Metric Domains for Completeness

by

Stephen G. Matthews

A dissertation submitted for
the degree of
Doctor of Philosophy

Department of Computer Science
University of Warwick
Coventry
United Kingdom

March 1985

Abstract

Completeness is a semantic non-operational notion of program correctness suggested (but not pursued) by W.W.Wadge. Program verification can be simplified using completeness, firstly by removing the approximation relation $\sqsubseteq$ from proofs, and secondly by removing partial objects from proofs. The dissertation proves the validity of this approach by demonstrating how it can work in the class of **metric domains**. We show how the use of Tarski's least fixed point theorem can be replaced by a non-operational unique fixed point theorem for many well behaved programs. The proof of this theorem is also non-operational. After this we consider the problem of deciding what it means for a function to be "complete". It is shown that combinators such as function composition are not complete, although they are traditionally assumed to be so. Complete versions for these combinators are given. **Absolute** functions are proposed as a general model for the notion of a complete function. The theory of mategories is introduced as a vehicle for studying absolute functions.

Contents

Acknowledgements

May I first express my gratitude to my supervisor Dr.W.W.Wadge for proposing an idea which has now become this dissertation. His support through torturous grilling sessions (1980-83) will not be forgotten.

Many thanks are due to the now disbanded Warwick Dataflow Research group led by Bill Wadge. In alphabetical order, Pete Cameron, Tony Faustini, Forouzan Golshani, Paul Pilgram, and Ali Yaghi. Their work on the experimental programming language Lucid provided me with constant support whenever my faith in the notion of "completeness" dwindled.

May I thank the Department of Computer Science at Warwick University for providing a very suitable environment in which I carried out my postgraduate research (1980-83). My study there was only made possible by a three year postgraduate research grant from the Science & Engineering Research Council of Great Britain. At this point may I also thank the Department of Computer Science at the University of Victoria of British Columbia in Canada for employment as a sessional lecturer (1983-4). This enabled my researches to continue. May I thank again my friends at Warwick for employing me (1984-5) as a temporary lecturer (1984-5), as this has enabled me to complete my dissertation and finally realise some of the goals of completeness.

Let me acknowledge here discussions with Prof. David Park prior to my submission which have proved very helpful.

Most importantly let me acknowledge the moral support provided my parents, as it has only been their belief in the value of further education which has

made me persist to the end.

Declaration

The work described in this dissertation, except where stated explicitly in the text, is my own original work. Furthermore, no portion of this dissertation has been submitted in support of any other degree at any other University.

Stephen G. Matthews

Chapter 1

Introduction

Section 1.1) : Background to the Thesis

Research into reasoning about functional programs forms the background for this dissertation. While functional languages are the most appropriate class of languages for developing our ideas on reasoning about programs it should be possible to apply the ideas presented in this dissertation to other languages as well. Functional languages are the most appropriate because they are the languages with the simplest denotational semantics, and it is this semantics which we use as the basis for judging the correctness of rules for reasoning about programs. A typical definition in such a language is the following.

$$\begin{array}{lll} f(n) & ::= & \textbf{if } n = 0 \\ & & \textbf{then } 0 \\ & & \textbf{else } f(n-1) + 2n - 1 \end{array}$$

It's denotation is the least fixed point $Y(F)$ of the following combinator F, where for each chain continuous function f over Ω, $F(f)$ is the function

$$\forall\ x \in \Omega\ .\ \ F(f)(x) = \begin{cases} 0 & \text{if } x = 0 \\ f(x-1) + 2x - 1 & \textit{otherwise} \end{cases}$$

The property which we now wish to prove of this least fixed point is that it is the square function $\lambda x \in \Omega . x^2$ (Ω denotes the flat domain of all non-negative integers with $\perp$, where $\perp^2 = \perp$). The usual proof consists of showing that the only fixed point of F is the square function and so must be the least one. This requires the following inductive proof. For each function f satisfying the definition show that

(i) $f(\perp) = \perp^2$

(ii) $f(0) = 0^2$

(iii) $\forall n > 0 . \ f(n-1) = (n-1)^2 \quad ==> \quad f(n) = n^2$

Such a proof manages to reason about the denotations of programs without direct reference to the theory of least fixed points. This is in contrast to the following alternative way we could have proved that the denotation of the definition is the square function. Our second proof consists of showing that $Y(F)$ is the limit of the chain of partial functions $f^0, f^1, \cdots$ where for each f^n

$$\forall x \in \Omega . \ f^n(x) = \begin{cases} x^2 & \text{if } x < n \\ \perp & \textit{otherwise} \end{cases}$$

It can easily be shown that the limit of this chain is the square function.

The second proof is clearly more tedious and unnatural than the first, using partial functions, chains and limits. The first proof avoids using partial orderings by making the assumption that for **welldefined** definitions the combinator F will have a unique fixed point. And so, our two example proofs show that a proof technique which uses a theory of unique fixed points may be more appropriate than techniques which use a theory of least fixed points. To establish a theory of the former kind first requires a test to verify programs "well defined". With such a test we would construct a proof of the following kind for the above example definition. Firstly show that the definition passes the test. Secondly, show that the square function is a fixed point of the definition by proving (using high school algebra) that

$$0^2 = 0 , \quad \text{and}$$

$$\forall\ n > 0 \quad n^2 = (n-1)^2 + 2n - 1$$

Then by a theorem of unique fixed points for well defined definitions it can be deduced that the denotation of the definition must be the square function. This proof is (what we shall call) **non–operational** in the sense that it does not use the approximation relation $\sqsubseteq$ and the theory of least fixed points. That is, the relation is not used to describe notions such as "computation" or "partially defined" which are often used in proofs of program properties.

McCarthy's **recursioninduction** rule for proving properties of recursive definitions also uses the idea of "well-definedness". Unfortunately his rule only works for flat domains. The following formulation is based upon [].

To prove that two functions $f_1 , f_2 : D \to D$ over a flat domain D are the same find a combinator $F : (D \to D) \to D$ such that

(i) $f_1 = F(f_1)$

(ii) $f_2 = F(f_2)$

(iii) $\forall\ x \in D\ .\ Y(F)(x)\ ne\ \perp$

Condition (iii) says that the least fixed point of F must be well defined.

The potential for proofs using "well definedness" in this way depends upon whether or not tests for proving programs "well defined" can be constructed. These tests may be divided into two types, **operational** tests and **non–operational** tests. Operational tests use $\sqsubseteq$. Recursion induction is such an operational test as reference is made directly in (iii) to the least fixed point combinator Y. Languages based on the von-Neumann model [Ba78] tend to be most ammenable to operational tests due to their reliance upon the notion of "state transition". A typical functional programming language will have an "evaluation" based semantics as well as a semantics of functions. Thus in such languages it is possible to design non-operational tests for "well definedness" which use only the properties of functions. Such non-operational tests for a program in a language are those tests which involve only an analysis of data structures.

Now that tests for proving programs "well defined" have been introduced it is time to talk about those tests which inspire the work in this dissertation. These are the operational tests for proving termination. Termination has become outdated as a notion of program correctness. This can be seen by the need for non-terminating software such as operating systems, and secondly with the arrival of languages involving the evaluation of infinite data structures. However, such tests are still needed as is clearly demonstrated in Hoare's work on communicating sequential processes [Ho83]. In such work, rules for reasoning about properties of programs tend to have termination as a precondition. Due to the unsolvability of the Halting Problem [Kr83] [Tu36] it is impossible to design a test to decide whether or not an arbitrary program terminates. However, restrictive tests can be designed to prove that structurally simple programs terminate. Unfortunately it does not seem to be possible to generalise the notion of termination to extend to more recent languages. An explanation to this problem has been suggested by W.W.Wadge [Wa81] based upon the following fact. The more recent languages have introduced new operational ideas, for example, infinite evaluation in functional programming [Mc63] [Fau83], evaluation with backtracking in logic programming [Ko79], and deadlock in processor based languages [Ho83]. Thus he has proposed that termination will not generalise because it is a notion no longer present in many recent languages.

Wadge has proposed (although not pursued) a format for a possible solution based upon the following observation. Although operational notions may have changed, non-operational notions expressed in terms of data structures have not changed to the same extent. For example, the notion of a function has survived the test of time. His suggestion is that an extensional notion of correctness to replace termination would do better with todays modern languages than termination itself. **Completeness** is Wadge's candidate for such a notion. In his own words,

> "A complete object (in a domain of data objects) is, roughly speaking, one which has no holes or gaps in it, one which cannot be further completed"

Wadge uses Kahn's model of dataflow networks [Ka74] to show how the notion of completeness can be used to prove programs well defined. The extensional "Cycle Sum Test" is constructed to prove that simple networks do not deadlock. While this test has an extensional formulation it's original proof is not conventional, that is, Wadge presented an operational proof. An extensional proof would have made his argument for completeness much stronger. It is the problem over this proof which leads us now to summarise the overall aims and results of this dissertation.

The thesis of this dissertation is that the notion of completeness is a worthy candidate for an extensional replacement to the notion of termination. Wadge's work has provided a lead, however, before completeness can be seriously considered operational proofs such as that mentioned above must be removed. Also, the notion of completeness must be made independent of Kahn's dataflow model. These two aims are achieved in this dissertation by producing a purely extensional general theory of completeness.

Section 1.2) : Aims of the Dissertation

Section 1.1) has introduced the notion of completeness, and has described the way in which it has been suggested as a tool for reasoning about programs. The overall aim of this dissertation is to show that the notion of completeness does have a place in denotational semantics. That is we aim to show that the notion of completeness can be formalised in the theory of domains. As mentioned in Section 1.1), our principal approach is to generalise the example given in [Wa81]. The first step in this approach is to choose a class of structures in which completeness can both be defined, and which generalises the semantic domain used in Kahn's networks. The class suggested by the author is the class of **Metric Domains**. These domains allow a natural distinction to be made between "complete" and "partial" (non complete) objects. Metric Domains can be used for both Kahn networks [Ka74] and the programming language LUCID [Fau83] [Wa85]. As the name "metric" suggests, metric domains take their inspiration from the theory of metric spaces [Suth]. The complete objects of a metric domain form a metric space, while the partial objects have a very similar structure. Although the structure of metric domains is a very simple extension of that for metric spaces, it does not involve any use of the approximation relation $\sqsubseteq$.

Chapter 2 constructs a general theory of completeness for metric domains. Even without the relation $\sqsubseteq$ it is shown in Section 2.2) that a theorem of unique fixed points can be constructed to provide a non-operational semantics for many well behaved programs. This theorem is a direct generalisation of The **Banach contraction mapping** fixed point theorem for metric spaces. Such a theorem provides much appeal to the philosophy of deriving tools for domains by generalising tools from the complete objects in the domains. An interesting implication of the existence of our fixed point theorem is that it refutes a suggestion made by Wadge [Wa81] that the cycle sum test cannot be extended to all metric spaces. Our work has taken his test beyond metric spaces and into metric domains. The following important question is raised by our theorem. If we can take Banach's theorem for complete objects (a theorem well used in the past by mathematicians) and use it with such ease in domains, can other "complete" theories be extended similarly. Section 2.3) applies the fixed point theorem for metric domains by using it to give a non-operational proof of a generalisation of Wadge's Cycle Sum Test. The fact that our proof is non-operational is an improvement upon Wadge's operational proof. Our **Cycle Product Test** is introduced in this work in order to show how Wadge's work could have been taken much further in demonstrating the use of completeness in reasoning non-operationaly about programs. Section 2.5) gives an example of how the work of Sections 2.2) & 2.3) can be used to define an alternative semantics for Kahn's deterministic networks. The example is not given as an attempt to improve Kahn's original semantics. However, similar but non-deterministic networks have been used by Park to study the "Fairness" Problem [Pa84]. Thus our example semantics is presented here to demonstrate that metric domains provide a very appealing non-operational framework in which such studies as Park's can be made. The example has a second purpose. It is easily noted of the domain used in the example that there are only complete objects, no partial objects. The example thus shows how metric domains of only complete objects are useful.

Chapter 3 considers the problem of removing partial objects from fixed point semantics. It is argued that this is done by "Completeness Rules", although the problems are formidable. The explanation of these problems is really quite simple. Interesting metric domains such as the LUCID domain of intermittent streams do not have their whole structure described by means of a metric. For categories of metric domains to exist with interesting closure properties our domains will have to have a structure in addition to that of a metric. To begin an investigation into the nature of such additional structure is outside the scope of this dissertation. It must be said however, that such an investigation is the way forward to finding a theory of categories for metric domains. Even though categories of arbitrary metric domains are beyond this work there are still interesting domains such as the one used in Section 2.4). That example demonstrates how metric domains of complete objects (that is, metric spaces) can still be useful for defining the semantics of a language. Chapter 3 considers fixed point semantics using only complete objects. **Agreement Spaces** have been suggested by Wadge [Wa81] as a class of metric spaces which could be used to establish a fixed point semantics not involving partial objects or the approximation relation. The principle aim of Chapter 3 is to show that it is sufficient to consider only compact agreement spaces for a conventional fixed point semantics of complete pograms not using higher order functions. This we do by constructing a theory of recursion equations for specifying compact agreement spaces.

Chapter 4 considers the problems of formulating a notion of completeness for functions. It is shown that function composition is not complete, and so we suggest a complete form of composition to replace it. The model of domains used for this Chapter is a **powerdomain** model. This model can be used to describe both the Kahn and Lucid domains. In fact, if the universe of the powerdomain is a metric domain then so is the powerdomain itself. It is shown that a theory of completeness for functions can be formulated. However, combinators such as composition must first be made complete. Also it is shown that the notion of "function" must first be restricted to "absolute function".

A conclusion from Chapter 4 is that absolute functions are necessary in a higher order theory of domains.

Chapter 5 considers the problems involved in setting up categories of absolute functions. It is shown that tight restrictions must be placed upon the kind of allowed absolute function in order to obtain categorical products and sums. Finally in Chapter 5 a suggestion is made to weaken the associativity axiom on composition of morphisms in category theory in order to obtain what we call **mategories**. In a mategory equality is replaced by a partial order. Thus mategories are extensions of categories just as metric domains are extensions of metric spaces. Accordingly all the usual categorical definitions of product, sum, and exponentiation are extended.

Chapter 2

Metric Domains as a model for Completeness

Section 2.1) : Introduction

This Chapter introduces the notion of **Metric Domain** in order to promote the notion of completeness in domain theory. The first intended application of metric domains is as a unifying model to describe completeness for two well used domains. These are the Kahn Domain of finite & infinite sequences [Ka74], and the Lucid Domain of intermittent infinite sequences [Wa85]. The second intended application of metric domains is an alternative approach to defining computability on metric spaces. The approach taken by Klaus Weihrauch [W&S] is to embed metric spaces into weighted algebraic cpos. Instead of adding extra objects as in his approach we reformulate the notion of "metric space" to get "metric domain". It will be pointed out in this chapter that there is a one to one correspondence between the class of metric domains and the class of metric spaces. However, a distinction between complete and partial objects can be made in a metric domain which cannot be made in a metric space. As will be shown in Chapter 3, there are metric domains in which a very natural notion of computability can be defined using completeness. However, in this chapter consideration is given to arbitrary metric domains. In order to justify our reformulation of metric spaces we show

how the **Banach Contraction Mapping Theorem** [Suth] from elementary metric space theory can be used in metric domain theory as a tool for program verification.

The work in this Chapter is divided into two parts. Section 2.2) takes Banach's theorem for complete metric spaces and proves it for complete metric domains. The interesting observation from this section is that Banach's theorem requires no reformulation even though metric domains have partial objects. Section 2.3) applies the reformulated theorem by formalising and proving a technique for verifying simple programs correct. The **Cycle Product Test** is a generalisation of Wadge's **Cycle Sum Test** for Kahn Dataflow Networks [Wa81] to arbitrary metric domains. The interesting observation of this section is that the correctness proof for the Cycle Product Test requires no use of the approximation relation $\sqsubseteq$ in contrast to Wadge's proof of the Cycle Sum Test. It is intersting because it shows how metric domains can have a notion of computability different to that based upon the approximation relation. This theme of an alternative notion is considered in Chapter 3. Section 2.4) justifies the work of the previous two sections by means of an example. It is shown how metric domains and Banach's theorem can be used to construct a fixed point semantics for Kahn Networks equivalent to Kahn's semantics [Ka74].

Section 2.2) : A Theorem for Unique Fixed Points

Notation

The set of all non-negative real numbers is denoted by 'R^+'.

Definition

A **Metric Domain** is a pair $<D,d>$ where D is a non-empty set, and d is a function from $D \times D$ to R^+ such that

$$\text{(i)} \quad \forall\, x,y \in D \qquad d(x,y) = 0 \implies x = y$$

$$\text{(ii)} \quad \forall\, x,y \in D \qquad d(x,y) = d(y,x)$$

$$\text{(iii)} \quad \forall\, x,y,z \in D \qquad d(x,z) \leq d(x,y) + d(y,z)$$

The Kahn Domain can be formulated as the metric domain $< Ka\ , kd >$ where for all x , y in Ka $d(x,y)$ is 2^{-n} for n the largest integer (or infinity) such that

$$n \leq \min\{\, |x|\, ,\, |y|\, \} \quad \text{and,}$$

$$\forall\, i < n \qquad x_i = y_i$$

In other words $d(x,y)$ is 2 to the minus the length of the common initial segment of x and y.

Definition

In a metric domain $<D,d>$, D is the set of **points**. A point x is **complete** if $d(x,x) = 0$.

A point in the Kahn Domain is thus complete precisely when it is an infinite sequence.

Notation

'ω' denotes the set of all non-negative integers.

As mentioned in the introduction, the unique fixed point theorem formulated and proved in this section is a generalisation of the Contraction Mapping Theorem [Suth] used in the theory of metric spaces. The latter theorem involves the use of convergent sequences, and so we do the same for domains. It is interesting to note here that this "metrical" form of convergence is, in effect, replacing Tarski's use of convergent chains in least fixed point semantics [Tar55].

Definition

In a metric domain $<D,d>$ $X \in {}^{\omega}D$ **converges** to a point y if

$$\forall \varepsilon > 0 \quad \exists\, n \geq 0 \quad \forall\, m \geq n \qquad d(X_m\, , y) \leq \varepsilon$$

Theorem 1

For each metric domain $<D,d>$, if $X \in {}^{\omega}D$ converges to $y \in D$ then y is complete.

Proof:

Let $<D,d>$ be a metric domain.

Let $X \in {}^{\omega}D$, and let $y \in D$.

Suppose X converges to y.

Let $\varepsilon > 0$.

Then there exists $n \geq 0$ such that

$$\forall\ m \geq n \quad d(X_m, y) \leq \frac{\varepsilon}{2}$$

Thus $\quad d(X_n, y) \leq \frac{\varepsilon}{2}$

However, $\quad d(y, y) \leq d(y, X_n) + d(X_n, y)$

$$= 2 * d(X_n, y)$$

$$\leq 2 * \frac{\varepsilon}{2}$$

$$= \varepsilon$$

Thus $\quad d(y, y) = 0,$

thus y is complete.

□

The next theorem tidies up a small point which is probably already obvious. A sequence can only converge to one point.

Theorem 2

For each metric domain $<D,d>$, if $X \in {}^{\omega}D$ converges to both $y \in D$ and $y' \in D$ then y=y'.

Proof:

Let $<D,d>$ be a metric domain.

Let $X \in {}^{\omega}D$, $y \in D$, $y' \in D$.

Suppose that X converges to both y and y'.

Let $\varepsilon > 0$.

Then there exists $n \geq 0$ and $n' \geq 0$ such that

$$\forall\ m \geq n \quad d(X_m, y) \leq \frac{\varepsilon}{2}$$

$$\forall\ m \geq n' \quad d(X_m, y') \leq \frac{\varepsilon}{2}$$

Let $t = \max\{n, n'\}$

Then, $d(X_t, y) \leq \frac{\varepsilon}{2}$ and $d(X_t, y') \leq \frac{\varepsilon}{2}$

$$\begin{aligned} \text{but, } d(y, y') &\leq d(y, X_t) + d(X_t, y') \\ &= d(X_t, y) + d(X_t, y') \\ &\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} \\ &= \varepsilon \end{aligned}$$

Thus $d(y, y') = 0$,

thus $y = y'$.

