stichting
mathematisch
centrum

MC

AFDELING INFORMATICA IW 39/75 DECEMBER

L. AMMERAAL

HOW PROGRAM STATEMENTS TRANSFORM PREDICATES

Prepublication

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

AMS(MOS) subject classification scheme (1970): 68A05

ACM-Computing Reviews-categories: 4.20, 5.24

How program statements transform predicates *)

by

L. Ammeraal
Mathematical Centre, Amsterdam

ABSTRACT

This paper deals with relationships between conditions that hold before the initiation of a statement and on its completion. Statements are semantically defined by statement functions which are transformations in the state space. They induce predicate transformers which map state space subsets to such subsets. The predicate transformers and their inverses are explicitly given for some well-known constructs including the conditional statement and the while statement. A number of examples illustrate how predicate transformers can be used.

KEY WORDS & PHRASES: *program semantics, correctness proofs, program correctness, predicate transformers*

---

*) This paper is not for review; it is meant for publication elsewhere.

INTRODUCTION

Sequences of the form

$$\{X\}\ S\ \{Y\}$$

are frequently found in papers on program correctness. Usually X and Y are assertions or conditions which hold before initiation of a statement S and on its completion, respectively. Since X and Y are comments, they are not very precisely defined and anything that improves program readability will do. However, X, Y and S are not mutually independent. It would be nice if two of them determined the third, like the sides of a rectangular triangle. Unfortunately we have not yet discovered a relationship between X,Y and S as elegant as Pythagoras' theorem. In cases like

$$\{x > 0\}\ x := x + 1\ \{x > 1\}$$

this relationship is intuitively clear without precise definitions or any other mathematical equipment, but in general it is not evident how one unknown in the triple (X,S,Y) can be derived from both other items. It is not difficult to realize that

$$\{X\}\ \underline{\text{while}}\ B\ \underline{\text{do}}\ S\ \underline{\text{od}}\ \{X \wedge \neg B\}$$

is a consequence of

$$\{X \wedge B\}\ S\ \{X\},$$

but it is less well-known how Y in

$$\{X\}\ \underline{\text{while}}\ B\ \underline{\text{do}}\ S\ \underline{\text{od}}\ \{Y\}$$

can be expressed as a function of X if Y' is given as a function of X' in

$$\{X'\}\ S\ \{Y'\}.$$

Such functions and their inverses are discussed in this paper. Because functions have unique values we will only accept the strongest Y or X that fol-

lows from a given X or Y. If we would tolerate information to be lost then, e.g., 1 + 1 = 2 could serve as a Y for any X and S.

In this paper we will avail ourselves of the conventional notation of elementary set theory. This has two pleasant aspects. First, anyone who is not altogether unfamiliar with modern mathematics understands it. Secondly, useful theorems from standard text-books can immediately be applied.

STATES AND PREDICATES

There are two distinct ways of writing things down, viz. either in mathematical or in symbolic notation. E.g. x := x + 1 is in symbolic notation. In mathematics we would write something like $x' = x + 1$. Objects in symbolic notation can be discussed with the aid of mathematical notation. In the program

(1)
```
begin real x, y; int i, j; bool b;
      S1 ;...; Sm
end,
```

where $S_1,\ldots,S_m$ stand for statements, the symbols x, y, i, j, b are usually called "variables". This is not a definition. The term "variable" is a source of troubles when we use both notations at the same time. We will therefore neither define this term nor use it. This is possible if we replace program (1) by

(2)
```
begin real x1, x2; int x3, x4; bool x5;
      S1;...;Sm
end.
```

If R is the set of real numbers and Z is the set of integers, we will say that the set

$$V = R \times R \times Z \times Z \times \{\underline{\text{true}}, \underline{\text{false}}\}$$

is the state space of program (2). It is the set of all 5-tuples $(\xi_1,\xi_2,\xi_3,\xi_4,\xi_5)$, where

$\xi_1, \xi_2 \in R$

$\xi_3, \xi_4 \in Z$

$\xi_5 \in \{\underline{true}, \underline{false}\}$.

Greek letters ξ_i were written here to emphasize that mathematical notation was used. When this is clear from the context we can safely use x_i instead of ξ_i. We will do so and keep in mind that such x_i is just an element of some set, say R. The resemblance of x_3 in $x_3 \in Z$ and in program (2) is based on pure accident. The example leads to the following definitions.

