Program Specifications for Program Simulations

M.M. Fokkinga and J.C.S.P. van der Woude, and J. Zwiers

Revised version, April 1995

We give a concise, structured, and calculational proof of a theorem that yields a program specification for a given, specific, program simulation.

Keywords: program correctness, programming calculi.

Introduction. Our aim is to prove a certain theorem in a concise, structured, and calculational way, in the spirit of The Calculational Method [1]. The theorem yields a specification of a 'concrete program (an implementation)' that simulates, via a given retrieve relation, a given specification of an 'abstract program'. Throughout the paper we abstract away all aspects of a program and a program specification except for the input-output; what is left over is a binary relation. Thus, formally, we are dealing with relations only.

On the basis of uninterpreted formal considerations about the relational formulation of the Hoare-style partial correctness statement, we are led to define three operations that turn out to be well-known: a *box* and *diamond* operation (from dynamic logic) that are related to the weakest precondition and the weakest liberal precondition, and a *leadsto* operation as investigated by Zwiers et al [11] (and not to be confused with the leadsto operation of Unity). We observe several Galois connections between these operations, as well as a duality, thus getting a host of useful properties for free.

The main tool in the proof of the theorem is a so-called fusion theorem, indicating under what conditions and in what way a relation can be *fused* into a leadsto operation. The importance of fusion properties in a theory dealing with compositions has been pointed out earlier in the literature, for example by Fokkinga [4].

Basic notions. We assume familiarity with the following concepts and notations about relations:

inclusion, intersection, union:	$R \subseteq S,\ R \cap S,\ R \cup S$
least, identity, greatest relation:	$\perp\!\!\!\perp,\ I,\ \top\!\!\!\top$
composition, converse:	$R \mathbin{;} S,\ R^{\cup}$
conditions (relations included in I):	p, q
domain and range:	$R^{<},\ R^{>}$.

The domain $R^{<}$ and range $R^{>}$ are conditions, that is, relations included in I. For example, $R^{<} = \{(x,x) \mid \exists y :: xRy\}$. A relation R is called *total* if $R^{<} = I$. So, the left side of R is its 'input', and the right side is its 'output'. We use negation on conditions: $\neg p = I \cap$ the complement of p. We need just one axiom involving negation and relational composition:

(1) $\quad p \mathbin{;} R \subseteq R \mathbin{;} q \quad \equiv \quad R \mathbin{;} \neg q \subseteq \neg p \mathbin{;} R$ $\quad$ NEGATION

General complementation is nowhere needed, though it may be used to prove this axiom in a few steps (which we won't do). We shall often use the principle of 'indirect equality', that is, we shall prove an equality $R{=}S$ by showing $\forall T :: T{\subseteq}R \equiv T{\subseteq}S$. All of our reasoning is valid in every (incomplete) axiomatisation of relations that we know of, see for example Backhouse et al. [2]. However, even without knowledge of such an axiomatisation, the steps are clear and convincing when interpreting the formulas.

Simulation. Let R, S, T be relations that can be composed as follows:

$$\begin{tikzcd} {} \arrow[r, "T"] & {} \\ {} \arrow[u, "R"] \arrow[r, "S"'] & {} \arrow[u, "R"'] \end{tikzcd}$$

We say that S simulates T via R whenever the diagram sub-commutes with S at the lower side and T at the upper side. There are four variants:

$$\begin{array}{lrll}
S & \text{strongly } R\text{-simulates} & T & \equiv \quad R^{\cup} \mathbin{;} S \mathbin{;} R \subseteq T \\
S & \text{up } R\text{-simulates} & T & \equiv \quad S \mathbin{;} R \subseteq R \mathbin{;} T \\
S & \text{down } R\text{-simulates} & T & \equiv \quad R^{\cup} \mathbin{;} S \subseteq T \mathbin{;} R^{\cup} \\
S & \text{weakly } R\text{-simulates} & T & \equiv \quad S \subseteq R \mathbin{;} T \mathbin{;} R^{\cup} \quad .
\end{array}$$

Here, R, S, T may be interpreted as a retrieve relation, a concrete program or implementation, and an abstract program or specification, respectively. Also, if R and T are given, S may be considered as a specification of a concrete program. *Our theorem will give a solution for S when R and a specific T are given.* Our aim is to give a concise, calculational proof at the level of relations (rather than at the point level, in predicate logic).

