

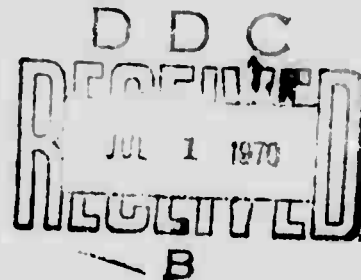
MARCH 1970

AD708077

SECOND-ORDER MATHEMATICAL
THEORY OF COMPUTATION

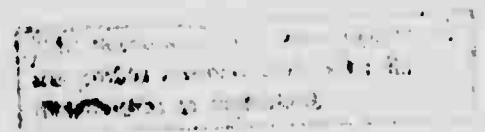
BY

ZOHAR MANNA



Reproduced by the
CLEARINGHOUSE
for Federal Scientific & Technical
Information Springfield Va 22151

SPONSORED BY
ADVANCED RESEARCH PROJECTS AGENCY
ARPA ORDER NO. 457



**BEST
AVAILABLE COPY**

MARCH 1970

SECOND-ORDER MATHEMATICAL THEORY OF COMPUTATION

by

Zohar Manna
Computer Science Department
Stanford University

ABSTRACT: In this work we show that it is possible to formalize all properties regularly observed in (deterministic and non-deterministic) algorithms in second-order predicate calculus.

Moreover, we show that for any given algorithm it suffices to know how to formalize its 'partial correctness' by a second-order formula in order to formalize all other properties by second-order formulas.

This result is of special interest since 'partial correctness' has already been formalized in second-order predicate calculus for many classes of algorithms.

This paper will be presented at the ACM Symposium on Theory of Computing (May 1970).

The research reported here was supported in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense (SD-183).

Reproduced in the USA. Available from the Clearinghouse for Federal Scientific and Technical Information, Springfield, Virginia 22151.
Price: Full size copy \$3.00; microfiche copy \$.65.

Introduction

We normally distinguish between two classes of algorithms: deterministic algorithms and non-deterministic algorithms. A deterministic algorithm defines a single-valued (partial) function, while a non-deterministic algorithm defines a many-valued function. Therefore, while there are only few properties of interest (mainly, termination, partial correctness, total correctness, partial equivalence, equivalence and total equivalence) for deterministic algorithms, there are many more (including partial determinacy, total determinacy and several additional varieties of termination, correctness and equivalence) for non-deterministic algorithms.

Several works have recently formalized properties of algorithms in first-order predicate calculus (see Manna [8]). The importance of such formalization is clear considering the current power of mechanical theorem proving techniques, which hopefully will be further improved in the next few years. Unfortunately there are properties (such as equivalence) that cannot be formalized by a first-order formula; however, they can be formalized by a second-order formula (see Cooper [4]).

In this work we show that for any given algorithm, it is always possible to formalize all its properties by second-order formulas, if one knows how to formalize its 'partial correctness' by a second-order formula.

This result is of special interest since 'partial correctness' has already been formalized for many classes of deterministic algorithms, such as flowchart programs (Floyd [6] and Manna [7]), functional programs (Manna and Pnueli [10]), and Algol-like programs (Ashcroft [1] and Burstall [3]); and also for certain classes of non-deterministic algorithms, such as non-deterministic programs (Manna [9]) and parallel programs (Ashcroft and Manna [2]).

Papers closely related to this work are those of Cooper [5] and Park [11].

I. PARTIAL FUNCTIONS

Let $y = f(x)$ be a partial function mapping D_x (called the input domain) into D_y (called the output domain). That is, for every $x \in D_x$, $f(x)$ is either defined (notation: $*f(x)$) or undefined. A function that is defined for all values of its input domain is called total. A function whose output domain is $\{\text{true}, \text{false}\}$, $\{T, F\}$ for short, is called predicate.

Basic definitions

Let $\psi(x, y)$ be a total predicate over $D_x \times D_y$ and let $x \in D_x$. We say that

- 1.(a) $f(x)$ is partially correct w.r.t. ψ if $*f(x) \supset \psi(x, f(x))$;
- (b) $f(x)$ is totally correct w.r.t. ψ if $*f(x) \wedge \psi(x, f(x))$.

Let $y = f_1(x)$ and $y = f_2(x)$ be any two comparable partial functions, i.e., partial functions with the same input domain D_x and the same output domain D_y . We say that:^{1/}

- 2.(a) $f_1(x)$ and $f_2(x)$ are partially equivalent if

$$*f_1(x) \wedge *f_2(x) \supset f_1(x) = f_2(x) ;$$

- (b) $f_1(x)$ is an extension of $f_2(x)$ if

$$*f_2(x) \supset [*f_1(x) \wedge f_1(x) = f_2(x)] ;$$

- (c) $f_1(x)$ and $f_2(x)$ are equivalent if

$$[*f_1(x) \equiv *f_2(x)] \wedge [*f_1(x) \wedge *f_2(x) \supset f_1(x) = f_2(x)] ;$$

- (d) $f_1(x)$ and $f_2(x)$ are totally equivalent if

$$*f_1(x) \wedge *f_2(x) \wedge f_1(x) = f_2(x) .$$

^{1/} Throughout the paper we are assuming that the connectives have the following precedence: $\sim, \wedge, \vee, \supset$ and $\equiv$. Thus ' $\sim$ ' is more binding than ' $\wedge$ ', ' $\wedge$ ' is more binding than ' $\vee$ ', and so on.

Let $y_1 = f_1(x_1), \dots, y_n = f_n(x_n)$ be partial functions with input domains $D_{x_1}, \dots, D_{x_n}$ and output domains $D_{y_1}, \dots, D_{y_n}$, respectively. Let $\tilde{\Psi}(x_1, y_1, \dots, x_n, y_n)$ be any total predicate over $D_{x_1} \times D_{y_1} \times \dots \times D_{x_n} \times D_{y_n}$. We say that:

- 3.(a) $f_1(\xi_1), \dots, f_n(\xi_n)$ are partially correct w.r.t. $\tilde{\Psi}$ if

$$*f_1(\xi_1) \wedge \dots \wedge *f_n(\xi_n) \supset \tilde{\Psi}(\xi_1, f_1(\xi_1), \dots, \xi_n, f_n(\xi_n)) ;$$
- (b) $f_1(\xi_1), \dots, f_n(\xi_n)$ are totally correct w.r.t. $\tilde{\Psi}$ if

$$*f_1(\xi_1) \wedge \dots \wedge *f_n(\xi_n) \wedge \tilde{\Psi}(\xi_1, f_1(\xi_1), \dots, \xi_n, f_n(\xi_n)) .$$

For $k = 1$ we obtain properties 1(a) and 1(b) as special cases of properties 3(a) and 3(b), respectively. Note that the case $k = 2$ and f_1 is identical to f_2 , can be used to define some properties of a single function which cannot be defined by 1(a) or 1(b).

For example, the property that a function f mapping integers into integers is defined and monotonically increasing (i.e., $\xi > \xi' \supset f(\xi) > f(\xi')$), is exactly the case where the functions f and f' (where f' is identical to f) are correct w.r.t. $\tilde{\Psi}(x, y, x', y') : x > x' \supset y > y'$.

For $k = 2$ and $\tilde{\Psi}(x_1, y_1, x_2, y_2) : x_1 = x_2 \supset y_1 = y_2$ we obtain properties 2(a) and 2(d) as special cases of 3(a) and 3(b), respectively.

The formalization

Suppose that we can formalize the property of f being partially correct by a second-order formula $w(x,q)$ ^{2/} in the following sense:

For every $x \in D_x$ and for every predicate $\psi(x,y)$ over $D_x \times D_y$:

$w(x,\psi)$ if and only if $\neg f(x) \supset \psi(x, f(x))$.

I.e., $w(x,\psi)$ is true if and only if either $f(x)$ is undefined, or $f(x)$ is defined and $\psi(x, f(x))$ is true.

Note that the following two properties of $w(x,q)$ are always true.

For every $x \in D_x$: (i) $w(x,T)$ and therefore $\exists q w(x,q)$, and
(ii) $\neg *f(x) \supset \forall q w(x,q)$.

Theorem 1

0. $f(x)$ is defined if and only if $\neg w(x, \bar{f})$;
- 1.(a) $f(x)$ is partially correct w.r.t. ψ if and only if $w(x, \psi)$;
(b) $f(x)$ is totally correct w.r.t. ψ if and only if $\neg w(x, \neg \psi)$;
- 2.(a) $f_1(x)$ and $f_2(x)$ are partially equivalent if and only if $\forall q [w_1(x, q) \vee w_2(x, \neg q)]$;
(b) $f_1(x)$ is an extension of $f_2(x)$ if and only if $\forall q [w_1(x, q) \supset w_2(x, q)]$;
(c) $f_1(x)$ and $f_2(x)$ are equivalent if and only if $\forall q [w_1(x, q) \equiv w_2(x, q)]$;
(d) $f_1(x)$ and $f_2(x)$ are totally equivalent if and only if $\forall q [\neg w_1(x, q) \vee \neg w_2(x, \neg q)]$;

^{2/} We write $w(x,q)$ to indicate that the wff w has no free variables except the individual variable x and the predicate variable q .

