34. A Mathematical Theory of Randomized Computation. I

By Shinichi YAMADA
Waseda University and Nihon Unisys, Ltd.

(Communicated by Shokichi IYANAGA, M. J. A., April 12, 1988)

1. Introduction. Randomized algorithms, or probabilistic algorithms, are the extended algorithms which incorporate random input data and random choices. They have recently been recognized as an important area of information science and undergone intensive studies. Whereas a mathematical theory of randomized computation is indispensable to guarantee the reliability of stochastic software, there has been no theory comparable to Scott's theory [3, 4] for deterministic computation. The present series of communications outlines a new randomized domain theory [5] which, while naturally extending Scott's deterministic domain theory, readily provides a denotational semantics for a class of high level probabilistic programming languages and is also applicable to machine learning and algorithmic information theory.

2. The postulates for the probability theory. There have been pointed out many pathological phenomena that arise within the framework of the axiomatic theory of probability. In order to avoid these defects, we postulate the following Axioms 1 and 2 for the probability theory. The space Ω in a measurable space $(\Omega, \mathcal{A})$ is called a *basic space*.

Axiom 1. *Every basic space is compact, Hausdorff and second countable, if it is endowed with a topology. Every basic space has power* $\leq \aleph_1$, *if it is not endowed with any topology.*

Axiom 2. *Every probability space is separable and perfect.*

Given a measurable space $(\Omega, \mathcal{A})$, we shall call a measure μ on $\mathcal{A}$ a *subprobability measure* if it satisfies $\mu(\Omega) \leq 1$.

3. The Scott topology. The essence of Scott's theory is the idea of finite approximation coupled with the Kleene first recursion theorem. A computation is thought as a sequence or a directed set of increasingly refined approximations whose supremum is the desired result. To formulate the idea, we introduce "undefined value" $\perp$ and "approximation ordering" $\sqsubseteq$ in a domain of computation C and think of the domain as a poset $\boldsymbol{D}=(D, \sqsubseteq)$, where $D := C \cup \{\perp\}$. The intuitive meaning of $\sqsubseteq$ is as follows: Every x in C is a definite value "well defined" compared with the undefined value $\perp$. So $\perp$ is considered to "approximate x with respect to the degree of definition". In other words, "x is better defined than $\perp$", and written $\perp \sqsubseteq x$. For $x, y \in C$, if $x \neq y$ then "x and y do not approximate each other". Technically we define it as follows:

(1) A poset $\boldsymbol{D}=(D, \leq)$ is a *conditionally complete poset* (*ccp* for short)