Simulation has been explored first by Milner [8], and was used later in a relational setting by Hoare and Sanders [7]. A short summary is given by Fokkinga [5].

Correctness. Let R be a relation (standing for a program or a specification), and p, q conditions. Then the classical Hoare-style partial correctness assertion:

for all x, y: if $p(x)$ and xRy, then $q(y)$

is expressed at the relation level by:

(2) $\qquad p \mathbin{;} R \quad \subseteq \quad R \mathbin{;} q \quad .$

Let us consider the extremal (least, greatest) solutions for p, q or R in this statement. Some of the extremal solutions are trivial, that is, they are the extremal condition or relation without restriction. This is the case for the least p, the greatest q, and the least R for which $p \mathbin{;} R \subseteq R \mathbin{;} q$, as the reader may check. The non-trivial cases are these: the greatest p, the least q, and the greatest R such that $p \mathbin{;} R \subseteq R \mathbin{;} q$. We use the following notation for them: $[R]q$ ("box"), $\langle R^{\cup}\rangle p$ ("diamond"), and $p \rightsquigarrow q$ ("leadsto"). So their characterisations read:

(3) $\qquad p \subseteq [R]q \quad \equiv \quad p \mathbin{;} R \subseteq R \mathbin{;} q \qquad$ boxCharn

(4) $\qquad \langle R^{\cup}\rangle p \subseteq q \quad \equiv \quad p \mathbin{;} R \subseteq R \mathbin{;} q \qquad$ diaCharn

(5) $\qquad R \subseteq p \rightsquigarrow q \quad \equiv \quad p \mathbin{;} R \subseteq R \mathbin{;} q \quad .$ leadsto$\textsc{Charn}$

It is left implicit that $[R]q$ and $\langle R^{\cup}\rangle p$ are conditions: included in I.

Remark on the notation. Operation $^{\cup}$ in the characterisation of diamond can be shifted to the right-hand side by substituting $R^{\cup}$ for R and using $R^{\cup\cup} = R$. The current formulation has the advantage that all three right-hand sides are the same, and consequently all three left-hand sides are equivalent. This will be exploited in observation (6) and further. However, many formulas in the sequel would have been slightly nicer, if the symbol '$^{\cup}$' is omitted altogether from the notation, thus writing, say, $\{R\}p$ instead of $\langle R^{\cup}\rangle p$. Nevertheless we keep the notation '$\langle\rangle$', possibly decorated with the awkward '$^{\cup}$', because the diamond operation itself also has nice properties, namely the interpretation below, and the duality with the box operation, (11).

Interpreted at the point level we recognise some familiar notions:

$$
\begin{array}{lcl}
[R]q\ (x) & \equiv & \forall y:\ xRy:\ q(y) \\
\langle R\rangle q\ (x) & \equiv & \exists y:\ xRy:\ q(y) \\
x\,(p \rightsquigarrow q)\,y & \equiv & p(x) \ \Rightarrow\ q(y) \quad .
\end{array}
$$

So, $[R]q$ and $\langle R\rangle q$ are related to the weakest (liberal) precondition for R with respect to q. The precise relationship depends on decisions that we don't want to go into here, concerning the way non-termination of programs is modeled (if at all) in the relational framework. (One may also notice that nowhere in this paper the relational composition is meant to model sequential program composition — although it may be so, of course.) Finally, $p \rightsquigarrow q$ is a Hoare-style specification of a (yet to be developed) program R satisfying $p \mathbin{;} R \subseteq R \mathbin{;} q$.

The theorem. Having introduced the notion of simulation and the operation leadsto, we are now able to formalise the problem statement. Let a simple specification $p \rightsquigarrow q$ of an (abstract) program be given, as well as a fixed retrieve relation R. We are looking for conditions x and y (and possibly some further requirements on p, q, R) such that:

$$x \rightsquigarrow y \quad \text{strongly } R\text{-simulates} \quad p \rightsquigarrow q \quad .$$

And similarly for up, down, and weak simulation. It will turn out that x and y can be expressed in terms of operations box and diamond (and the given p, q, R).