3.(a) $f_1(\xi_1), \dots, f_n(\xi_n)$ are partially correct w.r.t. $\tilde{\psi}$ if and only if

$$\exists q_1 \dots \exists q_n \{w_1(\xi_1, q_1) \wedge \dots \wedge w_n(\xi_n, q_n) \wedge$$

$$\forall \eta_1 \dots \forall \eta_n [q_1(\xi_1, \eta_1) \wedge \dots \wedge q_n(\xi_n, \eta_n) \supset \tilde{\psi}(\xi_1, \eta_1, \dots, \xi_n, \eta_n)]\} ;$$

(b) $f_1(\xi_1), \dots, f_n(\xi_n)$ are totally correct w.r.t. $\tilde{\psi}$ if and only if

$$\forall q_1 \dots \forall q_n \{w_1(\xi_1, q_1) \wedge \dots \wedge w_n(\xi_n, q_n) \supset$$

$$\exists \eta_1 \dots \exists \eta_n [q_1(\xi_1, \eta_1) \wedge \dots \wedge q_n(\xi_n, \eta_n) \wedge \tilde{\psi}(\xi_1, \eta_1, \dots, \xi_n, \eta_n)]\} .$$

Proof of Theorem 1

$$0. \quad \sim w(\xi, \mathcal{F}) \Leftrightarrow *f(\xi) \supset \mathcal{F} \Leftrightarrow \sim *f(\xi) .$$

$$1.(a) \quad w(\xi, \psi) \Leftrightarrow *f(\xi) \supset \psi(\xi, f(\xi)) .$$

$$(b) \quad \sim w(\xi, \sim \psi) \Leftrightarrow \sim [*f(\xi) \supset \sim \psi(\xi, f(\xi))] \Leftrightarrow *f(\xi) \wedge \psi(\xi, f(\xi)) .$$

$$2.(a) \quad \sim \forall q [w_1(\xi, q) \vee w_2(\xi, \sim q)] \Leftrightarrow \exists q [\sim w_1(\xi, q) \wedge \sim w_2(\xi, \sim q)]$$

$$\Leftrightarrow \exists q [*f_1(\xi) \wedge \sim q(\xi, f_1(\xi)) \wedge *f_2(\xi) \wedge q(\xi, f_2(\xi))]$$

$$\Leftrightarrow *f_1(\xi) \wedge *f_2(\xi) \wedge f_1(\xi) \neq f_2(\xi)$$

$$\Leftrightarrow \sim [*f_1(\xi) \wedge *f_2(\xi) \supset f_1(\xi) = f_2(\xi)] .$$

$$(b) \quad \sim \forall q [w_1(\xi, q) \supset w_2(\xi, q)] \Leftrightarrow \exists q [w_1(\xi, q) \wedge \sim w_2(\xi, q)]$$

$$\Leftrightarrow \exists q \{[\sim *f_1(\xi) \vee q(\xi, f_1(\xi))] \wedge [*f_2(\xi) \wedge \sim q(\xi, f_2(\xi))]\}$$

$$\stackrel{3/}{\Leftrightarrow} \exists q \{[\sim *f_1(\xi) \wedge *f_2(\xi) \wedge \sim q(\xi, f_2(\xi))]$$

$$\vee [*f_1(\xi) \wedge *f_2(\xi) \wedge q(\xi, f_1(\xi)) \wedge \sim q(\xi, f_2(\xi))]\}$$

$$\Leftrightarrow \exists q [\sim *f_1(\xi) \wedge *f_2(\xi) \wedge \sim q(\xi, f_2(\xi))]$$

$$\vee \exists q [*f_1(\xi) \wedge *f_2(\xi) \wedge q(\xi, f_1(\xi)) \wedge \sim q(\xi, f_2(\xi))]$$

$$\Leftrightarrow [\sim *f_1(\xi) \wedge *f_2(\xi)]$$

$$\vee \exists q [*f_1(\xi) \wedge *f_2(\xi) \wedge q(\xi, f_1(\xi)) \wedge \sim q(\xi, f_2(\xi))]$$

$$\Leftrightarrow [\sim *f_1(\xi) \wedge *f_2(\xi)] \vee [*f_1(\xi) \wedge *f_2(\xi) \wedge f_1(\xi) \neq f_2(\xi)]$$

$$\stackrel{4/}{\Leftrightarrow} \sim \{ *f_2(\xi) \supset [*f_1(\xi) \wedge f_1(\xi) = f_2(\xi)] \} .$$

$$\stackrel{3/}{\Leftrightarrow} [\sim A \vee C] \wedge [B \wedge \sim D] \text{ is logically equivalent to } [\sim A \wedge B \wedge \sim D] \vee [A \wedge B \wedge C \wedge \sim D] .$$

$$\stackrel{4/}{\Leftrightarrow} [\sim A \wedge B] \vee [A \wedge B \wedge \sim C] \text{ is logically equivalent to } \sim [B \supset A \wedge C] .$$

$$(c) \quad \forall q[w_1(x, q) \equiv w_2(x, q)]$$

$$\Leftrightarrow \forall q[(w_1(x, q) \supset w_2(x, q)) \wedge (w_2(x, q) \supset w_1(x, q))]$$

$$\Leftrightarrow \forall q[w_1(x, q) \supset w_2(x, q)] \wedge \forall q[w_2(x, q) \supset w_1(x, q)] ;$$

then use 2(b).

$$(d) \quad \sim \forall q[\sim w_1(x, q) \vee \sim w_2(x, \sim q)] \Leftrightarrow \exists q[w_1(x, q) \wedge w_2(x, \sim q)]$$

$$\Leftrightarrow \exists q[(\sim *f_1(x) \supset q(x, f_1(x))) \wedge (*f_2(x) \supset \sim q(x, f_2(x)))]$$

$$\stackrel{5/}{\Leftrightarrow} \exists q[(\sim *f_1(x) \wedge \sim *f_2(x)) \vee (\sim *f_1(x) \wedge \sim q(x, f_2(x)))]$$

$$\vee [q(x, f_1(x)) \wedge \sim *f_2(x)] \vee [q(x, f_1(x)) \wedge \sim q(x, f_2(x))]$$

$$\Leftrightarrow \exists q[(\sim *f_1(x) \wedge \sim *f_2(x)) \vee \sim *f_1(x) \vee \sim *f_2(x)$$

$$\vee [q(x, f_1(x)) \wedge \sim q(x, f_2(x))]]$$

$$\stackrel{6/}{\Leftrightarrow} \exists q[(\sim *f_1(x) \vee \sim *f_2(x))$$

$$\vee (*f_1(x) \wedge *f_2(x) \wedge q(x, f_1(x)) \wedge \sim q(x, f_2(x)))]$$

$$\Leftrightarrow \sim *f_1(x) \vee \sim *f_2(x) \vee \exists q[*f_1(x) \wedge *f_2(x) \wedge q(x, f_1(x)) \wedge \sim q(x, f_2(x))]$$

$$\Leftrightarrow \sim *f_1(x) \vee \sim *f_2(x) \vee [*f_1(x) \wedge *f_2(x) \wedge f_1(x) \neq f_2(x)]$$

$$\stackrel{7/}{\Leftrightarrow} \sim [*f_1(x) \wedge *f_2(x) \wedge f_1(x) = f_2(x)] .$$

$$3.(a) \quad \exists q_1 \dots \exists q_n [w_1(x_1, q_1) \wedge \dots \wedge w_n(x_n, q_n) \wedge$$

$$\forall \eta_1 \dots \forall \eta_n [q_1(x_1, \eta_1) \wedge \dots \wedge q_n(x_n, \eta_n) \supset \tilde{\Psi}(x_1, \eta_1, \dots, x_n, \eta_n)]]$$

$$\Leftrightarrow \exists q_1 \dots \exists q_n [(*f_1(x_1) \supset q_1(x_1, f_1(x_1))) \wedge \dots \wedge (*f_n(x_n) \supset q_n(x_n, f_n(x_n)))]$$

$$\wedge \forall \eta_1 \dots \forall \eta_n [q_1(x_1, \eta_1) \wedge \dots \wedge q_n(x_n, \eta_n) \supset \tilde{\Psi}(x_1, \eta_1, \dots, x_n, \eta_n)]$$

$$\Leftrightarrow *f_1(x_1) \wedge \dots \wedge *f_n(x_n) \supset \tilde{\Psi}(x_1, f_1(x_1), \dots, x_n, f_n(x_n)) .$$

$$(b) \quad \sim \forall q_1 \dots \forall q_n [w_1(x_1, q_1) \wedge \dots \wedge w_n(x_n, q_n) \supset$$

$$\exists \eta_1 \dots \exists \eta_n [q_1(x_1, \eta_1) \wedge \dots \wedge q_n(x_n, \eta_n) \wedge \tilde{\Psi}(x_1, \eta_1, \dots, x_n, \eta_n)]]$$

$$\Leftrightarrow \exists q_1 \dots \exists q_n [(*f_1(x_1) \supset q_1(x_1, f_1(x_1))) \wedge \dots \wedge (*f_n(x_n) \supset q_n(x_n, f_n(x_n)))] \wedge$$

$$\forall \eta_1 \dots \forall \eta_n [q_1(x_1, \eta_1) \wedge \dots \wedge q_n(x_n, \eta_n) \supset \sim \tilde{\Psi}(x_1, \eta_1, \dots, x_n, \eta_n)]$$

$$\Leftrightarrow *f_1(x_1) \wedge \dots \wedge *f_n(x_n) \supset \sim \tilde{\Psi}(x_1, f_1(x_1), \dots, x_n, f_n(x_n))$$

$$\Leftrightarrow \sim [*f_1(x_1) \wedge \dots \wedge *f_n(x_n) \wedge \tilde{\Psi}(x_1, f_1(x_1), \dots, x_n, f_n(x_n))] .$$

