Domain Theory and Continuous Data Types

Abbas Edalat

20 lectures + 10 tutorials

Independent Study Option: 2003–2004

Department of Computing
Imperial College London
180 Queen's Gate
London SW7 2BZ

1 Introduction

The basic idea of domain theory is to use simple recursion to find increasingly better approximations to a desired object. To formalise the notion of approximation, we use *partial orders*.

Definition 1.1 A *partial order* (or a *partially ordered set* or a *poset*) $(D, \leq)$ is a set D with a binary relation $\leq$ which is

(i) reflexive: $a \leq a$,

(ii) anti-symmetric: $a \leq b \ \wedge \ b \leq a \ \Rightarrow \ a = b$, and

(iii) transitive: $a \leq b \ \wedge \ b \leq c \ \Rightarrow \ a \leq c$.

Example 1.2 (i) $(\mathbb{N}, \leq)$, the set of natural numbers with the usual ordering '$\leq$' is a poset.

(ii) For any set A, $(\mathcal{P}(A), \subseteq)$, the set of subsets of A ordered by inclusion is a poset. Similarly, $(\mathcal{P}(A), \supseteq)$ is a poset.

Sometimes we write the binary relation of a partial order as $\sqsubseteq$. Then $a \sqsubseteq b$ is taken to mean that a is an *approximation* to b or that b gives us more *information* than a.

1.1 Complete partial orders

If we have a chain $d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq d_3 \sqsubseteq \cdots$ we want an element which gives exactly the total information provided by the elements of the chain.

This leads us to the idea of a *complete* poset.

If $A \subseteq D$ is a subset of a poset D, we say A has a *least upper bound* (*lub*) or *supremum* if there exists an element $d \in D$ such that

- d is an upper bound for A, i.e. $a \sqsubseteq d$ for all $a \in A$
- If d' is any upper bound for A then $d \sqsubseteq d'$.

We usually write $d = \bigsqcup A$.

Definition 1.3 A *cpo* (*complete partial order*) or a *domain* is a poset with a least element denoted $\perp$ such that every chain $d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq d_3 \sqsubseteq \cdots$ has a least upper bound, denoted by $\bigsqcup_{i \geq 0} d_i$ or, often for convenience, simply by $\bigsqcup_i d_i$.

Thus $\bigsqcup_i d_i$ gives precisely the total information contained in the chain (and no more).

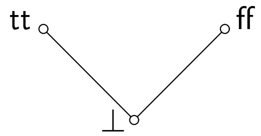
Example 1.4 $(\mathcal{P}(A), \subseteq)$ is a cpo for any set A. Similarly, $(\mathcal{P}(A), \supseteq)$ is a cpo.

1.2 Flat domains

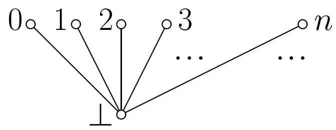
Let S be any set. We define the *flat domain* $S_\perp$ as follows. Consider the set $S \cup \{\perp\}$ with the partial order $\sqsubseteq$ defined by $a \sqsubseteq b$ iff $a = b$ or $a = \perp$.

All chains in $S_\perp$ will have at most two different elements. Hence $S_\perp$, called the *lift* of S, is a cpo. Similarly, we can define the lift $D_\perp$ of a cpo D.

Example 1.5 Let $\mathbb{B} = \{\mathsf{tt}, \mathsf{ff}\}$ be the set of Booleans consisting of true denoted by tt and false denoted by ff. Then, $\mathbb{B}_\perp$ of $\mathbb{B}$ is given by:



Example 1.6 The flat domain $\mathbb{N}_\perp$ of natural numbers $\mathbb{N}$ is given by:



Example 1.7 For any pair of sets S and T, the set of *strict* maps $f : S_\perp \to T_\perp$ with $f(\perp_S) = \perp_T$ ordered pointwise, i.e. $f \sqsubseteq g \stackrel{\text{def}}{\iff} \forall x \in S.\ f(x) \sqsubseteq g(x)$, is a cpo.

1.3 Domain of streams

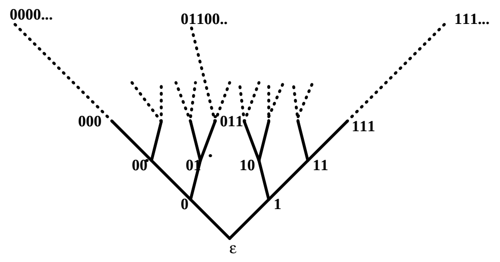
For any set Σ, denote by $\mathsf{Str}(\Sigma)$ the set of all finite and infinite sequences over Σ, including the empty sequence ϵ, ordered by prefix ordering:

$$a \sqsubseteq b \overset{\text{def}}{\iff} a \text{ is an initial subsequence of } b.$$

So if $\Sigma = \{0, 1\}$ we have e.g. $\epsilon \sqsubseteq 0 \sqsubseteq 01 \sqsubseteq 011$. It can be checked that this gives a cpo.

For example, the lub of the chain $\epsilon \sqsubseteq 01 \sqsubseteq 010 \sqsubseteq 0100 \sqsubseteq \cdots \sqsubseteq 010^n \sqsubseteq \cdots$ is just $010^\omega = 01000\cdots$.

We picture this cpo as a binary tree as follows:



Each node of the tree corresponds to a finite string. The maximal elements of the cpo are precisely the infinite strings. (An element x of a poset P is maximal if $x \sqsubseteq y$ implies $x = y$ for all elements $y \in P$.) In general, if $\Sigma = \{1, 2, \ldots, N\}$ then $\mathsf{Str}(\Sigma)$ is a cpo giving an N-ary tree.

1.4 Domain theory in semantics

Domain theory was introduced by Dana Scott in 1970 as a mathematical theory of programming languages. Two basic problems in computer science have given rise to domain theory. One is the need to canonically solve fixed point equations or recursive equations of procedures and data-structures.

As an example, consider negation as a function $\neg : \mathbb{B} \to \mathbb{B}$. This has no fixed point x with $\neg x = x$. But the strict extension $\neg : \mathbb{B}_\perp \to \mathbb{B}_\perp$ on the flat domain $\mathbb{B}_\perp$ has a fixed point, namely $\perp$. In general, every well-behaved function on a domain has a canonical fixed point. We can then give meaning to functions and objects which are recursively defined.

For example, consider the function

$$\mathsf{fac}(n) = \begin{cases} 1 & \text{if } n = 0 \\ n \times \mathsf{fac}(n-1) & \text{if } n > 0. \end{cases}$$

We will see later that we can "unfold" this recursive equation to get a chain of functions $f_i : \mathbb{N}_\perp \to \mathbb{N}_\perp$ each finitely presented:

$$f_i(x) = \begin{cases} x! & \text{if } 0 \leq x \leq i-1 \\ \perp & \text{otherwise} \end{cases}$$

Indeed, we will obtain the function fac as the fixed point of another function on a domain.

1.5 Recursive data types

The other basic problem in semantics which gave rise to domain theory is the need to canonically obtain recursively defined datatypes.

For example, suppose a data type is defined by $L = (L + L)_\perp$. This means that an element in our data type is either undefined ($\perp$) or is given by 0 (left) or 1 (right) followed by another such element.

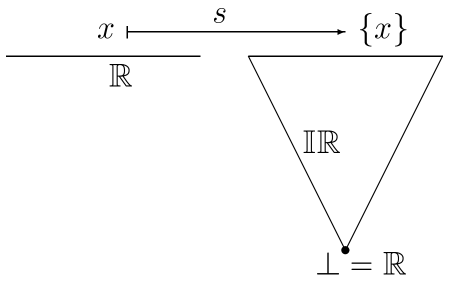
Then, we can check directly that the cpo $\mathsf{Str}\{0, 1\}$ is a solution of this recursive equation. Indeed, the empty sequence in $\mathsf{Str}\{0, 1\}$ can be regarded as the undefined element and any other element in $\mathsf{Str}\{0, 1\}$ is given by 0 or 1 followed by another element of $\mathsf{Str}\{0, 1\}$.

Domain theory provides a general technique to solve such equations. Starting from a domain equation as above one can then systematically construct a domain which satisfies the required specification in the domain equation.

In order to solve domain equations, we will introduce the notion of a domain approximating another domain and will use a technique which is very similar to finding the fixed point of well-behaved functions on domains.

1.6 Exact real arithmetic

Consider the set $\mathbb{IR}$ consisting of the real line $\mathbb{R}$ and all its closed intervals $[a, b] = \{x \mid a \leq x \leq b\}$ ordered by reverse inclusion. This is a cpo. Any real number $x \in \mathbb{R}$ is represented in this cpo as the singleton element $\{x\}$, i.e. $\mathbb{IR}$ contains a copy of $\mathbb{R}$ namely the image of the singleton map $s : \mathbb{R} \to \mathbb{IR}$:

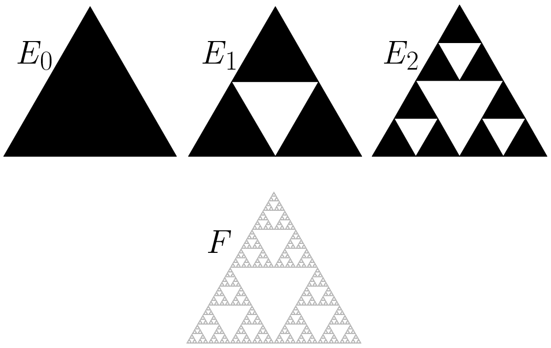


A chain in $\mathbb{IR}$ is simply a shrinking sequence of nested closed intervals. Any real number can be represented by such a sequence of intervals which give better and better approximations to the real number. Eg., π identified as $\{\pi\}$ is the lub of the chain $[0, 10] \supseteq [3, 4] \supseteq [3.1, 3.2] \supseteq [3.14, 3.15] \supseteq \ldots$. We will construct a framework for exact real arithmetic using this model, in particular, for computing elementary functions such as $+, -, \times, \div, \sin, \cos$ etc.

1.7 Fractal geometry

Since 1993, domain theory has also found new applications in computations in dynamical systems and fractal geometry. Here we give a simple example.

Example 1.8 The Sierpinski triangle is obtained by starting with an equilateral triangle E_0, dividing it into four smaller equilateral triangles, removing the interior of the middle one to obtain E_1 and repeating the same with each of the three remaining triangles to get E_2 and so on. Then, E_n, for $n = 0, 1, 2\cdots$, forms a shrinking sequence of subsets of $\mathbb{R}^2$. The Sierpinski triangle F is the intersection of all E_n's, i.e. $F = \bigcap_{n\geq 0} E_n$. We will see that F is in fact the fixed point of a function on a cpo.



1.8 References

- *Fractals Everywhere*, M. F. Barnsley, Academic Press, 2nd edition, 1993. (Covers the background on fractals and image compression.)
- *Domains*, G. Plotkin, 1983. (The first comprehensive text on domain theory.) Available on WWW with URL

  http://hypatia.dcs.qmw.ac.uk/sites/other/domain.notes.other/

  as domains.ps.Z
- *Domain Theory*, S. Abramsky and A. Jung, Handbook of Logic in Computer Science, vol. 3, Clarendon Press, 1994. (Advanced domain theory.) Available on WWW:

  http://theory.doc.ic.ac.uk/people/Edalat/handbook.ps
- *Mathematical Theory of Domains*, V. Stoltenberg-Hansen and I. Lindström and E. R. Griffor, Cambridge University Press, 1994.
- *Domains and Lambda-Calculi*, R. Amadio and P. Curien, Cambridge University Press, 1998. (Comprehensive text on domain theory and semantics.)
- *Denotational Semantics*, D. A. Schmidt, Wm. C. Brown Publishers, 1986. (Basic application of domain theory in semantics.)
- *Semantic domains*, C. A. Gunter and D. Scott, Handbook of theoretical computer science / managing editor, J. van Leeuwen. 1990. (A basic introduction to domain theory.)

- *Semantics of Programming Languages*, C. A. Gunter, MIT Press, 1992. (Application of domain theory in semantics.)
- *Formal Semantics of Programming languages*, G. Winskel, MIT Press, 1993. (Basic application of domain theory in semantics.)
- *The Lambda Calculus, Its Syntax and Semantics*, H. P. Barendregt, North Holland, 1984. (Contains an introductory chapter on domain theory.)
- There are no introductory texts yet on applications of domain theory in integration theory, fractal geometry or exact arithmetic. However, there is the following survey article:

  Domains for Computation in Mathematics, Physics and Exact Real Arithmetic, A. Edalat, Bulletin of Symbolic Logic 3(4), 1997, 401–452, Available from: `http://theory.doc.ic.ac.uk/~ae`

2 Fractals

A distinction can be broadly made between human-made objects and natural objects.

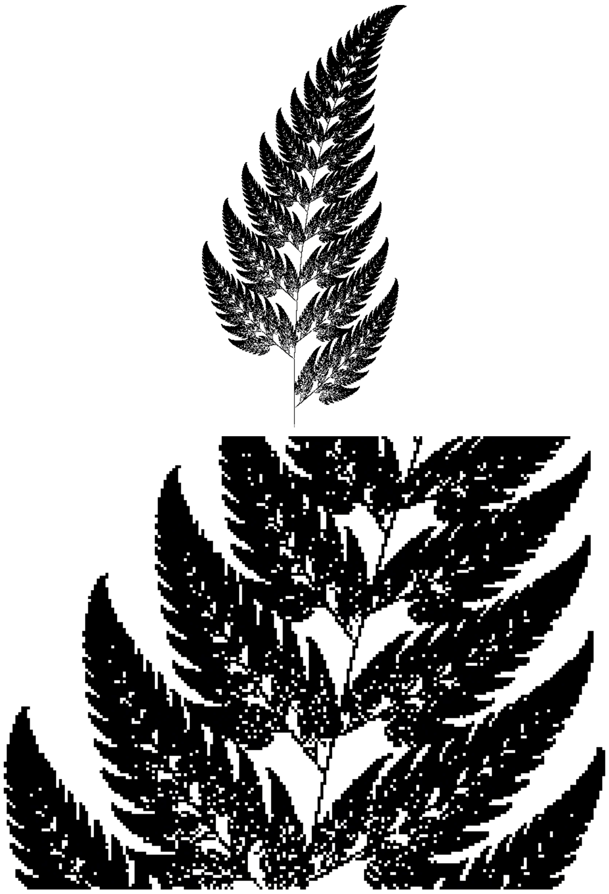
Human-made objects usually have simple structures and piecewise smooth boundaries.