The proof, or rather the construction of the unknowns x and y, is structured as follows. Firstly, we observe some Galois connections and some duality among the operations box, diamond and leadsto. The duality will halve the proof obligations; the Galois connections give several useful properties for free. Secondly, we investigate the fusion properties of leadsto. These properties express $R \mathbin{;} (p \rightsquigarrow q)$ and $(p \rightsquigarrow q) \mathbin{;} R$ as a single leadsto expression and form the main tool to prove and construct simulation statements. After that, the construction of x and y is almost a triviality.

Galois connections and Duality. The characterisations (and, indeed, the introduction of the operations as *extremal solutions for certain inclusions*) cry out for looking for Galois connections.

(*Galois connection*) Let $F : A \to B$ and $g : B \to A$ be functions between complete lattices. Then (f, g) are $(\subseteq_B, \subseteq_A)$-Galois connected if: for every $a \in A$ and $b \in B$,

$$f(a) \subseteq_B b \quad \equiv \quad a \subseteq_A g(b) \quad .$$

In that case f is called the lower adjoint of g, and g the upper adjoint of f. Galois theory [2, 3, 6] tells us that f has an upper adjoint precisely when f is universally disjunctive, and, dually, g has a lower adjoint precisely when g is universally conjunctive. Moreover, for a Galois connected pair (f, g) we have, amongst others:

$g(b)$ is the greatest solution for a in $f(a) \subseteq_B b$
$f(a)$ is the least solution for b in $a \subseteq_A g(b)$
$f \circ g \circ f = f$ and $g \circ f \circ g = g$
$id_A \subseteq g \circ f$ and $f \circ g \subseteq id_B$.

From the characterisations CHARN(3)(4)(5) it is immediate that:

(6) $\quad \langle R^\cup \rangle p \subseteq q \quad \equiv \quad p \subseteq [R]q \quad \equiv \quad R \subseteq p \leadsto q \quad .$ GALOIS

Hence the following are Galois connected:

$\langle R^\cup \rangle\_$ with $[R]\_$
$\langle \_^\cup \rangle p$ with $p \leadsto \_$
$\_ \leadsto q$ with $[\_]q$ contravariantly (that is, one $\subseteq$ is replaced by $\supseteq$) ,

and therefore we have the following properties, of which only the monotonicity of leadsto and boxdia/diabox will be used in the sequel. For all p, q, R:

(7) $\quad \langle R^\cup \rangle\_$, $[R]\_$, $\langle \_^\cup \rangle p$ and $p \leadsto \_$ are monotonic MONO

(8) $\quad [\_]q$ and $\_ \leadsto q$ are antimonotonic

(9) $\quad p \subseteq [R]\langle R^\cup \rangle p$ and $\langle R^\cup \rangle [R] q \subseteq q$ boxdia/diabox

(10) $\quad \langle R^\cup \rangle [R] \langle R^\cup \rangle p = \langle R^\cup \rangle p$ and $[R]\langle R^\cup \rangle [R] q = [R]q$.

Moreover, NEGATION(1) immediately translates into another Galois connection for $\langle R^\cup \rangle\_$:

$\quad \langle R^\cup \rangle p \subseteq q$
$\equiv \quad p \,;\, R \subseteq R \,;\, q$ (by diaCHARN(4))
$\equiv \quad \neg q \,;\, R^\cup \subseteq R^\cup \,;\, \neg p$ (by NEGATION(1), converse)
$\equiv \quad \langle R \rangle \neg q \subseteq \neg p$ (by diaCHARN(4))
$\equiv \quad p \subseteq \neg \langle R \rangle \neg q$. (by NEGATION(1))

So, since $\langle R^\cup \rangle\_$ is Galois connected with $[R]\_$, it follows by uniqueness of adjoints (or by indirect equality) that:

(11) $\quad [R]q \quad = \quad \neg \langle R \rangle \neg q \quad .$ boxdiaDUAL

This duality carries over to $\rightsquigarrow$:

$$(12) \qquad (p \rightsquigarrow q)^{\cup} \quad = \quad \neg q \rightsquigarrow \neg p \quad , \qquad \text{leadsto}\textsc{Dual}$$

since:

$$\begin{aligned}
& R \subseteq p \rightsquigarrow q \\
\equiv \quad & p \subseteq [R]q && (\text{by } \textsc{Galois}^{(6)}) \\
\equiv \quad & \langle R^{\cup\cup}\rangle \neg q \subseteq \neg p && (\text{by boxdia}\textsc{Dual}^{(11)}, \textsc{Negation}^{(1)}) \\
\equiv \quad & R^{\cup} \subseteq \neg q \rightsquigarrow \neg p \quad . && (\text{by } \textsc{Galois}^{(6)})
\end{aligned}$$

We shall use these dualities in the following way.

> Let *expr* be a relational expression, and let $expr^{\circ}$ result from *expr* by subjecting it to $^{\cup}$ and substituting $\neg p, R^{\cup}$ for p, R (for some variables p, R). Clearly:
>
> $$(\, expr_0 \subseteq expr_1 \text{ is universally valid}) \quad \equiv \quad (\, expr_0^{\circ} \subseteq expr_1^{\circ} \text{ is universally valid}).$$
>
> Moreover, this remains true if $expr_0^{\circ}$ and $expr_1^{\circ}$ are simplifyed according to the rules $(R \,;\, S)^{\cup} = S^{\cup} \,;\, R^{\cup}$, $\neg\neg p = p$, $R^{\cup\cup} = R$, and the duality rules (11, 12) above.

For example, the following two statements are equivalent "by duality":

$$\begin{aligned}
&\forall p, q, R, S, T :: \qquad R \,;\, (\langle S\rangle p \rightsquigarrow [S]q) \subseteq T \\
&\forall p, q, R, S, T :: \qquad (\langle S\rangle q \rightsquigarrow [S]p) \,;\, R \subseteq T \quad .
\end{aligned}$$

Backhouse and Van der Woude [3] also consider the box operation in a relational framework. They denote $[R]q$ by $q \not/ R$ in view of the similarities of operation $\not/$ with the factor operation $/$. They have no specific notation for $\langle R\rangle p$ (neither for $p \rightsquigarrow q$), so that several dualities in their statements are not present notationally, and therefore not easily recognised. For example, compare their cancellation properties for $\not/$ [3, Table 1] with our formulation in (9, 10).

Fusion properties. In view of the theorem we are searching for, let us investigate under what conditions a relation can be fused into a leadsto.

As it turns out, the authors need the following explicit form for the leadsto operation:

$$(13) \qquad p \rightsquigarrow q \quad = \quad (\neg p \,;\, \mathbb{T}) \cup (\mathbb{T} \,;\, q) \qquad \text{leadsto}\textsc{Def}$$

It follows by indirect equality from:

$$\begin{aligned}
& R \subseteq (\neg p \,;\, \mathbb{T}) \cup (\mathbb{T} \,;\, q) \\
\equiv \quad & (p \,;\, R) \cup (\neg p \,;\, R) \subseteq (\neg p \,;\, \mathbb{T}) \cup (\mathbb{T} \,;\, q) && (\text{by } I = p \cup \neg p) \\
\equiv \quad & p \,;\, R \subseteq \mathbb{T} \,;\, q \\
\equiv \quad & p \,;\, R \subseteq R \,;\, q \\
\equiv \quad & R \subseteq p \rightsquigarrow q \quad . && (\text{by leadsto}\textsc{Charn}^{(5)})
\end{aligned}$$