Q.E.D.

$$\stackrel{5/}{[A \supset C] \wedge [B \supset \sim D] \text{ is logically equivalent to } [\sim A \wedge \sim B] \vee [\sim A \wedge \sim D] \vee [C \wedge \sim B] \vee [C \wedge \sim D].}$$

$$\stackrel{6/}{[\sim A \wedge \sim B] \vee \sim A \vee \sim B \vee [C \wedge \sim D] \text{ is logically equivalent to } \sim A \vee \sim B \vee [A \wedge B \wedge C \wedge \sim D].}$$

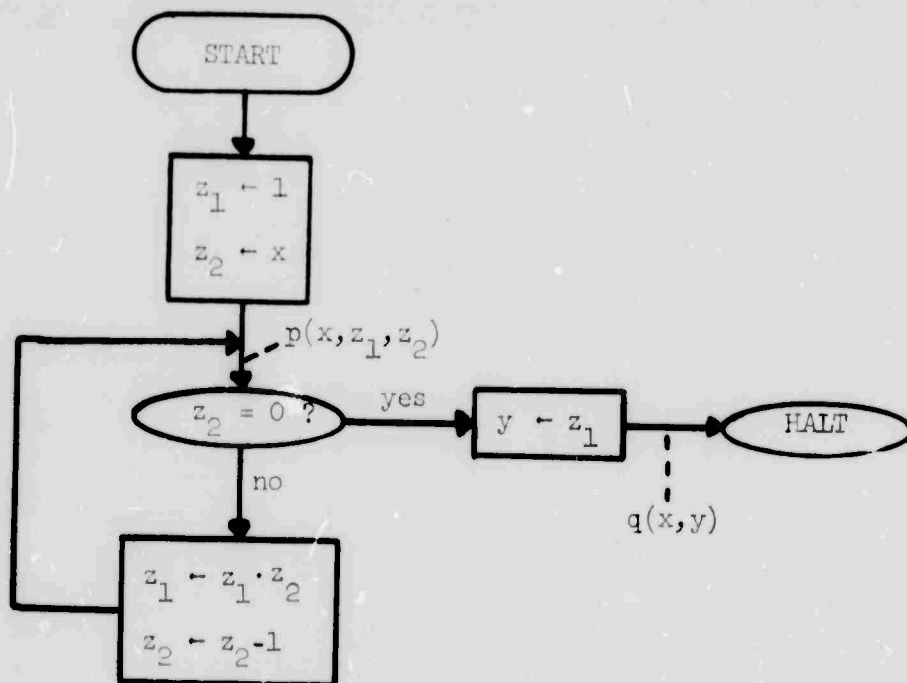
$$\stackrel{7/}{\sim A \vee \sim B \vee [A \wedge B \wedge \sim C] \text{ is logically equivalent to } \sim [A \wedge B \wedge C].}$$

Example

Our theory is based on the assumption that for a given partial function f , one knows how to construct the appropriate second-order formula $w(x,q)$. The construction depends, in general, on the (deterministic) algorithm defining f . However, as mentioned in the Introduction, the construction of $w(x,q)$ has already been described for many classes of deterministic algorithms, such as: flowchart programs (Floyd [6], Manna [7]), functional programs (Manna and Pnueli [10]), and Algol-like programs (Ashcroft [1], Burstall [3]).

We shall illustrate the construction of $w(x,q)$ for the factorial function over the integers (undefined for negative integers) defined by four different algorithms; the first two are flowchart programs and the other two are functional programs. Note that the formulas reflect the computations of the algorithms in a very natural way.^{8/}

1.

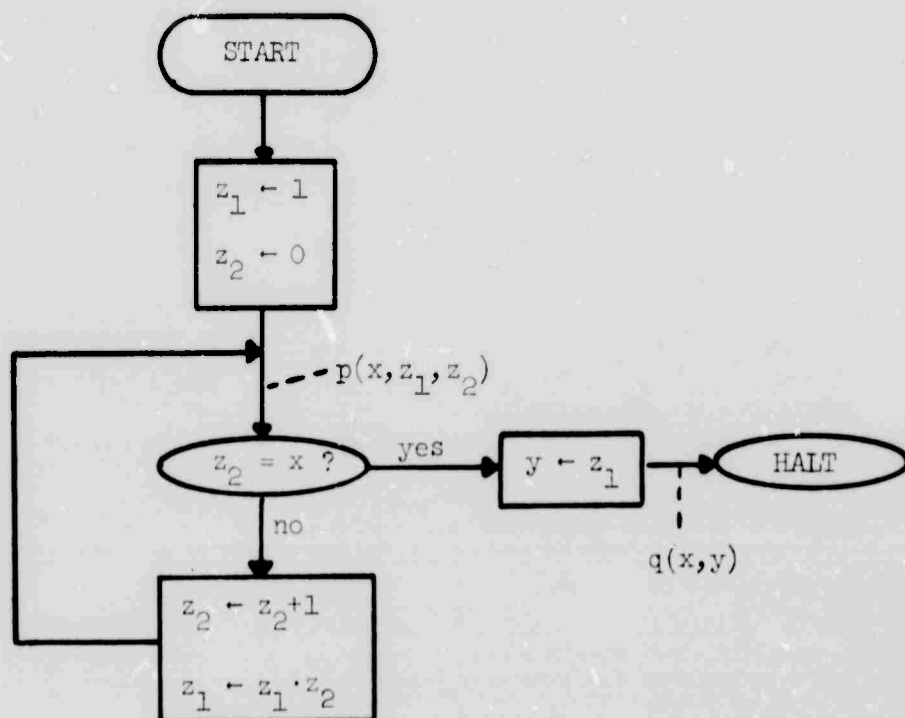


$w_1(x,q)$ is

$\exists p \{ p(x,1,x) \wedge \forall z_1 \forall z_2 [p(x,z_1,z_2) \supset \text{if } z_2 = 0 \text{ then } q(x,z_1) \text{ else } p(x,z_1 \cdot z_2, z_2 - 1)] \}$.

^{8/} In the formulas, 'if A then B else C' stands for $[A \supset B] \wedge [\sim A \supset C]$.

2.



$w_2(x, q)$ is

$\exists p\{p(x, 1, 0) \wedge \forall z_1 \forall z_2 [p(x, z_1, z_2) \supset \text{if } z_2 = x \text{ then } q(x, z_1) \text{ else } q(x, z_1 \cdot (z_2 + 1), z_2 + 1)]\}$.

3. $y = f(x)$, where $f(x) \leftarrow \text{if } x = 0 \text{ then } 1 \text{ else } x \cdot f(x-1)$.

$w_3(x, q)$ is: $\text{if } x = 0 \text{ then } q(x, 1) \text{ else } \forall z [q(x-1, z) \supset q(x, x \cdot z)]$.

4. $y = g(x, 0)$, where $g(x, z) \leftarrow \text{if } x = z \text{ then } 1 \text{ else } (z+1) \cdot g(x, z+1)$.

$w_4(x, q)$ is: $\exists p\{\forall y [p(x, 0, y) \supset q(x, y)] \wedge$

$\forall x \forall z [\text{if } x = z \text{ then } p(x, z, 1)$

$\text{else } \forall t [p(x, z+1, t) \supset p(x, z, (z+1) \cdot t)]\}$.

II. MANY-VALUED FUNCTIONS

One natural extension of our results is obtained by considering many-valued functions rather than single-valued functions.

Let $y = F(x)$ be a many-valued function mapping elements of D_x into subsets of D_y ; that is, for every $\xi \in D_x$, $F(\xi)$ is a (possibly empty) subset of D_y . We say that:

1. $F(\xi)$ is defined if $F(\xi) \neq \emptyset$.