We rely on traditional geometry based on the mathematics of polynomials to model these objects: Points, lines, circles, cubes, spheres, etc. are used to model the design of buildings, roads, wheels, and cars.

Natural objects, however, usually have very complex and fine structures with non-smooth boundaries. Viewed at greater magnification, natural objects such as landscapes, coastlines, leaves or clouds reveal more and more structure.

Here, classical geometry can only provide a crude approximation in modelling such objects by suppressing and smoothing out their fine details. It does not therefore provide a suitable framework for modelling natural objects. In order to encode a digitised picture of a fern or a cloud exhibiting its fine structure in a graphics system, we have to specify the address and the colour attribute of each point in the fern or the cloud, which gives us a huge data base.

A Spleenwort fern and a close-up.



We need a richer class of geometrical shapes which conform to natural objects. Fractal geometry provides such a framework.

Mandelbrot, a French mathematician, coined the word fractal, originating from the Latin *fractus* (meaning fractured or broken), to describe objects which are too irregular to fit into a traditional geometric setting.

He developed a new geometry called fractal geometry to describe the "geometry of nature". Classical geometry provides only a first approximation to natural objects. Fractal geometry is an extension of classical geometry.

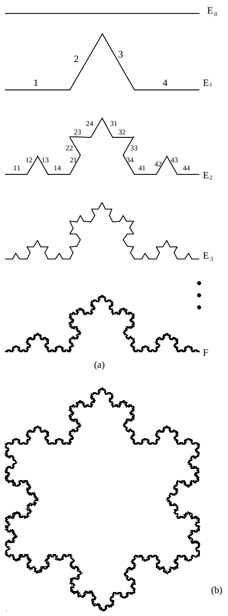
We will now study a simple but rich class of fractals which are generated by simple recursion using a finite number of functions.

It turns out that *all* natural objects can be approximated by this class of fractals. This means that they are useful to model natural objects.

We begin by giving three examples of fractals which we will study in some detail. They have a number of common features. One is that we can model them by cpo's. These examples will motivate us to introduce the general case.

2.1 The Koch snowflake

Start with the unit interval. Remove the middle third of the interval and replace it by the other two sides of the equilateral triangle based on the removed segment. The resulting set E_1 has four segments. Apply the algorithm to each of these to get E_2. Repeat to get E_n for all $n \geq 1$. The limiting curve F is the Koch curve, figure (a). Joining three such curves gives us the snowflake as in figure (b).



2.2 Affine maps

An *affine* map of $\mathbb{R}^m$ is a linear map followed by a translation. In $\mathbb{R}^2$, an affine map $f : \mathbb{R}^2 \to \mathbb{R}^2$ has, in matrix notation, the following action on a point (x, y) of the plane:

$$f\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} k \\ l \end{pmatrix}.$$

For the generation of the Koch curve there are four contracting affine transformations of the plane f_i, $1 \leq i \leq 4$, each a combination of a simple contraction, a rotation and a translation, such that

$$E_{n+1} = f_1(E_n) \cup f_2(E_n) \cup f_3(E_n) \cup f_4(E_n).$$

We can label the sides of the polygons E_n with finite sequences of 1,2,3 and 4 with length n as follows. The four segments of E_1 are labelled by 1 to 4 from left to right. Each of these segments gives rise to four segments in E_2, which are now labelled by two digits: The first is the label of the original segment while the second corresponds to the new relative position they occupy. In other words, the 4^n segments of E_n are labelled by the 4^n strings of 1,2,3 and 4 such that $m_1 m_2 \ldots m_n$, where $m_i = 1, 2, 3$ or 4 $(0 \leq i \leq n)$, is the label of the segment $f_{m_1}(f_{m_2}(\ldots(f_{m_n}(E_0))\ldots))$ in E_n.

2.3 A cpo for the Koch curve

Any infinite string of 1,2,3 and 4 now determines an address for a point of the Koch curve.

This means that we can model the generation of the Koch curve by the cpo $S_4 = \mathsf{Str}(\{1,2,3,4\})$.

Then the E_n's are made up of segments labelled by the finite sequences of $1,2,3,4$ and the points of the Koch curve correspond to the infinite sequences. A point represented by $m_1 m_2 m_3 m_4 \cdots$ is then the "limit" of the chain of segments

$$m_1 \sqsubseteq m_1 m_2 \sqsubseteq m_1 m_2 m_3 \sqsubseteq \cdots.$$

It is easily seen that f_1, f_2, f_3, f_4 correspond to the functions $f_1', f_2', f_3', f_4' : S_4 \to S_4$ by

$$f_i'(a) = ia \qquad i = 1,2,3,4.$$

Furthermore, the Koch curve F is the solution of the recursive equation

$$F = f_1(F) \cup f_2(F) \cup f_3(F) \cup f_4(F) \stackrel{\text{def}}{=} f(F).$$

Similarly the set F' of the infinite sequences of $\mathsf{Str}(\{1,2,3,4\})$ is the solution of the equation

$$F' = f_1'(F') \cup f_2'(F') \cup f_3'(F') \cup f_4'(F') \stackrel{\text{def}}{=} f'(F').$$

2.4 Properties of the Koch curve F

1. F is the limit of a sequence of simple polygons which are recursively defined.

2. Fine structure, self-similarity: It is made up of four parts, each similar to F but scaled by $\frac{1}{3}$.

3. Complicated local structure: F is nowhere smooth (no tangents anywhere). Think about the chain of segments $2 \sqsubseteq 22 \sqsubseteq 222 \sqsubseteq \cdots$. It spirals around infinitely many times!

4. F has infinite length (E_n has length $(\frac{4}{3})^n$ and this tends to infinity with n) but occupies zero area. The snowflake can be painted but you cannot make a string go around it!

5. Although F is defined recursively in a simple way, its geometry is not easily described classically (cf. definition of a circle). But the cpo $\mathsf{Str}(\{1,2,3,4\})$ gives a good model for F.

6. It is difficult to see what the dimension of F can be: since it has infinite length within a bounded region, it is too big to have dimension one. (It also looks thicker than an ordinary line.) But, since it has zero area, it is too small to have dimension two. Does it have a dimension strictly between 1 and 2?

2.5 Towards fractal dimension

We have the following intuition for the dimension of man-made objects:

Object	**Dimension**
finite set of points	zero
(piecewise) smooth curve	one
(piecewise) smooth surface	two
solid volume	three

Classical geometry confirms this intuition. But it also tells us that the dimension of any curve, in particular the Koch curve, is one, which we find difficult to accept.

[In classical geometry the dimension of a subset of $\mathbb{R}^m$ is always an integer and is recursively defined: It is zero if the set is totally disconnected (i.e. for any two points in the set there is a closed surface not in the set, containing one of the two but not the other), it is one if each point has arbitrarily small neighbourhoods with boundary of dimension zero and so on.]

We want a new notion of dimension which coincides with the traditional notion for the objects of the above table but gives a more satisfactory value for the dimension of the Koch curve, as well as the other examples of fractals that we will study.

2.6 Similarity dimension

Consider the following simple method to calculate dimension using recursion and self-similarity. Take the interval $[0,1]$ and any integer $n > 0$. Reduce the interval by $\frac{1}{n}$. Then we need n^1 copies of this reduced interval to make up the original interval and the dimension of unit interval is $\frac{\log n^1}{\log n} = 1$. Similarly, if we reduce the unit square $[0,1]^2$ by $\frac{1}{n}$ we need n^2 copies of the reduced square to make up the original square and the dimension of $[0,1]^2$ is $\frac{\log n^2}{\log n} = 2$. The dimension of $[0,1]^3$ is $\frac{\log n^3}{\log n} = 3$.

Definition 2.1
A set which can be made up with m copies of itself scaled by $\frac{1}{n}$ has *similarity* dimension $\frac{\log m}{\log n}$.

The Koch curve, made up of four copies of itself scaled by $\frac{1}{3}$, has therefore similarity dimension $\frac{\log 4}{\log 3} = 1.261\ldots$. This is between 1 and 2, consistent with our expectation.

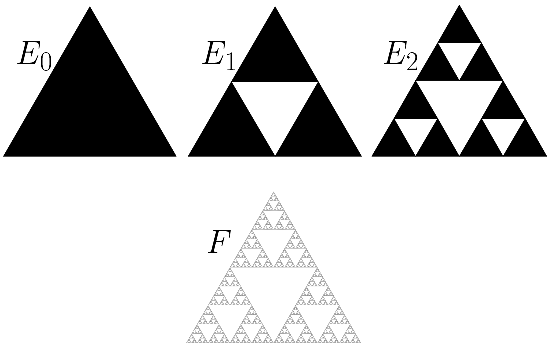
The above definition works only if the object is strictly self-similar. (There is a general notion of fractal dimension which reduces to the similarity dimension when the latter is defined.)

2.7 The Sierpinski triangle

Recall that the Sierpinski triangle is obtained by starting with an equilateral triangle E_0, dividing it into four smaller equilateral triangles, removing the interior of the middle one to obtain E_1 and repeating the same with each of the three remaining triangles to get E_2 and so on. Again, there are three contracting affine transformations f_1, f_2, f_3 such that

$$E_{n+1} \;=\; f_1(E_n) \cup f_2(E_n) \cup f_3(E_n).$$

The Sierpinski triangle F is the intersection of all E_n's, i.e. $F = \bigcap_{n\geq 0} E_n$.



2.8 Properties of the Sierpinski triangle

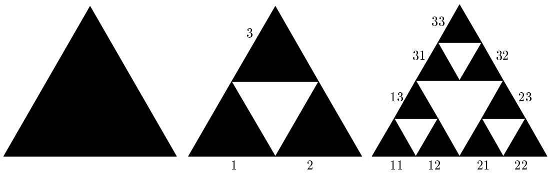
1. It is the intersection of a shrinking sequence of simple sets which are recursively defined.

2. Although F contains a polygon with infinite length, it occupies zero area.

3. Fine structure, self similarity: It is made up of three copies of itself scaled by $\frac{1}{2}$. It has dimension $\frac{\log 3}{\log 2} \approx 1.5$ and satisfies:

$$F = f_1(F) \cup f_2(F) \cup f_3(F) \stackrel{\text{def}}{=} f(F).$$

4. We can model the generation of the Sierpinski triangle by the cpo $S_3 = \mathsf{Str}\{1,2,3\}$: Label the triangles in E_1 by $1, 2$ and 3 as shown.



Repeat this for each of the bigger triangles in E_2 and so on. The Sierpinski triangle is then represented by F', the infinite sequences in S_3. If we define f_1', f_2', f_3' as in the case of the Koch curve we find that F' is the solution of the recursive equation:

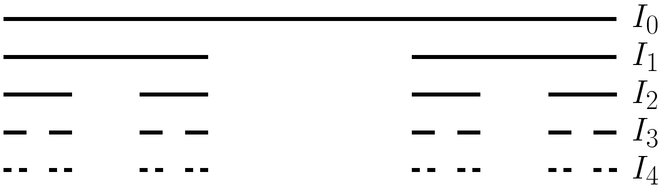
$$F' = f_1'(F') \cup f_2'(F') \cup f_3'(F') \stackrel{\text{def}}{=} f'(F').$$

2.9 The Cantor set

Start with $I_0 = [0,1]$. Remove the interior of its middle third to get $I_1 = [0, \frac{1}{3}] \cup [\frac{2}{3}, 1]$. Do the same with each interval in I_1 to get I_2 and so on. We have

$$I_{n+1} \;=\; f_1(I_n) \cup f_2(I_n)$$

where the affine transformations f_1 and f_2 are given by: $f_1(x) = \frac{x}{3}$ and $f_2(x) = \frac{2}{3} + \frac{x}{3}$. The Cantor set C is the intersection of all I_n's.



How many points does C have?

All the end points of the intervals in I_n are in C. But C has many more points. In fact, C consists of all numbers in the unit interval whose expansions in base 3 do not contain the digit 1, i.e. numbers of the form

$$\frac{a_1}{3} + \frac{a_2}{3^2} + \frac{a_3}{3^3} + \cdots \qquad \text{where } a_i = 0 \text{ or } 2.$$

This is an uncountable set of numbers.

2.10 Properties of the Cantor set

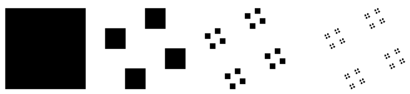
1. It is the intersection of a shrinking sequence of simple sets (each being a finite union of intervals) which are recursively defined.

2. Fine structure and self-similarity: It is made up of two copies of itself scaled by $\frac{1}{3}$. It satisfies: $C = f_1(C) \cup f_2(C)$.

3. C is uncountable but has zero length.

4. Complicated local structure: C is totally disconnected (between any two points in C there is always a point not in C) and therefore contains no intervals; yet it has no isolated points either (in any neighbourhood of a point in C there are infinitely many points of C). As before it can be modelled by a cpo, say, $\mathsf{Str}\{1,2\}$.

The Cantor dust is obtained by repeatedly dividing up a unit square into 16 smaller squares and removing all but four of them as in the picture below. What is its similarity dimension?



2.11 Some conclusions

Having studied the Koch curve, the Sierpinski triangle and the Cantor set we can infer that fractals are objects with complicated local shape, very fine structure and usually non-integral dimension. But a rich class of fractals, like the above examples, are obtained as a solution of a simple recursive equation involving a finite number of contracting affine transformations.