Here, then, are the fusion properties of leadsto:

$$(14) \qquad R \,;\, (p \rightsquigarrow q) \quad = \quad R^{<} \,;\, ([R]p \rightsquigarrow q) \qquad \text{leadsto}\textsc{Fusion}$$

$$\begin{array}{lrcll} & & \subseteq & x \leadsto q & \text{if } x \subseteq [R]p \\ (15) & (p \leadsto q) ; R & = & (p \leadsto \langle R^{\cup} \rangle q) ; R^{>} & \qquad\qquad \text{leadsto}\textsc{Fusion} \\ & & \subseteq & p \leadsto y & \text{if } \langle R^{\cup} \rangle q \subseteq y \end{array}$$

By duality it suffices to prove (15) only; this reads:

$$\begin{array}{lcll} & & (p \leadsto q) ; R & \\ & = & ((\neg p ; \mathbb{T}) \cup (\mathbb{T} ; q)) ; R & (\text{by leadsto}\textsc{Def}^{(13)}) \\ & = & ((\neg p ; \mathbb{T}) \cup (\mathbb{T} ; (q ; R)^{>})) ; R^{>} & (\text{by distribution, domains}) \\ (*) & = & ((\neg p ; \mathbb{T}) \cup (\mathbb{T} ; \langle R^{\cup} \rangle q)) ; R^{>} & (\text{by observation below: } \langle R^{\cup} \rangle q = (q ; R)^{>}) \\ & = & (p \leadsto \langle R^{\cup} \rangle q) ; R^{>} & (\text{by leadsto}\textsc{Def}^{(13)}) \\ & \subseteq & p \leadsto \langle R^{\cup} \rangle q & (\text{by domains}) \\ & \subseteq & p \leadsto y, \quad \text{if } \langle S^{\cup} \rangle q \subseteq y \quad . & (\text{by } \textsc{Mono}^{(7)}) \end{array}$$

In step $(*)$ we have used the explicit form $\langle R^{\cup} \rangle q = (q ; R)^{>}$. This equality follows by indirect equality from:

$$\langle R^{\cup} \rangle q \subseteq r \quad \equiv \quad q ; R \subseteq R ; r \quad \equiv \quad (q ; R)^{>} \subseteq r \quad ,$$

in which the left-most equivalence is just dia$\textsc{Charn}^{(4)}$, and the right-most equivalence is verified by elementary relational calculus.

The '='-versions of the fusion laws specialise to nice equalities if R is total or surjective, respectively. By boxdia$^{(9)}$ ($p \subseteq [R]\langle R^{\cup} \rangle p$) and diabox$^{(9)}$ ($\langle R^{\cup} \rangle [R]q \subseteq q$) the '$\subseteq$'-versions specialise to:

$$\begin{array}{llcl} (16) & R ; (\langle R^{\cup} \rangle p \leadsto q) & \subseteq & p \leadsto q \\ (17) & (p \leadsto [R]q) ; R & \subseteq & p \leadsto q \quad . \end{array}$$

Actually, under some conditions on R, these inclusions are equalities as well.

Theorem.

$$\begin{array}{lllll} (18) & \langle R \rangle p \leadsto [R]q & \text{strongly } R\text{-simulates} & p \leadsto q, & \text{for all } R \\ (19) & [R]p \leadsto [R]q & \text{up } R\text{-simulates} & p \leadsto q, & \text{if } R \text{ is total} \\ (20) & \langle R \rangle p \leadsto \langle R \rangle q & \text{down } R\text{-simulates} & p \leadsto q, & \text{if } R \text{ is total} \\ (21) & [R]p \leadsto \langle R \rangle q & \text{weakly } R\text{-simulates} & p \leadsto q, & \text{if } R \text{ is total} \quad . \end{array}$$

The conditions $\langle R \rangle p$, $[R]p$, $\langle R \rangle q$, $[R]q$ in the left-hand side have been constructed in the course of the proof; for clarity we present them as if we know them already before beginning the proof.

