MATHEMATICAL CENTRE TRACTS 24

J.W. DE BAKKER

RECURSIVE PROCEDURES

MATHEMATISCH CENTRUM AMSTERDAM 1973

AMS (MOS) subject classification scheme (1970): 68A05
ACM - Computing Reviews - category: 5.24

1st printing: 1971

2nd printing: 1973

ACKNOWLEDGEMENTS

I am deeply indebted to Professor Dana Scott, whose work forms the basis for these investigations.

To J.D. Alanen and W.P. de Roever I am grateful for their critical reading of the manuscript. Their comments resulted in an improved presentation in many cases.

I also want to thank Mrs. S.J. van den Bergh-Koerts for her excellent typing, and Mr. J. Suiker and Mr. D. Zwarst for the printing of this Tract.

CONTENTS

1. INTRODUCTION

1.1. GENERAL

This monograph is devoted to the study of recursive procedures. After clarification of some of their mathematical properties, a proof technique is developed for showing equivalence of programs containing recursive procedures, and a number of properties of this technique are derived.

Our study originated with some attempts to develop a generalization of McCarthy's rule of recursion induction [23], which rule can be summarized as follows: If one has two functions g and h, which both satisfy the defining equation of a recursively defined function f, then $g(x) = h(x)$ for all those x, for which $f(x)$ is defined. E.g., if one has for f the recursive definition

$$f(x) \Leftarrow (x=0 \rightarrow 1, x \times f(x-1))$$

where "$\Leftarrow$" stands for "is recursively defined by", and where in the right hand side we have used McCarthy's notation for conditional expressions [23], and if one has established by some means that g and h satisfy

$$g(x) = (x=0 \rightarrow 1,\ x \times g(x-1))$$
$$h(x) = (x=0 \rightarrow 1,\ x \times h(x-1))$$

then one concludes that $g(x) = h(x)$ for all x for which $f(x)$ is defined.

At first glance one might consider this a rather obvious rule. However, when one tries to justify it mathematically, some problems arise. In particular one is confronted with the question as to what is the precise status of the "$\Leftarrow$" relationship, and what is the connection between the "$\Leftarrow$" and "=" relations. Note that it is certainly not the case that "$\Leftarrow$" and "=" are the same relations. E.g., each function f satisfies $f(x) = f(x)$, but it is clearly not true that for each f, we have $f(x) \Leftarrow f(x)$, since if f were defined recursively in this way, then f would be undefined for all x.

We were led to the consideration of these problems when we tried to apply recursion induction to two problems: An investigation of Yanov's axiomatization of the equivalence of his "logical schemes of algorithms" [39], and an axiomatic treatment of the equivalence of while statements. Although we obtained some partial results, our framework remained unsatisfactory, precisely because of this problem about the meaning of "$\Leftarrow$". The solution to this was provided by Scott, who both showed what mathematical objects correspond to recursive procedures, and how to exploit this insight in the development of a formal system in which a proof rule which generalizes recursion induction plays a central role.

We can therefore summarize the contents of our paper as: The description, application and investigation of Scott's mathematical theory of recursive procedures. (This theory was described for the first time in the unpublished notes [32].)

Scott's approach is based on the following central idea: Recursive procedures are *minimal fixed points* of *monotonic* and *continuous* transformations. This way of looking at procedures is explained in section 2. In this section we shall take as a starting point an ALGOL-like simple programming language, and we use this as a tool to clarify what the mathematical properties of recursive procedures are. (The functional notation we used in this introduction will be used only occasionally in the sequel, since the imperative notation of section 2 provides a better transition to the formal system to be developed in section 3.)

The notion of *minimal* fixed point apparently presupposes some partial ordering, and the use of this ordering, together with the notions of monotonicity and continuity, is essential to Scott's theory.

These notions have in fact a much wider scope in the theory of computation than just the theory of procedures, as Scott has shown in a number of subsequent papers. For a general treatment see [34], for other applications [35] and [36].

Once it had been clarified what procedures "really are", it became possible to develop a formal system based on this insight, namely Scott's μ-calculus, which we describe in section 3. The core of this formal system is the μ-induction rule, which is a generalization of recursion induction.

In section 4, we apply the μ-calculus to a large number of examples, partly taken from various places in the literature, partly new. We hope to demonstrate in this section that the μ-calculus can be used to prove results taken from rather divergent sources (e.g. various properties of while statements, McCarthy's 91-function) and, thus, to convince the reader of its general scope.

The main result of our paper follows in sections 5 and 6. We prove the completeness of the μ-calculus for a restricted type of procedures, viz. the "regular procedures", which can be considered as corresponding to flow diagrams. The proof is given in two parts. In section 5, we introduce a normal form for this restricted class of procedures, and prove that each regular procedure is equivalent to a regular procedure in normal form. In section 6 we show that two regular procedures, which may be assumed to be in normal form by section 5, can be proved to be equivalent in the μ-calculus if and only if they are semantically equivalent (i.e., they denote the same functions under all interpretations).

Section 7 contains some conclusions. We mention some ways in which the μ-calculus has already been extended, and areas which may be of interest for future investigation.

We have tried to make our paper more or less self contained. However, we recommend the reader who is not familiar with the problem area of the mathematical theory of computation to read first the classical papers by McCarthy [22,23].

1.2. RELATED WORK

Recursion induction was introduced by McCarthy [23]. It has been applied and generalized in a number of papers, e.g. Cooper [4,5], and Kaplan [14]. A variant is Burstall's structural induction [3]. Some steps towards Scott's theory were made independently by Morris [25,26].

The relationship between procedures and fixed points (expressed via Curry's Y-operator) has been more or less well known for some time. See e.g. Landin [16] and Strachey [37]. However, the minimality of the fixed points was not exploited there. Minimality considerations, including the application of Knaster's result (see section 2) were introduced independently by Park [27] and Bekić [1]; Bekić also obtained the simultaneous versus iterated fixed point property, stated in section 2. The Y-operator in relation to minimal fixed points is discussed in Scott [35]; for a comment see Park [28].

For the results of section 5 and 6 compare Yanov [39], Kaplan [13] and Ito [12], who uses Salomaa's complete axiom system for regular events [30].

It may be of interest to clarify the relation between our normal form and the one defined by Engeler [9].

Another attempt at an axiomatic treatment of recursive procedures is given in Hoare [11], who uses "pre-Scott" notions in his statement of the induction rule.

A rather different approach is taken by a number of authors who base themselves on the predicate calculus. This work originated with Floyd [10], and was developed further by Manna, e.g. [17,18] and Cooper [6,7]. See also the "relational theory" in Scott and de Bakker [32].

For references to some extensions of the μ-calculus see section 7.

2. RECURSIVE PROCEDURES

In section 3, we shall present a formalism for the treatment of recursive procedures: Scott's μ-calculus. In this section, we give an intuitive explanation of some of the notions to be used later, and derive some of their properties.

We shall be concerned with programs in a simple language. This language comprises first of all a class of elementary statements, the structure of which is not analyzed in the present context. Next, it has three constructions to build up more complex statements from simpler ones, starting with the elementary statements. Two of these are straightforward: The *composition* $S_1;S_2$ of two statements S_1 and S_2, and the construction of the *conditional* statement $(p \to S_1, S_2)$, where p is some boolean expression. (Throughout the paper, we shall use McCarthy's notation [23] for conditionals, in which $(p \to S_1, S_2)$ is short for <u>if</u> p <u>then</u> S_1 <u>else</u> S_2.) The third construction is that of, possibly recursive, *procedures* which are introduced by means of declarations of the form

(1) <u>procedure</u> P; T(P)

where the procedure body T(P) is some statement which may contain one or more occurrences of P.

As an instance of (1), we have

(2) <u>procedure</u> P; $(p \to A_1;P, A_2)$

where A_1, A_2 are elementary statements.

The reader who is more used to a functional notation may read this as the declaration of the function $P(\sigma)$ by means of the definition

$$P(\sigma) \Leftarrow (p(\sigma) \to P(A_1(\sigma)), A_2(\sigma))$$

where "$\Leftarrow$" stands for "is recursively defined by". This notation emphasizes the *functional* aspect of statements: A statement S may be considered as prescribing a mapping from a state σ of the computation to a new state σ', or, in functional notation, $S(\sigma) = \sigma'$. For our purposes, we are not interested in the structure of the state. If one wishes to be more specific, one may think of the state as consisting of all variables manipulated by the program. This is the approach taken by McCarthy in [22] with his state vectors: E.g., let the elementary statement A be the assignment statement $x := x+1$, let σ be the state vector, the x-th component of which has the value 0, say. Then, the new state vector $\sigma' = A(\sigma)$ is equal to σ in all its components, except for the x-th component which now has the value 1. However, such articulation of the state vector is not considered in our paper. All we need is the fact that each statement S determines some functional relationship between σ and σ'.

The functional approach to statements gives us a way of expressing equivalence between two statements: S_1 and S_2 are equivalent if and only if the functions associated with them are the same, i.e., iff $S_1(\sigma) = S_2(\sigma)$, for all σ.

The next step is the realization of the fact that the function determined by some S may be *partial*, i.e., it may be undefined for certain arguments. E.g., the statement L: goto L (using an example outside our language) has an undefined effect for all σ, i.e., there is no σ' such that (L:gotoL) $(\sigma) = \sigma'$. Also, the statement $(p\rightarrow L$:gotoL$,A)$ is undefined for all those σ for which $p(\sigma)$ happens to be true.

Once one has observed this, it becomes natural to introduce, besides the relation "=" of *equivalence* between statements, a second relationship of *inclusion*, $S_1 \subseteq S_2$, meaning that for all those σ for which S_1 is defined, S_2 is also defined, and yields the same answer. Formally,

$$S_1 \subseteq S_2 \text{ iff } [S_1(\sigma) = \sigma' \Longrightarrow S_2(\sigma) = \sigma'] .$$

By the introduction of the notion of *graph* of a function, we can give another formulation of the "$\subseteq$" relation:

The graph of the function S is the set of all pairs (σ,σ') such that $\sigma' = S(\sigma)$. Note that

a. The graph of L:gotoL is the empty set.

b. The graph of $S_1;S_2$ is the set of all pairs (σ,σ') such that there exists $\bar{\sigma}$ with $S_1(\sigma) = \bar{\sigma}$ and $S_2(\bar{\sigma}) = \sigma'$.

c. The graph of $(p \to S_1, S_2)$ is the set of all (σ, σ') such that either

(α) $p(\sigma)$ is true, and $S_1(\sigma) = \sigma'$

or

(β) $p(\sigma)$ is false, and $S_2(\sigma) = \sigma'$.

Thus, $S_1 \subseteq S_2$ can be phrased as: The graph of S_1 is included in the graph of S_2.

As examples of "$\subseteq$" we have:

a. L:goto L $\subseteq$ S , for each S;

b. $(p \to$ L:goto L$, S) \subseteq S$, for each S.

It is clear that "$\subseteq$" determines a partial ordering between statements, which means in particular that "$\subseteq$" is reflexive and transitive, and that $S_1 \subseteq S_2$ and $S_2 \subseteq S_1$ together imply $S_1 = S_2$.

The introduction of the "=" and "$\subseteq$" relations between statements gives us the tools to discuss procedures in a proper framework. Consider again the procedure declaration

(3) procedure P; T(P) .

According to the usual meaning of procedure declarations, one may replace each call of P, as declared by (3), in the program by its body T(P); i.e., one uses the fact that

(4) $P = T(P)$.

For our purposes, it is convenient to consider T as a transformation of statements to statements, or, in other words, as a functional, which has functions as arguments and values: For each statement S, T(S) yields another statement S'. E.g., if we take the instance of T in (2), then $T(S) = (p \to A_1; S, A_2) = S'$. In this terminology, we can now formulate (3) as: *P is a fixed point of the transformation T*. This fixed point property is the first basic fact to be noted about procedures. That this is not the whole story will be explained by another example, where for the transformation T we have taken the identity transformation:

(5) <u>procedure</u> $P_1;P_1$.

Of course, it is again true that

(6) $P_1 = P_1$

i.e., (4) is satisfied. However, we expect more from (5): If P_1 is called in the program, it gives the infinite loop, i.e., it has the same effect as L:<u>goto</u>L, say. Let us give this infinite loop a name: Ω. Then, we may say that (6) is not all we want from (5): we want that $P_1 = \Omega$. Now observe that all statements are fixed points of the identity transformation ($S=S$ holds for all S), but that Ω is distinguished among all those statements by being the *minimal* statement satisfying (6), where minimal is meant with respect to the partial ordering "$\subseteq$". This follows from $\Omega \subseteq S$ for all S. Thus, we have now some evidence that a procedure P, as declared by (3) in the general case, has to be not only a *fixed point* of T, but even its *minimal fixed point*. Some more evidence is obtained by considering

(7) <u>procedure</u> P_2; $(p\rightarrow P_2,A)$.

Again, we have a whole class of statements satisfying the functional equation

$$P_2 = (p\rightarrow P_2,A)$$

viz., all statements of the form $(p\rightarrow S,A)$, for arbitrary S. This follows from the equivalence of $(p\rightarrow S,A)$ and $(p\rightarrow(p\rightarrow S,A),A)$, which can easily be seen to hold as a result of the properties of conditional statements. Again, we choose the minimal element in this class: the statement $(p\rightarrow\Omega,A)$, which satisfies $(p\rightarrow\Omega,A) \subseteq (p\rightarrow S,A)$, for all S. This choice correspond to our idea that (7) declares a procedure which loops when called at a moment when the state σ is such that $p(\sigma)$ is true.

These two examples may illustrate the general argument for the minimal fixed point approach: Consider again the procedure P declared by (3). By (4), the graph of P must be the same as the graph of $T(P)$. However, according to the usual meaning of procedures, there is no reason to expect this graph to contain more pairs than are necessary in order that it satisfy this equality. (That it is always possible to achieve equality will be proved below.)

Thus, we have as the first central idea of our paper:

Procedures are minimal fixed points of the transformations given by the body of their declaration.

Next, we introduce another important aspect of our system, namely that the transformations T are monotonic with respect to $\subseteq$:

(8) If $S_1 \subseteq S_2$, then $T(S_1) \subseteq T(S_2)$.

For the simplest transformations, this will be clear. Consider e.g. the following transformations:

a. $T(S) = S$,
b. $T(S) = A;S$, for some fixed A,
c. $T(S) = (p \to S, S')$, for some fixed S'.

For these cases, (8) reads:

a. If $S_1 \subseteq S_2$, then $S_1 \subseteq S_2$,
b. If $S_1 \subseteq S_2$, then $A;S_1 \subseteq A;S_2$,
c. If $S_1 \subseteq S_2$, then $(p \to S_1, S') \subseteq (p \to S_2, S')$.

It is easily verified that these assertions are valid.

A somewhat more complicated instance of (8) is provided by the following example:

<u>procedure</u> P; T(S,P) .

Here the procedure P depends on S, i.e., $P = P(S)$. Hence, P can be considered as a transformation on S. In this case, (8) becomes: If $S_1 \subseteq S_2$, then $P(S_1) \subseteq P(S_2)$. We shall at this stage not give the detailed justification of this and similar assertions; this will be postponed to section 3. However, we can already introduce a first important consequence of the monotonicity property.

First we introduce some notation. In the sequel, we will be interested in the greatest lower bound (g.l.b.) and the least upper bound (l.u.b.) of families of functions, with respect to the partial ordering

"$\subseteq$". The greatest lower bound of a family $\mathcal{F}$ will be denoted by $\bigcap\{F:F\in\mathcal{F}\}$. This notation is chosen since it is clear that the graph of the greatest lower bound of $\mathcal{F}$ is the intersection of the graphs of the functions $F \in \mathcal{F}$. Similarly, for the least upper bound we write $\bigcup\{F:F\in\mathcal{F}\}$. Note that the l.u.b. does not necessarily exist: If F_1 and F_2 are such that for some σ, $F_1(\sigma) = \sigma_1$, $F_2(\sigma) = \sigma_2$, and $\sigma_1 \neq \sigma_2$, then there does not exist a function F such that both $F_1 \subseteq F$ and $F_2 \subseteq F$. However, if the l.u.b. of $\mathcal{F}$ does exist then, clearly, its graph is the union of the graphs of the functions $F \in \mathcal{F}$.

Now consider the family of all functions X such that $X = T(X)$. We are interested in its l.u.b. $F = \bigcap\{X:X=T(X)\}$. We shall prove that F itself is a fixed point of T (and, hence, its minimal fixed point), i.e., that $T(F) = F$. This will follow from the following, more general result: Let $G = \bigcap\{X:T(X)\subseteq X\}$. We show that

$$(9) \qquad F = \cap\{X:T(X)=X\} = \cap\{X:T(X)\subseteq X\} = G \ .$$

In this proof, the monotonicity of T will be essential:

1. $G \subseteq F$ is clear.
2. Proof of $F \subseteq G$. It is sufficient to show that $T(G) = G$.
   - 2a. Proof of $T(G) \subseteq G$. Let X be such that $T(X) \subseteq X$. Then $G \subseteq X$, and, by monotonicity, $T(G) \subseteq T(X)$; thus, $T(G) \subseteq X$. We see that $T(G)$ is included in all X such that $T(X) \subseteq X$; hence, $T(G)$ is included in the l.u.b. of all such X, i.e., $T(G) \subseteq G$.
   - 2b. Proof of $G \subseteq T(G)$. Since $T(G) \subseteq G$, we have $T(T(G)) \subseteq T(G)$. But, G is minimal with this property; hence, $G \subseteq T(G)$.

That $F = G$ is in fact an old result in set theory, cf. Knaster [15] and Tarski [38].

We are now in a position to explain recursion induction. Remember that recursion induction can be phrased as follows: Let the procedure P_1 be defined by

(10) <u>procedure</u> $P_1;T(P_1)$

and let P_2, P_3 be such that

$$(11) \qquad \begin{aligned} P_2 &= T(P_2) \\ P_3 &= T(P_3) \end{aligned} \quad .$$

Then, by recursion induction, we conclude that $P_2(\sigma) = P_3(\sigma)$ for all those σ for which $P_1(\sigma)$ is defined. The argument for this is the following: By (10), P_1 is the least function satisfying $P_1 = T(P_1)$. Hence, by (11), $P_1 \subseteq P_2$ and $P_1 \subseteq P_3$. By the definition of $\subseteq$, we have: For all σ for which $P_1(\sigma)$ is defined, $P_1(\sigma) = P_2(\sigma)$, and $P_1(\sigma) = P_3(\sigma)$. Thus we obtain that, for these σ, $P_2(\sigma) = P_3(\sigma)$.

The transformations T are not only monotonic, they are also *continuous* in a sense which we shall make precise presently. The introduction of the notion of continuity arises from another way of looking at the minimal fixed point property of procedures. This other approach is inspired by the construction of the minimal fixed point of a monotonic and continuous real function f, from [0,1] to [0,1], say. The minimal fixed point x_0 of f can be found as

$$x_0 = \lim_{i\to\infty} f^i(0)$$

where $f^i(0) = \underbrace{f(f(\ldots f(0)}_{i \text{ times } f} \ldots))$.

That x_0 is a fixed point follows from

$$f(x_0) = f(\lim_{i\to\infty} f^i(0)) = \lim_{i\to\infty} f(f^i(0)) = \lim_{i\to\infty} f^i(0) = x_0 .$$

x_0 is also minimal: Let x_1 be such that $f(x_1) = x_1$. Then

$$\begin{aligned} 0 &\leq x_1 \\ f(0) &\leq f(x_1) = x_1 \\ f^2(0) &\leq f^2(x_1) = x_1 \\ &\vdots \\ x_0 = \lim_{i\to\infty} f^i(0) &\leq x_1 \quad . \end{aligned}$$

The minimal fixed point of a transformation T can in a similar way be obtained by approximation from below. Consider the sequence

(12) $\Omega, T(\Omega), T^2(\Omega),\dots,T^i(\Omega),\dots$.

Note that, by monotonicity,

(13) $\Omega \subseteq T(\Omega) \subseteq T^2(\Omega) \subseteq \dots \subseteq T^i(\Omega) \subseteq \dots$.

The sequence of partial functions (12) has a l.u.b. $P_0 = \bigcup_{i=0}^{\infty} T^i(\Omega)$. The existence of this l.u.b. for (12) is a result of (13). We assert that P_0 is the minimal fixed point we are looking for. Clearly, the above proof for $x_0 = \lim_{i\to\infty} f^i(0)$ can be taken over, provided that we have given a proper meaning to the continuity for T. This is done as follows:

T is called continuous if, for each sequence

$$X_0 \subseteq X_1 \subseteq X_2 \subseteq \dots \subseteq X_i \subseteq \dots$$

we have

(14) $T(\bigcup_{i=0}^{\infty} X_i) = \bigcup_{i=0}^{\infty} T(X_i)$.

Just as with our assertion on the monotonicity of T, (14) can easily be verified for simple T, involving only composition and conditionals, but some further argument is needed for those T which themselves involve procedures. Again, a further discussion of this is postponed to section 3.

Using (14) as definition of continuity, we thus have the result that for monotonic and continuous T, its minimal fixed point is given by $\bigcup_{i=0}^{\infty} T^i(\Omega)$.

The result that for the procedure declared by

procedure P; T(P)

we have

$$P = \bigcup_{i=0}^{\infty} T^i(\Omega)$$

can also be explained in a somewhat different way. We have to justify the two inclusions

$$\bigcup_{i=0}^{\infty} T^i(\Omega) \subseteq P$$

$$P \subseteq \bigcup_{i=0}^{\infty} T^i(\Omega) \ .$$

The argument for the first inclusion is the same as above: We have

$$\begin{array}{rcl} \Omega & \subseteq & P \\ T(\Omega) & \subseteq & T(P) = P \\ & \vdots & \\ T^i(\Omega) & \subseteq & T^i(P) = P \\ & \vdots & \end{array}$$

Hence, for the l.u.b. of the family $\{\Omega, T(\Omega), \ldots, T^i(\Omega), \ldots\}$ we have

$$\bigcup_{i=0}^{\infty} T^i(\Omega) \subseteq P$$

Next, we consider the second inclusion: $P \subseteq \bigcup_{i=0}^{\infty} T^i(\Omega)$. According to the definition of "$\subseteq$", this means that, for all σ, if $P(\sigma)$ is defined, say $P(\sigma) = \sigma'$, then $(\bigcup_{i=0}^{\infty} T^i(\Omega))\ (\sigma) = \sigma'$.

What does it mean that $P(\sigma)$ is defined? In order to explain this, we look at the way in which the programmer determines $P(\sigma)$. He applies a rule of *rewriting*, and in order to determine $P(\sigma)$, he tries to determine $(T(P))\ (\sigma)$. Executing the procedure body for this argument may lead to another "inner call" of P, which then, in turn, is rewritten as T(P), etc. Eventually, this rewriting process must come to an end, since, otherwise, σ' would not have been obtained as a result. This can be expressed by saying that there exists an $i \geq 1$ such that, in the application of $T^i(P)$

to the given σ, P is not encountered anymore. Because of this we may as well replace P in $T^i(P)$ everywhere by Ω, i.e., we use the fact that, for this σ, $(T^i(P))\ (\sigma) = (T^i(\Omega))\ (\sigma) = \sigma'$. Then, clearly, $(\bigcup\limits_{i=0}^{\infty} T^i(\Omega))\ (\sigma) = \sigma'$, which is the desired result.

Example: Let $T(P) = (p\to A_1;P,A_2)$, and let σ be such that $p(\sigma)$ and $p(A_1(\sigma))$ are true, but $p(A_1(A_1(\sigma)))$ is false. Then

$$\begin{aligned} P(\sigma) &= (p\to A_1;(p\to A_1;(p\to A_1;P,A_2),A_2),A_2)\ (\sigma) = \\ &\ (p\to A_1;(p\to A_1;(p\to A_1;\Omega,A_2),A_2),A_2)\ (\sigma) = \\ &\ A_2(A_1(A_1(\sigma))) = \sigma'. \end{aligned}$$

Finally, we make some remarks on systems of procedures. For explanatory purposes, it is sufficient to consider a system of two procedures

(15) $$\begin{aligned} &\underline{\text{procedure}}\ P_1;T_1(P_1,P_2) \\ &\underline{\text{procedure}}\ P_2;T_2(P_1,P_2) \end{aligned}$$

The minimal fixed point approach in this case means that for P_1 and P_2 we have

(16) $$(P_1,P_2) = \bigcap\{(X,Y):\ X = T_1(X,Y),\ Y = T_2(X,Y)\}$$

and the analogon of Knaster's result (see (9)) is that

(17) $$(P_1,P_2) = \bigcap\{(X,Y):\ T_1(X,Y) \subseteq X,\ T_2(X,Y) \subseteq Y\}\ .$$

This way of dealing with systems of procedures may be described as the *simultaneous* minimal fixed point technique. Now it turns out that this technique can be replaced by an *iterated* taking of fixed points. This is made precise in the following assertion: Let

(18) $$P_1' = \bigcap\{X':X' = T_1(X',\bigcap\{Y':Y' = T_2(X',Y')\})\}$$

(19) $$P_2' = \bigcap\{Y':Y' = T_2(P_1',Y')\}\ .$$

Then,

(20) $$P_1 = P'_1$$

(21) $$P_2 = P'_2 \ .$$

This is shown as follows: By the definitions of P'_1 and P'_2 we have

$$P'_1 = T_1(P'_1, \cap\{Y' : Y' = T_2(P'_1,Y')\}) = T_1(P'_1,P'_2)$$

$$P'_2 = T_2(P'_1,P'_2) \ .$$

By the minimality of P_1 and P_2, we infer that $P_1 \subseteq P'_1$, and $P_2 \subseteq P'_2$. Let $P''_2 = \cap\{Y'' : Y'' = T_2(P_1,Y'')\}$. Since $T_2(P_1,P_2) = P_2$, we have $P''_2 \subseteq P_2$; hence, $T_1(P_1,P''_2) \subseteq T_1(P_1,P_2) = P_1$, and, from this, $T_1(P_1, \cap\{Y'' : Y'' = T_2(P_1,Y'')\}) \subseteq P_1$.

By (18) and (9) (Knaster's result), we see that $P'_1 \subseteq P_1$. From this, $T_2(P'_1,P_2) \subseteq T_2(P_1,P_2) = P_2$.

By (19) and (9), we finally infer that $P'_2 \subseteq P_2$.

Thus, (20) and (21) are proved.

The fact that we may replace simultaneous fixed points by iterated fixed points provides a simplification of the formal system to be described in the next section. Before we preceed with this section, we summarize the contents of section 2:

Procedures are minimal fixed points of monotonic and continuous transformations.

3. THE μ-CALCULUS

The μ-calculus is a formal system developed for the investigation of properties of recursive procedures. We shall first describe the well formed formulae of the system (section 3.1), then we explain how to interpret these formulae (section 3.2), next we give the axioms and rules of inference of the system (section 3.3), we justify the validity of these (section 3.4), and finally we give some first applications of the μ-calculus (section 3.5).

3.1. LANGUAGE

In the formal language, we have

a. Two function constants, denoted by Ω and E.

b. Function variables, denoted by any, possibly indexed, upper case letter (different from Ω and E), e.g. $A, A_1, \ldots, F, \ldots, X, Y, \ldots$.

c. Predicate variables, denoted by, possibly indexed, lower case letters, e.g. $p, p_1, q, r, \ldots$.

We shall in the sequel shorten the words "function constant" and "function variable" to "constant" and "variable".

(In programming terminology:

a. Ω corresponds to the undefined statement, E to the dummy statement.

b. Each variable corresponds to some elementary statement.

c. Each predicate variable corresponds to some boolean expression.)

We now give the definition of a *term* in the formal language, which is the counterpart of the notion of statement in the programming language.

Definition 3.1.

a. Each constant or variable is a term.

b. If τ_1 and τ_2 are terms, then $\tau_1;\tau_2$ is a term.

c. If τ_1 and τ_2 are terms, and p is a predicate variable, then $(p\to\tau_1,\tau_2)$ is a term.

d. If τ is a term and X is a variable, then $\mu X[\tau]$ is a term.

