Preserving Progress Under Program Composition

Notes on UNITY: 17-90

Jayadev Misra*
Department of Computer Sciences
The University of Texas at Austin
Austin, Texas 78712
(512) 471-9547
misra@cs.utexas.edu

7/17/90

1 Introduction

The question considered in this note is this: Under what condition is a progress property of program F preserved when F is composed with another program? For safety properties and progress properties of the form p *ensures* q, the corresponding question is answered by the union theorem. For general progress properties, however, there seems to be no easy answer; plausible rules, such as the following, are all invalid.

$$\frac{p \mapsto q \text{ in } F \text{ , } p \text{ stable in } G}{p \mapsto q \text{ in } F \,[\!]\, G}$$

$$\frac{p \mapsto q \text{ in } F \text{ , } p \mapsto q \text{ in } G}{p \mapsto q \text{ in } F \,[\!]\, G}$$

One restriction we can put on G is that it should not write into any variable that it shares with F. It is then true that $p \mapsto q$ in $F \,[\!]\, G$ if $p \mapsto q$ in F; this is, in fact, a special case of the superposition theorem [1]. However, this is a stringent restriction on G. We propose a rule whose moral is "progress is achieved when everyone pushes in the same direction." The proposed rule is obtained by simplifying and generalizing a result due to Ambuj Singh [2].

The inference rule, given below, tells us when $p \mapsto q$ in $F \,[\!]\, G$ can be established from $p \mapsto q$ in F. The condition is that both F and G should only "decrease" the values of their shared variables along a well-founded ordering. Formally, let x be the variables shared between F and G and "$<$" is a well-founded ordering relation among the values of x where $p \wedge \neg q$ holds. We assume that p names no program variables of G other than x (if it does, all such variables are treated free in interpreting $p \mapsto q$ in F; they should be renamed to avoid name clashes with G's variables). In the following, m is free.

$$\frac{p \mapsto q \text{ in } F \text{ ,} \quad p \wedge x = m \;\; unless \;\; (p \wedge x < m) \vee q \text{ in } F \,[\!]\, G}{p \mapsto q \text{ in } F \,[\!]\, G}$$

*This material is based in part upon work supported by the Texas Advanced Research Program under Grant No. 003658–065 and by the Office of Naval Research Contract N00014-90-J-1640.

To see the validity of this rule in operational terms, consider any execution of $F \,[\!]\, G$ starting from a state where p holds. If G changes the value of x infinite number of times in this execution then, from $p \wedge x = m$ *unless* $(p \wedge x < m) \vee q$, eventually q will hold (because as long as p holds the value of x decreases each time that G changes it and F does not increase x; from well-foundedness, x cannot decrease forever and hence, q will be established). If G changes x only a finite number of times and q has not been established by the time G last changes x then p holds at that point—again from the given *unless* property; G no longer interferes by changing x and hence, from $p \mapsto q$ in F, eventually q is established.

2 Proof of the Inference Rule

Consider a predicate p and a program G. The variables named in p are either (1) program variables of G, (2) bound variables or (3) free variables. Let z be the set of program variables of G named in p. Then in G, once p holds it continues to hold as long as variables in z do not change their values. That is (in the following, m is free),

Axiom A :: $p \wedge z = m$ *unless* $z \neq m$ in G

Note: The above axiom still holds if z is a superset of all program variables of G named in p. □

Let F, G share variables x, i.e., each variable in x is a program variable of both F and G. Let predicate p name only program variables of F, and free or bound variables; in particular, p does not name any program variable of G other than x.

Lemma 1: $$\frac{p \text{ } unless \text{ } q \text{ in } F}{p \wedge x = m \text{ } unless \text{ } q \vee x \neq m \text{ in } F \,[\!]\, G}$$

Proof:

p *unless* q in F	, given
$x = m$ *unless* $x \neq m$ in F	, antireflexivity of *unless*
$p \wedge x = m$ *unless* $q \vee x \neq m$ in F	, simple conjunction
$p \wedge x = m$ *unless* $x \neq m$ in G	, Axiom A
$p \wedge x = m$ *unless* $q \vee x \neq m$ in $F \,[\!]\, G$	, union theorem on the above two □

Lemma 2: $$\frac{p \text{ } ensures \text{ } q \text{ in } F}{p \wedge x = m \text{ } ensures \text{ } q \vee x \neq m \text{ in } F \,[\!]\, G}$$

Proof: Similar to that of Lemma 1. □

Lemma 3: $$\frac{p \mapsto q \text{ in } F}{p \wedge x = m \mapsto q \vee x \neq m \text{ in } F \,[\!]\, G}$$

Proof: We apply induction on the structure of the proof of $p \mapsto q$ in F.

