"Natural Proofs Vs Derandomization"

Akash Kumar

1 PRELIMINARY

This report is based on the paper "Natural Proofs Vs Derandomization" by Ryan Williams. (STOC, 2013). I've tried to give details about the purpose of the paper, the underlying technique to solve the key results of the paper.

2 RESULTS

Razborov-Rudich gave Natural proofs. For the past two decades, complexity theorists have tried to give lower bounds which have been naturalizing i.e. a Natural Proof exists. It has been a point of study, if algorithms can be given avoiding the conditions of a Natural Proofs:

• Constructivity
• Largeness
• Usefullness

In this paper, Ryan gives connection between :

• NEXP lower bounds are constructive and useful : P-natural property useful against a class of typical circuits $\mathcal{C}$ containment of $\text{NEXP} \not\subseteq \mathcal{C}$ and thus proving that Constructivity is necessary.

• Natural Property and Derandomization : P-natural property useful against $\mathcal{C}$ and derandomization of probabilistic exponential time using $\mathcal{C}$ random seeds.
3 Some Propositions and Lemmas

**Typical Class**: A non-uniform class C is called typical if it belongs to the set 
\[ \mathcal{C} = \{ \text{AC}^0, \text{ACC}, \text{TC}^0, \text{NC}^1, \text{NC}, \text{P}/\text{poly} \} \]

3.1 Some Conventions

For any circuit \( C(x_1, x_2, \ldots, x_n) \), \( i < j \) and \( a_1, a_2, \ldots, a_n \in \{0, 1\} \), \( C(a_1, a_2, \ldots, a_j, \ldots, a_n) \) is a circuit with all but integers in \((i,j)\), are fixed with constants \( a_i \)'s. Here, \( . \) represents free assignment of the variables.

For a boolean function \( f : \{0, 1\}^n \to \{0, 1\} \), truth table of \( f \) is defined as \( \text{tt}(f) = f(y_1) f(y_2) \ldots f(y_n) \) where \( y_i \) is the \( i \)th string of length \( n \) in lexicographical order.

For string \( T \) of length \( n \) which is not power of 2, the following encoding rule is being used: Let \( C \) be \( s \)-size \( d \)-depth circuit for \( f \), \( m \) is the \( \text{ith} \) minimum integer such that, \( 2^m \leq n \). Let \( y_i \) be the \( i \)th string of length \( n \) in lexicographical order.

For string \( T \) of length \( n \) which is not power of 2, the following encoding rule is being used: Let \( C \) be \( s \)-size \( d \)-depth circuit for \( f \), \( m \) is the \( \text{ith} \) minimum integer such that, \( 2^m \leq n \). Let \( y_i \) be the \( i \)th string of length \( n \) in lexicographical order.

Circuit complexity of a string \( T \) is denoted by \( \text{CC}(T) \). The convention remains same if we consider a \( d \)-depth circuit complexity of \( T \), i.e. \( \text{CC}_d(T) \). For, MOD gates with modulus \( m \) and depth bound \( d \), \( d \)-depth mod-\( m \) complexity of \( T \) is denoted by \( \text{CC}_{d,m}(T) \).

**Proposition-1**: Suppose \( T = T_1 \ldots T_{2^k} \) is a string of length \( 2^{k+1} \), where \( T_i \)'s have length \( 2^i \). Then, \( \text{CC}(T) \leq \text{CC}(T) \), \( \text{CC}_d(T_i) \leq \text{CC}_d(T) \), and \( \text{CC}_{d,m}(T) \leq \text{CC}_{d,m}(T) \).

**Proof**: Note that using the convention for a function \( f_T \) representing \( T \), \( T_i \) is obtained from \( f_T \) by fixing \( k \) variables in \( x_1, x_2, \ldots, x_k \) such that a string of length \( 2^i \), \( T_i \) is obtained using lexicographical ordered strings of length \( k \) after plugging them into \( x_i \)'s. Note that, \( T_1 T_2 \ldots T_{i-1} \) is obtained by first \( i-1 \) input variables. Now, \( T_i \) is obtained by plugging strings into the variables from index \( i-1 \) to \( i+1 \). Thus, fixing the rest of \( k \) variables gives \( T_i \).