2.(a) $F(\xi)$ is partially determinate if

$$\forall y_1 \forall y_2 [y_1 \in F(\xi) \wedge y_2 \in F(\xi) \supset y_1 = y_2],$$

i.e., $F(\xi)$ is either empty or a singleton;

(b) $F(\xi)$ is totally determinate if

$$F(\xi) \neq \emptyset \wedge \forall y_1 \forall y_2 [y_1 \in F(\xi) \wedge y_2 \in F(\xi) \supset y_1 = y_2],$$

i.e., $F(\xi)$ is a singleton.

Let $\psi(x, y)$ be a total predicate over $D_x \times D_y$. We say that:

3.(a) $F(\xi)$ is partially $\exists$ -correct w.r.t. ψ if

$$F(\xi) = \emptyset \vee \exists y [y \in F(\xi) \wedge \psi(\xi, y)];$$

(b) $F(\xi)$ is totally $\exists$ -correct w.r.t. ψ if

$$\exists y [y \in F(\xi) \wedge \psi(\xi, y)];$$

4.(a) $F(\xi)$ is partially $\forall$ -correct w.r.t. ψ if

$$\forall y [y \in F(\xi) \supset \psi(\xi, y)];$$

(b) $F(\xi)$ is totally $\forall$ -correct w.r.t. ψ if

$$F(\xi) \neq \emptyset \wedge \forall y [y \in F(\xi) \supset \psi(\xi, y)].$$

Let $y = F_1(x)$ and $y = F_2(x)$ be any two comparable many-valued functions, i.e., many-valued functions with the same input domain D_x and the same output domain D_y . We say that:

5.(a) $F_1(\xi)$ and $F_2(\xi)$ are partially non-disjoint if

$$F_1(\xi) = \emptyset \vee F_2(\xi) = \emptyset \vee [F_1(\xi) \cap F_2(\xi) \neq \emptyset] ;$$

(b) $F_1(\xi)$ and $F_2(\xi)$ are totally non-disjoint if

$$F_1(\xi) \cap F_2(\xi) \neq \emptyset ,$$

$$\text{i.e., } \exists y_1 \exists y_2 [y_1 \in F_1(\xi) \wedge y_2 \in F_2(\xi) \wedge y_1 = y_2] ;$$

6.(a) $F_1(\xi)$ and $F_2(\xi)$ are partially determinate-equivalent if

$$\forall y_1 \forall y_2 [y_1 \in F_1(\xi) \wedge y_2 \in F_2(\xi) \supset y_1 = y_2] ;$$

(b) $F_1(\xi)$ and $F_2(\xi)$ are totally determinate-equivalent if

$$F_1(\xi) \neq \emptyset \wedge F_2(\xi) \neq \emptyset \wedge \forall y_1 \forall y_2 [y_1 \in F_1(\xi) \wedge y_2 \in F_2(\xi) \supset y_1 = y_2] ,$$

$$\text{i.e., } F_1(\xi) = F_2(\xi) \text{ and they are singletons;}$$

7.(a) $F_1(\xi)$ is an extension of $F_2(\xi)$ if $F_1(\xi) \supseteq F_2(\xi)$;

(b) $F_1(\xi)$ and $F_2(\xi)$ are equivalent if $F_1(\xi) = F_2(\xi)$;

8.(a) $F_1(\xi)$ and $F_2(\xi)$ are partially equivalent if

$$F_1(\xi) = \emptyset \vee F_2(\xi) = \emptyset \vee F_1(\xi) = F_2(\xi) ;$$

(b) $F_1(\xi)$ and $F_2(\xi)$ are totally equivalent if

$$F_1(\xi) \neq \emptyset \wedge F_2(\xi) \neq \emptyset \wedge F_1(\xi) = F_2(\xi) .$$

Suppose that we can formalize the property of F being partially correct by a second-order formula $W(x,q)$ in the following sense:

For every $\xi \in D_x$ and for every predicate $\psi(x,y) : W(\xi, \psi)$ if and only if $\forall y [y \in F(\xi) \supset \psi(\xi, y)]$.

Note that the following two properties of $W(x,q)$ are always true.
 For every $\xi \in D_x$: (i) $W(\xi, T)$ and therefore $\exists q W(\xi, q)$, and
 (ii) $F(\xi) = \emptyset \supset \forall q W(\xi, q)$.

Theorem 2

1. $F(\xi)$ is defined if and only if $\sim W(\xi, \mathcal{F})$;
- 2.(a) $F(\xi)$ is partially determinate if and only if
 $\forall q [W(\xi, q) \vee W(\xi, \sim q)]$;
 (b) $F(\xi)$ is totally determinate if and only if
 $\sim W(\xi, \mathcal{F}) \wedge \forall q [\sim W(\xi, q) \vee \sim W(\xi, \sim q)]$;
- 3.(a) $F(\xi)$ is partially $\mathcal{F}$ -correct w.r.t. ψ if and only if
 $W(\xi, \mathcal{F}) \vee \sim W(\xi, \sim \psi)$;
 (b) $F(\xi)$ is totally $\mathcal{F}$ -correct w.r.t. ψ if and only if
 $\sim W(\xi, \sim \psi)$;
- 4.(a) $F(\xi)$ is partially $\mathcal{V}$ -correct w.r.t. ψ if and only if $W(\xi, \psi)$;
 (b) $F(\xi)$ is totally $\mathcal{V}$ -correct w.r.t. ψ if and only if
 $\sim W(\xi, \mathcal{F}) \wedge W(\xi, \psi)$;
- 5.(a) $F_1(\xi)$ and $F_2(\xi)$ are partially non-disjoint if and only if
 $W_1(\xi, \mathcal{F}) \vee W_2(\xi, \mathcal{F}) \vee \forall q [\sim W_1(\xi, q) \vee \sim W_2(\xi, \sim q)]$;
 (b) $F_1(\xi)$ and $F_2(\xi)$ are totally non-disjoint if and only if
 $\forall q [\sim W_1(\xi, q) \vee \sim W_2(\xi, \sim q)]$;

- 6.(a) $F_1(x)$ and $F_2(x)$ are partially determinate-equivalent if and only if $\forall q[W_1(x, q) \vee W_2(x, \sim q)]$;
- (b) $F_1(x)$ and $F_2(x)$ are totally determinate-equivalent if and only if $\sim W_1(x, \mathcal{F}) \wedge \sim W_2(x, \mathcal{F}) \wedge \forall q[W_1(x, q) \vee W_2(x, \sim q)]$;
- 7.(a) $F_1(x)$ is an extension of $F_2(x)$ if and only if $\forall q[W_1(x, q) \supset W_2(x, q)]$;
- (b) $F_1(x)$ and $F_2(x)$ are equivalent if and only if $\forall q[W_1(x, q) \equiv W_2(x, q)]$;
- 8.(a) $F_1(x)$ and $F_2(x)$ are partially equivalent if and only if $W_1(x, \mathcal{F}) \vee W_2(x, \mathcal{F}) \vee \forall q[W_1(x, q) \equiv W_2(x, q)]$;
- (b) $F_1(x)$ and $F_2(x)$ are totally equivalent if and only if $\sim W_1(x, \mathcal{F}) \wedge \sim W_2(x, \mathcal{F}) \wedge \forall q[W_1(x, q) \equiv W_2(x, q)]$.

Proof of Theorem 2

1. $\sim W(x, \mathcal{F}) \Leftrightarrow \sim \forall y[y \in F(x) \supset \mathcal{F}] \Leftrightarrow \exists y[y \in F(x)] \Leftrightarrow F(x) \neq \emptyset$.
- 2.(a) $\sim \forall q[W(x, q) \vee W(x, \sim q)] \Leftrightarrow \exists q[\sim W(x, q) \wedge \sim W(x, \sim q)]$
 $\Leftrightarrow \exists q[\exists y[y \in F(x) \wedge \sim q(x, y)] \wedge \exists y[y \in F(x) \wedge q(x, y)]]$
 $\Leftrightarrow \exists y_1 \exists y_2[y_1 \in F(x) \wedge y_2 \in F(x) \wedge y_1 \neq y_2]$
 $\Leftrightarrow \sim \forall y_1 \forall y_2[y_1 \in F(x) \wedge y_2 \in F(x) \supset y_1 = y_2]$.
- (b) Follows from 1 and 2(a).
- 3.(a) Follows from 1 and 3(b).
- (b) $\sim W(x, \sim \psi) \Leftrightarrow \sim \forall y[y \in F(x) \supset \sim \psi(x, y)] \Leftrightarrow \exists y[y \in F(x) \wedge \psi(x, y)]$.

$$4.(a) \quad W(\xi, \psi) \Leftrightarrow \forall y[y \in F(\xi) \supset \psi(\xi, y)] .$$

(b) Follows from 1 and 4(a).