□

The next few definitions are analogues of those used in the theory of metric spaces. They will enable us to establish a domain version of the contraction mapping theorem.

Definition

In a metric domain $<D,d>$, $X \in {}^{\omega}D$ is **Cauchy** if

$$\forall\ \varepsilon > 0\ \exists\ n \geq 0\ \forall\ i,j \geq n \quad d(X_i, X_j) \leq \varepsilon$$

Definition

A metric domain is **complete** if every cauchy sequence converges.

Notation

For each $n \geq 0$, a function f composed with itself n times is denoted by 'f^n'.

Definition

A function $f : D \to D$ is a **contraction function** if there exists $c \in R^+$ such that $0 \leq c < 1$ and,

$$\forall\ x, y \in D\ .\ \ d(f(x), f(y)) \ \leq\ c * d(x,y)$$

All constant functions over metric domains returning complete points are contractions, as is the function f over the Kahn Domain such that,

$$f(<>) = <0>$$

$$f(<x_0, \ldots, x_n>) = <0, x_0{+}1, \ldots, x_n{+}1>$$

$$f(<x_0, x_1, \ldots>) = <0, x_0{+}1, x_1{+}1, \ldots>$$

We will be able to show later that f has the unique fixed point

$$<0, 1, 2, \ldots>$$

Theorem 3

For each contraction function f in a metric domain $<D,d>$, and for each $x \in D$, $\lambda n \in \omega . f^n(x)$ is contraction.

Proof:

Let $<D,d>$ be a metric domain.

Suppose $f : D \to D$ is a contraction function,

then there exists $c \in R^+$ such that $0 \le c < 1$ and

$$\forall\ y, y' \in D \quad d(f(y), f(y')) \le c * d(y, y')$$

Let $x \in D$.

$$\text{Now, } \forall\ i \ge 0 \quad d(f^i(x), f^{i+1}(x)) \le c * d(f^{i-1}(x), f^i(x))$$

$$\le c^2 * d(f^{i-2}(x), f^{i-1}(x))$$

$$\le c^3 * d(f^{i-3}(x), f^{i-2}(x))$$

and so on ...

$$\text{thus } \forall\ i \ge 0 \quad d(f^i(x), f^{i+1}(x)) \le c^i * d(x, f(x))$$

Let $\varepsilon > 0$.

Let $n \ge 0$ be such that, $\quad d(x, f(x)) * \dfrac{c^n}{1-c} \le \varepsilon$

and, $\quad c^n * d(x, x) \le \varepsilon$

Let $i, j \ge n$.

There are three cases to consider.

case 1 : $i = j$

$$\begin{aligned} d(f^i(x), f^j(x)) &= d(f^i(x), f^i(x)) \\ &\le c^i * d(x, x) \\ &\le \varepsilon \end{aligned}$$

case 2 : $i < j$

$$d(f^i(x), f^j(x)) \le \sum_{k=i}^{j-1} d(f^k(x), f^{k+1}(x))$$

$$\leq \sum_{k=i}^{j-1} c^k * d(x, f(x))$$

$$= d(x, f(x)) * \sum_{k=i}^{j-1} c^k$$

$$= d(x, f(x)) * \frac{c^i * (1 - c^{j-i})}{1 - c}$$

$$\leq d(x, f(x)) * \frac{c^i}{1 - c}$$

$$\leq d(x, f(x)) * \frac{c^n}{1 - c}$$

$$\leq \varepsilon$$

case 3 : $j > i$

similar to case 2.

□

Theorem 4

Each contraction function in a complete metric domain has a fixed point.

Proof:

Let $<D,d>$ be a complete metric domain.

Suppose $f : D \to D$ is a contraction function.

Then we can find $c \in R^+$ such that $0 \leq c < 1$ and,

$$\forall\ y, y' \in D \quad d(f(y), f(y')) \leq c * d(y, y')$$

Let $x \in D$.

Then by Theorem 3 $\lambda n \in \omega . f^n(x)$ is a cauchy sequence.

But as $<D,d>$ is complete $\lambda n \in \omega . f^n(x)$ converges to a point $l \in D$.

We will show that $\lambda n \in \omega . f^n(x)$ also converges to $f(l)$.

Let $\varepsilon > 0$.

Then we can find $m \geq 0$ such that

$$\forall\ i \geq m \quad d(f^i(x), l) \leq \frac{\varepsilon}{c+1}$$

thus, $\forall\ i \geq m+1 \quad d(f^i(x), f(l)) \leq c * d(f^{i-1}(x), l)$

$$\leq c * \frac{\varepsilon}{c+1}$$

$$\leq \varepsilon$$

Thus $\lambda n \in \omega . f^n(x)$ converges to $f(l)$.

But by Theorem 2 $l = f(l)$.

Thus f has a fixed point.

□

Theorem 5

Each contraction function in a complete metric domain has a unique fixed point, and this point is complete.

Proof:

Let $<D,d>$ be a complete metric domain.

Suppose $f : D \to D$ is a contraction function,

then we can find $c \in R^+$ such that $0 \leq c < 1$ and,

$$\forall\ y, y' \in D \quad d(f(y), f(y')) \leq c * d(y, y')$$

By Theorem 4, f has a fixed point $l \in D$.

Suppose that $l' \in D$ is a fixed point of f such that $l \neq l'$.

As f is a contraction function,

$$d(f(l),f(l')) \leq c * d(l,l')$$

thus,

$$d(l,l') \leq c * d(l,l')$$

but $c < 1$, and so $d(l,l') = 0$.

thus $l = l'$, a contradiction, and so $l = l'$.

Also, $d(l,l) = 0$, and so l is complete.

□

Theorem 5 gives us a metric domain version of the contraction mapping theorem for metric spaces. However, for domains there is a more interesting generalisation of this result concerning the following type of function.

Definition

In a metric domain $<D,d>$, a function $f:D \to D$ is a **virtual contraction** function if there exists $n > 0$ such that f^n is a contraction function.

Theorem 6

Each virtual contraction function in a complete metric domain has a complete fixed point.

Proof:

Let $<D,d>$ be a complete metric domain.

Suppose that $f:D \to D$ is a virtual contraction function,

then we can find $n > 0$ such that f^n is

a contraction function.

By Theorem 5 f^n has a unique complete fixed point l .

Thus, $f^n(l) = l$

thus $f(f^n(l)) = f(l)$

thus $f^n(f(l)) = f(l)$

thus $f(l)$ is also a fixed point of f^n

thus by Theorem 5 $l = f(l)$

thus f has a complete fixed point.

□

The following theorem is the unique fixed point theorem needed in the next section to prove the Cycle Product Theorem.

Theorem 7

Each virtual contraction function in a complete metric domain has a unique fixed point, and this point is complete.

Proof:

Let $<D,d>$ be a complete metric domain.

Let $f:D \to D$ be a virtual contraction function,

then by Theorem 6 f has a complete fixed point l say.

Suppose that l' is a fixed point of f .

Then $l = f(l)$ and $l' = f(l')$,

thus $\forall\ i \geq 0 \quad f^i(l) = f^{i+1}(l)$ and $f^i(l') = f^{i+1}(l')$

thus $\forall\ i \geq 0 \quad l = f^{i}(l)$ and $l' = f^{i+1}(l')$

As f is a virtual contraction function we can find $n > 0$ such that f^{n} is a contraction function.

But f^{n} has fixed points l and l' ,

thus by Theorem 5 $l = l'$

thus f has a unique fixed point l .

$\square$

Section 2.3) : The Cycle Product Test

Theorem 7 of the previous section is the required unique fixed point result for proving the **Cycle Product Test**. The Cycle Product Test is essentially Wadge's **Cycle Sum Test** [Wa81] generalised to all metric domains, The latter test was constructed to prove that certain well behaved Kahn Networks would not deadlock. A Kahn Network is a directed graph, the arcs of which are communication channels down which "tokens" travel, and the nodes of which are processing stations. The simplest nodes are those like "+", which correspond to ordinary operations on data items. The "+" node repeatedly awaits the arrival of tokens on it's two input arcs. As soon as there are tokens on both arcs, the two tokens are removed and a token representing their sum is sent out along the output arc. Input on different arcs need not arrive simultaneously or even at the same rate, and tokens awaiting processing queue on the arcs.

Some networks have cycles, that is, there are networks in which the tokens output by a node will (directly or via other nodes) be processed to become input tokens for that node. In such networks the possibility occurs that a node may be waiting for itself to produce a token which it needs for input. Such a situation is called **deadlock**. The Cycle Sum Test is a test which can be applied to Kahn Networks, such that every network passing the test is guaranteed not to deadlock. The work of Kahn [Ka74] and Faustini [Fau82] has shown an equivalence between Kahn Networks and sets of equations over the domain Ka . Wadge formulates his test both in terms of networks and equations, however, his justification of the test is described in terms of a "loose" operational semantics. This justification cannot be taken as a proof, as a proof requires a mathematical formulation of the operational semantics (e.g. that of Faustini [Fau82]) semantics for networks introduced above. Also, a proof of the validity of the equational formulation of the Cycle Sum Test has not been produced in the literature, and so we can say that the Cycle Sum Test has not yet been proved.

This Chapter introduces the **Cycle Product Test** as a generalisation of the Cycle Sum Test. Our formulation is solely in terms of equations over metric domains. Theorem 7 is used to show that any set of equations passing the Cycle Product Test has a unique fixed point, and that this point is complete. This Cycle Product Theorem and its proof make no use of the approximation relation $\sqsubseteq$. A result from this Chapter will thus be the first proof of not only the Cycle Sum Test, but also a test which can be applied to languages such as Lucid [Wa85].

Our notion of "sets of equations" will first be formalised using **systems**.

Definition

For each $n > 0$, the **n'th product** of a metric domain $<D,d>$ (denoted '$<D,d>^n$') is the metric domain $<D^n,d^n>$ where,

$$\forall \; x , y \in D^n \quad d^n(x , y) = \max\{ d(x_i , y_i) : i < n \}$$

Definition

A **system** (of equations) over a metric domain $<D,d>$ is a triple $< n , f , c >$ such that

(i) $n > 0$

(ii) $f \in (D^n \to D)^n$

(iii) $c \in \{0, 1, \cdots n-1\}^2 \rightarrow R^+$

(iv) $\forall\ i < n\ \forall\ x, y \in D^n$

$$d(f_i(x), f_i(y)) \leq \max\{\ c_{ij} * d(x_j, y_j) : j < n\ \}$$

An example of a system over the Kahn Domain is $< 2, f, c >$ where

$$f_0(x, <>) = <1, 2>$$

$$f_0(x, <y_0, \ldots, y_n>) = <1, 2, y_0, \ldots, y_n>$$

$$f_0(x, <y_0, y_1, \ldots>) = <1, 2, y_0, y_1, \ldots>$$

$$f_1(<>, y) = <>$$

$$f_1(<x_0, \ldots, x_n>, y) = <x_1, \ldots, x_n>$$

$$f_1(<x_0, x_1, \ldots>, y) = <x_1, x_2, \ldots>$$

and

$$c_{00} = 0,\quad c_{01} = 2^{-2},\quad c_{11} = 0,\quad c_{10} = 2$$

This example will be used later in this section.

By "solution" to a system of equations we mean a fixed point of the "key" of the system.

Definition

The **key** of a system $< n, f, c >$ in a metric domain $<D,d>$ is the function $s : D^n \rightarrow D^n$ such that,

$$\forall\ i < n\ \forall\ x \in D^n \quad s(x)_i = f_i(x)$$

Kahn takes the least solution of the key of a system in his work over Ka, however, in a system passing the Cycle Product Test there will only be one solution to choose from.

Theorem 1

For each system $< n , f , c >$ in a metric domain $<D,d>$,

$$\forall \; x , y \in D^n$$
$$d^n(s(x), s(y)) \leq \max\{ c_{ij} * d(x_j , y_j) : i , j < n \}$$

Proof:

Suppose $< n , f , c >$ is a system in a metric domain $<D,d>$.

Then for all $x , y \in D^n$

$$d^n(s(x), s(y)) = \max\{ d(s(x)_i , s(y)_i) : i < n \}$$
$$= \max\{ d(f_i(x), f_i(y)) : i < n \}$$
$$\leq \max\{ \max\{ c_{ij} * d(x_j , y_j) : j < n \} : i < n \}$$
$$= \max\{ c_{ij} * d(x_j , y_j) : i , j < n \}$$

□

Theorem 2

For each system $< n , f , c >$ in a metric domain $<D,d>$, and for each $k > 0$

$$\forall x , y \in D^n \quad d^n(s^k(x), s^k(y)) \leq$$
$$\max\{ c_{ij} * d(s^{k-1}(x)_j , s^{k-1}(y)_j) : i , j < n \}$$

Proof:

Suppose $< n , f , c >$ is a system in a metric domain.

Let $k > 0$.

Then, $\forall\ x , y \in D^n \quad d^n (s^k (x) , s^k (y))$

$$= \max\{ \ d(s^k (x)_i , s^k (y)_i) \ : \ i < n \ \}$$

$$= \max\{ \ d(s(s^{k-1}(x))_i , s(s^{k-1}(y))_i) \ : \ i < n \ \}$$

$$= \max\{ \ d(f_i(s^{k-1}(x)) , f_i(s^{k-1}(y))) \ : \ i < n \ \}$$

$$\leq \max \{ \ \max\{ \ c_{ij} * d((s^{k-1}(x))_j , (s^{k-1}(y))_j) \ : \ j < n \ \} : \ i < n \ \}$$

$$= \max\{ \ c_{ij} * d((s^{k-1}(x))_j , (s^{k-1}(y))_j) \ : \ i , j < n \ \}$$

□

Theorem 3

For each system $< n , f , c >$ in a metric domain $<D,d>$, and for each $k > 0$

$$\forall\ x , y \in D^n \quad d^n (s^k (x) , s^k (y)) \ \leq$$

$$\max \{ \ c_{ij} \ : i , j < n \ \} \ * \ \max \{ \ d(x_i , y_j) \ : i , j < n \ \}$$

Proof:

by induction using Theorem 2

□

The following definition of a "path" is used in the proof of the Cycle Product Theorem.

Definition

A **path** over $n > 0$ is a sequence $< p_0 , \cdots , p_k >$ of integers such that $k > 0$, and

$$\forall\ i < k \quad 0 \leq p_i < n$$

When we talk of a path $< p_0 , \cdots , p_k >$, what we are really talking about are the cycle product constants

$$c_{p_0 p_1} \ , \ \cdots \ , \ c_{p_{k-1} p_k}$$

What is particularly interesting (as we shall see later) is the product of all these constants.

Definition

A **cycle** over $n > 0$ is a path $< p_0 , \cdots , p_k >$ such that $p_0 = p_k$, and $k \leq n$.

Definition

A **sub cycle** of a path $< p_0 , p_1 , \cdots , p_k >$ is a sub-sequence of the form $< p_i , p_{i+1} , \cdots , p_j >$ which is also a cycle. The **remainder** of $< p_0 , p_1 , \cdots , p_k >$ after extracting $< p_i , p_{i+1} , \cdots , p_j >$ is $< p_0 , p_1 , \cdots , p_j , p_{j+1} , \cdots , p_k >$.

Notation

The length of a sequence x is denoted by ' $|x|$ '. The absolute value of an integer n is denoted by ' $|n|$ '.

Definition

A **cycle set** for a path p over $n > 0$ is a sequence $A = < A_0 , \cdots , A_{|A|-1} >$ of cycles such that

(i) $|p| \leq \left(\sum_{q \in A} (|q| - 1) \right) + n$

(ii) $|A| \geq \frac{|p|}{n} - 1$

(iii) There exists a sequence $< B_0, B_1, \cdots, B_{|A|} >$ of paths such that $B_0 = p$, and for each $0 \leq i < |A|$ A_i is a sub cycle of B_i, and B_{i+1} is the remainder of B_i after extracting A_i.

Theorem 4

Each path in a system has a cycle-set

Proof:

Suppose $n > 0$, and let p be a path over n.

Construct a cycle set A using the following algorithm.

Step 1 : Let X be the path p

Step 2 : If the length of X is more than n then we can choose a cycle $< X_i, \cdots, X_j >$ with length not more than $n+1$. Append the cycle to A. Let X now be $< X_0, \ldots, X_i, X_{j+1}, \ldots, X_{|X|-1} >$. Repeat Step 2.

Now, each time step 2 is executed the length of X is reduced by $| < X_i, \ldots, X_j > | - 1$.

Thus the final length of X is,

$$|p| - \sum_{q \in A} (|q| - 1)$$

But the algorithm finishes when the length of X is at most n , thus

$$|p| - \sum_{q \in A} (|q| - 1) \leq n$$

and so,

$$|p| \leq \sum_{q \in A} (|q| - 1) + n$$

Now, in each execution of step 2 the length of X is reduced by at most n , thus

$$\sum_{q \in A} (|q| - 1) \leq \sum_{q \in A} n$$

$$\leq n * |A|$$

Thus from above,

$$|p| \leq (n * |A|) + n$$

thus,

$$|A| \geq \frac{|p|}{n} - 1$$

□

Definition

The **product** of a path p in a system $< n , f , c >$ is

$$\prod_{i=0}^{|p|-2} c_{p_i p_{i+1}}$$

and is denoted by ('prod(p)').

Theorem 5

For each path p in a system $< n , f , c >$, and for each cycle-set A for p,

$$\text{prod}(p) \leq \left(\prod_{q \in A} prod(q) \right) * \text{M}$$

$$where\ M = \begin{cases} 1 & \text{if } \{ prod(q) : |q| \leq n \} = 0 \\ \max\{ prod(q) : |q| \leq n \} & otherwise \end{cases}$$

Proof:

as in Theorem 4.

□

Definition

A system is **cycle product complete** if the product of each cycle is less than 1. The example system given on p.24 is cycle product complete.

This then is our generalisation of the Cycle Sum Test. A system is said to pass the **Cycle Product Test** if it is cycle product complete.

Theorem 6

For each cycle product complete system $< n , f , c >$ there exists $m \geq 2$ such that for any path p of length m,

$$\left(N^{\left(\frac{|p|}{n} - 1\right)} \right) * \text{M} < 1$$

where $\text{N} = \max\{ \text{prod}(q) : q \text{ is a cycle, and } |q| \leq n+1 \}$

$$M = \begin{cases} 1 & \text{if } \{ prod(q) : |q| \leq n \} = 0 \\ \max\{ prod(q) : |q| \leq n \} & otherwise \end{cases}$$

Proof:

The product of each cycle in a cycle product complete system $< n , f , c >$ is less than 1, and so as there are only a finite number of cycles with length at most $n+1$,

$$\max\{ \text{prod}(q) : q \text{ is a cycle, and } |q| \leq n+1 \} < 1$$

□

Theorem 7

For each cycle product complete system there exists $m \geq 2$ such that for any path p of length m ,

$$\text{prod}(p) < 1$$

Proof:

Suppose $< n , f , c >$ is a cycle product complete system.

Let $N = \max\{ \text{prod}(q) : q \text{ is a cycle, and } |q| \leq n+1 \}$

Let $M = 1$ if $\{ \text{prod}(q) : |q| \leq n \} = 0$, and otherwise

$$\max\{ \text{prod}(q) : q \text{ is a cycle, and } |q| \leq n+1 \}$$

By Theorem 6 we can choose $m \geq 2$ such that for any path p of length m

$$(N^{(\frac{|p|}{n}-1)}) * M < 1$$

By Theorem 4 we can choose a cycle-set A for p , thus by Theorem 5

$$\text{prod}(p) \leq \left(\prod_{q \in A} prod(q) \right) * M$$

thus,

$$\text{prod}(p) \leq (N^{|A|}) * M$$

But $N < 1$ as $< n , f , c >$ is cycle product complete,

and so as A is a cycle set,

$$N^{|A|} \leq N^{(\frac{|p|}{n} - 1)}$$

thus,

$$\text{prod}(p) \leq (N^{(\frac{|p|}{n} - 1)}) * M$$

$$< 1$$

□

Theorem 8

The key of a cycle product complete system is a virtual contraction function.