We have also seen that the generation of these fractals can be modelled by complete partial orders.

A number of questions now remain to be answered.

How general is our theory? When we generate e.g. the Koch curve, does it matter where we start the iteration? Do we still get the same curve in the limit if we start not with unit interval but, say, with the unit square, or a finite set of points?

Since in practice we can only iterate our function a finite number of times, we also need to measure the degree our finite iterates approximate the limiting fractal object.

We answer these questions in the next section by studying fractals in a proper framework. This framework is given by *Iterated Function Systems.*

3 Fixed points of Contracting Maps

Definition 3.1 A *distance function* d defined on a set X assigns to every two points $x, y \in X$ a non-negative number $0 \leq d(x, y) \leq \infty$ with

- $d(x, x) = 0$ (reflexive).
- $d(x, y) = d(y, x)$ (symmetric).
- $d(x, z) \leq d(x, y) + d(y, z)$ (triangular inequality).
- $d(x, y) = 0 \Rightarrow x = y$ (antisymmetric).

We write (X, d) to indicate that we have a set X with a distance function d defined on it.

Example 3.2

- In $\mathbb{R}$ (the real line), $\mathbb{R}^2$ (the plane), and $\mathbb{R}^3$ (the three-space) the Euclidean distance $|x - y|$ between points defines a distance function. In $\mathbb{R}^2$:
$$|x-y| = \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2}.$$
- For two distinct sequences $a = a_0 a_1 \ldots$ and $b = b_0 b_1 \ldots$ (finite or infinite) in $\mathsf{Str}(\Sigma)$, define $d(a, b) = \frac{1}{2^n}$ where $n \geq 0$ is the least integer such that $a_n \neq b_n$ (or one of them is undefined). For example $d(0101, 011\bar{0}) = \frac{1}{2^2} = \frac{1}{4}$. Then d gives a distance function on $\mathsf{Str}(\Sigma)$.

3.1 Contracting maps

A map $f : (X, d) \to (X', d')$ is *contracting* if there exists $0 \leq s < 1$ such that f reduces all distances by at least a factor s, i.e. if $d'(f(x), f(y)) \leq s d(x, y)$ for all $x, y \in X$. We say s is a *contracting factor* for f. The smallest such s is called the *contractivity* of f. Eg., each of the maps in the generation of the Koch curve, the Sierpinski triangle and the Cantor set is a contracting map.

Given a map $f : X \to X$ on any set, we say $x \in X$ is a *fixed point* of f if $f(x) = x$.

Exercise 3.3 Each of the maps in the generation of the Koch curve, the Sierpinski triangle and the Cantor set has a unique fixed point.

Any contracting map $f : \mathbb{R}^m \to \mathbb{R}^m$ has a unique fixed point obtained as follows. Choose $r > 0$ large enough so that the disk D_r of radius r with centre the origin is mapped into itself by f. Eg., put $r = \frac{|f(0)|}{1-s}$, then for $x \in D_r$ we get $|f(x)| \leq |f(x) - f(0)| + |f(0)| \leq s|x| + |f(0)| \leq sr + |f(0)| = \frac{s|f(0)|}{1-s} + |f(0)| = \frac{|f(0)|}{1-s} = r$, i.e. $f(x) \in D_r$.

The sequence $D_r \supseteq f(D_r) \supseteq f^2(D_r) \supseteq \cdots$ shrinks to a single point which is the fixed point of f.

3.2 Computing the fixed point

The unique fixed point x^* of f is therefore given by
$$\{x^*\} = \bigcap_{n \geq 0} f^n(D_r).$$

In the cpo $(\mathcal{P}(D_r), \supseteq)$ we have $x^* = \bigsqcup_{n \geq 0} f^n(D_r)$.

We want to compute the fixed point up to some given accuracy $\epsilon > 0$, i.e., find a point whose distance from x^* is at most ϵ. Usually, $\epsilon = \frac{1}{2^k}$ or $\frac{1}{10^k}$.

The diameter of D_r is $2r$ and f has contractivity factor s. Hence, the diameter of the set $f^n(D_r)$ is at most $2rs^n$.

Let n be the first positive integer such that this diameter satisfies $2rs^n \leq \epsilon$, i.e. $n = \left\lceil \frac{\log(\epsilon/2r)}{\log s} \right\rceil$.

Since $x^* \in f^n(D_r)$ and this set has diameter at most ϵ, it follows that any point in $f^n(D_r)$, for example $f^n(0)$, would be the required approximation.

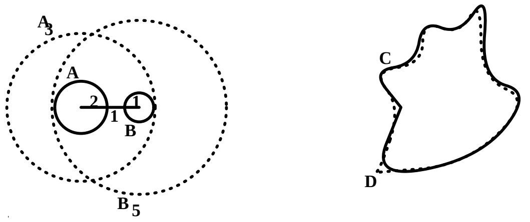
In applications to computer graphics, we rescale and translate so that f maps the screen represented as the unit square $[0, 1]^2$ into itself, i.e.
$$f : [0, 1]^2 \to [0, 1]^2.$$

We then take ϵ to be of pixel size to plot the best approximation to the fixed point.

3.3 Iterated Function Systems

Definition 3.4 An *Iterated Function System (IFS)* in $\mathbb{R}^m$ consists of a finite number of contracting maps $f_i : \mathbb{R}^m \to \mathbb{R}^m$, $i = 1, 2 \dots N$.

We obtain a map $f : \mathcal{P}(\mathbb{R}^m) \to \mathcal{P}(\mathbb{R}^m)$ defined by $f(A) = \bigcup_{1 \le i \le N} f_i(A)$. Is it contracting?
We define the *Hausdorff distance*, $d_H(A, B)$, between two non-empty elements A and B in $\mathcal{P}(\mathbb{R}^m)$ as the least number d such that every point of A is within distance d of some point of B and every point of B is within distance d of some point of A. E.g. in the figure below $d_H(A, B) = 5$. If $d_H(C, D)$ is small then C and D are 'visually' close together:



We can obtain $d_H(A, B)$ as follows. For $r \ge 0$, the *r-neighbourhood* of A is:

$$A_r = \{x \in \mathbb{R}^m \mid |x - a| \le r \text{ for some } a \in A\}.$$

Then: $d_H(A, B) = \min\{r \mid B \subseteq A_r \text{ and } A \subseteq B_r\}$.

3.4 The Attractor of an IFS

The map $f : \mathcal{P}(\mathbb{R}^m) \to \mathcal{P}(\mathbb{R}^m)$ given by $f(A) = \bigcup_{1 \le i \le N} f_i(A)$ is contracting for non-empty subsets in $\mathcal{P}(\mathbb{R}^m)$ with a contracting factor $s = \max_i s_i$ where s_i is a contracting factor for f_i. (Check this!) There is a largest subset A^* satisfying $f(A^*) = A^*$, i.e. a fixed point of f, which is called the *attractor* of the IFS. In fact, A^* is the most interesting fixed point of f. Any other fixed point is a subset of A^*.

The attractor is obtained as follows. Let D_{r_i} be a disk with radius r_i centred at the origin which is mapped by f_i into itself. Put $r = \max_i r_i$. Then $D = D_r$ is mapped by f into itself. We get:

$$D \supseteq f(D) \supseteq f^2(D) \supseteq \cdots$$

and $A^* = \bigcap_{n \ge 0} f^n(D)$.
In the cpo $(\mathcal{P}(D), \supseteq)$ we have $A^* = \bigsqcup_{n \ge 0} f^n(D)$.

Example 3.5

(i) The Koch curve is the attractor of the IFS $\{f_1, f_2, f_3, f_4\}$ with a contracting factor $1/3$.

(ii) The Sierpinski triangle is the attractor of the IFS $\{f_1, f_2, f_3\}$ with a contracting factor $1/2$.

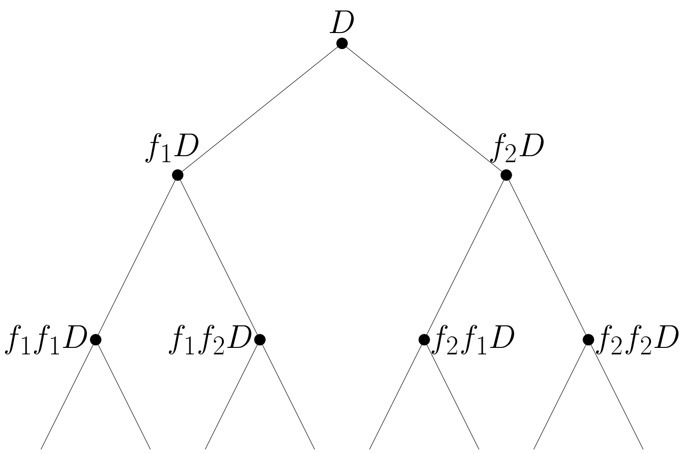
(iii) The Cantor set is the attractor of the IFS $\{f_1, f_2\}$ with a contracting factor $1/3$.

3.5 IFS tree

The nth iterate

$$f^n D = \bigcup_{i_1, i_2, \cdots i_n = 1}^{N} f_{i_1} f_{i_2} \cdots f_{i_n} D$$

generates the N^n nodes of the nth level of the *IFS tree*. For $N = 2$, three levels of this tree are shown below. Each node is a subset of its parent node.



Each branch of the tree is a sequence of subsets $D \supseteq f_{i_1}(D) \supseteq f_{i_1} f_{i_2} D \supseteq f_{i_1} f_{i_2} f_{i_3}(D) \supseteq \cdots$ where $i_1, i_2, \cdots \in \{1, 2, \cdots N\}$ whose intersection contains a single point as each map f_j is contracting. The attractor A^* is the set of all such points for all the branches of the IFS tree.

3.6 Approximation of the Attractor

We use the IFS tree to obtain an algorithm to generate a discrete approximation to the attractor A^* up to a given $\epsilon > 0$ accuracy. In other words, we will obtain a finite set A such that $d_H(A, A^*) \le \epsilon$.

Let $n = \left\lceil \frac{\log(\epsilon / 2r)}{\log s} \right\rceil$. Consider the truncated tree at level n. Then the diameters of the N^n leaves of the tree are at most ϵ. Pick the distinguished point

$$f_{i_1} f_{i_2} \cdots f_{i_n}(0) \in f_{i_1} f_{i_2} \cdots f_{i_n}(D)$$

for each of the N^n leaves. Let A be the set of these N^n points.

Each point in A^* is in one of the N^n leaves each of which has diameter at most ϵ and contains one of the distinguished points and hence one point of A. It follows that $d_H(A, A^*) \le \epsilon$ as required.

The complexity of the algorithm is $O(N^n)$, i.e. of the order of N^n. Improve the efficiency of the algorithm by taking a smaller set of leaves. For each branch

$$D \text{ --- } f_{i_1} D \text{ --- } f_{i_1} f_{i_2} D \text{ --- } f_{i_1} f_{i_2} f_{i_3} D \text{ --- } \cdots$$

of the tree find an integer k such that the diameter of $f_{i_1} \cdots f_{i_k} D$ is at most ϵ by taking the first integer k such that $2 r s_{i_1} s_{i_2} \cdots s_{i_k} \le \epsilon$. Then take this node as a leaf. Do as before with this new set of leaves.

3.7 An alternative Algorithm

Assume we have an IFS with contracting factor s defined in $\mathbb{R}^m$.

Let $A_0 \subseteq \mathbb{R}^n$ be any non-empty bounded subset, i.e. one which is contained in some disk around the origin. Then $d_H(A_0, A^*) < \infty$ since both A^* and A_0 are bounded.

Proposition 3.6 *For any positive integer $n > 0$:*

$$d_H(f^n(A_0), A^*) \leq s^n d_H(A_0, A^*).$$

Proof We use induction. For $n = 1$, we have:

$$d_H(f(A_0), A^*) = d_H(f(A_0), f(A^*)) \leq s d_H(A_0, A^*).$$

The inductive step is similar.□

Therefore, for large n, $d_H(f^n(A_0), A^*)$ becomes small, i.e. the iterates $f^n(A_0)$ give better and better approximations to A^*.

We can use this as the basis of an algorithm to generate approximations to A^* on a digitized screen. We start with a simple non-empty set A_0, e.g. the set of fixed points of the maps f_j ($j = 1, \cdots, N$). We find the images $f^n(A_0)$ for $n = 1, 2, \cdots$ until the image becomes constant.

The code of an IFS.

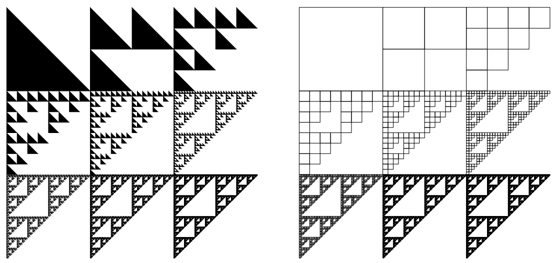
Any affine transformation $f : \mathbb{R}^2 \to \mathbb{R}^2$ of the plane can be written in matrix notation as:

$$f\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} k \\ l \end{pmatrix}.$$

We can therefore tabulate the a, b, c, d, k and l values of all the maps in the IFS. The resulting table is called the IFS code. The following is the code for an IFS whose attractor is the (right angled) Sierpinski triangle.

f_i	a	b	c	d	k	l
1	0.5	0	0	0.5	0	0
2	0.5	0	0	0.5	0	50
3	0.5	0	0	0.5	50	50

Below, you can see the result of nine iterations for this IFS with two different starting sets.



Encoding of images and **Image compression** is based on the following theorem, which says that any image can be approximated by the fixed point of an IFS.