Let \( f_T \) be the function encoding \( T \) then \( f_T(x_1, \ldots, x_i) = f(a_1, \ldots, a_{i-1+i}, a_{i+1+i}, \ldots, a_k) \), where \( \forall \ i, \ a_i = 0 \).

This proves that \( T_i \) is obtained from the circuit for \( T \). Hence the inequalities for the circuit sizes follow.

**Lemma-1**: There is a universal \( c \geq 1 \) such that the following holds. Let \( T \) be a binary string, and let \( S \) be a any substring of \( T \). Then for all \( d \) and \( m \), \( \text{CC}(f_S) \leq \text{CC}(f_T) + (c \log |T|) \), \( \text{CC}_d(f_S) \leq \text{CC}_d(f_T) + (c \log |T|)^{1+o(1)} \), and \( \text{CC}_{d,m}(f_S) \leq \text{CC}_{d,m}(f_T) + (c \log |T|)^{1+o(1)} \)

**Proof**: Assume \( c' \) be large enough constant (which can be evaluated later). Let \( k \) be the minimum interger such that \( 2^k \geq |T| \). Let \( f_T \) be a function encoding \( T \) such that \( tt(f_T) = T0^{2^k - |T|} \).

Let \( C \) be \( s \)-size \( d \)-depth circuit for \( f_T \)

Now, \( S \) is a substring such that \( S = t_A \ldots t_B \) where \( A, B \in \{1, 2, \ldots, 2^k\} \) and \( T = t_1 t_2 \ldots t_{2^k} \). Approximate minimum \( l \) such that, \( 2^l \geq B - A \). Now, we construct a circuit \( D \) which computes \( S0^{2^k - |S|} \).

Let \( x_1, x_2, \ldots, x_{2^l} \) be lexicographical ordered strings of length \( l \). Given string \( x_i \), First compute
\[ \log i + \log A \leq \log B \text{ then output } C(y[\log(i+A)]), \text{ where } y[\log(i+A)] = \log x_i + y_A, \text{ where } y_A \text{ is the } A\text{th string in lexicographical ordering of strings of length } k. \]

Now, using [CFL85], we can have a depth-\(c'\) circuits of size at most \(c'n \log^* n\) for addition of two \(n\)-bit strings. Also, the inequality can be checked in \((c' \cdot l)\)-size depth-\(c'\) circuits. So finally using the three circuits for addition, inequality and \(C, D\) can be implemented by a circuit of size at most \(s + c'(k \log^* k) + l + 1\). Now, we can assume that \(c \geq 2c'\) and we are done. \(\square\)

4 NEXP LOWER BOUNDS ARE CONSTRUCTIVE AND USEFUL

**Theorem 1.** For all typical \(C\), NEXP \(\not\subset C\) iff there is a polynomial-time property of Boolean functions useful against \(C\).

4.1 **Definitions**

**Natural Proofs:** A property of Boolean functions \(P\) is a subset of the set of all Boolean functions. Let \(\Gamma\) be a complexity class and let \(C\) be a circuit class (typically, \(\Gamma = \text{P} \text{ and } C = \text{P/poly}\)). A \(\Gamma\)-natural property useful against \(C\) is a property of Boolean functions \(P\) that satisfies the axioms:

- (Constructivity) \(P\) is decidable in \(\Gamma\),
- (Largeness) for all \(n\), \(P\) contains \(1/2^{O(n)}\) fraction of all \(n\)-bit inputs,
- (Usefulness) Let \(f = \{f_n\}\) be a sequence of functions \(\{f_n\}\) such that \(f_n \in P\) for all \(n\). Then for all \(k\) and infinitely many \(n\), \(f_n\) does not have \(n^k\)-size \(C\)-circuits.