5.(a) Follows from 1 and 5(b).

$$\begin{aligned} (b) \quad & \sim \forall q[\sim W_1(\xi, q) \vee \sim W_2(\xi, \sim q)] \Leftrightarrow \exists q[W_1(\xi, q) \wedge W_2(\xi, \sim q)] \\ & \Leftrightarrow \exists q[\forall y[y \in F_1(\xi) \supset q(\xi, y)] \wedge \forall y[y \in F_2(\xi) \supset \sim q(\xi, y)]] \\ & \Leftrightarrow \forall y_1 \forall y_2[y_1 \in F_1(\xi) \wedge y_2 \in F_2(\xi) \supset y_1 \neq y_2] \\ & \Leftrightarrow \sim \exists y_1 \exists y_2[y_1 \in F_1(\xi) \wedge y_2 \in F_2(\xi) \wedge y_1 = y_2] . \end{aligned}$$

$$\begin{aligned} 6.(a) \quad & \sim \forall q[W_1(\xi, q) \vee W_2(\xi, \sim q)] \Leftrightarrow \exists q[\sim W_1(\xi, q) \wedge \sim W_2(\xi, \sim q)] \\ & \Leftrightarrow \exists q[\exists y[y \in F_1(\xi) \wedge \sim q(\xi, y)] \wedge \exists y[y \in F_2(\xi) \wedge q(\xi, y)]] \\ & \Leftrightarrow \exists y_1 \exists y_2[y_1 \in F_1(\xi) \wedge y_2 \in F_2(\xi) \wedge y_1 \neq y_2] \\ & \Leftrightarrow \sim \forall y_1 \forall y_2[y_1 \in F_1(\xi) \wedge y_2 \in F_2(\xi) \supset y_1 = y_2] . \end{aligned}$$

(b) Follows from 1 and 6(a).

$$\begin{aligned} 7.(a) \quad & \sim \forall q[W_1(\xi, q) \supset W_2(\xi, q)] \Leftrightarrow \exists q[W_1(\xi, q) \wedge \sim W_2(\xi, q)] \\ & \Leftrightarrow \exists q[\forall y[y \in F_1(\xi) \supset q(\xi, y)] \wedge \exists y[y \in F_2(\xi) \wedge \sim q(\xi, y)]] \\ & \Leftrightarrow \exists y[y \in F_2(\xi) \wedge y \notin F_1(\xi)] \Leftrightarrow \sim [F_1(\xi) \supseteq F_2(\xi)] . \end{aligned}$$

(b) Follows from 7(a).

8.(a) and (b) Follows from 1 and 7(b).

Q.E.D.

III. AUGMENTED MANY-VALUED FUNCTIONS

In order to formalize several more natural properties of a non-deterministic algorithm it is usually not sufficient to consider it as defining a regular many-valued function F (mapping elements of D_x into subsets of D_y), but rather as defining an augmented many-valued function F^+ , mapping elements of D_x into non-empty subsets of $D_y \cup \{\infty\}$. Thus, for example, for some algorithm with $D_x = D_y = \{\text{the integers}\}$ we write $F^+(7) = \{3, 5, \infty\}$ to mean that for input $x = 7$: there is at least one finite computation of the algorithm yielding $y = 3$, there is at least one finite computation of the algorithm yielding $y = 5$, and there is at least one infinite computation. We say that:

- 1.(a) $F^+(\xi)$ is $\exists$ -defined if $\exists y[y \in F^+(\xi) \wedge y \neq \infty]$;
- (b) $F^+(\xi)$ is $\forall$ -defined if $\forall y[y \in F^+(\xi) \supset y \neq \infty]$.
- 2.(a) $F^+(\xi)$ is partially determinate if $\forall y_1 \forall y_2 [y_1 \in F^+(\xi) \wedge y_2 \in F^+(\xi) \wedge y_1 \neq \infty \wedge y_2 \neq \infty \supset y_1 = y_2]$;
- (b) $F^+(\xi)$ is totally determinate if $\forall y_1 \forall y_2 [y_1 \in F^+(\xi) \wedge y_2 \in F^+(\xi) \supset y_1 \neq \infty \wedge y_2 \neq \infty \wedge y_1 = y_2]$.
- 3.(a) $F^+(\xi)$ is partially $\exists$ -correct w.r.t. ψ if $\exists y[y \in F^+(\xi) \wedge [y \neq \infty \supset \psi(\xi, y)]]$;
- (b) $F^+(\xi)$ is totally $\exists$ -correct w.r.t. ψ if $\exists y[y \in F^+(\xi) \wedge y \neq \infty \wedge \psi(\xi, y)]$.
- 4.(a) $F^+(\xi)$ is partially $\forall$ -correct w.r.t. ψ if $\forall y[y \in F^+(\xi) \wedge y \neq \infty \supset \psi(\xi, y)]$;
- (b) $F^+(\xi)$ is totally $\forall$ -correct w.r.t. ψ if $\forall y[y \in F^+(\xi) \supset y \neq \infty \wedge \psi(\xi, y)]$.

Let $y = F_1^+(x)$ and $y = F_2^+(x)$ be any two comparable augmented many-valued functions, i.e., functions with the same input domain D_x and the same output domain D_y . We say that:

5.(a) $F_1^+(\xi)$ and $F_2^+(\xi)$ are partially $\exists$ -equivalent if

$$\exists y_1 \exists y_2 [y_1 \in F_1^+(\xi) \wedge y_2 \in F_2^+(\xi) \wedge [y_1 \neq \infty \wedge y_2 \neq \infty \supset y_1 = y_2]] ;$$

(b) $F_1^+(\xi)$ and $F_2^+(\xi)$ are totally $\exists$ -equivalent if

$$\exists y_1 \exists y_2 [y_1 \in F_1^+(\xi) \wedge y_2 \in F_2^+(\xi) \wedge y_1 \neq \infty \wedge y_2 \neq \infty \wedge y_1 = y_2] ;$$

6.(a) $F_1^+(\xi)$ and $F_2^+(\xi)$ are partially determinate-equivalent if

$$\forall y_1 \forall y_2 [y_1 \in F_1^+(\xi) \wedge y_2 \in F_2^+(\xi) \wedge y_1 \neq \infty \wedge y_2 \neq \infty \supset y_1 = y_2] ;$$

(b) $F_1^+(\xi)$ and $F_2^+(\xi)$ are totally determinate-equivalent if

$$\forall y_1 \forall y_2 [y_1 \in F_1^+(\xi) \wedge y_2 \in F_2^+(\xi) \supset y_1 \neq \infty \wedge y_2 \neq \infty \wedge y_1 = y_2] ;$$

7.(a) $F_1^+(\xi)$ partially extends $F_2^+(\xi)$ if $[F_1^+(\xi) - \{\infty\}] \supseteq [F_2^+(\xi) - \{\infty\}]$;

(b) $F_1^+(\xi)$ totally extends $F_2^+(\xi)$ if $F_1^+(\xi) \supseteq F_2^+(\xi)$;

8.(a) $F_1^+(\xi)$ and $F_2^+(\xi)$ are partially equivalent if

$$[F_1^+(\xi) - \{\infty\}] = [F_2^+(\xi) - \{\infty\}] ;$$

(b) $F_1^+(\xi)$ and $F_2^+(\xi)$ are totally equivalent if $F_1^+(\xi) = F_2^+(\xi)$.

Suppose that we can formalize the properties of F^+ being partially $\exists$ -correct and partially $\forall$ -correct by second-order formulas $W^{\exists}(x, q)$ and $W^{\forall}(x, q)$, respectively, in the following sense:

For every $\xi \in D_x$ and for every predicate $\psi(x, y)$ over $D_x \times D_y$:

$$W^{\exists}(\xi, \psi) \text{ if and only if } \exists y [y \in F^+(\xi) \wedge [y \neq \infty \supset \psi(\xi, y)]] ,$$

and

$$W^{\forall}(\xi, \psi) \text{ if and only if } \forall y [y \in F^+(\xi) \wedge y \neq \infty \supset \psi(\xi, y)] .$$

Note that the following properties of $W^{\exists}(x, q)$ and $W^{\forall}(x, q)$ are always true. For every $\xi \in D_x$: (i) $W^{\exists}(\xi, T)$ and therefore $\exists q W^{\exists}(\xi, q)$, (ii) $W^{\forall}(\xi, T)$ and therefore $\exists q W^{\forall}(\xi, q)$, (iii) $\infty \in F^+(\xi) \supset \forall q W^{\exists}(\xi, q)$, and (iv) $F^+(\xi) = \{\infty\} \supset \forall q W^{\forall}(\xi, q)$.

