 C.B. Jones

IBM LABORATORY VIENNA, Austria

YET ANOTHER PROOF OF THE CORRECTNESS OF BLOCK IMPLEMENTATION

C.B. JONES

ABSTRACT:
A number of proofs of the correctness of implementations for the "Block Concept" have been given. These proofs have been based on a definition using an abstract machine. This note attempts to repeat the exercise with an alternative definition. The relative merits of the approaches are reviewed.

LN 25.3.075
24 August 1970

INTRODUCTION

This note is an attempt to indicate how proofs of correctness of "block implementations" might be simplified in contrast to /2/, /1/ etc. The approach is to base the proof on a different style of definition: instead of having a base abstract interpreter machine, an attempt is made to give properties required of a model in terms of an equivalence relation over occurences of identifiers in blocks.

The current form of the note relies heavily on /1/ with which the reader is assumed to be familiar. The two methods which are proved to be correct models of the definition are basically the defining model and mechanism 1 of /1/ except that call by reference has been included. This choice of mechanisms will facilitate the more complete discussion which is reserved for a later section, which will also review to what extent the given definition can be considered to be "machine-free".

NOTATION

The notation of /1/ will be used throughout the sequel. Certain formulas should be compared or contrasted to those of /1/ and in such cases the appropriate reference is given on the right.

Equivalence relations are used below and the relevant facts are now presented:

$\mathscr{R}^S$ is said to be an equivalence relation on a set S if it is symmetric, reflexive and transitive over that set. (The name of the set is omitted when there is no danger of confusion.)

$\mathscr{R}^S$ partitions S into disjoint subsets or cosets such that

$$\alpha \mathscr{R}^S \underset{\text{Df}}{=} \{\beta \mid \alpha \mathscr{R} \beta \wedge \beta \in S\} \qquad \text{for } \alpha \in S$$

$\mathscr{R}^{S \cup T}$ is an extension of $\mathscr{R}^S$ providing

$$\alpha, \beta \in S \quad \supset \quad (\alpha \mathscr{R}^{S \cup T} \beta \equiv \alpha \mathscr{R}^S \beta)$$

The following ways of relating elements, γ, of T are examples of how the extension can be defined to satisfy the above

i) for exactly one $\alpha, \alpha \in S$, specify $\gamma \mathscr{R}^{S \cup T} \alpha$

ii) specify $\gamma \mathscr{R}^{S \cup T} = \{\gamma\}$ (no element except γ is in the coset of γ)

DEFINITION

A "reference" can occur in an active block/procedure and is represented by an identifier, activation pair.

References are related using:

a) a set of all possible references
b) an equivalence relation over the set a)

for each active (dynamic) occurence of a block/procedure. The effect of blocks/procedures on relations between identifiers can be expressed by showing how new sets and relations are formed.

basic sets:

D1 ID identifiers
D2 PT activation markers
D3 $REF \subseteq ID \times PT$
the interpretation of $(id,i) \in REF$ is id is known in activation i.

the following abbreviation is used:

$$ID^{P,REF} \underset{Df}{=} \{id \mid (id,p) \in REF\}$$

relation:

D4 $\mathcal{R}^{REF}$ is an equivalence relation;
the interpretation of $r1 \mathcal{R}^{REF} r2$, for $r1, r2 \in REF$, is r1 and r2 refer to the same entity.

values:

D5 a value is an object associated with a coset of $\mathcal{R}^{REF}$, and, if a procedure introduced in activation r, ID^r can be determined.

initial:

I1 $REF^1 = \{\}$

block: suppose the block to be interpreted is encountered in activation p.

let: REF, $\mathcal{R}^{REF}$ be the set and relation at activation p

D be the set of names declared in the block

B1 choose $q \in PT$ such that $\neg(\exists r)(r \in REF \wedge 2nd(r) = q)$

then the block is interpreted with:

B2 $ID^{q} = ID^{p,REF} \cup D$

B3 $REF' = REF \cup \{(id,q) \mid id \in ID^{q}\}$

extend $\mathcal{R}^{REF}$ to $\mathcal{R}^{REF'}$ as follows:

B4 $\alpha, \beta \in REF \supset (\alpha \, \mathcal{R}^{REF'} \beta \equiv \alpha \, \mathcal{R}^{REF} \beta)$

B5 $id \notin D \supset ((id,q) \, \mathcal{R}^{REF'} \, (id,p))$

B6 $id \in D \supset ((id,q) \, \mathcal{R}^{REF'} = \{(id,q)\})$

B7 execution of the nested block has no influence on REF or $\mathcal{R}$ of activation p

B8 for $id \in D$: value associated with $(id,q)\,\mathcal{R}^{REF'}$ is introduced in q.

procedure: suppose the call is encountered in activation p, and the procedure invoked was introduced in activation r.

let: REF, $\mathcal{R}^{REF}$ be the set and relation at activation p

P be the names of the parameter list

A be the names of the argument list

(P and A: used both as sets and lists)

P0 if id is the name of the procedure called, then:
$(id,p) \in REF$
value associated with $(id,p)\,\mathcal{R}^{REF}$ is the denotation of the invoked procedure.

P1 choose $q \in PT$ such that $\neg(\exists r)(r \in REF \wedge 2nd(r) = q)$

then the procedure is interpreted with:

P2 $ID^{q} = ID^{r,REF} \cup P$

P3 $REF' = REF \cup \{(id,q) \mid id \in ID^{q}\}$

extend $\mathcal{R}^{REF}$ to $\mathcal{R}^{REF'}$ as follows:

P4 $\alpha, \beta \in REF \supset (\alpha\ \mathcal{R}^{REF'} \beta \equiv \alpha\ \mathcal{R}^{REF} \beta)$

P5 $id \notin P \supset ((id,q)\,\mathcal{R}^{REF'}(id,r))$

P6 $id = P_i \supset ((P_i,q)\,\mathcal{R}^{REF'}(A_i,p))$

P7 execution of the called procedure has no influence on REF, $\mathcal{R}$ of activation p.

To show that a model satisfies these properties, it is necessary to define D1 - D5 in terms of its state and show that these realizations satisfy I1; B1 - B8 and P0 - P7.

<u>First Model</u>

State:

s1 $is\text{-}state(\xi) \supset (is\text{-}dump(s\text{-}d(\xi)) \wedge is\text{-}dendir(s\text{-}dn(\xi)) \wedge s\text{-}U(\xi) \subseteq UN)$ (S1)

s2 $is\text{-}dump(d) \supset is\text{-}de(d_i)$ for $1 \leq i \leq l(d)$ (S2)

s3 $is\text{-}de(de) \supset (s\text{-}tp(de) \in TP \wedge is\text{-}env(s\text{-}e(de)))$ (S3)

s4 $is\text{-}env(e) \supset (D(e) \subseteq ID \wedge R(e) \subseteq UN)$ (S4)

s5 $is\text{-}dendir(dn) \supset (D(dn) \subseteq UN \wedge R(dn) \subseteq DEN)$ (S5)

s6 $is\text{-}proc\text{-}den(den) \supset (den \in DEN \wedge s\text{-}tp(den) \in TP \wedge is\text{-}env(s\text{-}e(den)))$ (S6)

State transitions:

Interpretation of a program begins with init

Interpretation of a block consists of term$\circ \ldots \circ$bloc

and of a procedure term$\circ \ldots \circ$proc

Only elements of a computation formed by init, bloc, proc or term are considered below.