Theorem 3.7 (The Collage Theorem) *Let E be a non-empty bounded subset of $\mathbb{R}^2$. Given an IFS with s, f and F as before, we have*

$$d_H(E, F) \leq d_H(E, f(E))/(1 - s).$$

Proof We have:

$$\begin{aligned} d_H(E, F) &\leq d_H(E, f(E)) + d_H(f(E), F) \\ &= d_H(E, f(E)) + d_H(f(E), f(F)) \\ &\leq d_H(E, f(E)) + s\, d_H(E, F). \end{aligned}$$

Hence, $d_H(E, F) \leq d_H(E, f(E))/(1 - s)$.□

Suppose that E is an image that we want to encode. If we can find an IFS with contractivity s such that $d_H(E, f(E)) \leq \epsilon$, then by the above theorem $d_H(E, F) \leq \epsilon/(1 - s)$, i.e. if ϵ is small then F is a good approximation to E and we can take the code of the IFS as the code for E, and use it for design, animation, pattern recognition, etc.

But, how do we find an IFS with

$$d_H(E, f(E)) = d_H(E, f_1(E) \cup f_2(E) \cup \ldots \cup f_N(E))$$

small?

We have to find a set of affine mappings that shrink distances and cause the target image to be approximated by the union of the affine mappings of the image.

We can view an object as the union of several subobjects and consider that each subobject is actually an instance of the original object, obtained by applying an affine mapping to the object. Each subobject is then a *tile* and our job is the *self-tiling* of the image.

The tiling scheme should completely cover the object, even if this necessitates overlapping the tiles or overhanging the target image slightly. We also want to keep the number of tiles small. The smaller the overlapping and the overhanging the more accurate our result will be: The fixed point of the IFS obtained will be a better approximation to our object.

This gives us the following elementary interactive scheme to encode an image. [There is a more advanced fractal image compression technique which is automated but we will not study it here.]

The algorithm starts with the image E displayed within a viewing window, taken to be $[0,1]\times[0,1]$, on the graphics workstation monitor.

An affine transformation f_1 with $a=d=\frac{1}{4}$, say, and $b=c=k=l=0$ is introduced and the image $f_1(E)$, a quarter-sized copy of E, is displayed on the monitor in a different colour from E. The user now interactively adjusts the values of a,b,c,d,k,l by specifying changes with, e.g., a mouse, so that the image $f_1(E)$ is variously translated, rotated, and sheared on the screen. The goal is to transform $f_1(E)$ so that it lies over part of E, with the smallest possible overhang.

Once $f_1(E)$ is suitably positioned, it is fixed, and a new subcopy of the target, $f_2(E)$ is introduced. Then, f_2 is interactively adjusted until $f_2(E)$ covers a subset of those pixels in E not in $f_1(E)$, with the smallest possible overlap.

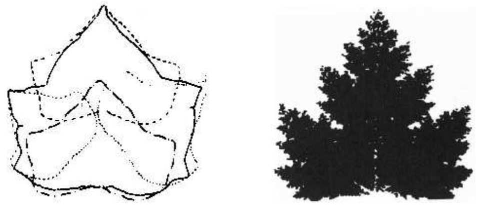
In this way, contracting maps $f_1,\ldots,f_N$ are determined so that E and the tiling

$$f_1(E)\cup\ldots\cup f_N(E)$$

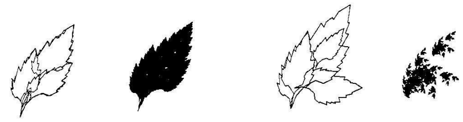
are visually close and N is as small as possible.

Below you see the approximate tiling or collage of a leaf with four contracting affine maps, whose codes are given in the following table. The attractor of the IFS is shown on the right hand side.

f	a	b	c	d	k	l
1	0.6	0	0	0.6	0.18	0.36
2	0.6	0	0	0.6	0.18	0.12
3	0.4	0.3	−0.3	0.4	0.27	0.36
4	0.4	−0.3	0.3	0.4	0.27	0.09



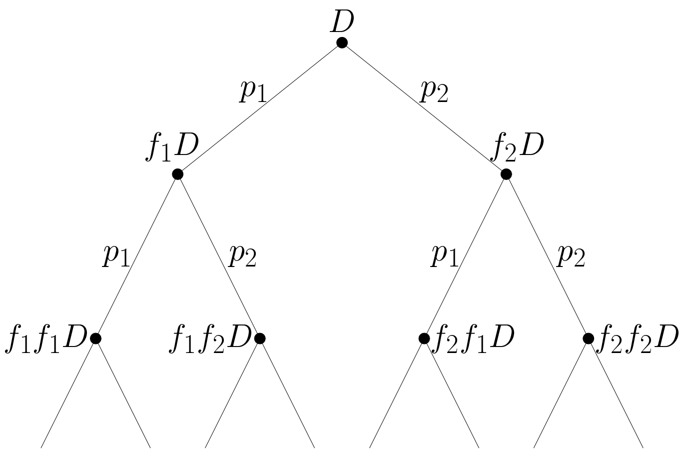
In the figure below, two tilings of a leaf and the corresponding attractors on the RHS are shown. The first tiling is good and the attractor resembles the leaf. The second is very poor and produces a bad approximation to the leaf.



3.8 Colours on Fractals

We can get a probability distribution or colours on an attractor by assigning probabilities to the IFS.

Definition 3.8 An *IFS with probabilities* in $\mathbb{R}^n$ is an IFS $\{f_1,\ldots,f_N\}$ such that each f_i is assigned a probability $p_i>0$ with $p_1+\cdots+p_N=1$. We get an *IFS tree with probabilities* shown for $N=2$ below, where D is, as before, a disk centred at the origin which is mapped into itself by every map f_i.



Consider one unit of mass on the root D. It is distributed in the ratio $p_1:p_2$ to its children f_1D and f_2D. The mass on each of these nodes is distributed in the ratio $p_1:p_2$ to its children and so on.

An IFS with probabilities gives rise to a probability distribution on the attractor of the underlying IFS. We will explain how we obtain a discrete probability distribution on a digitized approximation of the attractor that we constructed before. We construct as before a finite tree whose leaves give the required digitized approximation to the attractor. Note that different leaves can give rise to the same pixel since they can correspond in the digitized approximation to the same pixel.

We now associate a probability weight to each of the leaves of this tree. A leaf such as $f_{i_1}f_{i_2}\cdots f_{i_n}(D)$ gets weight $p_{i_1}p_{i_2}\cdots p_{i_n}$ which is precisely the amount of mass which it gets from the distribution of one unit mass on the root D.

For each pixel in the attractor, we obtain a probability weight by adding up all the weights of the leaves of the finite tree which are associated with that pixel.

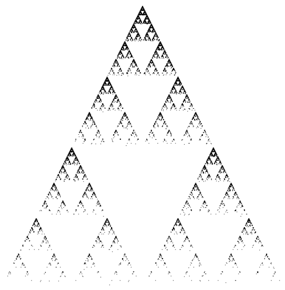
This gives us a probability distribution on the digitized approximation to the attractor. In fact, the root node has mass one and, given any node on the IFS tree, the total weights of the descendents of this node on each level of the IFS tree is equal to the weight of the node.

We can now use a gray-scale colouring scheme to shade each pixel in the attractor such that a darker gray colour for a pixel corresponds to a greater probability weight on that pixel. This is shown in the following example.

Example 3.9
Consider the IFS with probabilities:

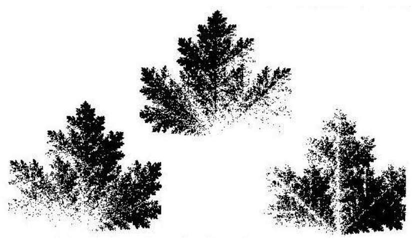
f_i	a	b	c	d	k	l	p
1	0.5	0	0	0.5	0	0	.25
2	0.5	0	0	0.5	48	0	.25
3	0.5	0	0	0.5	24	48	.5

The attractor is a Sierpinski triangle. In the figure below, the result of computing one thousand points is shown. The number of points in the upper triangle is roughly twice that of either of the two lower ones, and the same thing is true in each of the smaller triangles. This is repeated at all scales.



Different sets of probabilities assigned to the mappings of an IFS produce different density distributions and thus different textures of the attractor.

In the figure below three different sets of probabilities have produced three different renderings of the maple leaf.



We can colour the fractal by assigning different colour indices to different density distributions.

This concludes our study of fractals for now. Towards the end of the course we will study how *power domains*, which model *non-determinism*, are constructed from cpo's.

Although we will not give details, let us just mention that the attractors of IFSs can be obtained as the fixed point of a *continuous* map on a *convex* power domain. The probability distribution on the attractor of an IFS with probabilities can be obtained as the fixed point of a map on the *probabilistic* power domain.

We will now study fixed point semantics in cpo's in general.

4 Continuous functions on cpo's

In the last section, we saw two examples in which a map $f : D \to D$ on a cpo D had a fixed point which was obtained as the lub $\bigsqcup_n f^n(\perp)$ of the chain

$$\perp \sqsubseteq f(\perp) \sqsubseteq f^2(\perp) \sqsubseteq \dots .$$

We want to formulate a property which ensures that a map on a cpo has such a fixed point.

In computer science, we think of a map

$$f : D \to E$$

between cpo's D and E as a computation process: Given an input $d \in D$, f computes the output $f(d) \in E$.

Definition 4.1
A map $f : D \to E$ between cpo's D and E is called *monotone* if whenever $d \sqsubseteq_D d'$ in D, we have $f(d) \sqsubseteq_E f(d')$ in E. A monotone map f is called *continuous* if whenever

$$d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq \dots$$

is a chain in D, we have $\bigsqcup_i f(d_i) = f(\bigsqcup_i d_i)$.

In other words, a monotone map preserves the information order, and a continuous map preserves lubs of chains.

Intuitively, a computable map should be monotone: The more information provided as input the more information we should obtain as output.

We can also expect a computable map to be continuous: The output information $f(\bigsqcup_i d_i)$ computed with the total information $\bigsqcup_i d_i$ in a chain should be equal to the total information $\bigsqcup_i f(d_i)$ computed with the elements of the chain.

Note that if f is monotone we always have $f(d_i) \sqsubseteq f(\bigsqcup_i d_i)$ and hence $\bigsqcup_i f(d_i) \sqsubseteq f(\bigsqcup_i d_i)$. This means that if f is continuous we are *not* going to get more output information $f(\bigsqcup_i d_i)$ from the lub of a chain than the total output information $\bigsqcup_i f(d_i)$ computed with the elements of the chain. Intuitively, we require the same from a computable map.

Our intuition therefore tells us that a computable function must be continuous. This is called Scott's thesis after Dana Scott, who together with Christopher Strachey formulated the mathematical theory of semantics of programming languages.

It turns out that continuity is precisely what we need to ensure that a map on a cpo has a canonical fixed point. We first look at some examples.

Example 4.2

- Let $D = E = (\mathbb{N} \cup \{\infty\}, \leq)$ and f be given by $f(n) = n+1$ for $n \in \mathbb{N}$ and $f(\infty) = \infty$. Then f is continuous.
- Let $D = E = (\mathbb{N} \cup \{\infty\}, \leq)$ and let $f(n) = 0$ for $n \in \mathbb{N}$ and $f(\infty) = \infty$. Then f is clearly monotone but it is not continuous: Take the chain $a_n = n$ for $n \in \mathbb{N}$. Then $f(\bigsqcup_n n) = f(\infty) = \infty$ but $\bigsqcup_n f(n) = \bigsqcup_n 0 = 0$.
- Let $f : \mathsf{Str}\{0,1\} \to \mathsf{Str}\{0,1\}$ be given by $f(a) = 0a$. Then f is continuous.
- Let A be any set and $B \subseteq A$ a fixed subset. Let $f : (\mathcal{P}(A), \subseteq) \to (\mathcal{P}(A), \subseteq)$ be given by $f(X) = B \cup X$. Then f is continuous.
- The identity map $1_D : D \to D$ with $1_D(d) = d$ (for all $d \in D$) is continuous.
- Any constant map $f : D \to E$ is continuous.
- If $f : D \to E$ is monotone and D is finite, then f is continuous.
- More generally, if $f : D \to E$ is monotone and all chains in D are eventually constant then f is continuous.

Theorem 4.3 (Fixed Point Theorem) *Let $f : D \to D$ be a continuous map on the cpo D. Then f has a least fixed point given by $\mathsf{fix}(f) = \bigsqcup_n f^n(\perp)$.*

Proof We claim that $f^n(\perp) \sqsubseteq f^{n+1}(\perp)$ for all $n \geq 0$. We show this by induction on n. Since $\perp$ is the least element of D we have $\perp \sqsubseteq f(\perp)$. This establishes our claim for the base case $n = 0$. Assume that $f^n(\perp) \sqsubseteq f^{n+1}(\perp)$. Since f is continuous and hence monotone, we get $f^{n+1}(\perp) = f(f^n(\perp)) \sqsubseteq f(f^{n+1}(\perp)) = f^{n+2}(\perp)$. This completes the inductive proof. Therefore we have a chain

$$\perp \sqsubseteq f(\perp) \sqsubseteq f^2(\perp) \sqsubseteq f^3(\perp) \sqsubseteq \cdots.$$

This chain is mapped by f to the chain

$$f(\perp) \sqsubseteq f^2(\perp) \sqsubseteq f^3(\perp) \sqsubseteq \cdots.$$

Since f is continuous, it sends the lub of the first chain to the lub of the second chain. But the second chain is the same as the first except for the first element and therefore the two chains have the same set of upper bounds and hence the same lub. It follows that $f(\bigsqcup_n f^n(\perp)) = \bigsqcup_n f^n(\perp)$. Hence $\mathsf{fix}(f) = \bigsqcup_n f^n(\perp)$ is a fixed point for f.

Suppose x is any fixed point of f. Then we show by induction that $f^n(\perp) \sqsubseteq x$ for all $n \geq 0$.