Examples:

Ω, E, $A_1;A_2$,

$A;(p\to(q\to X,Y),Z)$

$\mu X[X]$

$\mu X[(p_1\to A_1;X,(p_2\to A_2;\mu Y[(p_1\to A_1;X,(p_2\to A_2;Y,E))],E))]$

There are three ways of constructing terms, starting from the basic ones given in clause a. Clause b introduces *composition*, clause c gives the *conditional* terms, and clause d introduces, by means of the *variable binding operator* μ, terms we shall see presently correspond to *procedures*. Specifically, $\mu X[\tau]$ denotes the minimal fixed point of the transformation corresponding to the term τ, i.e., $\mu X[\tau]$ corresponds to the procedure declared by <u>procedure</u> $P;\tau(P)$, where $\tau(P)$ is a provisional notation for the result of substituting P for all occurrences of X in τ.

Examples:

1. $\mu X[X]$ corresponds to the procedure declared by <u>procedure</u> P;P.
2. $\mu X[\ (p\to A_1;X,A_2)$ corresponds to the procedure declared by <u>procedure</u> $P;(p\to A_1;P,A_2)$.
3. Consider the system of declarations
   <u>procedure</u> $P_1;(p_1\to A_1;P_1,(p_2\to A_2;P_2,E))$;
   <u>procedure</u> $P_2;(p_1\to A_1;P_2,(p_2\to A_2;P_1,E))$.
   By the results of section 2, in particular (20) and (21), we have:

$$(P_1,P_2) = \bigcap\{(X,Y):\ X = (p_1\to A_1;X,(p_2\to A_2;Y,E)),\ Y = (p_1\to A_1;Y,(p_2\to A_2;X,E))\}$$

and, moreover,

$$P_1 = \bigcap\{X:X = (p_1\to A_1;X,(p_2\to A_2;\bigcap\{Y:Y=(p_1\to A_1;Y,(p_2\to A_2;X,E))\},E))\}$$
$$P_2 = \bigcap\{Y:Y = (p_1\to A_1;Y,(p_2\to A_2;P_1,E))\}\quad .$$

Hence, in the formal language, P_1 corresponds to the term

$$\mu X[(p_1\to A_1;X,(p_2\to A_2;\mu Y[(p_1\to A_1;Y,(p_2\to A_2;X,E))],E))],$$

and P_2 to the term

$$\begin{aligned}&\mu Y[(p_1 \rightarrow A_1;Y,\\ &\qquad (p_2 \rightarrow A_2;\mu X[(p_1 \rightarrow A_1;X,(p_2 \rightarrow A_2;\mu Y[(p_1 \rightarrow A_1;Y,(p_2 \rightarrow A_2;X,E))],\\ &\qquad\quad E))];\end{aligned}$$

The use of the μ-operator has the usual consequences for the distinction between free and bound variables. In particular, all occurrences of X in $\mu X[\tau]$ are bound, and an occurrence of Y in τ is free iff it is not a bound occurrence.

We shall use the notation $\tau_1[\tau_2/X]$ for the result of substituting τ_2 for all free occurrences of X in τ_1. Again, we have to take the usual precautions in a case like $\mu X[\tau_1][\tau_2/Y]$; this is defined as $\mu X'[\tau_1[X'/X][\tau_2/Y]]$, where X' is some variable which does not occur free in τ_2.

Examples:

$$X[\tau/X] = \tau$$

$$(p_1 \rightarrow A_1;X,(p_2 \rightarrow A_2;X,E))[A/X] = (p_1 \rightarrow A_1;A,(p_2 \rightarrow A_2;A,E))$$

$$\mu X[(p \rightarrow A_1;X,Y)][A_2;X/Y] = \mu X'[(p \rightarrow A_1;X',A_2;X)]$$

Using the notation for substitution, we can now describe in general how systems of procedures fit into the language. Consider the procedure declarations

$$\begin{aligned}&\underline{\text{procedure}}\ P_1;\tau_1[P_1/X][P_2/Y];\\ &\underline{\text{procedure}}\ P_2;\tau_2[P_1/X][P_2/Y].\end{aligned}$$

Then, in the formal language we have for the terms corresponding to P_1 and P_2 respectively:

$$\begin{aligned}&\mu X[\tau_1[\mu Y[\tau_2]/Y]]\\ &\mu Y[\tau_2[\mu X[\tau_1[\mu Y[\tau_2]/Y]]/X]].\end{aligned}$$

The generalization to a system of more than two procedures is straightforward.

Terms are used to construct *atomic formulae*, *formulae* and *assertions* as follows:

Definition 3.2. Let τ_1 and τ_2 be terms.

a. An *atomic formula* is either an *equivalence* $\tau_1 = \tau_2$, or an *inclusion* $\tau_1 \subseteq \tau_2$.

b. A *formula* is a list of zero or more atomic formulae, written as $\Phi_1, \Phi_2, \ldots, \Phi_n$, each Φ_i, $1 \leq i \leq n$, an atomic formula.

c. An *assertion* has the form $\Phi \vdash \psi$, where Φ, ψ are formulae.

Anticipating the formal rules for interpreting assertions, to be given in the next subsection, we can already indicate their intended meaning:

a. The atomic formulae are the counterparts of the relationships $S_1 = S_2$ and $S_1 \subseteq S_2$ of section 2.

b. A formula $\Phi = \Phi_1, \Phi_2, \ldots, \Phi_n$ holds iff the conjunction of its elements, i.e. $\Phi_1 \wedge \Phi_2 \wedge \ldots \wedge \Phi_n$, holds.

c. An assertion $\Phi \vdash \psi$ holds iff "Φ holds implies that ψ holds" holds.

Examples of assertions:

1. $X \subseteq Y,\ Y \subseteq X \vdash X = Y$

2. $\vdash \mu X[(p \rightarrow X, A)] = (p \rightarrow \Omega, A)$

   (cf. the comments on the procedure declared by equation (7) of section 2).

3. $$\left.\begin{array}{l} A;(p \rightarrow E, E) = (p \rightarrow A, A), \\ A;(p \rightarrow E, \Omega) = (p \rightarrow A, \Omega), \\ A;(p \rightarrow \Omega, E) = (p \rightarrow \Omega, A), \\ A;(p \rightarrow \Omega, \Omega) = (p \rightarrow \Omega, \Omega). \end{array}\right\} \quad A;(p \rightarrow X, Y) = (p \rightarrow A;X, A;X)$$

In the sequel, we shall occasionally omit the ";" symbol between terms. The notation for substitution is extended in an obvious way from terms to assertions:

$$\begin{aligned}(\Phi \vdash \psi)[\tau/X] &= \Phi[\tau/X] \vdash \psi[\tau/X]\\ (\Phi_1,\dots,\Phi_n)[\tau/X] &= (\Phi_1[\tau/X],\dots,\Phi_n[\tau/X])\\ (\tau_1 = \tau_2)[\tau/X] &= (\tau_1[\tau/X] = \tau_2[\tau/X])\\ (\tau_1 \subseteq \tau_2)[\tau/X] &= (\tau_1[\tau/X] \subseteq \tau_2[\tau/X]) .\end{aligned}$$

3.2. INTERPRETATION

We give a set of rules to give an *interpretation* I to an assertion α: $\Phi \vdash \psi$.

Let $p_1,p_2,\dots,p_i,\dots$ be the predicate variables occurring in α, and let $A_1,A_2,\dots,A_i,\dots$ be the <u>free</u> variables of α.

1. Choose some domain $\mathcal{D}$.

2a. With Ω, associate the nowhere defined function Ω^I: $\mathcal{D} \to \mathcal{D}$. With E, associate the identity function E^I: $\mathcal{D} \to \mathcal{D}$. (For all $x \in \mathcal{D}$, $E^I(x) = x$).

2b. Associate with each variable A_i a partial function A_i^I: $\mathcal{D} \to \mathcal{D}$.

2c. Associate with each predicate variable p_i a partial function p_i^I: $\mathcal{D} \to \{0,1\}$.

3. Given the interpretation I of the constants, the A_i and the p_i, we now define how to extend I to terms, formulae and assertions.

4. Interpretation of terms.

4a. Let τ_1^I, τ_2^I and p^I be determined already. Then $(\tau_1;\tau_2)^I$ and $(p\to\tau_1,\tau_2)^I$ are defined by: Let $x \in \mathcal{D}$.

If $\tau_1^I(x)$ is undefined, then $(\tau_1;\tau_2)^I(x)$ is undefined.

If $\tau_1^I(x) = y$, and $\tau_2^I(y)$ is undefined, then $(\tau_1;\tau_2)^I(x)$ is undefined.

If $\tau_1^I(x) = y$, and $\tau_2^I(y) = z$, then $(\tau_1;\tau_2)^I(x) = z$.

If $p^I(x)$ is undefined, then $(p\to\tau_1,\tau_2)^I(x)$ is undefined.

If $p^I(x) = 1$, and $\tau_1^I(x)$ is undefined, then $(p\to\tau_1,\tau_2)^I(x)$ is undefined.

If $p^I(x) = 1$, and $\tau_1^I(x) = y$, then $(p\to\tau_1,\tau_2)^I(x) = y$.

If $p^I(x) = 0$, and $\tau_2^I(x)$ is undefined, then $(p\to\tau_1,\tau_2)^I(x)$ is undefined.

If $p^I(x) = 0$, and $\tau_2^I(x) = z$, then $(p\to\tau_1,\tau_2)^I(x) = z$.

4b. Let τ^I be determined already, except for the interpretation of the variable X. Then

$$\mu X[\tau]^I = \bigcap_{X^I:\ \mathcal{D}\to\mathcal{D}} \{X^I : X^I = \tau^I\}$$

i.e., $\mu X[\tau]^I$ is the g.l.b. of the family of all partial functions X^I, which satisfy $X^I = \tau^I$.

5. Interpretation of atomic formulae.

5a. $(\tau_1 \subseteq \tau_2)^I$ is true iff $\tau_1^I \subseteq \tau_2^I$ is true, i.e., iff for all $x \in \mathcal{D}$, if $\tau_1^I(x) = y$, then $\tau_2^I(x) = y$.

5b. $(\tau_1 = \tau_2)^I$ is true iff $\tau_1^I = \tau_2^I$ is true, i.e., iff for all $x \in \mathcal{D}$,
if $\tau_1^I(x) = y$, then $\tau_2^I(x) = y$, and
if $\tau_2^I(x) = y$, then $\tau_1^I(x) = y$.

6. Interpretation of formulae.
A list $\Phi^I = (\Phi_1,\Phi_2,\ldots,\Phi_n)^I$ is true iff each Φ_i^I is true, $1 \leq i \leq n$.

7. Interpretation of assertions.
An assertion $(\Phi \vdash \psi)^I$ is true iff the implication $\Phi^I \supset \psi^I$ is true.

An assertion α: $\Phi \vdash \psi$ is called *valid* if $(\Phi \vdash \psi)^I$ is true for all interpretations. It should be emphasized that this means that for all domains $\mathcal{D}$, and for all choices of partial functions for the free variables occurring in α (which are then extended in the given way to interpretations I of α), we have that $(\Phi \vdash \psi)^I$ is true.
The statement "$\Phi \vdash \psi$ is valid" is abbreviated to "$\Phi \models \psi$".

For examples of valid assertions, see the examples following definition 3.2. in subsection 3.1. Consider the second example. In programming terminology, its validity asserts that, whatever choice we make for the boolean expression p and the elementary statement A (i.e. whatever functions p^I and A^I on $\mathcal{D} \to \{0,1\}$ and $\mathcal{D} \to \mathcal{D}$ we choose), it is always true that, if P is the procedure declared by procedure $P;(p\to P,A)$, then P is equivalent to $(p\to\Omega,A)$.

3.3. AXIOMS AND RULES

We shall not stretch the formalism to its limits: The usual formal rules for dealing with $\vdash$, $=$, and substitution will not be listed here, let alone systematically used. This would detract from the understanding of the central part of our formal system, consisting of the following axioms and rules:

1. Axioms for composition

   a. $\vdash E;X = X$ $\qquad \vdash X;E = X$

   b. $\vdash X;(Y;Z) = (X;Y);Z$

   c. $\vdash \Omega;X = \Omega$

   $\vdash X;\Omega = \Omega$

(We have anticipated the associativity of ";", by omitting parentheses around $\tau_1;\tau_2$ in definition 3.1.)

2. Axioms for $\subseteq$

   a. $\vdash X \subseteq X$

   b. $X \subseteq Y,\ Y \subseteq X \vdash X = Y$

   c. $X \subseteq Y,\ Y \subseteq Z \vdash X \subseteq Z$

   d. $\vdash \Omega \subseteq X$

   e. $X \subseteq Y \vdash \tau \subseteq \tau[Y/X]$

3. Axioms for conditionals

   a. $\vdash (p\to E,E) \subseteq E$

   b. $\vdash (p\to(p\to X,Y),Z) = (p\to X,Z)$

   $\vdash (p\to X,(p\to Y,Z)) = (p\to X,Z)$

   c. $\vdash (p\to(q\to X,Y),\ (q\to U,V)) =$

   $(q\to(p\to X,U),\ (p\to Y,V))$

   d. $\vdash (p\to X,Y);Z = (p\to X;Z,Y;Z)$

4. Axiom for the μ-operator

   $\vdash \tau[\mu X[\tau]/X] \subseteq \mu X[\tau]$

5. Rule of inference for the μ-operator

$$\frac{\psi \vdash \Phi[\Omega/X] \qquad \psi,\Phi \vdash \Phi[\tau/X]}{\psi \vdash \Phi[\mu X[\tau]/X]}$$

   provided that X does not occur free in ψ.

This rule of inference, which will be called the μ-induction rule, is the foundation of the μ-calculus.

3.4. JUSTIFICATION

1. The three axioms for composition assert that
   a. E is its identity element
   b. It is associative
   c. Ω is its zero element

These three assertions are clearly valid.

2. The first three axioms for "$\subseteq$" assert that it is a reflexive, antisymmetric and transitive relation. Moreover, it is clear that the undefined function is included in every function. The fifth axiom will be discussed below.

3. Consider the first axiom for conditionals. Observe that for each domain $\mathcal{D}$ and each $x \in \mathcal{D}$, either $(p \to E,E)^I(x)$ is undefined (in case $p^I(x)$ is undefined) or $(p \to E,E)^I(x) = x$. This proves the validity of $\vdash (p \to E,E) \subseteq E$. Note that replacement of this axiom by $\vdash (p \to E,E) = E$ would be valid only if the predicates were interpreted as <u>total</u> functions on $\mathcal{D}$. The validity of the other three axioms for conditionals, which are taken from McCarthy [23] is also easily verified.

4. In the language of the μ-calculus, the fixed point property of procedures (if the procedure P is declared by <u>procedure</u> P; $\tau(P)$, then $P = \tau(P)$) is expressed by $\vdash \mu X[\tau] = \tau[\mu X[\tau]/X]$. The axiom for the μ-operator gives one half of this equivalence. The other half and the minimality of $\mu X[\tau]$ will be proved in subsection 3.5.

5. Before we discuss the μ-induction rule, we first prove the validity of the main axiom for inclusion

$$X \subseteq Y \vdash \tau \subseteq \tau[Y/X]$$

The proof of its validity

$$X \subseteq Y \models \tau \subseteq \tau[Y/X]$$

amounts to showing that each term is monotonic in all of its free variables. It is not difficult to verify this for those terms which involve only composition and conditionals. Also, it can be proved that such simple terms (without μ's) are continuous in all their free variables. These two facts provide the basis for an inductive argument -on the complexity of τ- that monotonicity and continuity hold for arbitrary τ. Now assume that τ is monotonic and continuous in two of its free variables, X and Y, say. We indicate our special interest in these by writing $\tau = \tau(X,Y)$. We show that then $\mu Y[\tau(X,Y)]$ is monotonic and continuous in X.

1. Monotonicity

Since $\tau(X,Y)$ is continuous in Y, we have, using the notation $\tau(X)(Y)$ for $\tau(X,Y)$:

$$\mu Y[\tau(X,Y)] = \bigcup_{i=0}^{\infty} \tau(X)^i(\Omega)$$

where $\tau(X)^0(\Omega) = \Omega$

$$\tau(X)^{i+1}(\Omega) = \tau(X,\tau(X)^i(\Omega)) \ .$$

Thus, in order to prove

$$X \subseteq X' \models \mu Y[\tau(X,Y)] \subseteq \mu Y[\tau(X',Y)]$$

it is sufficient to show, for each i,

$$X \subseteq X' \models \tau(X)^i(\Omega) \subseteq \tau(X')^i(\Omega) \ .$$

We use induction on i.

a. $i = 0$: $X \subseteq X' \models \Omega \subseteq \Omega$ is clear.

b. Assume the assertion for i. Then

$$\begin{aligned}\tau(X)^{i+1}(\Omega) &= \tau(X,\tau(X)^i(\Omega))\\ &\subseteq \tau(X',\tau(X)^i(\Omega))\\ &\subseteq \tau(X',\tau(X')^i(\Omega))\\ &= \tau(X')^{i+1}(\Omega)\end{aligned}$$

where we have used the monotonicity of τ in X, the induction assumption, and the monotonicity of τ in Y.

An alternative proof, which does not use the continuity of τ can be based on Knaster's result. Let μ and μ' be short for $\mu Y[\tau(X,Y)]$ and $\mu Y[\tau(X',Y)]$ respectively. We have to show: If $X \subseteq X'$, then $\mu \subseteq \mu'$. By the monotonicity of τ in Y we have

$$\mu = \bigcap\{Y\colon \tau(X,Y) \subseteq Y\}$$

$$\mu' = \bigcap\{Y\colon \tau(X',Y) \subseteq Y\}$$

In order to prove $\mu \subseteq \mu'$, it is sufficient to show that, for all Y such that $\tau(X',Y) \subseteq Y$, we have $\mu \subseteq Y$. But, if $\tau(X',Y) \subseteq Y$, then $\tau(X,Y) \subseteq \tau(X',Y) \subseteq Y$, by the monotonicity of τ in X. Since μ is the l.u.b. of all Y such that $\tau(X,Y) \subseteq Y$, we see that $\mu \subseteq Y$, and the result follows

2. Continuity

Let $X_0 \subseteq X_1 \subseteq \dots \subseteq X_i \subseteq \dots$.

We have to show that

$$\mu Y[\tau(\bigcup_{i=0}^{\infty} X_i, Y)] = \bigcup_{i=0}^{\infty} \mu Y[\tau(X_i,Y)]$$

(Cf. definition (14) of section 2) or, using the continuity of τ in Y, that

$$\bigcup_{j=0}^{\infty} \tau(\bigcup_{i=0}^{\infty} X_i)^j(\Omega) = \bigcup_{i=0}^{\infty}\bigcup_{j=0}^{\infty} \tau(X_i)^j(\Omega) \ .$$

Since

$$\bigcup_{i=0}^{\infty}\bigcup_{j=0}^{\infty} \tau(X_i)^j(\Omega) = \bigcup_{j=0}^{\infty}\bigcup_{i=0}^{\infty} \tau(X_i)^j(\Omega)$$

it is sufficient to prove

$$\tau(\bigcup_{i=0}^{\infty} X_i)^j(\Omega) = \bigcup_{i=0}^{\infty} \tau(X_i)^j(\Omega) \ .$$

We use induction on j.

a. If $j = 0$, the assertion is clear.

b. Assume the assertion for j. Then

$$\begin{aligned}
\tau(\bigcup_{i=0}^{\infty} X_i)^{j+1}(\Omega) &= \tau(\bigcup_{i=0}^{\infty} X_i, \tau(\bigcup_{i=0}^{\infty} X_i)^{j}(\Omega)) \\
\text{(ind. assumption)} &= \tau(\bigcup_{i=0}^{\infty} X_i, \bigcup_{i=0}^{\infty} \tau(X_i)^{j}(\Omega)) \\
\text{(cont.in X and Y)} &= \bigcup_{i=0}^{\infty} \bigcup_{k=0}^{\infty} \tau(X_i, \tau(X_k)^{j}(\Omega)) \\
\text{(mon.in X and Y)} &= \bigcup_{i=0}^{\infty} \bigcup_{k=0}^{\infty} \tau(X_{max(i,k)}, \tau(X_{max(i,k)})^{j}(\Omega)) \\
&= \bigcup_{n=0}^{\infty} \tau(X_n, \tau(X_n)^{j}(\Omega)) \\
&= \bigcup_{n=0}^{\infty} \tau(X_n)^{j+1}(\Omega) \quad .
\end{aligned}$$

This completes the core of the proof that each term is monotonic and continuous in each of its free variables. Extension to the full proof is straightforward.

Now that the continuity of each τ has been established, we first of all conclude that each τ does have a minimal fixed point, viz. $\bigcup_{i=0}^{\infty} \tau^{i}(\Omega)$; from this it follows that, for each interpretation I, the definition of $\mu X[\tau]^{I}$ as $\bigcap_{X^I:\mathcal{D}\to\mathcal{D}} \{X^I : X^I = \tau^I\}$, as given in section 3.2, makes sense, since we have now proved that, for each τ, the set $\{X^I : X^I = \tau^I\}$ is non-empty.

Secondly, we can now justify the μ-induction rule, which we repeat here for convenience:

$$\begin{array}{l}
\psi \vdash \Phi[\Omega/X] \\
\underline{\psi, \Phi \vdash \Phi[\tau/X]} \\
\psi \vdash \Phi[\mu X[\tau]/X]
\end{array}$$

provided that X does not occur free in ψ.

The inductive pattern in this rule is clarified by phrasing it informally as follows: If one wishes to prove an assertion α about a procedure $\mu X[\tau]$, one shows that

a. The basis step $\alpha(\Omega)$ holds,

b. If $\alpha(X)$ holds, then $\alpha(\tau(X))$ holds,

and from these two results one then infers that $\alpha(\mu X[\tau])$ holds.

For the formal justification of the μ-induction rule, we introduce the following notation:

a. $\bigcup_{i=0}^{\infty} \Phi = (\bigcup_{i=0}^{\infty} \Phi_1, \bigcup_{i=0}^{\infty} \Phi_2, \ldots, \bigcup_{i=0}^{\infty} \Phi_n)$,

where Φ is the list of atomic formulae $\Phi_1, \Phi_2, \ldots, \Phi_n$.

b. $\bigcup_{i=0}^{\infty} (\tau_1 \subseteq \tau_2) = (\bigcup_{i=0}^{\infty} \tau_1 \subseteq \bigcup_{i=0}^{\infty} \tau_2)$

$\bigcup_{i=0}^{\infty} (\tau_1 = \tau_2) = (\bigcup_{i=0}^{\infty} \tau_1 = \bigcup_{i=0}^{\infty} \tau_2)$

Using this notation, our continuity result reads: If $X_0 \subseteq X_1 \subseteq \ldots \subseteq X_i \subseteq \ldots$ then

$$\bigcup_{i=0}^{\infty} \Phi(X_i) = \Phi(\bigcup_{i=0}^{\infty} X_i)$$

The validity of the μ-induction rule then follows easily:

Suppose

$$\psi \models \Phi[\Omega/X]$$
$$\psi, \Phi \models \Phi[\tau/X]$$

have been proved, or, is a more suggestive notation, that

$$\psi \models \Phi(\Omega)$$
$$\psi, \Phi(X) \models \Phi(\tau(X))$$

hold. Starting from the first result, and repeatedly applying the second, we then have

$$\psi \models \Phi(\Omega)$$
$$\psi \models \Phi(\tau(\Omega))$$
$$\psi \models \Phi(\tau^2(\Omega))$$
$$\vdots$$
$$\psi \models \Phi(\tau^i(\Omega))$$
$$\vdots$$

Thus, we conclude that

$$\psi \models \bigcup_{i=0}^{\infty} \Phi(\tau^i(\Omega))$$

holds. By continuity, this means that

$$\psi \models \Phi(\bigcup_{i=0}^{\infty} \tau^i(\Omega))$$

also holds, i.e., we have

$$\psi \models \Phi(\mu X[\tau(X)])$$

which is the desired result.

3.5. FIRST APPLICATIONS

We give some first applications of the μ-calculus; many other examples will be given in section 4.

1. Proof that $\mu X[\tau]$ is the minimal fixed point of τ. First we show that $\mu X[\tau]$ is a fixed point, i.e., that

$$\vdash \mu X[\tau] = \tau[\mu X[\tau]/X] \ .$$

One half of this is given by the μ-axiom

$$\vdash \tau[\mu X[\tau]/X] \subseteq \mu X[\tau] \ .$$

In order to prove

$$\vdash \mu X[\tau] \subseteq \tau[\mu X[\tau]/X]$$

we use the μ-induction rule. We have to establish

$$\vdash \Omega \subseteq \tau[\Omega/X]$$

which is clear, and

$$X \subseteq \tau \vdash \tau \subseteq \tau[\tau/X]$$

which holds by monotonicity. (When we say that an assertion follows by monotonicity, we mean that it can be derived from the axiom $X \subseteq Y \vdash \tau \subseteq \tau[Y/X]$, by suitable application of the other axioms and rules. Usually, this will need some use of the rules for substitution, deduction and equivalence, which we have not presented here formally.) By the μ-induction rule, from these two assertions we may infer that

$$\vdash \mu X[\tau] \subseteq \tau[\mu X[\tau]/X]$$

holds. Next we show that $\mu X[\tau]$ is the minimal fixed point, i.e., that

$$Y = \tau[Y/X] \vdash \mu X[\tau] \subseteq Y \ .$$

Again, we use the μ-induction rule. We have

$$Y = \tau[Y/X] \vdash \Omega \subseteq Y$$

and, by monotonicity,

$$Y = \tau[Y/X],\ X \subseteq Y \vdash \tau \subseteq Y \ .$$

From these two assertions, the desired result follows.

2. Proof of the monotonicity of $\mu X[\tau]$ from the monotonicity of τ.
We have to show

$$Y \subseteq Y' \vdash \mu X[\tau] \subseteq \mu X[\tau][Y'/Y] \ .$$

In order to apply the μ-induction rule, we have to prove the two assertions

$$Y \subseteq Y' \vdash \Omega \subseteq \mu X[\tau[Y'/Y]]$$

and

$$Y \subseteq Y',\ X \subseteq \mu X[\tau[Y'/Y]] \vdash \tau \subseteq \mu X[\tau[Y'/Y]]\ .$$

The first is clear. By the fixed point property, for the second we can write

$$Y \subseteq Y',\ X \subseteq \mu X[\tau[Y'/Y]] \vdash \tau \subseteq \tau[Y'/Y][\mu X[\tau[Y'/Y]]/X]$$

and this follows by the monotonicity of τ in X and Y.

3. An elementary property of while statements. Consider the procedure declared by

$$\underline{\text{procedure}}\ P;\ (p\rightarrow A;P,E)$$

(remember that E is the identity function, or, in programming terminology, the dummy statement.) The action of this P can be described by: Perform A as long as p is true, or, equivalently, as that of the while statement <u>while</u> p <u>do</u> A, for which we shall use the shorter notation $p * A$. Thus, in our formalism, $p * A = \mu X[p\rightarrow A;X,E)]$. We apply the μ-calculus to prove the following simple property of while statements:

$$\vdash p * A_1;A_2 = \mu X[(p\rightarrow A_1;X,A_2)]$$

a. Proof of $\supseteq$. By the fixed point property of $p * A_1$, we have
$p * A_1 = (p\rightarrow A;p*A_1,E)$; hence,
$p * A_1;A_2 = (p\rightarrow A_1;p*A_1,E);\ A_2 = (p\rightarrow A_1;p*A_1;A_2,A_2)$.
Thus, $p * A_1;A_2$ is a fixed point of $\tau(X) = (p\rightarrow A_1;X,A_2)$.
Since $\mu X[(p\rightarrow A_1;X,A_2)]$ is its minimal fixed point, the result follows.

b. Proof of $\subseteq$. We apply the μ-induction rule to show

$$\vdash \mu X[(p\rightarrow A_1;X,E)];\ A_2 \subseteq \mu X[(p\rightarrow A_1;X,A_2)]$$

That $\vdash \Omega;A_2 \subseteq \mu X[(p\to A_1;X,A_2)]$ is clear. Let $R = \mu X[(p\to A_1;X,A_2)]$. For the second step of the μ-induction rule, we have to verify

$$X;A_2 \subseteq R \vdash (p\to A_1;X,E);A_2 \subseteq R \ .$$

Using the fixed point property of R, the last assertion follows from

$$X;A_2 \subseteq R \vdash (p\to A_1;X;A_2,A_2) \subseteq (p\to A_1;R,A_2) \ .$$

(This last example, however simple its result, provides one of the basic steps for the proof of the completeness theorem in sections 5 and 6. The proper generalization of it can be found in lemma's 5.10, 5.11 and 5.12).