Proof:

Suppose $< n , f , c >$ is a cycle product complete system

in a metric domain $< D , d >$.

Then by Theorem 7 we can choose $m \geq 2$ such that for any

path p of length m,

$$\text{prod}(p) < 1$$

Thus by Theorem 3, for all $x , y \in D^n$

$$d^n (s^{m-1}(x) , s^{m-1}(y)) \leq$$

$$\max\{ \text{prod}(p) * d(x_{p_{m-1}} , y_{p_{m-1}}) : |p| = m \}$$
$$\leq \max\{ (\max\{ \text{prod}(q) : |q| = m \}) * d(x_{p_{m-1}} , y_{p_{m-1}}) : |p| = m \}$$
$$= (\max\{ \text{prod}(q) : |q| = m \}) * (\max\{ d(x_{p_{m-1}} , y_{p_{m-1}} : |p| = m \})$$
$$\leq (\max\{ \text{prod}(q) : |q| = m \}) * (\max\{ d(x_i , y_i) : i < n \})$$
$$= (\max\{ \text{prod}(q) : |q| = m \}) * d^n(x , y)$$

Thus s^{m-1} is a contraction function.

Thus s is a virtual contraction function.

□

Theorem 9 (The Cycle Product Theorem)

The key of a cycle product complete system in a complete metric domain has a unique fixed point, and this point is complete.

Proof:

Suppose $< n , f , c >$ is a cycle product complete system in a complete metric domain $< D , d >$.

Then $< D , d >^n$ is a complete metric domain.

Thus by Theorem 7 of Section 2.2), and by Theorem 8, the key of $< n , f , c >$ has a unique fixed point, and this point is complete.

□

Section 2.4) : Convergence in Metric Domains

The previous sections have introduced the notion of completeness using metric domains. In particular, Banach's theorem from metric topology has been applied to metric domains, thus providing a verification rule for proving programs complete. As was noted at the time, the theorem required no reformulation in its transfer from metric spaces to metric domains. The aim of that work was to demonstrate how theorems such as Banach's can be extended from a metric space of complete objects to a metric domain which included partial objects. This was relatively easy as we were only interested in establishing a fixed point theorem for the complete objects in metric domains. That is, a theorem to prove that certain recursive definitions have unique complete solutions. However, Banach's theorem is not solely restricted to the verification of complete programs. It can easily be generalised to produce a theorem which verifies that a program has a unique fixed point, a point which may be either partial or complete. This section will show how Banach's theorem can be made to work in metric domains for both complete and partial objects. The first result of this is a theorem which allows us to give a unique fixed point semantics to a larger class of programs than before. The second result is a clearer demonstration than before of how Banach's original theorem for metric spaces can be reformulated to accomodate partial objects.

In Section 2.2) the notion of completeness was introduced into metric spaces by reformulating the notion of a metric. Other notions such as "convergence" and "cauchy sequence" remained unaltered in our work on Banach's theorem. Such notions do not have to be altered if all convergent sequences of interest converge to complete objects. However, what about sequences converging to a partial object? The definition of convergence used for metric domains was the following definition usually used for metric spaces.

$$X \in {}^{\omega}D \quad \text{"converges"} \quad \text{to} \quad y \in D \quad \text{if}$$
$$\forall \varepsilon > 0 \;\; \exists n \geq 0 \;\; \forall m \geq n \quad d(X_m, y) < \varepsilon$$

This definition is satisfactory when y is complete. However, what if y is partial and each X_n is equal to y ? In this case X does not converge according to this definition. To allow sequences to converge to partial objects the following reformulated definition of convergence is needed.

Definition

A sequence $X \in {}^{\omega}D$ is **K–convergent** in a metric domain $<D,d>$ if there exists an object $y \in D$ such that

$$\forall \varepsilon > 0 \;\; \exists n \geq 0 \;\; \forall m \geq n$$
$$d(X_m, y) < \varepsilon \quad \text{or} \quad X_m = y$$

If $<D,d>$ is a metric space then K-convergence is equivalent to the usual notion of convergence for metric spaces. K-convergence captures precisely the intuitive notion of convergence in the Kahn metric domain of finite & infinite sequences, however, it will be shown later in this section that there are other interesting domains for which K-convergence is not good enough. As the Kahn Domain is an important domain we will first show how K-convergence can be used to formulate a new version of Banach's theorem. A sequence in a metric domain is K-convergent if and only if it is convergent in one (or both) of the following two ways.

Definition

A sequence $X \in {}^{\omega}D$ is **completely K–convergent** in a metric domain $<D,d>$ if there exists a complete object $y \in D$ such that

$$\forall \varepsilon > 0 \quad \exists n \geq 0 \quad \forall m \geq n \qquad d(X_m, y) < \varepsilon$$

Definition

A sequence $X \in {}^{\omega}D$ is **partially K–convergent** in a metric domain $<D,d>$ if there exists a partial object $y \in D$ such that

$$\exists n \geq 0 \quad \forall m \geq n \qquad X_m = y$$

In metric spaces each convergent sequence is a cauchy sequence. In order that this result should hold in metric domains the notion of a cauchy sequence must be reformulated as follows.

Definition

A sequence $X \in {}^{\omega}D$ is **K–cauchy** if

$$\forall \varepsilon > 0 \quad \exists n \geq 0 \quad \forall k, m \geq n$$
$$d(X_k, X_m) < \varepsilon \quad \text{or} \quad X_k = X_m$$

Similarly, we have a notions of K-complete domain, and of K-contraction function.

Definition

A metric domain is **K–complete** if every K-cauchy sequence is K-convergent.

Definition

A function $f: D \to D$ in a metric domain $<D,d>$ is a K-contraction function if there exists $c \in R^+$ such that

$$0 \leq c < 1 \text{ , and}$$

$$\forall\, x,y \in D \quad d(f(x), f(y)) \leq c \,.\, d(x, y)$$
$$\text{or} \quad f(x) = f(y)$$

Theorem 1 (The K–Banach Theorem)

Each K-contraction function over a K-complete metric domain has a unique fixed point, and for each $x \in D$ this point is the limit of the sequence

$$\lambda\, n \in \omega \;.\; f^n(x)$$

Proof:

> The proof of this theorem is a simple generalisation of Theorems 3, 4, & 5 from Section 2.2) .

□

The generalisation of notions such as convergence is not as easy as the past few definitions may have suggested. This chapter concludes with an example of how K-convergence is not a good enough notion of convergence for arbitrary metric domains, although it is exactly what is wanted for the Kahn Domain. Our example is the problem of how to construct product domains in the category of all metric domains with K-continuous functions. Such domains are needed in the first instance to define multi-argument functions, and later for domain equations.

Definition

A function $f : D \to D$ is **K–continuous** if for each K-convergent sequence $X \in {}^{\omega}D$ the sequence

$$\lambda n \in \omega \,.\, f(X_n)$$

converges to $f(\mathit{lim}\, X)$.

This category is different from the contrasting category of all metric spaces and continuous functions. The category for spaces is closed under finite products, while the category for domains is not. In the former category, the product of two spaces $\langle D, d \rangle$ and $\langle D', d' \rangle$ is $\langle D \times D', d \times d' \rangle$ where,

$$\forall\ \langle x, x' \rangle, \langle y, y' \rangle \in D \times D'$$
$$d \times d'\,(\langle x, x' \rangle, \langle y, y' \rangle) = \mathit{sup}\{\, d(x, y),\ d'(x', y') \,\}$$

Let $p_0 : D \times D' \to D$ and $p_1 : D \times D' \to D'$ be the continuous functions such that,

$$\forall\ \langle x, x' \rangle \in D \times D' \qquad p_0(\langle x, x' \rangle) = x, \quad \text{and}$$
$$p_1(\langle x, x' \rangle) = x'$$

Then the product is shown to exist by proving that for arbitary continuous functions $f : A \to D$ and $g : A \to D'$, there exists a unique function $h : A \to D \times D'$ such that the following diagram commutes.

$$\begin{tikzcd}
 & A \arrow[dl, "f"'] \arrow[d, "h"] \arrow[dr, "g"] & \\
D & D \times D' \arrow[l, "p_0"] \arrow[r, "p_1"'] & D'
\end{tikzcd}$$

h is the function such that,

$$\forall x \in D \quad h(x) = \langle f(x), g(x) \rangle$$

while p_0 and p_1 are the usual projection functions. The projection functions are easily shown to be continuous, When $\langle D, d \rangle$ and $\langle D', d' \rangle$ are spaces h is easily shown to be continuous. However, when they are domains h is not in general K-continuous. For example, suppose $X \in {}^{\omega}D$ is a completely K-convergent sequence. Also, suppose that $f(X)$ is partially K-convergent, and that $g(X)$ is completely K-convergent. Then $h(X)$ is neither partially nor completely K-convergent, and so h is not K-continuous. The conclusion from this example is that K-convergence is not an adequate notion of convergence for metric domains which are to have any decent closure properties.

Section 2.5) : An Alternative Semantics for Kahn Networks

This section constructs an alternative denotational semantics for deterministic Kahn Networks in order to show how the results of the previous sections can be used to define a fixed point semantics for languages using metric domains. The correctness of the alternative semantics is proven. Our alternative semantics for deterministic networks is an obvious overkill in comparison to the simplicity and elegance of Kahn's semantics [Ka74]. This fact does not trivialise the value of our semantics however, as the approach used in our deterministic semantics is that used by Park [Pa84] in his study of nondeterministic networks. This section thus shows how an approach to nondeterminism such as Park's can be formalised using metric domains and Banach's theorem.

In Kahn Networks processes communicate via "first in, first out" queues. Inevitably processes sometimes have to wait if their input queues become empty. For deterministic networks these delays do not affect the final result of the network computation, and so do not occur in Kahn's semantics. However, for nondeterministic networks such delays may be the very cause of the nondeterminism. Our alternative semantics is in essence a Kahnian semantics which does take account of delays. Although we only consider deterministic networks, it is a necessary first step in Park's approach to non-determinism.

The Kahn Domain **Ka** is the set of all finite & infinite sequences under the "initial segment" partial ordering, taken over some universe of discourse such as ω. The complete objects in Ka are the infinite sequences, while the partial objects are the finite sequences. A new object τ is added to the universe of discourse to represent a delay. This object called a **hiaton** was originally suggested (but as of yet unpublished) by W.W.Wadge for the treatment of nondeterministic networks (The notation "τ" is from Park's use of hiatons in the "Fairness Problem" [Pa84]). For example, the sequence of factorials

$$< 1 , 2 , 6 , 24 , 120 , \ldots >$$

might be "hiatonised" to get

$$< 1 , \tau , 2 , \tau , \tau , 24 , \tau , \tau , \tau , 120 , \ldots >$$

In this example sequence the hiatons represent increasing delays. More precisely we have the following definitions.

Definition

Suppose Σ is the universe of discourse for the Kahn Domain. Then the **Hiatonic Kahn Domain "HKa"** is the metric domain of all finite & infinite sequences over $\Sigma \cup \{\tau\}$ where if n is the length of the common initial segment between x & y then the distance between x & y is 2^{-n}.

Definition

HKa is retracted onto Ka by the **essence** function $\varepsilon : HKa \rightarrow Ka$ where for each u in HKa, $\varepsilon(u)$ is the sequence u with all it's hiatons removed.

(The notation "essence" and "ε" are taken from [Pa84]) The equivalence relation upon HKa induced by epsilon is such that

$$\forall\, u,v \in \mathrm{HKa} \qquad u \sim v \quad \text{iff} \quad \varepsilon(u) = \varepsilon(v)$$

Chain continuous functions used by Kahn in his least fixed point semantics have analogues over HKa which are contraction functions. A technique for constructing analogues over a hiatonised Kahn Domain with partial objects has been used by Park [Pa84] in his studies of "fair parallelism". In our language, Park's technique is to add hiatons to sequences so that all continuous functions over Ka become contraction functions over HKa. We require functions over HKa to be contractions in order that Theorem 7 of Section 2.1) may be used to construct a unique fixed point semantics. A good example of how the hiatonisation of Kahn's functions works is the "whenever" function. $wvr : Ka^2 \rightarrow Ka$ can be turned into a contraction function (this is based on the Lucid "whenever" function [Fau83]). wvr filters out a subsequence of x according to a sequence of booleans y. Thus for example,

$$wvr(\ <a,b,c,d,e,f,g,h,i,j>\ ,\ <1,0,1,1,0,0,0,1>\ )$$
$$= \ <a,c,d,h,>$$

The contraction function $wvr^{\tau} : HKa^2 \rightarrow HKa$ can be defined by placing hiatons into $wvr(x,y)$. First one τ is placed at the begining, and then one τ is added for each 0 in y. Thus,

$$wvr^{\tau}(\ <a,b,c,d,e,f,g,h,i,j>\ ,\ <1,0,1,1,0,0,0,1>\ )$$
$$= \ <tau,a,tau,c,d,tau,tau,tau,h>$$

Hiatons occuring in the arguments to wvr^{τ} are propagated through to the result. The correctness of a function such as wvr is expressed by the by the following "commutativity" equation.

$$\forall\ x, y \in HKa \qquad \varepsilon(\ wvr^{\tau}(x,y)\ )\ =\ wvr(\ \varepsilon(x)\ ,\ \varepsilon(y)\ )$$

Having demonstrated that the chain continuous functions over Ka used by

Kahn have correct contraction analogues over HKa we now have to show that the fixed point theory of Section 2.2) is a correct analogue for Kahn's least fixed point theory. That is, the following theorem has to be proved.

Theorem 1

For each chain continuous function f over Ka with least fixed point $Y(f)$ and with a correct contraction analogue f^τ over HKa, the unique fixed point $\mu(f^\tau)$ of f^τ is such that

$$\varepsilon(\ \mu(\ f^\tau\ )\ )\quad =\quad Y(\ f\ )$$

Proof:

First it is shown that $\varepsilon(\mu(f^\tau))$ is a fixed point of f .

$$\begin{aligned} \text{Now,}\quad \mu(f^\tau) &= f^\tau(\ \mu(f^\tau)\ ) \\ \text{thus,}\quad \varepsilon(\ \mu(f^\tau)\ ) &= \varepsilon(\ f^\tau(\ \mu(f^\tau)\ )\ ) \\ &= f(\ \varepsilon(\ \mu(\ f^\tau\ )\ )\ ) \end{aligned}$$

as f^τ is a correct analogue of f

Thus $\varepsilon(\mu(f^\tau))$ is a fixed point of f .

Now it is shown that this is the least fixed point.

Suppose that p is a fixed point of f , then we will show that

$$\varepsilon(\ \mu(\ f^\tau\ )\ )\ \sqsubseteq\ p$$

Let $u \in (\Sigma^\omega)^\omega$ be such that

$u_0 \; = \; h(p)$, and

$\forall\, n > 0 \quad u_n \; = \; f^\tau(\, u_{n-1}\,)$

(the function h gives p a "tail" of hiatons)

Then as f^τ is a contraction mapping, u converges to $\mu(f^\tau)$.

Also, it can be shown inductively that

$$\forall\, n \geq 0 \quad \varepsilon(\, u_n\,) \;=\; p$$

But as u is convergent,

$$\forall\, n \geq 0 \quad \varepsilon(\, lim\, u\,) \;\sqsubseteq\; \varepsilon(\, u_n\,)$$

Thus,

$$\varepsilon(\, \mu(\, f^{\tau}\,)\,) \;\sqsubseteq\; p$$

Thus,

$$\varepsilon(\, \mu(\, f^{\tau}\,)\,) \;=\; Y(\, f\,)$$

□

Thus the domain HKa can be used to reformulate Kahn's semantics for deterministic networks. A reformulation which uses Banach's fixed point theorem, and no partial ordering. It is admitedly an "overkill" for deterministic networks, however there are reasons why the example is not trivial. The hiaton has proved useful in the analysis of nondeterministic networks. As Park [Pa84] shows, non-determinism and least fixed points do not always go together. His use of hiatons enables him to use a "unique fixed point" theorem to overcome this problem. There is thus a reason for considering hiatons in non-deterministic Kahn Networks. It is now a natural question to ask whether or not the notion of a hiaton can be generalised to more interesting domains. Our contribution towards giving a positive answer to this question is as follows. Using the powerdomain operator P() introduced in Chapter 3, we can construct our own theory of hiatonic nondeterminism using metric domains. Due to our restriction in Chapter 3 to compact spaces, it would be a theory of "unfair nondeterminism". This is in contrast to Park who is interested in "fair nondeterminism".

Chapter 3

Fixed Point Semantics without Partial Objects

Section 3.1) : Introduction

The previous Chapter has shown how the fixed point semantics of many correct recursive programs can be given without using the usual theory of approximation. This was done by firstly using Wadge's notion of completeness as a notion of correctness, and secondly by working in our own specially constructed metric domains. The result of that work has been to refute the following suggestion of Wadge [Wa81],

> It is not possible (as far as we know) to formulate the cycle sum theorem purely in terms of functions on an abstract metric space.

We have not only formulated it (as the Cycle Product Theorem) for metric spaces, but for metric domains as well. Also, our proof is extensional, that is, it does not use the approximation relation $\sqsubseteq$. However, being able to reason about correct programs without $\sqsubseteq$ is not the main reason for extending the Cycle Sum Theorem to metric spaces. The main reason is that it is the first step in realising Wadge's dream of being able to define the fixed point semantics of correct programs without using partial objects at all. In his own words,

> ..... a fixed point semantics for a large class of "obviously terminating" recursive programs which would be mathematically conventional in that it could completely avoid reference to partial objects and approximation.

First let us describe how this idea of removing partial objects works in our theory of metric domains. Suppose that $< n , f , c >$ is a cycle product complete system of equations over a domain D. Then the functions f_i map complete objects in D^n to complete objects in D. (For non-recursive f_i this is ensured by the existence of the constants c_{ij} ; the recursive case is covered by the Cycle Product Theorem) Let f'_i be the functions f_i restricted to the complete objects, then the system $< n , f' , c >$ is cycle product complete. Also, the keys of the two systems have the same unique fixed point. This means that the following rule can be developed for proving the completeness of programs.

> Suppose that $f \in (D^n \to D)^n$ is a set of equations we wish to prove has a unique fixed point which is complete. Suppose first that we can find a cycle product complete system $< n , f' , c' >$ over the complete objects of D such that each $f_i{}'$ is the restriction of f_i to the complete objects. Suppose that from such a system we can deduce the existence of a cycle product complete system $< n , f , c >$. Then we can conclude that the set of equations f has a unique fixed point, and that this point is complete.

Agreement spaces are then shown to be topologically equivalent to the class of **ultrametric spaces**. Next it is shown in Section 3.2) that agreements are more appropriate than are ultrametrics for reasoning about completeness. Finally in that Section, in preparation for constructing our categorical theory of equations, other topological properties of agreement spaces are considered. The most important of these is the proof of the equivalence between **separable complete agreement spaces** and **closed subspaces** of the **Baire Null Space**. Section 3.3) constructs the category **CAS** of **compact agreement spaces**., while Section 3.4) proves the above mentioned closure properties. Section 3.5) shows how we solve recursive equations over compact agreement spaces using initial fixed points. Finally, Section 3.6) gives some brief example solutions to recrusive equations.

Section 3.2) : Agreement Spaces

This section begins by introducing Wadge's notion of an **agreement**. We will use it later as an alternative to a metric in the construction of a category of metric spaces. Firstly we show the topological equivalence between agreement spaces and ultrametric spaces. The main aim of this section is to introduce agreement spaces in preparation for the categorical construction of the next section. The following notions are all shown to be equivalent in the sense of uniform homeomorphism.

compact agreement space

compact ultrametric space

compact subspace of the Baire Null Space

compact totally disconnected metric space

The proofs of these equivalences have all been completed by the author, however, some are lengthy and technically uninteresting. Thus their inclusion would not benefit the theme of completeness in the dissertation. For this reason the proofs have not been included. For background topology see [Suth] or [Sim].