For $n = 0$ this is simply $\perp \sqsubseteq x$. If $f^n(\perp) \sqsubseteq x$, then by monotonicity we get $f^{n+1}(\perp) \sqsubseteq f(x) = x$, completing the inductive proof. Hence, x is an upper bound of the chain

$$\perp \sqsubseteq f(\perp) \sqsubseteq f^2(\perp) \sqsubseteq f^3(\perp) \sqsubseteq \ldots.$$

Since $\mathsf{fix}(f)$ is the lub of this chain, we get
$\mathsf{fix}(f) \sqsubseteq x$.□

Example 4.4 Let Σ be an alphabet. Let Σ^* denote the set of finite sequences of letters from Σ. We use context-free grammar in the following examples to specify subsets of Σ^*.

(i) $E ::= \epsilon \mid Ea$ defines finite strings of a's including the empty string ϵ.

(ii) $E ::= a \mid bEb$ defines finite strings consisting (for all $n \geq 0$) of n b's followed by an a followed by another n b's.

Note that $(\mathcal{P}(\Sigma^*), \subseteq)$ is a cpo. Now for each case define $f : \mathcal{P}(\Sigma^*) \to \mathcal{P}(\Sigma^*)$ respectively by:

(i) $f(X) = \{\epsilon\} \cup X\{a\}$

(ii) $f(X) = \{a\} \cup \{b\}X\{b\}$.

Then f is continuous in both cases and its least fixed point is the desired solution.

4.1 Function space

Let D and E be cpo's and let $[D \to E]$ denote the set of all continuous functions from D to E ordered pointwise, i.e. for $f, g \in [D \to E]$ we put $f \sqsubseteq g$ iff $f(d) \sqsubseteq g(d)$ for all $d \in D$.

Then $([D \to E], \sqsubseteq)$ is easily seen to be a poset. We will show that it is in fact a cpo. The bottom element $\perp_{[D \to E]}$ is the constant map $\perp_{[D \to E]} : D \to E$ with $\perp_{[D \to E]}(d) = \perp_E$ for all $d \in D$.

Suppose we have a chain

$$f_0 \sqsubseteq f_1 \sqsubseteq f_2 \sqsubseteq \cdots$$

in $D \to E$.

What is the lub $\bigsqcup_n f_n$? Let $f : D \to E$ be defined by $f(d) = \bigsqcup_n f_n(d)$ for $d \in D$.

First, we check that f is indeed the lub of $\langle f_i \rangle_{i \geq 0}$. Since for each $j \geq 0$, $f_j(d) \sqsubseteq \bigsqcup_i f_i(d)$ for all $d \in D$, we have $f_j \sqsubseteq f$ and f is an upper bound of the chain. If g is any upper bound, then for each j we have $f_j \sqsubseteq g$ and hence $f_j(d) \sqsubseteq g(d)$ for all $d \in D$. Therefore $\bigsqcup_i f_i(d) \sqsubseteq g(d)$ for all $d \in D$. We conclude that $f \sqsubseteq g$, i.e. $f = \bigsqcup_i f_i$.

To conclude the proof, we have to check that f is continuous.

Monotonicity of $f : D \to E$.

Let $d \sqsubseteq d'$. Then $f_n(d) \sqsubseteq f_n(d')$ for all $n \geq 0$. But $f_n(d') \sqsubseteq f(d')$, for all $n \geq 0$, therefore $f(d')$ is an upper bound of the chain $\langle f_n(d)\rangle_{n\geq 0}$. Hence,

$$f(d) = \bigsqcup_{n\geq 0} f_n(d) \sqsubseteq f(d')$$

and we conclude that f is monotonic.

Continuity of $f : D \to E$.

Let $\langle x_j\rangle_{j\geq 0}$ be a chain in D. We must show that $f(\bigsqcup_j x_j) = \bigsqcup_j f(x_j)$. We have:

$$\begin{aligned}
f(\textstyle\bigsqcup_j x_j) &= \textstyle\bigsqcup_i f_i(\bigsqcup_j x_j) && \text{definition of } f,\\
&= \textstyle\bigsqcup_i \bigsqcup_j f_i(x_j) && f_i \text{ is continuous},\\
&= \textstyle\bigsqcup_j \bigsqcup_i f_i(x_j) && \text{see next slide!}\\
&= \textstyle\bigsqcup_j f(x_j) && \text{definition of } f.
\end{aligned}$$

This shows that f is continuous.

We have therefore proved:

Theorem 4.5
For any pair of cpo's D and E, $[D\to E]$, the set of continuous functions from D to E, ordered pointwise, is a cpo.

The equality $\bigsqcup_i \bigsqcup_j f_i(x_j) = \bigsqcup_j \bigsqcup_i f_i(x_j)$ needed above is a special instance of the proposition on the next slide.

Proposition 4.6
Let A be a cpo and let $a_{ij} \in A$ $(i, j \geq 0)$ be such that for each fixed i, $\langle a_{ij}\rangle_{j\geq 0}$ is increasing in j, i.e.

$$a_{i0} \sqsubseteq a_{i1} \sqsubseteq a_{i2} \sqsubseteq \dots$$

and for each fixed j, $\langle a_{ij}\rangle_{i\geq 0}$ is increasing in i i.e.

$$a_{0j} \sqsubseteq a_{1j} \sqsubseteq a_{2j} \sqsubseteq \dots.$$

Then

$$\bigsqcup_i \bigsqcup_j a_{ij} = \bigsqcup_j \bigsqcup_i a_{ij} = \bigsqcup_n a_{nn}.$$

Proof It is easy to check that each of the lubs in the above formula does exist. For example, $\langle a_{nn}\rangle_{n\geq 0}$ is increasing since $a_{nn} \sqsubseteq a_{n,n+1} \sqsubseteq a_{n+1,n+1}$.
For each n we have $a_{nn} \sqsubseteq \bigsqcup_j a_{nj}$ hence

$$\bigsqcup_n a_{nn} \sqsubseteq \bigsqcup_i \bigsqcup_j a_{ij}.$$

For each pair i, j, we have $a_{ij} \sqsubseteq a_{nn}$ where $n = \max(i, j)$. Hence $a_{ij} \sqsubseteq \bigsqcup_n a_{nn}$. Therefore $\bigsqcup_j a_{ij} \sqsubseteq \bigsqcup_n a_{nn}$. We conclude that

$$\bigsqcup_i \bigsqcup_j a_{ij} \sqsubseteq \bigsqcup_n a_{nn}.$$

It follows from anti-symmetry that

$$\bigsqcup_i \bigsqcup_j a_{ij} = \bigsqcup_n a_{nn}.$$

The other equation follows in a similar way.□

Recall that a continuous function $f : D \to E$ between cpo's is *strict* if $f(\perp_D) = \perp_E$. We denote the subset of strict functions in $[D\to E]$ by $[D\to_s E]$.

Exercise 4.7 $[D\to_s E]$ is a cpo.

Recall also that for any set S, the *flat domain* $S_\perp$ is the set $S \cup \{\perp\}$ with the partial order $\sqsubseteq$ defined by

$$a \sqsubseteq b \text{ iff } a = b \text{ or } a = \perp.$$

All chains in $S_\perp$ will have at most two different elements. Hence $S_\perp$ is cpo.

We can use the least fixed point theorem to solve recursive equations for functions.

Example 4.8 Consider the factorial function:

$$\mathsf{fac}(n) = \begin{cases} 1 & \text{if } n = 0\\ n \times \mathsf{fac}(n-1) & \text{if } n > 0 \end{cases}$$

This function is recursively defined in terms of itself. We want to overcome the circularity in this definition. We introduce a new function $F : [\mathbb{N}_\perp \to_s \mathbb{N}_\perp] \to [\mathbb{N}_\perp \to_s \mathbb{N}_\perp]$ defined by:

$$F(f)(n) = \begin{cases} 1 & \text{if } n = 0\\ n \times f(n-1) & \text{if } n > 0\\ \perp & \text{if } n = \perp \end{cases}$$

Exercise 4.9 F is a continuous function.

The map fac is the least fixed point of F. In fact, if $\perp_{[\mathbb{N}_\perp \to_s \mathbb{N}_\perp]} : \mathbb{N}_\perp \to \mathbb{N}_\perp$ denotes the least element of $[\mathbb{N}_\perp \to_s \mathbb{N}_\perp]$, i.e. the constant map with value $\perp$, then

$$F^i(\perp_{[\mathbb{N}_\perp \to_s \mathbb{N}_\perp]})(n) = \begin{cases} \mathsf{fac}(n) & \text{if } n < i\\ \perp & \text{otherwise.} \end{cases}$$

There are good reasons why we choose the least fixed point rather than any other fixed point.

Firstly, the least fixed point provides precisely the information contained in the recursive equation (and no more!).

Secondly, it is the canonical solution of the fixed point equation in a sense which we will make precise.

Definition 4.10 A *fixed point operator*, $F_{(-)}$, is a class of continuous functions

$$F_D : [D\to D] \to D$$

for each cpo D, such that for every continuous $f : D \to D$ we have $F_D(f) = f(F_D(f))$.

Exercise 4.11 Show that $\mathsf{fix} : [D\to D] \to D$ with $\mathsf{fix}(f) = \bigsqcup_i f^i(\perp)$ is continuous. Hence fix is a fixed point operator.

We say the fixed point operator $F_{(-)}$ is *uniform* if for any pair of continuous maps $f : D \to D$ and $g : E \to E$ and strict continuous map $h : D \to E$ which make the following commute:

$$\begin{tikzcd} D \arrow[r, "f"] \arrow[d, "h"'] & D \arrow[d, "h"] \\ E \arrow[r, "g"] & E \end{tikzcd}$$

(i.e. $h \circ f = g \circ h$) we have $F_E(g) = h(F_D(f))$.

Proposition 4.12

fix *is the unique uniform fixed point operator.*

Proof That fix is uniform is left as an exercise. Let $g : E \to E$ be a continuous function on a cpo $(E, \sqsubseteq)$. Consider the subset $D \subseteq E$ defined by $D = \{a \mid a \sqsubseteq \mathsf{fix}(g)\}$. Then $(D, \sqsubseteq)$ is a cpo and the restriction, $f : D \to D$ say, of g to D is continuous and has a unique fixed point $\mathsf{fix}(g)$. The inclusion map $i : D \to E$ (with $i(d) = d$ for all $d \in D$) is strict and continuous with:

$$\begin{tikzcd} D \arrow[r, "f"] \arrow[d, "i"'] & D \arrow[d, "i"] \\ E \arrow[r, "g"] & E \end{tikzcd}$$

Therefore, for any uniform fixed point operator F we must have $F_E(g) = i(F_D(f)) = F_D(f)$. But $F_D(f)$ is a fixed point of f so it must be equal to $\mathsf{fix}(g)$, i.e. $F_E(g) = \mathsf{fix}(g)$.$\square$

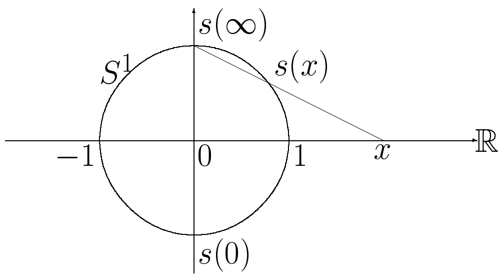
5 Exact real number computation

Consider a real number $r \in \mathbb{R}$ as the intersection of a shrinking nested sequence of rational intervals $\{r\} = \bigcap_n [a_n, b_n]$. The real number is *computable* if there is a master program which generates the intervals $[a_n, b_n]$.

The usual predicates such as $=$, $\leq$ and $<$ on computable real numbers are not decidable, e.g. we cannot decide in finite time if $r = 0$.

Since there is no test for zero, we have to deal with the problem of dividing say 1 by 0. Therefore, we allow ∞ to be the output of a program.

We work with the simple model of the extended real line $\mathbb{R}^* = \mathbb{R} \cup \{\infty\}$, represented by the unit circle S^1 with the map $s : \mathbb{R}^* \to S^1$:



The interval $[a, b] \subsetneqq S^1$ is defined to be the closed arc going anti-clockwise from a to b.

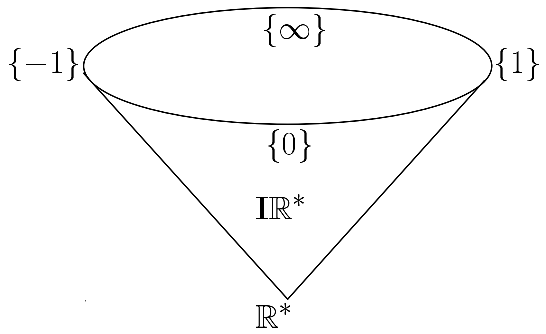
A suitable distance function on S^1 is defined as follows. For extended reals x and y which are both non-negative or both non-positive,

$$\rho(x, y) = \left| \frac{|x| - 1}{|x| + 1} - \frac{|y| - 1}{|y| + 1} \right|.$$

Otherwise, if x and y have different signs, then

$$\rho(x, y) = \min(\rho(x, 0) + \rho(0, y), \rho(x, \infty) + \rho(\infty, y)).$$

Expressions such as $\infty - \infty$, $0/0$ and 0^0 must be denoted by $\perp = \mathbb{R}^*$. This leads us to the domain $\mathbb{IR}^* = \{[a, b] \subsetneqq \mathbb{R}^*\} \cup \{\mathbb{R}^*\}$ of the closed intervals of $\mathbb{R}^*$ ordered by reverse inclusion. Any continuous function $f : \mathbb{R}^* \to \mathbb{R}^*$ has a canonical extension $f : \mathbb{IR}^* \to \mathbb{IR}^*$, given by $f(A) = \{f(x) \mid x \in A\}$.