Let \(f = \{f_n\} : \{0,1\}^n \rightarrow \{0,1\}\) be a sequence of Boolean functions. A \(\Gamma\)-natural proof that \(f \notin C\) establishes the existence of a \(\Gamma\)-natural property \(P\) useful against \(C\) such that \(P(f_n) = 1\) for all \(n\) \(\square\).

**Witness Class:** Let \(L \in \text{NTIME}[t(n)]\) where \(t(n) \geq n\) is constructible, and let \(C\) be a circuit class. An algorithm \(V(x,y)\) is a predicate for \(L\) if \(V\) runs in time \(O(|y|) + t(|x|)\) and for all strings \(x, x \in L \leftrightarrow\) there is a \(y\) of length \(O(t(n))\) (a witness for \(x\)) such that \(V(x,y)\) accepts. We denote \(L(V)\) to be the language accepted by \(V\). \(V\) has \(C\) witness of size \(s(n)\) if for all strings \(x, x \in L\) then there is a \(C\)-circuit \(C_x\) of size at most \(s(n)\) such that \(V(x,C_x(.))\) accepts. \(L\) has \(C\)'s of polynomial size if for all predicates \(V\) for \(L\), there is a polynomial \(p(n)\) such that \(V\) has \(C\) witness of size \(O(p(n))\). \(\text{NTIME}[t(n)]\) has \(C\) witness if for every infinite language \(L \in \text{NTIME}[t(n)]\), \(L\) has \(C\) witness of polynomial size. \(\square\)

**Oblivious Witness Class:** Let \(L \in \text{NTIME}[t(n)]\), let \(C\) be a circuit class, and \(V\) be a predicate for \(L\). \(L\) has oblivious \(C\) witness of size \(s(n)\) if for every predicate \(V\) for \(L\), there is a \(C\) circuit
family \{C_n\} of size s(n) such that for all \(x \in \{0,1\}^*\), if \(x \in L\) then \(V(x, C|x)(x,.)\) accepts. Also, \(\text{NTIME}[t(n)]\) has oblivious \(C\) witnesses if every infinite \(L \in \text{NTIME}[t(n)]\) has oblivious witnesses. \(\square\)

We'll move on to prove the theorem-1 after proving a series of Theorems.

### 4.2 Some Necessary Theorems

**Theorem 2.** The following are equivalent:

1. \(\text{NEXP} \subset C\)
2. \(\text{NEXP}\) has \(C\) witnesses of polynomial size
3. \(\text{NEXP}\) has oblivious \(C\) witnesses of polynomial size

**Proof:**

(1) \(\Rightarrow\) (2) : This side is proven in the papers [IKW02],[Wil10] and [Wil11]. \(\square\)

(2) \(\Rightarrow\) (3) :
Let \(V(x,y)\) be a \(\text{NEXP}\) predicate which for some natural \(k\), takes witnesses \(y\) of length \(2^{\mid x \mid^k}\). Note, \(y\) is the truth table of of a circuit \(R\).
We are supposed to construct a \(C\)-circuit family \{\(C_n\)\} such that \(x \in L\) iff \(V(x, \text{tt}(C|x)(x,.)))\) (by definition) accepts.

Let \(x_1, x_2, ..., x_{2^n}\) be the list of strings of length \(n\) in lexicographical order. Now, define a predicate \(V'\) which takes ((\(x, r\)), \(y\)) where \(x \in \{0,1\}^n\) and \(r = 0, 1, ..., 2^n\) and \(|y| = 2^{n+k}\).
\(V'((x,r),y)\) accepts iff \(y\) can be partitioned into \(2^{\mid x \mid}\) strings of length \(2^{\mid x \mid^k}\) as \(y = z_1 z_2 ... z_{2^n}\) where for all \(i\), \(|z_i| = 2^{\mid x \mid^k}\) such that exactly \(2^{\mid x \mid} - q\) of the strings \(z_i\)'s equal to \(0^{2^{\mid x \mid^k}}\) (all zeros) and rest of the \(q\) strings \(z_j\)'s are such that \(V(x_j, z_j)\) accepts.