init: $\Longrightarrow \xi^1$

i1 $l(d^1) = 1$ (T1)

i2 $e^1 = \Omega$ (T3)

i3 $dn^1 = \Omega$ (T6)

i4 $U^1 = \{\}$ (T7)

bloc : $\xi \Longrightarrow \xi'$

b1	$D(eo\text{-}b) = D$	(T8)
b2	$R(eo\text{-}b) \cap U = \{\}$	(T9)
b3	$id1(eo\text{-}b) = id2(eo\text{-}b) \neq \Omega \supset id1 = id2$	
b4	$rest(d') = d$	(T10)
b5	for $i \leq l(d): u \in R(s\text{-}e(d_i)) \supset u(dn') = u(dn)$	(T13)
b6	$U' = R(eo\text{-}b) \cup U$	(T14)
b7	$e' = update(e, eo\text{-}b)$	(T17)
b8	for $u \in R(eo\text{-}b) \wedge is\text{-}proc\text{-}den(u(dn'))$: $s\text{-}e(u(dn')) = e'$	(T19b)

proc: $\xi \Longrightarrow \xi'$

p1	$id\text{-}p \in D(e)$	(T20)
p2	$u\text{-}p = id\text{-}p(e)$ $e\text{-}p = s\text{-}e(u\text{-}p(dn))$	(T22)
p3	$is\text{-}proc\text{-}den(u\text{-}p(dn))$	(T21)
p4	$D(eo\text{-}b) = P$ [1)]	(T8)
p5	for $1 \leq i \leq l(P): P_i(eo\text{-}b) = A_i(e)$	
p6	$rest(d') = d$	(T10)
p7	for $i \leq l(d): u \in R(s\text{-}e(d_i)) \supset u(dn') = u(dn)$	(T13)
p8	$U' = U$	(T14)
p9	$e' = update(e\text{-}p, eo\text{-}b)$	(T23)

term: $\xi \Longrightarrow \xi'$

t1	$d' = rest(d)$	(T26)
t2	for $i \leq l(d'): u \in R(s\text{-}e(d_i)) \supset u(dn') = u(dn)$	(T27)
t3	$U' = U$	(T28)

---

1) Strictly, the model should exhibit a way of storing P with the denotation, the omission derives from the use of /1/ as base.

Further, let (for some state ξ):

m1 $ID^i = D(e_i)$

m2 $PT = N$ indexes to the dump

m3 $REF = \{(id,i) \mid id \in ID^i \wedge 0 \le i \le l(d)\}$

m4 $(id1,i)\mathcal{R}(id2,j) \underset{Df}{=} id1(e_i) \neq \Omega \wedge id1(e_i) = id2(e_j)$

Two lemmas can be proved about the model:

l1 for ξ: $u \in U \supset \neg(\exists\, id,j)(j \le l(d) \wedge u = id(e_j))$

A proof by induction (similar to L5 of /1/) of a strengthened proposition can be given using:

basis	i4	
induction	bloc	b4, b6, b7, b8
	proc	p5, p6, p8, p9, PO, m3
	term	t1, t3

l2 for : for a procedure introduced in activation r such that id(e) gives its unique name:

$$s\text{-}e(id(e)(dn)) = e_r$$

A proof by induction (similar to L6 of /1/, but made simpler by the differences b5 to T13 etc.) can be given using:

basis	i2	
induction	bloc	b4, b5, b7, b8
	proc	p5, p6, p7, p9, PO, m3
	term	t1, t2

<u>Theorem:</u> The above model satisfies the properties required by the definition.

Proof:

D1, D2, D3:	The basic sets are defined	m1,s4,m2,m3
D4:	$\mathcal{R}$ is an equivalence relation over the given sets	m1,m3,m4
D5:	values are associated with the cosets of $\mathcal{R}$ via	m4,s5
	unique names, and ID^r for procedures can be found.	P0,12
I1:	$ID^1 = \{\}$	m1,i2
	$\therefore\ REF^1 = \{\}$	m3,i1
B1:	$(id,i) \in REF \supset i \leq l(d)$	m3
	$\therefore\ q = l(d')$ fulfils the conditions	b4
B2:	$ID^q = D(e')$	B1,m1
	$= D(e_p) \cup D(eo\text{-}b)$	b7
	$= ID^{p,REF} \cup D$	m1,b1
B3:	$REF' = REF \cup \{(id,q) \mid id \in ID^q\}$	m3,b4,B2
B4:	$\alpha,\beta \in REF \supset 2nd(\alpha) \leq l(d) \wedge 2nd(\beta) \leq l(d)$	m3
	$\therefore\ \alpha \mathcal{R}^{REF'} \beta \equiv \alpha \mathcal{R}^{REF} \beta$	m4,b4,B3
B5:	$id \notin D \supset id(e') = id(e)$	b7,b1
	$\therefore\ (id,q) \mathcal{R}^{REF'} (id,p)$	B1,b4,m4
B6:	$id \in D \supset id(e') = id(eo\text{-}b)$	b7,b1
	$id(e') \notin U$	b2
	$\neg(\exists\, id,j)(j \leq l(d) \wedge id(e') = id(e_j))$	l1
	$\therefore\ (id,q) \mathcal{R}^{REF'} = \{(id,q)\}$	B3,m4,b3
B7:	term∘bloc is an identity with respect to:	
	REF	m3,b4,t1
	and thus $\mathcal{R}$	m4,B4,B5,B6