5.1 Lft's and continued fractions

A *linear fractional transformation (lft)* is a map:

$$f : x \mapsto \frac{ax + c}{bx + d} : \mathbb{R}^* \to \mathbb{R}^*$$

with $a, b, c, d \in \mathbb{R}$ and $ad - bc \neq 0$. An lft is a continuous map of $\mathbb{R}^*$ with a continuous inverse; it is orientation preserving if $ad - bc > 0$ and orientation reversing if $ad - bc < 0$. The lft above can be represented up to scaling by the matrix $M = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$; a real number x is represented by the vector $\begin{pmatrix} x \\ 1 \end{pmatrix} \equiv \frac{x}{1}$ and ∞ by $\begin{pmatrix} 1 \\ 0 \end{pmatrix} \equiv \frac{1}{0}$, called *homogeneous coordinate*. Then composition of two lft's corresponds to multiplication of the two matrices. The lft's are linked to continued fractions.

A *continued fraction* is an infinite development

$$a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{a_3 + \ddots}}}$$

with $a_n, b_n \in \mathbb{Z}$. The rational number r_n, obtained from the above expression by replacing b_n with 0, is the nth rational approximation. If $\lim_{n \to \infty} r_n = r \in \mathbb{R}$ then the continued fraction *converges* to r.

Any real number has many continued fraction representations. The *regular* continued fraction expansion of a real number r is obtained as follows. If r is not an integer then $r = a_0 + 1/x_1$ where a_0 is the integer part of r and $x_1 > 1$. Similarly, if x_1 is not an integer we have $x_1 = a_1 + 1/x_2$ where $a_1 \geq 1$ is the integer part of x_1 and $x_2 > 1$. This scheme is repeated to obtain $a_2, a_3, \cdots$. We then have

$$r = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \ddots}}}$$

where $a_0 \in \mathbb{Z}$ and $a_i \in \mathbb{Z}^+$ for $i \geq 1$. The regular expansion is finite if and only if r is a rational number. For example, we have:

$$\sqrt{2} = 1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2 + \ddots}}}$$

$$\text{The golden ratio} = \frac{1+\sqrt{5}}{2} = 1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \ddots}}}$$

Any continued fraction expansion of a real number can be expressed as an infinite composition of lft's. In fact, a continued fraction expansion

$$r = a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{a_3 + \ddots}}}$$

of a real number r can be expressed as $r = f_0(x_0)$ with

$$x_0 = a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{a_3 + \ddots}}$$

and $f_0(x) = a_0 + \dfrac{b_0}{x}$. Iterating the above scheme, we obtain $r = f_0 f_1 \cdots f_n(x_n)$ with

$$x_n = a_{n+1} + \cfrac{b_{n+1}}{a_{n+2} + \cfrac{b_{n+2}}{a_{n+3} + \ddots}}$$

and $f_i(x) = a_i + \dfrac{b_i}{x}$ for $0 \leq i \leq n$.

One can therefore identify the original continued fraction for r with the infinite composition $f_0 f_1 f_2 \cdots$.

We now take $[0, \infty]$ as our base interval. Then any other rational interval $[c/d, a/b]$ can be expressed as $[c/d, a/b] = f[0, \infty]$ where

$$f : x \mapsto \frac{ax+c}{bx+d} : \mathbb{R}^* \to \mathbb{R}^*.$$

The set of all lft's is denoted by $\mathbb{M}$. Let $\mathbb{M}^+ \subseteq \mathbb{M}$ be those lft's whose coefficients are all non-negative or, equivalently, all non-positive. Check that for any lft f we have $f[0, \infty] \subseteq [0, \infty]$ if and only if $f \in \mathbb{M}^+$.

Proposition 5.1
For lft's f and g we have $f[0, \infty] \supseteq g[0, \infty]$ if and only if $g = f \circ h$ where $h \in \mathbb{M}^+$.

Proof We have $f[0, \infty] \supseteq g[0, \infty]$ if and only if $[0, \infty] \supseteq f^{-1} \circ g[0, \infty]$ if and only if $f^{-1} \circ g \in \mathbb{M}^+$. Therefore, we put $h = f^{-1} \circ g$. $\square$

It follows that for any shrinking sequence of nested intervals $[p_0, q_0] \supseteq [p_1, q_1] \supseteq [p_2, q_2] \supseteq \cdots$ we have $[p_n, q_n] = f_0 f_1 \cdots f_n[0, \infty]$ where $f_0 \in \mathbb{M}$ and $f_i \in \mathbb{M}^+$ for $1 \leq i \leq n$. Therefore, the sequence can be expressed as an infinite composition of lft's, or equivalently infinite product of matrices, $f_0 f_1 f_2 \cdots$.

Note that we have here a generalization of IFS with, in particular, a countable set of continuous (not necessarily contracting) maps on $\mathbb{R}^*$.

We have therefore shown that any real number can be represented as the intersection $\bigcap_{n \geq 0} f_0 f_1 f_2 \cdots f_n[0, \infty]$ with $f_0 \in \mathbb{M}$ and $f_i \in \mathbb{M}^+$ $(i \geq 1)$ such that f_n has integer coefficients for all $n \geq 0$. If

$$f_n : x \mapsto \frac{a_n x + c_n}{b_n x + d_n}$$

then in matrix notation, the real number can be expressed as the infinite product

$$\begin{pmatrix} a_0 & c_0 \\ b_0 & d_0 \end{pmatrix} \begin{pmatrix} a_1 & c_1 \\ b_1 & d_1 \end{pmatrix} \begin{pmatrix} a_2 & c_2 \\ b_2 & d_2 \end{pmatrix} \begin{pmatrix} a_3 & c_3 \\ b_3 & d_3 \end{pmatrix} \cdots.$$

We call this a *normal product*. If the first matrix is in $\mathbb{M}$ we call it a *signed normal product (snp)*; otherwise it is called an *unsigned normal product (unp)*.

The information, $\mathbf{Info}(K)$, given by an lft K is an interval of $\mathbb{R}^*$ defined by: $\mathbf{Info}(M) = M([0, \infty])$.

The first matrix tells us that the result is contained in the interval $\left[\frac{a_0}{b_0}, \frac{c_0}{d_0}\right]$ or $\left[\frac{c_0}{d_0}, \frac{a_0}{b_0}\right]$ according to the sign of the determinant of the matrix. The other matrices will successively refine this interval to give better and better approximations to the real number. In analogy with the decimal representation of real numbers, the first matrix is called a *sign* matrix whereas the other matrices are *digit* matrices.

Example 5.2
Our continued fraction for $\sqrt{2}$

$$\sqrt{2} = 1 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{2+ \ddots}}}$$

gives us the normal product

$$\sqrt{2} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} \cdots .$$

We therefore obtain the following shrinking nested sequence of intervals for approximating $\sqrt{2}$:

$$\begin{aligned}
\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}[0,\infty] &= [1,\infty] \\
\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}[0,\infty] &= [1,1.5] \\
\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}[0,\infty] &= [1.4,1.5] \\
\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}[0,\infty] &= [1.4,\tfrac{17}{12}] \\
&\cong [1.4,1.46] \\
\cdots\cdots\cdots\cdots\cdots\cdots &= \cdots\cdots
\end{aligned}$$

Recall that the intersection of all these intervals is precisely $\{\sqrt{2}\}$.

We can represent rational functions with lft's of two arguments. These are of the form $f : \mathbb{R}^* \times \mathbb{R}^* \to \mathbb{R}^*$ where

$$f(x,y) = \frac{axy + cx + ey + g}{bxy + dx + fy + h}$$

with integer coefficients which in homogenous coordinates can be represented by a *tensor*:

$$\left[\begin{pmatrix} x \\ x' \end{pmatrix}, \begin{pmatrix} y \\ y' \end{pmatrix}\right] \mapsto \begin{pmatrix} a & c & e & g \\ b & d & f & h \end{pmatrix} \left[\begin{pmatrix} x \\ x' \end{pmatrix}, \begin{pmatrix} y \\ y' \end{pmatrix}\right].$$

If we fix any of the two arguments of an lft with two arguments we obtain a usual lft with one argument. We can express the following basic arithmetic operations by tensors:

$$\begin{aligned}
\begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}(x,y) &= x + y \\
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}(x,y) &= x \times y \\
\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}(x,y) &= x \div y
\end{aligned}$$

The set of tensors and the subset of non-negative tensors are denoted by $\mathbb{T}$ and $\mathbb{T}^+$ respectively. The information given by a tensor is the interval of $\mathbb{R}^*$ defined by: $\mathbf{Info}(f) = f([0,\infty],[0,\infty])$.

Check that $\mathbf{Info}(f) = [\min(\frac{a}{b},\frac{c}{d},\frac{e}{f},\frac{g}{h}), \max(\frac{a}{b},\frac{c}{d},\frac{e}{f},\frac{g}{h})]$ or $[\max(\frac{a}{b},\frac{c}{d},\frac{e}{f},\frac{g}{h}), \min(\frac{a}{b},\frac{c}{d},\frac{e}{f},\frac{g}{h})]$, in which any fraction of the form $0/0$ should be ignored.

Signed and unsigned normal products are generalized to signed and unsigned expression trees by allowing tensors as follows. Denote the set of all vectors and all non-negative vectors by $\mathbb{V}$ and $\mathbb{V}^+$ respectively.

Definition 5.3 A *signed expression tree (sext)* and an *unsigned expression tree (uext)* are finite or infinite binary trees defined by:

$$\begin{aligned}
sext &::= V \mid M(uext) \mid T(uext, uext) \\
&\quad\ \text{where } V \in \mathbb{V},\ M \in \mathbb{M},\ T \in \mathbb{T} \\
uext &::= V \mid M(uext) \mid T(uext, uext) \\
&\quad\ \text{where } V \in \mathbb{V}^+,\ M \in \mathbb{M}^+,\ T \in \mathbb{T}^+
\end{aligned}$$

A finite truncation of such an expression tree corresponds to a finite subtree such that the removed nodes are replaced by the base interval $[0,\infty]$. Any such truncation denotes the composition of a finite number of lft's applied to $[0,\infty]$. Hence, it denotes a compact interval. The intersection of all compact intervals obtained gives the value of the expression tree.

In order to compute the value of an expression tree, we need a way to transform such a tree into a snp. We first need the rules for composition of lft's of different dimensions.

We say that a tensor T is composed of two matrices $T = (T_0, T_1)$: If $T = \begin{pmatrix} a & c & e & g \\ b & d & f & h \end{pmatrix}$ then $T_0 = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$ and $T_1 = \begin{pmatrix} e & g \\ f & h \end{pmatrix}$. Similarly, a matrix M consists of two vectors $M = (M_0, M_1)$.

Definition 5.4 The *dot product, left product, and the right product*, denoted respectively by $\cdot$, $\circ_l$ and $\circ_r$, are defined by:

$$\begin{aligned}
(M \cdot V)_i &= \textstyle\sum_{j=0,1} M_{ij} V_j & T \circ_r V &= (T_0 \cdot V, T_1 \cdot V) \\
(M \cdot N) &= (M \cdot N_0, M \cdot N_1) & T \circ_r M &= (T_0 \cdot M, T_1 \cdot M) \\
(M \cdot T) &= (M \cdot T_0, M \cdot T_1) & T \circ_l V &= T^t \circ_r V \\
& & T \circ_l M &= (T^t \circ_r M)^t
\end{aligned}$$

Here, T^{t} is the *transpose* of T obtained by swapping the two middle columns of T, in other words, we have $T^{\mathsf{t}} = \begin{pmatrix} a & e & c & g \\ b & f & d & h \end{pmatrix}$ if $T = \begin{pmatrix} a & c & e & g \\ b & d & f & h \end{pmatrix}$.

Let $\mathcal{T} = T(\mathcal{T}_1, \mathcal{T}_2)$ be a signed expression tree having the tensor T as its root node with left and right unsigned expression subtrees $\mathcal{T}_1$ and $\mathcal{T}_2$ respectively. In order to obtain the sign matrix of the snp, we choose M such that $M([0,\infty]) \supseteq T([0,\infty],[0,\infty])$. Then, M is the sign matrix of the snp. We now replace T in $\mathcal{T}$ by $M^{-1} \cdot T$ which has non-negative coefficients because $M([0,\infty]) \supseteq T([0,\infty],[0,\infty])$. This is called *emission*.

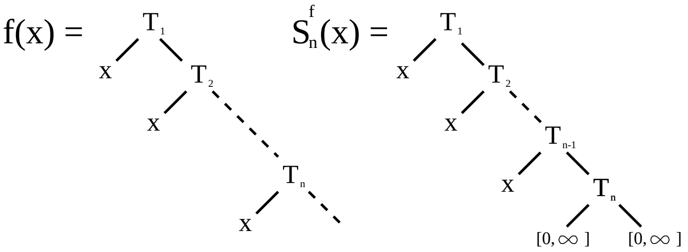
In order to have a more precise result, we absorb some information in the tensor T. That means emitting matrices M_1 and M_2 from $\mathcal{T}_1$ and $\mathcal{T}_2$ and replacing T with $(T \circ_l M_1) \circ_r M_2$. This gives an incremental algorithm to compute an expression tree.

We can also represent all basic elementary functions such as $f = \tan, \arctan, \log$ and $\exp$ in terms of expression trees with an entry x of the form given in the figure below.

Let x be a real number. The value $f(x)$ of the infinite expression tree in the figure below is given in terms of the truncated expression trees $S_n^f(x)$ by the intersection of a nested sequence of closed intervals $f(x) = \bigcap_n S_n^f(x)$, where

$$S_n^f(x) = \{T_1(x, T_2(x, , \cdots, T_n(y, z) \cdots)) | y, z \in [0, \infty]\}.$$

In general, $f(x)$ may be a closed interval.



For example, the function arctan has the following continued fraction expansion:

$$\arctan x = \cfrac{x}{1 + \cfrac{\frac{x^2}{3}}{1 + \cfrac{\frac{4x^2}{15}}{1+ \ddots}}},$$

which can be transformed into

$$\arctan x = \prod_{n=1}^{\infty} \begin{pmatrix} 0 & x \\ n^2 x & 2n-1 \end{pmatrix}.$$

This is an infinite composition of lft's with non-negative coefficients but now each lft has x as a parameter, i.e., it is a function of two arguments. For example, when $n = 1$, we have:

$$\begin{pmatrix} 0 & x \\ x & 1 \end{pmatrix} \begin{pmatrix} y \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} [\begin{pmatrix} x \\ 1 \end{pmatrix}, \begin{pmatrix} y \\ 1 \end{pmatrix}],$$

as it can be directly checked using the left and right absorption rules.