Note that, \(V\) runs in \(2^{O(n)}\) and \(V'\) checks for each \(z_i\) for \(2^{\mid x \mid}\) strings, so the overall running time of \(V'\) is \(2^{O(n)}\). It implies that \(V'\) is a predicate for a language in \(\text{NEXP}\) and thus it has \(C\) witnesses of polynomial size.

Now, construction of oblivious witnesses for \(V:\)
Let \(q_n = \text{no.\ strings x of length n such that x \in L(V)}\).
Say if \(u\) is such that \(V'((x,q_n),u)\) accepts \(\Rightarrow u = u_1 u_2 ... u_{2^n}\) such that exactly \(2^{\mid x \mid} - q_n\) strings are all zeros and rest \(q_n\) strings \(u_j\)'s are such that \(V(x_j, u_j)\) accepts. It implies \(u\) encodes witnesses for all the strings of length \(|x'|\).

Note that since \(V'\) is an \(\text{NEXP}\) predicate thus there exists a circuit \(C_{x',q_n}\) such that \(C_{x',q_n}(\log i)\) outputs the \(i\)th bit of \(u\).

**Claim :** \(C_{(x',q_n)}\) is the required witness circuit for all the strings of length \(|x'|\).

**Proof :** Let \(D_{ij}(x, j) := C_{(x',q_n)}(x \circ j) (x \circ j\) is the concatenation of \(x\) and \(j\) binary strings) outputs the \(j\)th bit of a valid witness for \(x\) if it exists or else outputs 0. Note that the computation of \(V'\) doesn't depend upon \(x\). So, \(q_n\) captures the no. of strings in \(L(V)\). So, if \(x \in L(V)\) then \(\exists u\) s.t \(V(x,u)\) is accepted. But then collecting all the witnesses \(u_j\) for all the strings \(x_j \in L(V)\) and
keeping rest of the strings $u_j$ all zero forms a witness for $V'$ for $(x, q_n)$. Now, we know $u$ is being implemented by $C_{(x, q_n)}$. This completes the claim and which completes the entire proof.

(3) $\Rightarrow$ (1):
Let NEXP has oblivious $C$ witnesses. Now, assume $M$ be a non-deterministic T.M such that
$L(M) \in \text{NEXP}$. Construction of circuit class $\{C_n\}$ from $C$ for $M$:
Define $V_k(x, y)$ as
For all circuits $C$ of size $|x|^k + k$,
If $tt(C)$ encodes an accepting run for $M(x)$, then
accept iff first bit of $y$ is 1
End For
Accept iff first bit of $y$ is 0.