A <u>state space</u> is a Cartesian product

$$V = V_1 \times V_2 \times \ldots \times V_n.$$

The sets $V_1, \ldots, V_n$ are sometimes called <u>types</u>. The elements $\underline{x} = (x_1, x_2, \ldots, x_n)$ of V are called <u>states</u>.

If now, e.g., the statement

(3) $\quad x_3 := 50$

is chosen for S_1 in program (2), its effect (semantics) can be described in terms of the state space: any state $(x_1, x_2, x_3, x_4, x_5)$ is mapped to the state $(x_1, x_2, 50, x_4, x_5)$.

We will write this as

$$\forall\ \underline{x}=(x_1, x_2, x_3, x_4, x_5) \in V : f(\underline{x}) = (x_1, x_2, 50, x_4, x_5)$$

or simply as

(4) $\quad f(x_1, x_2, x_3, x_4, x_5) = (x_1, x_2, 50, x_4, x_5)$,

when it is clear that f is defined for all $\underline{x} \in V$.

In the context of program (2) we regard (3) just as a symbolic notation of (4). The subset

$$Y = \{\underline{x} \in V \mid \underline{x} = (x_1, x_2, 50, x_4, x_5)\}$$

of V is the mathematical equivalent of the symbolically written condition $x_3 = 50$ that could be inserted as a comment after S_1 in program (2). Thus subsets of the state space V are equivalent to "restrictions" or "conditions" imposed on V. We will use the term predicate for a subset of V when we think of the corresponding condition at the same time. We should be aware of the following correspondence between conditions (Boolean expressions) and set expressions.

Boolean expression	set expression
$A \vee B$	$A \cup B$
$A \wedge B$	$A \cap B$
$\neg A$	$\bar{A}$ (=V-A;A's complement)
$A \Rightarrow B$	$A \subset B$
true	V
false	$\emptyset$

STATEMENT FUNCTIONS AND PREDICATE TRANSFORMERS

Program statements are strings of symbols composed according to syntactic rules. Thus x_3 := 50 and x_3 := 5 * 10 are different statements. Their effects on the state space, however, are the same. We define:

The statement function of a statement S is a function

$f: W \to V$,

defined on some domain $W \subset V$.

EXAMPLE 1. Let $V = R^2$, i.e. the program has the form

```
begin real x1,x2;
      ...
end.
```

We consider the statement

S: x_1 := sqrt(x_1+x_2).

Then

$$f(x_1,x_2) = (\sqrt{x_1+x_2},\ x_2)$$

Here f is only defined on the domain

$$W = \{(x_1,x_2) \in V \mid x_1 + x_2 \geq 0\}$$

We are interested in mappings not only from states to states but also from predicates to predicates. Predicates are subsets of the state space V. The set of all subsets of a set A is called the *power set* $P(A)$ of A. As before W denotes the domain of the statement function f. Then f induces a function from $P(W)$ to $P(V)$. This function is also written as f and is given by

$$f(X) = \{f(\underline{x}) \in V \mid \underline{x} \in X\} \quad \text{for all } X \subset W.$$

This new function f will be called *predicate transformer*.

EXAMPLE 2.

$$V = R^2$$

$$X = \{(x_1,x_2) \in V \mid x_1 > 0\}$$

$$S:\ x_1 := x_1 + 1$$

Then the statement function and the predicate transformer are given by

$$f(x_1,x_2) = (x_1 + 1,\ x_2)$$

$$f(X) = \{(x_1,x_2) \in V \mid x_1 > 1\}.$$

In the next section some general rules to find f(X) will be given. In this and some other examples x_2 seems superfluous. Its only purpose is to prevent us from identifying state spaces with the set of real numbers, which would be highly unrealistic.

Set theory (cf.[7]) provides many interesting properties of mappings which we may use for our purposes by virtue of our definition of predicate transformers. For a given statement function f with domain W we define the *inverse predicate transformer* f^{-1} as a function from $P(V)$ to $P(W)$ given by

$$f^{-1}(Y) = \{\underline{x} \in W \mid f(\underline{x}) \in Y\} \quad \text{for any } Y \subset V.$$

Then for any X, X_1, $X_2 \subset W$ and Y, Y_1, $Y_2 \subset V$ we mention the following properties, which are easy to prove.

