P ≠ NP

Author

Robert DiGregorio
0x51B4908DdCD986A41e2f8522BB5B68E563A358De

Abstract

A problem exists that’s hard to solve but easy to verify a solution for.

```
∃ V ∈ {V : V is a deterministic polynomial time Turing machine}
∀ M ∈ {M : M is a deterministic Turing machine ∧
  ∀ S ⊆ ℕ
  ∀ k ∈ {k ∈ ℕ : k ≤ |S|}
  ∀ w ∈ {w ∈ {0, 1}* : w is S and k encoded as a binary string ∧ M accepts w} [
    M(w) = ∃ b ∈ {b ⊆ {a ⊆ S : |a| = k} : |b| = k} [
note: b is a k-subset of a k-subset of S
      V(b)
note: V is a polynomial time verifier for M using b as a certificate
    ]
  ]
} [
  ∀ S ∈ ℕ
  ∀ k ∈ {k ∈ ℕ : k ≤ |S|}
  ∀ w ∈ {w ∈ {0, 1}* : w is S and k encoded as a binary string ∧ M accepts w}
  ∀ n ∈ ℕ
  ∀ F ∈ {F : F is a deterministic Turing machine ∧ (F accepts w ⇔ |w| = n)} [
    F(w) = M(w)
  ] ⇒
  ∀ S ∈ ℕ
  ∀ k ∈ {k ∈ ℕ : k ≤ |S|}
  ∀ w ∈ {w ∈ {0, 1}* : w is S and k encoded as a binary string ∧ M accepts w}
  ∀ n ∈ ℕ
  ∀ F ∈ {F : F is a deterministic Turing machine ∧ (F accepts w ⇔ |w| = n)} [
    F has O(n ^ log(n) ^ log(n)) certificates
  ] ⇒
  ∀ S ∈ ℕ
  ∀ k ∈ {k ∈ ℕ : k ≤ |S|}
  ∀ w ∈ {w ∈ {0, 1}* : w is S and k encoded as a binary string ∧ M accepts w}
  ∀ n ∈ ℕ
  ∀ F ∈ {F : F is a deterministic Turing machine ∧ (F accepts w ⇔ |w| = n)} [
    F runs in ≥ O(2 ^ log(n)) steps
note: this is because each certificate is in a different node on the decision tree and a balanced
decision tree requires the least amount of steps
  ] ⇒
  ∀ w ∈ {w ∈ {0, 1}* : w is S and k encoded as a binary string ∧ M accepts w} [
    M runs in ≥ O(2 ^ log(|w|)) steps
  ] ⇒ the decision problem of M ∉ P
] ⇒ P ≠ NP
note: the decision problem of M is in NP because YES solutions can be verified in polynomial time by V
using b as a certificate
```