Since $M$ is an NEXP machine thus there is a polynomial $p(n)$ such that $M$ has witness
circuit class family $\{C_n\}$. This $p(n)$ can be approximated by a $n^k + k$ for some $k$. It implies for that
$k$, $V_k$ finds all the circuits for a string $x \in L(M)$. Now, note that $V_k$ runs for at most $2^{O(n^k)}$, for all
enumeration of a string of length $n^k + k$. Thus, $V_k$ (one which encodes circuit family for $M$) is
an NEXP predicate.
Now, since $V_k$ is an NEXP predicate thus it has an oblivious $C$-circuit family $\{C_n\}$ encoding
witnesses for $V_k$.
Now, define the circuit family $\{D_n\}$ as $D_n$ outputs the first bit of the witness encoded by $C_n(x, .)$
Claim : $\{D_n\}$ is the required circuit family for $L(M)$.
Proof : By definition $D_n(x)$ where $|x| = n$, outputs the first bit of the witness encoded by $C_n(x, .)$. Now, if $x \in L(M)$, but $V_k$ rejects, then $V_k$ doesn’t have a witness circuit encoding accepting run
for $M(x)$ and first bit of witness encoded by $C_n(x, .)$ is 1. Contradiction!! Since $x \in L(M)$ thus
it has an accepting run for $V_k$. Thus it is rejected in the loop only when first bit of the witness
encoded by $C_n(x, .)$ is 0. But then it has to be accepted in the last statement of the algorithm
and thus accepted by $V_k$ itself. Hence, if $x \in L(M)$, then $V_k(x, tt(C_n(x, .)))$ accepts but for that
first bit has to be 1, thus $D_n(x) = 1$.
Now, if $x \not\in L(M)$ then plug in a dummy $y$ with first bit 0 thus $V_k(x, y)$ accepts in the last state-
ment. Thus, $C_n(x, .)$ must encoding an accepting witness for $V_k$. But $V_k$ accepts with first bit of
witness encoded by $C_n(x, .)$ as 0 when $x \in L(M)$.
This completes the claim and hence the entire proof.

Since we have shown the relation between all the three statements, thus the theorem 2 is
proven.

Theorem 3. For all $s(n)$, the following are equivalent :

1. There is a $c \in (0,1]$ such that $\text{NTIME}(2^{O(n)})$ does not have $s(cn)$ size witness circuits from $C$.

2. There is a $c \in (0,1]$ and a $P$-computable property that is useful against $C$-circuits of size at
most $s(cn)$.
Proof:

(2) ⇒ (1) : Consider A to be an algorithm that is P-computable and takes input T of length n and is useful against size-s(cn) C. Using this we construct a T.M M as follows : M takes (x,T) which accepts iff |T| = 2^{|x|} and A(T_x) accepts where T = T_1 T_2 ... T_{2^{|x|}} s.t |T_i| = 2^{|x_i|} for all i and T_x represents the xth string in the partitioning of T. Assume x to be a no. from 1 to 2^{|x|}.

Now, let L = {x | M(x,T) accepts for some string T s.t |T| = 2^{|x|}}. By definition, L ∈ NEXP.

Claim : ∃ c ∈ (0,1] such that L doesn’t have s(cn)-size witness circuits from C.

Proof : Let forall d in the set, NTIME has s(dn)-size witness circuits from C. Let us fix a d ∈ (0,1].

Now, for almost all x ∈ L there is a witness circuit C with inputs 2^{|x|} and size at most s(dn) such that M(x,tt(C)) accepts. It implies the xth, T_x block in the partitioning of tt(C) is accepted by A. Assume x to be a no. from 1 to 2^{|x|}.

Now, let L = {x | M(x,T) accepts for some string T s.t |T| = 2^{|x|}}. By definition, L ∈ NEXP.

(1) ⇒ (2) :

We’ll prove it for the case when c = 1. We can conclude for a specific c < 1. So, assume that NTIME doesn’t have s(n)-size witness circuits. Thus, we have a predicate V which runs in 2^{an} for a ≥ 1 that doesn’t have s(n)-size witness C-circuits such that L(V) is infinite.

Now, we can obtain a sequence of strings {x_i} such that x_i ∈ L(V) but for every y such that V(x_i, y) accepts, y requires at least s(|x_i|) size circuits as witness.

Construction of P-computable algorithm useful against s(dn)-size circuits C for some d : Define an algorithm A as follows : It takes a string y and rejects if y ≠ y_1 y_2 ... y_{2^l} 01^k for some k = 0,1,...,2^l and l s.t |y_i| = 2^c l.

Otherwise, A obtains k from y and accepts iff there are exactly k partitions y_i’s s.t V(x_i, y_i) accepts (x_i is the ith string of l bits in lexicographical order).