The notion of an agreement was introduced by Wadge [Wa81] in the concluding paragraph to his paper.

> It is not possible (as far as we know) to formulate the cycle sum theorem purely in terms of functions on an abstract metric space. But it is possible, however, if we use instead of a metric a dual notion which we call an **agreement**: a function which assigns to any two points a nonnegative

element of $\omega \bigcup \{\infty\}$ which measures how close together the points are, yielding ∞ if they coincide. This approach could allow a fixed point semantics for a large class of "obviously terminating" recursive programs which would be mathematically conventional in that it could completely avoid reference to partial objects and approximation.

This suggestion has neither been formalised nor developed until the work in this dissertation. We have formalised it in so far as we regard it as a first step in establishing Completeness Rules. Also, we develop the notion of agreement in three different ways. Firstly, we relate agreement spaces to the more general notion of a metric. Secondly, other interesting topological properties of agreement spaces are established. Thirdly, we later construct a category of agreement spaces for use in Completeness Rules.

Definition

An **agreement space** is an ordered pair $<D,\alpha>$ where D is a non-empty set, and the **agreement** α is a function from $D \times D$ to $\omega \bigcup \{\infty\}$ such that for all x, y, z in D

$$x = y \quad \text{iff} \quad \alpha(x,y) = \infty$$

$$\text{and,} \quad \alpha(x,y) = \alpha(y,x)$$

$$\text{and,} \quad \alpha(x,z) \geq \inf\{\ \alpha(x,y)\ ,\ \alpha(y,z)\ \}$$

An example of an agreement space is the **Baire Null Space** $<\omega^\omega,\alpha>$ of all infinite sequences of non-negative integers. The agreement between any two sequences is the length of their common initial segment.

Definition

An **Ultrametric Space** is an ordered pair $<D,d>$ where D is a non-empty set, and the **ultrametric** d is a function from $D \times D$ to the non-negative reals such that for all x,y,z in D

$$d(x,y) = 0 \quad \text{iff} \quad x = y$$

$$\text{and,} \quad d(x,y) = d(y,x)$$

$$\text{and,} \quad d(x,z) \leq \sup\{\, d(x,y)\ ,\ d(y,z)\,\}$$

Each agreement space $<D,\alpha>$ is topologically equivalent to the ultrametric space $<D,d>$ where,

$$\forall\, x\,,y \in D\,.\quad d(x,y) = 2^{-\alpha(x,y)}$$

In fact, each agreement space is uniformly homeomorphic to an ultrametric space. The reverse uniform homeomorphism is constructed as follows. Suppose that $<D,d>$ is an ultrametric space. Let $<D,\alpha>$ be the agreement space such that for all x,y in D

$$d(x,y) \geq 1 \quad ==> \quad \alpha(x,y) = 0$$

$$\forall\, n > 0 \quad 2^{-n} \leq d(x,y) < 2^{-n+1} \quad ==> \quad \alpha(x,y) = n$$

$$d(x,y) = 0 \quad ==> \quad \alpha(x,y) = \infty$$

Thus all metrical and topological notions associated with ultrametric spaces can be carried over to agreement spaces. Wadge did not prove the equivalence between ultrametric and agreement spaces, however, his belief was that partial objects could be avoided by a device such as this in a fixed point semantics for "obviously terminating recursive programs". His notion of a partial object [private communication] is embodied in the following idea of an agreement domain.

Definition

An **agreement domain** is an ordered pair $<D,\alpha>$ where D is a non-empty set, and $\alpha : D \times D \to \omega \cup \{\infty\}$ is such that for all x, y, z in D

$$\alpha(x,y) = \infty \quad ==> \quad x = y$$

$$\text{and,} \quad \alpha(x,y) = \alpha(y,x)$$

$$\text{and,} \quad \alpha(x,z) \geq \inf\{\ \alpha(x,y)\ ,\ \alpha(y,z)\ \}$$

As in the case of agreement spaces above, each agreement domain can be mapped onto an ultrametric domain by the identity function which is bicontinuous. An agreement domain appears to be more useful for reasoning about programs than Wadge's suggestion of agreement spaces. For example, in the agreement domain formulation of the Kahn Domain the agreement between two sequences is the length of their common initial segment. In this domain the following rule holds.

$$\forall\, x,y \in D \quad (\ \forall\, z \in D \quad \alpha(x,z) = \alpha(y,z)\ ) \quad ==> \quad x = y$$

In such domains the usual partial ordering $\sqsubseteq$ can be defined as follows

$$\forall\, x,y \in D \quad x \sqsubseteq y \quad \text{iff} \quad \forall\, z \in D \quad \alpha(x,z) \leq \alpha(y,z)$$

Thus in principle anything approximation theory can do in these domains so can agreements. A second example of how agreements are more useful for reasoning about programs than ultrametrics is in inductive proofs. Induction over ω is clearly easier than induction over the reals, and so rules such as (remember $\infty > \infty$),

x is complete if

$$\exists\, y \in D^{\omega} \quad \forall\, n > 0 \quad \alpha(x\,,y_{n+1}) \;>\; \alpha(x\,,y_n\,)$$

are simpler than the ultrametric analogue,

x is complete if

$$\exists\, c \geq 0 \quad \exists\, y \in D^{\omega} \quad c < 1 \quad \text{and,}$$

$$\forall\, n > 0 \quad d(x\,,y_{n+1}) \;\leq\; c\,.\,d(x\,,y_n\,)$$

Now that agreements have been introduced we return to the topology of agreements. The remainder of this section establishes the equivalent formulations of agreement spaces, thus justifying the use of agreements in the category theory of Section 3.3).

Definition

A topological space is **separable** if its set of points has a countable dense subset.

Theorem 1

Each separable agreement space is uniformly homeomorphic to a subspace of the Baire Null Space (Two spaces are said to be uniformly homeomorphic if both the homeomorphism and it's inverse are uniformly continuous).

Proof:

Let $< D\,, \alpha >$ be a separable agreement space.

For each $x \in D$ and for each $n \in \omega$ let $D_n{}^{\alpha}(x)$ denote

$\{\; y \mid y \in D$ and $\alpha(x,y) > n \;\}$

Let D^{α} denote

$\{\; D_n{}^{\alpha}(x) \mid x \in D$ and $n \in \omega \;\}$

Let C be an injective function from a subset of ω onto a countable dense subset of D.

Let $f : D^\alpha \to \omega$ be such that

$$x \in D^\alpha \implies C_{f(x)} \in x$$

Such a function exists as the range of C is dense in D.

Let $g : D \to \omega^\omega$ be such that

$$\forall x \in D \ \forall n \in \omega \quad g(x)_n = f(\ D_n{}^\alpha(x)\ )$$

Then it can be shown that g is a uniform homeomorphism.

□

Theorem 2

Each subspace of the Baire Null Space is uniformly homeomorphic to a separable agreement space.

Proof:

Let D be a subset of the Baire Null Space.

Then as there are a countable number of finite sequences of non-negative integers, we can for each $n > 0$ construct a countable subset D^n of D such that,

$$\forall x \in D \ \exists y \in D^n \quad \alpha(x,y) > n$$

Thus, $\bigcup_{n>0} D^n$ is a countable dense subset of D.

□

Definition

A metric space is **complete** if each cauchy sequence converges.

Theorem 3

Each separable complete agreement space is uniformly homeomorphic to a closed subspace of the Baire Null Space.

Theorem 4

Each closed subspace of the Baire Null Space is uniformly homeomorphic to a separable complete agreement space.

The proofs of Theorems 3 & 4 reduce to the proof of showing that a subset of a complete metric space is complete precisely when it is closed. Such a proof can be found in [Sim].

Definition

A metric space is **compact** if each sequence of points has a convergent subsequence.

By [Suth] p.82, each compact subspace of a metric space is closed, thus a compact subspace of the Baire Null Space is a closed subspace. By [Sim] p.111, the continuous image of a compact space is compact, thus by Theorem 4, each compact subspace of the Baire Null Space is uniformly homeomorphic to a compact agreement space. Now for the reverse equivalence. By [Sim] p.125 if a metric space is compact then it is complete and totally bounded. Also, by [Suth] p.111 if a metric space is totally bounded then it is separable. Thus each compact agreement space is a compact complete separable agreement space. Thus by Theorem 3 (and as by [Sim] p.111 the continuous image of a compact space is compact), each compact agreement space is uniformly homeomorphic to a compact subspace of the Baire Null Space.

Definition

A **disconnection** of a topological space is a pair of disjoint non-empty open sets whose union is the whole space. A space is **totally disconnected** if each pair of distinct points can be separated by a disconnection.

Theorem 5

Each compact totally disconnected metric space is uniformly homeomorphic to a compact agreement space.

Theorem 6

Each compact agreement space is uniformly homeomorphic to a compact totally disconnected metric space.

To recap, in a compact world the following notions are all equivalent.

agreement space

ultrametric space

subspace of the Baire Null Space

totally disconnected metric space

Section 3.3) : A Category of Agreement Spaces

Results from Section 3.2) show that the compact agreement spaces are equivalent to the compact ultrametric spaces, which are equivalent to the compact subspaces of the Baire Null Space. This section defines the category **CAS** of compact agreement spaces which we use for giving a semantics to metric space equations such as

$$D \equiv A + P(B) \times D$$

Such equations use the operators **disjoint sum** $+$, **cartesian product** $\times$, and a **powerdomain** operator $P()$. There are two steps in constructing the semantics. Firstly each equation is translated into the form

$$D \equiv F(D)$$

where F is a **finite continuous** functor over CAS . Thus the above example becomes

$$D \equiv (FA + (FB \times FI))(D)$$

where FA , FB , and FI are all finite continuous functors over CAS . In Section 3.5) we show that each finite continuous functor F has an **initial fixed point**, which is taken to be the semantics. The approach is similar to that used by D.Lehmann in his Ph.D. dissertation (University of Jerusalem), and involves many lengthy proofs of closure properties.

CAS is defined to be the category whose objects are the non-empty compact agreement spaces, and whose arrows are the continuous functions over those spaces. The identity arrows in CAS are the identity functions, while the isomorphisms are the bicontinuous functions. Sums and products exist in CAS . A sum of two compact spaces $<D,\alpha>$ & $<D',\alpha'>$ is the compact space $<D+D', \alpha+\alpha'>$ where $D+D'$ denotes the disjoint sum of D & D', and $\alpha+\alpha'$ denotes the agreement on $D+D'$ such that

$$
\begin{aligned}
\forall\, x, y \in D+D'.\quad (\alpha+\alpha')(x, y) &= 0 && \text{if } x \in D \ \&\ y \in D' \\
&= 0 && \text{if } x \in D' \ \&\ y \in D \\
&= \alpha(x, y) + 1 && \text{if } x \in D \ \&\ y \in D \\
&= \alpha'(x, y) + 1 && \text{if } x \in D' \ \&\ y \in D'
\end{aligned}
$$

A product of $<D,\alpha>$ and $<D',\alpha'>$ is the compact space $<D\times D', \alpha\times\alpha'>$ where $D\times D'$ denotes the cartesian product of D & D', and $\alpha\times\alpha'$ denotes the agreement on $D\times D'$ such that

$$
\begin{aligned}
&\forall\, x, y \in D.\ \forall\, x', y' \in D'. \\
&\quad \alpha\times\alpha'(\ <x, x'>, <y, y'>\ ) \ = \ \inf\{\ \alpha(x, y)\ ,\ \alpha'(x', y')\ \}
\end{aligned}
$$

Definition

A **chain** in *CAS* is a sequence of objects and arrows of the form

$$<A_0,\alpha_0> \xleftarrow{f_0} <A_1,\alpha_1> \xleftarrow{f_1} <A_2,\alpha_2> \xleftarrow{f_2} \cdots$$

CAS is a **chain complete** category, that is, each chain has a "limit". The notion of a "limit" is formalised in category theory by means of a **cone** which we now introduce. For each sequence α (called a **chain**)

$$a_0 \xleftarrow{\alpha_0} a_1 \xleftarrow{\alpha_1} a_2 \xleftarrow{\alpha_2} \cdots$$

of objects & arrows in CAS, and for each object D, a cone with base α and vertex D is a commutative diagram

$$\begin{tikzcd}
a_0 & a_1 \arrow[l, "\alpha_0"'] & a_2 \arrow[l, "\alpha_1"'] & \cdots \arrow[l, "\alpha_2"'] \\
D \arrow[u, "\tau_0"'] \arrow[r, equal] & D \arrow[u, "\tau_1"'] \arrow[r, equal] & D \arrow[u, "\tau_2"'] \arrow[r, equal] & \cdots
\end{tikzcd}$$

and is denoted Δ_α^τ. The limit of a chain α (denoted $lim\,\alpha$) may or may not exist, but if it does then it is unique up to isomorphism. $lim\,\alpha$ is an object for which there exists a cone Δ_α^τ (called a **limiting cone**) with vertex $lim\,\alpha$ having the following property. For each object D, and for each cone Δ_α^δ with vertex D, there exists a unique arrow g from D to $lim\,\alpha$ such that

$$\forall\, n \geq 0 \,.\; \delta_n \;=\; \tau_n \cdot g$$

(more precise definitions of cone and limit are given in [Ma71] p68). The first aim of this section is to show that the chain

$$<A_0,\alpha_0> \xleftarrow{f_0} <A_1,\alpha_1> \xleftarrow{f_1} <A_2,\alpha_2> \xleftarrow{f_2} \cdots$$

has the limit $<L,\beta>$ where

$$L \;=\; \{\, <x_0,x_1,\ldots> \mid \forall\, n \geq 0 \,.\, x_n \in A_n \text{ and } f_n(x_{n+1}) = x_n \,\}$$
$$\forall\, x,y \in L \,.\;\; \beta(x,y) \;=\; \inf\{\, n + \alpha_n(x_n,y_n) \mid n \geq 0 \,\}$$

It is easy to see that β is in fact an agreement. L is a subspace of the cartesian product of a countable number of metric spaces ([Kur1] pp.212-3). The product topology is a topology of infinite sequences x such that for each $n \geq 0$ $x_n \in A_n$. Each sequence X of points in the product converges to a point l if and only if for each $n \geq 0$ the sequence

$$(X_0)_n \;,\; (X_1)_n \;,\; (X_2)_n \;,\; \ldots$$

converges to l_n . The following theorem shows that β induces the product topology on L .

Theorem 1

For each $n \geq 0$ let $<A_n, \alpha_n>$ be a compact agreement space. Let $<PA, \beta>$ be the agreement space such that

$$PA \;=\; \{\ <x_0, x_1, \ldots> \mid \forall\, n \geq 0 \,.\; x_n \in A_n\ \}$$
$$\forall\, x\ y \in PA \,.\;\; \beta(x, y) \;=\; \inf\{\ n + \alpha_n(x_n, y_n) \mid n \geq 0\ \}$$

Then $<PA, \beta>$ is a cartesian product. More precisely we will show that a sequence X of points in PA converges to a point l if and only if for each $n \geq 0$ the sequence

$$(X_0)_n \,,\, (X_1)_n \,,\, (X_2)_n \,,\, \ldots$$

converges to l_n .

Proof:

Suppose firstly that X converges to l . Let $n \geq 0$. Then for each $k \geq 0$ there exists $m \geq 0$ such that

$$\forall\, j \geq m \,.\;\; \beta(X_j, l) \;\geq\; k + n$$

Thus

$$\forall\, j \geq m \,.\;\; \inf\{\ i + \alpha_i(\,(X_j)_i \,, l_i\,) \mid i \geq 0\ \} \;\geq\; k + n$$

Thus

$$\forall\, j \geq m \,.\;\; n \,+\, \alpha_n(\,(X_j)_n \,, l_n\,) \;\geq\; k + n$$

Thus

$$\forall\, j \geq m \,.\quad \alpha_n(\,(X_j)_n\,,l_n\,) \;\geq\; k$$

Thus

$$(X_0)_n\,,(X_1)_n\,,(X_2)_n\,,\;\ldots$$

converges to l_n .

Now for the converse argument. Suppose that X is a sequence of points in PA such that for some $l \in PA$ and for each $n \geq 0$ the sequence

$$(X_0)_n\,,(X_1)_n\,,(X_2)_n\,,\;\ldots$$

converges to l_n . Let $k \geq 0$. Let $m \geq 0$ be such that

$$\forall\, i < k \,.\, \forall\, j \geq m \,.\quad \alpha_i(\,(X_j)_i\,,l_i\,) \;\geq\; k$$

Then

$$\forall\, j \geq m \,.\;\; \inf\{\; i + \alpha_i(\,(X_j)_i\,,l_i\,) \mid \mathrm{i} \geq k \;\} \;\geq\; k$$

Thus

$$\forall\, j \geq m \,.\quad \beta(\,X_j\,,l) \;\geq\; k$$

Thus X converges to L .

$\Box$

To see that L is compact we firstly appeal to Tychonov's theorem, which states that the cartesian product of a countable number of compact spaces is compact ([Kur2] p.17). We show L to be compact by showing it to be complete. Suppose that X is a cauchy sequence of points in L , then for each $n \geq 0$ so is the sequence

$$(X_0)_n \ , \ (X_1)_n \ , \ (X_2)_n \ , \ \ldots$$

As L is a subspace of a compact product space, there exists a point l in the product such that for each $n \geq 0$ l_n is a limit for the above sequence. Thus

$$\lim_{m \to \infty} X_m = \langle l_0, l_1, \ldots \rangle$$

To show that $l \in L$ it must be shown that

$$\forall\, n \geq 0 \,.\ f_n(l_{n+1}) = l_n$$

For each $n \geq 0$

$$\begin{aligned} \mathrm{f}(\, l_{n+1}\,) &= f_n(\, \lim_{m \to \infty}(X_m)_{n+1}\,) \\ &= \lim_{m \to \infty} f_n(\, (X_m)_{n+1}\,) \\ &\qquad\qquad (\text{ as } f_n \text{ is continuous }) \\ &= \lim_{m \to \infty} (X_m)_n \\ &\qquad\qquad (\text{ as } X_m \in L\ ) \\ &= l_n \end{aligned}$$

Thus $l \in L$, and in consequence L is complete and so compact. Now we can complete our proof that L is a limit of of the chain

$$\langle A_0, \alpha_0 \rangle \xleftarrow{f_0} \langle A_1, \alpha_1 \rangle \xleftarrow{f_1} \langle A_2, \alpha_2 \rangle \xleftarrow{f_2} \ldots$$

More precisely we will construct a limiting cone $\Delta \mathcal{J}$ with vertex L. For each $n \geq 0$ let τ_n:L$\rightarrow A_n$ be the function such that

$$\forall\, x \in L \,.\ \tau_n(x) = x_n$$

Each τ_n is continuous as for every convergent sequence X of points in L

$$\left(\lim_{m \to \infty} X_m \right)_n = \lim_{m \to \infty} (X_m)_n$$

We now show that $\Delta\mathcal{J}$ is limiting. Suppose that $\Delta\mathcal{J}$ is a cone with a vertex D. Then there is a unique function $g : D \to L$ such that

$$\forall\, n \geq 0 \,.\quad \delta_n = \tau_n \cdot g$$

that is, the function g such that

$$\forall\, d \in D \,.\ \forall\, n \geq 0 \,.\quad (g(d))_n = \delta_n(d)$$

It just remains to show that g is continuous. Let $d_0, d_1, \ldots$ be a convergent sequence of points in D. Then as each δ_n is continuous

$$\forall\, n \geq 0 \,.\quad \delta_n\left(\lim_{m \to \infty} d_m \right) = \lim_{m \to \infty} \delta_n(d_m)$$

Thus

$$\forall\, n \geq 0 \,.\quad \left(g\left(\lim_{m \to \infty} d_m \right)\right)_n = \lim_{m \to \infty} (g(d_m))_n$$

That is, for each $n \geq 0$ the sequence

$$(g(d_0))_n \,,\ (g(d_1))_n \,,\ (g(d_2))_n \,,\ \ldots$$

converges to

$$\left(g\left(\lim_{m \to \infty} d_m \right)\right)_n$$

Thus the sequence

$$g(d_0) \,,\ g(d_1) \,,\ g(d_2) \,,\ \ldots$$

converges to

$$g\left(\lim_{m \to \infty} d_m \right)$$

Thus

$$g\left(\lim_{m \to \infty} d_m\right) = \lim_{m \to \infty} g(d_m)$$

Thus g is continuous.