Theorem 3

- 1.(a) $F^+(\xi)$ is $\exists$ -defined if and only if $\sim W^V(\xi, \mathcal{F})$;
 (b) $F^+(\xi)$ is $\forall$ -defined if and only if $\sim W^H(\xi, \mathcal{F})$;
- 2.(a) $F^+(\xi)$ is partially determinate if and only if $\forall q[W^V(\xi, q) \vee W^V(\xi, \sim q)]$;
 (b) $F^+(\xi)$ is totally determinate if and only if $\forall q[\sim W^H(\xi, q) \vee \sim W^H(\xi, \sim q)]$;
- 3.(a) $F^+(\xi)$ is partially $\exists$ -correct w.r.t. ψ if and only if $W^H(\xi, \psi)$;
 (b) $F^+(\xi)$ is totally $\exists$ -correct w.r.t. ψ if and only if $\sim W^V(\xi, \sim \psi)$;
- 4.(a) $F^+(\xi)$ is partially $\forall$ -correct w.r.t. ψ if and only if $W^V(\xi, \psi)$;
 (b) $F^+(\xi)$ is totally $\forall$ -correct w.r.t. ψ if and only if $\sim W^H(\xi, \sim \psi)$;
- 5.(a) $F_1^+(\xi)$ and $F_2^+(\xi)$ are partially $\exists$ -equivalent if and only if
 $\forall q[W_1^H(\xi, q) \vee W_2^H(\xi, \sim q)]$;
 (b) $F_1^+(\xi)$ and $F_2^+(\xi)$ are totally $\exists$ -equivalent if and only if
 $\forall q[\sim W_1^V(\xi, q) \vee \sim W_2^V(\xi, \sim q)]$;
- 6.(a) $F_1^+(\xi)$ and $F_2^+(\xi)$ are partially determinate-equivalent if and
 only if $\forall q[W_1^V(\xi, q) \vee W_2^V(\xi, \sim q)]$;
 (b) $F_1^+(\xi)$ and $F_2^+(\xi)$ are totally determinate-equivalent if and
 only if $\forall q[\sim W_1^H(\xi, q) \vee \sim W_2^H(\xi, \sim q)]$;
- 7.(a) $F_1^+(\xi)$ partially extends $F_2^+(\xi)$ if and only if $\forall q[W_1^V(\xi, q) \supset W_2^V(\xi, q)]$;
 (b) $F_1^+(\xi)$ totally extends $F_2^+(\xi)$ if and only if $\forall q[W_2^H(\xi, q) \supset W_1^H(\xi, q)]$;
- 8.(a) $F_1^+(\xi)$ and $F_2^+(\xi)$ are partially equivalent if and only if
 $\forall q[W_1^V(\xi, q) \equiv W_2^V(\xi, q)]$;
 (b) $F_1^+(\xi)$ and $F_2^+(\xi)$ are totally equivalent if and only if
 $\forall q[W_1^H(\xi, q) \equiv W_2^H(\xi, q)]$.

Proof of Theorem 3

$$1.(a) \sim W^V(\xi, \mathcal{F}) \Leftrightarrow \sim \forall y[y \in F^+(\xi) \wedge y \neq \infty \supset \mathcal{F}] \Leftrightarrow \exists y[y \in F^+(\xi) \wedge y \neq \infty] .$$

$$(b) \sim W^H(\xi, \mathcal{F}) \Leftrightarrow \sim \forall y\{y \in F^+(\xi) \wedge [y \neq \infty \supset \mathcal{F}]\} \Leftrightarrow \forall y[y \in F^+(\xi) \supset y \neq \infty] .$$

$$\begin{aligned} 2.(a) \sim \forall q[W^V(\xi, q) \vee W^V(\xi, \sim q)] &\Leftrightarrow \exists q[\sim W^V(\xi, q) \wedge \sim W^V(\xi, \sim q)] \\ &\Leftrightarrow \exists q[\sim \forall y[y \in F^+(\xi) \wedge y \neq \infty \supset q(\xi, y)] \wedge \sim \forall y[y \in F^+(\xi) \wedge y \neq \infty \supset \sim q(\xi, y)]] \\ &\Leftrightarrow \exists q\{\exists y[y \in F^+(\xi) \wedge y \neq \infty \wedge \sim q(\xi, y)] \wedge \exists y[y \in F^+(\xi) \wedge y \neq \infty \wedge q(\xi, y)]\} \\ &\Leftrightarrow \exists y_1 \exists y_2[y_1 \in F^+(\xi) \wedge y_1 \neq \infty \wedge y_2 \in F^+(\xi) \wedge y_2 \neq \infty \wedge y_1 \neq y_2] \\ &\Leftrightarrow \sim \forall y_1 \forall y_2[y_1 \in F^+(\xi) \wedge y_2 \in F^+(\xi) \wedge y_1 \neq \infty \wedge y_2 \neq \infty \supset y_1 = y_2] . \end{aligned}$$

$$\begin{aligned} (b) \sim \forall q[\sim W^H(\xi, q) \vee \sim W^H(\xi, \sim q)] &\Leftrightarrow \exists q[W^H(\xi, q) \wedge W^H(\xi, \sim q)] \\ &\Leftrightarrow \exists q\{\exists y\{y \in F^+(\xi) \wedge [y \neq \infty \supset q(\xi, y)]\} \wedge \exists y\{y \in F^+(\xi) \wedge [y \neq \infty \supset \sim q(\xi, y)]\}\} \\ &\Leftrightarrow \exists y_1 \exists y_2\{y_1 \in F^+(\xi) \wedge y_2 \in F^+(\xi) \wedge [y_1 \neq \infty \wedge y_2 \neq \infty \supset y_1 \neq y_2]\} \\ &\Leftrightarrow \sim \forall y_1 \forall y_2[y_1 \in F^+(\xi) \wedge y_2 \in F^+(\xi) \supset y_1 \neq \infty \wedge y_2 \neq \infty \wedge y_1 = y_2] . \end{aligned}$$

$$3.(a) W^H(\xi, \psi) \Leftrightarrow \exists y\{y \in F^+(\xi) \wedge [y \neq \infty \supset \psi(\xi, y)]\} .$$

$$\begin{aligned} (b) \sim W^V(\xi, \sim \psi) &\Leftrightarrow \sim \forall y[y \in F^+(\xi) \wedge y \neq \infty \supset \sim \psi(\xi, y)] \\ &\Leftrightarrow \exists y[y \in F^+(\xi) \wedge y \neq \infty \wedge \psi(\xi, y)] . \end{aligned}$$

$$4.(a) W^V(\xi, \psi) \Leftrightarrow \forall y[y \in F^+(\xi) \wedge y \neq \infty \supset \psi(\xi, y)] .$$

$$\begin{aligned} (b) \sim W^H(\xi, \sim \psi) &\Leftrightarrow \sim \exists y\{y \in F^+(\xi) \wedge [y \neq \infty \supset \sim \psi(\xi, y)]\} \\ &\Leftrightarrow \forall y[y \in F^+(\xi) \supset y \neq \infty \wedge \psi(\xi, y)] . \end{aligned}$$

$$\begin{aligned} 5.(a) \sim \forall q[W_1^H(\xi, q) \vee W_2^H(\xi, \sim q)] &\Leftrightarrow \exists q[\sim W_1^H(\xi, q) \wedge \sim W_2^H(\xi, \sim q)] \\ &\Leftrightarrow \exists q\{\forall y[y \in F_1^+(\xi) \supset y \neq \infty \wedge \sim q(\xi, y)] \wedge \forall y[y \in F_2^+(\xi) \supset y \neq \infty \wedge q(\xi, y)]\} \\ &\Leftrightarrow \forall y_1 \forall y_2[y_1 \in F_1^+(\xi) \wedge y_2 \in F_2^+(\xi) \supset y_1 \neq \infty \wedge y_2 \neq \infty \wedge y_1 \neq y_2] \\ &\Leftrightarrow \sim \exists y_1 \exists y_2\{y_1 \in F_1^+(\xi) \wedge y_2 \in F_2^+(\xi) \wedge [y_1 \neq \infty \wedge y_2 \neq \infty \supset y_1 = y_2]\} . \end{aligned}$$

$$\begin{aligned} (b) \sim \forall q[\sim W_1^V(\xi, q) \vee \sim W_2^V(\xi, \sim q)] &\Leftrightarrow \exists q[W_1^V(\xi, q) \wedge W_2^V(\xi, \sim q)] \\ &\Leftrightarrow \exists q\{\forall y[y \in F_1^+(\xi) \wedge y \neq \infty \supset q(\xi, y)] \wedge \forall y[y \in F_2^+(\xi) \wedge y \neq \infty \supset \sim q(\xi, y)]\} \\ &\Leftrightarrow \forall y_1 \forall y_2[y_1 \in F_1^+(\xi) \wedge y_1 \neq \infty \wedge y_2 \in F_2^+(\xi) \wedge y_2 \neq \infty \supset y_1 \neq y_2] \\ &\Leftrightarrow \sim \exists y_1 \exists y_2\{y_1 \in F_1^+(\xi) \wedge y_2 \in F_2^+(\xi) \wedge y_1 \neq \infty \wedge y_2 \neq \infty \wedge y_1 = y_2\} . \end{aligned}$$