- p *ensures* q in F: Result follows from Lemma 2.
- $p \mapsto r$ in F and $r \mapsto q$ in F:

$p \wedge x = m \mapsto r \vee x \neq m$ in $F \,[\!]\, G$	, induction hypothesis
$p \wedge x = m \mapsto (r \wedge x = m) \vee x \neq m$ in $F \,[\!]\, G$	, rewriting the rhs
$r \wedge x = m \mapsto q \vee x \neq m$ in $F \,[\!]\, G$	, induction hypothesis
$p \wedge x = m \mapsto q \vee x \neq m$ in $F \,[\!]\, G$	, cancellation on the above two

Note: The predicate r, arising in the proof of $p \mapsto q$ in F, cannot name any program variable of G other than x.

- $\langle \forall\, i \;::\; p.i \;\mapsto\; q\rangle$ in F where $p \;\equiv\; \langle \exists\, i \;::\; p.i\rangle$
  $p.i \;\wedge\; x = m \;\mapsto q \;\vee\; x \neq m$ in $F \,[\!]\, G$, induction hypothesis
  $\langle \exists\, i \;::\; p.i \;\wedge\; x = m\rangle \mapsto q \;\vee\; x \neq m$ in $F \,[\!]\, G$, disjunction
  $p \;\wedge\; x = m \;\mapsto q \;\vee\; x \neq m$ in $F \,[\!]\, G$, rewriting lhs □

Theorem: Let x be the variables shared between F, G. Let p be a predicate that names no program variable of G other than x. Let "<" be a well-founded ordering relation among the values of x where $p \wedge \neg q$ holds. Then,

$$\frac{p \;\mapsto\; q \text{ in } F\ , \quad p \;\wedge\; x = m \;\; \textit{unless} \;\; (p \;\wedge\; x < m) \;\vee\; q \text{ in } F \,[\!]\, G}{p \;\mapsto\; q \text{ in } F \,[\!]\, G}$$

Proof (due to Edgar Knapp):

$p \;\mapsto\; q$ in F , given
$p \;\wedge\; x = m \mapsto q \;\vee\; x \neq m$ in $F \,[\!]\, G$, above and Lemma 3
$p \;\wedge\; x = m$ *unless* $(p \;\wedge\; x < m) \;\vee\; q$ in $F \,[\!]\, G$, given
$p \;\wedge\; x = m \mapsto [(p \;\wedge\; x = m) \;\wedge\; (q \;\vee\; x \neq m)] \;\vee\; (p \;\wedge\; x < m) \;\vee\; q$ in $F \,[\!]\, G$
, PSP on the above two
$p \;\wedge\; x = m \mapsto (p \;\wedge\; x < m) \;\vee\; q$ in $F \,[\!]\, G$, simplifying the rhs of the above
$p \;\mapsto\; q$ in $F \,[\!]\, G$, induction on the above □

The reader should note that the following plausible rules are all invalid.

- $$\frac{p \;\textit{unless}\; q \text{ in } F\ ,\; p \;\mapsto\; q \text{ in } F \quad p \;\wedge\; x = m \;\; \textit{unless} \;\; (p \;\wedge\; x < m) \;\vee\; q \text{ in } G}{p \;\mapsto\; q \text{ in } F \,[\!]\, G}$$

- $$\frac{p \;\wedge\; x = m \;\; \textit{unless} \;\; (p \;\wedge\; x < m) \;\vee\; q \text{ in } G \quad p \;\wedge\; x = m \;\mapsto\; (p \;\wedge\; x < m) \;\vee\; q \text{ in } F}{p \;\mapsto\; q \text{ in } F \,[\!]\, G}$$

3 References

1. K. Mani Chandy and Jayadev Misra, *Parallel Program Design: A Foundation* (Section 7.3.2), Addison-Wesley, 1988.

2. Ambuj Singh, "*Leads-to* and Program Union," *Notes on UNITY; 06–89*, Austin, Texas, June 20, 1989.