P1: $(id,i) \in REF \supset i \leq l(d)$ m3

$\therefore\; q = l(d')$ fulfils the condition p6

P2: $ID^{q} = D(e')$ B1,m1

$= D(e_r) \cup D(eo\text{-}b)$ P0,12,p9,p2

$= ID^{r,REF} \cup P$ m1,p6,p4

P3: $REF' = REF \cup \{(id,q) \mid id \in ID^{q}\}$ m3,p6,P2

P4: $\alpha, \beta\ REF \supset 2nd(\alpha) \leq l(d) \wedge 2nd(\beta) \leq l(d)$ m3

$\therefore\; \alpha \mathcal{R}^{REF'} \beta \equiv \alpha \mathcal{R}^{REF} \beta$ p6,m4

P5: $id \notin P \supset id(e'_q) = id(e_r)$ p9,p4,P0,12,p2

$\therefore\; (id,q) \mathcal{R}^{REF'} (id,r)$ P1,p6,m4

P6: $id \in P \supset P_i(e'_q) = P_i(eo\text{-}b)$ p9,p4

$= A_i(e)$ p5

$\therefore\; (P_i,q) \mathcal{R}^{REF'} (A_i,p)$ p6,m4

P7: term∘proc is an identity with respect to:

REF m3,p6,t1

and thus $\mathcal{R}$ m4,P4,P5,P6

Second Model:

State:

s1 $\text{is-state}(\xi) \supset \text{is-dump}(\text{s-d}(\xi))$ (S1)

s2 $\text{is-dump}(d) \supset \text{is-de}(d_i) \quad \text{for } 1 \leq i \leq l(d)$ (S2)

s3 $\text{is-de}(de) \supset (\text{s-tp}(de) \in TP \wedge$
$\text{is-dendir}(\text{s-dn}(de)) \wedge$
$\text{s-epa}(de) \in N)$ (S3)

s4 $\text{is-dendir}(dn) \supset (D(dn) \subseteq ID \wedge$
$R(dn) \subseteq DEN)$ (S5)

s5 $\text{is-proc-den}(den) \supset (den \in DEN \wedge$
$\text{s-tp}(den) \in TP \wedge$
$\text{s-epa}(den) \in N)$ (S6)

s6 $\text{is-parm-den}(den) \supset (den \in DEN \wedge$
$\text{s-id}(den) \in ID)$

State transitions:

init: $\Longrightarrow \xi^1$

i1 $l(d^1) = 1$ (T1)

i2 $dn^1 = \Omega$ (T4)

i3 $epa^1 = 0$ (T5)

bloc: $\xi \Longrightarrow \xi'$

b1 $D(dn') = D$ (T8)

b2 $\neg(\exists dn)(dn \in R(dn') \wedge \text{is-parm-den}(dn))$

b3 $\text{rest}(d') = d$ (T10)

b4 $epa' = l(d)$ (T18)

b5 for $\text{is-proc-den}(\text{id}(dn'))$: (T19)
$\text{s-epa}(\text{id}(dn')) = l(d)$

proc: $\xi \Longrightarrow \xi'$

p1 is-proc-den(find-d(id-p,d)) (T21)

p2 epa-p = s-epa(find-d(id-p,d)) (T22)

p3 D(dn') = P (T8)

p4 for $1 \le i \le l(P)$:s-id(P_i(dn')) = A_i

p5 rest(d') = d (T10)

p6 epa' = epa-p (T24)

term: $\xi \Longrightarrow \xi'$

t1 d' = rest(d) (T26)