Now, let k be the no. of strings of length l which are accepted by V. Thus, for strings y with trailing 1’s are k in no. s.t A accept encodes witnesses for every string x in L(V).

Now, Let for some i and j, x_i = x_j then witness for x_i must have circuit of size at least s(l).

Using lemma-1, it implies that y that has k trailing 1’s and is accepted by A has circuit size s(l) − (c’ l)^{1+o(1)} ≥ s(l)/2 for some c’ constant.

Now, note that for every l and every k = 0,1,...,2^l, there is exactly one input length n = 2^{(c+1)l} + k + 1 on which this value of c is considered. Thus, for all those strings of length n such k equals the no. inputs of l bits accepted by V, A is circuit s(l)/2-size useful. Note that s(d log n) < s(l)/2 for some constant d and this completes the proof.

Theorem 4. Let \{s_k(n)\} be an infinite family of functions such that for all k, there is a P-computable property P_k that is useful against all C-circuits of s_k(n) size. Then there is a single property P* such that, for all k, there is a c > 0 such that P* is useful against all C-circuits of s_k(cn) size.
Proof:
First we define an algorithm HISTORY(y) which will turn out to be the required property.
Let b(n) denote the nth string of \{0, 1\}\* in lexicographical order.
We define SUCCINCT HALTING problem : \{⟨M, x, b(n)⟩ | Nondeterministic TM M accepts x within at most n steps⟩
HISTORY(y) : Compute z = b(|y|). If z does not have the form ⟨M, x, b(n)⟩, reject. Accept iff there is a prefix y’ of y with length equal to a power of two such that y’ encodes an accepting computation history to z ∈ SUCCINCT HALTING.

Note that |z| ≤ |y|. SUCCINCT HALTING problem runs for O(b(n)). Running time of HISTORY is upper-bounded by \log |z|.O(b(|y|)) ≤ |z|^{O(1)} ≤ |y|^{O(1)}. Hence, HISTORY runs in polynomial in |y|.

Now, using the following claim we can conclude Theorem-4:
Claim : HISTORY is useful against C circuits of size s(cn) for some c > 0 iff there is some P-time property that is useful against C circuits of size s(c’n).
Proof:
Since, HISTORY runs in polynomial time in its input length thus if HISTORY is useful against C circuits of size s(cn) for some c > 0, then we get a P-time natural property for free. So, one side of claim is proven.
For the other direction assume that there is P-time property useful against C circuits of size s(cn). Using, Theorem-3, we conclude there is a constant d > 0, such that NTIME[2^{O(n)}] doesn’t have witnesses of size s(dn) from C circuits. So, let V be a predicate which runs in 2^{kn}, k ≥ 1 and doesn’t have C-circuits as witness of size s(dn). Let M be a Nondeterministic TM such that L(M) ∈ NTIME and L(M) = L(V). This implies that there are infinitely many inputs x_i of M which doesn’t have witness of size s(dn) from C circuits. It means that there are infinitely many z_i = ⟨M, x_i, b(2^{kn})⟩ where M(x_i) doesn’t have accepting run y_i which is encoded by a circuit of size s(dn).
Let us fix a z_i with above condition. Now, for a y such that |y| = n and z_i = b(n), HISTORY(y) accepts but for all y’ which has circuit size s(dn) and |y’| = n, HISTORY(y’) rejects. Note, y’ has length a power of 2 using Proposition-1. Thus, HISTORY is useful against C circuits of size s(dn). This completes the other side of the claim.
So, the is proven both sides

Now, we show that Theorem-4 follows from the previous claim : If there is P-computable P_k property useful against C circuits of size s_k(n) then HISTORY is useful against C circuits of size s_k(cn) for some c > 0. But, since this is true for all k, hence HISTORY is useful for all integer k. Thus the claim follows.

