Probabilistic invariants for probabilistic machines

Annabelle McIver
Carroll Morgan
Thai Son Hoang
Zhendong Jin
Ken Robinson

May 30, 2003

1 What is pGSL?

- *probabilistic Generalised Substitution Language* (*pGSL*) is the extension of *Generalised Substitution Language* (*GSL*).
- In *pGSL*, predicates (functions from state to Boolean) has been widened to functions from state to real number.
- For consistency with Boolean logic, $false \mapsto 0$, $true \mapsto 1$. In other words, it acts over 'expectations' rather than predicates.
- Notationally, we have kept the predicate as much as possible.
- Example of an expression in *pGSL*:

2 New pB construct

- *probabilistic Generalised Substitution Language* (*pGSL*): Probabilistic choice substitution $S \,{}_p\!\oplus T$.
- *probabilistic AMN* (*pAMN*):

PCHOICE p **OF**
  S
OR
  T
END

3 Example of probabilistic Library

Consider the specification of a simple Library in Fig. 1.

- The state of the machine contains three variables, namely *booksInLibrary*, *loansStarted* and *loansEnded* *booksLost* representing: the number of books in the library; the number of book loans initiated by the library; the number of book loans completed by the library; and the number of books lost, respectively.
- Initially, *booksInLibrary* has value *totalBooks* (a parameter of the machine). Both *loansStarted* and *loansEnded* are assigned 0 initially.
- The StartLoan operation (for starting a loan of a book) has a precondition that there are books available for loan; it decrements the books held and increments the book loans.
- The loss of a book will be modelled by the EndLoan operation so that, with some probability *pp*, the user fails to return a book to the library; in that case the effect of EndLoan is to consider the book lost. In this operation, the chance of a book being lost is *pp* — where *booksInLibrary* fails to increase; the other 1−*pp* of the time, *booksInLibrary* increases.
- The first term of the invariant on the left-hand side is the number of books not in the on-loan database; the second term is the number of books that are in the on-loan database.

We have two operations that can modify the state of the machine, StartLoan, for starting a loan of a book, and EndLoan, for ending the loan of a book. The StartLoan operation has a precondition that there are books available for loan; it decrements the books held and increments the book loans.

The loss of a book will be modelled by the EndLoan operation so that, with some probability *pp*, the user fails to return a book to the library; in that case the effect of EndLoan is to consider the book lost. The *PCHOICE* construct is the *probabilistic AMN* (*pAMN*) counterpart of ${}_{p}\oplus$. In this operation, the chance of a book being lost is *pp* — where *booksInLibrary* fails to increase; the other 1−*pp* of the time, *booksInLibrary* increases. The variable *booksLost* is introduced to record the number of books lost and is initialised to 0.

The first term of the invariant on the left-hand side is the number of books not in the on-loan database; the second term is the number of books that are in the on-loan database. This specification is simply modelling the effect of loss, without attempting to identify where it occurs. In practice, loss could be the consequence of a faulty (unreliable) loan or return operation. At some point, "loss" needs to be recognised and that is modelled by the probabilistic *booksLost*:=*booksLost*+1.

A new *EXPECTATION* clause is introduced into *pAMN* for declaring the probabilistic invariant. It gives an expression V over the program variables, denoting the random-variable invariant, and an initial expression e which is evaluated over the program variables when the machine is initialised. We write it $e \Rrightarrow V$. Its interpretation is that the expected value of V, at any point, is always at least the value of e initially. The value of e can be depended on the context of the machine (machine's parameters, constants, etc.) but often e will just be a constant.

When we claim that the expectation $V \mathrel{\widehat{=}} pp * loansEnded - booksLost$ is an expected-value invariant for this machine, we mean that, if we check the value of V many times during the running operations of the Library, then the average of our observation of V will be at least 0. That's the mathematical meaning of our probabilistic invariant. From that we can conclude for our probabilistic library machine, the expected number of books lost (value of *booksLost*) is bounded above by *pp*∗*loansEnded*.

MACHINE $ProbabilisticLibrary\ (\ totalBooks\ )$

SEES $Real\_TYPE$

CONSTANTS pp

PROPERTIES $pp \in REAL \wedge pp \leq real\ (\ 1\ ) \wedge real\ (\ 0\ ) \leq pp$

VARIABLES

$booksInLibrary\ ,\ loansStarted\ ,\ loansEnded\ ,\ booksLost$

INVARIANT

$booksInLibrary \in \mathbb{N} \wedge loansStarted \in \mathbb{N} \wedge loansEnded \in \mathbb{N} \wedge booksLost \in \mathbb{N} \wedge$
$loansEnded \leq loansStarted \wedge$
$booksInLibrary + booksLost + loansStarted - loansEnded = totalBooks$

EXPECTATIONS

$real\ (\ 0\ ) \Rrightarrow pp \times real\ (\ loansEnded\ ) - real\ (\ booksLost\ )$

INITIALISATION

$booksInLibrary\ ,\ loansStarted\ ,\ loansEnded\ ,\ booksLost := totalBooks\ ,\ 0\ ,\ 0\ ,\ 0$

OPERATIONS

StartLoan $\widehat{=}$
PRE $booksInLibrary > 0$ **THEN**
$booksInLibrary := booksInLibrary - 1$ ||
$loansStarted := loansStarted + 1$
END ;

EndLoan $\widehat{=}$
PRE $loansEnded < loansStarted$ **THEN**
PCHOICE pp **OF**
$booksLost := booksLost + 1$
OR
$booksInLibrary := booksInLibrary + 1$
END ||
$loansEnded := loansEnded + 1$
END

END

Figure 1: Simple probabilistic Library

4 Proof obligation generator

Proof obligations are generated similarly to standard specification with the exception that the invariant is now a real-value, not a Boolean. In order to prove that the *real* invariant is bounded below, we have to prove the following:

P1: The initialisation needs to establish the lower bound of the probabilistic invariant, given the context of the machine (information about sets and constants)

$$e \Rrightarrow [Init]I \ .$$

P2: The operations do not decrease the expected value of the probabilistic invariant, i.e. the expected value of the invariant after the operation is at least the expected value before the operation

$$I \Rrightarrow [Op]I \ .$$