Honesty and Time-Constructibility in Type-2 Computation

Chung-Chih Li

School of Information Technology
Illinois State University
Normal, IL 61790, USA

One of the reviewers of my CIE 2009 paper, *Union Theorems in Type-2 Computation*, proposed a related and interesting question: "What happens to the honesty theorem in type-2 computation?" In response to the question, I would like to present my finding in this presentation.

Informally, we say that a function is *honest* if we do not need too much resource in order to compute a small value (e.g. a 0-1 function can be highly dishonest). The problem with dishonest functions is that, if they are used as some resource bounds, some unnatural phenomenon may occur such as the gap theorem stating that, by allowing more resource, we not necessarily obtain a bigger complexity class. Blum fixes this problem with his measured sets (this is stated as the *compression theorem.*) Any measured set is a collection of honest functions, but not every honest function is in the measured set. Thus, do we lose any complexity classes if we restrict our complexity-bound functions to a measured set? McCreight and Meyer answer this question with a no in their *honesty theorem*. How about Type-2 computation? Based on my definitions given in a series of previous works, we have an immediate honesty theorem as follows:

Theorem: *There exists a recursive function g such that $\{\alpha_{g(i)} | i \in \mathbf{N}\}$ is a measured set, and for all $\beta \in \mathbf{T}_2\mathbf{TB}$, if $\beta = \varphi_i$, then $\mathbf{C}(\beta) = \mathbf{C}(\alpha_{g(i)})$.*

All techniques used in the proof of the original honesty theorem can be used in the proof of the theorem above. But we are not interested because there is no compelling reason to have it as the gap phenomenon does not exist in type-2 computation under our framework (this is stated in a theorem we call *Inflation Theorem*). Instead, we are interested in a non-trivial notion of *Time-Constructibility* for type-2 computation as follows:

Definition: *We say that β is type-2 time-constructable if and only if $\beta \in \mathbf{T}_2\mathbf{TB}$ and β is useful and there is a φ-program i for β such that, for every $(\sigma, x) \in \mathcal{F} \times \mathbf{N}$, $\varphi_i(\langle \sigma, x \rangle) = \beta(\sigma, x)$ and, for every $(f, x) \in \mathcal{T} \times \mathbf{N}$, there exists $y \in \mathbf{N}$ such that*

$$\lim_{\sigma \to f} \Phi_i(\langle \sigma, x \rangle) = y + |\langle \sigma, x \rangle|.$$

We will review some needed definitions and argue that, $\beta \in \mathbf{T}_2\mathbf{TB}$ is type-2 time-constructable if and only if β is *useful* and the cost of the computation is convergent. Also, if β is type-2 time-constructable, then it is *locking detectable* and polynomial time computable.