4. EXAMPLES

4.1. WHILE STATEMENTS

A rich source of, mostly simple, examples on program equivalence is provided by the while statement. First we introduce some auxiliary notation and corresponding axioms. We shall allow conditionals of the form

$$(p_1 \wedge p_2 \rightarrow X, Y)$$
$$(p_1 \vee p_2 \rightarrow X, Y)$$
$$(\neg p \rightarrow X, Y)$$

which are characterized by the axioms

$$\vdash (p_1 \wedge p_2 \rightarrow X, Y) = (p_1 \rightarrow (p_2 \rightarrow X, Y), Y)$$
$$\vdash (p_1 \vee p_2 \rightarrow X, Y) = (p_1 \rightarrow X, (p_2 \rightarrow X, Y))$$
$$\vdash (\neg p \rightarrow X, Y) = (p \rightarrow Y, X) \ .$$

Note that "$\wedge$" and "$\vee$" are not commutative: E.g., $(p_1 \rightarrow (p_2 \rightarrow X, Y), Y)$ and $(p_2 \rightarrow (p_1 \rightarrow X, Y), Y)$ are not necessarily equivalent, since p_1 may be undefined by some argument for which p_2 is false.

Using this notation and the one for while statements introduced in section 3.5, we have

$$p * A = \mu X[(p \rightarrow AX, E)]$$
$$p_1 \wedge p_2 * A = \mu X[(p_1 \rightarrow (p_2 \rightarrow AX, E), E)]$$
$$p_1 \vee p_2 * A = \mu X[(p_1 \rightarrow AX, (p_2 \rightarrow AX, E))]$$
$$\neg p * A = \mu X[(p \rightarrow E, AX)]$$

The following equivalences all hold for the predicates concerned total ($\vdash (p \rightarrow E, E) = E$) and some of them also hold for partial predicates ($\vdash (p \rightarrow E, E) \subseteq E$):

1. $\vdash p * A = (p\to A;p*A,E)$
2. $\vdash p * A = p * p * A$
3. $\vdash p * A = p * (A;p*A)$
4. $\vdash p * A_1;(p\to A_2,A_3) = p * A_1;A_3$
5. $\vdash p * E = (p\to\Omega,E)$
6. $\vdash p_1 * p_2 * E = p_1 * E$
7. $\vdash p_1 * (A;p_2*A) = p_1 * p_1\vee p_2 * A$
8. $\vdash p_1\wedge p_2 * A;p_1 * A = p_1 * A$
9. $\vdash p_1 * A;p_1\vee p_2 * A = p_1\vee p_2 * A$
10. $\vdash p_1\vee p_2 * A = p_2 * A;p_1 * (A;p_2 * A)$
11. $\vdash p_1 * A;p_2 * A = p_1 * A;\neg p_1\wedge p_2 * p_2 * A$
12. $\vdash p_1 * p_2 * A = p_1 * (p_1\wedge p_2*A;\neg p_1\wedge p_2*A)$
13. $\vdash p_1 * A = p_1 * (p_1\wedge p_2*A;p_1\wedge\neg p_2*A)$
14. $\vdash p_1 * p_2 * A = p_1 * p_1\wedge p_2 * p_2 * A$
15. $\vdash p_1 * p_2 * p_1 * p_2 * A = p_1 * p_2 * A$

etc.

We shall prove here only examples 1, 2, 5, 10 and 15; the remaining ones are left as exercises to the reader. A mechanical way of proving these and other equivalences for while statements will follow from the completeness proof of sections 5 and 6.

In our comments, we shall not indicate explicitly use of the axioms on composition and conditionals.

1. $\vdash p * A = (p\to A;p*A,E)$.

   This is a direct consequence of the fixed point property (f.p.p.):

   $\vdash p * A = \mu X[(p\to AX,E)] = (p\to AX,E)[\mu X[(p\to AX,E)]/X] =$

   $(p\to A;\mu X[(p\to AX,E)],E) = (p\to A;p*A,E)$

2. $\vdash p * A = p * p * A$.

   By the f.p.p.

   $\vdash p * p * A = (p\to p*A;p*p*A,E)$.

   Also, using the last result of section 3.5., and the f.p.p.,

$\vdash p * A;p * p * A =$
$\mu X[(p\to AX,E)];p * p * A =$
$\mu X[(p\to AX,p*p*A)] =$
$\mu X[(p\to AX,(p\to p*A;p*p*A,E))] =$
$\mu X[(p\to AX,E)] =$
$p * A.$

Hence,

$\vdash p * A;p * p * A = p * A.$

Thus,

$\vdash p * p * A = (p\to p*A,E) = (p\to (p\to A;p*A,E),E) =$
$(p\to A;p*A,E) = p * A$

which proves the desired result.

5. $\vdash p * E = (p\to\Omega,E).$

We have

$\vdash p * E = \mu X[(p\to EX,E)] = \mu X[(p\to X,E)]$

a. Proof of $\supseteq$. By monotonicity and the f.p.p.,

$\vdash (p\to\Omega,E) \subseteq (p\to\mu X[(p\to X,E)],E) = \mu X[(p\to X,E)]$

b. Proof of $\subseteq$. We apply the μ-induction rule. Thus, we must show

$\vdash \Omega \subseteq (p\to\Omega,E)$, which is clear, and

$X \subseteq (p\to\Omega,E) \vdash (p\to X,E) \subseteq (p\to\Omega,E)$.

Assume $X \subseteq (p\to\Omega,E)$. Then

$\vdash (p\to X,E) \subseteq (p\to (p\to\Omega,E),E) = (p\to\Omega,E)$,

and the result follows.

A generalization of example 5 will be given in lemma 5.2.

Before we give the proofs of 10 and 15, we first list some auxiliary results:

R_1: $\vdash \mu X[\mu Y[\tau]] = \mu Y[\tau[Y/X]]$

R_2: $\vdash \mu X[A;\tau] = A;\mu X[\tau[AX/X]]$

R_3: Let Y not occur free in τ. Then
$\vdash \mu X[(p\to\tau,E)] = (p\to\mu X[\tau[(p\to Y,E)/X]],E)$

R_4: Let τ' be the result of replacing, in τ, one or more free occurrences of X by $\mu Y[\tau[Y/X]]$, where Y is a variable which does not occur free in τ.

Then

$\vdash \mu X[\tau] = \mu X[\tau']$.

These results can be transliterated back into results on the equivalence of programs. We shall do this here for R_1 and R_4.

R_1:

Let $\mathcal{P}_1$ be the program

```
begin procedure $P_1$;$P_2$;
      procedure $P_2$;$\tau(P_1,P_2)$;
      $P_1$
end
```

and let $\mathcal{P}_2$ be the program

```
begin procedure P;$\tau(P,P)$;
      P
end.
```

Then $\mathcal{P}_1$ and $\mathcal{P}_2$ are equivalent.

R_4: We give only a specific instance.

Let $\mathcal{P}_3$ be the program

```
begin procedure P;
      begin ... P ... P ... end;
      P
end
```

and let $\mathcal{P}_4$ be the program

```
begin procedure P;
      begin ... begin procedure Q;
                      begin ... Q ... Q ... end;
                      Q
                end
            ...
            P
            ...
      end;
      P
end.
```

Then $\mathcal{P}_3$ and $\mathcal{P}_4$ are equivalent.

We prove only R_3. Call the left hand side P_1 and the right hand side P_2.

a. $P_1 \subseteq P_2$.

We apply the μ-induction rule. We have to show

$X \subseteq P_2 \vdash (p \to \tau, E) \subseteq P_2$.

By monotonicity, it is sufficient to show

$X \subseteq P_2 \vdash \tau \subseteq \mu Y[\tau[(p \to Y, E)/X]]$.

By the f.p.p., this is the same as

$X \subseteq P_2 \vdash \tau \subseteq \tau[(p \to Y, E)/X][\mu Y[\tau[(p \to Y, E)/X]]/Y]$.

By a property of substitution, and since Y does not occur free in τ, this reduces to

$X \subseteq P_2 \vdash \tau \subseteq \tau[(p \to \mu Y[\tau[(p \to Y, E)/X]], E)/X]$

or

$X \subseteq P_2 \vdash \tau \subseteq \tau[P_2/X]$

which holds by monotonicity.

b. $P_2 \subseteq P_1$.

By the μ-induction rule, it is sufficient to show

$(p \to Y, E) \subseteq P_1 \vdash (p \to \tau[(p \to Y, E)/X], E) \subseteq P_1$.

By the f.p.p., this is the same as

$(p \to Y, E) \subseteq P_1 \vdash (p \to \tau[(p \to Y, E)/X], E) \subseteq (p \to \tau[P_1/X], E)$

and the result follows by monotonicity.

(A somewhat more general version of R_3 is given in lemma 5.7.)

We now proceed with the proof of

10. If p_1 and p_2 are total predicates, then

$$(1) \qquad \vdash p_1 \vee p_2 * A = p_2 * A; p_1 * (A; p_2 * A)$$

First we rewrite the left hand side of (1). Since $\vdash (p_1 \to E, E) = (p_2 \to E, E) = E$, we have $\vdash (p_1 \to X, X) = (p_2 \to X, X) = X$. Using this, we obtain

$$\begin{aligned} \vdash p_1 \vee p_2 * A = {} & \mu X[(p_1 \to AX, (p_2 \to AX, E))] = \\ & \mu X[(p_1 \to (p_2 \to AX, AX), (p_2 \to AX, E))] = \\ & \mu X[(p_2 \to (p_1 \to AX, AX), (p_1 \to AX, E))] = \\ & \mu X[(p_2 \to AX, (p_1 \to AX, E))]. \end{aligned}$$

Thus,

(2) $$\vdash p_1 \vee p_2 * A = \mu X[(p_2 \to AX,(p_1 \to AX,E))] \text{ .}$$

(What we have shown here is in effect that for total predicates, "v" is a commutative operator.)

For the right hand side of (1) we write, using the last result of section 3.5. twice:

(3) $$\vdash p_2 * A;p_1 * (A;p_2 * A) = \\ \mu X[(p_2 \to AX,\mu Y[(p_1 \to A;\mu Z[(p_2 \to AZ,Y)],E)])].$$

Applying R_3 to $\mu Y[(p_1 \to A;\mu Z[(p_2 \to AZ,Y)],E)]$ we obtain

$$\vdash \mu Y[(p_1 \to A;\mu Z[(p_2 \to AZ,Y)],E)] = \\ (p_1 \to \mu U[A;\mu Z[(p_2 \to AZ,(p_1 \to U,E))]],E).$$

By R_1 and R_2 we then have

$$\vdash \mu Y[(p_1 \to A;\mu Z[(p_2 \to AZ,Y)],E)] = \\ (p_1 \to A;\mu U[(p_2 \to AU,(p_1 \to AU,E))],E).$$

Combining this with (3), we see that

(4) $$\vdash p_2 * A;p_1 * (A;p_2 * A) = \\ \mu X[(p_2 \to AX,(p_1 \to A;\mu U[(p_2 \to AU,(p_1 \to AU,E))],E))].$$

Comparing (2) and (4), we see that we can apply R_4; thus, (1) follows. Using the notation $A * p$ for $A;p * A$, result (1) can be phrased concisely as:

$$\vdash A * p_1 \vee p_2 = A * p_2 * p_1.$$

Finally, we prove

15. $$\vdash p_1 * p_2 * A = p_1 * p_2 * p_1 * p_2 * A.$$

For the left hand side we write, using R_3 and a result similar to example 5:

$$\begin{aligned}&\vdash p_1 * p_2 * A = \\ &\quad \mu X[(p_1 \to \mu Y[(p_2 \to AY, E)]; X, E)] = \\ &\quad \mu X[(p_1 \to \mu Y[(p_2 \to AY, X)], E)] = \\ &\quad (p_1 \to \mu Y[(p_2 \to AY, (p_1 \to Y, E))], E) = \\ &\quad (p_1 \to \mu Y[(p_2 \to AY, (p_1 \to \Omega, E))], E).\end{aligned}$$

Replacing in this result A by $p_1 * A$, and using an analogue of R_3, an analogue of example 5, and the conditional axioms, we derive

$$\begin{aligned}&\vdash p_1 * p_2 * p_1 * A = \\ &\quad (p_1 \to \mu Y[(p_2 \to \mu Z[(p_1 \to AZ, Y)], (p_1 \to \Omega, E))], E) = \\ &\quad (p_1 \to (p_2 \to \mu Z[(p_1 \to AZ, (p_2 \to Z, (p_1 \to \Omega, E)))], (p_1 \to \Omega, E)), E) = \\ &\quad (p_1 \to (p_2 \to \mu Z[(p_1 \to AZ, (p_2 \to \Omega, E))], \Omega), E).\end{aligned}$$

Replacing in this result A by $p_2 * A$, using similar arguments as above, and the f.p.p., it then follows that

$$\vdash p_1 * p_2 * p_1 * p_2 * A = p_1 * p_2 * A.$$

4.2. A WHILE STATEMENT EXAMPLE FROM DIJKSTRA

In [8], p.31, the following example is discussed:
If the booleans B_1 and B_2 have no side effects, and if B_2 is unaffected either by S_1 or S_2, then the following two programs are equivalent:

$$\begin{aligned}&\underline{\text{if}}\ B_2\ \underline{\text{then}}\ \underline{\text{while}}\ B_1\ \underline{\text{do}}\ S_1 \\ &\qquad\quad \underline{\text{else}}\ \underline{\text{while}}\ B_1\ \underline{\text{do}}\ S_2\end{aligned}$$

and

$$\underline{\text{while}}\ B_1\ \underline{\text{do}}\ \underline{\text{if}}\ B_2\ \underline{\text{then}}\ S_1\ \underline{\text{else}}\ S_2 \quad .$$

In the μ-calculus, this can be formulated as

$$\mathcal{A} \vdash (p_2 \to p_1 * A_1, p_1 * A_2) = p_1 * (p_2 \to A_1, A_2)$$

where $\mathcal{A}$ will have to reflect the assumptions made above.

Since no predicate in our formalism is assumed to have side effects, Dijkstra's first assumption does not concern us. The second assumption can be formulated as

$$A_1(p_2\to X,Y) = (p_2\to A_1X,A_1Y)$$
$$A_2(p_2\to X,Y) = (p_2\to A_2X,A_2Y) \ .$$

As a third assumption, not made explicit by Dijkstra, we need the totality of p_2:

$$(p_2\to E,E) = E \ .$$

Thus, our formulation becomes

$$(5)\qquad \left.\begin{array}{l} A_1(p_2\to X,Y) = (p_2\to A_1X,A_1Y), \\ A_2(p_2\to X,Y) = (p_2\to A_2X,A_2Y), \\ \quad (p_2\to E,E) = E \end{array}\right| \!\!\vdash \begin{array}{l} (p_2\to p_1*A_1,p_1*A_2) \\ = \\ p_1 * (p_2\to A_1,A_2) \end{array} .$$

It is possible also to assume somewhat less, by using the following assertion:

$$(6)\qquad \left.\begin{array}{l} A(p\to\Omega,E) = (p\to\Omega,A), \\ A(p\to E,\Omega) = (p\to A,\Omega), \\ A(p\to E,E) = (p\to A,A) \end{array}\right| \!\!\vdash A(p\to X,Y) = (p\to AX,AY) \ .$$

The proof of (6) goes as follows:

$$\begin{array}{l} \vdash A(p\to X,Y) = A(p\to(p\to X,Y),(p\to X,Y)) = \\ \quad A(p\to E,E)(p\to X,Y) = (p\to A,A)(p\to X,Y) = \\ \quad (p\to(p\to E,\Omega)A,(p\to\Omega,E)A)(p\to X,Y) = \\ \quad (p\to A(p\to E,\Omega),A(p\to\Omega,E))(p\to X,Y) = \\ \quad (p\to A(p\to E,\Omega)(p\to X,Y),A(p\to\Omega,E)(p\to X,Y)) = \\ \quad (p\to A(p\to(p\to X,Y),\Omega),A(p\to\Omega,(p\to X,Y))) = \\ \quad (p\to A(p\to X,\Omega),A(p\to\Omega,Y)) = \\ \quad (p\to A(p\to E,\Omega)X,A(p\to\Omega,E)Y) = \\ \quad (p\to(p\to E,\Omega)AX,(p\to\Omega,E)AY) = \\ \quad (p\to AX,AY). \end{array}$$

(6) is in fact a special case of a much more general result, the proof of which is omitted here. Using (6), we can write for (5):

$$\left.\begin{array}{rcl} A_1(p_2\to\Omega,E) & = & (p_2\to\Omega,A_1), \\ A_1(p_2\to E,\Omega) & = & (p_2\to A_1,\Omega), \\ A_2(p_2\to\Omega,E) & = & (p_2\to\Omega,A_2), \\ A_2(p_2\to E,\Omega) & = & (p_2\to A_2,\Omega), \\ (p_2\to E,E) & = & E \end{array}\right| \vdash \begin{array}{c} (p_2\to p_1*A_1,p_1*A_2) \\ = \\ p_1 * (p_2\to A_1,A_2) \end{array} .$$

The proof of (5) offers no difficulties:
Let $P_1 = (p_2\to p_1*A_1,p_1*A_2)$. Then

$$\vdash P_1 = (p_2\to\mu X[(p_1\to A_1X,E)],\mu Y[(p_1\to A_2Y,E)]) .$$

Let $P_2 = p_1 * (p_2\to A_1,A_2)$. Then

$$\begin{aligned} \vdash P_2 &= \mu Z[(p_1\to(p_2\to A_1,A_2)Z,E)] \\ &= \mu Z[(p_1\to(p_2\to A_1Z,A_2Z),E)] \\ &= \mu Z[(p_1\to(p_2\to A_1Z,A_2Z),(p_2\to E,E))] \\ &= \mu Z[(p_2\to(p_1\to A_1Z,E),(p_1\to A_2Z,E))] \end{aligned}$$

Let A be short for the assumptions of (5).

a. Proof of $A \vdash P_1 \subseteq P_2$.

Assume A. We first show

$\vdash (p_2\to\Omega,\mu Y[(p_1\to A_2Y,E)]) \subseteq P_2$.

Clearly,

$\vdash (p_2\to\Omega,\Omega) \subseteq P_2$.

Next, we have to show

$(p_2\to\Omega,Y) \subseteq P_2 \vdash (p_2\to\Omega,(p_1\to A_2Y,E)) \subseteq P_2$.

Applying A and the conditional axioms, we have

$\vdash (p_2\to\Omega,(p_1\to A_2Y,E)) =$
$(p_2\to\Omega,(p_1\to A_2(p_2\to\Omega,Y),E))$.

Applying the f.p.p. and monotonicity, we see that, indeed,

$$(p_2 \to \Omega, Y) \subseteq P_2 \vdash (p_2 \to \Omega, (p_1 \to A_2(p_2 \to \Omega, Y), E)) \subseteq$$
$$(p_2 \to (p_1 \to A_1 P_2, E), (p_1 \to A_2 P_2, E)) \;.$$

Thus, we have proved

$$\vdash (p_2 \to \Omega, \mu Y[(p_1 \to A_2 Y, E)]) \subseteq P_2,$$

by the μ-induction rule.

By symmetry, we have

$$\vdash (p_2 \to \mu X[(p_1 \to A_1 X, E)], \Omega) \subseteq P_2.$$

Combining these, we obtain

$$\begin{aligned} &\vdash (p_2 \to \mu X[(p_1 \to A_1 X, E)], \mu Y[(p_1 \to A_2 Y, E)]) = \\ &\quad (p_2 \to (p_2 \to \mu X[(p_1 \to A_1 X, E)], \Omega), \\ &\qquad (p_2 \to \Omega, \mu Y[(p_1 \to A_2 Y, E)])) \\ &\quad \subseteq (p_2 \to P_2, P_2) = P_2. \end{aligned}$$

b. Proof of $A \vdash P_2 \subseteq P_1$.

Using A, it is easy to verify that

$$\begin{aligned} Z \subseteq P_1 \vdash\; & (p_2 \to (p_1 \to A_1 Z, E), (p_1 \to A_2 Z, E)) \subseteq \\ & (p_2 \to (p_1 \to A_1 P_1, E), (p_1 \to A_2 P_1, E)) = \\ & (p_2 \to (p_1 \to A_1 \mu X[(p_1 \to A_1 X, E)], E), \\ & \qquad (p_1 \to A_2 \mu Y[(p_1 \to A_2 Y, E)], E)) = \\ & (p_2 \to \mu X[(p_1 \to A_1 X, E)], \mu Y[(p_1 \to A_2 Y, E)]) = \\ & P_2. \end{aligned}$$

The result then follows by the μ-induction rule.

4.3. A WHILE STATEMENT EXAMPLE FROM COOPER

In [6], Cooper considers the following three programs:

P_1 is the program

```
1:  if p1(x) then goto 2 else goto 4;
2:  x := f(x);
3:  goto 1;
4:  if p2(y) then goto 5 else goto 7;
5:  y := g(y);
6:  goto 4;
7:  halt.
```

P_2 is the program

```
1:  if $p_2(y)$ then goto 2 else goto 4;
2:  $y := g(y)$;
3:  goto 1;
4:  if $p_1(x)$ then goto 5 else goto 7;
5:  $x := f(x)$;
6:  goto 4;
7:  halt
```

P_3 is the program

```
1:  if $p_1(x)$ then goto 2 else goto 6;
2:  if $p_2(y)$ then goto 3 else goto 9;
3:  $x := f(x)$;
4:  $y := g(y)$;
5:  goto 1;
6:  if $p_2(x)$ then goto 7 else goto 12;
7:  $y := g(y)$;
8:  goto 6;
9:  if $p_1(x)$ then goto 10 else goto 12;
10: $x := f(x)$;
11. goto 9;
12: halt.
```

He proves the equivalence of P_1, P_2 and P_3 in the Manna-Floyd framework, using the predicate calculus. (See the references in section 1; for this specific example, compare also Park [27,29].)

We show how to transliterate this example into the μ-calculus.
Let A_1 correspond to $x := f(x)$ and A_2 to $y := g(y)$.
We note that

1. $A_1A_2 = A_2A_1$.
2. A_1 does not change the value of p_2,
   A_2 does not change the value of p_1,
   hence,
   $A_1(p_2 \rightarrow X, Y) = (p_2 \rightarrow A_1X, A_1Y)$,
   $A_2(p_1 \rightarrow X, Y) = (p_1 \rightarrow A_2X, A_2Y)$.

Moreover, it is apparently necessary to require that p_1 and p_2 be total.

The μ-terms corresponding to P_1, P_2 and P_3 are:

$$P_1 = p_1 * A_1;p_2 * A_2$$

$$P_2 = p_2 * A_2;p_1 * A_1$$

$$P_3 = \mu X[(p_1 \rightarrow (p_2 \rightarrow A_1A_2X, p_1 * A_1), p_2 * A_2)]$$

In the derivation of the μ-term for P_3 we have used the well known technique of associating systems of recursive procedures with flow chart programs. A possible version of this technique runs in this is case as follows: To the program P_3 a system of four procedures is associated, one for each occurrence of a predicate in P_3, with the following declarations:

<u>procedure</u> $P_1'; (p_1 \rightarrow P_2', P_3');$

<u>procedure</u> $P_2'; (p_2 \rightarrow A_1A_2P_1', P_4');$

<u>procedure</u> $P_3'; (p_2 \rightarrow A_2P_3', E);$

<u>procedure</u> $P_4'; (p_1 \rightarrow A_1P_4', E).$

According to the method for associating a term in the μ-language with a system of procedures, as described in section 3.1, we obtain for P_3:

$$P_3 = \mu X[(p_1 \rightarrow \mu Y[(p_2 \rightarrow A_1A_2X, \mu Z[(p_1 \rightarrow A_1Z, E)])], \\ \mu S[(p_2 \rightarrow A_2S, E)])]$$

or , after simplification, and using the * notation for while statements,

$$P_3 = \mu X[(p_1 \rightarrow (p_2 \rightarrow A_1A_2X, p_1 * A_1), p_2 * A_2)] \ .$$

Let Á be the list of equivalences

$$A:\begin{cases} A_1A_2 = A_2A_1 \\ A_1(p_2\to X,Y) = (p_2\to A_1X,A_1Y) \\ A_2(p_1\to X,Y) = (p_1\to A_2X,A_2Y) \\ \quad (p_1\to E,E) = E \\ \quad (p_2\to E,E) = E \end{cases} .$$

Cooper's result can then be formulated as

$$A \vdash P_1 = P_2,\ P_1 = P_3.$$

We prove only $A \vdash P_1 = P_2$; once this is understood, the reader should have no difficulty in proving $A \vdash P_1 = P_3$ himself.
We use the notation

$$A \bullet p = p \quad \text{for} \quad A(p\to X,Y) = (p\to AX,AY).$$

By result (6) of section 4.2, we have, for p_2 total,

$$\begin{aligned} A_1 \circ p_2 = p_2 \text{ iff} \quad & A_1(p_2\to\Omega,E) = (p_2\to\Omega,E)A_1 \\ & A_1(p_2\to E,\Omega) = (p_2\to E,\Omega)A_1 \ . \end{aligned}$$

First we show

(7) $$A_1A_2 = A_2A_1,\ A_1 \circ p_2 = p_2 \vdash A_1;\ p_2 * A_2 = p_2 * A_2;\ A_1 \ .$$

Clearly, $\vdash A_1;\Omega = \Omega;A_1$. Also, it is easily seen that

$$A_1A_2 = A_2A_1,\ A_1 \circ p_2 = p_2,\ A_1Y = YA_1 \vdash A_1(p_2\to A_2Y,E) = (p_2\to A_2Y,E)A_1 \ .$$

Next, we show

(8) $$A_1 \circ p_2 = p_2 \vdash (p_1*A_1) \circ p_2 = p_2 \ .$$

Assume $A_1 \circ p_2 = p_2$. To prove $(p_1*A_1) \circ p_2 = p_2$, we must show

$$\vdash p_1 * A_1;(p_2 \to \Omega, E) = (p_2 \to \Omega, E);p_1 * A_1$$

and

$$\vdash p_1 * A_1;(p_2 \to E, \Omega) = (p_2 \to E, \Omega);p_1 * A_1 \; .$$

The second one will follow from the first by symmetry. In order to prove the first, we have to verify whether

$$X(p_2 \to \Omega, E) = (p_2 \to \Omega, E)X \vdash (p_1 \to A_1X, E)(p_2 \to \Omega, E) = (p_2 \to \Omega, E)(p_1 \to A_1X, E).$$

Using $A_1 \circ p_2 = p_2$, and some manipulations with conditionals, this easily follows.
Replacing, in (7), A_1 by $p_1 * A_1$, we obtain

(9) $$p_1 * A_1;A_2 = A_2;p_1 * A_1, (p_1 * A_1) \circ p_2 = p_2 \vdash p_1 * A_1;p_2 * A_2 = p_2 * A_2;p_1 * A_1.$$

Since the assumptions of (9) follow from A, using (7) with the indices 1 and 2 interchanged, and (8), the proof of

$$A \vdash p_1 * A_1;p_2 * A_2 = p_2 * A_2;p_1 * A_1$$

is complete.

4.4. REDUCTION OF FLOW CHARTS TO WHILE STATEMENTS

In [2], Böhm and Jacopini study the problem whether and how flow charts can be rewritten as while statements. To be more specific, let L_f and L_w be two languages such that

a. L_f and L_w have the same elementary statements and the same predicates.
b. L_f has composition, conditionals and goto statements.
c. L_w has composition, conditionals and only while statements (which can, of course, be considered as a restricted type of goto statements).

They conjecture that it is not the case that for each program in L_f there exists an equivalent program in L_w. (This conjecture has been proved in the mean time by Scott [31].) Next, they introduce an extension of L_w to L_w', which has, in addition to L_w, three special elementary statements (function constants in our terminology), F, T and K, and a special predicate constant ω. These have as intended interpretation:

$$
\begin{aligned}
&F(x) &&= (x,1) \;,\\
&T(x) &&= (x,0) \;,\\
&K((x,1)) &&= x \;,\\
&K((x,0)) &&= x \;,\\
&\omega((x,1)) &&= 1 \;,\\
&\omega((x,0)) &&= 0 \;.
\end{aligned}
$$

They then show that each program in L_f is equivalent to a program in $L_{w'}$.

We give one of their examples of such an equivalence, and present its proof in the μ-calculus.