Section 3.4) : The Category FCAS

Now that CAS has been introduced we must consider the category **FCAS** of finite continuous functors over CAS. A functor F over CAS is a function which assigns to each object D in CAS an object $F(D)$, and to each arrow $f:D\to D'$ an arrow $F(f):F(D)\to F(D')$ such that,

$$\forall\ D \in CAS \qquad F(\,1_D\,) \;=\; 1_{F(D)}$$

$$\forall\ \ f:D\to D'\ \ g:D'\to D'' \qquad F(\,g\,.\,f\,) \;=\; F(g)\,.\,F(f)$$

An example of a functor is the powerspace functor $P()$ now described. The space $<D,\alpha>$ is a **subspace** of the space $<D',\alpha'>$ if $D\subseteq D'$, and if α is α' restricted to D. The compact subspaces of a compact space are the closed subsets, as each subset of a compact space is compact precisely when it is closed. [Kur1] (p.214) describes the metric space $(2^D)_m$ of all closed and bounded subsets of a space D with a metric m. If m is a compact metric induced by an agreement α then $(2^D)_m - \{\{\}\}$ is the space induced by the **powerspace** of compact subspaces of $<D,\alpha>$ (denoted $<P(D),P(\alpha)>$). The agreement $P(\alpha)$ for this powerspace is such that

$$\begin{aligned}\forall\ A\,,B \in P(D) \quad (P(\alpha))(\,A\,,B\,) \;=&\\ \inf\,\{\ \sup\{\,\alpha(x\,,y)\,|\,x \in A\,\}\ |\ y\in B\ \}&\\ \cup\ \{\ \sup\{\,\alpha(x\,,y)\,|\,y\in B\,\}\ |\ x\in A\ \}&\end{aligned}$$

The following theorem shows that $P(\alpha)$ is indeed an agreement.

Theorem 1

For each compact agreement space $<D,\alpha>$, $P(\alpha)$ is an agreement.

Proof:

Let $<D,\alpha>$ be a compact agreement space. Clearly $P(\alpha)$ is symmetric, also it is easy to see that

$$\forall\, A\,, B\,. \qquad A = B \quad ==> \quad (P(\alpha))(A,B) = \infty$$

As compact subspaces are complete the reverse implication is also clear. It only remains to prove the "triangle inequality"

$$\forall\, A\,, B\,, C\,. \quad (P(\alpha))(A,C) \geq \min\{\, (P(\alpha))(A,B)\,,\, (P(\alpha))(B,C)\,\}$$

Now,

$$\forall\, y \in B\,. \quad \sup\{\, \alpha(\, y\,, z\,) \mid z \in C\,\} \;\geq\; (P(\alpha))(B,C)$$

and,

$$\forall\, x \in A\,.\ \forall\, y \in B\,.\ \forall\, z \in C\,.\ \alpha(x,z) \geq \min\{\, \alpha(x,y)\,,\, \alpha(y,z)\,\}$$

thus,

$$\begin{aligned}\forall\, x \in A\,.\ \forall\, y \in B\,.\ &\sup\{\, \alpha(x,z) \mid z \in C\,\}\\ &\geq \min\{\, \alpha(x,y)\,,\, \sup\{\, \alpha(y,z) \mid z \in C\,\}\,\}\\ &\geq \min\{\, \alpha(x,y)\,,\, (P(\alpha))(B,C)\,\}\end{aligned}$$

Thus,

$$\begin{aligned}\forall\, x \in A\,.\ &\sup\{\, \alpha(x,z) \mid z \in C\,\}\\ &\geq \min\{\, \sup\{\, \alpha(x,y) \mid y \in B\,\}\,,\, (P(\alpha))(B,C)\,\}\end{aligned}$$

but,

$$\forall\, x \in A\,.\ \sup\{\, \alpha(x,y) \mid y \in B\,\} \geq (P(\alpha))(A,B)$$

thus,

$$\forall\, x \in A\,.\ \sup\{\, \alpha(x,z) \mid z \in C\,\}$$

$$\geq \min\{\, (P(\alpha))(A,B)\,,\, (P(\alpha))(B,C)\,\}$$

By similar argument,

$$\forall\, z \in C\,.\ \sup\{\, \alpha(z,x) \mid x \in A \,\}$$
$$\geq \min\{\, (P(\alpha))(C,B)\,,\, (P(\alpha))(B,A)\,\}$$

thus,

$$(P(\alpha))(A,C) \geq \min\{\, (P(\alpha))(A,B)\,,\, (P(\alpha))(B,C)\,\}$$

Thus we have proved that $P(\alpha)$ is an agreement.

□

To see that $P(D)$ is compact we consider the exponential topology 2^D of all closed subsets of D. As D is a compact metric space, 2^D is the same as $(2^D)_m$ ([Kur2] p.47). Also, 2^D is compact as D is compact ([Kur2] p.45), thus $(2^D)_m$ is compact. Thus as $\{\}$ is an isolated point in $(2^D)_m$, $(2^D)_m - \{\{\}\}$ is compact. That is, $P(D)$ is compact. $P()$ can be made a functor over CAS as follows. For each continuous function $f:D \to D'$, let $P(f):P(D) \to P(D')$ be the continuous function such that for each compact subspace A of D

$$(P(f))(A) \quad = \quad \{\, f(x) \mid x \in A \,\}$$

$(P(f))(A)$ is compact as the image of a compact space under a continuous function is compact ([Kur2] p.11). f is a closed mapping as it is continuous and as D is compact ([Kur2] p.11), thus $P(f)$ is continuous ([Kur1] p.165). $P()$ is a **finite functor**, that is, for each finite space A $P(A)$ is also finite. $P()$ is an example of a **continuous functor** as it maps each limiting cone to a limiting cone. A continuous functor is a generalisation of **chain complete function** as used by [Tar55] in his least fixed point theorem over partial orders. As an example, the following theorem shows that the powerspace functor $P()$ is continuous.

Theorem 2

$P()$ is a continuous functor.

Proof:

Let f be a chain

$$<A_0,\alpha_0> \xleftarrow{f_0} <A_1,\alpha_1> \xleftarrow{f_1} <A_2,\alpha_2> \xleftarrow{f_2} \cdots$$

in CAS. It has been shown earlier that f has a limit $<\beta,L>$ where

$$L = \{ <x_0,x_1,\ldots> \mid \forall n \geq 0 \;\; x_n \in A_n \;\; \& \;\; f_n(x_{n+1}) = x_n \}$$

$$\forall x\; y \in L \,.\;\; \beta(x,y) = \inf\{ n + \alpha_n(x_n,y_n) \mid n \geq 0 \}$$

$P(f)$ is the chain

$$P(A_0) \xleftarrow{P(f_0)} P(A_1) \xleftarrow{P(f_1)} P(A_2) \xleftarrow{P(f_2)} \cdots$$

Similarly $P(f)$ has a limit $<L',\beta'>$ where

$$L' = \{ <B_0,B_1,\ldots> \mid \forall n \geq 0 \;\; B_n \in P(A_n)$$

$$\& \;\; (P(f_n))(B_{n+1}) = B_n \}$$

$$\forall x\; y \in L' \;\; \beta'(x,y) = \inf\{ n + (P(\alpha_n))(x_n,y_n) \mid n \geq 0 \}$$

Let $i{:}P(L) \rightarrow L'$ be the function such that

$$\forall C \in P(L) \;\; \forall n \geq 0 \;\; (i(C))_n = \{ x_n \mid x \in C \}$$

To show that the functor $P()$ is continuous in CAS it is sufficient to show that the function i is an isomorphism. That is, it must be shown that i is a homeomorphism. But by [Kur2] (pp.11-12) each injective continuous mapping of a compact agreement space into an agreement space is a homeomorphism. Thus as $P(L)$ is compact, it is sufficient to show that i is an injective surjective continuous function. We begin firstly by showing that i is injective. Suppose that $C, C' \in P(L)$. Let $x \in C$. Then there exists a sequence $Y_0, Y_1, \ldots$ in C' such that

$$\forall\, n \geq 0 \quad x_n \;=\; (Y_n)_n$$

Thus

$$\forall\, n \geq 0 \;\; \forall\, m \leq n \quad x_m \;=\; (Y_n)_m$$

Thus

$$\forall\, n \geq 0 \quad \beta(x\,,Y_n\,) \;\geq\; n$$

Thus

$$x \;=\; \lim_{n \to \infty} Y_n$$

Therefore $x \in C'$ as each $Y_n \in C'$ and as C' is complete. Hence $C \subseteq C'$. By similar argument $C' \subseteq C$, thus $C = C'$. Thus i is injective.

We now show that i is surjective. Suppose that $B \in L'$. Let C be

$$\{\ \langle x_0, x_1, \ldots \rangle \mid \forall\, n \geq 0 \;\; x_n \in B_n \;\; \& \;\; f_n(x_{n+1}) = x_n\ \}$$

Then $C \subseteq L$. To show that $C \in P(L)$ we must show that C is compact. As L is compact it is sufficient to show that C is complete. Suppose that $X_0, X_1, \ldots$ is a cauchy sequence in C. Then as L is compact, X converges to a point $l \in L$. We need to show that

$l \in C$, that is we need to show that

$$\forall\, n \geq 0 \qquad f_n(l_{n+1}) \;=\; l_n$$

This is easily seen as for each $n \geq 0$

$$\begin{aligned}
f_n(l_{n+1}) \;&=\; f_n(\lim_{m \to \infty} (X_m)_{n+1}) \\
&=\; \lim_{m \to \infty} f_n((X_m)_{n+1}) \\
&\qquad\qquad (\text{ as } f_n \text{ is continuous }) \\
&=\; \lim_{m \to \infty} (X_m)_n \\
&\qquad\qquad (\text{ as } X_m \in L\ ) \\
&=\; (\lim_{m \to \infty} X_m\ )_n \\
&=\; l_n
\end{aligned}$$

Thus C is complete, and so compact. Thus $C \in P(L)$. Thus i is surjective, as by inspection $i(C) = B$.

Next we show that i is continuous. To do this it is sufficient to show that

$$\forall\, C\,,C' \in P(L) \quad \beta'(\,i(C)\,,i(C')\,) \;\geq\; (P(\beta))(\,C\,,C'\,)$$

Now, for all $C\,,C' \in P(L)$

$$\begin{aligned}
&(P(\beta))(\,C\,,C'\,) \\
&= \inf\{\ \sup\{\,\beta(x,y)\,|\,x \in C\,\}\ |\ y \in C'\ \} \\
&\qquad \cup \{\ \sup\{\,\beta(x,y)\,|\,y \in C'\,\}\ |\ x \in C\ \} \\
&= \inf\{\,\sup\{\,\inf\{\,n + \alpha_n(x_n,y_n)\,|\,n \geq 0\,\}\,x \in C\,\}\,|\,y \in C'\ \} \\
&\qquad \cup \{\,\sup\{\,\inf\{\,n + \alpha_n(x_n,y_n)\,|\,n \geq 0\,\}\,y \in C'\,\}\,|\,x \in C\,\}
\end{aligned}$$

Thus for all $C\,,C' \in P(L)$, and for all $n \geq 0$

$$\inf\{\ \sup\{\ n + \alpha_n(x_n, y_n) \mid x \in C\ \} \mid y \in C\ \}$$
$$\cup \{\ \sup\{\ n + \alpha_n(x_n, y_n) \mid y \in C\ \} \mid x \in C\ \}$$
$$\geq\ (P(\beta))(\ C\ ,\ C\ )$$

But for all $C\ ,\ C \in P(L)$

$$\beta'(\ i(C)\ ,\ i(C\ )\ )$$
$$=\ \inf\{\ n\ +\ (P(\alpha_n))(\ (i(C))_n\ ,\ (i(C\ ))_n\ )\ \mid\ n \geq 0\ \}$$
$$=\ \inf\{\ n + \inf\{\ \sup\{\ \alpha_n(u, v) \mid u \in (i(C))_n\ \} \mid v \in (i(C\ ))_n\ \}$$
$$\cup \{\ \sup\{\ \alpha_n(u, v) \mid v \in (i(C\ ))_n\ \} \mid u \in (i(C))_n\ \} \mid\ n \geq 0\ \}$$
$$=\ \inf\{\ n + \inf\{\ \sup\{\ \alpha_n(x_n, y_n) \mid x \in C\ \} \mid y \in C\ \}$$
$$\cup \{\ \sup\{\ \alpha_n(x_n, y_n) \mid y \in C\ \} \mid x \in C\ \}\ \mid\ n \geq 0\ \}$$
$$=\ \inf\{\ \inf\{\ \sup\{\ n + \alpha_n(x_n, y_n) \mid x \in C\ \} \mid y \in C\ \}$$
$$\cup \{\ \sup\{\ n + \alpha_n(x_n, y_n) \mid y \in C\ \} \mid x \in C\ \} \mid n \geq 0\ \}$$

Thus

$$\beta'(\ i(C)\ ,\ i(C\ )\ )\ \geq\ (P(\beta))(\ C\ ,\ C\ )$$

Thus i is continuous. We have now proved that i is an injective surjective continuous function, and thus that $P()$ is a continuous functor in CAS .

□

To conclude this section we show that all sums and products of continuous functors over CAS are continuous. First we look at sums. Suppose that F & G are continuous functors over CAS, and suppose that Δ_α^τ is a limiting cone with vertex V. Then as F & G are continuous the cones $F(\Delta_\alpha^\tau)$ and $G(\Delta_\alpha^\tau)$ are both limiting. Suppose we have a cone $\Delta^\delta_{(F+G)(\alpha)}$ with vertex D. As D is compact, and as δ_0 is a continuous function, there exists a unique partition of D into compact spaces D' & D'' such that the restrictions $\delta_0 | D'$ & $\delta_0 | D''$ are continuous and have the ranges $F(a_0)$ & $G(a_0)$ (where a_0 is the first object in α). Thus there exist unique cones $\Delta\delta'_{F(\alpha)}$ & $\Delta\delta''_{G(\alpha)}$ with vertices D' & D'' respectively which are restrictions of the cones $F(\Delta^\tau_\alpha)$ & $G(\Delta^\tau_\alpha)$. But, as $F(\Delta^\tau_\alpha)$ and $G(\Delta^\tau_\alpha)$ are limiting, there exist unique continuous functions $g' : D' \to F(V)$ & $g'' : D'' \to G(V)$ such that

$$\forall\, n \geq 0 \quad \delta'_n = F(\tau_n) \,.\, g' \quad \text{and} \quad \delta''_n = G(\tau_n) \,.\, g''$$

Thus there exists a unique continuous function $g : D \to (F+G)(V)$ such that

$$\forall\, n \geq 0 \quad \delta_n = (F+G)(\tau_n) \,.\, g$$

Thus $F + G$ is a continuous functor over CAS.

We now show that the product $F \times G$ is continuous. Let Δ^τ_α be a cone with vertex V as above. Then as F & G are continuous the cones $F(\Delta^\tau_\alpha)$ and $G(\Delta^\tau_\alpha)$ are both limiting. Suppose we have a cone $\Delta\delta_{(F\times G)(\alpha)}$ with vertex D. For each $n \geq 0$ let $\delta'_n : D \to F(a_n)$ and $\delta''_n : D \to G(a_n)$ be the left and right components of δ_n (where a_n is the n'th object in α). Then there exist cones $\Delta\delta'_{F(\alpha)}$ & $\Delta\delta''_{G(\alpha)}$ both with vertex D. But, as $F(\Delta^\tau_\alpha)$ & $G(\Delta^\tau_\alpha)$ are both limiting, there exist unique continuous functions $g' : D \to F(V)$ & $g'' : D \to G(V)$ such that

$$\forall\, n \geq 0 \quad \delta'_n = F(\tau_n) \,.\, g' \quad \text{and} \quad \delta''_n = G(\tau_n) \,.\, g''$$

Thus there exists a unique continuous function $g : D \to (F\times G)(V)$ such that

$$\forall\, n \geq 0 \quad \delta_n = (F\times G)(\tau_n) \,.\, g$$

Thus $F \times G$ is a continuous functor over CAS.

Section 3.5) : Recursive Equations over CAS

Now an **initial fixed point** semantics over CAS is given to equations of the form

$$D \equiv F(D)$$

for the class of continuous functors F. To formalise the notion of a "fixed point" for a continuous functor F we introduce below the category **ICAS(F)** of all isomorphisms $i:D \to F(D)$ (fixed points) of F. The aim of this section is to show that an object is **initial** (see below) in $ICAS(F)$ precisely when it is isomorphic to the fixed point constructed by the **inverse limit method** (see below). This method for solving recursive equations in category theory is a generalisation of Tarski's method for cpos [Tar55]. The approach is similar to that envisaged by, among others, Daniel Lehmann in this Ph.D. dissertation (University of Jerusalem). This section thus has two parts. The first part constructs a fixed point to the above equation using the inverse limit method. The second part defines the category $ICAS(F)$, and then shows that an object is initial precisely when it is isomorphic to our constructed fixed point.

We begin the first part of this section by describing the inverse limit method for continuous functors. A continuous functor F maps each limiting cone to a limiting cone. Thus for each such F, if the cone

$$\begin{tikzcd}
I & F(I) \arrow[l, "\tau"'] & F(F(I)) \arrow[l, "F(\tau)"'] & \cdots \arrow[l, "F(F(\tau))"'] \\
D \arrow[u, "\tau_0"'] \arrow[r, equal] & D \arrow[u, "\tau_1"'] \arrow[r, equal] & D \arrow[u, "\tau_2"'] \arrow[r, equal] & \cdots
\end{tikzcd}$$

is limiting, then so is the cone

$$\begin{array}{ccccccc}
F(I) & \xleftarrow{F(\tau)} & F(F(I)) & \xleftarrow{F(F(\tau))} & F(F(F(I))) & \xleftarrow{F(F(F(\tau)))} & \cdots \\
\Big\uparrow{\scriptstyle F(\tau_0)} & & \Big\uparrow{\scriptstyle F(\tau_1)} & & \Big\uparrow{\scriptstyle F(\tau_2)} & & \cdots \\
F(D) & = & F(D) & = & F(D) & = & \cdots
\end{array}$$

Thus this cone with the object I added to the base is a limiting cone. However, as the vertices of any two limiting cones with the same base must be unique (up to isomorphism), we have that

$$D \equiv F(D)$$

Consequently we can use limits to find a fixed point of a continuous functor F if we can prove that the chain (denoted ωF)

$$I \xleftarrow{\tau} F(I) \xleftarrow{F(\tau)} F(F(I)) \xleftarrow{F(F(\tau))} \cdots$$

has a limit. In the remainder of the first part of this section we will show that if F is finite continuous then ωF does in fact have a limit.

First let us look at the "retraction" properties of the arrows in ωF. A **retraction** for the arrow $f : A \to B$ in CAS is an arrow $\tau : B \to A$ such that

$$\tau \, . \, f \;=\; 1_A$$

Such an arrow f is always monic (see [Ma71] p19), thus in CAS f is a topological embedding of A into B. The arrows in ωF are all retractions, as for any arrow $e : I \to F(I)$

$$\begin{aligned}
\tau \, . \, e &= 1_I \\
F(\tau) \, . \, F(e) &= 1_{F(I)} \\
F(F(\tau)) \, . \, F(F(e)) &= 1_{F(F(I))} \\
& \cdots\cdots
\end{aligned}$$

Consequently the objects in ωF form an increasing sequence of finite sub-spaces of CAS. This enables us to use the usual **inverse limit** construction to find a limit. Let S be the agreement space of all infinite sequences x such that

$$\forall\, n \geq 0 \quad x_n \in F^n(I) \text{ and } x_n = (F^n(\tau))(x_{n+1})$$

S is clearly complete, and can be shown to be totally bounded as follows. A basis for S is

$$\{ \ \{ \ y \in S \mid \alpha(x,y) \geq n \ \} \mid x \in S \text{ and } n \geq 0 \ \}$$

Thus by [Suth] (p.111) S is totally bounded. Thus by [Sim] (p.125) S is compact, and so it is shown that $S \in CAS$.