For each claim we merely combine two fusion properties, using the premiss $R^{<} = R^{\cup >} = I$ in case R is total (and using the fact that $R^{\cup\cup} = R$). To be specific, claim (18) is immediate from (16, 17):

$$R^{\cup} ; (\langle R \rangle p \leadsto [R]q) ; R \quad \subseteq \quad p \leadsto q \quad .$$

Claim (19) is immediate from (17) and $I = R^<$ and (14), respectively:

$$([R]p \rightsquigarrow [R]q) \mathbin{;} R \quad \subseteq \quad [R]p \rightsquigarrow q \quad = \quad R^< \mathbin{;} ([R]p \rightsquigarrow q) \quad = \quad R \mathbin{;} (p \rightsquigarrow q) \quad .$$

Claim (20) is equivalent to claim (19) by duality.
Claim (21) is immediate from $I = R^< = R^{\cup>}$ and (14, 15), respectively:

$$[R]p \rightsquigarrow \langle R\rangle q \quad = \quad R^< \mathbin{;} ([R]p \rightsquigarrow \langle R\rangle q) \mathbin{;} R^{\cup>} \quad = \quad R \mathbin{;} (p \rightsquigarrow q) \mathbin{;} R^{\cup} \quad .$$

So, in this case there is an equality, whereas only an inclusion is required.

Discussion. The value of our work is not so much the theorem that we have proved, but more the way we have done it. There are two alternative ways conceivable.

The first alternative is to fully interpret the formulas and reason with intuition only. In this case (as always with intuition), the theorems are quite "obvious". But the reasoning is hardly a proof: for example, it is too easy to "see" the correctness without using the totality of the retrieve relation.

The second alternative is to translate the theorem into predicate logic. Then a proof of each claim is not that difficult, as the reader may try and check: a tedious calculation suffices, manipulating one huge formula involving six quantifications with various scopes. Machine assistance would be welcome here. In fact, Remco Hoetmer has done this exercise using PVS [9, 10]. After translating the theorem into PVS predicate logic, it took him a few seconds to prove the strong simulation statement, and a quarter of an hour for the next statement. Using the experience so obtained, the remaining statements were again straightforward. So, the machine supported proof in predicate logic took *much* less time than our proof at the level of relations! It must be said, however, that Remco was given the final form of the theorem, and that he knew the theorem to be true, and therefore he was sure that some proof strategies were going to terminate. See the appendix for an excerpt of his proof.

In contrast to these alternatives, our method has brought structure and modularity into the proof: the fusion properties. For example, it is now perfectly clear where and for what purpose the totality of the retrieve relation has been used. We have found several lemmas that are worthwhile in their own. In fact, we have *constructed* the conditions of the simulating program in the course of the proof. And last but not least, the proof consists of several concise calculations, involving short formulas only.

Acknowledgement. We thank to Remco Hoetmer for his proof using the PVS system.

References

[1] R.C. Backhouse, editor. *The Calculational Method, Special Issue of* Information Processing Letters*, 53(3):121–172.* Elsevier, North-Holland, February 1995.

[2] R.C. Backhouse et al. A relational theory of datatypes, part I and II. Presented at the STOP 1992 Summerschool on Constructive Algorithmics, Ameland, Netherlands, 1992.

[3] R.C. Backhouse and J.C.S.P. van der Woude. Demonic operators and monotype factors. *Mathematical Structures in Computer Science*, 4(3):417–433, December 1993.

[4] M.M. Fokkinga. Calculate categorically! *Formal Aspects of Computing*, 4(4):673–692, 1992.

[5] M.M. Fokkinga. A summary of refinement theory. June 1994.

[6] G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove, and D.S. Scott. *A Compendium of Continuous Lattices*. Springer Verlag, 1980.

[7] C.A.R. Hoare, J. He, and J.W. Sanders. Prespecification in data refinement. *Information Processing Letters*, 25:71–76, 1987.

[8] R. Milner. An algebraic definition of simulation between programs. Technical Report CS-205, Stanford University, February 1971.

[9] S. Owre, J.M. Rushby, and N. Shankar. PVS: A Prototype Verification System. In D. Kapur, editor, *11th International Conference on Automated Deduction (CADE)*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752. Springer Verlag, 1992.

[10] N. Shankar, S. Owre, and J.M. Rushby. The PVS Proof Checker: A reference manual. Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA, 1993.

[11] J. Zwiers and W.-P. de Roever. Predicates are predicate transformers. In *Proc. of the 8th Symp. on Distributed Computing*, pages 265–279, 1989.