Reasoning About States of Probabilistic Sequential Programs*

R. Chadha, P. Mateus, and A. Sernadas

SQIG – IT and IST, Portugal
{rch, pmat, acs}@math.ist.utl.pt

Abstract. A complete and decidable propositional logic for reasoning about states of probabilistic sequential programs is presented. The state logic is then used to obtain a sound Hoare-style calculus for basic probabilistic sequential programs. The Hoare calculus presented herein is the first probabilistic Hoare calculus with a complete and decidable state logic that has truth-functional propositional (not arithmetical) connectives. The models of the state logic are obtained exogenously by attaching sub-probability measures to valuations over memory cells. In order to achieve complete and recursive axiomatization of the state logic, the probabilities are taken in arbitrary real closed fields.

1 Introduction

Reasoning about probabilistic systems is very important due to applications of probability in distributed systems, security, reliability, and randomized and quantum algorithms. Logics supporting such reasoning have branched in two main directions. Firstly, Hoare-style [27,21,6] and dynamic logics [9,17] have been developed building upon denotational semantics of probabilistic programs [16]. The second approach enriches temporal modalities with probabilistic bounds [10, 13, 23].

Our work is in the area of Hoare-style reasoning about probabilistic sequential programs. A Hoare assertion [11] is a triple of the form $\{\xi_1\}\, s\, \{\xi_2\}$ meaning that if program s starts in state satisfying the state assertion formula ξ_1 and s halts then s ends in a state satisfying the state transition formula ξ_2. The formula ξ_1 is known as the pre-condition and the formula ξ_2 is known as the post-condition. For probabilistic programs the development of Hoare logic has taken primarily two different paths. The common denominator of the two approaches is forward denotational semantics of sequential probabilistic programs [16]: program states are (sub)-probability measures over valuations of memory cells and denotations of programs are (sub)-probability transformations.

The first sound Hoare logic for probabilistic programs was given in [27]. The state assertion language is *truth-functional*, *i.e.*, the formulas of the logic are interpreted as either true and false and the truth value of a formulas is determined

* Supported by FCT and FEDER through POCI via CLC QuantLog POCI/MAT/55796/2004 Project. Additional support for Rohit Chadha came from FCT and FEDER grant SFRH/BPD/26137/2005.

Z. Ésik (Ed.): CSL 2006, LNCS 4207, pp. 240–255, 2006.

by the truth values of the sub-formulas. The state assertion language in [27] consists of two levels: one classical state formulas γ interpreted over the valuations of memory cells and the second probabilistic state formulas ξ which interpreted over (sub)-probability measures of the valuations. The state assertion language contain terms $(\int\gamma)$ representing probability of γ being true. The language at the probabilistic level is extremely restrictive and is built from term equality using conjunction. Furthermore, the Hoare rule for the alternative if-then-else is incomplete and even simple valid assertions may not be provable.

The reason for incompleteness of the Hoare rule for the alternative composition in [27] as observed in [27,17] is that the Hoare rule tries to combine absolute information of the two alternates truth-functionally to get absolute information of the alternative composition. This fails because the effects of the two alternatives are not independent. In order to avoid this problem, a probabilistic dynamic logic is given in [17] with an *arithmetical* state assertion logic: the state formulas are interpreted as measurable functions and the connectives are arithmetical operations such as addition and subtraction.

Inspired by the dynamic logic in [17], there are several important works in the probabilistic Hoare logic, *e.g.* [14,21], in which the state formulas are either measurable functions or arithmetical formulas interpreted as measurable functions. Intuitively, the Hoare triple $\{f\}\, s\, \{g\}$ means that the expected value of the function g after the execution of s is at least as much as the expected value of the function f before the execution. Although research in probabilistic Hoare logic with arithmetical state logics has yielded several interesting results, the Hoare triples themselves do not seem very intuitive. A high degree of sophistication is required to write down the Hoare assertions needed to verify relatively simple programs. For this reason, it is worthwhile to investigate Hoare logics with truth-functional state logics.

A sound Hoare logic with a truth-functional state logic was presented in [6] and completeness for a fragment of the Hoare-logic is shown for iteration-free programs. In order to deal with alternative composition, a probabilistic sum construct $(\xi_1 + \xi_2)$ is introduced in [6]. Intuitively, the formula $(\xi_1 + \xi_2)$ is satisfied by a (sub)-probability measure μ if μ can be be written as the sum of two measures μ_1 and μ_2 which satisfy ξ_1 and ξ_2 respectively. The drawback of [6] is that no axiomatization is given for the state assertion logic. The essential obstacle in achieving a complete axiomatization for the state language in [6] is the probabilistic sum construct.

This paper addresses the gap between [27] and [6] and provides a sound Hoare logic for iteration-free probabilistic programs with a truth-functional state assertion logic. Our main contribution is that the Hoare logic herein is the first sound probabilistic Hoare logic with a truth-functional state assertion logic that enjoys a complete and decidable axiomatization.

We tackle the Hoare rule for the alternative composition in two steps. The first step is that our alternative choice construct is a slight modification of the usual if-then-else construct: we mark a boolean memory variable bm with the choice taken at the end of the execution of the conditional branch. Please note that this