First we introduce part of their notation:

For each p, $\underline{p}$ is defined by

$$\underline{p} = (p\rightarrow TF,FT) \;.$$

For each X, (X) is defined by

$$(X) = \omega * X \;.$$

Their equivalence then reads:

$$
\begin{aligned}
&\mu X[(p\rightarrow E,A_1(q\rightarrow E,A_2X))] = \\
&F(K\;\underline{p}\;(KTT)\;K(KA_1\;\underline{q}\;(KTT)\;K(KA_2FT))K)K \;.
\end{aligned}
$$

In order to prove this in the μ-calculus, we need a number of equivalences characterizing F, T, K and ω:

$$
A\colon \begin{cases}
FK = E,\\
TK = E,\\
F(\omega\rightarrow X,Y) = FX,\\
T(\omega\rightarrow X,Y) = TY.
\end{cases}
$$

Thus, we have to show

$$A \vdash \mu X[(p\to E, A_1(q\to E, A_2X))] =$$
$$F(K\ \underline{p}\ (KTT)\ K(KA_1\ \underline{q}\ (KTT)\ K(KA_2FT))\,K)K\ .$$

This is done as follows. Assume A. We first define five auxiliary procedures:

(1)	$P_1 = (KTT)$	
(2)	$P_2 = (KA_2FT)$	
(3)	$P_3 = (KA_1\ \underline{q}\ P_1KP_2)$	
(4)	$P_4 = (K\ \underline{p}\ P_1KP_3K)$	
(5)	$P_5 = \mu X[(p\to E, A_1(q\to E, A_2X))]$	

Using these definitions, we have

(6)	$\vdash P_1 = (\omega\to KTTP_1, E)$	(1)
(7)	$\vdash TP_1 = T$	(6), A
(8)	$\vdash P_1 = (\omega\to KTT, E)$	(6), (7)
(9)	$\vdash \underline{p}\ P_1 = (p\to TF, FT)(\omega\to KTT, E)$	
	$= (p\to TTT, FT)$	(8), A
(10)	$\vdash \underline{p}\ P_1K = (p\to TT, F)$	(9), A
(11)	$\vdash \underline{q}\ P_1K = (q\to TT, F)$	(10)
(12)	$\vdash P_2 = (\omega\to KA_2FT, E)$	(2) and similar to (8)
(13)	$\vdash \underline{q}\ P_1KP_2 = (q\to TT, A_2FT)$	(11), (12), A
(14)	$\vdash P_3 = (\omega\to KA_1(q\to TT, A_2FT), E)$	(3), (13) and similar to (8)
(15)	$\vdash \underline{p}\ P_1KP_3 = (p\to TT, A_1(q\to TT, A_2FT))$	(10), (14), A
(16)	$\vdash \underline{p}\ P_1KP_3K = (p\to T, A_1(q\to T, A_2F))$	(15), A
(17)	$\vdash P_4 = \mu X[(\omega\to K(p\to T, A_1(q\to T, A_2F))X, E)]$	(4), (16)
(18)	$P_6 = \mu Y[(\omega\to K(p\to T, A_1(q\to T, A_2FY)), E)]$	definition
(19)	$\vdash TP_4 = T,\ TP_6 = T$	(17), (18), A
(20)	$\vdash P_4 = (\omega\to K(p\to T, A_1(q\to T, A_2FP_4)), E)$	(17), (19)
(21)	$\vdash P_6 \subseteq P_4$	(18), (20)

(22) $\vdash P_6 = (\omega \to K(p \to TP_6, A_1(q \to TP_6, A_2FP_6)), E)$ (18),(19)

(23) $\vdash P_4 \subseteq P_6$ (17),(22)

(24) $\vdash P_4 = P_6$ (21),(23)

There remains the proof that

$$\vdash FP_6K = P_5 \ .$$

a. $\subseteq$.

It is sufficient to show

$FYK \subseteq P_5 \vdash F(\omega \to K(p \to T, A_1(q \to T, A_2FY)), E)K \subseteq P_5$.

This follows from

$FYK \subseteq P_5 \vdash (p \to E, A_1(q \to E, A_2FYK)) \subseteq$
$(p \to E, A_1(q \to E, A_2P_5))$.

b. $\supseteq$.

$\vdash FP_6K = F(\omega \to K(p \to T, A_1(q \to T, A_2FP_6)), E)K =$
$(p \to E, A_1(q \to E, A_2FP_6K))$.

Hence,

$\vdash P_5 \subseteq FP_6K$.

This completes the proof of Böhm and Jacopini's example.

4.5. MCCARTHY'S 91-FUNCTION

McCarthy's 91-function has become a wellknown test case in programming theory. It has been used in particular in Manna's work (Manna and Pnueli [20,21], Manna and McCarthy [19]).

Its original form is:

Let, for integer x,

$$f(x) = (x > 100 \to x-10, f(f(x+11)))$$
$$g(x) = (x > 100 \to x-10, 91)$$

Then $f(x) = g(x)$.

We shall concern ourselves with a slight generalization. Let

$$f'(x) = (x>a \rightarrow x-b, f'(f'(x+b+1)))$$
$$g'(x) = (x>a \rightarrow x-b, a+1-b)$$

for integer $x \geq 0$, $a \geq 0$, $b \geq 1$. (We do not allow $b = 0$ in order to avoid some uninteresting special cases.)

In order to apply the μ-calculus to obtain the proof of $f'(x) = g'(x)$, we have to reflect in some way in our assumptions that we are dealing with the special domain of integers. This is done as follows: We introduce three function constants S_1, M_1, and A_0, and the predicate constant p_0, where S_1 stands for the successor function, M_1 for the predecessor function, A_0 for the zero-function, and p_0 for the test for zero predicate. In other words, S_1, M_1, A_0 and p_0 have as intended interpretation over the domain N of non-negative integers:

$$\begin{array}{lll} S_1(x) & = x + 1 & , \text{for all } x \in N, \\ M_1(x) & = x - 1 & , \text{for all } x \in N,\ x \neq 0 \\ & = \text{undefined} & , \text{for } x = 0, \\ A_0(x) & = 0 & , \text{for all } x \in N, \\ p_0(x) & = 0 & , \text{for all } x \in N,\ x \neq 0 \\ & = 1 & , \text{for } x = 0 \end{array}$$

These four constants are characterized by the list of four equivalences $A = A_1, A_2, A_3, A_4$:

$$\begin{array}{ll} A_1\text{:} & S_1M_1 = E \\ A_2\text{:} & S_1A_0 = A_0 \\ A_3\text{:} & A_0M_1 = \Omega \\ A_4\text{:} & \mu X[(p_0 \rightarrow A_0, M_1XS_1)] = E \quad . \end{array}$$

These equivalences are based on the axioms of Scott [33]; note that A_4 asserts that the function h, defined by $h(x) = (x=0 \rightarrow 0, h(x-1)+1)$, is the identity function.

A_4 can be used to prove the following result in the μ-calculus which is the counterpart of mathematical induction:

$$\frac{\begin{array}{r}\vdash A_0F \subseteq A_0G\\ XF \subseteq XG \vdash XS_1F \subseteq XS_1G\end{array}}{\vdash F \subseteq G}$$

(see also Milner [24])

PROOF. Assume the two premises. We show that then $\vdash F \subseteq G$ follows. By $\mathcal{A}_4$,

$$\vdash F = EF = \mu X[(p_0 \to A_0, M_1XS_1)]F,$$

$$\vdash G = EG = \mu X[(p_0 \to A_0, M_1XS_1)]G.$$

Thus, by the μ-induction rule, it is sufficient to show

$$\vdash \Omega F \subseteq \Omega G$$

which is clear, and

$$XF \subseteq XG \vdash (p_0 \to A_0, M_1XS_1)F \subseteq (p_0 \to A_0, M_1XS_1)G$$

which is easily seen to hold by the two premises.

Next, we introduce some other constants, defined in terms of the given ones, for repeated addition and subtraction, and for dealing with the ">" relation. From now on, we assume that a, b are given integers, which remain fixed throughout the proof. Let $c = \max(a+1, b)$. We define $2c - 2$ new function constants $S_2, S_3, \ldots, S_c$, $M_2, M_3, \ldots, M_c$, and $c + 1$ new predicate constants $p_{>0}, p_{>1}, \ldots, p_{>c}$.

Let, for $1 \leq \gamma \leq c$, $\mathcal{B}_\gamma$ be the equivalence

$$S_\gamma = \underbrace{S_1S_1\ldots S_1}_{\gamma \text{ times}} \quad .$$

Let $\mathcal{C}_\gamma$ be the equivalence

$$M_\gamma = \underbrace{M_1M_1\ldots M_1}_{\gamma \text{ times}} \quad .$$

Let

$$\mathcal{B}^c = \mathcal{B}_2,\mathcal{B}_3,\ldots,\mathcal{B}_c$$
$$\mathcal{C}^c = \mathcal{C}_2,\mathcal{C}_3,\ldots,\mathcal{C}_c \ .$$

Let $\mathcal{D}_0$ be the equivalence

$$(p_{>0}\rightarrow X,Y) = (p_0\rightarrow Y,X)$$

and let $\mathcal{D}_\gamma$ be

$$(p_{>\gamma}\rightarrow X,Y) = (p_0\rightarrow Y,M_1(p_{>\gamma-1}\rightarrow S_1X,S_1Y)) \ .$$

$\mathcal{D}_0$ and $\mathcal{D}_\gamma(\underline{1\leq\gamma\leq c})$ can be understood intuitively as expressing the equivalences

$$(x>0 \rightarrow X(x),Y(x)) = (x=0 \rightarrow Y(x),X(x))$$

and

$$(x>\gamma \rightarrow X(x),Y(x)) =$$
$$(x=0 \rightarrow Y(x),(x-1>\gamma-1 \rightarrow X(x-1+1),Y(x-1+1)))$$

respectively.
Let $\mathcal{D}^c$ be

$$\mathcal{D}^c = \mathcal{D}_0,\mathcal{D}_1,\ldots,\mathcal{D}_c.$$

After these preparations, we are now in a position to formulate McCarthy's result as

(1) $$A,\mathcal{B}^c,\mathcal{C}^c,\mathcal{D}^c \vdash \mu X[(p_{>a}\rightarrow M_b,S_bS_1XX)] = (p_{>a}\rightarrow M_b,A_0S_{a+1}M_b) \ .$$

The proof of this is given in two parts. In the first part, we derive some intermediate results, contained in the list $\mathcal{E}$:

(2) $$A,\mathcal{B}^c,\mathcal{C}^c,\mathcal{D}^c \vdash \mathcal{E}$$

where $\mathcal{E}$ is the list

$$\begin{aligned}
&\mathcal{E}_1: S_b M_b = E\\
&\mathcal{E}_2: S_b S_1 = S_1 S_b\\
&\mathcal{E}_3: (p_{>a} \to E, E) = E\\
&\mathcal{E}_4: (p_{>a} \to X, M_b Y) = (p_{>a} \to X, M_b(p_{>a} \to Z, Y))\\
&\mathcal{E}_5: S_b(p_{>a} \to X, Y) = S_b(p_{>a} \to X, M_b S_b Y)\\
&\mathcal{E}_6: \mu Y[(p_{>a} \to E, S_1 Y)] = (p_{>a} \to E, A_0 S_{a+1})
\end{aligned}$$

and in the second part we show that

$$(3) \qquad \mathcal{E} \vdash \mu X[(p_{>a} \to M_b, S_b S_1 XX)] = (p_{>a} \to M_b, A_0 S_{a+1} M_b)$$

without using the elements of $\mathcal{A}$, $\mathcal{B}^c$, $\mathcal{C}^c$, $\mathcal{D}^c$ again.
We begin with the proof of (3).
Assume $\mathcal{E}$. By $\mathcal{E}_6$,

$$\vdash (p_{>a} \to M_b, A_0 S_{a+1} M_b) = \mu Y[(p_{>a} \to E, S_1 Y)] M_b .$$

By the last result of section 3.5.,

$$\vdash \mu Y[(p_{>a} \to E, S_1 Y)] M_b = \mu Y[(p_{>a} \to M_b, S_1 Y)] \quad .$$

Thus, we can replace (3) by

$$\vdash \mu X[(p_{>a} \to M_b, S_b S_1 XX)] = \mu Y[(p_{>a} \to M_b, S_1 Y)] \ .$$

Call the left hand side P_1 and the right hand side P_2.

a. Proof of $P_1 \subseteq P_2$.

It is sufficient to show that
$\vdash S_b P_2 P_2 \subseteq P_2$.
We verify whether
$\vdash S_b \Omega P_2 \subseteq P_2$
and
$S_b Y P_2 \subseteq P_2 \vdash S_b(p_{>a} \to M_b, S_1 Y) P_2 \subseteq P_2$.

The first is clear. The second assertion is derived as follows:

$$\begin{aligned}
S_bYP_2 \subseteq P_2 \vdash\ & S_b(p_{>a}\rightarrow M_b, S_1Y)P_2 = && (\mathcal{E}_5)\\
& S_b(p_{>a}\rightarrow M_b, M_bS_bS_1Y)P_2 = && \\
& S_b(p_{>a}\rightarrow M_bP_2, M_bS_bS_1YP_2) = && (\mathcal{E}_2)\\
& S_b(p_{>a}\rightarrow M_bP_2, M_bS_1S_bYP_2) \subseteq && (\text{assumption})\\
& S_b(p_{>a}\rightarrow M_bP_2, M_bS_1P_2) = && (\mathcal{E}_4)\\
& S_b(p_{>a}\rightarrow M_bP_2, M_b(p_{>a}\ M_b, S_1P_2)) = && (\text{f.p.p.})\\
& S_b(p_{>a}\rightarrow M_bP_2, M_bP_2) = && (\mathcal{E}_3)\\
& S_bM_bP_2 = && (\mathcal{E}_1)\\
& P_2 &&
\end{aligned}$$

b. Proof of $\vdash P_2 \subseteq P_1$.

It is sufficient to show that

$\vdash P_1 \subseteq S_bP_1P_1$.

By the μ-induction rule, the proof of this follows from the proof of

$X \subseteq P_1,\ X \subseteq S_bS_1S_1 \vdash (p_{>a}\rightarrow M_b, S_bS_1XX) \subseteq P_1,\ (p_{>a}\rightarrow M_b, S_bS_1XX) \subseteq S_bP_1P_1$.

The first conclusion of this assertion follows from the f.p.p. In order to prove

$$X \subseteq P_1,\ X \subseteq S_bP_1P_1 \vdash (p_{>a}\rightarrow M_b, S_bS_1XX) \subseteq S_bP_1P_1$$

we write for the left hand side of its conclusion

$$\begin{aligned}
\vdash\ & (p_{>a}\rightarrow M_b, S_bS_1XX) = && (\mathcal{E}_1)\\
& S_bM_b(p_{>a}\rightarrow M_b, S_bS_1XX) = && (\mathcal{E}_3)\\
& S_b(p_{>a}\rightarrow M_b(p_{>a}\rightarrow M_b, S_bS_1XX),\ M_b(p_{>a}\rightarrow M_b, S_bS_1XX)) = && (\mathcal{E}_4)\\
& S_b(p_{>a}\rightarrow M_b(p_{>a}\rightarrow M_b, S_bS_1XX),\ M_bS_bS_1XX) = && (\mathcal{E}_5)\\
& S_b(p_{>a}\rightarrow M_b(p_{>a}\rightarrow M_b, S_bS_1XX), S_1XX) &&
\end{aligned}$$

and for its right hand side

$$\begin{aligned}
\vdash\ & S_bP_1P_1 = && (\text{f.p.p.})\\
& S_b(p_{>a}\rightarrow M_b, S_bS_1P_1P_1)P_1 = && (\text{f.p.p.}, \mathcal{E}_2)\\
& S_b(p_{>a}\rightarrow M_b(p_{>a}\rightarrow M_b, S_bS_1P_1P_1), S_1S_bP_1P_1P_1) &&
\end{aligned}$$

Comparing the results for the left hand side and right hand side, and using, from left to right, the assumptions $X \subseteq P_1$, $X \subseteq P_1$, $X \subseteq S_bP_1P_1$, and $X \subseteq P_1$, the desired result follows, and the proof of (3) is complete. There remains the proof of

$$\mathcal{A},\ \mathcal{B}^c,\ \mathcal{C}^c,\ \mathcal{D}^c \vdash \mathcal{E}.$$

First we derive some consequences from $\mathcal{A}$.

$$\begin{aligned}
&R_1\colon\ \mathcal{A} \vdash (p_0 \to E, E) = E\\
&R_2\colon\ \mathcal{A} \vdash (p_0 \to E, M_1S_1) = E\\
&R_3\colon\ \mathcal{A} \vdash M_1S_1 = (p_0 \to \Omega, E)\\
&R_4\colon\ \mathcal{A} \vdash (p_0 \to M_1X, Y) = (p_0 \to \Omega, Y)\\
&R_5\colon\ \mathcal{A} \vdash S_1(p_0 \to X, Y) = S_1Y \qquad .
\end{aligned}$$

We prove only R_1, R_3 and R_5.

Proof of R_1: By $\mathcal{A}_4$ and the f.p.p., we have

$$\vdash E = (p_0 \to A_0, M_1ES_1) = (p_0 \to A_0, M_1S_1) =$$
$$(p_0 \to (p_0 \to A_0, M_1S_1),\ (p_0 \to A_0, M_1S_1)) = (p_0 \to E, E).$$

Proof of R_3:

$$\vdash M_1S_1 = EM_1S_1 = (p_0 \to A_0, M_1S_1)M_1S_1 = (p_0 \to A_0M_1S_1, M_1S_1M_1S_1)$$
$$(p_0 \to \Omega, M_1S_1) = (p_0 \to \Omega, (p_0 \to A_0, M_1S_1)) = (p_0 \to \Omega, E) \ .$$

Proof of R_5:

$$\vdash S_1(p_0 \to X, Y) = S_1M_1S_1(p_0 \to X, Y) = S_1(p_0 \to \Omega, E)(p_0 \to X, Y) =$$
$$S_1(p_0 \to \Omega, (p_0 \to X, Y)) = S_1(p_0 \to \Omega, Y) = S_1(p_0 \to \Omega, E)Y =$$
$$S_1M_1S_1Y = S_1Y.$$

We use R_1 to R_5 in the proofs of $\mathcal{E}_1$ to $\mathcal{E}_6$.

a. The proofs of $\mathcal{E}_1$ and $\mathcal{E}_2$ are clear.

b. $\mathcal{E}_3$ follows from R_1, R_2 and $\mathcal{D}^c$: Let $\mathcal{E}_3 = \mathcal{E}_3(a)$.
   $\mathcal{E}_3(0)$ follows from

$\vdash (p_{>0} \rightarrow E, E) = (p_0 \rightarrow E, E) = E.$

Assuming $\mathcal{E}(a-1)$, we have

$\vdash (p_{>a} \rightarrow E, E) = (p_0 \rightarrow E, M_1(p_{>a-1} \rightarrow S_1, S_1)) =$

$(p_0 \rightarrow E, M_1(p_{>a-1} \rightarrow E, E)S_1) = (p_0 \rightarrow E, M_1ES_1) = E.$

c. Proof of $\mathcal{E}_4$. Let $\mathcal{E}_4 = \mathcal{E}_4(a,b)$. We prove $\mathcal{E}_4(a,b)$ by mathematical induction on a and b.

c1. $\mathcal{E}(0,1)$ follows from R_4.

c2. Proof of $\mathcal{E}(a-1,1) \vdash \mathcal{E}(a,1)$. Assume $\mathcal{E}(a-1,1)$. For the left hand side of $\mathcal{E}(a,1)$, we have

$\vdash (p_{>a} \rightarrow X, M_1Y) = (p_0 \rightarrow M_1Y, M_1(p_{>a-1} \rightarrow S_1X, S_1M_1Y)) =$

$(p_0 \rightarrow \Omega, M_1(p_{>a-1} \rightarrow S_1X, Y))$

and for the right hand side of $\mathcal{E}(a,1)$:

$\vdash (p_{>a} \rightarrow X, M_1(p_{>a} \rightarrow Z, Y)) =$

$(p_0 \rightarrow \Omega, M_1(p_{>a-1} \rightarrow S_1X, S_1M_1(p_{>a} \rightarrow Z, Y)))$

$(p_0 \rightarrow \Omega, M_1(p_{>a-1} \rightarrow S_1X, (p_{>a} \rightarrow Z, Y))).$

To establish $\mathcal{E}(a,1)$, it is thus sufficient to show that

$\vdash (p_{>a-1} \rightarrow S_1X, Y) = (p_{>a-1} \rightarrow S_1X, (p_{>a} \rightarrow Z, Y)).$

This follows from

$\mathcal{E}(a-1,1) \vdash (p_{>a-1} \rightarrow U, (p_{>a} \rightarrow V, W)) = (p_{>a-1} \rightarrow U, W)$

which is derived as follows:

$\vdash (p_{>a-1} \rightarrow U, (p_{>a} \rightarrow V, W)) =$

$(p_{>a-1} \rightarrow U, (p_0 \rightarrow W, M_1(p_{>a-1} \rightarrow S_1V, S_1W))) =$

$(p_{>a-1} \rightarrow U, (p_0 \rightarrow W, (p_{>a-1} \rightarrow X, M_1(p_{>a-1} \rightarrow S_1V, S_1W)))) = (\mathcal{E}(a-1,1))$

$(p_{>a-1} \rightarrow U, (p_0 \rightarrow W, (p_{>a-1} \rightarrow X, M_1S_1W))) =$

$(p_{>a-1} \rightarrow U, (p_0 \rightarrow W, M_1S_1W)) =$

$(p_{>a-1} \rightarrow U, (p_0 \rightarrow E, M_1S_1)W) =$

$(p_{>a-1} \rightarrow U, W).$

c3. From c_1 and c_2, the proof of $\mathcal{E}(a,1)$ can be constructed for each given a.

c4. The proof of $\mathcal{E}(a,b-1) \vdash \mathcal{E}(a,b)$ is similar to the above ones, and therefore omitted.

d. Proof of $\mathcal{E}_5$.

d1. Using mathematical induction on b, one first shows that

$\vdash S_b(p_{>b-1} \to X, Y) = S_b X$

where, for $b = 1$, R_5 is used.

d2. Using mathematical induction on b again, one shows that

$\vdash M_b S_b = (p_{>b-1} \to E, \Omega)$

using R_3 for the basis step.

d3. Using these results, we prove $\mathcal{E}_5$:

$\vdash S_b(p_{>a} \to X, M_b S_b Y) =$ (by d2)

$S_b(p_{>a} \to X, (p_{>b-1} \to E, \Omega)Y) =$ (by d1)

$S_b(p_{>b-1} \to (p_{>a} \to X, (p_{>b-1} \to Y, \Omega)), Z) =$

$S_b(p_{>b-1} \to (p_{>a} \to X, Y), Z) =$ (by d1)

$S_b(p_{>a} \to X, Y)$.

e. Proof of $\mathcal{E}_6$.

We use mathematical induction on a. The proof of the basis step

$\vdash \mu Y[(p_{>0} \to E, S_1 Y)] = (p_{>0} \to E, A_0 S_1)$

offers no difficulties.

Assume $\mathcal{E}_6(a-1)$. For the right hand side of $\mathcal{E}_6(a)$ we write

$\vdash (p_{>a} \to E, A_0 S_{a+1}) =$

$(p_0 \to A_0 S_{a+1}, M_1(p_{>a-1} \to S_1, S_1 A_0 S_{a+1})) =$ ($\mathcal{A}_2$)

$(p_0 \to (p_{>a-1} \to S_1, A_0 S_{a+1}), M_1(p_{>a-1} \to S_1, A_0 S_{a+1})) =$

$(p_0 \to (p_{>a-1} \to E, A_0 S_a) S_1, M_1(p_{>a-1} \to E, A_0 S_a) S_1) =$ ($\mathcal{E}_6(a-1)$)

$(p_0 \to \mu Z[(p_{>a-1} \to E, S_1 Z)] S_1, M_1 \mu Z[(p_{>a-1} \to E, S_1 Z)] S_1)$.

(We have specially indicated use of $\mathcal{A}_2$ here, since this is the only place in the proof of the 91-function where this axiom is used.)

There remains the proof of

$$\vdash \mu Y[(p_{>a} \to E, S_1 Y)] = (p_0 \to \mu Z[(p_{>a-1} \to E, S_1 Z)] S_1, M_1 \mu Z[(p_{>a-1} \to E, S_1 Z)] S_1) \;. \tag{4}$$

Call the left hand side P_1 and the right hand side P_2.

For P_1 we have

$$\vdash P_1 = \mu Y[(p_0 \rightarrow S_1Y, M_1(p_{>a-1} \rightarrow S_1, S_1S_1Y))] \quad .$$

Let P_3 be defined by

$$P_3 = \mu Z[(p_{>a-1} \rightarrow S_1, S_1Z)] \ .$$

Then we can write for P_2:

$$\vdash P_2 = (p_0 \rightarrow P_3, M_1P_3) \ .$$

(α) Proof of $P_1 \subseteq P_2$.

It is sufficient to show

$Y \subseteq (p_0 \rightarrow P_3, M_1P_3) \vdash (p_0 \rightarrow S_1Y, M_1(p_{>a-1} \rightarrow S_1, S_1S_1Y)) \subseteq$
$(p_0 \rightarrow P_3, M_1P_3)$

From the assumption

$Y \subseteq (p_0 \rightarrow P_3, M_1P_3)$

we derive

$S_1Y \subseteq S_1(p_0 \rightarrow P_3, M_1P_3)$.

Hence, by R_5,

$S_1Y \subseteq S_1M_1P_3 = P_3$.

Using this last result and the f.p.p., we see that, indeed,

$\vdash (p_0 \rightarrow S_1Y, M_1(p_{>a-1} \rightarrow S_1, S_1S_1Y)) \subseteq$
$(p_0 \rightarrow P_3, M_1(p_{>a-1} \rightarrow S_1, S_1P_3))$.

(β) Proof of $P_2 \subseteq P_1$.

We have to verify

(5) $$(p_0 \rightarrow Z, M_1Z) \subseteq P_1 \vdash (p_0 \rightarrow (p_{>a-1} \rightarrow S_1, S_1Z), M_1(p_{>a-1} \rightarrow S_1, S_1Z)) \subseteq P_1.$$

For P_1 we have

$\vdash P_1 = (p_0 \rightarrow S_1P_1, M_1(p_{>a-1} \rightarrow S_1, S_1S_1P_1))$.

Hence,

$\vdash S_1P_1 = S_1M_1(p_{>a-1} \rightarrow S_1, S_1S_1P_1) = (p_{>a-1} \rightarrow S_1, S_1S_1P_1)$.

Thus,

(6) $$\vdash P_1 = (p_0 \to (p_{>a-1} \to S_1, S_1 S_1 P_1), M_1(p_{>a-1} \to S_1, S_1 S_1 P_1)) \\ = (p_0 \to S_1 S_1 P_1, M_1(p_{>a-1} \to S_1, S_1 S_1 P_1)).$$

Also, from the assumption
$(p_0 \to Z, M_1 Z) \subseteq P_1$, we derive, as above

(7) $$Z \subseteq S_1 P_1 \ .$$

Using (6), we write for (5)
$$(p_0 \to Z, M_1 Z) \subseteq P_1 \vdash (p_0 \to S_1 Z, M_1(p_{>a-1} \to S_1, S_1 Z)) \subseteq \\ (p_0 \to S_1 S_1 P_1, M_1(p_{>a-1} \to S_1, S_1 S_1 P_1)).$$
That this last assertion holds follows from (7).

This completes the proof of E_6; hence, the proof of (2), and, with this, the proof of the generalization of McCarthy's 91-function, i.e. of (1), is completed.

5. REGULAR TERMS AND THEIR NORMAL FORM

In this section, we introduce a subset of the set of terms of our formal language, viz. the set of *regular terms*. Intuitively, the regular terms correspond to flow charts, the predicate boxes of which are labelled by elements of our set of predicate variables, and the action boxes of which are labelled by our set of function variables. It is well known that each such flow chart can be represented by a system of recursive procedures (see e.g. McCarthy [22]). This system can then in turn be represented by a term in the μ-language, in the way described in section 3.1, and such a term will then be a regular term. It is also well known that there is not a one-one correspondence between such flow charts and (systems of) recursive procedures (or, equivalently, between flow charts and terms in the μ-language): No flow chart corresponds e.g. to $\mu X[(p \rightarrow A_1 X A_2, E)]$. Since the equivalence problem for flow charts of the indicated type is decidable (this follows e.g. from Yanov [39]), one may look for a completeness theorem for the μ-calculus restricted to regular terms. The completeness theorem we shall prove in this and the next section is the following:

Let τ_1 *and* τ_2 *be regular terms. Then*
$\models \tau_1 = \tau_2$ *if and only if* $\vdash \tau_1 = \tau_2$

In words, $\vdash \tau_1 = \tau_2$ is a valid assertion, i.e., τ_1^I and τ_2^I denote the same function in all interpretations I, if and only if $\vdash \tau_1 = \tau_2$ is a theorem of the μ-calculus.

