SMALL JUMP WITH NEGATION-UTM TRAMPOLINE

KOJI KOBAYASHI

1. Introduction

This paper divide some complexity class by using fixpoint and fixpointless area of Decidable Universal Turing Machine (UTM). Decidable Deterministic Turing Machine (DTM) have fixpointless combinator that add no extra resources (like Negation), but UTM makes some fixpoint in the combinator. This means that we can jump out of the fixpointless combinator system by making more complex problem from diagonalisation argument of UTM.

As a concrete example, we proof L is not P . We can make Polynomial time UTM that emulate all Logarithm space DTM (LDTM). LDTM set close under Negation, therefore UTM does not close under LDTM set. (We can proof this theorem like halting problem and time/space hierarchy theorem, and also we can extend this proof to divide time/space limited DTM set.) In the same way, we proof P is not NP. These are new hierarchy that use UTM and Negation.

2. L is not P

Definition 1. "DTM" is defined as Decidable Deterministic Turing Machine set. "$LDTM$" is defined as logarithmic space DTM. "$pDTM$" is defined as polynomial time DTM. "$\bigcirc DTM$" is defined as DTM that some resource (time, space) limited.

"UTM" is defined as Universal Turing Machine set. "$UTM\left(C\right)$" is defined as minimum UTM that can emulate all $M \in C$. $\langle M\rangle$ is defined as code number of a $M \in DTM$ that $U \in UTM$ emulate. That is, $\forall w\left[U\left(\langle M\rangle, w\right)=M\left(w\right)\right]$ and $U\left(\langle M\rangle\right)=M$.

"$Negate\left(C\right)$" is defined as minimum Negation system that include C. That is, $\forall C\left[\left(C\subset Negate\left(C\right)\right)\wedge\left(\forall c\in Negate\left(C\right)\left[\neg c\in Negate\left(C\right)\right]\right)\right]$.

Theorem 2. $\forall r\in\bigcirc DTM\left(\neg r\in\bigcirc DTM\right)$

Proof. It is trivial from DTM structure.

If DTM is

$M=\left(Q,\Sigma,\Gamma,\delta,q_0,q_1,q_2\right)$

then this dual machine

$\overline{M}=\left(Q,\Sigma,\Gamma,\delta,q_0,q_2,q_1\right)$

compute $\neg M$ without extra resources.

Therefore negation of $\bigcirc DTM$ is also in $\bigcirc DTM$. □

Theorem 3. $\exists U\in UTM\left(LDTM\right)\left[U\in pDTM\right]$

Proof. It is trivial because some $U'\in UTM$ can emulate all $LDTM$ in polynomial time. Therefore, we can make $U\in pDTM$ by limiting at polynomial time (if U' compute over polynomial time, U reject these input). □

Theorem 4. $L \subsetneq P$

Proof. We can proof this theorem like halting problem and time/space hierarchy theorem.

Because of

$\forall U \in UTM\left(LDTM\right), M \in LDTM\left[U\left(\langle M\rangle\right) = M\right]$ 1

all $M \in LDTM$ have index $\langle M\rangle$. Therefore we can make H which is diagonalization of U.

$H\left(\langle M\rangle\right) = U\left(\langle M\rangle, \langle M\rangle\right)$

$$\begin{array}{llccccc} & & \langle M_0\rangle & \langle M_1\rangle & \langle M_2\rangle & \langle M_3\rangle & \cdots \\ M_0 & =\{ & \underline{\top} & \top & \bot & \top & \cdots \\ M_1 & =\{ & \bot & \underline{\bot} & \top & \bot & \cdots \\ M_2 & =\{ & \bot & \bot & \underline{\bot} & \top & \cdots \\ M_3 & =\{ & \top & \top & \bot & \underline{\bot} & \cdots \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \\ H & =\{ & \top & \bot & \bot & \bot & \cdots \end{array}$$

$H \in pDTM$ because $U \in pDTM$ 2 and H input size is at least half of U.

Mentioned above 2,

$\forall r \in LDTM\left(\neg r \in LDTM\right)$

we can make G which is Negation of diagonalization.

$G\left(\langle M\rangle\right) = \neg H\left(\langle M\rangle\right) = \neg U\left(\langle M\rangle, \langle M\rangle\right)$

$$\begin{array}{llccccc} & & \langle M_0\rangle & \langle M_1\rangle & \langle M_2\rangle & \langle M_3\rangle & \cdots \\ M_0 & =\{ & \underline{\top} & \top & \bot & \top & \cdots \\ M_1 & =\{ & \bot & \underline{\bot} & \top & \bot & \cdots \\ M_2 & =\{ & \bot & \bot & \underline{\bot} & \top & \cdots \\ M_3 & =\{ & \top & \top & \bot & \underline{\bot} & \cdots \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \\ H & =\{ & \top & \bot & \bot & \bot & \cdots \\ G & =\{ & \bot & \top & \top & \top & \cdots \end{array}$$

$G \notin LDTM$ because $\forall M \in LDTM\left[G\left(\langle M\rangle\right) \neq M\left(\langle M\rangle\right)\right]$. On the other hand, $G \in pDTM$ because $H \in pDTM$.

Therefore, $G \in pDTM\left(G \notin LDTM\right)$ and $L \subsetneq P$. □

We can expand above result to general DTM.

Theorem 5. $\forall CC \subset DTM\left[Negate\left(UTM\left(Negate\left(CC\right)\right)\right) \nsubseteq Negate\left(CC\right)\right]$

Proof. We omit the proof because this proof is same as previous. □

Corollary 6. $Negate\left(UTM\left(\bigcirc DTM\right)\right) \nsubseteq \bigcirc DTM$

3. P is not NP

Theorem 7. $\exists U \in UTM\left(pDTM\right)\left[U \in pNTM\right]$

Proof. We can make some oracle TM which oracle emulate transition function.

$np^p \mid np \in NP, p \in P$

$p\left(\langle t\rangle, w\right) = t\left(w\right)$

$t \in P$: transition function

$\langle t\rangle$:code number of transition function t

w: t's input (state and symbol)

$p(\langle t \rangle, w)$ accept if and only if $t(w)$ accept and output $t(w)$.

Oracle TM $np^p \in pNTM$ and np^p can emulate all $pDTM$. Therefore $np^p \in UTM(pDTM)$. □

Note 8. np^p can change transition function more flexible than $pDTM$ in less time. In fact, np^p can increase transition function with logarithm time (by computing these transition functions in parallel). Therefore, np^p have more chance to compute more complex problems.

Theorem 9. $P \subsetneq NP$

Proof. (Proof by contradiction.) Assume to the contrary that $P = NP$.

$P = NP$ means that $P = NP = coNP = PH$, therefore NP close under Nagation.

In the same way as mentioned above 4, we get $P \subsetneq NP$. This result contradicting assumption $P = NP$.

Therefore $P \subsetneq NP$. □

Corollary 10. $P \subsetneq coNP$

4. Trampoline Hierarchy between Negation and UTM

This result shows that we can jump over border of asymptotic analysis by using Negation (fixpointless combinator) and UTM (fixpoint creator). Therefore, combination of UTM and Negation make new complexity class. That is, there are some Hierarchy of UTM and Negation.

References

[1] Michael Sipser, (translation) Kazuo OHTA, Keisuke TANAKA, Masayuki ABE, Hiroki UEDA, Atsushi FUJIOKA, Osamu WATANABE, "Introduction to the Theory of COMPUTATION Second Edition (Japanese version)", 2008