Further, let (for some state ξ):

m0 find(id,d) = l(d) = 1 $\longrightarrow \Omega$
is-parm-den(id(dn)) $\longrightarrow$ find(id(dn),rest(d))
id $\in$ D(dn) $\longrightarrow$ id(dn)
T $\longrightarrow$ find(id,rest(d,epa))

where: dn = s-dn(top(d))
epa = s-epa(top(d))

To simplify the equivalence relation m4,find-p is used which differs from find only in that the third case distinction yields the pair (id,l(d)).

m1 ID^i = {id | find(id,rest(d,i)) $\neq \Omega$}

m2 PT = N

m3 REF = {(id,j) | id $\in ID^j \wedge 0 \le j \le l(d)$}

m4 (id1,i)$\mathcal{R}$(id2,j) $\underset{Df}{=}$ find-p(id1,rest(d,i)) $\neq \Omega \wedge$
find-p(id1,rest(d,i)) = find-p(id2,rest(d,j))

The following lemma can be proved about the model:

l1 for ξ: for a procedure introduced in activation r such that find(id,d) gives its denotation:

s-epa(find(id,d)) = r

A proof by induction can be given:

basis	i1,m0	
induction	bloc	b3,b4,b5,m0
	proc	p2,p3,p4,p5,p6,P0,m3
	term	t1

Theorem: The above model satisfies the properties required by the definition.

Proof:

D1,D2,D3:	The basic sets are defined	m1,m0,s4,m2,m3
D4:	$\mathcal{R}$ is an equivalence relation over the given sets	m1,m3,m4
D5:	values are associated with the cosets of $\mathcal{R}$	m4,m0,s4
	via find, and ID^r for procedures can be found.	P0,l1
I1:	$ID^1 = \{\}$	m1,m0,i1
	$\therefore REF^1 = \{\}$	m3,i1

B1: $(id,i) \in REF \supset i \leq l(d)$ m3

$\therefore$ $q = l(d')$ is satisfactory b3

B2: $ID^q = \{id \mid find(id,d') \neq \Omega\}$ B1,m1

$= \{id \mid find(id,rest(d)) \neq \Omega\} \cup D(dn')$ m0,b2,b4

$= ID^{p,REF} \cup D$ m1,b3,b1

B3: $REF' = REF \cup \{(id,q) \mid id \in ID^q\}$ m3,B2,b3

B4: $\alpha, \beta\ REF \supset 2nd(\alpha) \leq l(d) \wedge 2nd(\beta) \leq l(d)$ m3

$\therefore \alpha \mathcal{R}^{REF'}\beta \equiv \alpha \mathcal{R}^{REF}\beta$ B3,m4,b3

B5: $id \notin ID \supset find\text{-}p(id,d') = find\text{-}p(id,rest(d'))$ m0,b1,b4

$\therefore (id,q)\ \mathcal{R}^{REF'}(id,p)$ B1,b3,m4

B6: $id \in D \supset find\text{-}p(id,d') = (id,q)$ m0,b1,b2

$(id,i) \in REF \supset i \neq q$ B1

$\therefore (id,q)\ \mathcal{R}^{REF'} = \{(id,q)\}$ B3,m4

B7 term$\circ$bloc is an identity with respect to:

REF m3,b3,t1

and thus $\mathcal{R}$ m4,B4,B5,B6

P1: $(id,i) \in REF \supset i \le l(d)$ m3

$\therefore q = l(d')$ is satisfactory p5

P2: $ID^{q} = \{id \mid find(id,d') \neq \Omega\}$ B1,m1

$= \{id \mid find(id,rest(d,r)) \neq \Omega\} \cup D(dn')$ m0,P0,m3,l1,m1,p6

$= ID^{r,REF} \cup P$ p5,m1,p3

P3: $REF' = REF \cup \{(id,q) \mid id \in ID^{q}\}$ P2,m3,p5

P4: α,β REF $\supset 2nd(\alpha) \le l(d) \wedge 2nd(\beta) \le p$ m3

$\therefore \alpha \mathscr{R}^{REF'} \beta \equiv \alpha \mathscr{R}^{REF} \beta$ P3,m4