(5) $f(X_1 \cup X_2) = f(X_1) \cup f(X_2)$

(6) $f(X_1 \cap X_2) \subset f(X_1) \cup f(X_2)$

(7) $X_1 \subset X_2$ implies $f(X_1) \subset f(X_2)$

(8) $f^{-1}(Y_1 \cup Y_2) = f^{-1}(Y_1) \cup f^{-1}(Y_2)$

(9) $f^{-1}(Y_1 \cap Y_2) = f^{-1}(Y_1) \cap f^{-1}(Y_2)$

(10) $Y_1 \subset Y_2$ implies $f^{-1}(Y_1) \subset f^{-1}(Y_2)$

(11) $f(X) = \emptyset$ if and only if $X = \emptyset$

(12) $f(f^{-1}(Y)) \subset Y$

(13) $X \subset f^{-1}(f(X))$

(14) $f^{-1}(V-Y) = W - f^{-1}(Y)$

If the statement function f maps W <u>onto</u> V, i.e. $f(W) = V$, then

(15) $f(f^{-1}(Y)) = Y$

If f is one-to-one, i.e. $\underline{x}_1 = x_2$ whenever $f(\underline{x}_1) = f(\underline{x}_2)$, then

(16) $f^{-1}(f(X)) = X$

(17) $f(X_1 \cap X_2) = f(X_1) \cap f(X_2)$

Often $W = V$, as in the following examples.

<u>EXAMPLE 3.</u>

$V = R^2$

$X = \{(x_1, x_2) \in V \mid x_1 > 3\}$

S: $x_1 := x_1 * x_1$

Then

$$f(x_1,x_2) = (x_1^2,x_2)$$

$$f(X) = \{(x_1,x_2) \in V \mid x_1 > 9\}$$

$$f^{-1}(f(X)) = \{(x_1,x_2) \in V \mid x_1 < -3 \vee x_1 > +3\}$$

This example illustrates (13). Since f is not one-to-one, (16) does not apply here.

EXAMPLE 4.

$$V = R^2$$

$$Y = \{(x_1,x_2) \in V \mid x_1 > -2\}$$

$$S: \quad x_1 := x_1 * x_1$$

Then

$$f^{-1}(Y) = V$$

$$f(f^{-1}(Y)) = \{(x_1,x_2) \in V \mid x_1 \geq 0\}$$

Here (12) applies, but (15) does not, because f does not map V onto V.

PREDICATE TRANSFORMERS FOR SOME STATEMENTS

For each program statement the statement function is its semantic definition. The predicate transformer and its inverse are then determined. In other words, $f(\underline{x})$ is given by definition; $f(X)$ and $f^{-1}(Y)$ by theorems. They will be presented in this section for

a. the dummy statement
b. the assignment statement
c. the compound statement
d. the conditional statement
e. the while statement.

The given expressions for $f(X)$ and $f^{-1}(Y)$ are rather evident. We will omit their proofs but insert some elementary examples.

a. The dummy statement.

The semantics of the dummy statement are defined by $f(\underline{x}) = \underline{x}$ for all

$x \in V$. We denote this statement by the symbol skip. Evidently, $f(X) = X$ and $f^{-1}(X) = X$ for all $X \subset V$.

b. The assignment statement

Let the state space be $V = V_1 \times V_2 \times \ldots \times V_n$, and let for some i $(1 \leq i \leq n)$ a function

$$\phi: W \to V_i \quad (W \subset V)$$

be given. Then the statement function f with domain W and given by

$$\begin{cases} f(\underline{x}) = (x_1,\ldots,x_{i-1}, \phi(\underline{x}), x_{i+1},\ldots,x_n) \\ \underline{x} = (x_1,\ldots,x_i,\ldots,x_n) \end{cases}$$

defines the semantics of the assignment statement symbolically denoted by

$$x_i := \phi(\underline{x}).$$

It then follows that, for all $X \subset W$ and $Y \subset V$

$$f(X) = \{(x_1,\ldots,x_i,\ldots,x_n) \in V \mid \exists x_i^\circ : \underline{x}^\circ = (x_1,\ldots,x_{i-1}, x_i^\circ, x_{i+1},\ldots,x_n) \in X \wedge x_i = \phi(\underline{x}^\circ)\}, \tag{18}$$

$$f^{-1}(Y) = \{\underline{x} \in W \mid \underline{x} = (x_1,\ldots,x_n) \wedge (x_1,\ldots,x_{i-1}, \phi(\underline{x}), x_{i+1},\ldots,x_n) \in Y\} \tag{19}$$

EXAMPLE 5.

$$V = R^2$$
$$X = \{(x_1,x_2) \mid x_1 + x_2 > 0\}$$
$$S: \quad x_1 := x_1 - x_2$$

Then

$$f(x_1,x_2) = (x_1-x_2,x_2)$$
$$f(X) = \{(x_1,x_2) \mid \exists x_1^\circ : (x_1^\circ,x_2) \in X \wedge x_1 = x_1^\circ - x_2\}$$
$$= \{(x_1,x_2) \mid \exists x_1^\circ : x_1^\circ + x_2 > 0 \wedge x_1 = x_1^\circ - x_2\}$$
$$= \{(x_1,x_2) \mid x_1 + 2x_2 > 0\}.$$

The last step consisted of eliminating x_1°.