For each $n \geq 0$ let $\tau_n : S \to F^n(I)$ be such that

$$\forall\, x \in S \quad \tau_n(x) = x_n$$

Then for each $m < n$

$$\tau_m = F^m(\tau) \,.\, F^{m+1}(\tau) \,.\, \cdots \,.\, F^{n-1}(\tau) \,.\, \tau_n$$

Then the diagram

$$\begin{tikzcd}
I & F(I) \arrow[l, "\tau"'] & F(F(I)) \arrow[l, "F(\tau)"'] & \cdots \arrow[l, "F(F(\tau))"'] \\
\bar{S} \arrow[u, "\tau_0"'] \arrow[r, phantom, "="] & \bar{S} \arrow[u, "\tau_1"'] \arrow[r, phantom, "="] & \bar{S} \arrow[u, "\tau_2"'] \arrow[r, phantom, "="] & \cdots
\end{tikzcd}$$

commutes, and is thus the cone $\Delta_{\omega F}^{I}$. To show that this cone is limiting, we must show that for each cone $\Delta^{\delta}_{\omega F}$ with base D, there exists a unique continuous map $g : D \to \overline{S}$ such that

$$\forall\, n \geq 0 \quad \delta_n \;=\; \tau_n \,.\, g$$

We will show that such a g exists. The only g satisfying the above equation is the g such that

$$\forall\, x \in D \quad \forall\, n \geq 0 \quad (g\,(x\,))_n \;=\; \delta_n\,(x\,)$$

It only remains to prove that g is continuous.

Suppose that y is a convergent sequence in D, then for each $n \geq 0$

$$\begin{aligned}
(g\,(\lim y\,))_n &= \delta_n\,(\lim y\,) \\
&= \lim_{m \to \infty} \delta_n\,(y_m) \\
&\qquad\qquad \text{(as } \delta_n \text{ is continuous)} \\
&= \lim_{m \to \infty} (g\,(y_m))_n
\end{aligned}$$

But for each $n \geq 0$, as $F^n(I)$ is finite there exists an integer $i_n \geq 0$ such that

$$\forall\, m \geq i_n \quad (g\,(y_m))_n \;=\; (g\,(y_{i_n}))_n$$

and

$$\lim_{m \to \infty} (g\,(y_m))_n \;=\; (g\,(y_{i_n}))_n$$

thus there exists $\lim\limits_{m \to \infty} g\,(y_m)$, and

$$\forall\, n \geq 0 \quad (\lim_{m \to \infty} g\,(y_m))_n \;=\; (g\,(y_{i_n}))$$

therefore

$$\lim_{m \to \infty} g\,(y_m) \;=\; g\,(\lim_{m \to \infty} y_m\,)$$

Thus g is continuous. We conclude the first part of this section by summarising its results. For each finite continuous functor F over CAS, it has been shown that the chain

$$I \xleftarrow{\ r\ } F(I) \xleftarrow{\ F(r)\ } F(F(I)) \xleftarrow{\ F(F(r))\ } \cdots$$

(denoted ωF) has a limit which is a fixed point of F.

Now we go on to define the category $ICAS(F)$, and to show that an object is initial in this category precisely when it is the limit constructed above. The objects of $ICAS(F)$ are the isomorphisms $i{:}A \to F(A)$ in CAS. For all such isomorphisms $i{:}A \to F(A)$ & $j{:}B \to F(B)$, and for each arrow $f{:}A \to B$ in CAS, the triple $<j, f, i>$ is an arrow in $ICAS(F)$ iff

$$F(f) \,.\, i \quad = \quad j \,.\, f$$

that is, if the following diagram commutes.

$$\begin{tikzcd} B \arrow[r, "j"] & F(B) \\ A \arrow[u, "f"] \arrow[r, "i"'] & F(A) \arrow[u, "F(f)"'] \end{tikzcd}$$

Composition in $ICAS(F)$ is similar to that in CAS. The composition of any two arrows $<k, g, j>$ & $<j, f, i>$ is the arrow $<k, g.f, i>$. For each isomorphism $i{:}A \to F(A)$ in CAS, an identity in $ICAS(F)$ is an arrow of the form $<i, 1_A, i>$. The isomorphisms in $ICAS(F)$ are those arrows $<j, f, i>$ for which f is an isomorphism in CAS. Although $ICAS(F)$ is not a subcategory of CAS, there is a forgetful functor which reduces $ICAS(F)$ to a **fixed point** subcategory of CAS. This is the functor which maps each object $i{:}A \to F(A)$ in $ICAS(F)$ to A, and maps each arrow $<j, f, i>$ of $ICAS(F)$ to f.

In the remainder of this section it is shown that for any object B in CAS, an object $j{:}B \to F(B)$ is **initial** in $ICAS(F)$ iff there exists a limiting cone $\Delta^{\tau}_{\omega F}$ with vertex B such that

$$\forall\, n > 0 \quad \tau_n \;=\; F(\tau_{n-1}) \,.\, j$$

The following has been shown for each finite continuous functor F over CAS. There exists an isomorphism $j{:}B \to F(B)$ such that B is the vertex of a limiting cone $\Delta^{\tau}_{\omega F}$. In the second part of this section it will first be shown (in Lemma 1) that j is an initial object in $ICAS(F)$. Secondly (Lemmas 2 & 3), it will be shown that for any object B in CAS, an isomorphism $j{:}B \to F(B)$ is initial in $ICAS(F)$ if and only if there exists a limiting cone $\Delta^{\tau}_{\omega F}$ with vertex B such that

$$\forall\, n > 0 \quad \tau_n \;=\; F(\tau_{n-1}) \,.\, j$$

Lemma 1

From above there exists an object B which is the vertex of a limiting cone $\Delta^{\tau}_{\omega F}$. Then there exists an isomorphism $j{:}B \to F(B)$ which is an initial object in $ICAS(F)$.

Proof:

As F is continuous, the cone

$$\begin{tikzcd}
I & F(I) \arrow[l, "\tau"'] & F(F(I)) \arrow[l, "F(\tau)"'] & \cdots \arrow[l, "F(F(\tau))"'] \\
F(B) \arrow[u, "s"'] \arrow[r, equal] & F(B) \arrow[u, "F(\tau_0)"'] \arrow[r, equal] & F(B) \arrow[u, "F(\tau_1)"'] \arrow[r, equal] & \cdots \arrow[u, phantom, "\cdots"]
\end{tikzcd}$$

is limiting. Thus B and $F(B)$ are vertices of limiting cones with the same base. Thus there exists an isomorphism $j:B \to F(B)$ such that

$$\tau_0 = s \, . \, j$$

$$\text{and, } \forall\, n > 0 \quad \tau_n = F(\tau_{n-1}) \, . \, j$$

To show that j is initial in $ICAS(F)$, we must show that for each isomorphism $i:A \to F(A)$ in CAS, there exists precisely one arrow $f:A \to B$ in CAS such that $<j,f,i>$ is an arrow in $ICAS(F)$. First we will prove that such an arrow $<j,f,i>$ exists, and then that it is unique.

Let $i:A \to F(A)$ be an isomorphism in CAS. Let $\Delta^{\delta}_{\omega F}$ be the unique cone with vertex A such that

$$\forall\, n > 0 \quad \delta_n = F(\delta_{n-1}) \, . \, i$$

As $\Delta^{\tau}_{\omega F}$ is a limiting cone there exists a unique arrow $f:A \to B$ such that

$$\forall\, n \geq 0 \quad \delta_n = \tau_n \, . \, f$$

Similarly for the above diagram, there exists a unique arrow $g:A \to F(B)$ such that

$$\delta_0 = s \, . \, g$$

$$\text{and, } \forall\, n > 0 \quad \delta_n = F(\tau_{n-1}) \, . \, g$$

But,

$$\delta_0 = s \,.\, (j \,.\, f)$$

and, $\delta_0 = s \,.\, (F(f) \,.\, i)$

and, $\forall\, n > 0 \quad \delta_n = F(\tau_{n-1}) \,.\, (j \,.\, f)$

$\qquad$ (as, $\delta_n = \tau_n \,.\, f$

$\qquad$ and, $\tau_n = F(\tau_{n-1}) \,.\, j$)

and, $\forall\, n > 0 \quad \delta_n = F(\tau_{n-1}) \,.\, (F(f) \,.\, i)$

$\qquad$ (as, $\delta_n = F(\delta_{n-1}) \,.\, i$

$\qquad$ and, $\delta_{n-1} = \tau_{n-1} \,.\, f$)

Thus as g is unique,

$$g = j \,.\, f = F(f) \,.\, i$$

Thus $<j,f,i>$ is an arrow in $ICAS(F)$. We must now show that there is no other arrow $f':A \to B$ in CAS such that $<j,f',i>$ is an arrow in $ICAS(F)$. Remembering from above that f is the unique arrow with the property

$$\forall\, n \geq 0 \quad \delta_n = \tau_n \,.\, f$$

we will show that f' has this property if $<j,f',i>$ is an arrow in $ICAS(F)$, and thus that $f = f'$. The proof is by induction on n.

Trivially, $\delta_0 = \tau_0 \,.\, f'$

Suppose we have shown that $\delta_{n-1} = \tau_{n-1} \,.\, f'$, then

$$\delta_n = F(\delta_{n-1}) \,.\, i$$

(by definition of δ_n)

$$= F(\tau_{n-1} \,.\, f') \,.\, i$$

(by induction hypothesis)

$$= (F(\tau_{n-1}) \,.\, F(f')) \,.\, i$$

$$= F(\tau_{n-1}) \,.\, (F(f') \,.\, i)$$

$$= F(\tau_{n-1}) \,.\, (j \,.\, f')$$

(as $<j, f', i>$ is an arrow in $ICAS(F)$)

$$= \ (F(\tau_{n-1}) \cdot j) \ . \ f'$$

$$= \ \tau_n \ . \ f'$$

(by definition of τ_n)

Thus,

$$\forall\, n \geq 0 \quad \delta_n \ = \ \tau_n \ . \ f'$$

Therefore $f = f'$, completing the proof of Lemma 1.

Lemma 2

For each initial object $i{:}A \rightarrow F(A)$ of $ICAS(F)$ there exists a limiting cone $\Delta^{\delta}_{\omega F}$ with vertex A such that

$$\forall\, n > 0 \quad \delta_n \ = \ F(\delta_{n-1}) \ . \ i$$

Proof:

Let $i{:}A \rightarrow F(A)$ be an initial object in $ICAS(F)$.

Let $j{:}B \rightarrow F(B)$ and $\Delta^{\tau}_{\omega F}$ be constructed as in the proof of Lemma 1.

Then B is a limit of ωF , j is an initial object of $ICAS(F)$, and

$$\forall\, n > 0 \quad \tau_n \ = \ F(\tau_{n-1}) \ . \ j$$

Initial objects in any category are isomorphic, thus there exists an arrow $f{:}A \rightarrow B$ in CAS such that $<j, f, i>$ is an isomorphism in $ICAS(F)$. Therefore f is an isomorphism in CAS.

Let $\Delta^{\delta}_{\omega F}$ be the unique cone such that

$$\forall n > 0 \quad \delta_n = \tau_n \, . \, f$$

This cone is limiting as $\Delta_{\omega F}^{\tau}$ so is, and as f is an isomorphism (limits of chains are unique up to isomorphism). Then for each $n \geq 0$

$$\begin{aligned} \delta_n &= \tau_n \, . \, f \\ &= (F(\tau_{n-1}) \, . \, j \,) \, . \, f \\ &\qquad (\text{ by definition of } \tau_n \,) \\ &= F(\tau_{n-1}) \, . \, (j \, . \, f \,) \\ &= F(\tau_{n-1}) \, . \, (F(f) \, . \, i \,) \\ &\qquad (\text{ as } <j, f, i> \text{ is an arrow in } ICAS(F) \,) \\ &= (F(\tau_{n-1}) \, . \, F(f) \,) \, . \, i \\ &= F(\tau_{n-1} \, . \, f \,) \, . \, i \\ &= F(\delta_{n-1}) \, . \, i \end{aligned}$$

Thus

$$\forall n > 0 \quad \delta_n = F(\delta_{n-1}) \, . \, i$$

This completes the proof of Lemma 2.

Lemma 3

Each isomorphism $i : A \rightarrow F(A)$ in CAS with a limiting cone $\Delta_{\omega F}^{\delta}$ such that

$$\forall n > 0 \quad \delta_n = F(\delta_{n-1}) \, . \, i$$

is an initial object in $ICAS(F)$.

Proof:

Let $i : A \rightarrow F(A)$ be an isomorphism in CAS.

Suppose that $\Delta^{\delta}_{\omega F}$ is a limiting cone such that

$$\forall\, n > 0 \quad \delta_n \;=\; F(\delta_{n-1}) \,.\, i$$

Let $j{:}B \to F(B)$ and $\Delta^{\tau}_{\omega F}$ be constructed as in the proof of Lemma 1.

As $\Delta^{\delta}_{\omega F}$ and $\Delta^{\tau}_{\omega F}$ are both limiting, there exists an isomorphism $f{:}A \to B$ such that

$$\forall\, n \geq 0 \quad \delta_n \;=\; \tau_n \,.\, f$$

By an argument similar to that in the proof of Lemma 1 we can show that

$$j \,.\, f \;=\; F(f) \,.\, i$$

Thus as f is an isomorphism in CAS, $<j, f, i>$ is an isomorphism in $ICAS(F)$. Thus as j is initial in $ICAS(F)$, and as initial objects in a category are unique up to isomorphism, i is initial.

This completes the proof of Lemma 3, and thus completes our proof that initial objects exist in $ICAS(F)$, and that they are precisely the isomorphisms $j{:}B \to F(B)$ for which there exists a limiting cone $\Delta^{\tau}_{\omega F}$ with vertex B such that

$$\forall\, n > 0 \quad \tau_n \;=\; F(\tau_{n-1}) \,.\, j$$

Section 3.6) : Solutions to Example Equations

In this section we give some example equations together with their solutions constructed via the inverse limit method.

The first example is an equation to generate the natural numbers with infinity.

$$X \equiv I + X$$

The solution is the compact space $< INT , \alpha_{INT} >$ where

$$INT = \{ 0 , 1 , 2 , \cdots \} \cup \{ \infty \}$$

$$\forall \ x \ y \in INT \ . \quad x < y \quad ==> \quad \alpha_{INT}(x , y) = x$$

Topologically, INT is the denumerable space with precisely one limit point.

The second example is one to generate the space SEQ of all finite & infinite sequences over an arbitrary space $<B,\beta>$.

$$X \equiv I + B \times X$$

The solution generated by the inverse limit method is $< SEQ , \alpha_{SEQ} >$ where

$$\forall \ x \ y \in SEQ \ . \quad \alpha_{SEQ}(x , y) =$$

$$\inf\{ \ n + \beta(x_n , y_n) \mid n < |x| \ \} \qquad \text{if } |x| = |y|$$

$$\inf\{ \ n + \beta(x_n , y_n) \mid n < |x| \ \} \cup \{ \ |x| \ \} \quad \text{if } |x| < |y|$$

For each $x \in SEQ$ $|x|$ denotes the length of x i.e.

$$| <> | = 0$$

$$| < x_0 , x_1 , \ldots , x_{n-1} > | = n$$

$$|<x_0, x_1, \ldots>| \quad = \quad \infty$$

Also note that by convention $\inf\{\} = \infty$. When B is a **flat** (that is a domain such that $\beta(x,y)=0$ precisely when $x \neq y$) α_{SEQ} becomes the **Baire agreement**. In this case for all distinct x & y in SEQ $\alpha_{SEQ}(x,y)$ is the length of the common initial segment of x and y.

Chapter 4

Function Domains for Completeness Rules

Section 4.1) : Introduction

Section 3.1) described the problem of choosing the right domain of functions in order to establish a Completeness Rule. The first problem in constructing any domain in this dissertation is deciding which objects in the domain deserve to be called "complete". A satisfactory answer to this question for function domains will take us half way to establishing a theory of Completeness Rules. This is because Completeness Rules are used to prove functions complete, just as the Cycle Product Test is used to show that equations have a complete (unique) solution. We propose here a simple function domain for investigating the completeness properties of functions. A function f is defined to be **complete** in this Chapter if for each complete object c , $f(c)$ is complete. This definition is unfortunately too simple, as we will see it needs refinement when discussing agreements upon functions. The function application combinator is shown to be complete, while the usual function composition combinator is shown to be partial. The failure of this composition to be complete appears to be the main stumbling block in constructing domains of functions. A new complete composition combinator is introduced. This discussion is carried out using a set theoretic model for domains. This model includes both the Kahn & Lucid Domains. A consequence of this Chapter will be that functions in a higher order theory of completeness should be **absolute**. The Conclusion of this Chapter discusses the implications of absolute functions and the problems with composition.

Section 4.2) : A Model For Domains

The examples of complete functions given so far have been all first order. Also they have been restricted to either the Kahn & Lucid Domains, or else domains of the like kind. It is possible to talk about the completeness of higher order functions such as function application, function composition, and fixed point functions. For example, function application is implemented in programming languages by a variety of "eager" and "lazy" techniques [He76]. These different techniques can be described by a variety of partial & complete objects in a function domain. To illustrate this we construct a simple theory of domains. A domain D in this theory is a collection of subsets of a set $\mathcal{D}$ partially ordered by the set inclusion relation $\supseteq$. The complete objects in these domains are the singletons, while the partial objects are the non-singletons. A more precise definition is given below. In this theory "eagerness" and "laziness" can be expressed in terms of completeness. Another interesting example for these domains is function composition. This corresponds to the way in which modules are "plugged" together. The way in which the plugging is done determines the efficiency of the resulting program, this efficency can be described using completeness. The final example for this theory of domains will be an examination of various fixed point combinators with respect to their completeness.

The construction of the domains themselves is straightforward.

Definition

A **domain** D with **universe** $\mathcal{D}$ is a partial order of subsets of $\mathcal{D}$ such that,

(i) The order $\sqsubseteq$ on D is set inclusion $\supseteq$

(ii) $\mathcal{D} \in D$

(iii) $\phi \notin D$

(iv) $\forall\ c \in \underline{D} \quad \{c\} \in D$

(v) All non-empty meets exist

(vi) D is chain complete

The operations $\sqcup$ and $\sqcap$ are not necessarily set union and set intersection, this is because domains do not have to include all non-empty sets. The Kahn Domain can be formalised as a domain D by letting $\underline{D}$ be the set of all infinite sequences in Ka , and by letting each non-singleton A in D be as follows. A corresponds to a finite sequence x in Ka , where A is the set of all infinite sequences having x as an initial segment. The Lucid Domain can be defined similarly in terms of these set theoretic domains.

Definition

The **complete** members of a domain are the singletons, while the **partial** objects are the non-singletons.

The following properties of a domain may be noted.

(i) Each member of a domain is either complete or else has a complete member above it. This is because each and every singleton is in the domain.

(ii) Each member of a domain is the meet of the maximal members above it. This is because,

$$\forall A \in D \quad A = \bigcup \{ \{a\} \mid a \in A \}$$

These domains are examples of posets which we shall consider later called

acpos, however, for the moment the only domains to be considered are these set theoretic ones. The notion of "set" quite naturally models the idea that a partial object should correspond to a collection of complete objects. Simple operations such as the cartesian product $D \times E$ of two universes can quite easily be extended to a **product** $D^{*}E$ of two domains.

