Proof that P ≠ NP

Author

Robert DiGregorio
0x51B4908DdCD986A41e2f8522BB5B68E563A358De

Abstract

Using sorting keys, we prove that P ≠ NP.

Part 1

We define SS:

- let $\forall$ list x [SS(x) = the list of all sublist sums of x]
  - example: $SS(x)[01100101_2] = x[0] + x[2] + x[5] + x[6]$
  - note: SS(x) can be constructed lazily
- let $\forall$ list x [SS accepts x $\Leftrightarrow |SS(x)| = 2^{|x|}$]
  - note: this forces every element of SS(x) to be unique

We define keys and sorting keys:

- let $\forall$ list x $\forall k \in \mathbb{N}[k \in KEYS(x) \Leftrightarrow k < |x|]$
- let $\forall$ list x $\forall k \in KEYS(SS(x))\ \forall sk \in SKEYS(SS(x))\ [SS(x)[k \oplus sk] = \text{SORT}(SS(x))[k]]$
  - note: because every element of SS(x) is unique, SS(x) has 1 sorting key

We define L_0:

- let L_0 be all possible inputs for deterministic Turing machine M_0 such that
  - $M_0(k \in KEYS(SS(A))$, list $A) = k \notin SKEYS(SS(A))$
- let C_0 be all possible inputs for deterministic Turing machine V_0 such that
  - $V_0($list A, $k \in KEYS(SS(A))$, $x \in KEYS(SS(A))$, $y \in KEYS(SS(A)))$ = (compare x to y $\neq$ compare $SS(A)[x \oplus k]$ to $SS(A)[y \oplus k]$)
- all YES expressions in L_0 can be verified with YES certificates in C_0 in polynomial time $\Rightarrow$
  - $L_0 \in NP$

Part 2

We presume part 1 does not prove $P \neq NP$:

- let $L_0 \in P$
  - note: if L_0 is not in P, $P \neq NP$

We define digest functions (these are not one way functions):

- let $\forall x \in \mathbb{N}[DIGEST_0(x) =$ even bits of $x \oplus$ odd bits of $x]$
  - example: $DIGEST_0(00011011_2) = 0110_2$
- let $\forall x \in \mathbb{N}[DIGEST_1(x) = x > 1\ ?\ DIGEST_1(DIGEST_0(x)) : x]$

We define L_1:

- let L_1 be all possible inputs for deterministic Turing machine M_1 such that
  - $M_1($list $A) = \exists p \in \mathbb{N} \exists sk \in SKEYS(SS($PERMUTATION$(A, p)))\ [DIGEST_1(sk) = 0]$
    - note: it is possible that every sorting key of every permutation of A digests to 0
    - note: it is possible that every sorting key of every permutation of A digests to 1
    - note: there are $|A|!$ permutations of A
    - note: no specific search algorithm is implied
- $M_1($list $A)$ searches $p \in \mathbb{N}$ searches $SKEYS(SS($PERMUTATION$(A, p)))$ for $[DIGEST_1(sk) = 0] \wedge$
  - $\exists$ list $A\ [\forall p \in \mathbb{N}\ \forall sk \in SKEYS(SS($PERMUTATION$(A, p)))\ [DIGEST_1(sk) = 1]] \Rightarrow$
  - $\exists$ list $A\ [M_1$ runs in $\Omega(|A|!)$ time$] \Rightarrow$
  - $L_1 \notin P$
- let C_1 be all possible inputs for deterministic Turing machine V_1 such that
  - $V_1($list $A, p \in \mathbb{N}, k \in \mathbb{N}) = \neg L_0($PERMUTATION$(A, p), k) \wedge DIGEST_1(k) = 0$
- L_0 is in $P \Rightarrow$
  - all YES expressions in L_1 can be verified with YES certificates in C_1 in polynomial time $\Rightarrow$
  - $L_1 \in NP$

We prove that $P \neq NP$:

- $L_1 \notin P \wedge L_1 \in NP \Rightarrow P \neq NP$