/

Ziegler
Department of Mathematics
University of North Texas
1155 Union Circle #311430 Denton, Texas 76203
USA
caleb.ziegler@my.unt.edu