A theorem on the equivalence of a precondition—postcondition specification and an LD-relational specification

Robert L. Baber
2003 September – 2004 January

Background: Traditionally postconditions are viewed as subsets of the state space or Boolean expressions defining subsets of the state space (characteristic predicates of the state space). In order to provide sufficient generality for practical applications, such postconditions must be allowed to refer to (to depend on) the initial values of program variables. In effect, this changes postconditions into Boolean functions of the initial state and the final state, or, correspondingly, into relations on the state space.

Context: Consider two types of specifications to be satisfied by a program pgm:

1. {V} pgm {P} strictly [a precondition–postcondition specification]
2. (Cs, Rs) [an LD-relational specification]

where the precondition V is a subset of the state space (the set of all data environments), the postcondition P is a relation on the state space, and Cs and Rs are the competence set and the relation respectively constituting an LD-relation specifying a program pgm. The specifications (V, P) and (Cs, Rs) are not necessarily deterministic.

The meanings of the above two types of specifications for initial states not satisfying V or Cs are different. The specification {V} pgm {P} strictly permits any behaviour of the program pgm for initial states not satisfying V. However, the LD-relational specification (Cs, Rs) places certain restrictions on the permitted behaviour of the program for initial states not satisfying Cs. In particular, if the program terminates for such initial states, the pair of initial and final states must be in the relation Rs. I.e., the LD-relational specification (Cs, Rs) requires that the program executed on an initial state not satisfying Cs either (1) does not terminate or (2) terminates in a state satisfying the relation Rs.

Note that a program pgm satisfies the precondition—postcondition specification (V, P) if and only if {V} pgm {P} strictly. The program represented by the LD-relation (Cp, Rp) satisfies the LD-relational specification (Cs, Rs) if and only if

Cs□Cp □ Rp□Rs

Note also that if (Cp, Rp) is the LD-relation for a program pgm, then (Cp□ S, Rp|S) is the LD-relation for the program pgm restricted to any subset S of the state space. For deterministic programs, this follows from certain definitions. For non-deterministic programs, this can be viewed as the definition of restricting a program to a certain domain.

Theorem (informally stated): When V=Cs and P=Rs, specifications of the above two types are equivalent for deterministic programs in the sense that a deterministic program pgm satisfies the precondition—postcondition specification if and only if the program pgm restricted to the competence set Cs satisfies the LD-relational specification. I.e., pgm satisfies the specification (V, P) if and only if (pgm|Cs) satisfies the specification (Cs, Rs).

Theorem (formally stated):

If

pgm is a deterministic program, [I.e., pgm is a function on the state space.]
(Cp, Rp) is the LD-relation for pgm, [Note: Rp is the relation for the function pgm.]
V = Cs and
P = Rs

then

{V} pgm {P} strictly = (Cs □ Cp□ Cs □ Rp|Cs □ Rs)

Proof:

{V} pgm {P} strictly

= [def. extended to relational P, see Baber, R.L., *Prak. Anw. ...*, section 3.1.6 (1), p. 38]

(□ d : d□V : (d, pgm.d)□P) □ V □ dom(pgm)

= [hypotheses of the theorem, Cp is the domain of pgm]

(□ d : d□Cs : (d, pgm.d)□Rs) □ Cs □ Cp

= [(Cp, Rp) is the LD-relation for pgm, d□Cs□Cp □ (d, pgm.d)□Rp]

(□ d : d□Cs □ (d, pgm.d)□Rp : (d, pgm.d)□Rs) □ Cs □ Cp

=

(□ d : (d, pgm.d)□Rp|Cs : (d, pgm.d)□Rs) □ Cs □ Cp

= [Rp is the relation for pgm, pgm is deterministic]

Rp|Cs □ Rs □ Cs □ Cp

= [□ is commutative, properties of sets and their intersection]

Cs □ Cp□ Cs □ Rp|Cs □ Rs

End of proof

Some implications of the above theorem:

If a program pgm satisfies a precondition—postcondition specification, then pgm restricted to Cs will satisfy the corresponding LD-relational specification. If pgm restricted to Cs satisfies an LD-relational specification, then pgm will satisfy the corresponding precondition—postcondition specification.

If a program pgm satisfies an LD-relational specification, pgm restricted to Cs will also satisfy the same LD-relational specification, and pgm will satisfy the corresponding precondition—postcondition specification. If pgm satisfies a precondition—postcondition specification, pgm restricted to Cs will, as stated above, satisfy the corresponding LD-relational specification, but pgm may or may not satisfy that LD-relational specification, depending on what pgm does when executed on initial states outside Cs.