Note that this result is only a special case of the general problem: If we know that $\Phi \vdash \psi$ is valid, can we then always obtain a proof of $\Phi \vdash \psi$ as a theorem of the μ-calculus? It has been proved by Scott [33] that the answer to this general question is negative. (Clearly, the validity of the μ-calculus, as justified in section 3.4, guarantees that if $\Phi \vdash \psi$ is a theorem of the μ-calculus, then $\Phi \vdash \psi$ is a valid assertion.)

The present section contains the first half of our proof of the completeness theorem for regular terms. After the definition of these terms, we introduce a normal form for them, derive a number of properties of regular terms in normal form, and show that each regular term is equivalent to a regular term in normal form. In section 6, we then prove that, for σ_1, σ_2 regular terms in normal form, we have

$$\models \sigma_1 = \sigma_2 \text{ iff } \vdash \sigma_1 = \sigma_2 \quad .$$

Definition 5.2 gives the definition of a regular term; it uses definition 5.1, in which the notion of a term being regular in a variable is introduced.

DEFINITION 5.1

a. X is regular in X.
b. If τ_1 does not contain X free, and τ_2 is regular in X, then τ_1 and $\tau_1;\tau_2$ are regular in X.
c. If τ_1 and τ_2 are regular in X, then $(p\rightarrow\tau_1,\tau_2)$ is regular in X, for each predicate variable p.
d. If τ is regular in X and τ is regular in Y, then $\mu Y[\tau]$ is regular in X, for each variable Y.

Examples.
1. AX and $\mu Y[(p\rightarrow AY,X)]$ are regular in X.
2. XA, $(p\rightarrow A_1XA_2,E)$ and $\mu Z[(p\rightarrow ZX,E)]$ are not regular in X.

DEFINITION 5.2

a. Each constant or variable is regular.
b. If τ_1 and τ_2 are regular, then $\tau_1;\tau_2$ and $(p\rightarrow\tau_1,\tau_2)$ are regular, for each predicate variable p.
c. If τ is regular and τ is regular in X, then $\mu X[\tau]$ is regular.

Examples.
1. AX and $\mu Y[(p\rightarrow AY,X)]$ are regular.
2. $\mu X[XA]$, $\mu X[(p\rightarrow A_1XA_2,E)]$ and $\mu Z[(p\rightarrow ZX,E)]$ are not regular.

We now proceed with the definition of a normal form for regular terms. One part of this definition is for technical convenience. This concerns the notions introduced in definitions 5.3, 5.4 and 5.5. By technical convenience we mean that a proof of the completeness theorem which does not use these notions in some way might well be possible. Another part, however, seems to be essential, viz. that property of regular terms which amounts to the fact that for each term $\mu X[\tau_1];\tau_2$ there exists τ' such that $\vdash \mu X[\tau_1];\tau_2 = \mu X[\tau']$. This is the property which was already indicated in the third example of section 3.5, played an important role in the proofs of section 4, and which will be stated precisely in lemmas 5.10, 5.11 and 5.12.

Before we present the definition of normal form for regular terms, we first give the following auxiliary definitions:

DEFINITION 5.3

A term τ may be *free from a predicate* p.

a. Each constant or variable is free from p.

b. $A;\tau$ is free from p, for each variable A and term τ.

c. If τ_1 and τ_2 are free from p, then $(q\to\tau_1,\tau_2)$ is free from p, for each predicate variable q which is different from p.

Examples.

1. E, AX, and $(q\to A(p\to X,E),\Omega)$ are free from p.
2. $(q\to(p\to A_1,A_2),A_3)$ and $\mu X[X]$ are not free from p.

Semantically, if a term τ is free from p, then, for each interpretation I with domain $\mathcal{D}$, and for each $x \in \mathcal{D}$, $\tau^I(x)$ may be assumed not to depend on $p^I(x)$. A more precise statement of this fact will be given in the course of the proof of the completeness theorem (Lemma 6.4.) Roughly speaking, we want to be able to reduce a certain argument on $(p\to\sigma_1,\sigma_2)$ to arguments on σ_1 and σ_2. In our approach, this is, in general, allowed only if σ_1 and σ_2 are free from p. To be more specific, we want to be able to infer from $\models (p\to\sigma_1,\sigma_2) \subseteq \sigma$ that $\models \sigma_1 \subseteq \sigma'$ and $\models \sigma_2 \subseteq \sigma''$, where σ' and σ'' are derived from σ in a certain way to be stated below. This inference is not allowed in general: Although e.g. $\models (p\to(p\to\Omega,E),\Omega) \subseteq \Omega$ is valid, it is not true that $\models (p\to\Omega,E) \subseteq \Omega$ is valid.

Note that no procedure is said to be free from p (this is again a matter of convenience) and that, for $q \neq p$, $(q\to\tau_1,\tau_2)$ is free from p iff both τ_1 and τ_2 are free from p.

Next we introduce the concepts of a *(directly) open* occurrence of a variable in a term (definition 5.4) and of a variable being *shielded* by a term (definition 5.5). The main application of the notion of a directly open occurrence can be found in Lemmas 5.1 and 5.2. For the notion of shielding compare the third remark following the definition of normal form (definition 5.6).

DEFINITION 5.4

A variable X may occur (directly) open in a term τ which is regular in X.

a. X occurs directly open in $(p\to X,\tau_2)$ or in $(p\to\tau_1,X)$, for each p, τ_1, τ_2.

b. If X occurs (directly) open in τ_1 or τ_2, then X occurs (directly) open in $(p\to\tau_1,\tau_2)$ for each p.

c. If X occurs directly open in τ, then X occurs open in τ.

d. If X occurs open in τ, then X occurs open in $A;\tau$, for each variable A (which is $\neq$ X, by regularity).

e. If X occurs open in τ, then X occurs open in $\mu Y[\tau]$, for each Y $\neq$ X (where τ is regular in Y).

DEFINITION 5.5

Let τ be regular in X. τ is said to shield X iff τ contains no open occurrences of X.

Examples.

1. X occurs directly open (and, hence, open) in $(p\to X,E)$ and in $(p\to(q\to X,A_1),A_2)$.
2. X occurs open (but not directly open) in $(p\to A_1(q\to X,E),A_2)$ and in $\mu Y[(p\to AY,X)]$.
3. X does not occur open in X or in $(p\to A_1X,\mu Y[(q\to A_2Y,A_3X)])$.

Note that X shields X, that $A;\tau$ shields X iff τ shields X, that $(p\to\tau_1,\tau_2)$ shields X iff τ_1 and τ_2 shield X, and that $\mu Y[\tau]$ shields X iff τ shields X.

First consequences of definitions 5.4 and 5.5 are lemmas 5.1 and 5.2:

LEMMA 5.1

Let τ be regular in X. Let τ' be the result of replacing, in τ, one or more directly open occurrences of X by τ. Then $\vdash \tau = \tau'$.

PROOF. This follows easily by suitable application of the axioms on conditionals.

Examples:

1. $\vdash (p\rightarrow X,E) = (p\rightarrow(p\rightarrow X,E),E)$
2. $\vdash (p\rightarrow(q\rightarrow X,A_1),A_2) =$
   $(p\rightarrow(q\rightarrow(p\rightarrow(q\rightarrow X,A_1),A_2),A_1),A_2)$.

We prove the second example. First we need an auxiliary result:

$$\vdash (p\rightarrow(q\rightarrow(p\rightarrow X,Y),Z),T) = (p\rightarrow(q\rightarrow X,Z),T).$$

This is shown as follows:

$$\begin{aligned}\vdash\ & (p\rightarrow(q\rightarrow(p\rightarrow X,Y),Z),T) = \\ & (p\rightarrow(p\rightarrow(q\rightarrow(p\rightarrow X,Y),Z),\ (q\rightarrow(p\rightarrow X,Y),Z)),T) = \\ & (p\rightarrow(q\rightarrow(p\rightarrow(p\rightarrow X,Y),(p\rightarrow X,Y)),(p\rightarrow Z,Z)),T) = \\ & (p\rightarrow(q\rightarrow(p\rightarrow X,Y),(p\rightarrow Z,Z)),T) = \\ & (p\rightarrow(p\rightarrow(q\rightarrow X,Z),(q\rightarrow Y,Z)),T) = \\ & (p\rightarrow(q\rightarrow X,Z),T)\ .\end{aligned}$$

Replacing in this auxiliary result X by $(q\rightarrow X,A_1)$, Y by A_2, Z by A_1 and T by A_2 we obtain

$$\begin{aligned}\vdash\ & (p\rightarrow(q\rightarrow(p\rightarrow(q\rightarrow X,A_1),A_2),A_1),A_2) = \\ & (p\rightarrow(q\rightarrow(q\rightarrow X,A_1),A_1,A_2)\ .\end{aligned}$$

Hence,

$$\begin{aligned}\vdash\ & (p\rightarrow(q\rightarrow(p\rightarrow(q\rightarrow X,A_1),A_2),A_1),A_2) = \\ & (p\rightarrow(q\rightarrow X,A_1),A_2).\end{aligned}$$

LEMMA 5.2

Let τ be regular in X. Let τ' be the result of replacing, in τ, one or more directly open occurrences of X by Ω. Then $\vdash \mu X[\tau] = \mu X[\tau']$.

Examples:

1. $\vdash \mu X[(p\to X,E)] = \mu X[(p\to\Omega,E)]$
2. $\vdash \mu X[(p\to(q\to X,A_1),A_2X)] =$
   $\mu X[(p\to(q\to\Omega,A_1),A_2X)]$.

PROOF. If τ has no directly open occurrences of X, we have nothing to prove. Otherwise, let Y_1, Y_2 be two variables which do not occur free in τ. Then there exists τ'' such that $\tau = \tau''[X/Y_1][X/Y_2]$, where τ'' is regular in Y_1 and Y_2, and Y_1 occurs directly open in τ''. (In example 2 above, $\tau'' = (p\to(q\to Y_1,A_1),A_2Y_2))$. It is sufficient to show that

$$\vdash \mu Y_1[\mu Y_2[\tau'']] = \mu Y_2[\tau''[\Omega/Y_1]].$$

a. $\supseteq$ is clear.

b. Proof of $\subseteq$. Applying the μ-induction rule twice, we see that it is sufficient to show that

$$\left.\begin{array}{l} Y_1 \subseteq \mu Y_2[\tau''[\Omega/Y_1]] \\ Y_2 \subseteq \mu Y_2[\tau''[\Omega/Y_1]] \end{array}\right| \vdash \tau'' \subseteq \mu Y_2[\tau''[\Omega/Y_1]] .$$

Let $\tau''' = \mu Y_2[\tau''[\Omega/Y_1]]$. Then

$\vdash \tau''' = \tau''[\Omega/Y_1][\tau'''/Y_2]$.

Thus, we have to show

$Y_1 \subseteq \tau'''$, $Y_2 \subseteq \tau''' \vdash \tau'' \subseteq \tau''[\Omega/Y_1][\tau'''/Y_2]$.

Since Y_1 occurs directly open in τ'', we have, using lemma 5.1,

$\vdash \tau''[\Omega/Y_1][\tau'''/Y_2] = \tau''[\tau'''/Y_1][\tau'''/Y_2]$.

The desired result then follows by monotonicity.

We are now in a position to give the definition of a regular term in normal form:

DEFINITION 5.6

a. Each constant or variable is in normal form.

b. If σ is in normal form, then $A;\sigma$ is in normal form, for each variable A.

c. If σ_1 and σ_2 are in normal form, and are both free from p, then $(p\to\sigma_1,\sigma_2)$ is in normal form, for each p.

d. If σ is in normal form, σ is regular in X, σ shields X, and σ is not a procedure, then $\mu X[\sigma]$ is in normal form.

Examples:

1. Ω, E, A, $A_1(p\to A_2,A_3)$, $\mu X[(p\to AX,E)]$,
   $\mu X[(p_1\to A_1\mu Y[(p\to A_2Y,A_3X)],E)]$
   are in normal form.
2. $(p\to A_1,A_2)A_3$, $(p\to(p\to A,E),E)$, $(p\to\mu X[X],E)$,
   $\mu X[XA]$, $\mu X[\mu Y[E]]$, $\mu X[(p\to X,E)]$,
   $\mu X[(p\to A(q\to X,E),E)]$, $\mu X[X];A$
   are not in normal form.

Remarks:

1. A regular term in normal form will also be called a normal term. σ, σ_1, σ' etc. always stand for normal terms.
2. Some of the general forms of regular terms which are <u>not</u> in normal form are: $(p\to\tau_1,\tau_2);\tau_3$, $\Omega;\tau$ or $E;\tau$, $\mu X[\tau_1];\tau_2$, $(p\to(p\to\tau_1,\tau_2),\tau_3)$ and its analogues, $(p\to\mu X[\tau_1],\tau_2)$ and similarly, $\mu X[\mu Y[\tau]]$, and $\mu X[\tau]$, where τ contains one or more open occurrences of X.
   Some instances of such terms not in normal form, and their corresponding equivalent terms in normal form are
   $(p\to A_1,A_2)A_3$ and $(p\to A_1A_3,A_2A_3)$,
   $\Omega;A$ and Ω,
   $E;A$ and A,
   $\mu X[(p\to A_1X,E)]A_2$ and $\mu X[(p\to A_1X,A_2)]$,
   $(p\to(p\to A_1,A_2),A_3)$ and $(p\to A_1,A_3)$,
   $(p\to\mu X[(p\to AX,E)],E)$ and $(p\to A\mu X[(p\to AX,E)],E)$,
   $\mu X[\mu Y[(p\to A_1X,A_2Y)]]$ and $\mu X[(p\to A_1X,A_2\mu Y[(p\to A_1X,A_2Y)])]$,
   $\mu X[(p\to A(p\to X,E),E)]$ and $(p\to A\mu U[(p\to AU,E)],E)$.
3. If $\mu X[\sigma]$ is in normal form, we want $\sigma[\mu X[\sigma]/X]$ also to be in normal form. Without the restriction on shielding in clause d, this would not

be true in general. See
$\mu X[\tau] = \mu X[(p \to A(p \to X, E), E)]$ and
$\tau[\mu X[\tau]/X] =$
$(p \to A(p \to \mu X[(p \to A(p \to X, E), E)], E), E)$.
Since $(p \to \mu X[.], E)$ is not in normal form, the whole last term is not in normal form.

Before we can give the proof that each regular term is equivalent to a normal term, we need a number of properties of normal terms, contained in lemmas 5.3 to 5.12.

LEMMA 5.3

Let σ_1, σ_2 be normal terms, let σ_1 be regular in X and shield X. Then

a. $\sigma_1[\sigma_2/X]$ is a normal term.

b. If σ_1 is free from p, and $\sigma_1 \not\equiv X$, then $\sigma_1[\sigma_2/X]$ is free from p, for each variable p.

PROOF. We use induction on the complexity of σ_1, to prove both statements simultaneously.

1. If σ_1 is a constant or variable, both statements are clear.
2. Let $\sigma_1 = A;\sigma_{11}$, for some variable $A(\not\equiv X$, by regularity).
   Since σ_{11} is regular in X and shields X, we have that $\sigma_{11}[\sigma_2/X]$ is a normal term, by the induction hypothesis. Since $(A;\sigma_{11})[\sigma_2/X] =$ $= A;(\sigma_{11}[\sigma_2/X])$, statement a follows. Statement b is clear.
3. Let $\sigma_1 = (q \to \sigma_{11}, \sigma_{12})$. Since σ_{11} and σ_{12} are regular in X and shield X, both $\sigma_{11}[\sigma_2/X]$ and $\sigma_{12}[\sigma_2/X]$ are normal terms, by the induction hypothesis is for a. Since σ_1 shields X, both $\sigma_{11} \not\equiv X$ and $\sigma_{12} \not\equiv X$. Since σ_{11} and σ_{12} are free from q, we have that $\sigma_{11}[\sigma_2/X]$ and $\sigma_{12}[\sigma_2/X]$ are free from q, by the induction hypothesis for b. Since $(q \to \sigma_{11}, \sigma_{12})[\sigma_2/X]$ $= (q \to \sigma_{11}[\sigma_2/X], \sigma_{12}[\sigma_2/X])$, statement a follows. As to b, if $p = q$, we have nothing to prove. Otherwise, suppose that $(q \to \sigma_{11}, \sigma_{12})$ is free from p. Then σ_{11}, σ_{12} are free from p, and both are $\not\equiv X$. Thus, $\sigma_{11}[\sigma_2/X]$ and $\sigma_{12}[\sigma_2/X]$ are free from p, by the induction hypothesis for b. Thus, $(q \to \sigma_{11}, \sigma_{12})[\sigma_2/X]$ is free from p.

4. Let $\sigma_1 = \mu Y[\sigma_{11}]$. Since σ_{11} is regular in X and shields X, $\sigma_{11}[\sigma_2/X]$ is a normal term, by induction. We have $\mu Y[\sigma_{11}][\sigma_2/X] = \mu Y[\sigma_{11}[\sigma_2/X]]$, assuming that Y does not occur free in σ_2, and, moreover, $\sigma_{11}[\sigma_2/X]$ shields Y. Thus, $\mu Y[\sigma_{11}[\sigma_2/X]]$ is a normal term. This proves a. Since no procedure is free from p, b is trivially satisfied.

COROLLARY 5.3

If $\mu X[\sigma]$ is a normal term, then $\sigma[\mu X[\sigma]/X]$ is a normal term which is not a procedure.

PROOF. Follows from lemma 5.3 and the fact that, if $\mu X[\sigma]$ is a normal term, then σ is not a procedure.

LEMMA 5.4

If σ is a normal term which is regular in X, then, for each variable A, $\sigma[AX/X]$ is a normal term, regular in X and shielding X.

PROOF. Follows easily from the definition of shielding.

For the next two lemmas, we need the notion of *subterm* of a normal term. In this definition, we use the following notation: If T is a set of terms $\{\tau_1,\tau_2,\ldots,\tau_n\}$, then $T[\tau/X]$ is the set of terms $\{\tau_1[\tau/X], \tau_2[\tau/X],\ldots,\tau_n[\tau/X]\}$.

DEFINITION 5.7

The set $\Sigma(\sigma)$ of all subterms of a normal term σ is defined by

1. $\Sigma(\sigma) = \{\Omega\} \cup \Sigma_0(\sigma)$.

2.1. $\Sigma_0(\Omega) = \{\Omega\}$, $\Sigma_0(E) = \{E\}$, $\Sigma_0(A) = \{A\}$.

2.2. $\Sigma_0(A;\sigma) = \{A, A;\sigma\} \cup \Sigma_0(\sigma)$

2.3. $\Sigma_0((p\to\sigma_1,\sigma_2)) = \Sigma_0(\sigma_1) \cup \Sigma_0(\sigma_2) \cup \{(p\to\sigma',\sigma'') \mid \sigma' \in \Sigma_0(\sigma_1),\ \sigma'' \in \Sigma_0(\sigma_2)\}$.

2.4. $\Sigma_0(\mu X[\sigma]) = \{\mu X[\sigma]\} \cup \Sigma_0(\sigma) \cup \Sigma_0(\sigma)[\mu X[\sigma]/X]$.

Remarks:

1. For each σ, $\Sigma(\sigma)$ is a finite set.
2. For each σ, $\sigma \in \Sigma_0(\sigma)$.
3. It is convenient to have Ω as a subterm of each term (hence clause 1) and to have e.g. $(p\to\sigma_1,\sigma_2)$ as a subterm of $(p\to(p\to\sigma_1,\sigma_3),\sigma_2)$ (hence clause 2.3).
4. Since $\sigma \in \Sigma_0(\sigma)$, we have $\sigma[\mu X[\sigma]/X] \in \Sigma_0(\sigma)[\mu X[\sigma]/X] \subseteq \Sigma_0(\mu X[\sigma])$.
5. The notion of subterm will be of importance in a certain termination argument in section 6.

LEMMA 5.5

For each normal term $\mu X[\sigma]$ there exists a σ', such that $\vdash \mu X[\sigma] = \sigma'$, σ' not a procedure, and σ' a subterm of $\mu X[\sigma]$.

PROOF. Follows from corollary 5.3, definition 5.7 and the remarks following it.

LEMMA 5.6

Let σ_1, σ_2 be two normal terms. There exist σ', σ'' such that, for each predicate variable p,

a. $\vdash (p\to\sigma_1,\sigma_2) = (p\to\sigma',\sigma'')$.

b. $(p\to\sigma',\sigma'')$ is a normal term.

c. σ', σ'' are subterms of σ_1, σ_2 respectively.

PROOF. We show the existence of σ'; the proof for σ'' is similar. We use induction on the complexity of σ_1. By lemma 5.5, we may assume that σ_1 is not a procedure. If σ_1 is a constant or variable, or if σ_1 has the form $A;\sigma_{11}$, then σ_1 is clearly free from p, and $\sigma' = \sigma_1$. Now suppose $\sigma_1 = (q\to\sigma_{11},\sigma_{12})$, for some predicate variable q.

If $q \equiv p$, we have

$$\vdash (p\to(p\to\sigma_{11},\sigma_{12}),\sigma_2) = (p\to\sigma_{11},\sigma_2).$$

Since σ_{11} has less complexity than σ_1, there exists, by induction, a normal σ''', free from p, such that σ''' is a subterm of σ_{11}, and such that $\vdash (p\to\sigma_{11},\sigma_2) = (p\to\sigma''',\sigma_2)$. Since σ_{11} is a subterm of $(p\to\sigma_{11},\sigma_{12})$, the statement of the lemma follows.

If $q \neq p$, we use the equivalence

$$\vdash (p\to(q\to\sigma_{11},\sigma_{12}),\sigma_2) = \\ (p\to(q\to(p\to\sigma_{11},\sigma_2), (p\to\sigma_{12},\sigma_2)),\sigma_2).$$

By the induction hypothesis, applied to $(p\to\sigma_{11},\sigma_2)$ and $(p\to\sigma_{12},\sigma_2)$, there exist σ''', $\sigma^{'V}$ such that

1. σ''', $\sigma^{'V}$ are free from p.
2. $\vdash (p\to\sigma_{11},\sigma_2) = (p\to\sigma''',\sigma_2)$,
   $\vdash (p\to\sigma_{12},\sigma_2) = (p\to\sigma^{'V},\sigma_2)$.
3. σ''', $\sigma^{'V}$ are subterms of σ_{11}, σ_{12} respectively.

Then

$$\vdash (p\to(q\to\sigma_{11},\sigma_{12}),\sigma_2) = \\ (p\to(q\to(p\to\sigma_{11},\sigma_2),(p\to\sigma_{12},\sigma_2)),\sigma_2) = \\ (p\to(q\to(p\to\sigma''',\sigma_2),(p\to\sigma^{'V},\sigma_2)),\sigma_2) = \\ (p\to(q\to\sigma''',\sigma^{'V}),\sigma_2).$$

Since $(q\to\sigma''',\sigma^{'V})$ is free from p and is a subterm of $\sigma_1 = (q\to\sigma_{11},\sigma_{12})$, the statement of the lemma follows.

In the proof of lemma 5.8, which is one of the two key results in the proof of the normal form theorem, we need the following auxiliary lemma:

LEMMA 5.7

Let σ_1, σ_2 be terms which do not contain U, V, W or S free. Then

$$\vdash \mu X[(p\to\sigma_1,\sigma_2)] = \\ (p\to\mu U[\sigma_1[(p\to U,\mu V[\sigma_2[(p\to U,V)/X]])/X]], \\ \mu W[\sigma_2[(p\to\mu S[\sigma_1[(p\to S,W)/X]],W)/X]]).$$

PROOF. The straightforward, but somewhat tedious proof of this lemma is left to the reader. A special case of lemma 5.7 was proved as result R_3 of section 4.1.

Example:

$$\vdash \mu X[(p \to A_1(q \to X, E),\ A_2(r \to X, E))] =$$
$$(p \to \mu U[A_1(q \to (p \to U, \mu V[A_2(r \to (p \to U, V), E)]), E)],$$
$$\mu W[A_2(r \to (p \to \mu S[A_1(q \to (p \to S, W), E)], W), E)]) \ .$$

The idea behind this lemma may become clear if one constructs the μ-term for the procedure P_1 which is declared by the system

$$\underline{\text{procedure}}\ P_1;(p \to P_2, P_3);$$
$$\underline{\text{procedure}}\ P_2;\sigma_1[P_1/X];$$
$$\underline{\text{procedure}}\ P_2;\sigma_2[P_2/X].$$

LEMMA 5.8

Let σ be a normal term which is regular in X and which is not a procedure. There exists a normal term σ', σ' not a procedure, such that $\vdash \mu X[\sigma] = \sigma'$.

PROOF. Note that $\mu X[\sigma]$ itself is not necessarily a normal term, since σ may contain open occurrences of X.
We use an auxiliary function $\alpha(X,\sigma)$ which is defined for σ as in the statement of the lemma by

a. $\alpha(X,E) = \alpha(X,\Omega) = \alpha(X,A) = \alpha(X,A;\sigma) = 0$.

b. $\alpha(X,X) = \infty$.

c. $\alpha(X,(p \to \sigma_1,\sigma_2)) = 1+\max(\alpha(X,\sigma_1),\alpha(X,\sigma_2))$.

Examples are

$$\alpha(X,(p \to X,E)) = \infty, \text{ and}$$

$$\alpha(X,(p \to A_1(q \to X,E),(r \to A_2(s \to X,E),A_3X)))) = 2.$$

Since neither σ_1 nor σ_2 in clause c is a procedure, $\alpha(X,\sigma)$ is either a non-negative integer, or ∞. First we consider the case that $\alpha(X,\sigma) = \infty$. This case occurs iff either

a. $\sigma = X$, or

b. σ contains one or more directly open occurrences of X.

In case a, we use $\mu X[X] = \Omega$, in case b we apply lemma 5.2, yielding $\vdash \mu X[\sigma] = \mu X[\sigma']$, where σ' results from σ by replacing all directly open occurrences of X in σ by Ω. Then, $\alpha(X,\sigma') < \infty$.

Next, we give the proof for $\alpha(X,\sigma) < \infty$ by induction on the integer $\alpha(X,\sigma)$. (See also the example following this proof.)

a. $\alpha(X,\sigma) = 0$. If $\sigma = E$, Ω or A, the statement follows immediately. If $\sigma = A;\sigma_1$, we apply $\vdash \mu X[A;\sigma_1] = A;\mu X[\sigma_1[AX/X]]$ (see result R_2 of section 4.1.) By lemma 5.5, we may assume that σ_1 is not a procedure. Then, by lemma 5.4, $\mu X[\sigma_1[AX/X]]$ is in normal form, and the desired result follows.

b. $\alpha(X,\sigma) = n > 0$. From the definition of $\alpha(X,\sigma)$ it follows that then $\sigma = (p\rightarrow\sigma_1,\sigma_2)$. By lemma 5.7, we have

$$\vdash \mu X[(p\rightarrow\sigma_1,\sigma_2)] =$$
$$(p\rightarrow\mu U[\sigma_1[(p\rightarrow U,\mu V[\sigma_2[(p\rightarrow U,V)/X]])/X]],$$
$$\mu W[.]).$$

First we consider $\mu V[\sigma_2[(p\rightarrow U,V)/X]]$. If $\sigma_2[(p\rightarrow U,V)/X]$ is not in normal form, it can be brought into normal form by suitable application of the conditional axioms. Let the result be σ_2'. Consider $\mu V[\sigma_2']$. Clearly, σ_2' is not a procedure, and, also, $\alpha(V,\sigma_2') < n$. Thus, by the induction hypothesis, there exists a normal σ_2'', with $\vdash \mu V[\sigma_2'] = \sigma_2''$.

Next, consider $\mu U[\sigma_1[(p\rightarrow U,\sigma_2'')/X]]$. Again, after suitable application of the conditional axioms, we obtain a normal σ_1', such that $\vdash \sigma_1[(p\rightarrow U,\sigma_2'')/X] = \sigma_1'$, σ_1' not a procedure. Also, $\alpha(U,\sigma_1') < n$. Thus, by induction, there exists a normal term σ_1'' such that $\vdash \mu U[\sigma_1'] = \sigma_1''$. Similarly, we derive $\vdash \mu W[.] = \sigma_2'''$, for some normal σ_2'''. Then $\mu X[(p\rightarrow\sigma_1,\sigma_2)] = (p\rightarrow\sigma_1'',\sigma_2''')$. From the construction it follows that σ_1'' and σ_2''' are free from p. Hence, $(p\rightarrow\sigma_1'',\sigma_2''')$ is in normal form.