We now take Y = f(X) and apply (19):

$$f^{-1}(Y) = \{(x_1,x_2) \mid (x_1-x_2,x_2) \in Y\}$$
$$= \{(x_1,x_2) \mid (x_1-x_2) + 2x_2 > 0\}$$
$$= \{(x_1,x_2) \mid x_1 + x_2 > 0\} = X$$

Because f is one-to-one, we could have predicted this by using (16).

Remark on the symbolic notation of statements.

We have used the symbol S for statements and f for the corresponding statement functions. When several statements are involved, it is more convenient to denote them by capital letters F, G,..., and their statement functions by the corresponding small letters f, g,... .

c. The compound statement

The sequence G;H is considered a new statement F, semantically defined by

$$f(\underline{x}) = h(g(\underline{x})) \quad \text{for all } \underline{x} \in g^{-1}(h^{-1}(V)).$$

Then

$$f(X) = h(g(X)) \quad \text{for all } X \subset g^{-1}(h^{-1}(V)),$$

$$f^{-1}(Y) = g^{-1}(h^{-1}(Y)) \quad \text{for all } Y \subset V.$$

d. The conditional statement

For any predicate $B \subset V$ the sequence

if B then G else H fi

is called a *conditional statement* F, semantically defined by

$$(20) \qquad f(x) = \begin{cases} g(x) & \text{if } \underline{x} \in B \cap g^{-1}(V) \\ h(x) & \text{if } \underline{x} \in \bar{B} \cap h^{-1}(V). \end{cases}$$

Here $\bar{B} = V - B$, the complement of B. Clearly the domain of f is $B \cap g^{-1}(V) \cup \bar{B} \cap h^{-1}(V)$. (Intersection "$\cap$" has precedence over "$\cup$" throughout this paper).

For all subsets X of this domain and for all $Y \subset V$:

(21) $\quad f(X) = g(B \cap X) \cup h(\bar{B} \cap X)$

(22) $\quad f^{-1}(Y) = B \cap g^{-1}(Y) \cup B \cap h^{-1}(Y)$.

EXAMPLE 6.

$$V = R^2$$

$$X = \{(x,y) \mid x + y > 1\}$$

(For convenience's sake we write (x,y) instead of (x_1,x_2))

F: **if** $x < y$ **then** $x := x + y$ **else** $y := x + y$ **fi**

Consequently

$$B = \{(x,y) \in V \mid x < y\}$$
$$G: \quad x := x + y$$
$$H: \quad y := x + y .$$

Then

$$g(x,y) = (x+y,y)$$
$$h(x,y) = (x,x+y).$$

Our goal is to find f(X) by using (21).

$$B \cap X = \{(x,y) \mid x < y \wedge x + y > 1\}$$

$$\bar{B} \cap X = \{(x,y) \mid x \geq y \wedge x + y > 1\}$$

We apply (18):

$$g(B \cap X) = \{(x,y) \in V \mid \exists x^\circ : (x^\circ,y) \in B \cap X \wedge x = x^\circ + y\}$$
$$= \{(x,y) \in V \mid (x-y,y) \in B \cap X\}$$
$$= \{(x,y) \in V \mid x-y < y \wedge (x-y)+y > 1\}$$
$$= \{(x,y) \in V \mid 1 < x < 2y\}$$

We can find $h(\bar{B} \cap X) = \{(x,y) \in V \mid 1 < y \leq 2x\}$ in a similar way. Then (21) yields

$$f(X) = \{(x,y) \in V \mid 1 < x < 2y \vee 1 < y \leq 2x\}.$$

EXAMPLE 7.

$$V = R^2$$

$$Y = \{(x,y) \mid y = 2\}$$

F: **if** $x > 0$ **then** $y := x + 1$ **else skip fi**

Thus

$$B = \{(x,y) \mid x > 0\}$$

G: $y := x + 1$

H: **skip**.