Now, we prove Theorem-1 : By Theorem-2, NEXP ⊄ C iff NEXP doesn’t have C witnesses of polynomial size.
Let s(n) = n^k for arbitrary k. From previous statement, NTIME doesn’t have s(n) size C circuits as witness. But this implies, there is a P-computable property that is useful against C circuits of size s(cn) for some c > 0. But since k is arbitrary, thus it is true for every k.
So, we have \( s_k(c_k n) = (c_k n)^k \) be an infinite family of functions such that for all \( k \), there is a P-computable property \( P_k \) that is useful against all circuits of \( s_k(c_k n) \) size for constant \( c_k \). By Theorem-4, for all \( k \), there is a \( c > 0 \) such that a property \( P^* \) is useful against all \( C \)-circuits of \( (cn)^k \) size. Hence, NEXP \( \not\subset C \) iff for all \( k \), there is a \( c > 0 \) such that a property \( P^* \) is useful against all \( C \)-circuits of \( (cn)^k \) size.

\[ \square \]

5 NATURAL PROPERTY AND DERANDOMIZATION

**Theorem 5.** Let \( C \) be a polynomial-size typical circuit class. The following are equivalent:

1. There are no P-natural properties useful (respectively, ae-useful) against \( C \).
2. ZPE has \( C \) seeds for almost all (resp., infinitely many) input lengths.

5.1 DEFINITIONS

Notations :

- \( ZPE = \text{ZPTIME}[2^{O(n)}] \) : Class of languages solvable in \( 2^{O(n)} \) time with randomness and no error (machine can output ? or output “don’t know”).
- \( RPE = \text{RPTIME}[2^{O(n)}] \) : Class of languages solvable in \( 2^{O(n)} \) time with randomness in one side and no error (machine can output ? or output “don’t know”) i.e. if a string \( c \not\in L \) where \( L \in RPE \) then a machine \( M \) s.t \( L(M) = L \) outputs 0 with no error.

Seeds :

Let \( L \in ZPE \). A ZPE predicate for \( L \) is a procedure \( M(x,y) \) that runs in time \( \text{poly}(|y|).2^{O(|x|)} \), such that on every \( x \):

- The output of \( M(x,y) \) is in the set \( \{1,0,?\} \).
- \( x \in L \Rightarrow \Pr_y[M(x,y) \text{ outputs } 1] \geq 2/3 \), and for all \( y \), \( M(x,y) \in \{1,?\} \).
- \( x \not\in L \Rightarrow \Pr_y[M(x,y) \text{ outputs } 0] \geq 2/3 \), and for all \( y \), \( M(x,y) \in \{0,?\} \).

ZPE has \( C \) seeds if for every ZPE predicate \( M \), there is a \( k \) such that for all \( x \), there is a \( C \)-circuit \( C_x \) of size at most \( n^k + k \) such that \( M(x, t t(C_x)) \neq ? \). Similarly, seeds can be defined for RPE.

**Proof of theorem-5** : We give proof assuming the negation of statements for each side.

\( \neg (1) \Rightarrow \neg (2) \) :

It implies there is a P-computable property that is ae-useful (infinitely-many resp.). Thus, we have an algorithm \( A \) that runs for \( n^c \) and accepts at least \( 1/n^d \) for some constants \( c,d \) and for almost all \( n \) and for all \( k \), \( A \) rejects inputs of length \( n \) with \( CC = (\log n)^k \).

Contraction of ZPE predicate :

Let \( b(n) \) be the nth string in the lexicographical ordering. Consider \( \epsilon > 0 \). Now, define a predicate \( V(x,z) \) as:

If \( x \neq b(|z|) \) then output ?, else partition \( z = z_1 z_2 ... z_s \) where for all \( |z_i| = |z|^\epsilon/d \) and \( s = |z|^{1-\epsilon/d} \) and output 1 iff \( A(z_i) \) accepts for some \( i \).
Assume $y$'s are of length 2.