P5: $id \notin P \supset find\text{-}p(id,d') = find\text{-}p(id,rest(d',r))$ p3,m0,p6,p2,P0,l1

$\therefore (id,q) \mathscr{R}^{REF'} (id,r)$ B1,m4

P6: $id \in P \supset find\text{-}p(P_i,d') = find\text{-}p(A_i,rest(d))$ p3,m0,p4

$\therefore (P_i,q) \mathscr{R}^{REF'} (A_i,p)$ m4

P7: term$\circ$proc is an identity with respect to

REF m3,p5,t1

and thus $\mathscr{R}$ m4,P4,P5,P6

The remaining steps of the proof to obtain the same result as in /1/ consist of:

show $find(id,d) = id(s\text{-}eo(d_{index(s(id,d),d)}))$ by R.I.

then (L9), Theorem II, (L10), (L11), Theorem III

Notice that the transition made above, of incorporating the denotations in the stack was <u>not</u> formally justified in /1/.

Discussion

The previous proofs of block implementations (e.g. /1/ and /2/) have taken as the definition an abstract machine giving a, hopefully simple, model. The disadvantages of this approach can be seen:

(a) It is not clear where the division between the essential and inessential properties of the defining model lays. It is possible that this difficulty could be minimized by supplying additional notes.

(b) Proving the correctness (i.e. equivalence) of an implementation often requires that difficult lemmas on the base model are established. Notice that Lemmas 5 and 6 of /1/ occur above in the proof of the base model not in the proof of the implementation. It is possible that this difficulty could be overcome by proving a suitable set of lemmas about the base model and incorporating them with the definition.

(c) If a convenient common range for the results of the definition and implementation functions is not available, the equivalence may be difficult to state. For example, a formal proof of the inclusion of the denotations in the stack in /1/ would not have had the set of unique names as a common range.

This paper is an attempt to go beyond noting what is important in, or what lemmas can be deduced from, the defining model: facts like these replace the model as a definition. Naturally, it is not possible to simply write down these defining properties. Not only is it an easier task to formulate one model but the study of a range of models is the key to finding the essential properites. (In fact the idea to use equivalence relations came to the author when trying to formulate a correctness condition between the first two models of /2/.)

Although the above definition fulfils the required role for a number of models, it is certainly not general enough. In particular, it would be necessary to change the equivalence relation to one references, as in the text, in order to cover the copy rule. At least for this reason the above definition cannot be claimed to be machine-free.

As well as being easier to formulate, and possibly to read, definitions by models have the advantage that when an implementation uses precisely the same algorithm as the definition for some, or all, sub-problems, these points can be ignored by the proof (this is done at several points in /1/). However, the author suspects that to obtain, in general, more straightforward proofs of implementations, the more promising approach is to develop properties which models could be shown to fulfil. To substantiate this suspicion other areas of languages must be investigated.

REFERENCES

/1/ JONES,C.B., LUCAS,P.: Proving Correctness of Implementation Techniques.-
IBM Lab Vienna, Techn.Report TR 25.110, August 1970;
also to be published in Springer Verlag lecture note series.

/2/ HENHAPL,W., JONES,C.B.: The Block Concept and Some Possible Implementations.-
IBM Lab Vienna, Techn.Report TR 25.104, April 1970.

/3/ LANDIN,P.: A Correspondence between Algol 60 and Church's Lambda-Notation.- (Part 1)
CACM 8 (1956), No.2, p.89.

ACKNOWLEDGEMENT

The author is grateful to W.Henhapl and P.Lucas for stimulating discussions. The latter has also pointed out that P.Landin in /3/ uses an equivalence relation to informally define parts of his Sharing Machine for interpretation of IAEs.