Definition

The **product** of domains D & E is the domain

$$D^{*}E \;=\; \{\; A \times B \mid A \in D \text{ and } B \in E \;\}$$

$D^{*}E$ is a domain with universe $D \times E$. The complete objects in $D^{*}E$ are the singletons $\{<d, e>\}$ for $d \in D$ and $e \in E$. This fits with the intuition that the complete objects of a product should be formed from the complete objects of the component domains. The meet of a set of members

$$\{\; A_i \times B_i \;:\; i \in I \;\}$$

of $D^{*}E$ is

$$<\; \sqcap\{A_i : i \in I\} \;,\; \sqcap\{B_i : i \in I\} \;>$$

Also there is a one-one correspondence between $D^{*}E$ and $D \times E$ given by the interpretation function $\downarrow \in ((D^{*}E \to (D \times E))$ where

$$\forall\, A \in D \;\forall\, B \in E \quad \downarrow(A \times B) \;=\; <A, B>$$

At this point we can justify the omission of the empty set ϕ from domains. If ϕ was allowed in domains then $\downarrow$ could not be defined as, for example, $\{\phi\}$ would be a domain, and so

$$\{\phi\} * \{\{3\}\} = \{\{3\}\} * \{\phi\}$$

This omission is desirable as we do not want "overdefined" (i.e. "overcomplete") objects in domains. The claim that $*$ is an extension of $\times$ is justified by the fact that $\perp_D \times \perp_E$ is always a member of $D*E$.

The **disjoint sum** $\perp_D + \perp_E$ of universes can also be extended easily to a **sum** $D\#E$ of domains.

Definition

The **disjoint sum** of sets A & B is the set

$$A+B = \{ <a,0> \mid a \in A \} \cup \{ <b,1> \mid b \in B \}$$

Definition

The **sum** of domains D & E is the set

$$\begin{aligned} D\#E = & \{ \{ <a,0> \mid a \in A \} \mid A \in D-\{\perp_D\} \} \cup \\ & \{ \{ <b,1> \mid b \in B \} \mid B \in E-\{\perp_E\} \} \cup \\ & \{ \perp_D + \perp_E \} \end{aligned}$$

This sum is the so called **coalesced sum**, where the bottom elements $\perp_D$ and $\perp_E$ are "merged" into the new bottom $\perp_D + \perp_E$. The universe of $D\#E$ is $\perp_D + \perp_E$. The claim that $\#$ is an extension of $+$ is justified by the fact that $\perp_D + \perp_E$ is always a member of $D\#E$.

Section 4.3) : Function Domains

The final operator on universes which we wish to extend is the following

Definition

The **function space** $A \to B$ is the space of all functions from the set A to the set B.

As with the previous domain constructions of product & sum, we conform to the intuition that a partial object should be the meet of the complete objects above it. A complete object in a function domain will be a singleton $\{f\}$ for f in a function space $D \to E$. In general, an object in the domain will thus be a non-empty set of such functions f.

Definition

The **function domain** $D \multimap E$ is the set of all non-empty subsets A of $D \to E$ such that

$$\text{(i)} \quad \forall c \in D \quad \{ f(c) \mid f \in A \} \in E$$

$$\text{(ii)} \quad A = \{ f \in D \to E \mid \forall c \in D \ \ f(c) \in \{ g(c) \mid g \in A \} \}$$

Using (i) we can map each complete member $\{c\}$ of D into an element of E. (ii) implies the following property. If two members of $D \multimap E$ agree on the complete points of D then they are identical.

It still has to be shown that $D \multimap E$ is in general a domain.

Theorem 1

For all domains D & E, $D \to E \in D \blacktriangleright E$

Proof:

(proof technically uninteresting)

□

Theorem 2

For all domains D & E, $D \blacktriangleright E$ is chain complete.

Proof:

(proof technically uninteresting)

□

The previous two theorems show that in general $D \blacktriangleright E$ is a domain. As with product and sum, there is the following property to show that $\blacktriangleright$ is an extension of $\to$.

$$\forall\, D, E \quad \{D\} \blacktriangleright \{E\} = \{D\} \to \{E\}$$

A function domain is no good unless its members can actually be interpreted as functions. This we can do via an **interpretation** function.

Definition

For all domains D & E, the **interpretation function** for $D \blacktriangleright E$ is the function $\downarrow \in ((D \blacktriangleright E) \to (D \to E))$ such that

$$\forall A \in D \blacktriangleright E \quad \forall B \in D$$

$$(\downarrow A)(B) \quad = \quad \sqcap \{ \{ f(c) \mid f \in A \} \mid c \in B \}$$

Absolute functions occur in languages using lazy evaluation [He76]. In such languages calls are made for only the minimum amount of input needed to produce the required output. Absolute functions embody the ultimate (or absolute) notion of laziness. This notion turns out to be lazier than that employed by present users of lazy evaluation. In other words, absolute functions can be regarded as specifications for current lazy implementations.

The interpretation function gives a one-one correspondence between the function domain $D \multimap E$ and the set of all **absolute** functions in $D \to E$.

Definition

A function $F \in (D \to E)$ over domains D & E is **absolute** if

$$\forall\ A \in D \qquad F(A) \ = \ \sqcap \{ \ F(\{c\}) \mid c \in A \ \}$$

The one-one correspondence between each domain $D \multimap E$ and the set of all absolute functions from D to E is established by the next theorem.

Theorem 3

For all domains D & E, the interpretation function $\downarrow$ for $D \multimap E$ is injective. Also, the range of $\downarrow$ is the set of all absolute functions in $D \to E$.

Proof:

Suppose $\downarrow : ((D \multimap E) \to (D \to E))$

is an interpretation function.

Then $\downarrow$ is injective as for each $A \in D \blacktriangleleft\!\!- E$

$$A \quad = \quad \{\ f \in D \to E \ \mid\ \forall\ c \in D\ .\ f(c) \in (\downarrow A)(\{c\})\ \}$$

Also $\downarrow$ is surjective as for each absolute function $F \in D \to E$

$$\downarrow(\ \{\ f \in D \to E \mid \forall\ c \in D\ .\ f(c) \in F(\{c\})\ \}\ ) \quad = \quad F$$

$\square$

The complete objects in a function domain $D \blacktriangleleft\!\!- E$ are, of course, the singletons $\{f\}$ for any f in $(D \to E)$. However, the interpretation $\downarrow$ for $D \blacktriangleleft\!\!- E$ puts them in one-one correspondence with the set of absolute functions in $(D \to E)$ which maps each complete object in D to a complete object in E. Thus a **complete function** in our theory is an absolute function which maps complete objects to complete objects. The product & function interpretations can be combined to obtain interpretation functions such as

$$\downarrow\ \in\ (D \ast (D \blacktriangleleft\!\!- E)) \blacktriangleleft\!\!- E\ \to\ (D \times (D \to E)) \to E$$

For example, when the set

$$\{\mathrm{app}\}\ \in\ (D \ast (D \blacktriangleleft\!\!- E)) \blacktriangleleft\!\!- E$$

is interpreted, where app is the usual function application

$$\mathrm{app}\ \in\ (D \times (D \to E)) \to E$$

we get the function $\downarrow\{\mathrm{app}\}$ where for each A in D and absoulute F in $D \to E$

$$(\ \downarrow\{\mathrm{app}\}\ )(\ \langle A, F\rangle\ ) \quad = \quad F(A)$$

That is, {app} is interpreted as the usual function application. By similar argument we can interpret members of domains such as

$$((D \to E) * (E \to F)) \to (D \to F)$$

as members of

$$((D \to E) \times (E \to F)) \to (D \to F)$$

Thus if comp is the usual function composition in

$$((\underline{D} \to \underline{E}) \times (\underline{E} \to \underline{F})) \to (\underline{D} \to \underline{F})$$

then the interpretation of {comp} in the above domain is such that for all absolute functions F & G in $D \to E$ & $E \to F$ respectively

$$\forall B \in D \quad (\downarrow(\{\text{comp}\}) (<F,G>))(B) \quad = \quad \sqcap \{ \; G(F(\{c\})) \mid c \in B \; \}$$

This is not the following definition of function composition that one would expect

$$\text{dcomp}(<F,G>) \quad = \quad \lambda \; A \in D \; . \; G(F(A))$$

dcomp will not in general be absolute, but a weaker function as,

$$\forall A \in D \quad (\text{dcomp}(<F,G>))(A) \sqsubseteq (\downarrow(\{\text{comp}\})(<F,G>))(A)$$

An example of when this relation is not an equality is included in the proof of Theorem 1 of Section 5.2. However, the two functions $\downarrow(\{\text{comp}\})(<F,G>)$ and dcomp($<F,G>$) do agree on the complete members of D .

This suggests that $\downarrow(\{\text{comp}\})$ is to be regarded as a specification of the usual composition dcomp.

$\downarrow(\{\text{comp}\})$ does not share all the properties of dcomp, for example, dcomp is associative while $\downarrow(\{\text{comp}\})$ is not.

Section 4.4) : Conclusions

The first conclusion from this Chapter is that the usual function composition combinator cannot be easily extended from the complete objects to the remainder of the domain. We can in this model show what complete composition is, as well as complete application, as opposed to just speculating what they might be. Our version of composition can be regarded as a specification for all other forms of composition, an upper bound for which the usual form of composition is an approximation. Our specification does not have nice properties such as associativity. Is there a reason though why composition goes wrong? It is this question which needs an answer before completeness for function domains can really be understood.

A second conclusion from this Chapter is that absolute functions are the most natural functions to consider for complete objects in a function domain. While this claim is justified in our model it is not justified in functions used by many programming languages. This is firstly because the normal composition of absolute functions is not in general absolute. Secondly, it is because our complete composition is (to say the least) not easy to implement when the complete objects used by the functions are infinite data structures. However, having made these negative points about absolute functions we should remember that such functions are used in programming languages having lazy evaluation. Thus absolute functions, however difficult they may seem from the point of view of function composition, are used by programmers as specifications for the functions which they actually write. Hence this Chapter has provided an argument in support of Completeness Rules for lazy programming languages. By consequence, we have provided an argument in support of completeness in general.

Chapter 5

Absolute Functions

Section 5.1) : Introduction

Previous discussions in Chapter 4 lead to a conclusion that a theory of function domains for Completeness Rules using the naive set-theoretic approach necessitates a restriction to **absolute** functions. This section discusses the possibility of setting up categories of such functions, and thus considers the feasability of the naive approach. It is shown that categories of absolute functions (using o for morphism composition) do not have the basic categorical properties required for a theory of domains. This is achieved by restricting the allowed absolute functions, firstly in Section 5.2) to **super absolute** functions, and secondly in Section 5.3) to **meet preserving** functions. This is more difficult than the work of Chapter 3, as there we did not consider functions over domains, but only functions over complete objects. The conclusion of Sections 5.2 and 5.3 is that a reappraisal of the role of function composition is necessary to generalise completeness (note that this is consistent with the conclusion of Chapter 4). It is this conclusion which in Section 5.4 leads us to weaken the fundamental categorical axiom of composition associativity, the result being a category-like structure which we call a **mategory**, and which in its definition incorporates the notion of completeness.

Definition

A **cpo** is a chain-complete partially ordered set with a least element.

Definition

An **absolute cpo** (acpo) is a cpo such that

(i) the meet of each set of points exists.

(ii) each point has a maximal element above it.

(iii) each point is the meet of the maximal points above it.

Definition

For all acpos A & B, a function $f : A \to B$ is **absolute** if for each point x in A

$$f(x) = \sqcap \{ f(m) \mid x \sqsubseteq m \text{ and } m \text{ is maximal} \}$$

This then is the world we are interested in. This Chapter looks at categories of absolute functions over acpos. Do there exist products, sums or exponentiations in such categories? First it must be established as to whether or not any category of absolute functions exists at all.

Section 5.2) : Super Absolute Functions

Theorem 1

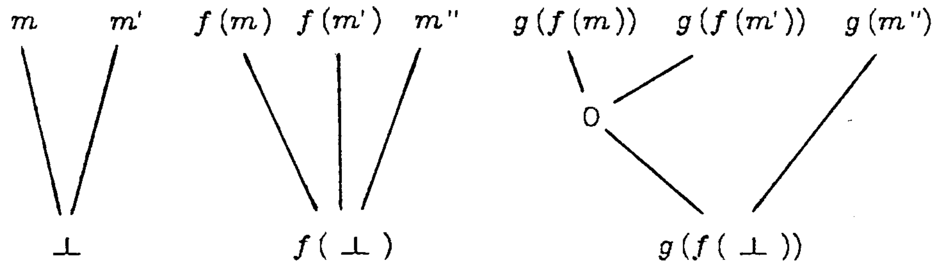
There exist two absolute functions whose composition is not absolute.

Proof:

Let A , B , & C be the acpos with

functions $f:A \to B$ and $g:B \to C$

given by the following diagrams.



f and g are absolute, however

$$g(f(\perp)) \neq \sqcap \{ g(f(m)) , g(f(m')) \}$$

Hence $g o f$ is not absolute.

□

Theorem 1 appears to down any hopes of setting up a category of cpos and absolute functions. This is not the case if a more restrictive notion of "absolute function" is used.

Definition

For all acpos A & B , an absolute function $f:A \to B$ is **super absolute** if for each x in A ,

for each maximal m in B, if $f(x) \sqsubseteq m$ then
there exists a maximal m' in A such that $x \sqsubseteq m'$
and $f(m') \sqsubseteq m$

The identity function is always super absolute as are many other absolute functions. The following function f is an example of a super absolute function.

$$f(<>) = <>$$

$$f(< x_0 , \dots , x_n >) = < x_1 , \dots , x_n >$$

$$f(< x_0 , x_1 , \dots >) = < x_1 , x_2 , \dots >$$

The next theorem will be needed to show that super absolute functions are closed under function composition.

Theorem 2

Each absolute function is monotonic.

Proof:

trivial.

□

Theorem 3

For each set Λ of subsets of an acpo

$$\sqcap (\bigcup \Lambda) = \sqcap \{ \sqcap A \mid A \in \Lambda \}$$

Proof:

Let Λ be a set of subsets of an acpo, and let

x be in that acpo. Then,

$$x \sqsubseteq \sqcap(\bigcup \Lambda) \iff \forall y \in \bigcup \Lambda \,.\, x \sqsubseteq y$$
$$\iff \forall A \in \Lambda \;\; \forall y \in A \,.\, x \sqsubseteq y$$
$$\iff \forall A \in \Lambda \,.\, x \sqsubseteq \sqcap A$$
$$\iff x \sqsubseteq \sqcap \{ \sqcap A \mid A \in \Lambda \}$$

thus, $\sqcap(\bigcup \Lambda) = \sqcap \{ \sqcap A \mid A \in \Lambda \}$

$\square$

Theorem 4

Super absolute functions are closed under function composition.

Proof:

Let A, B, & C be acpos.

Suppose that $f : A \to B$ and $g : B \to C$ are super absolute.

Now, for each x in A,

$g(f(x))$

$= \sqcap \{ g(m) \mid m$ is maximal, and $f(x) \sqsubseteq m \}$

(as g is absolute)

$= \sqcap \{ g(m) \mid m$ is maximal, and there exists m' such that

$x \sqsubseteq m'$ & $f(m') \sqsubseteq m \}$

(as f is super absolute)

$= \sqcap \{ \sqcap \{ g(m) \mid m$ is maximal and $f(m') \sqsubseteq m \}$

$\mid m'$ is maximal and $x \sqsubseteq m' \}$

(by Theorem 3)

$$= \quad \sqcap \{ \; g(f(m')) \mid m' \text{ is maximal, and } x \sqsubseteq m' \; \}$$

(as g is absolute)

Thus for each x in A ,

$$g(f(x)) \quad = \quad \sqcap \{ \; g(f(m')) \mid m' \text{ is maximal, and } x \sqsubseteq m' \; \} \quad (*)$$

Let x be in A , and suppose $m \in C$ is such that $g(f(x)) \sqsubseteq m$.

Suppose m' is maximal in B , and is such that

$f(x) \sqsubseteq m'$ and $g(m') \sqsubseteq m$

(m' exists as g is super absolute)

Suppose m'' is maximal in A , and is such that

$x \sqsubseteq m''$ and $f(m'') \sqsubseteq m'$

(m'' exists as f is super absolute, and as $f(x) \sqsubseteq m'$)

thus $g(f(m'')) \sqsubseteq g(m')$

(g is monotonic by Theorem 2)

thus $g(f(m'')) \sqsubseteq m$

thus $x \sqsubseteq m''$ and $g(f(m'')) \sqsubseteq m$ (as $g(f(x)) \sqsubseteq m$)

thus by (*) $g \circ f$ is absolute.

□

Using Theorem 4 a category of absolute functions can now be built. The objects of the category are precisely the acpos, while the arrows are precisely the super absolute functions (Theorem 5 proves that the identity morphisms exist in the category).

Theorem 5

The identity function on each acpo is super absolute.

Proof:

trivial

□

If this category is to be useful for solving domain equations then it must have, among other things, products. The following definition defines the notion of a categorical product.

Definition

A **product** of objects A & B in a category is a triple $< c\ , p_0 , p_1 >$ where C is an object, and $p_0{:}C \rightarrow A$ & $p_1{:}C \rightarrow B$ are morphisms such that for any morphisms $f{:}X \rightarrow A$ & $g{:}X \rightarrow B$ for some object X there exists a unique morphism $h{:}X \rightarrow C$ for which the following diagram commutes

$$\begin{tikzcd} & X \arrow[d, "h"] \arrow[ddl, bend right, "f"'] \arrow[ddr, bend left, "g"] & \\ & G \arrow[dl, "p_0"'] \arrow[dr, "p_1"] & \\ A & & B \end{tikzcd}$$

Notation

For all posets A & B, '$A \times B$' denotes the poset with universe

$$\{ <a,b> \mid a \in A \text{ and } b \in B \}$$

under the pointwise ordering.

Definition

The **cartesian product** of posets A & B is the triple $< A \times B, p_0, p_1 >$ where $p_0 : A \times B \to A$ and $p_1 : A \times B \to B$ are the obvious projection functions.

The following theorem suggests that cartesian products are categorical products.

Theorem 6

For any acpos A & B, $A \times B$ is an acpo. Also the projection functions $p_0 : A \times B \to A$ and $p_1 : A \times B \to B$ are super absolute.

Proof:

trivial

□

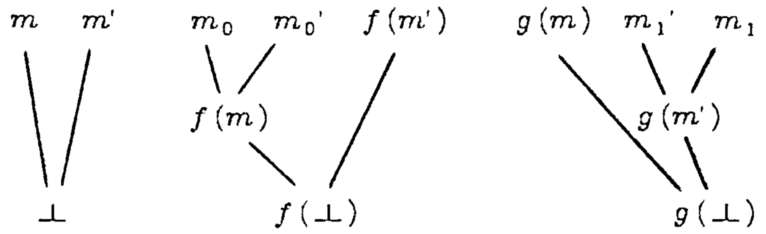
Unfortunately the following theorem rules out the cartesian product as a categorical product.

Theorem 7

There exist two acpos whose cartesian product is not a categorical product in the category of all acpos and super absolute functions.

Proof:

Let X, A, & B be the acpos, and let $f : X \rightarrow A$ & $g : X \rightarrow B$ be the super absolute functions given by the following diagrams.



Suppose that the cartesian product $<A \times B, p_0, p_1>$ of A & B is a product in the category of all acpos & super absolute functions.

Then there exists a unique super absolute function $h : X \rightarrow A \times B$ such that $p_0 o h = f$ and $p_1 o h = g$ and

$$\forall\, x \in X \;\; h(x) = <f(x), g(x)>$$

Thus from the definition of super absolute function,

as $<m_0, m_1>$ is maximal in $A \times B$, and as

$h(\perp) \sqsubseteq <m_0, m_1>$ we have that either,

$h(m) \sqsubseteq <m_0, m_1>$ or $h(m') \sqsubseteq <m_0, m_1>$

However, this is a contradiction as both

$h(m) \not\leq <m_0, m_1>$ and $h(m') \not\leq <m_0, m_1>$

Hence $<A \times B, p_0, p_1>$ is not a categorical product.