$$\begin{aligned}
6.(a) \quad & \sim \forall q [W_1^V(\xi, q) \vee W_2^V(\xi, \sim q)] \Leftrightarrow \exists q [\sim W_1^V(\xi, q) \wedge \sim W_2^V(\xi, \sim q)] \\
& \Leftrightarrow \exists q \{ \exists y [y \in F_1^+(\xi) \wedge y \neq \infty \wedge \sim q(\xi, y)] \wedge \exists y [y \in F_2^+(\xi) \wedge y \neq \infty \wedge q(\xi, y)] \} \\
& \Leftrightarrow \exists y_1 \exists y_2 [y_1 \in F_1^+(\xi) \wedge y_1 \neq \infty \wedge y_2 \in F_2^+(\xi) \wedge y_2 \neq \infty \wedge y_1 \neq y_2] \\
& \Leftrightarrow \sim \forall y_1 \forall y_2 [y_1 \in F_1^+(\xi) \wedge y_2 \in F_2^+(\xi) \wedge y_1 \neq \infty \wedge y_2 \neq \infty \supset y_1 = y_2] .
\end{aligned}$$

$$\begin{aligned}
(b) \quad & \sim \forall q [\sim W_1^H(\xi, q) \vee \sim W_2^H(\xi, \sim q)] \Leftrightarrow \exists q [W_1^H(\xi, q) \wedge W_2^H(\xi, \sim q)] \\
& \Leftrightarrow \exists q \{ \exists y \{ y \in F_1^+(\xi) \wedge [y \neq \infty \supset q(\xi, y)] \} \wedge \exists y \{ y \in F_2^+(\xi) \wedge [y \neq \infty \supset \sim q(\xi, y)] \} \} \\
& \Leftrightarrow \exists y_1 \exists y_2 [y_1 \in F_1^+(\xi) \wedge y_2 \in F_2^+(\xi) \wedge [y_1 \neq \infty \wedge y_2 \neq \infty \supset y_1 \neq y_2]] \\
& \Leftrightarrow \sim \forall y_1 \forall y_2 [y_1 \in F_1^+(\xi) \wedge y_2 \in F_2^+(\xi) \supset y_1 \neq \infty \wedge y_2 \neq \infty \wedge y_1 = y_2] .
\end{aligned}$$

$$\begin{aligned}
7.(a) \quad & \sim \forall q [W_1^V(\xi, q) \supset W_2^V(\xi, q)] \Leftrightarrow \exists q [W_1^V(\xi, q) \wedge \sim W_2^V(\xi, q)] \\
& \Leftrightarrow \exists q \{ \forall y [y \in F_1^+(\xi) \wedge y \neq \infty \supset q(\xi, y)] \wedge \exists y [y \in F_2^+(\xi) \wedge y \neq \infty \wedge \sim q(\xi, y)] \} \\
& \Leftrightarrow \exists y [y \in F_2^+(\xi) \wedge y \neq \infty \wedge y \notin F_1^+(\xi)] \\
& \Leftrightarrow \sim \forall y [y \in F_2^+(\xi) \wedge y \neq \infty \supset y \in F_1^+(\xi)] .
\end{aligned}$$

$$\begin{aligned}
(b) \quad & \sim \forall q [W_2^H(\xi, q) \supset W_1^H(\xi, q)] \Leftrightarrow \exists q [W_2^H(\xi, q) \wedge \sim W_1^H(\xi, q)] \\
& \Leftrightarrow \exists q \{ \exists y [y \in F_2^+(\xi) \wedge [y \neq \infty \supset q(\xi, y)]] \wedge \forall y [y \in F_1^+(\xi) \supset y \neq \infty \wedge \sim q(\xi, y)] \} \\
& \Leftrightarrow \exists y [y \in F_2^+(\xi) \wedge y \notin F_1^+(\xi)] \\
& \Leftrightarrow \sim \forall y [y \in F_2^+(\xi) \supset y \in F_1^+(\xi)] .
\end{aligned}$$

8.(a) Follows from 7(a).

(b) Follows from 7(b).

Q.E.D.

Example

The construction of $W^V(x,q)$ and $W^H(x,q)$ has already been described for several classes of non-deterministic algorithms, such as non-deterministic programs (Manna [9]) and parallel programs (Ashcroft and Manna [2]).

We shall illustrate the construction of $W^V(x,q)$ and $W^H(x,q)$ for a non-deterministic program computing the factorial function.

In the program below a branch of the form



is called a choice branch and means that upon execution of the program, at this point we are allowed to proceed with either branch, chosen arbitrarily. The execution of the program proceeds until $z_2 = z'_2$; then $y = z_1 \cdot z'_1$.

For $x = 3$, for example, there are 30 different possible executions of the program: 5 of them are represented by table 1 below, 10 by table 2, 10 by table 3, and 5 by table 4.

z_1	z_2	z'_1	z'_2
1	3	1	0
3	2		
3·2	1		
3·2·1	0		

table 1

z_1	z_2	z'_1	z'_2
1	3	1	0
3	2	1	1
3·2	1		

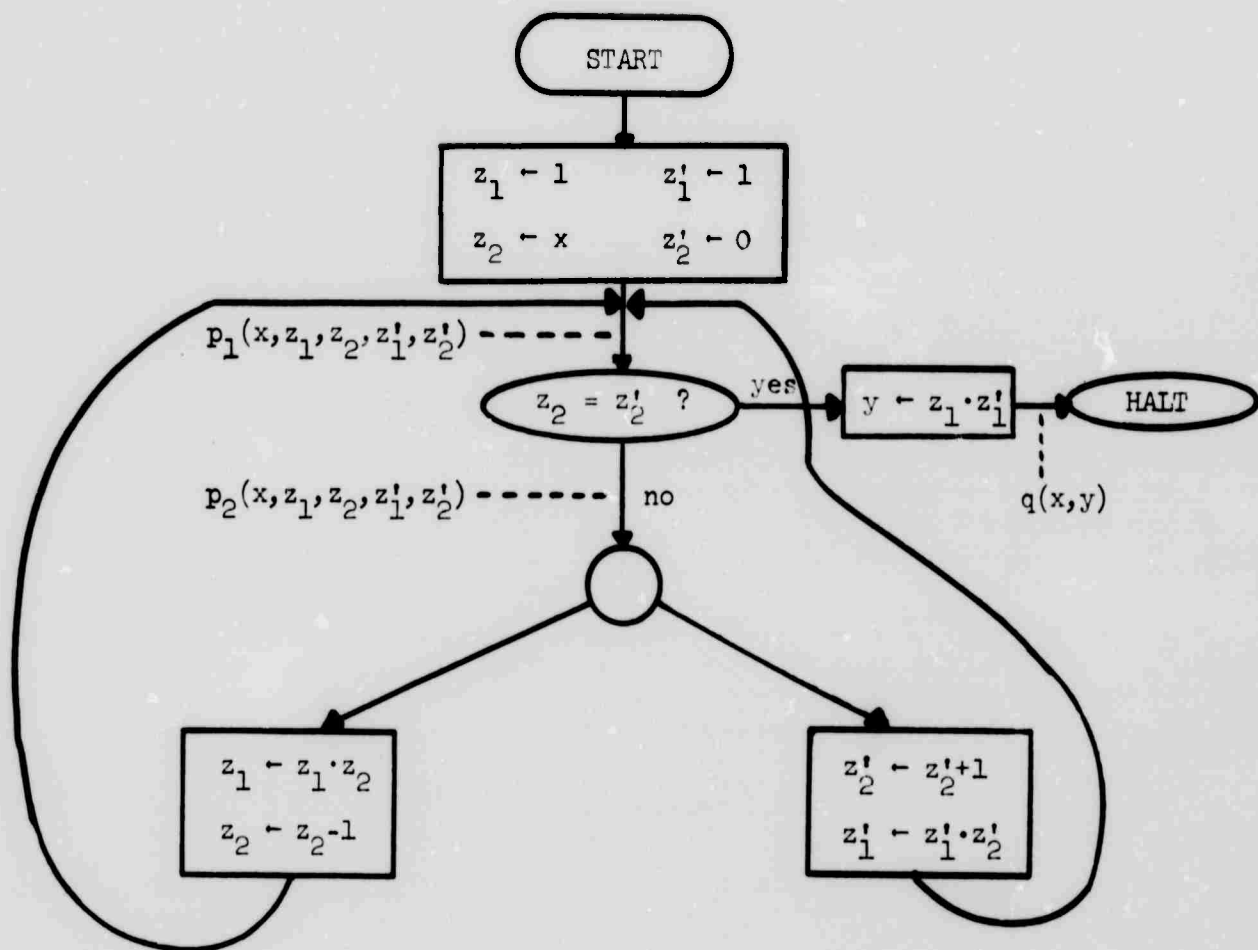
table 2

z_1	z_2	z'_1	z'_2
1	3	1	0
3	2	1	1
		1·2	2

table 3

z_1	z_2	z'_1	z'_2
1	3	1	0
		1	1
		1·2	2
		1·2·3	3

table 4



$W^{\exists}(x, q)$ is

$\exists p_1 \exists p_2 \{ p_1(x, 1, x, 1, 0)$

$\wedge \forall z_1 \forall z_2 \forall z'_1 \forall z'_2 [p_1(x, z_1, z_2, z'_1, z'_2) \supset \text{if } z_2 = z'_2 \text{ then } q(x, z_1 \cdot z'_1) \\ \text{else } p_2(x, z_1, z_2, z'_1, z'_2)]$

$\wedge \forall z_1 \forall z_2 \forall z'_1 \forall z'_2 [p_2(x, z_1, z_2, z'_1, z'_2) \supset p_1(x, z_1 \cdot z_2, z_2 - 1, z'_1, z'_2) \vee \\ p_1(x, z_1, z_2, z'_1 \cdot (z'_2 + 1), z'_2 + 1)] \} .$

$W^{\forall}(x, q)$ is similar, with the ' $\vee$ ' connective replaced by ' $\wedge$ '.