Then we find by using (22):

$$\begin{aligned} f^{-1}(Y) &= B \cap \{(x,y) \mid (x,x+1) \in Y\} \cup \\ &\quad \bar{B} \cap \{(x,y) \mid y = 2\} \\ &= \{(x,y) \mid x > 0\} \cap \{(x,y) \mid x + 1 = 2\} \cup \\ &\quad \{(x,y) \mid x \leq 0\} \cap \{(x,y) \mid y = 2\} \\ &= \{(x,y) \mid x = 1 \ \vee \ (x \leq 0 \ \wedge \ y = 2)\}. \end{aligned}$$

e. The while statement

We will adopt the obvious notation

$$g^{-k}(Y) = \begin{cases} Y & \text{if } k = 0 \\ g^{-1}(g^{-k+1}(Y)) & \text{if } k > 0. \end{cases}$$

The sequence

(23) **while** B **do** G **od**

is called a *while statement* F. We will define its statement function f in terms of statement G (with statement function g) and predicate $B \subset V$. The domain of f is

(24) $$W = \bigcup_{k=0}^{\infty} g^{-k}(\bar{B}).$$

Intuitively, W is the set of all states $\underline{x} \in V$ which have the property that repeated application of g eventually results in a state $\underline{x}' \in \bar{B}$. Then the statement function for (23) is

$$(25)\quad f(\underline{x}) = \begin{cases} f(g(\underline{x})) & \text{if } \underline{x} \in B \cap W \\ \underline{x} & \text{if } x \in \bar{B} \cap W \end{cases}$$

(The domain W is closely related to the recursive nature of this definition. A (not allowed) attempt to find $f(\underline{x})$ for some $\underline{x} \in V - W$ may result not only in an undefined $g(\underline{x})$ as in Example 1, but also in an infinite process.)
The predicate transformer and its inverse are

$$(26)\quad f(X) = \bar{B} \cap \bigcup_{k=0}^{\infty} T_k \quad \text{for any } X \subseteq W,$$

where

$$\begin{cases} T_0 = X \\ T_{k+1} = g\,(B \cap T_k) \quad (k=0,1,2,\ldots), \end{cases}$$

$$(27)\quad f^{-1}(Y) = \bigcup_{k=0}^{\infty} S_k \quad \text{for any } Y \subset V,$$

where

$$\begin{cases} S_0 = \bar{B} \cap Y \\ S_{k+1} = B \cap g^{-1}(S_k) \quad (k=0,1,2,\ldots) \end{cases}$$

EXAMPLE 8.

$V = Z$ (the set of all integers)

$X = \{x \in Z \mid x > 0 \wedge x \text{ even}\}$

F: while $x \leq 10 \vee (x \geq 20 \wedge x \leq 30)$ do $x := 2 * x + 1$ od.

Then

$B = \{x \in Z \mid x \leq 10 \vee 20 \leq x \leq 30\}$

$\bar{B} = \{x \in Z \mid 10 < x < 20 \vee x > 30\}$

$T_0 = X = \{2,4,6,\ldots\}$, $B \cap T_0 = \{2,4,6,8,10,20,22,24,26,28,30\}$,

$T_1 = g(B \cap T_0) = \{5,9,13,17,21,41,45,49,53,57,61\}$, $B \cap T_1 = \{5,9,21\}$,

$T_2 = g(B \cap T_1) = \{11,19,43\}$, $B \cap T_2 = \emptyset$,

$T_3 = g(B \cap T_2) = \emptyset$ etc.

$$f(X) = \bar{B} \cap (T_0 \cup T_1 \cup T_2) =$$

$$\{11,12,13,14,16,17,18,19,41,43,45,49,53,57,61\} \cup \{32,34,36,\ldots\}$$

Here X satisfies $X \subset W$, because $W = \bigcup_{k=0}^{\infty} g^{-k}(\bar{B}) = \{x \in Z \mid x \geq 0\}$.