Thus the infinite product can be rewritten as an infinite expression tree:

$$\arctan x = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \left[x, \begin{pmatrix} 0 & 1 & 0 & 0 \\ 4 & 0 & 0 & 3 \end{pmatrix} \left[x, \begin{pmatrix} 0 & 1 & 0 & 0 \\ 9 & 0 & 0 & 5 \end{pmatrix} [x, \cdots]\right]\right]$$

We need an algorithm to compute the value of an expression tree $\mathcal{T}(x)$ when the entry x corresponds to an extended real number represented by an unp $D_1 D_2 \cdots$.

Let $\mathcal{T}_n(x)$ denote the *truncation* of the $\mathcal{T}(x)$ at depth n, i.e., any node with n edges away from the root node is replaced by $[0, \infty]$. In the case of the expression tree in the figure on page 65, $\mathcal{T}_n(x)$ is simply $S_n^f(x)$.

We let $(x)_n$ denote the truncation at depth n of the tree representing x. In the case of an unp,

$$(D_1 D_2 \cdots)_n = D_1 \cdots D_n [0, \infty].$$

We can compute the information given by the finite tree $\mathcal{T}_n((x)_n)$ using the absorption and emission rules we have defined.

The sequence

$$\langle \mathcal{T}_n((x)_n) \rangle_{n \in \mathbb{N}}$$

is a nested shrinking sequence of compact intervals which tends to the value of the expression tree $\mathcal{T}(D_1 D_2 \cdots)$.

Note that the algorithm is incremental, i.e., it is not necessary to recompute $\mathcal{T}_n((x)_n)$ in order to evaluate $\mathcal{T}_{n+1}((x)_{n+1})$.

5.2 Exact Floating Point

So far our representation allows arbitrary normal products of integer matrices $M_0 M_1 M_2 \cdots$ with $M_0 \in \mathbb{M}$ and $M_i \in \mathbb{M}^+$ for $i \geq 1$. This, in practice, results in some major problems. Firstly, intervals will be refined at an arbitrary rate, making any analysis of complexity of algorithms practically impossible. Secondly, matrix multiplication can quickly produce huge integers in a matrix quite disproportionate to the information contained it.

In analogy with floating point formats, where number representations in a given base are generated by two sign symbols and a finite number of digits, we restrict the sign and digit matrices to a finite set of specific matrices. Sign matrices are rotations of S^1 whereas digit matrices are contracting maps with respect to the metric ρ on $\mathbb{R}^*$.

We start with sign matrices. The information in sign matrices must overlap and cover S^1. If we further assume that they have the same length with respect to ρ and are evenly placed on S^1, then they will be generated by rotations of S^1. The lft $\phi_{\exp i\theta} : x \mapsto \dfrac{x \cos\frac{\theta}{2} + \sin\frac{\theta}{2}}{-x \sin\frac{\theta}{2} + \cos\frac{\theta}{2}}$ rotates S^1 by θ.

Moreover, $\phi_{\exp i\theta}$ generates a finite cyclic group iff θ is a rational multiple of 2π. Our choice will be further restricted if the lft is required to have integer coefficients.

Proposition 5.5 *Suppose θ is a non-integral rational multiple of 2π. Then the lft $\phi_{\exp i\theta}$ will have integer coefficients iff $\theta = \frac{\pi}{2}$ or $\theta = \pi$.*

For $\theta = \pi$, we get the cyclic group of order 2 consisting of $\phi_{\exp i\pi} : x \mapsto -\frac{1}{x}$ and the identity lft $\mathrm{Id} : x \mapsto x$. This gives the two intervals $\mathsf{info}(\phi_{\exp i\pi}) = [\infty, 0]$ and $\mathsf{info}(\mathrm{Id}) = [0, \infty]$ which are not overlapping. For $\theta = \pi/2$ we get the cyclic group of order 4 with elements

$$\phi_{\exp \frac{i\pi}{2}} : x \mapsto \frac{x+1}{-x+1}, \quad \phi_{\exp i\pi} : x \mapsto -\frac{1}{x},$$

$$\phi_{\exp \frac{3\pi i}{2}} : x \mapsto \frac{x-1}{x+1}, \quad \mathrm{Id} : x \mapsto x,$$

with information $[1, -1], [\infty, 0], [-1, 1]$ and $[0, \infty]$ respectively. The simplest matrices representing these lft's are, respectively:

$$S_\infty = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \quad S_- = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

$$S_0 = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \quad S_+ = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

We therefore take these as our sign matrices.

We now select an appropriate set of digit matrices from $\mathbb{M}^+$. Since compositions of digit matrices are required to represent shrinking sequences of intervals, we will look for matrices which contract distances in $[0, \infty]$ with respect to the metric ρ. Digit matrices must overlap and cover $[0, \infty]$.

Note that S_0 is a homeomorphism from $[0, \infty]$ to its image $S_0[0, \infty] = [-1, 1]$, i.e. it is a 1-1, onto and continuous map with a continuous inverse. Let $\phi \in \mathbb{M}^+$ and consider the restriction $\phi : [0, \infty] \to [0, \infty]$. Then $S_0 \phi S_0^{-1}$ is a homeomorphism from $[-1, 1]$ onto itself. For $x, y \in [0, \infty]$ we have $\rho(x, y) = |S_0(x) - S_0(y)|$ and we get:

Proposition 5.6 *The map $\phi : [0, \infty] \to [0, \infty]$ is contracting with respect to the ρ-metric iff*

$$S_0 \phi S_0^{-1} : [-1, 1] \to [-1, 1]$$

is contracting with respect to the Euclidean metric.

It follows that for any base $b > 1$, the signed digit representation on $[-1, 1]$ in base b induces via the homeomorphism S_0 a suitable set of digit matrices in $\mathbb{M}^+$.

The signed digit system in base $b > 1$ in $[-1, 1]$ is generated by an IFS on $[-1, 1]$ with contracting maps

$$f_k : [-1, 1] \mapsto [-1, 1]$$
$$x \mapsto \frac{x+k}{b}$$

with

$$k \in \mathrm{Dig}(b) = \{-b+n, b-n | n \in \mathbb{N}, 1 \leq n \leq \lfloor b \rfloor\},$$

where $\lfloor b \rfloor$ is the integral part of b. Here, b can be allowed to be a rational or an irrational number.

The case $b = 3/2$ was considered by Brouwer and the case $b = \frac{1+\sqrt{5}}{2}$, the golden ratio, has also been studied recently.

We now define the digit matrices in base b as the IFS on $[0, \infty]$ with ρ-contracting maps:

$$D_k = S_0^{-1} f_k S_0 = \begin{pmatrix} 1+b+k & -1+b+k \\ -1+b-k & 1+b-k \end{pmatrix}.$$

For example, for base 2, we have the four sign matrices S_+, S_∞, S_{-1} and S_0 together with the three digit matrices:

$$D_{-1} = \begin{pmatrix} 1 & 0 \\ 1 & 2 \end{pmatrix} \quad D_0 = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix} \quad D_1 = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}.$$

Exact floating point in base b is defined as the representation of real numbers by infinite composition of lft's, or, equivalently, infinite product of matrices, such that the first matrix is one of the sign matrices above and the subsequent matrices are digit matrices. For each finite composition $D_{k_1} D_{k_2} \cdots D_{k_n}$ of digit matrices we have:

$$S_0 D_{k_1} D_{k_2} \cdots D_{k_n}[0, \infty] = f_{k_1} f_{k_2} \cdots f_{k_n}[-1, 1].$$

Therefore, for every infinite composition of digit matrices, we obtain

$$\bigcap_{n \geq 0} S_0 D_{k_1} D_{k_2} \cdots D_{k_n}[0, \infty]$$
$$= \bigcap_{n \geq 0} f_{k_1} f_{k_2} \cdots f_{k_n}[-1, 1].$$

This gives us:

Proposition 5.7 *A real number with signed digit expansion $.k_1 k_2 k_3 \cdots$ (with $k_j \in \mathrm{Dig}(b)$ for $j \geq 1$) is represented in exact floating point by the infinite product*

$$S_0 D_{k_1} D_{k_2} D_{k_3} \cdots.$$

Thus we obtain a reasonable data-type for representing real numbers, with which we can obtain provably correct algorithms for exact real number computation.

6 A domain-theoretic model of geometry

A *topology* on a space X is given by a set of subsets of X, called the *open* subsets, which is closed under infinite unions and finite intersections. The empty set $\emptyset$, being the empty union, and X, being the empty intersection, are open. A *closed* subset is one whose complement is open. The *boundary* ∂A of a set $A \subseteq X$ is the set of points $x \in X$ such that every open set O with $x \in O$ intersects both A and its complement A^c. Check that closed sets are closed under taking finite unions and arbitrary intersections. The set X with its given topology is called a topological space.

For example the Euclidean topology on $\mathbb{R}^n$ has as non-empty open sets the union of open balls $O = \bigcup_{i\in I} O(c_i, r_i)$, where the open ball of centre c and radius r is the set $O(c,r) = \{x \in \mathbb{R}^n \mid |x - c| < r\}$. The closed ball of centre c and radius r is the set $C(c,r) = \{x \in \mathbb{R}^n \mid |x - c| \le r\}$. Check that $O(c,r)$ is open and $C(c,r)$ is closed.

Exercise 6.1 (i) What is the boundary of $O(c,r)$? How about that of $C(c,r)$?

(ii) Show that the Cantor set is closed. What is its boundary?

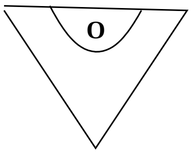
6.1 Semi-decidable predicate

Membership in an open set is a *semi-decidable* predicate or an *observable* property: we can confirm in finite time if a point belongs to an open set. For example consider a point $x \in \mathbb{R}^2$, given as the intersection of a shrinking nested sequence of rational rectangles and an open rational ball (rational centre and rational radius) $O(c,r)$. If $x \in O(c,r)$ then one of the rational rectangles will be contained in $O(c,r)$ and we can confirm that in finite time. However, if x lies on the boundary of $O(c,r)$, then we cannot verify it in finite time.

Let $(P, \sqsubseteq)$ be a poset. A non-empty subset $A \subseteq P$ is *directed* if for any pair of elements $a, b \in A$, there exists $c \in A$ with $a, b \sqsubseteq c$. A *directed complete partial order* or a *dcpo* is a poset in which every directed subset has a lub. Check that every dcpo with bottom is a cpo. Let $(D, \sqsubseteq)$ be a dcpo. The *Scott* topology on D is defined by characterising its closed sets as follows. A subset $C \subseteq D$ is closed if

- C is downward closed, i.e. $y \in C \;\&\; x \sqsubseteq y \;\Rightarrow\; x \in C$; and
- whenever $A \subseteq C$ is a directed subset, $\bigsqcup A \in C$.

Exercise 6.2 Check that a subset $O \subseteq D$ of a dcpo D is open if (i) O is upward closed, i.e. $x \in O \;\&\; x \sqsubseteq y \;\Rightarrow\; y \in O$; and (ii) whenever $A \subseteq D$ is a directed subset with $\bigsqcup A \in O$, then $A \cap O \neq \emptyset$.



We say that a sequence $(x_n)_{n\ge 0}$ of points of a topological space X converges to a point $x \in X$, if for every open set O with $x \in O$ there exists $N \ge 0$ such that $x_n \in O$ for all $n \ge N$; we write this as $\lim_{n\to\infty} x_n = x$.

Exercise 6.3 (i) Check that in Euclidean spaces, if a sequence has a limit then the limit is unique. Show that a convergent sequence in a dcpo may have more than one limit in general.

(ii) Show that a closed subset C of a topological space contains all its limit points, i.e. whenever $\lim_{n\to\infty} x_n = x$, with $x_n \in C$ for all $n \ge 0$, then $x \in C$.

(iii) Show by an example that membership in a closed set in Euclidean spaces is not semi-decidable.

6.2 Continuity and Computability

A function $f : X \to Y$ of topological spaces is *continuous* at $a \in X$ if for each open subset $O \subseteq Y$ with $f(a) \in O$, there exists an open set $U \subseteq X$ with $a \in U$ and $f(U) \subseteq O$. We say f is continuous if it is continuous at all points of X.

Exercise 6.4 (i) Check that a function $f : \mathbb{R}^n \to \mathbb{R}^n$ is continuous with respect to the Euclidean topology iff it preserves limits of convergent sequences, i.e. iff whenever $\lim_{n\to\infty} x_n = x$ then $\lim_{n\to\infty} f(x_n) = f(x)$.

(ii) Check that a function $f : D \to E$ between dcpo's D and E is continuous with respect to the Scott topology iff f is monotone and preserves the lubs of directed sets.

Recall the notion of computable real number (p. 54).

Exercise 6.5 Show that $r \in \mathbb{R}$ is computable iff there exists a master program which on input $n \in \mathbb{N}$ outputs a rational number q_n with $|q_n - r| \le 1/2^n$.

With respect to Exercise 6.5, we say that there exists an *effective* sequence $(q_n)_{n\ge 0}$ of rational numbers with $\lim_{n\to\infty} q_n = r$ and $|q_n - r| \le 1/2^n$; thus q_n gives an approximation with precision $1/2^n$ to r.

Intuitively, we expect a "computable" real-valued function $f : \mathbb{R} \to \mathbb{R}$ to take a computable real number to a computable real number: If $\lim_{n\to\infty} q_n = r$ as in Exercise 6.5, then we expect $\lim_{n\to\infty} f(q_n) = f(r)$, i.e. a sequence of rational approximations to r should be mapped to a sequence of approximations converging to $f(r)$. This is also called Scott's thesis: *A computable function is always continuous.*

6.3 Non-computability of predicates and operations in classical geometry

It is now easy to see that basic predicates and operations in classical geometry are not computable.