COMMENTS

1. There are clearly many natural extensions of our results. We shall present here just one example.

Let f_1 and f_2 be any two comparable partial functions, and let $TP(*f_1(x), *f_2(x))$ called the termination property, be any formula constructed from primitives $*f_1(x)$ and $*f_2(x)$ and propositional connectives $\sim, \supset, \wedge, \vee$ and $\equiv$.

We say that $f_1(\xi)$ and $f_2(\xi)$ are equivalent w.r.t. TP if $TP(*f_1(\xi), *f_2(\xi)) \wedge [*f_1(\xi) \wedge *f_2(\xi) \supset f_1(\xi) = f_2(\xi)]$; i.e., if $f_1(\xi)$ and $f_2(\xi)$ satisfy the termination property TP and if $f_1(\xi)$ and $f_2(\xi)$ are defined, then $f_1(\xi) = f_2(\xi)$.

By specifying TP we obtain as special cases all the notions of equivalence introduced in Part I (2(a)-(d)): (a) partial equivalence (TP is T), (b) extension (TP is $*f_2(x) \supset *f_1(x)$), (c) equivalence (TP is $*f_1(x) \equiv *f_2(x)$), and (d) total equivalence (TP is $*f_1(x) \wedge *f_2(x)$).

The following result follows from Theorem 1 (0 and 2(a)):

Theorem: $f_1(\xi)$ and $f_2(\xi)$ are equivalent w.r.t. TP if and only if $TP(\sim w_1(\xi, \mathcal{F}), \sim w_2(\xi, \mathcal{F})) \wedge \forall q[w_1(\xi, q) \vee w_2(\xi, \sim q)]$.

Thus the theorem gives second-order formulas for the above four properties by appropriate substitutions for TP. However, Theorem 1 (2(a)-(d)) gives simpler second-order formulas for the same properties.

Similarly, one can extend the notions of correctness. In general, any property can be formalized in second-order predicate calculus, if it can be expressed as a composition of some of the basic formulas (that were formalized in our theorems) using propositional connectives ($\sim$, $\wedge$, $\vee$, $\supset$ and $\equiv$). The appropriate second-order formula is then the propositional composition of the corresponding basic second-order formulas. This, for example, was the way we formalized several properties in Part II.

2. Note that among the 'equivalence properties' defined in Part I, only equivalence, i.e., property 2(c), is really an equivalence relation (i.e., reflexive, symmetric and transitive) as can be seen from the following table:

property	other names used in publications	reflexive relation	symmetric relation	transitive relation	equivalence relation
partial equivalence	weak equivalence	yes	yes	no	no
extension equivalence	inclusion	yes	no	yes	no
	strong equivalence	yes	yes	yes	yes
total equivalence	(termination) equivalence	no	yes	yes	no

Among the 'equivalence properties' defined in Parts II and III only property 7(b) in Part II and properties 8(a)(b) in Part III are equivalence relations.

3. Our results imply that the second-order formula formalizing the 'partial correctness' of a given algorithm represents, in some sense, all input-output relations of the computations of the algorithm.

In general, all our results hold even if the formulas w , W and $W^E - W^V$ formalize partial correctness in the following weaker sense:

For every $\xi \in D_x$ and for every predicate $\psi(x, y)$ over $D_x \times D_y$.

- (i) $w(\xi, \psi)$ if and only if $\exists q \{ [*f(\xi) \supset q(\xi, f(\xi))] \wedge \forall y [q(\xi, y) \supset \psi(\xi, y)] \}$;
- (ii) $W(\xi, \psi)$ if and only if $\exists q \{ \forall y [y \in F(\xi) \supset q(\xi, y)] \wedge \forall y [q(\xi, y) \supset \psi(\xi, y)] \}$;
- (iii) $W^E(\xi, \psi)$ if and only if $\exists q \{ \forall y [y \in F^+(\xi) \wedge [y \neq \infty \supset q(\xi, y)] \} \wedge \forall y [q(\xi, y) \supset \psi(\xi, y)] \}$

and

- $W^V(\xi, \psi)$ if and only if $\exists q \{ \forall y [y \in F^+(\xi) \wedge y \neq \infty \supset q(\xi, y)] \wedge \forall y [q(\xi, y) \supset \psi(\xi, y)] \}$.

4. All the properties mentioned so far were defined and formalized for fixed input values. One can extend all the definitions and the corresponding formulas to hold over some total input predicate $\varphi(x)$, which means that the property should hold for every $\xi \in D_x$ s.t. $\varphi(\xi) = T$. More precisely, if property P for $\xi \in D_x$ was formalized by $W_P(\xi)$, then the property P holds over input predicate $\varphi(x)$ if and only if $\forall x [\varphi(x) \supset W_P(x)]$.

5. The formulas w , W and $W^H - W^V$ constructed in previous publications for various classes of algorithms share an important common feature: all additional predicate symbols introduced in the formulas are existentially quantified (see examples above). This is because the additional symbols were always introduced for the same purpose, namely to cut the algorithm into pieces which can be formalized directly.

In this case certain properties happen to be formalized by first-order formulas, (i.e., all predicate symbols are universally quantified); for example, properties 0, 1(b) and 2(d) of Part I, 1, 2(b), 3(b) and 5(b) of Part II, and 1(a), 1(b), 2(b), 3(b), 4(b), 5(b) and 6(b) of Part III.

Acknowledgments

I am indebted to EDWARD ASHCROFT and STEPHEN NESS for many stimulating discussions and also for their critical reading of the manuscript and subsequent helpful suggestions.

References

- [1] E. A. ASHCROFT, "Mathematical Logic Applied to the Semantics of Computer Programs", Ph.D. Thesis, to be submitted to Imperial College, London (1970).
- [2] E. A. ASHCROFT and Z. MANNA, "Formalization of Properties of Parallel Programs", Computer Science Department, Stanford University, Artificial Intelligence Memo AIM-110 (February 1970).
- [3] R. M. BURSTALL, "Formal Description of Program Structure and Semantics in First-Order Logic", in Machine Intelligence 5 (Eds. Meltzer and Michie), Edinburgh University Press, 79-98 (1970).
- [4] D. C. COOPER, "Program Scheme Equivalences and Second-Order Logic", in Machine Intelligence 4 (Eds. Meltzer and Michie), Edinburgh University Press, 3-15 (1969).
- [5] D. C. COOPER, "Program Schemes, Programs and Logic", Computation Services Department, University College of Swansea, Memo No. 6 (April 1969).
- [6] R. W. FLOYD, "Assigning Meaning to Programs", in Proceedings of Symposia in Applied Mathematics, American Mathematical Society, Vol. 19, 19-32 (1967).
- [7] Z. MANNA, "The Correctness of Programs", J. of Computer and System Sciences, Vol. 3, No. 2, 119-127 (1969).
- [8] Z. MANNA, "Properties of Programs and the First-Order Predicate Calculus", JACM, 244-255 (1969).
- [9] Z. MANNA, "The Correctness of Non-deterministic Programs", Artificial Intelligence J., Vol. 1, No. 1 (1970).
- [10] Z. MANNA and A. PNUELI, "Formalization of Properties of Functional Programs", ACM Symposium on Theory of Computing (May 1969). To appear in the JACM (July 1970).
- [11] D. PARK, "Fixpoint Induction and Proofs of Program Properties", in Machine Intelligence 5 (Eds. Meltzer and Michie), Edinburgh University Press, 59-78 (1970).

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Stanford Artificial Intelligence Project
Computer Science Department
Stanford University

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

Second-Order Mathematical Theory of Computation

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

Zohar Manna

6. REPORT DATE

March, 1970

7a. TOTAL NO. OF PAGES

25

7b. NO. OF REFS

11

8a. CONTRACT OR GRANT NO.

d. PROJECT NO. ARPA SD-183

8b. ORIGINATOR'S REPORT NUMBER(S)

AIM-111

8c. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Statement No. 1 - Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

In this work we show that it is possible to formalize all properties regularly observed in (deterministic and non-deterministic) algorithms in second-order predicate calculus.

Moreover, we show that for any given algorithm it suffices to know how to formalize its "partial correctness" by a second-order formula in order to formalize all other properties by second-order formulas.

This result is of special interest since "partial correctness" has already been formalized in second-order predicate calculus for many classes of algorithms.

This paper will be presented at the ACM Symposium on Theory of Computing (May, 1970).