EXAMPLE 9.

$$V = Z$$

$$Y = \{x \in Z \mid x \text{ even}\}$$

F: **while** $x \leq 10$ **do** $x := x + 3$ **od**

Then

$$B = \{x \in Z \mid x \leq 10\}$$

$$S_0 = \bar{B} \cap Y = \{12,14,16,\ldots\}$$

$$S_1 = B \cap g^{-1}(S_0) = B \cap \{9,11,13,\ldots\} = \{9\}$$

$$S_2 = B \cap g^{-1}(S_1) = B \cap \{6\} = \{6\}$$

$$S_3 = B \cap g^{-1}(S_2) = B \cap \{3\} = \{3\}$$

etc.

$$f^{-1}(Y) = S_0 \cup S_1 \cup S_2 \cup \ldots$$

$$= \{\ldots,-6,-3,0,3,6,9,12,14,16,\ldots\}.$$

We will now mention some properties that can be derived from (25), (26) and (27). First, the equivalence of

while B **do** G **od**

and

if B **then**

G; **while** B **do** G **od**

else

skip

fi

follows immediately if we formulate the statement function of the latter conditional statement. If we write the predicate transformer and its inverse for this conditional statement, we find the recurrence relations

$$f(X) = f(g(B \cap X)) \cup \bar{B} \cap X$$
$$f^{-1}(Y) = B \cap g^{-1}(f^{-1}(Y)) \cup \bar{B} \cap Y.$$

Some less general results than (26) and (27) that are more practical can be derived from our predicate transformer. E.g. Hoare's "Rule of Iteration" (cf. [2]):

if $\vdash P \wedge B \ \{S\} \ P$ then $\vdash P \ \{\underline{\text{while}}\ B\ \underline{\text{do}}\ S\} \ \neg B \wedge P$
is written in our notation as :

if $g(X \cap B) \subset X$ then $f(X) \subset \bar{B} \cap X$, where
F: while B do G od.

It is proved as follows. It follows from (26) that $f(X) \subset \bar{B}$.
Furthermore

$$g(X \cap B) \subset X \overset{(\alpha)}{\Rightarrow} T_k \subset X \ (k=0,1,\ldots) \Rightarrow \bigcup_{k=0}^{\infty} T_k \subset X \Rightarrow f(X) \subset X.$$

(α) is proved by induction:

i. $T_0 = X$, thus $T_0 \subset X$

ii. Suppose $T_k \subset X$. Then $B \cap T_k \subset B \cap X \Rightarrow$

$g(B \cap T_k) \subset g(B \cap X) \Rightarrow T_{k+1} \subset g(B \cap X)$.

It is given that $g(B \cap X) \subset X$, thus $T_{k+1} \subset X$.

Thus $g(X \cap B) \subset X$ implies $f(X) \subset \bar{B} \cap X$.

We conclude with a rule for the while statement mentioned in DE BAKKER [5] and written there as

W5: $\forall\, u,v\ [\exists\, w[u \subseteq w,\ w;\ p;\ S \subseteq S;\ w,\ w;\ \bar{p} \subseteq v] \Rightarrow u;\ p * S \subseteq p * S;\ v]$

In a less sophisticated notation this theorem applied to our while statement

F: while B do G od

reads as follows.

For all predicates X and Y, it is true that $f(X) \subset Y$ if there is a predicate W satisfying

$$\left.\begin{array}{ll} (28) & X \subset W \\ (29) & g(W \cap B) \subset W \\ (30) & W \cap \bar{B} \subset Y. \end{array}\right\}$$

PROOF. We take W for X in our last example. This gives:

$$\text{if } g(W \cap B) \subset W \quad \text{then} \quad f(W) \subset \bar{B} \cap W.$$

Thus $f(W) \subset \bar{B} \cap W$, since (29) is given. Combining this with (30) yields $f(W) \subset Y$. From (28) it follows that $f(X) \subset f(W)$. Thus $f(X) \subset Y$. □

REFERENCES

[1] FLOYD, R.W., *Assigning Meanings to Programs*, Proc. Symp, Appl. Math. 19, American Math. Soc. (1967) 19-32.

[2] HOARE, C.A.R., *An axiomatic Basis of Computer Programming*, CACM, Vol. 12. No. 10 (October 1969), 576-580.

[3] DIJKSTRA, E.W., *A Simple Axiomatic Basis for Programming Language constructs*, Proc. Kon. Ned. Akad., Ser. A, 77 (or Indagationes Math., 36), 1-15 (1974).

[4] MANNA, Z. & A. PNUELI, *Axiomatic Approach to Total Correctness of Programs*, Report STAN-CS-73-382, Stanford University (1973)

[5] DE BAKKER, J.W., *Flow of control in the proof theory of structured programming*, Proc. 16th IEEE Symp. on Foundations of Computer Science (1975).

[6] MILLS, H.D., *The New Math of Computer Programming*, CACM, Vol 18, No. 1 (January 1975), 43-48

[7] HALMOS, P.R., *Naive Set Theory*, D. van Nostrand Company (1969).