This completes the proof of lemma 5.8.

Example:

We derive the normal form of

$$\mu X[(p\to(q\to A(p\to X,E),E),X)]\ .$$

We have

$$\begin{aligned}
&\vdash \mu X[(p\to(q\to A(p\to X,E),E),X)] = && (\alpha(X,\sigma) = \infty)\\
&\mu X[(p\to(q\to A(p\to X,E),E),\Omega)] = && (\alpha(X,\sigma') = 2)\\
&(p\to\mu U[(q\to A(p\to(p\to U,\mu V[\Omega]),E),E)],\\
&\quad \mu W[\Omega])\ .
\end{aligned}$$

Also,

$$\begin{aligned}
&\vdash \mu U[(q\to A(p\to(p\to U,\mu V[\Omega]),E),E)] =\\
&\mu U[(q\to A(p\to U,E),E)] = && (\alpha(U,\sigma'') = 1)\\
&(q\to\mu V[A(p\to(q\to V,\mu W[E]),E)],\ \mu S[E]) =\\
&(q\to A\mu V[(p\to(q\to AV,E),E)],E)\ .
\end{aligned}$$

Thus, the desired normal form is

$$(p\to(q\to A\mu V[(p\to(q\to AV,E),E)],E),\Omega).$$

The next group of lemmas contains the second key result to be used in the proof of the normal form theorem. They are concerned with the reduction of a term $\mu X[\sigma_1];\sigma_2$ to an equivalent normal term $\mu X[\sigma']$. This reduction is based on the following definition:

DEFINITION 5.8

Let ξ be a finite (possibly empty) set of variables, and let σ_1, σ_2 be normal terms.

$\sigma_1 \underset{\xi}{\circ} \sigma_2$ is defined by

a. $\Omega \underset{\xi}{\circ} \sigma_2 = \Omega$,

b. $E \underset{\xi}{\circ} \sigma_2 = \sigma_2$,

c. $A \underset{\xi}{\circ} \sigma_2 = A$, if $A \in \xi$

$A;\sigma_2$, if $A \notin \xi$,

d. $(A;\sigma_{11}) \underset{\xi}{\circ} \sigma_2 = A;(\sigma_{11} \underset{\xi}{\circ} \sigma_2)$,

e. $(p\rightarrow\sigma_{11},\sigma_{12}) \underset{\xi}{\circ} \sigma_2 = (p\rightarrow\sigma_{11} \underset{\xi}{\circ} \sigma_2,\sigma_{12} \underset{\xi}{\circ} \sigma_2)$,

f. $\mu X[\sigma_{11}] \underset{\xi}{\circ} \sigma_2 = \mu X[\sigma_{11} \underset{\xi\cup\{X\}}{\circ} \sigma_2]$,

provided that σ_2 does not contain X free.

LEMMA 5.9

If σ_1 is regular in X, σ_1 shields X, σ_1 is not a procedure, σ_2 is not a procedure, σ_2 does not contain X free, then $\sigma_1 \underset{\{X\}}{\circ} \sigma_2$ is regular in X, shields X and is not a procedure.

PROOF. Follows easily from definition 5.8.

Lemmas 5.10 and 5.11 are preparatory to lemma 5.12, which contains the main result on the "$\underset{\xi}{\circ}$" operation.

LEMMA 5.10

Let σ_1 be a normal term which is regular in X, let σ_2, σ_3 be normal terms which do not contain X free, and let ξ be a finite set of variables such that $X \notin \xi$. Then

$$X;\sigma_2 \subseteq \sigma_3 \vdash \sigma_1 \underset{\xi}{\circ} \sigma_2 \subseteq (\sigma_1 \underset{\xi\cup\{X\}}{\circ} \sigma_2)[\sigma_3/X]$$

PROOF. We use induction on the complexity of σ_1.

a. $\sigma_1 = \Omega$.

Since $\Omega \underset{\xi}{\circ} \sigma_2 = \Omega$, this case is clear.

b. $\sigma_1 = E$.

Then $\sigma_1 \underset{\xi}{\circ} \sigma_2 = \sigma_2$, and $(\sigma_1 \underset{\xi\cup\{X\}}{\circ} \sigma_2)[\sigma_3/X] = \sigma_2$,

since σ_2 does not contain X free.

c. $\sigma_1 = A \neq X$.
Then $(A \underset{\xi\cup\{X\}}{\circ} \sigma_2)[\sigma_3/X] = A \underset{\xi\cup\{X\}}{\circ} \sigma_2$, and

$A \underset{\xi}{\circ} \sigma_2 = A \underset{\xi\cup\{X\}}{\circ} \sigma_2$, and the assertion follows.

d. $\sigma_1 = X$.
Since $X \underset{\xi}{\circ} \sigma_2 = X;\sigma_2$ (because of $X \notin \xi$), and

$(X \underset{\xi\cup\{X\}}{\circ} \sigma_2)[\sigma_3/X] = X[\sigma_3/X] = \sigma_3$, the result follows from

$X;\sigma_2 \subseteq \sigma_3 \vdash X;\sigma_2 \subseteq \sigma_3$.

e. $\sigma_1 = A;\sigma_{11}$.
Note that $A \neq X$ by regularity. Then

$(A;\sigma_{11}) \underset{\xi}{\circ} \sigma_2 = A;(\sigma_{11} \underset{\xi}{\circ} \sigma_2)$, and

$(A;\sigma_{11} \underset{\xi\cup\{X\}}{\circ} \sigma_2)[\sigma_3/X] = A;(\sigma_{11} \underset{\xi\cup\{X\}}{\circ} \sigma_2)[\sigma_3/X]$.

Since

$X;\sigma_2 \subseteq \sigma_3 \vdash \sigma_{11} \underset{\xi}{\circ} \sigma_2 \subseteq (\sigma_{11} \underset{\xi\cup\{X\}}{\circ} \sigma_2)[\sigma_3/X]$,

holds by the induction hypothesis, the result follows by monotonicity.

f. $\sigma_1 = (p\rightarrow\sigma_{11},\sigma_{12})$.
This is similar to case e.

g. $\sigma_1 = \mu Y[\sigma_{11}]$.
We have to show

$X;\sigma_2 \subseteq \sigma_3 \vdash \mu Y[\sigma_{11}] \underset{\xi}{\circ} \sigma_2 \subseteq (\mu Y[\sigma_{11}] \underset{\xi\cup\{X\}}{\circ} \sigma_2)[\sigma_3/X]$,

or, by definition 5.8,

$X;\sigma_2 \subseteq \sigma_3 \vdash \mu Y[\sigma_{11} \underset{\xi\cup\{Y\}}{\circ} \sigma_2] \subseteq \mu Y[\sigma_{11} \underset{\xi\cup\{X,Y\}}{\circ} \sigma_2][\sigma_3/X]$

or, by definition of substitution,

$X;\sigma_2 \subseteq \sigma_3 \vdash \mu Y[\sigma_{11} \underset{\xi\cup\{Y\}}{\circ} \sigma_2] \subseteq \mu Y[(\sigma_{11} \underset{\xi\cup\{X,Y\}}{\circ} \sigma_2)[\sigma_3/X]]$.

Since, by the induction hypothesis,

$X;\sigma_2 \subseteq \sigma_3 \vdash \sigma_{11} \underset{\xi\cup\{Y\}}{\circ} \sigma_2 \subseteq (\sigma_{11} \underset{\xi\cup\{X,Y\}}{\circ} \sigma_2)[\sigma_3/X]$,

the desired result follows from the fact that

$$\frac{\Phi \vdash \tau_1 \subseteq \tau_2}{\Phi \vdash \mu Y[\tau_1] \subseteq \mu Y[\tau_2]}$$

which holds provided that Y does not occur free in Φ.

This completes the proof of lemma 5.10.

LEMMA 5.11

Let σ_i, $1 \leq i \leq n+1$, be regular in each $X_1, X_2, \ldots, X_n$ and let σ be a normal term which does not contain any of the X_i free. Then

$$X_1 \subseteq \sigma_1;\sigma, X_2 \subseteq \sigma_2;\sigma, \ldots, X_n \subseteq \sigma_n;\sigma \vdash$$
$$\sigma_{n+1} \underset{\{X_1,X_2,\ldots,X_n\}}{\circ} \sigma \subseteq \sigma_{n+1}[\sigma_1/X_1][\sigma_2/X_2]\ldots[\sigma_n/X_n];\sigma .$$

PROOF. Induction on the complexity of σ_{n+1}.

a. $\sigma_{n+1} = \Omega$ or $\sigma_{n+1} = E$.
Then the result is clear.

b. $\sigma_{n+1} = A \neq X_i$, $i = 1,2,\ldots,n$.
Then $\sigma_{n+1} \underset{\{X_1,X_2,\ldots,X_n\}}{\circ} \sigma = A;\sigma$, and

$\sigma_{n+1}[\sigma_1/X_1][\sigma_2/X_2]\ldots[\sigma_n/X_n];\sigma = A;\sigma$,

whence the result.

c. $\sigma_{n+1} = X_i$, for some i, $1 \leq i \leq n$.

Then $\sigma_{n+1} \underset{\{X_1,\ldots,X_n\}}{\circ} \sigma = X_i$, and

$\sigma_{n+1}[\sigma_1/X_1][\sigma_2/X_2]\ldots[\sigma_n/X_n];\sigma = \sigma_i;\sigma$. The result then follows from

$X_1 \subseteq \sigma_1;\sigma$, $X_2 \subseteq \sigma_2;\sigma, \ldots, X_n \subseteq \sigma_n;\sigma \vdash X_i \subseteq \sigma_i;\sigma$.

d. $\sigma_{n+1} = A;\sigma'$.
Follows easily by the induction hypothesis and monotonicity.

e. $\sigma_{n+1} = (p \rightarrow \sigma', \sigma'')$.
Similar to d.

f. $\sigma_{n+1} = \mu Y[\sigma']$.
We have to show

$$X_1 \subseteq \sigma_1;\sigma,\ X_2 \subseteq \sigma_2;\sigma,\ldots,X_n \subseteq \sigma_n;\sigma \vdash$$
$$\mu Y[\sigma'] \underset{\{X_1,X_2,\ldots,X_n\}}{\circ} \sigma \subseteq \mu Y[\sigma'][\sigma_1/X_1][\sigma_2/X_2]\ldots[\sigma_n/X_n];\sigma.$$

By definition 5.8, and the definition of substitution, this reduces to

$$X_1 \subseteq \sigma_1;\sigma,\ X_2 \subseteq \sigma_2;\sigma,\ldots,X_n \subseteq \sigma_n;\sigma \vdash$$
$$\mu Y[\sigma' \underset{\{X_1,X_2,\ldots,X_n,Y\}}{\circ} \sigma] \subseteq \mu Y[\sigma'[\sigma_1/X_1][\sigma_2/X_2]\ldots[\sigma_n/X_n]];\sigma\ .$$

By the μ-induction rule, it is sufficient to show

$$\left.\begin{array}{l} X_1 \subseteq \sigma_1;\sigma,\ X_2 \subseteq \sigma_2;\sigma,\ldots,X_n \subseteq \sigma_n;\sigma \\ Y \subseteq \mu Y[\sigma'[\sigma_1/X_1][\sigma_2/X_2]\ldots[\sigma_n/X_n]];\sigma \end{array}\right| \vdash$$
$$\sigma' \underset{\{X_1,X_2,\ldots,X_n,Y\}}{\circ} \sigma \subseteq$$
$$\sigma'[\sigma_1/X_1][\sigma_2/X_2]\ldots[\sigma_n/X_n][\mu Y[\sigma'[\sigma_1/X_1][\sigma_2/X_2]\ldots[\sigma_n/X_n]]/Y];\sigma$$

where we have used the f.p.p. of

$$\mu Y[\sigma'[\sigma_1/X_1][\sigma_2/X_2]\ldots[\sigma_n/X_n]].$$

It can be seen that the last assertion holds by the induction hypothesis, since it is of the form

$$X_1 \subseteq \sigma_1;\sigma,\ X_2 \subseteq \sigma_2;\sigma,\ldots,X_n \subseteq \sigma_n;\sigma,\ Y \subseteq \bar{\sigma};\sigma \vdash$$
$$\sigma' \underset{\{X_1,X_2,\ldots,X_n,Y\}}{\circ} \sigma \subseteq \sigma'[\sigma_1/X_1][\sigma_2/X_2]\ldots[\sigma_n/X_n][\bar{\sigma}/Y];\sigma$$

This completes the proof of lemma 5.11.

LEMMA 5.12

Let σ_1, σ_2 be two normal terms. Then

$$\vdash \sigma_1 \underset{\emptyset}{\circ} \sigma_2 = \sigma_1;\sigma_2\ \ .$$

PROOF. Induction on the complexity of σ_1. For the cases where σ_1 is not a procedure, the argument is similar to the proofs of lemmas 5.10 and 5.11.

There remains the proof of

$$\vdash \mu X[\sigma_{11}] \underset{\emptyset}{\circ} \sigma_2 = \mu X[\sigma_{11}];\sigma_2$$

or, by definition 5.8,

$$\vdash \mu X[\sigma_{11} \underset{\{X\}}{\circ} \sigma_2] = \mu X[\sigma_{11}];\sigma_2 \ .$$

a. Proof of $\subseteq$.

By the μ-induction rule, we have to show

$X \subseteq \mu X[\sigma_{11}];\sigma_2 \vdash \sigma_{11} \underset{\{X\}}{\circ} \sigma_2 \subseteq \mu X[\sigma_{11}];\sigma_2$

or

$X \subseteq \mu X[\sigma_{11}];\sigma_2 \vdash \sigma_{11} \underset{\{X\}}{\circ} \sigma_2 \subseteq \sigma_{11}[\mu X[\sigma_{11}]/X];\sigma_2$.

This assertion follows from lemma 5.11, with $n = 1$.

b. Proof of $\supseteq$.

By the μ-induction rule, we have to show

$X;\sigma_2 \subseteq \mu X[\sigma_{11} \underset{\{X\}}{\circ} \sigma_2] \vdash \sigma_{11};\sigma_2 \subseteq \mu X[\sigma_{11} \underset{\{X\}}{\circ} \sigma_2]$.

Since, by the induction hypothesis of lemma 5.12,

$\vdash \sigma_{11};\sigma_2 = \sigma_{11} \underset{\emptyset}{\circ} \sigma_2$,

and since

$\vdash \mu X[\sigma_{11} \underset{\{X\}}{\circ} \sigma_2] = (\sigma_{11} \underset{\{X\}}{\circ} \sigma_2)[\mu X[\sigma_{11} \underset{\{X\}}{\circ} \sigma_2]/X]$,

the assertion follows from lemma 5.10 with $\xi = \emptyset$.

This completes the proof of lemma 5.12.

We have now collected enough results to proceed with the proof of the normal form theorem:

THEOREM (Normal form theorem)

For each regular term τ there exists a normal term σ such that $\vdash \tau = \sigma$.

PROOF. We use induction on the complexity of τ.

a. If τ is a constant or variable, then τ is itself in normal form.

b. $\tau = \tau_1;\tau_2$.
By the induction hypothesis, there exist normal σ_1, σ_2 such that $\vdash \tau_1 = \sigma_1$, $\vdash \tau_2 = \sigma_2$. We show that there exists a normal σ such that $\vdash \sigma_1;\sigma_2 = \sigma$. We use induction on the complexity of σ_1.

b1. $\sigma_1 = \Omega$.
Then $\vdash \sigma_1;\sigma_2 = \Omega$; hence, $\sigma = \Omega$.

b2. $\sigma_1 = E$.
Then $\vdash \sigma_1;\sigma_2 = \sigma_2$; hence, $\sigma = \sigma_2$.

b3. $\sigma_1 = A$.
Then $\sigma_1;\sigma_2 = A;\sigma_2$, which is in normal form, and we take $\sigma = A;\sigma_2$.

b4. $\sigma_1 = A;\sigma_{11}$.
Then $\sigma_1;\sigma_2 = (A;\sigma_{11});\sigma_2 = A;(\sigma_{11};\sigma_2)$. By the induction hypothesis on the complexity of σ_1, there exists σ' such that $\vdash \sigma_{11};\sigma_2 = \sigma'$. Thus, we can take $\sigma = A;\sigma'$.

b5. $\sigma_1 = (p\to\sigma_{11},\sigma_{12})$.
We have $\vdash (p\to\sigma_{11},\sigma_{12});\sigma_2 = (p\to\sigma_{11};\sigma_2,\sigma_{12};\sigma_2)$. By the induction hypothesis, there exist σ',σ'' such that $\vdash \sigma_{11};\sigma_2 = \sigma'$, $\vdash \sigma_{12};\sigma_2 = \sigma''$. By lemma 5.6, there exist σ''',σ'^{V} such that $\vdash (p\to\sigma',\sigma'') = (p\to\sigma''',\sigma'^{V})$, and $(p\to\sigma''',\sigma'^{V})$ is in normal form. Thus, we can take $\sigma = (p\to\sigma''',\sigma'^{V})$.

b6. $\sigma_1 = \mu X[\sigma_{11}]$.
Then $\sigma_1;\sigma_2 = \mu X[\sigma_{11}];\sigma_2$. By lemma 5.12, $\vdash \mu X[\sigma_{11}];\sigma_2 = \mu X[\sigma_{11}] \underset{\emptyset}{o} \sigma_2$; hence, by definition 5.8, $\vdash \mu X[\sigma_{11}];\sigma_2 = \mu X[\sigma_{11} \underset{\{X\}}{o} \sigma_2]$. We may assume that σ_2 does not contain X free and is not a procedure. By lemma 5.9, $\sigma_{11} \underset{\{X\}}{o} \sigma_2$ is regular in X, shields X, and is not a procedure. After, if necessary, applying the conditional axioms, $\sigma_{11} \underset{\{X\}}{o} \sigma_2$ is in normal form. Then $\mu X[\sigma_{11} \underset{\{X\}}{o} \sigma_2]$ is in normal form, and we can take $\sigma = \mu X[\sigma_{11} \underset{\{X\}}{o} \sigma_2]$.

This completes the proof of case b.

c. $\tau = (p \rightarrow \tau_1, \tau_2)$.

By the induction hypothesis, there exist normal σ_1, σ_2 such that $\vdash \tau_1 = \sigma_1$, $\vdash \tau_2 = \sigma_2$. By lemma 5.6, there exist σ', σ'' such that $(p \rightarrow \sigma', \sigma'')$ is a normal term, and that $\vdash (p \rightarrow \sigma_1, \sigma_2) = (p \rightarrow \sigma', \sigma'')$. Thus, $\vdash \tau = (p \rightarrow \sigma', \sigma'')$.

d. $\tau = \mu X[\tau_1]$.

By the induction hypothesis, there exists a normal σ_1 such that $\vdash \tau_1 = \sigma_1$. It can be verified that, if τ_1 is regular in X, then σ_1 is regular in X. By lemma 5.5, we may assume that σ_1 is not a procedure. Application of lemma 5.8 to $\mu X[\sigma_1]$ yields the desired result.

This completes the proof of the normal form theorem.

6. COMPLETENESS THEOREM

6.1. OUTLINE OF THE PROOF

This section is devoted to the proof of the main result of this paper:

THEOREM (Completeness theorem)

Let τ_1, τ_2 be regular terms. Then

$$\models \tau_1 = \tau_2 \iff \vdash \tau_1 = \tau_2 \quad .$$

(Remember that, in general, $\Phi \models \psi$ holds iff $(\Phi \vdash \psi)^I$ is true for all interpretations I (see section 3.2), and that $\Phi \vdash \psi$ holds iff $\Phi \vdash \psi$ is a theorem of the μ-calculus.)

PROOF

1. Proof of $\Longleftarrow$. This follows from the validity of the μ-calculus, as discussed in sections 2 and 3.4.
2. Proof of $\Longrightarrow$. By the normal form theorem, it is sufficient to show that

(1) $$\models \sigma_1 \subseteq \sigma_2 \implies \vdash \sigma_1 \subseteq \sigma_2$$

where σ_1, σ_2 are normal terms.
Note that equivalent terms may well have different normal forms (e.g., $\mu X[(p \to AX, E)]$ and $(p \to A\mu X[(p \to AX, E)], E)$).

The proof of (1) proceeds essentially by an inductive argument on the complexity of σ_1. However, intermediate steps in this proof will be of the more general form

(2) $$\Phi \models \sigma_1 \subseteq \sigma_2 \implies \Phi \vdash \sigma_1 \subseteq \sigma_2$$

where Φ consists of the accumulated hypotheses which are generated by our treatment of procedures, and which will allow us, at suitable stages in the proof, to apply the μ-induction rule.

Let $\phi(\models)$ and $\phi(\vdash)$ be short for the assertions $\Phi \models \sigma_1 \subseteq \sigma_2$ and $\Phi \vdash \sigma_1 \subseteq \sigma_2$ respectively. The general scheme of the proof is the following:

First we give a precise description of the structure of the assertions $\phi(\models)$ (called *normal assertions*), which structure is determined by the way in which the successive hypotheses concerning procedures are generated.

Secondly, we shall introduce a complexity measure Γ for an assertion $\phi(\models)$. This measure involves the complexity of both σ_1 and Φ in $\phi(\models)$. Then, in order to prove $\phi(\models) \Longrightarrow \phi(\vdash)$ for normal assertions, we shall proceed by a case analysis of the various forms which σ_1 and σ_2 in $\phi(\models)$ can have, as determined by the normal form theorem, and we shall show that in each case, $\phi(\models) \Longrightarrow \phi(\vdash)$ can be seen to hold on the basis of one of three arguments (A_1, A_2 or A_3):

A_1 : $\phi(\models)$ is false, i.e., there exists at least one interpretation I for which $(\Phi \models \sigma_1 \subseteq \sigma_2)^I$ does not hold.
Example: $\Phi \models E \subseteq \Omega$.

A_2 : $\phi(\vdash)$ can be seen to be a theorem of the μ-calculus directly.
Example: $\Phi \vdash A \subseteq A$.

A_3 : Assume that $\phi(\models)$ holds. We exhibit $\phi^{(i)}(\models)$, $1 \leq i \leq n$, such that the following conditions are satisfied:

$A_{3.1}$: $\phi^{(i)}(\models)$ is a normal assertion, $1 \leq i \leq n$.

$A_{3.2}$: $\phi^{(i)}(\models)$ has less complexity than $\phi(\models)$, $1 \leq i \leq n$.

$A_{3.3}$: $\phi(\models) \Longrightarrow \phi^{(i)}(\models)$, $1 \leq i \leq n$.

$A_{3.4}$: From $\phi^{(1)}(\vdash)$, $\phi^{(2)}(\vdash)$,...,$\phi^{(n)}(\vdash)$ together, we can infer $\phi(\vdash)$.

The following picture illustrates argument A_3:

$$\begin{tikzcd}
\phi(\models) \arrow[r, Rightarrow, "1"] \arrow[d, Rightarrow, "2_i"'] & \phi(\vdash) \\
\left\{\begin{matrix} \phi^{(1)}(\models) \\ \vdots \\ \phi^{(n)}(\models) \end{matrix}\right. \arrow[r, Rightarrow, "3_i"] & \left\{\begin{matrix} \phi^{(1)}(\vdash) \\ \vdots \\ \phi^{(n)}(\vdash) \end{matrix}\right. \arrow[u, Rightarrow, "4"']
\end{tikzcd}$$

In order to show 1, we prove 2_i (for each i), and 4, and, assuming that 3_i holds (this is the induction assumption, since, for each i, $\phi^{(i)}(\models)$ has less complexity than $\phi(\models)$), we then have the desired result.
Example of A_3: $\phi(\models)$ is $\Phi \models A;\sigma_1 \subseteq A;\sigma_2$, $n = 1$, and $\phi^{(1)}(\models)$ is $\Phi \models \sigma_1 \subseteq \sigma_2$. Details of this example will be given below, in particular in lemma 6.3.

6.2. NORMAL ASSERTIONS

We now give the definition of a *normal assertion*. This definition will become clearer if it is considered together with the case analysis to be given below, in particular with the treatment of procedures (cases 6, 7 and 8). Moreover, the definition is followed by an informal explanation.

DEFINITION 6.1

An assertion $\Phi \models \sigma_1 \subseteq \sigma_2$ is a normal assertion iff

1. σ_1 and σ_2 are normal terms.
2. $\Phi = \Phi_1,\Phi_2,\ldots,\Phi_m$, $m \geq 0$. We use the convention that, if $m = 0$, then Φ is the empty list. If $m \geq 1$, then we require, for each i, $1 \leq i \leq m$,
   a. $\Phi_i = X_i \subseteq \rho_{i,0}, X_i \subseteq \rho_{i,1},\ldots,X_i \subseteq \rho_{i,n_i}$, $n_i \geq 0$.
   b. Each $\rho_{i,j}$, $0 \leq j \leq n_i$, is a normal term; $\rho_{i,0}$ is a normal term of the form $\mu X_i[\rho_i]$.
   c. σ_1 is regular in X_i and shields X_i.
   d. ρ_i is regular in X_j, $1 \leq j \leq m$, and shields X_j, $1 \leq j \leq m$.
   e. X_i does not occur free in σ_2 or in any $\rho_{j,k}$, $1 \leq j < i$, $0 \leq k \leq n_j$.
   f. If $j \neq k$, then $\rho_{i,j} \neq \rho_{i,k}$, $1 \leq j,k \leq n_i$.
   g. σ_1 and $\rho_{i,0}$ are subterms of $\rho_{1,0}$;
      σ_2 and $\rho_{i,j}$, $1 \leq j \leq n_i$, are subterms of $\rho_{1,1}$.
   h. $\Phi_1,\Phi_2,\ldots,\Phi_m \models \rho_i \subseteq \rho_{i,j}$, $0 \leq j \leq n_i$.

We give an intuitive exposition of the various points taken into account in this definition. Firstly, we distinguish two aspects in the conditions imposed upon the normal assertions, viz., those related to the application of the μ-induction rule, and those related to the termination of the inductive argument described via arguments A_1, A_2 and A_3 above.

Conditions 2f and 2g belong to the second category, the remaining ones to the first.

The conditions of definition 6.1 will be discussed by means of a simple example. In our treatment of this example, we shall not be complete - the full argument follows in the case analysis below - but we give only a first sketch.

Consider the following assertion:

$$(3)\qquad \models \underbrace{\mu X_1[\underbrace{\ldots\mu X_2[\underbrace{\ldots X_1\ldots X_2\ldots}_{\rho_2}]\ldots}_{\rho_1}]}_{\sigma_1} \subseteq \sigma_2$$

First we reduce this (case 6 below) to

$$(4)\qquad X_1 \subseteq \mu X_1[\rho_1] \models X_1 \subseteq \sigma_2 \text{ .}$$

(In the complete proof of this and the following reduction steps, argument A_3 has to be verified.)