□

Theorem 7 does not deny the existence of categorical products in the category of acpos & super absolute functions. However, it does confirm that if they do exist then they are not what they are expected to be. This fact should be combined with the fact that there are interesting non super absolute functions such as $check : (Ka \times Ka) \to Ka$

$$check(<>) \quad = \quad <>$$

$$\forall \text{ non empty } x \qquad check(x) \quad = \quad <1>$$

This function outputs precisely one daton as soon as an input daton has arrived. Analysis of the proof of Theorem 4 reveals that the tightening up of absolute functions to super absolute functions is unnecessarily tight. The tightening is to ensure the following in that proof.

$$\{ \ m \ \mid m \text{ is maximal, and } \ f(x) \sqsubseteq m \ \}$$

$$= \ \{ \ m \ \mid m \text{ is maximal, and } \exists m' \ . \ m' \text{ is maximal, and}$$
$$x \sqsubseteq m \ \& \ f(m') \sqsubseteq m \ \}$$

Now we try loosening the tightening to just ensuring that the meets of these sets are equal.

Definition

A function $f : A \to B$ over acpos A & B is **meet preserving** if for each pair A' & A'' of sets of points in A

$$A' \subseteq A'' \quad \text{and} \quad \sqcap A'' = \sqcap A' \qquad ==>$$

$$\sqcap \{ \ f(a'') \mid a'' \in A'' \ \} \ = \ \sqcap \{ \ f(a') \mid a' \in A' \ \}$$

The aim is now to see whether a category of acpos & absolute meet preserving functions can be defined. Such a category should have at least products in order to improve on super absolute functions.

Section 5.3) : Meet Preserving Functions

This section shows that the category of all meet preserving functions over acpos has finite products. However, it is also shown that none of the popular "disjoint", "coalesced", or "separated" sums [St77] are categorical.

Theorem 1

Each meet preserving function is monotonic

Proof:

Let A and B be acpos.

Suppose that $f : A \to B$ is meet preserving.

Suppose that a , $a' \in A$ are such that $a \sqsubseteq a'$.

Then, $\sqcap\{ a \} = \sqcap\{ a , a' \}$

thus as f is meet preserving, $\sqcap\{ f(a) \} = \sqcap\{ f(a) , f(a') \}$

thus $f(a) \sqsubseteq f(a')$

□

Theorem 2

Each meet preserving function is absolute

Proof:

Let A and B be acpos.

Suppose that $f : A \to B$ is meet preserving.

Let $a \in A$.

Now, $\sqcap \{ m \mid m \text{ is maximal, and } a \sqsubseteq m \}$

$$= \sqcap \{ \{ m \mid m \text{ is maximal, and } a \sqsubseteq m \} \cup \{ a \} \}$$

(as a is the meet of the maximal objects above it)

Thus as f is meet preserving,

$$\sqcap \{ f(m) \mid m \text{ is maximal, and } a \sqsubseteq m \}$$
$$= \sqcap \{ \{ f(m) \mid m \text{ is maximal, and } a \sqsubseteq m \} \cup \{ f(a) \} \}$$

Thus, $\sqcap \{ f(m) \mid m \text{ is maximal, and } a \sqsubseteq m \} \sqsubseteq f(a)$

But as f is monotonic by Theorem 1, the reverse implication holds.

Thus $f(a) = \sqcap \{ f(m) \mid m \text{ is maximal, and } a \sqsubseteq m \}$

Thus f is absolute.

□

Theorem 3

The class of meet preserving functions is closed under function composition.

Proof:

Let A, B, & C be acpos.

Suppose $f : A \to B$ and $g : B \to C$ are meet preserving.

Suppose that A' & A'' are subsets of A such that,

$$A' \subseteq A'' \quad \text{and} \quad \sqcap A' = \sqcap A''$$

Then as f is meet preserving,

$$\sqcap \{ f(a'') \mid a'' \in A'' \} = \sqcap \{ f(a') \mid a' \in A' \}$$

thus as g is meet preserving,

$$\sqcap \{ g(f(a'')) \mid a'' \in A'' \} = \sqcap \{ g(f(a')) \mid a' \in A' \}$$

hence $g \circ f$ is meet preserving.

□

Noting that each identity function and constant function is meet preserving, Theorem 3 shows that a category of meet preserving functions can be built. Now to the question of whether this category has products.

Theorem 4

For all acpos A & B, and for each $CNE\phi \subseteq A \times B$,

$$\sqcap C = < \sqcap \{ a \in A \mid \exists b \in B. <a,b> \in C \} ,$$
$$\sqcap \{ b \in B \mid \exists a \in A. <a,b> \in C \} >$$

Proof:

trivial.

□

Theorem 5

For all acpos A & B with cartesian product $< A \times B , p_0 , p_1 >$, p_0 & p_1 are both meet preserving.

Proof:

trivial (using Theorem 4)

□

Theorem 6

In the category of meet preserving functions the cartesian product of each pair of acpos is a product.

Proof:

Let A, B, & C be acpos.

Let $< A \times B , p_0 , p_1 >$ be the cartesian product of A & B.

Then by Theorem 6 of Section 5.2), & Theorem 10, $A \times B$ is an acpo, also p_0 & p_1 are absolute & meet preserving. (i)

Let X be an acpo.

Suppose $f : X \to A$ & $g : X \to B$ are meet preserving.

Let $h : X \to A \times B$ be the unique function such that

$$p_0 \circ h = f \quad \text{and} \quad p_1 \circ h = g \qquad \text{(ii)}$$

Then, $\forall x \in X \quad h(x) = <f(x), g(x)>$

Now to show that h is meet preserving.

Suppose that $X' \subseteq X'' \subseteq X$ are such that $\sqcap X'' = \sqcap X$

then, $\sqcap \{ h(x'') \mid x'' \in X'' \}$

$= \sqcap \{ <f(x''), g(x'')> \mid x'' \in X'' \}$

$= < \sqcap \{ f(x'') \mid x'' \in X'' \} , \sqcap \{ g(x'') \mid x'' \in X'' \} >$

$= < \sqcap \{ f(x') \mid x' \in X' \} , \sqcap \{ g(x') \mid x' \in X' \} >$

(as f & g are meet preserving)

$= \sqcap \{ <f(x'), g(x')> \mid x' \in X' \}$

(by Theorem 9)

$$= \sqcap \{ h(x') \mid x' \in X \}$$

thus h is meet preserving.

Thus by (i) & (ii), $< A \times B , p_0 , p_1 >$ is a product in the category of all acpos and meet preserving functions.

□

To summarise the results so far in this chapter. We now have a category of all acpos with a subclass of absolute functions that has at least finite products. The only absolute functions allowed are those which are meet preserving. The next interesting question for this category is whether or not it has sums (co-products). There are three potential sums, disjoint, coalesced, and separated. The disoint sum of two acpos is not an acpo as it does not have a least member. Coalesced sums involve "joining" the two least members, something which would have worked if all functions in the category had the same value on the least member $\perp$. The remainder of this section shows that separated sums will not work either.

Definition

A **sum** of objects A & B in a category is a triple $< C , i_0 , i_1 >$ where C is an object, and $i_0 : A \to C$ and $i_1 : B \to C$ are morphisms satisfying the following condition. For any object X and morphism $f : A \to X$ & $g : B \to X$ there exists a unique morphism $h : C \to X$ for which the following diagram commutes.

$$
\begin{tikzcd}
A \arrow[dr, "i_0"] \arrow[ddr, bend right, "f"'] & & B \arrow[dl, "i_1"'] \arrow[ddl, bend left, "g"] \\
& C \arrow[d, "h"] & \\
& X &
\end{tikzcd}
$$

C is denoted by $A+B$, while h is denoted by $f+g$.

Definition

The **separated sum** of acpos A & B is the triple $< A \pm B , i_0 , i_1 >$ where $A \pm B$ denotes $A+B$ with a new least member, and where $i_0 : A \rightarrow A \pm B$ & $i_1 : B \rightarrow A \pm B$ are the usual injections.

Theorem 7

The injection functions of each separated sum are absolute and meet preserving.

Proof:

trivial.

$\Box$

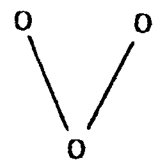
Unfortunately separated sums are not going to be categoical.

Theorem 8

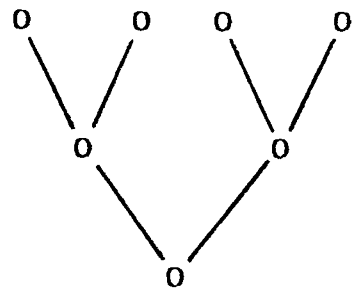
There exist acpos whose separated sum in the category of all acpos & absolute meet preserving functions is not categorical.

Proof:

Let A be the acpo



Then $A \pm A$ =



Let $id{:}A \to A$ be the identity function on A.

Let $h : (A \pm A) \to A$ be the unique absolute function such that the following diagram commutes.

$$
\begin{tikzcd}
A \arrow[dr, "i_0"] \arrow[ddr, "id"'] & & A \arrow[dl, "i_1"'] \arrow[ddl, "id"] \\
 & A \pm A \arrow[d, "h"] & \\
 & A &
\end{tikzcd}
$$

Then it can be verified that h is not meet preserving.

Hence $< A \pm A , i_0 , i_1 >$ is not a sum in the category of all acpos & meet preserving functions.

□

The overall conclusion from this chapter is that categories of meet preserving functions are worth considering because products do exist. However the lack of a suitable sum is a source of major concern. A category of strict meet preserving functions would make the coalesced sum categorical. This would rule out many necessary functions though, such as the constant valued functions. For this reason we are reluctant to consider meet preserving functions until other avenues such as that in Section 5.4) have been investigated.

Section 5.4) : Mategories

The previous two sections have throughly examined the possibility of constructing categories of absolute functions over acpos. The conclusion from those results is that absolute functions and category theory will not provide a general theory of domains which use the notion of completeness. However, the efforts of Chapter 4 to define the notion of a "complete function" forced our researches to consider absolute functions. The only parameter left open to us for reconsideration in that work is a change in the morphism composition operator. Instead of using o, we now suggest the **absolute composition** operator $|$, where for absolute functions f and g,

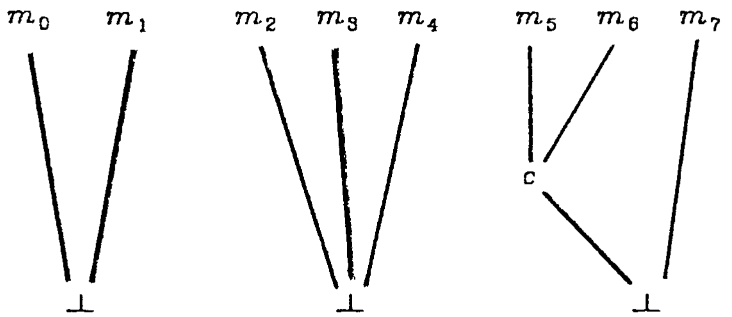
$$\forall x \quad (g \mid f)(x) = \sqcap \{ \, g(f(m)) : x \sqsubseteq m \text{ , and } m \text{ is maximal} \, \}$$

The trouble with o is that it is not complete, this is because the following is not in general an equality.

$$g \, o \, f \quad \sqsubseteq \quad g \mid f$$

We choose absolute composition because it is complete, that is, it is the best operator consistent with the value of o on complete objects. This section considers category theory using $|$, and shows that if it is to be successful then the notion of completeness must be incorporated in the very definition of a "category". Presented in this section is a category-like structure called a **mategory** which the author has defined, and is currently working on and which incorporates completeness. Hopefully mategories will in the future help generalise completeness.

The first problem with absolute composition is that it is non-associative, and thus cannot be used for categorical composition. To see this consider the following example. Let A, B, and C be the following acpos



Let $f : A \to A$, $g : A \to B$, and $h : B \to C$ be the absolute functions such that

$$f(m_0) = \perp \qquad f(m_1) = \perp$$

$$g(m_0) = m_2 \qquad g(m_1) = m_3$$

$$h(m_2) = m_5 \qquad h(m_3) = m_6 \qquad h(m_4) = m_7$$

Then $|$ is non-associative as,

$$(h \mid (g \mid f))(m_0) = \perp, \qquad \text{and}$$

$$((h \mid g) \mid f)(m_0) = c$$

However, $|$ is **partial associative**, that is,

$$h \mid (g \mid f) \sqsubseteq (h \mid g) \mid f$$

While the lack of associativity is fatal from the point of view of constructing categories with $|$ it is not totally unexpected. For example, it was seen in Section 4.3) that o is not complete, the source of the problems in constructing categories of absolute functions. Due to historical reasons categories were not intended to take account of a distinction between partial and complete objects [Ma71]. $|$ can however be regarded as a form of "partial composition". This is in contrast to category theory where the only composition is (what may be termed) "complete". Our structures called **mategories** are still under investigation. The complete objects and concepts are taken straight from category theory. However, extending notions such as "morphism" is more difficult. This problem cannot be separated from the redefinition of "product", "sum", and "exponentiation". Such definitions in category theory are used to define concepts up to isomorphism. But what is "partial isomorphism" in mategory theory. When such problems have been worked out we can begin to think about using mategories of absolute functions to construct higher theories of domains which have a built-in notion of completeness.

Chapter 6

Conclusions and Further Work

Section 6.1) : Conclusions

There are a number of conclusions from this work. The first is that the notion of completeness can be used to reason about some "obviously correct" programs. This is justified by the work in Chapter 2 on the Cycle Product Theorem. The second conclusion is that a theory of Domains which use completeness is compatible with traditional least fixed point semantics. This is justified by the reformulation of Kahn's Semantics in Section 3.2) where unique and least fixed fixed points are compared. Also from this example, our third conclusion is that domains of complete objects are worth considering whenever devices such as the hiaton can be used to turn partial objects into complete ones. The fourth conclusion is that the complete objects in a domain should form an ultrametric space. This is because in such domains the completeness of objects can be proved, as the degree of completeness of an object is definable.

Our main conclusion from this work concerns the problems of extending completeness from simple domains such as Kahn's to more interesting structured domains. Our experience is that completeness is a natural (but uncomputable) concept, and that a general semantics cannot be constructed until all the semantic tools such as category theory include completeness in their definitions. In other words, completeness cannot be another theory building on top of someone else's theory, but has to be thought out from the very bottom.

The final conclusion is that this study has tried to go too far too quickly. The natural way in which completeness can be formalised in the Kahn and Lucid Domains has encouraged the author to leap forward into category theory. Thus our last conclusion is that completeness has first to be formalised in other simple domains before the above mentioned rethink of semantic tools can take place.

Section 6.2) : Further Work

There are several directions in which the work started in this dissertation can be continued. The first direction is to consider lazy languages such as Lucid whose complete objects form an agreement space. As mentioned in the text, these languages provide the best hope for realising Completeness Rules in programming languages. This direction probably has the most potential of all possible directions. The second direction is to consider other interpretations of agreement spaces in existing languages. One such example is the consideration of sequences of state transitions. In this interpretation an intentionally non-terminating program such as an operating system is complete if it does not "crash". It is felt by the author that pursuit in this direction would shed a lot of badly needed computational light upon the notion of completeness.

Other directions for further work are more abstract. Mategories are a good example. In such work we believe that "static" mathematical notions such as category theory have to be extended to meet the challenge of partial objects. This is precisely what we did in replacing metric spaces by metric domains in Chapter 2.

References

[A&W76] E.A.Ashcroft, W.W.Wadge

"Lucid - A Formal System for Writing and Proving Programs" SIAM Journal on Computing, Vol.5 No.3 Sep.76, pp.336-354

[A&W77] E.A.Ashcroft, W.W.Wadge

"Lucid, a nonprocedural language with iteration" Communications of the ACM, Vol.20 No.7 July 1977, pp.519-526

[Ba78] J.W.Backus

"Can Programming be Liberated from the von Neumann style: a functional language and it's algebra of programs" Communications of the ACM, Vol.21 No.8 August 1978.

[Con63] M.E.Conway

"Design of a Separable Transition-Diagram Compiler" Communications of the ACM, Vol.6 No.7 July 1963, pp. 396-408

[Fau82] A.A.Faustini

"The equivalence of an operational and a denotational semantics for pure dataflow" Ph.D. thesis 1982, Dept. of Computer Science, University of Warwick, Coventry U.K.

[Fau83] A.A.Faustini, S.G.Matthews, A.AG.Yaghi

"The pLucid Programming Manual" Dept. of Computer Science Report TR-83-004, Arizona State University, Tempe, AZ 85287, USA.

[PBH70] P.Brinch Hansen

"The Nucleus of a Programming System" Communications of the ACM, Vol.13 No.4 April 1970 pp.238-241

[He76] P.Henderson, J.Morris

"A Lazy Evaluator" Proc. of the 3rd Symposium on the Principles of Programming Languages, Atlanta 76.

[Ho83] C.A.R.Hoare

"Notes on Communicating Sequential Processes" Technical Monograph PRG-33, Oxford University Computing Laboratory, August 1983.

[Ka74] Gilles Kahn

"The Semantics of a Simple Language for Parallel Programming" Information Proccessing 74, North-Holland Publishing Company (1974) pp.471-5

[Ko79] Robert Kowalski

"Algorithm = Logic + Control" Communications of the ACM (1979) Vol.22, No.7, pp.424-436

[Kr83] E.V.Krishnamurthy

"Introductory Theory of Computer Science" The Macmillan Computer Science Series, Macmillan Press Ltd., London 1983.

[Kur1] K.Kuratowski

"Topology" Vol.1 (1966), Academic Press

[Kur2] K.Kuratowski

"Topology" Vol.2 (1968), Academic Press

[Le76] Daniel J. Lehmann

"Categories for Fixed-point Semantics" Theory of Computation Report No.15, Dept. of Computer Science, University of Warwick.

[Ma71] Saunders MacLane

"Categories for the Working Mathematician" Graduate Texts in Mathematics No.5, Springer-Verlag, New York 1971

[Mc63] J.McCarthy

"Towards a Mathematical Science of Computation" Information Processing (1962), pp.21-28, North-Holland, Amsterdam (1963).

[Pa84] David Park

"The Fairness Problem and Nondeterministic Computing Networks" Foundations of Computer Science IV, Distributed Systems: Part 2, pp.133-161, J.W.de Bakker & J.van Leeuwen (editors), Mathematical Centre Tracts 158, Mathematisch Centrum, Amsterdam 83.

[Sim] George F. Simmons

"Introduction to Topology and Modern Analysis" International Student Edition, McGraw-Hill Kogakusha Ltd., Tokyo.

[St77] Joseph E. Stoy

"Denotational Semantics" The MIT Press Series in Computer Science.

[Suth] W.A.Sutherland

"Introduction to Metric and Topological Spaces" (1975), Clarendon Press, (Oxford University Press)

[Tar55] A.Tarski

"A Lattice-theoretical Fixpoint Theorem and its Applications" Pacific Journal of Mathematics (1955), 5, pp.285-309

[Tu36] A.M.Turing

"On computable numbers with an application to the Entscheidungs problem" Proc. London Mathematical Society (1936), 42 (Series 2), pp.230-65.

[Wa81] W.W.Wadge

"An Extensional Treatment of Dataflow Deadlock" Theoretical Computer Science, Vol.13 (1981) pp.3-15

[Wa85] W.W.Wadge, E.A.Ashcroft

"Lucid, The Data Flow Programming Language" Accademic Press 1985, London

[We81] Klaus Weihrauch

"Computability on Metric Spaces" NR.21, Informatik Berichte, Fern Universitat, Gesamthochschule, D-5800 Hagen, Feithstr.140, Postfach 940.

[W&S] Klaus Weihrauch, Ulrich Schreiber

"Metric Spaces defined by Weighted Algebraic CPOs" Lehrstuhl fuer Informatik I, Rheinisch-Westfaelische, Technische Hochschule Aachen, D-5100 Aachen, W.Germany.