Thus, assume that there is a ZPE predicate $V$ which doesn't have $C$-circuits for almost all (resp. infinitely many) input lengths $n$. It means that there are almost all (resp. infinitely many) inputs $n$, $A$ rejects with circuit size $(\log n)^k$.

Now, fix $n$ length string $x$, if $V$ accepts $x$ then witness $z$ must have fixed length. Thus for different input length $a$ different input length witness is required. Also, for all those witnesses such that $|z|$ length strings having $(\log |z|)$ size circuits are rejected by $A$, since $z$ is accepted so it has $CC$ at least $(\log |z|)$. This is true for almost all (resp. infinitely many) input lengths $|x|$. Also, whenever $z_i$ is accepted for those $i$ input lengths, $CC(z_i) \geq (\log((|z|^{c/d})^k \geq (\log k)^2 |z|)$. But using lemma-1, $CC(z) \geq \Omega((\log k^2 |z| - (\log |z|)^{1+o(1)})$ for all $k$. Hence, $V$ is a ZPE predicate which doesn't have $C$ seeds for almost all (resp. infinitely many) input lengths.

Now, $\neg(2) \Rightarrow \neg(1)$:

Thus, assume that there is a ZPE predicate $V$ which doesn't have $C$-circuits for almost all (resp. infinitely many) inputs. It means that there are almost all (resp. infinitely many) input lengths, there are strings $x_i$ accepted by $V$ such that for all $k$, for all $y$ s.t $V(x,y)$ accepts $CC(y) \geq (c|x_i|)^k$.

Assume $y$'s are of length $2^c|x_i|$.

Now, we define a predicate $V'$ as follows: If $|y| \neq 2^{l+c|x|}$ then it rejects where $l$ is the smallest integer such $2^{|x|} \leq 2^l$, otherwise partitions $y$ into $2^l$ strings $y = r_1 r_2 ... r_{2^l}$ where $|r_i| = 2^{|x|}$ and accepts iff $V(x, r_i) \neq ?$ for some $i$. Note that if $x$ is string which doesn't have a seed for $V$, then $CC(r) \geq |x|^k$ forall $k$ using Proposition-1.

Claim : $V'$ is ZPE predicate.
Proof : $Pr_{r_i}[\forall i, V(x, r_i) = ?$ for some string $x] < 1/3$ as $V$ is ZPE predicate and $Pr_{r_i}[V(x, r_i)$ accepts$] \geq 2/3$. Hence, a random string $y$ of length $2^{l+c|x|}$ is accepted by $V'$ with probability $2/3$.

Construction of a P-computable property:
Consider an algorithm $A$ which takes string $x$ of length $n$ and computes largest $n'$ such that $n \geq 2^{l+c'n'}$ then sets $r = 1$ to $2^{l+c'n'}$ bits of $x$ and partitions $r = r_1 r_2 ... r_{2^l}$ where $|r_i| = 2^{c|x|}$ and accepts iff for all strings $x$ of length $n$ s.t $V(x, r_i) \neq ?$ for some $r_i$.

Claim : $A$ is P-computable and the required property that is useful almost everywhere (resp. infinitely many) against $C$ circuits.
Proof: Running time of $A$ is upper-bounded by $2^{c'n'}2^l$ for the computation of $V$ for each $r_i$ for $2^{cn'}$ many strings. But then running time is poly$(n)$.

In the previous claim, we prove that $V'$ accepts with probability $2/3$ and hence $A$ accepts at least $1/2$ of the strings of length $n$.

Since, for almost all (resp. infinitely many) input lengths $A$ rejects strings with circuits size at most $O((\log n)^k)$ for all $k$, otherwise $r_i$ will have circuits of size $O((\log n)^k)$ as well.
Hence the three conditions for a P-computable property that is useful almost everywhere (resp. infinitely many) against $C$ circuits is proven and thus the claim follows. With the claim we have obtained the required property $P$ and this side of the theorem is proven.

So, finally the theorem-5 is proven.