(4) is in turn reduced to (case 7 below):

$$(5)\qquad \underbrace{X_1 \subseteq \mu X_1[\rho_1], X_1 \subseteq \sigma_2}_{\Phi_1} \models \rho_1 \subseteq \sigma_2$$

Note that Φ_1 has the form prescribed by conditions 2a and 2b of definition 6.1, with $\rho_{1,0} = \mu X_1[\rho_1]$, $\rho_{1,1} = \sigma_2$, $m = 1$, and $n_1 = 1$. The continuation of the process now depends on the particular form of ρ_1. In general, a successive reduction takes place; e.g., if ρ_1 and σ_2 are of the form $A;\rho_1'$ and $A;\sigma_2'$ respectively, we reduce (5) to

$$\Phi_1 \models \rho_1' \subseteq \sigma_2' \text{ ,}$$

(cf. lemma 6.3 and case 4 below; note that ρ_1' and σ_2' are subterms of ρ_1 and σ_2 respectively), and if ρ_1 is of the form $(p\rightarrow\rho_1',\rho_1'')$ we reduce (5) to both

$$\Phi_1 \models \rho_1' \leq \sigma_2'$$

and

$$\Phi_1 \models \rho_1'' \leq \sigma_2''$$

where σ_2' and σ_2'' are subterms of σ_2, such that $(p\to\sigma_2',\sigma_2'')$ is in normal form and that $\models (p\to\sigma,\sigma) = (p\to\sigma_2',\sigma_2'')$ holds. (Cf. lemma 6.5 and case 5 below.) Eventually, these and similar reductions will at some stage have reduced (5) to, a.o.,

(6) $$X_1 \leq \mu X_1[\rho_1], X_1 \leq \sigma_2 \models \mu X_2[\rho_2] \leq \sigma_2^{(1)}$$

where $\sigma_2^{(1)}$ is a subterm of σ_2.
We are now in a situation where case 6 and then case 7 are again applicable, and we derive

(7) $$X_1 \leq \mu X_1[\rho_1], X_1 \leq \sigma_2, X_2 \leq \mu X_2[\rho_2], X_2 \leq \sigma_2^{(1)} \models \rho_2 \leq \sigma_2^{(1)}.$$

Successive reduction of ρ_2 will at some stage lead to the assertion

(8) $$X_1 \leq \mu X_1[\rho_1], X_1 \leq \sigma_2, X_2 \leq \mu X_2[\rho_2], X_2 \leq \sigma_2^{(1)} \models X_1 \leq \sigma_2^{(2)}$$

where $\sigma_2^{(2)}$ is a subterm of $\sigma_2^{(1)}$ (and, thus, of $\sigma_2 = \rho_{1,1}$).
Now there are two possibilities:
(α) $\sigma_2^{(2)} = \sigma_2$.
Then, clearly,

$$X_1 \leq \mu X_1[\rho_1], X_1 \leq \sigma_2, X_2 \leq \mu X_2[\rho_2], X_2 \leq \sigma_2^{(1)} \vdash X_1 \leq \sigma_2^{(2)}$$

holds, and we have reached an end point in our inductive argument
(β) $\sigma_2^{(2)} \not\equiv \sigma_2$.
Then we continue the process with another application of case 7, and we reduce (8) to

(9) $$X_1 \subseteq \mu X_1[\rho_1], X_1 \subseteq \sigma_2, X_1 \subseteq \sigma_2^{(2)}, X_2 \subseteq \mu X_2[\rho_2], X_2 \subseteq \sigma_2^{(1)}$$
$$\models \rho_1 \subseteq \sigma_2^{(2)} \ .$$

Successive reduction of ρ_1 will then lead to, a.o.,

(10) $$X_1 \subseteq \mu X_1[\rho_1], X_1 \subseteq \sigma_2, X_1 \subseteq \sigma_2^{(2)}, X_2 \subseteq \mu X_2[\rho_2], X_2 \subseteq \sigma_2^{(1)}$$
$$\models \mu X_2[\rho_2] \subseteq \sigma_2^{(3)}$$

with $\sigma_2^{(3)}$ a subterm of $\sigma_2^{(2)}$. Now <u>case 8</u> applies: We encounter a procedure $(\mu X_2[\rho_2])$ which has already appeared before in the process (as can be seen from the presence of $X_2 \subseteq \mu X_2[\rho_2]$ in the list of assumptions). We then reduce (10) to

(11) $$\underbrace{X_1 \subseteq \mu X_1[\rho_1], X_1 \subseteq \sigma_2, X_1 \subseteq \sigma_2^{(2)}}_{\Phi_1}, \underbrace{X_2 \subseteq \mu X_2[\rho_2], X_2 \subseteq \sigma_2^{(1)}}_{\Phi_2} \models \rho_2 \subseteq \sigma_2^{(3)}$$

etc.

We now use this example to discuss the conditions of definition 6.1. It is easily seen that the assumptions $\Phi_1, \Phi_2, \ldots$ which are successively generated are indeed of the form prescribed by conditions 2a and 2b. It is also of importance to note that, in general, the number of Φ_i which will be generated is bounded by

a. For the assertion (3), the maximum number of "nested" procedures in σ_1;

b. For the assertions (4), (5),...,(11), the maximum number of "nested" procedures in $\rho_{1,0}$.

This maximum number is certainly bounded by the total number of subterms in σ_1 and $\rho_{1,0}$ respectively. This fact, which will be applied below in definition 6.2, case c (definition of $S_1(\phi)$) is of importance in the termination argument. See for this the comments on the use of the complexity measure Γ of a normal assertion in section 6.3.

Next, we consider clauses 2c and 2d of definition 6.1. These conditions are vacuously satisfied for assertion (3), and can easily be seen to hold for the remaining assertions. E.g., in (5), ρ_1 is regular in X_1 and shields X_1, since $\mu X_1[\rho_1]$ is in normal form.

As to condition 2e, by suitable rewriting of bound variables, we may assume that none of the bound variables of σ_1 in (3) occurs free in σ_2. Since all generated $\rho_{j,k}$, for $k \geq 1$, are subterms of the initial σ_2, these two facts imply that none of the X_i occurs free in any of the $\rho_{j,k}$, for $k \geq 1$. The absence of free occurrences of X_i in $\rho_{j,0}$, $1 \leq j < i$, is illustrated in our example by the absence of free occurrences of X_2 in $\mu X_1[\rho_1]$ (in assertions (7),...,(11)). Again, this can be achieved in the general case by a suitable rewrite.

For condition 2f compare the treatment of assertion (8): No addition of a new assumption to some Φ_i takes place, if this assumption is identical to some assumption already contained in the list Φ_i. Therefore, we know that, at each stage, all $\rho_{i,j}$ contained in Φ_i are different.

Next, we discuss condition 2g. Consider e.g. assertion (11). From the construction which led from (3) to (11) we see that

a. ρ_2 is a subterm of $\mu X_2[\rho_2]$ which is a subterm of ρ_1 which is a subterm of $\mu X_1[\rho_1] = \rho_{1,0}$.

b. $\sigma_2^{(3)}$ is a subterm of $\sigma_2^{(2)}$ which is a subterm of $\sigma_2^{(1)}$ which is a subterm of $\rho_2 = \rho_{1,1}$.

From these two facts we conclude that condition 2g is satisfied.

Combination of conditions 2f and 2g implies the following:
At each stage, we know an upper bound for the number of elements in each of the lists Φ_i: This is given by "1 + the number of subterms in $\rho_{1,1}$", or, if $\rho_{1,1}$ is not yet present (cf. assertions (3) or (4)), by "1 + the number of subterms in σ_2''. This fact is used in definition 6.2, clause c (definition of $S_r(\phi)$) and in the termination argument (cf. section 6.3).

Finally, we consider condition 2h. Applied e.g. to the assertion (11), this condition states that the following assertions all hold:

$$\Phi_1 \models \rho_1 \subseteq \rho_{1,0},\ \Phi_1 \models \rho_1 \subseteq \rho_{1,1},\ \Phi_1 \models \rho_1 \subseteq \rho_{1,2}$$

and

$$\Phi_1, \Phi_2 \models \rho_2 \subseteq \rho_{2,0}, \quad \Phi_1, \Phi_2 \models \rho_2 \subseteq \rho_{2,1}$$

or, specifically,

(12) $\Phi_1 \models \rho_1 \subseteq \mu X_1[\rho_1]$

(13) $\Phi_1 \models \rho_1 \subseteq \sigma_2$

(14) $\Phi_1 \models \rho_1 \subseteq \sigma_2^{(2)}$

(15) $\Phi_1, \Phi_2 \models \rho_2 \subseteq \mu X_2[\rho_2]$

(16) $\Phi_1, \Phi_2 \models \rho_2 \subseteq \sigma_2^{(1)}$.

That assertions (12) to (16) hold, can be seen from the "history" of the derivation of (11): (12) is clear from the fixed point property and monotonicity (since Φ_1 contains the assumption $X_1 \subseteq \mu X_1[\rho_1]$, we have $\rho_1 \subseteq \rho_1[\mu X_1[\rho_1]/X_1]$; hence, $\rho_1 \subseteq \mu X_1[\rho_1]$),(13) is implied by (5), (14) by (9), (15) follows also by the f.p.p. and monotonicity, and (16) follows from (7).

This example illustrates the function of condition 2h.
For each given assertion which occurs at some stage in the inductive argument (by repeated use of A_3), condition 2h states the validity of a number of assertions which precede the given assertion in the inductive process. As will be seen in the precise treatment of case 6, case 7 and case 8 below, we must have the validity of these preceding assertions available, in order to be able to verify, at suitable moments, arguments $A_{3.3}$ and $A_{3.4}$.

This transfer of information on the history of the derivation of an assertion to a property of the assumptions of the assertion, by means of condition 2h, allows us to do without the complications of the (implicity present) tree-like structure of the inductive argument.

6.3. COMPLEXITY OF NORMAL ASSERTIONS

The next definition introduces some notation for a normal assertion, which will be used in the discussion of its complexity.

DEFINITION 6.2

Let $\phi(\models) = \Phi \models \sigma_1 \subseteq \sigma_2$ be a normal assertion, as described in definition 6.1. We define

a. $X(\phi) = \{X_1, X_2, \ldots, X_m\}$

b. $L_a(\phi) = \sum_{i=1}^{m} (n_i+1)$

c. $S_l(\phi) = |\Sigma(\rho_{1,0})|$, if $m \geq 1$

$\qquad\quad = |\Sigma(\sigma_1)|$, if $m = 0$

$S_r(\phi) = |\Sigma(\rho_{1,1})|$, if $m \geq 1$ and $n_1 \geq 1$

$\qquad\quad = |\Sigma(\sigma_2)|$, otherwise

Here, $|\Sigma(\sigma)|$ is used to denote the number of elements in the set $\Sigma(\sigma)$ of all subterms of σ. See definition 5.7.

d. $L_{max}(\phi) = S_l(\phi)(1+S_r(\phi))$

e. $L_g(\phi) = L_{max}(\phi) - L_a(\phi)$

We now give the motivation for the definition of the complexity $\Gamma(\phi)$ of a normal assertion $\phi(\models) = \Phi \models \sigma_1 \subseteq \sigma_2$, which follows in definition 6.4 below.

Consider a sequence of reduction steps as illustrated in the example above. In general, we have, starting with the normal assertion $\phi(\models) = \phi^{(1)}(\models)$:

$$\begin{aligned}
\phi^{(1)}(\models) &= \Phi^{(1)} \models \sigma_1^{(1)} \subseteq \sigma_2^{(1)} \\
\phi^{(2)}(\models) &= \Phi^{(2)} \models \sigma_1^{(2)} \subseteq \sigma_2^{(2)} \\
&\vdots \\
\phi^{(i)}(\models) &= \Phi^{(i)} \models \sigma_1^{(i)} \subseteq \sigma_2^{(i)} \\
&\vdots
\end{aligned}$$

where each $\phi^{(i)}(\models)$ is reduced to $\phi^{(i+1)}(\models)$ as part of argument A_3. We need a complexity measure which decreases in each transition from $\phi^{(i)}(\models)$ to $\phi^{(i+1)}(\models)$. Clearly, it is not sufficient to use as such a measure simply the complexity of $\sigma_1^{(i)}$. Consider as a counter example assertions (4) and (5) above, where the complexity of $\sigma_1^{(i)} = X$, increases to that of $\sigma_1^{(i+1)} = \rho_1$. Therefore, another entity must be taken into account in order to make the inductive argument go through. For this, we use the function $L_g(\phi)$, as defined in clause e of definition 6.2, by $L_g(\phi) = L_{max}(\phi) - L_a(\phi)$. The idea behind its introduction is the following: We compare the number of elements in the lists $\Phi^{(i)}$ and $\Phi^{(i+1)}$. In each reduction step, this number either remains the same, or it increases. However, this increase cannot go on indefinitely: At each stage, it is possible to predict the maximum future growth of the number of elements in $\Phi^{(i)}$, and this is the number given by $L_g(\phi^{(i)})$, namely, as the difference between the actual length $L_a(\phi^{(i)})$ -cf. definition 6.2, clause b- and the upper bound for the number of elements in $\Phi^{(j)}$ which can occur in future steps, which upper bound is given by $L_{max}(\phi^{(i)}) = S_l(\phi^{(i)})(1 + S_r(\phi^{(i)}))$. The upper bound $L_{max}(\phi^{(i)})$ is determined as follows.

We note that Φ may grow in two ways:

a. Some Φ_i in Φ may be extended by the addition of another assumption $X_i \subseteq \rho_{i,n_i+1}$. Condition 2f of definition 6.1 ensures that in this newly added hypothesis, ρ_{i,n_i+1} is different from all previously added $\rho_{i,j}$, $1 \leq j \leq n_i$. (Cf. the reduction from (8) to (9) above.) Since, by condition 2g of definition 6.1, all $\rho_{i,j}$ are subterms of $\rho_{1,1}$, it follows that the number of elements in Φ_i is bounded by $1 + S_r(\phi)$. A proviso has to be made for the case that $\rho_{1,1}$ is not yet present. Then $\phi(\models)$ is of the form $\Phi \models \sigma_1 \subseteq \sigma_2$, with Φ empty, or $\Phi = X_1 \subseteq \rho_{1,0}$. (In the example above, this holds for assertions (3) and (4).) From the way in which assumptions are added to Φ -case 7 above- it will follow that in the possibly added assumption $X_1 \subseteq \rho_{1,1}$, $\rho_{1,1}$ will be either σ_2 itself, or a subterm of σ_2. This explains the second alternative in the definition of $S_r(\phi)$.

b. A new Φ_{m+1} may be added to $\Phi = \Phi_1, \Phi_2, \ldots, \Phi_m$. An upper bound for the number of times such an addition can occur, is given by the total number of subterms in $\rho_{1,0}$, or, in case Φ is empty, in σ_1.

Now we assert -and we shall have to verify this at each stage in the case analysis below- that, if according to argument A_3, the proof of $\phi^{(i)}(\models) \Longrightarrow \phi^{(i)}(\vdash)$ is reduced, a.o., to that of $\phi^{(i+1)}(\models) \Longrightarrow \phi^{(i+1)}(\vdash)$, then $L_g(\phi^{(i+1)}) \leq L_g(\phi^{(i)})$. In order to verify this, we must show in each case:

(α) $L_a(\phi^{(i+1)}) \geq L_a(\phi^{(i)})$

(β) $L_{max}(\phi^{(i+1)}) \leq L_{max}(\phi^{(i)})$.

After these explanations, the definition of the complexity $\Gamma(\phi)$ will offer no difficulties. First we give the definition of the complexity $\gamma(\sigma)$ of a normal term σ. This is the same notion we used already in section 5, but which is defined here formally for completeness sake.

DEFINITION 6.3

Let σ be a normal term.

a. If σ is a constant or variable, then $\gamma(\sigma) = 1$.

b. $\gamma(A;\sigma) = 1 + \gamma(\sigma)$.

c. $\gamma((p \to \sigma_1, \sigma_2)) = 1 + \gamma(\sigma_1) + \gamma(\sigma_2)$.

d. $\gamma(\mu X[\sigma]) = 1 + \gamma(\sigma)$.

DEFINITION 6.4

Let $\phi(\models) = \Phi \models \sigma_1 \subseteq \sigma_2$ be a normal assertion.

$\Gamma(\phi)$ is defined as a pair:

$$\Gamma(\phi) = (L_g(\phi), \gamma(\sigma_1)).$$

The ordering between these pairs is the lexicographical one:

Let:

$$\phi^{(1)}(\models) = \Phi^{(1)} \models \sigma_1^{(1)} \subseteq \sigma_2^{(1)}$$
$$\phi^{(2)}(\models) = \Phi^{(2)} \models \sigma_1^{(2)} \subseteq \sigma_2^{(2)} .$$

Then $\Gamma(\phi^{(1)}) < \Gamma(\phi^{(2)})$ iff either

(α) $L_g(\phi^{(1)}) < L_g(\phi^{(2)})$, or

(β) $L_g(\phi^{(1)}) = L_g(\phi^{(2)})$, and $\gamma(\sigma_1^{(1)}) < \gamma(\sigma_1^{(2)})$.

6.4. AUXILIARY LEMMAS

Before we proceed with the case analysis in section 6.5, we first collect in a number of lemmas some results to be used below.

LEMMA 6.1

a. $\Phi \vdash \Omega \subseteq \sigma$

b. $\Phi \vdash E \subseteq E$

c. $\Phi \vdash A \subseteq A$

d. $\Phi, X \subseteq \sigma \vdash X \subseteq \sigma$

PROOF. Clear.

LEMMA 6.2

None of the following normal assertions $\phi(\models)$ holds:

a. $\Phi \models E \subseteq \Omega$

b. $\Phi \models E \subseteq A$

c. $\Phi \models E \subseteq A;\sigma$

d. $\Phi \models E \subseteq (p\to\sigma_1,\sigma_2)$.

PROOF. a is clear. As to b, c and d, select some interpretation I with domain $\mathcal{D}$, and $x \in \mathcal{D}$, such that $A^I(x)$, $A^I(x)$, and $p^I(x)$ respectively are undefined, and, if necessary, X_i^I, for $X_i \in X(\phi)$, (cf. def.6.2) is undefined on $\mathcal{D}$. Since $E^I(x) = x$, the lemma follows.

LEMMA 6.3 [1]

Let $\phi(\models) = \Phi \models A;\sigma_{11} \subseteq \sigma_2$ be a normal assertion. Let σ_2 be neither a

[1] The need for this lemma, and its proof, were pointed out to us by Scott.

variable nor a procedure. If σ_2 has the form $\sigma_2 = A;\sigma_{21}$, define $\phi^{(1)}(\models)$ as
$\phi^{(1)}(\models) = \Phi \models \sigma_{11} \subseteq \sigma_{12}$. Otherwise, define
$\phi^{(1)}(\models) = \Phi \models \sigma_{11} \subseteq \Omega$.
Then $\phi(\models) \Longrightarrow \phi^{(1)}(\models)$ holds.

PROOF.

a. $\sigma_2 = A;\sigma_{21}$.

We have to show:

$$\Phi \models A;\sigma_{11} \subseteq A;\sigma_{21} \quad \Longrightarrow \quad \Phi \models \sigma_{11} \subseteq \sigma_{21}.$$

Suppose $\Phi \models \sigma_{11} \subseteq \sigma_{21}$ does not hold. Then there exists an interpretation I_0 with domain $\mathcal{D}_0$, satisfying Φ, and some $x_0 \in \mathcal{D}_0$, such that $\sigma_{11}^{I_0}(x_0) = y$, and $\sigma_{21}^{I_0}(x_0) \neq y$.
Let I_1 be the following interpretation: $\mathcal{D}_1 = \mathcal{D}_0 \cup \{x_1\}$, for some $x_1 \notin \mathcal{D}_0$. On $\mathcal{D}_0$, I_1 is defined to coincide with I_0. In x_1, we define $A^{I_1}(x_1) = x_0$. Note that, since $A;\sigma_{11}$ is regular in each $X_i \in X(\phi)$ (by clause 2c of definition 6.1), $A \notin X(\phi)$; hence, we are not restricted in our choice for A^{I_1}. We then have: $(A;\sigma_{11})^{I_1}(x_1) = \sigma_{11}^{I_1}(A^{I_1}(x_1)) =$
$= \sigma_{11}^{I_1}(x_0)$. Since $x_0 \in \mathcal{D}_0$, $\sigma_{11}^{I_1}(x_0) = \sigma_{11}^{I_0}(x_0) = y$. Also,
$(A;\sigma_{21})^{I_1}(x_1) = \sigma_{21}^{I_1}(A^{I_1}(x_1)) = \sigma_{21}^{I_1}(x_0) = \sigma_{21}^{I_0}(x_0) \neq y$. Thus, I_1 contradicts the validity of $\Phi \models A;\sigma_{11} \subseteq A;\sigma_{21}$.

b. σ_2 cannot be written as $A;\sigma_{21}$.

We have four possibilities for σ_2: Ω, E, $A';\sigma_2'$, or $(p\to\sigma',\sigma'')$. We give the proof of the third case, the other ones being similar. Assume $\Phi \models A;\sigma_{11} \subseteq A';\sigma_2'$, and suppose that $\Phi \models \sigma_{11} \subseteq \Omega$ does not hold. Then there exists I_0 and $\mathcal{D}_0$, and $x_0 \in \mathcal{D}_0$, such that $\sigma_{11}^{I_0}(x_0) = y$. Let $\mathcal{D}_1 = \mathcal{D}_0 \cup \{x_1\}$, for some $x_1 \notin \mathcal{D}_0$, let $A^{I_1}(x_1) = x_0$, let $A'^{I_1}(x_1)$ be undefined (by clause 2e of definition 6.1, the choice for A'^{I_1} is not restricted by Φ), and let I_1 be as I_0 on $\mathcal{D}_0$. This I_1 contradicts the validity of $\Phi \models A;\sigma_{11} \subseteq A';\sigma_2'$.

LEMMA 6.4

Let σ_1, σ_2 be normal terms, both of which are free from p. Then the following holds:

For all interpretations I_0 with domain $\mathcal{D}_0$, and all $x_0 \in \mathcal{D}_0$, if $\sigma_1^{I_0} x_0) = y$, then there exists an interpretation I_1 with domain $\mathcal{D}_1$ and $x_1 \in \mathcal{D}_1$ such that

a. $\sigma_1^{I_1}(x_1) = y$.

b. $p^{I_1}(x_1) = 1$.

c. For all $q \neq p$, $q^{I_1}(x_1) = q^{I_0}(x_0)$, if $q^{I_0}(x_0)$ is defined,

$q^{I_1}(x_1)$ is undefined, if $q^{I_0}(x_0)$ is undefined.

d. If $\sigma_2^{I_1}(x_1) = y$, then $\sigma_2^{I_0}(x_0) = y$.

Moreover, there also exists an interpretation I_2, which satisfies a, c, d (with I_1 replaced by I_2), and b': $p^{I_2}(x_1) = 0$.

The main statement of this lemma can be phrased as: If σ is free from p, then for each x_0 and I_0 we can find x_1 and I_1 such that $\sigma^{I_1}(x_1)$ delivers the same result as $\sigma^{I_0}(x_0)$, but, moreover, we are free to choose for $p^{I_1}(x_1)$ either 1 or 0, without influencing the value of $\sigma^{I_1}(x_1)$.

PROOF. We prove only the existence of I_1; the proof for I_2 follows by symmetry. We use induction on the complexity of the pair (σ_1, σ_2). The full proof will not be given, but only three representative cases.

1. $\sigma_1 = A;\sigma_{11}$, $\sigma_2 = A';\sigma_{21}$.
Choose I_1 and $\mathcal{D}_1$ as follows: $\mathcal{D}_1 = \mathcal{D}_0 \cup \{x_1\}$, for some $x_1 \notin \mathcal{D}_0$. I_1 coincides with I_0 on $\mathcal{D}_0$, $A^{I_1}(x_1) = A^{I_0}(x_0)$, $A'^{I_1}(x_1) = A'^{I_0}(x_0)$, $p^{I_1}(x_1) = 1$, and for all $q \neq p$, $q^{I_1}(x_1) = q^{I_0}(x_0)$ (or both are undefined). Assume $(A;\sigma_{11})^{I_0}(x_0) = y$. We verify a, b, c and d. We have

$(A;\sigma_{11})^{I_1}(x_1) = \sigma_{11}^{I_1}(A^{I_1}(x_1)) = \sigma_{11}^{I_1}(A^{I_0}(x_0)) = \sigma_{11}^{I_0}(A^{I_0}(x_0)) =$ $(A;\sigma_{11})^{I_0}(x_0) = y$, where we have used the fact that I_1 coincides I_0 on $\mathcal{D}_0$. b and c are clear. As to d, if $(A';\sigma_{21})^{I_1}(x_1) = y$, then $(A';\sigma_{21})^{I_0}(x_0) = \sigma_{21}^{I_0}(A'^{I_0}(x_0)) = \sigma_{21}^{I_1}(A'^{I_1}(x_1)) = y$.

2. $\sigma_2 = (q\to\sigma_{21},\sigma_{22})$.

Let $\sigma_1^{I_0}(x_0) = y$. Two cases arise:

2.1. $q^{I_0}(x_0)$ is defined, say $q^{I_0}(x_0) = 1$. We apply the induction hypothesis to the pair (σ_1,σ_{21}), both of which are free from p. Thus, there exists I_1 and $x_1 \in \mathcal{D}_1$ such that $\sigma_1^{I_1}(x_1) = y$, $p^{I_1}(x_1) = 1$, $r^{I_1}(x_1) = r^{I_0}(x_0)$, for all $r \neq p$ (or both are undefined), in particular, $q^{I_1}(x_1) = q^{I_0}(x_0) = 1$, and, if $\sigma_{21}^{I_1}(x_1) = y$, then $\sigma_{21}^{I_0}(x_0) = y$. This proves clauses a, b and c. As to clause d, suppose that $\sigma_2^{I_1}(x_1) = y$. Then $(q\to\sigma_{21},\sigma_{22})^{I_1}(x_1) = (q^{I_1}(x_1) \to \sigma_{21}^{I_1}(x_1), \sigma_{22}^{I_1}(x_1)) =$ $\sigma_{21}^{I_1}(x_1) = y$. Hence, $\sigma_{21}^{I_0}(x_0) = y$, by the induction hypothesis. From this, $\sigma_2^{I_0}(x_0) = (q\to\sigma_{21},\sigma_{22})^{I_0}(x_0) = y$ follows.

2.2. $q^{I_0}(x_0)$ is undefined. Apply the induction hypothesis to the pair (σ_1,σ_{21}). The argument is the same as for 2.1, apart from the fact that $q^{I_1}(x_1)$ is now undefined, which implies that clause d is trivially satisfied.

3. $\sigma_1 = (q\to\sigma_{11},\sigma_{12})$.

Assume $\sigma_1^{I_0}(x_0) = y$. Then either $q^{I_0}(x_0) = 1$ or $q^{I_0}(x_0) = 0$. Assume the first. Apply the induction hypothesis to the pair (σ_{11},σ_2), both of which are free from p. This yields an I_1 and $x_1 \in \mathcal{D}_1$, such that $\sigma_{11}^{I_1}(x_1) = y$, $p^{I_1}(x_1) = 1$, $r^{I_1}(x_1) = r^{I_0}(x_0)$ (or both are undefined), and, if $\sigma_2^{I_1}(x_1) = y$, then $\sigma_2^{I_0}(x_0) = y$. Then $(q\to\sigma_{11},\sigma_{12})^{I_1}(x_1) = y$;

hence, clause a. Clauses b, c and d are clear.

LEMMA 6.5

Let $\phi(\models) = \Phi \models (p\to\sigma_1,\sigma_2) \subseteq \sigma$ be a normal assertion. There exist σ',σ'' such that

a. σ',σ'' are subterms of σ, and $(p\to\sigma',\sigma'')$ is a normal term.

b. $\models (p\to\sigma',\sigma'') = (p\to\sigma,\sigma)$.

c. Let $\phi^{(1)}(\models) = \Phi \models \sigma_1 \subseteq \sigma'$

$\phi^{(2)}(\models) = \Phi \models \sigma_2 \subseteq \sigma''$

Then $\phi(\models) \Longrightarrow \phi^{(1)}(\models)$

$\phi(\models) \Longrightarrow \phi^{(2)}(\models)$.

PROOF. Assume that $\Phi \models (p\to\sigma_1,\sigma_2) \subseteq \sigma$ holds. This implies $\Phi \models (p\to(p\to\sigma_1,\sigma_2),(p\to\sigma_1,\sigma_2)) \subseteq (p\to\sigma,\sigma)$; hence, we have $\Phi \models (p\to\sigma_1,\sigma_2) \subseteq (p\to\sigma,\sigma)$. By lemma 5.6, there exists a normal term $(p\to\sigma',\sigma'')$ such that $\models (p\to\sigma',\sigma'') = (p\to\sigma,\sigma)$, with σ', σ'' subterms of σ. Hence,

$$(17) \qquad \Phi \models (p\to\sigma_{11},\sigma_{12}) \subseteq (p\to\sigma',\sigma'') \ .$$

Assume that, e.g., $\Phi \models \sigma_{11} \subseteq \sigma'$ does not hold. Then there exists I_0 and $x_0 \in \mathcal{D}_0$ such that I_0 satisfies Φ, and such that $\sigma_{11}^{I_0}(x_0) = y$, $\sigma'^{I_0}(x_0) \neq y$. By lemma 6.4, there exists I_1 and $x_1 \in \mathcal{D}_1$ such that $p^{I_1}(x_1) = 1$, and $\sigma_{11}^{I_1}(x_1) = y$. Since $(p\to\sigma_{11},\sigma_{12})$ is regular in all $X_i \in X(\phi)$ and shields all $X_i \in X(\phi)$, it can be verified that I_1 also satisfies Φ. Thus, $(p\to\sigma_{11},\sigma_{12})^{I_1}(x_1) = \sigma_{11}^{I_1}(x_1) = y$. Hence, by (17), $(p\to\sigma',\sigma'')^{I_1}(x_1) = \sigma'^{I_1}(x_1) = y$. Then, by lemma 6.4, $\sigma'^{I_0}(x_0) = y$. Contradiction.

6.5. CASE ANALYSIS

After the preparations of the preceding sections, we are in a position to give the proof of the completeness theorem: If $\phi(\models) = \Phi \models \sigma_1 \subseteq \sigma_2$ is a normal assertion, then

(18) $$\Phi \models \sigma_1 \subseteq \sigma_2 \Longrightarrow \Phi \vdash \sigma_1 \subseteq \sigma_2 \quad .$$

By lemma 5.5, we may assume that σ_2 is not a procedure. Since $\models A = A;E$, we may also assume that σ_2 is not a variable. Moreover, after suitable rewriting of bound variables, we may assume that in no term are there two occurrences of terms $\mu X[\sigma_1]$ and $\mu X[\sigma_2]$ with the same bound variable X. (E.g., a term $\mu X[\ldots\mu X[\sigma']\ldots\mu X[\sigma'']\ldots]$ is rewritten as $\mu X[\ldots\mu Y[\sigma'[Y/X]]\ldots\mu Z[\sigma''[Z/X]]\ldots]$.)

CASE 1. $\sigma_1 = \Omega$.
We have to show:

$$\Phi \models \Omega \subseteq \sigma \quad \Longrightarrow \quad \Phi \vdash \Omega \subseteq \sigma .$$

By lemma 6.1, argument $A2$ applies.

CASE 2. $\sigma_1 = E$.
2.1. $\sigma_2 = \Omega$. Follows by lemma 6.2 and argument A_1.
2.2. $\sigma_2 = E$. Follows by lemma 6.1 and argument A_2.
2.3 $\sigma_2 = A;\sigma_{21}$. Follows by lemma 6.2 and argument A_1.
2.4. $\sigma_2 = (p \rightarrow \sigma_{21}, \sigma_{22})$. Follows by lemma 6.2 and argument A_1.

CASE 3. $\sigma_1 = A$, $A \notin X(\phi)$.
Since $\models A = A;E$, we can apply the argument which follows in case 4. (The case that $\sigma_1 = X$, $X \in X(\phi)$ follows as case 7.)

CASE 4. $\sigma_1 = A;\sigma_{11}$.
Note that $A \notin X(\phi)$, since σ_1 is regular in all elements of $X(\phi)$. By lemma 6.3, either $\sigma_2 = A;\sigma_{21}$, and

$$\phi(\models) \Longrightarrow \phi^{(1)}(\models) = \Phi \models \sigma_{11} \subseteq \sigma_{21}$$

or σ_2 cannot be written as $A;\sigma_{21}$, and

$$\phi(\models) \Longrightarrow \phi^{(1)}(\models) = \Phi \models \sigma_{11} \subseteq \Omega \ .$$

We assert that in both cases, argument A_3 applies.
We have to verify $A_{3.1}$ to $A_{3.4}$.

$A_{3.1}$: $\phi^{(1)}(\models)$ is a normal assertion. Since Φ is not changed, we have nothing to verify for clauses 2a, 2b, 2d, 2f and 2h of definition 6.1. Clause 2c follows from the fact that, if $A;\sigma_{11}$ is regular in X and shields X, then σ_{11} is regular in X and shields X. If X_i does not occur free in $A;\sigma_{21}$, then it does not occur free in σ_{21}; hence, clause 2e. Clause 2g follows since σ_{11} is a subterm of $A;\sigma_{11}$, and since σ_{21} and Ω are subterms of $A;\sigma_{21}$ and σ_2 respectively.

$A_{3.2}$: $\Gamma(\phi^{(1)}) < \Gamma(\phi)$. Clearly, $\gamma(\sigma_{11}) < \gamma(A;\sigma_{11})$. If Φ is non empty, then $S_l(\phi) = S_l(\phi^{(1)}) = |\Sigma(\rho_{1,0})|$, and $S_r(\phi) = S_r(\phi^{(1)}) = |\Sigma(\rho_{1,1})|$. If Φ is empty, then $S_l(\phi) = |\Sigma(A;\sigma_{11})| > |\Sigma(\sigma_{11})| = S_l(\phi^{(1)})$, and either

$S_r(\phi) = |\Sigma(A;\sigma_{21})| > |\Sigma(\sigma_{21})| = S_r(\phi^{(1)})$, or

$S_r(\phi) = |\Sigma(\sigma_2)| > |\Sigma(\Omega)| = S_r(\phi^{(1)})$. Since, clearly, $L_a(\phi) = L_a(\phi^{(1)})$, we see that $L_g(\phi) \geq L_g(\phi^{(1)})$, from which $\Gamma(\phi) > \Gamma(\varphi^{(1)})$ follows.

$A_{3.3}$: That $\phi(\models) \Longrightarrow \phi^{(1)}(\models)$ follows from lemma 6.3.

$A_{3.4}$: We have to show that $\phi^{(1)}(\vdash) \Longrightarrow \phi(\vdash)$

(α)
$$\frac{\Phi \vdash \sigma_{11} \subseteq \sigma_{21}}{\Phi \vdash A;\sigma_{11} \subseteq A;\sigma_{21}} .$$

This follows by monotonicity.

(β)
$$\frac{\Phi \vdash \sigma_{11} \subseteq \Omega}{\Phi \vdash A;\sigma_{11} \subseteq A;\Omega} .$$

Since $\vdash A;\Omega = \Omega$, and $\vdash \Omega \subseteq \sigma_2$, the result follows.

CASE 5. $\sigma_1 = (p\to\sigma_{11},\sigma_{12})$.

Let σ', σ'' be as in the proof of lemma 6.5. Let

$$\phi^{(1)}(\models) = \Phi \models \sigma_{11} \subseteq \sigma'$$
$$\phi^{(2)}(\models) = \Phi \models \sigma_{12} \subseteq \sigma'' \ .$$

We shall verify A_3.

$A_{3.1}$: This is similar to the verification of $A_{3.1}$ in case 4 above. Note the use of the fact that σ' and σ'' are subterms of σ_2.

$A_{3.2}$: This is again similar to case 4.

$A_{3.3}$: Follows by lemma 6.5.

$A_{3.4}$: Clear, by monotonicity.

CASE 6. $\sigma_1 = \mu X[\sigma_{11}]$, and $X \notin X(\phi)$.
Then

$$\phi(\models) = \Phi \models \mu X[\sigma_{11}] \subseteq \sigma_2 \ .$$

(α) X does not occur free in σ_{11}. Let

$$\phi^{(1)}(\models) = \Phi \models \sigma_{11} \subseteq \sigma_2 \ .$$

We omit the simple proof that A_3 applies to $\phi^{(1)}(\models)$.

(β) X does occur free in σ_{11}.
Let $X_{m+1} = X$ and $\rho_{m+1} = \sigma_{11}$. Let $\phi^{(1)}(\models)$ be the assertion

$$\Phi,\ X_{m+1} \subseteq \mu X_{m+1}[\rho_{m+1}] \models X_{m+1} \subseteq \sigma_2 \ .$$

We prove that A_3 applies to $\phi^{(1)}(\models)$.

$A_{3.1}$: $\phi^{(1)}(\models)$ is a normal assertion. Clauses 2a, 2b and 2c are clear. After rewriting, if necessary, we may assume that X_{m+1} does not occur free in σ_2 nor in any $\rho_{j,k}$, $1 \le j \le m, 0 \le k \le n_j$. Together with the normality of $\phi(\models)$, this yields clause 2e of the normality of $\phi^{(1)}(\models)$. Since ρ_{m+1} is regular in X_{m+1} and shields X_{m+1}, and since X_{m+1} does not occur free in ρ_i, $1 \leq i \leq m$ (by clause 2e), clause 2d follows from the normality of $\phi(\models)$. Clause 2f is trivially satisfied. Clause 2g follows from the normality of $\phi(\models)$ and the fact that X_{m+1} and σ_2 are subterms of $\mu X_{m+1}[\sigma_{11}]$ (since X_{m+1} occurs free in σ_{11}) and σ_2 respectively. The proof of 2h follows from the normality of $\phi(\models)$ and the fact that

$$\Phi, X_{m+1} \subseteq \mu X_{m+1}[\rho_{m+1}] \models \rho_{m+1} \subseteq \mu X_{m+1}[\rho_{m+1}]$$

which is clear from monotonicity and the f.p.p.

$A_{3.2}$: It is easily seen that $L_{max}(\phi) = L_{max}(\phi^{(1)})$. Since $L_a(\phi) < L_a(\phi^{(1)})$, we have $L_g(\phi) > L_g(\phi^{(1)})$; hence, $\Gamma(\phi) > \Gamma(\phi^{(1)})$.

$A_{3.3}$: That $\phi(\models) \Longrightarrow \phi^{(1)}(\models)$ is clear.

$A_{3.4}$: Assume $\phi^{(1)}(\vdash)$, i.e.

$$\Phi, X_{m+1} \subseteq \mu X_{m+1}[\rho_{m+1}] \vdash X_{m+1} \subseteq \sigma_2.$$

Since X_{m+1} does not occur free in Φ or σ_2, substituting $\mu X_{m+1}[\rho_{m+1}]$ for X_{m+1} yields

$$\Phi, \mu X_{m+1}[\rho_{m+1}] \subseteq \mu X_{m+1}[\rho_{m+1}] \vdash \mu X_{m+1}[\rho_{m+1}] \subseteq \sigma_2.$$

Hence, $\phi(\vdash)$, i.e.,

$$\Phi \vdash \mu X_{m+1}[\rho_{m+1}] \subseteq \sigma_2$$

follows.

CASE 7. $\sigma_1 = X_i$, for some $X_i \in X(\phi)$.
Then $\phi(\models)$ is of the form

$$\Phi_1,\dots,\Phi_{i-1},X_i \subseteq \mu_i[\rho_i],X_i \subseteq \rho_{i,1},\dots,X_i \subseteq \rho_{i,n_i},\Phi_{i+1},\dots,\Phi_m \models X_i \subseteq \sigma_2.$$

We distinguish two cases:

(α) $\sigma_2 = \rho_{i,j}$, for some j, $1 \leq j \leq n_i$.

Then, by lemma 6.1, argument A_2 applies.

(β) $\sigma_2 \not\subseteq \rho_{i,j}$, for all j, $1 \le j \le n_i$.

Let $\rho_{i,n_i+1} = \sigma_2$, and let, for $0 \le j \le n_i+1$,

$$\phi^{(j)}(\models) =$$

$$\Phi_1,\dots,\Phi_{i-1},X_i \subseteq \mu X_i[\rho_i],X_i \subseteq \rho_{i,1},\dots,X_i \subseteq \rho_{i,n_i+1},$$
$$\Phi_{i+1},\dots,\Phi_m \models \rho_i \subseteq \rho_{i,j} \;.$$

We assert that $\mathcal{A}_3$ applies.

$\mathcal{A}_{3.1}$: For each j, $0 \le j \le n_i+1$, $\phi^{(j)}(\models)$ is a normal assertion. Clauses 2a and 2b of definition 6.1 are clear. Clauses 2c, 2d and 2e follow from the normality of $\phi(\models)$. Clause 2f follows, since $\sigma_2 \not\subseteq \rho_{i,j}$, $1 \le j \le n_i$. Since ρ_i is a subterm of $\rho_{i,0}$ and ρ_{i,n_i+1} $(=\sigma_2)$ is a subterm of σ_2, clause 2g follows from the normality of $\phi(\models)$. The proof of 2h follows under $\mathcal{A}_{3.3}$.

$\mathcal{A}_{3.2}$: Since $L_{max}(\phi) = L_{max}(\phi^{(j)})$, and $L_a(\phi) = L_a(\phi^{(j)}) - 1$, we have $L_g(\phi) > L_g(\phi^{(j)})$.

$\mathcal{A}_{3.3}$: Assume $\phi(\models)$. By clause 2h of the normality of $\phi(\models)$, we have

(19) $$\Phi_1,\dots,\Phi_{i-1},X_i \subseteq \mu X_i[\rho_i],X_i \subseteq \rho_{i,1},\dots,X_i \subseteq \rho_{i,n_i} \models \rho_i \subseteq \rho_{i,j}$$
$$0 \le j \le n_i \;.$$

Since none of $X_{i+1},\dots,X_m$ occurs free in $\Phi_1,\dots,\Phi_i$, from $\phi(\models)$ we derive

(20) $$\Phi_1,\dots,\Phi_{i-1},X_i \subseteq \mu X_i[\rho_i],X_i \subseteq \rho_{i,1},\dots,X_i \subseteq \rho_{i,n_i} \models X_i \subseteq \sigma_2 \;.$$

Combination of (19) and (20) yields

$$\Phi_1,\dots,\Phi_{i-1},X_i \subseteq \mu X_i[\rho_i],X_i \subseteq \rho_{i,1},\dots,X_i \subseteq \rho_{i,n_i} \models \rho_i \subseteq \sigma_2 \;.$$

Thus, â fortiori,

$$\Phi_1,\dots,\Phi_{i-1},X_i \subseteq \mu X_i[\rho_i],X_i \subseteq \rho_{i,1},\dots,X_i \subseteq \rho_{i,n_i},$$
$$X_i \subseteq \rho_{i,n_i+1} \models \rho_i \subseteq \sigma_2$$

which proves clause 2h of the normality of $\phi^{(n_i+1)}(\models)$, and

$$\Phi_1,\ldots,\Phi_{i-1},X_i \subseteq \mu X_i[\rho_i],X_i \subseteq \rho_{i,1},\ldots,X_i \subseteq \rho_{i,n_i+1},$$
$$\Phi_{i+1},\ldots,\Phi_m \models \rho_i \subseteq \sigma_2$$

which proves $\phi^{(n_i+1)}(\models)$.

That $\phi^{(0)}(\models),\ldots,\phi^{(n_i)}(\models)$ hold follows from clause 2h of the normality of $\phi(\models)$.

$A_{3.4}$: Assume $\phi^{(j)}(\vdash)$ holds, for each j, $0 \leq j \leq n_i+1$, i.e.,

$$\Phi_1,\ldots,\Phi_{i-1},X_i \subseteq \mu X_i[\rho_i],X_i \subseteq \rho_{i,1},\ldots,X_i \subseteq \rho_{i,n_i+1},$$
$$\Phi_{i+1},\ldots,\Phi_m \vdash \rho_i \subseteq \rho_{i,j}$$

holds, for $0 \leq j \leq n_i+1$.

Since none of $X_{i+1},\ldots,X_m$ occurs free in $\Phi_1,\ldots,\Phi_i$, we also have

(21) $$\Phi_1,\ldots,\Phi_{i-1},X_i \subseteq \mu X_i[\rho_i],X_i \subseteq \rho_{i,1},\ldots,X_i \subseteq \rho_{i,n_i+1} \vdash \rho_i \subseteq \rho_{i,j}$$
$$\text{for } 0 \leq j \leq n_i+1.$$

Since X_i does not occur free in $\Phi_1,\ldots,\Phi_{i-1}$, we can apply the μ-induction rule, yielding

$$\Phi_1,\ldots,\Phi_{i-1} \vdash \mu X_i[\rho_i] \subseteq \rho_{i,j} \qquad , 0 \leq j \leq n_i+1 .$$

In particular, for $j = n_i+1$,

$$\Phi_1,\ldots,\Phi_{i-1} \vdash \mu X_i[\rho_i] \subseteq \rho_{i,n_i+1} \quad (=\sigma_2) .$$

From this, $\phi(\vdash)$, i.e.,

$$\Phi_1,\ldots,\Phi_{i-1},X_i \subseteq \mu X_i[\rho_i],X_i \subseteq \rho_{i,1},\ldots,X_i \subseteq \rho_{i,n_i},$$
$$\Phi_{i+1},\ldots,\Phi_m \vdash X_i \subseteq \sigma_2$$

follows.

CASE 8. $\sigma_1 = \mu X[\sigma_{11}]$, for some $X \in X(\phi)$, say $X = X_i$.
$\phi(\models)$ then has the form

$$\Phi_1,\dots,\Phi_{i-1},X_i \subseteq \mu X_i[\rho_i],X_i \subseteq \rho_{i,1},\dots,X_i \subseteq \rho_{i,n_i},$$
$$\Phi_{i+1},\dots,\Phi_m \models \mu X_i[\rho_i] \subseteq \sigma_2$$

with $\rho_i = \sigma_{11}$. Let $\rho_{i,n_i+1} = \sigma_2$, and let $\phi^{(j)}(\models)$, $0 \leq j \leq n_i+1$ be defined as

$$\phi^{(j)}(\models) =$$
$$\Phi_1,\dots,\Phi_{i-1},X_i \subseteq \mu X_i[\rho_i],X_i \subseteq \rho_{i,1},\dots,X_i \subseteq \rho_{i,n_i},$$
$$\Phi_{i+1},\dots,\Phi_m \models \rho_i \subseteq \rho_{i,j} .$$

We verify A_3.

$A_{3.1}$: This is similar to case 7.

$A_{3.2}$: Clearly, $L_g(\phi) = L_g(\phi^{(j)})$, $0 \leq j \leq n_i+1$. Also, $\gamma(\rho_i) < \gamma(\mu X_i[\rho_i])$.

$A_{3.3}$: Assume $\phi(\models)$. This implies that

$$\Phi_1,\dots,\Phi_{i-1},X_i \subseteq \mu X_i[\rho_i],X_i \subseteq \rho_{i,1},\dots,X_i \subseteq \rho_{i,n_i},$$
$$\Phi_{i+1},\dots,\Phi_m \models X_i \subseteq \sigma_2$$

holds. The proof of $\phi(\models) \Longrightarrow \phi^{(j)}(\models)$, $0 \leq j \leq n_i+1$, is then similar to that of $A_{3.3}$ in case 7.

$A_{3.4}$: Assume $\phi^{(j)}(\vdash)$, $0 \leq j \leq n_i+1$. Similar to $A_{3.4}$ in case 7, we infer from this

$$\Phi_1,\dots,\Phi_{i-1} \vdash \mu X_i[\rho_i] \subseteq \rho_{i,j}, \quad 0 \leq j \leq n_i+1$$

and, à fortiori, taking $j = n_i+1$,

$$\Phi_1,\dots,\Phi_{i-1},X_i \subseteq \mu X_i[\rho_i],X_i \subseteq \rho_{i,1},\dots,X_i \subseteq \rho_{i,n_i},$$
$$\Phi_{i+1},\dots,\Phi_m \vdash \mu X_i[\rho_i] \subseteq \sigma_2$$

i.e., $\phi(\vdash)$ holds.

This completes the case analysis of the proof of (18). Thus, the proof of the completeness theorem is completed.

7. CONCLUSIONS

Ever since their first appearance in a number of programming languages, recursive procedures have been a much discussed concept. Initially, their practical feasibility was hotly debated, as caused by the alleged difficulties in their implementation. This argument settled down after some years, and interest turned to investigations centring around the question of what type of problems are particularly suited for recursive formulation. In parallel with this, the first proof technique (recursion induction) for proving equivalence of procedures was proposed, and then applied and extended in a number of papers (see the references in section 1.2). Recently, as part of the current heigthened activity in the field of the theory of programming, more exphasis has been given to the clarification of the mathematical properties of recursive procedures.

With the present paper, in which we have explained Scott's theory (sections 2 and 3), applied it to various examples (section 4), and investigated some of its properties (sections 5 and 6), we hope to have contributed to this clarification.

It will not have escaped the readers attention that we have dealt only with a restricted case of recursive procedures, viz., those which may be called "monadic", i.e., in which only functions of *one* variable occur. Extensions to a treatment of the general case -functions of more than one variable- have been made in two directions:

a. Milner [24] has developed a generalization of the μ-calculus in which the functions concerned are polyadic, i.e., they are interpreted as functions from $\mathcal{D}^n \to \mathcal{D}^m$, for arbitrary integer n, m $\geq$ 0. In this formalism, he is also able to deal with assignment statements.

b. As mentioned in section 1, Scott has exploited the notions of monotonicity and continuity in a framework where a number of problems in the theory of computation can be dealt with in a very general way. Central to this approach is the idea of building a hierarchy of domains. Starting with some initial domain $\mathcal{D}_0$, which is provided with a suitable partial ordering, one constructs domains $\mathcal{D}_0 \rightarrow \mathcal{D}_0$, $\mathcal{D}_0 \rightarrow (\mathcal{D}_0 \rightarrow \mathcal{D}_0)$, $(\mathcal{D}_0 \rightarrow \mathcal{D}_0) \rightarrow (\mathcal{D}_0 \rightarrow \mathcal{D}_0)$, etc. Each $\mathcal{D}' \rightarrow \mathcal{D}''$ consists of all *continuous* functions from $\mathcal{D}'$ to $\mathcal{D}''$. This gives -among many other results- a natural way of dealing with functions of more than one variable. E.g., a function of two variables is considered as an element of $\mathcal{D} \rightarrow (\mathcal{D} \rightarrow \mathcal{D})$. In unpublished notes Scott has shown how to extend the μ-calculus to such structures.

The "monadic" μ-calculus also offers a number of problems for further investigation. To the list of applications in section 4, many more might be added. Usually, the interesting part will be to develop a special set of axioms, adapted to a specific problem area. After one has obtained some experience with the μ-calculus, the formal proofs themselves are mostly straightforward. As a further example, we have a system for dealing with symbol manipulation, which we plan to publish in a forthcoming paper.

The *regular* procedures of sections 5 and 6 are clearly the best understood type of procedures. As a first extension of our completeness proof, one might wish to incorporate Yanov's shift distributions [39]. These are easily formulated as assumptions in the μ-calculus: If a certain variable A does not change the value of the predicate p, one assumes $A(p \rightarrow X,Y) = (p \rightarrow AX,AY)$ (cf. section 4.2). Some modifications will have to be made in the completeness proof, in order to deal with these assumptions.

One might also be interested in implementing the strategy of the completeness proof as a computer program.

Another possible extension is the introduction of compound predicates, in particular, of having predicates which themselves are defined as recursive (boolean) procedures.

Almost nothing is known about general properties of non-regular procedures. If the μ-calculus were to play a part in the solution of the many open problems in this area, we would have achieved one of our main goals.

BIBLIOGRAPHY

[1] BEKIĆ, H., Definable operations in general algebra, and the theory of automata and flow charts, to appear.

[2] BÖHM, C. & G. JACOPINI, Flow diagrams, Turing machines, and languages with only two formation rules, Comm. ACM, 9, 366-372(1966).

[3] BURSTALL, R.M., Proving properties of programs by structural induction, Comp. J., 12, 41-48(1969).

[4] COOPER, D.C., The equivalence of certain computations, Comp. J., 9, 45-52(1966).

[5] COOPER, D.C., Mathematical proofs about computer programs, in Machine Intelligence, vol. 1, 17-28(eds. N.L. Collins and D. Michie), Edinburgh, Oliver & Boyd (1967).

[6] COOPER, D.C., Program scheme equivalences and second order logic, in Machine Intelligence, vol. 4, 3-15(eds. B. Meltzer and D. Michie), Edinburgh, Edinburgh University Press (1969).

[7] COOPER, D.C., Program schemes, programs, and logic, in Symposium on Semantics of Algorithmic Languages, Lecture notes in mathematics, vol. 188, 62-70(ed. E. Engeler), Berlin, Springer-Verlag (1971).

[8] DIJKSTRA, E.W., Notes on structured programming, T.H. Report 70-WSK-03, Technological University Eindhoven, Second Ed. (1970).

[9] ENGELER, E., Structure and meaning of elementary programs, in Symposium on Semantics of Algorithmic Languages, Lecture notes in mathematics, vol. 188, 89-101(ed. E. Engeler), Berlin, Springer-Verlag (1971).

[10] FLOYD, R.W., Assigning meanings to programs, in Proc. of a Symposium in Applied Mathematics, Vol. 19, Mathematical Aspects of Computer Science, 19-32(ed. J.T. Schwartz), Providence, Rhode Island, American Mathematical Society (1967).

[11] HOARE, C.A.R., Procedures and parameters, an axiomatic approach, in Symposium on Semantics of Algorithmic Languages, Lecture notes in mathematics, Vol. 188, 102-116(ed. E. Engeler), Berlin, Springer-Verlag (1971).

[12] ITO, T., Some formal properties of a class of non-deterministic program schemata, IEEE Conf. Record of the Ninth Annual Symposium on Switching and Automata Theory, 85-98, New York, IEEE (1968).

[13] KAPLAN, D.M., Regular expressions and the equivalence of programs, J. Comp. Syst. Sci., 3, 361-386(1969).

[14] KAPLAN, D.M., Recursion induction applied to generalized flow charts, Proc. 24th Nat. ACM Conf., 491-504, New York, ACM(1969).

[15] KNASTER, B., Un théorème sur les fonctions d'ensembles, Ann. Soc. Pol. Math. 6, 133-134 (1928).

[16] LANDIN, P.J., The mechanical evaluation of expressions, Comp. J., 6, 308-320 (1964).

[17] MANNA, Z., Properties of programs and the first-order predicate calculus, J. ACM, 16, 244-255 (1969).

[18] MANNA, Z., The correctness of programs, J. Comp. Syst. Sci., 3, 119-127 (1969).

[19] MANNA, Z. & J. MCCARTHY, Properties of programs and partial function logic, in Machine Intelligence, Vol. 5, 27-37(eds. B. Meltzer and D. Michie), Edinburgh, Edinburgh University Press (1970).

[20] MANNA, Z. & A. PNUELI, The validity problem of the 91-function, Artificial Intelligence Memo 68, Stanford University (1968).

[21] MANNA, Z. & A. PNUELI, Formalization of properties of functional programs, J. ACM, 3, 555-569 (1970).

[22] MCCARTHY, J., Towards a mathematical science of computation, in Information Processing, Proc. of IFIP Congress 62, 21-28 (ed. C.M. Popplewell), Amsterdam, North-Holland (1963).

[23] MCCARTHY, J., A basis for a mathematical theory of computation, in Computer Programming and Formal Systems, 33-70(eds. P. Braffort and D. Hirschberg), Amsterdam, North-Holland (1963).

[24] MILNER, R., Algebraic theory of computable polyadic functions, Comp. Science Memorandum no.12, University College of Swansea (1970).

[25] MORRIS Jr, J.H., Another recursion induction principle, Comm. ACM 14, 351-354 (1971).

[26] MORRIS Jr, J.H., A correctness proof using recursively defined functions, Comp. Center Technical Report no.39, Univ. of Cal., Berkeley (1970).

[27] PARK, D., Fixpoint induction and proofs of program semantics, in Machine Intelligence, Vol. 5, 59-78(eds. B. Meltzer and D. Michie), Edinburgh, Edinburgh University Press (1970).

[28] PARK, D., The Y-combinator in Scott's Lambda Calculus Models, unpublished notes, University of Warwick (1970).

[29] PARK, D., Notes on a formalism for reasoning about schemes, unpublished notes, University of Warwick (1970).

[30] SALOMAA, A., Two complete axiom systems for the algebra of regular events, J. ACM, 13, 158-169 (1966).

[31] SCOTT, D., private communication (1969).

[32] SCOTT, D. & J.W. DE BAKKER, A theory of programs, unpublished notes, IBM Seminar, Vienna (1969).

[33] SCOTT, D., Unpublished notes (1969).

[34] SCOTT, D., Outline of a mathematical theory of computation, Proc. of the Fourth Annual Princeton Conference on Information Sciences and Systems, 169-176, Princeton (1970).

[35] SCOTT, D., Lattice-theoretic models for the λ-calculus, to appear.

[36] SCOTT, D., The lattice of flow diagrams, in Symposium on Semantics of Algorithmic Languages, Lecture notes in mathematics, Vol. 188, 311-364(ed. E. Engeler), Berlin, Springer-Verlag (1971).

[37] STRACHEY, C., Towards a formal semantics, in Formal language description languages, Proc. IFIP Working Conf. 1964, 198-220, (ed. T.B. Steel Jr.), Amsterdam, North-Holland (1966).

[38] TARSKI, A., A lattice-theoretical fixpoint theorem and its applications, Pacific J. of Math., 5, 285-309 (1955).

[39] YANOV, Y.I., The logical schemes of algorithms, in Problems of Cybernetics, 1, 82-140, New York, Pergamon Press (1958).