We consider the two element Boolean set $\{\mathsf{tt}, \mathsf{ff}\}$ with its discrete topology i.e. every subset is open and thus observable.

Take the membership predicate $\epsilon_D : \mathbb{R}^2 \to \{\mathsf{tt}, \mathsf{ff}\}$ of a simple object such as the unit disk

$$D = \{x \in \mathbb{R}^2 \mid |x| \leq 1\} :$$

$$\epsilon_D : x \mapsto \begin{cases} \mathsf{tt} & x \in D \\ \mathsf{ff} & x \notin D \end{cases}$$

Exercise 6.6 Show that ϵ_D is not continuous at any point on the boundary of D.

Therefore, the membership predicate of even a simple object is non-continuous and thus non-computable.

We can directly see the non-computability of the classical membership predicate. If $x \in D$ is on the boundary of D, i.e. if $|x| = 1$, and if x is approximated step by step by a shrinking nested sequence of rational rectangles (rectangles with rational vertices), then we cannot decide in finite time that $x \in D$.

Similarly, consider the intersection operator as a binary operator on the collection $\mathcal{C}(\mathbb{R}^n)$ of bounded closed subsets of $\mathbb{R}^n$ equipped with the Hausdorff distance d_H defined as before by $d_H(C, D) = \max(\max_{d\in D} \min_{c\in C} |c - d|, \max_{c\in C} \min_{d\in D} |c - d|)$, with the convention that $d_H(\emptyset, \emptyset) = 0$ and for $C \neq \emptyset$, $d_H(\emptyset, C) = \infty$:

$$\begin{array}{rll} - \cap - : \mathcal{C}(\mathbb{R}^n) \times \mathcal{C}(\mathbb{R}^n) & \to & \mathcal{C}(\mathbb{R}^n) \\ (A, B) & \mapsto & A \cap B \end{array}$$

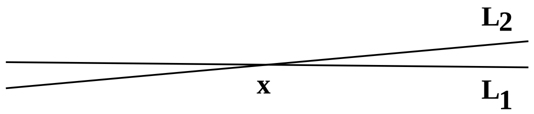
Exercise 6.7 Check that $- \cap -$ is discontinuous whenever A and B just touch each other.

Intuitively, when two objects touch each other, then by an arbitrary small perturbation their intersection can become empty.

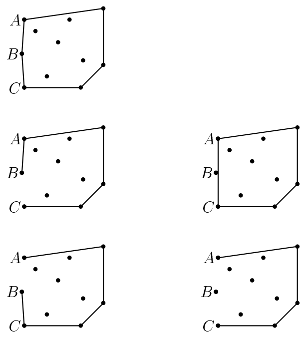
The non-continuity of the basic predicates and operations creates a foundational problem in computation, which has so far been essentially neglected. In fact, in order to construct a sound computational model for solids and geometry, one needs a framework in which these elementary building blocks are continuous and computable.

In practice, correctness of algorithms in computational geometry is usually proved using the Real RAM machine model of computation, in which comparison of real numbers is considered to be decidable. Since this model is not realistic, correct algorithms, when implemented, turn into unreliable programs.

A simple example is provided by computing, in any floating point format, first the intersection point x in the plane of two straight lines L_1 and L_2 meeting under a small angle, and then computing the minimum distance $d(x, L_1)$ and $d(x, L_2)$ from x to each of the two lines. In general, $d(x, L_1)$ and $d(x, L_2)$ are both positive and distinct, contradicting the expected values $d(x, L_1) = d(x, L_2) = 0$.



A more sophisticated example is given by the implementation in floating point of any algorithm to compute the convex hull of a finite number of points in the plane (i.e. the smallest convex set containing the points). If there are three nearly collinear points A, B, C as in the picture, then depending upon the floating point format, the program can give, instead of the edges AB and BC as in the first figure below, any of the following as in the next four figures:
(i) AB only. (ii) AC only.
(iii) BC only. (iv) none of them.



In any of the above four cases, we get a logical inconsistency as the edges returned by the program do not give the correct convex hull and in the cases (i), (iii) and (iv) do not give a closed polygon at all.

The solid modelling framework provided by classical analysis, which allows discontinuous behaviour and comparison of exact real numbers, is not realistic as a model of our interaction with the physical world in terms of measurement and manufacturing.

For example, as far as the process of design, manufacturing or any other practical application is concerned, the closed ball of radius one and the open ball of radius one are exactly the same despite the fact that these two objects are distinguished in classical geometry.

As we have seen the classical framework is also not realistic as a basis for the design of algorithms implemented on realistic machines, which can only deal with finite data.

A robust algorithm is one whose correctness is proved with the assumption of a realistic computable model. Domain theory defines precisely what it means, in the context of the realistic model of computation, to compute objects belonging to non-countable sets such as the set of real numbers.

Here, we use a domain-theoretic approach to develop the foundation of a computable framework for solid modelling and computational geometry.

6.4 The domain of geometric objects

In the domain-theoretic model, the basic predicates, such as membership and subset inclusion, and operations, such as union and intersection, are continuous and computable. The model provides a methodology for developing robust geometric algorithms and enables us to capture the uncertainties of input data in CAD (Computer Aided Design) situations.

For any subset S of a topological space, $\overline{S}$, S°, ∂S and S^c denote respectively the closure (i.e. the intersection of all closed sets containing S), the interior (i.e. the union of all open sets contained in S), the boundary and the complement of S.

Given any proper subset of $S \subseteq \mathbb{R}^n$ (i.e. $\emptyset \neq S \neq \mathbb{R}^n$), observe that the classical membership predicate $\in_S\colon \mathbb{R}^n \to \{\mathsf{tt}, \mathsf{ff}\}$ is continuous at every point in the interior S° and every point in $S^{c\circ}$, called the *exterior*, but not on ∂S. It therefore makes sense from a computational viewpoint to redefine the membership predicate as the function:

$$\begin{aligned} \in'_S\colon \mathbb{R}^n &\to \{\mathsf{tt}, \mathsf{ff}\}_\perp \\ x &\mapsto \begin{cases} \mathsf{tt} & \text{if } x \in S^\circ \\ \mathsf{ff} & \text{if } x \in S^{c\circ} \\ \perp & \text{otherwise.} \end{cases} \end{aligned}$$

Here, $\{\mathsf{tt}, \mathsf{ff}\}_\perp$ is the three element poset with least element $\perp$ and two incomparable elements tt and ff.

Exercise 6.8 (i) Check that in the Scott topology of $\{\mathsf{tt}, \mathsf{ff}\}_\perp$ the two subsets $\{\mathsf{tt}\}$ and $\{\mathsf{ff}\}$ are open sets but in contrast the subset $\{\perp\}$ is not open.

(ii) Show that $\in'_S$ is continuous for any subset S.

We therefore call $\in'_S$ the *continuous membership predicate.*

Exercise 6.9 Check that two subsets have the same continuous membership predicate iff they have the same interior and the same exterior (interior of complement).

Thus in this model the closed unit ball and the open unit ball are not distinguished as geometric objects.

By analogy with general set theory for which a set is completely defined by its membership predicate, we can define a geometric object in $\mathbb{R}^n$ to be any continuous map of type $\mathbb{R}^n \to \{\mathsf{tt}, \mathsf{ff}\}_\perp$.

The definition of the geometric domain is then consistent with the requirement that a computable membership predicate has to be continuous and that membership of the boundary of a set is in general not semi-decidable and not observable.

Note that a geometric object, given by a continuous map $f : \mathbb{R}^n \to \{\mathsf{tt}, \mathsf{ff}\}_\perp$, is determined precisely by two disjoint open sets, namely $f^{-1}(\mathsf{tt})$ and $f^{-1}(\mathsf{ff})$. Moreover, the interior $(f^{-1}(\mathsf{tt}) \cup f^{-1}(\mathsf{ff}))^{c^\circ}$ of the complement of the union of these two open sets can be non-empty.

If we now consider a second continuous function $g : \mathbb{R}^n \to \{\mathsf{tt}, \mathsf{ff}\}_\perp$ with $f \sqsubseteq g$, then we have:

$$f^{-1}(\mathsf{tt}) \subseteq g^{-1}(\mathsf{tt}) \qquad \& \qquad f^{-1}(\mathsf{ff}) \subseteq g^{-1}(\mathsf{ff}).$$

This means that a more defined geometric object has a larger interior and a larger exterior. We can think of the pair $f^{-1}(\mathsf{tt}), f^{-1}(\mathsf{ff})$ as the points of the interior and the exterior of a geometric object as determined at some finite stage of computation. At a later stage, we obtain a more refined approximation g which gives more information about the geometric object, i.e. more points of its interior and more points of its exterior.

Definition 6.10 The *domain of geometric objects* or the *solid domain* $(\mathbf{S}\mathbb{R}^n, \sqsubseteq)$ of $\mathbb{R}^n$ is the set of ordered pairs (A, B) of disjoint open subsets of $\mathbb{R}^n$ endowed with the information order:

$$(A_1, B_1) \sqsubseteq (A_2, B_2) \iff A_1 \subseteq A_2 \text{ and } B_1 \subseteq B_2.$$

An element (A, B) of $\mathbf{S}\mathbb{R}^n$ is called a *partial geometric object* or a *partial solid*. The sets A and B are intended to capture, respectively, the interior and the exterior of a geometric object, possibly, at some finite stage of computation.

We say that two dcpo's are *isomorphic* if there is a continuous 1-1 and onto map from one to the other with a continuous inverse.

Theorem 6.11 *The poset $(\mathbf{S}\mathbb{R}^n, \sqsubseteq)$ is a directed complete partial order with $\bigsqcup_{i\in I}(A_i, B_i) = (\bigcup_{i\in I} A_i, \bigcup_{i\in I} B_i)$ and is isomorphic with the function space $\mathbb{R}^n \to \{\mathsf{tt}, \mathsf{ff}\}_\perp$.*

Proof It is easy to check that if $(A_i, B_i)_{i\in I}$ is a directed set of disjoint open subsets then $\bigcup_{i\in I} A_i$ and $\bigcup_{i\in I} B_i$ are disjoint and $\bigsqcup_{i\in I}(A_i, B_i) = (\bigcup_{i\in I} A_i, \bigcup_{i\in I} B_i)$ is the lub of $(A_i, B_i)_{i\in I}$. The function $\Gamma : (\mathbb{R}^n \to \{\mathsf{tt}, \mathsf{ff}\}_\perp) \to \mathbf{S}(\mathbb{R}^n)$ given by $\Gamma(f) = (f^{-1}(\{\mathsf{tt}\}), f^{-1}(\{\mathsf{ff}\}))$ provides the isomorphism. $\square$

By duality of open and closed sets, $(\mathbf{S}\mathbb{R}^n, \sqsubseteq)$ is also isomorphic with the collection of ordered pairs (A, B) of closed subsets of $\mathbb{R}^n$ with $A \cup B = \mathbb{R}^n$ with the information ordering: $(A_1, B_1) \sqsubseteq (A_2, B_2) \iff A_2 \subseteq A_1$ and $B_2 \subseteq B_1$.

Proposition 6.12 *The partial geometric object $(A, B) \in (\mathbf{S}\mathbb{R}^n, \sqsubseteq)$ is a maximal element iff $A = B^{c\circ}$ and $B = A^{c\circ}$.*

Proof Let (A, B) be maximal. Since A and B are disjoint open sets, it follows that $A \subseteq B^{c\circ}$. Hence, $(A, B) \sqsubseteq (B^{c\circ}, B)$ and thus $A = B^{c\circ}$. Similarly, $B = A^{c\circ}$. This proves the "only if" part. For the "if" part, suppose that $A = B^{c\circ}$ and $B = A^{c\circ}$. Then, any proper open superset of A will have non-empty intersection with B and any proper open superset of B will have non-empty intersection with A. It follows that (A, B) is maximal. $\square$

An open set is *regular* if it is the interior of its closure; dually, a closed set is regular if it is the closure of its interior.

Exercise 6.13 Show that the interior of a closed set is a regular open set.

Corollary 6.14 *If (A, B) is a maximal element, then A and B are regular open sets.*

Proof Note that A is the interior of the closed set B^c and is, therefore, regular; similarly B is regular.$\square$

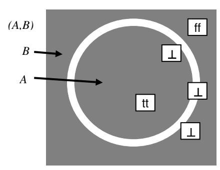
6.5 Predicates and Operations on Solids

Partial geometric objects, and, more generally, domain-theoretically defined data types allow us to capture partial, or uncertain input data encountered in realistic CAD situations. This means that the input data for a "point" is actually a rational rectangle, for example a point in the plane whose coordinates are two floating point numbers actually specifies a rational rectangle.

In order to be able to compute the continuous membership predicate, we extend it to the interval domain $\mathbf{I}\mathbb{R}^n$ and define $- \in - : \mathbf{I}\mathbb{R}^n \times \mathbf{S}\mathbb{R}^n \to \{\mathsf{tt}, \mathsf{ff}\}_\perp$ with:

$$C \in (A, B) = \begin{cases} \mathsf{tt} & \text{if } C \subseteq A \\ \mathsf{ff} & \text{if } C \subseteq B \\ \perp & \text{otherwise} \end{cases}$$

(see the figure below). Note that we use the infix notation for predicates and Boolean operations.

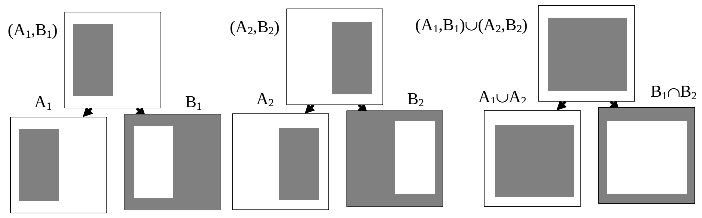


Starting with the continuous membership predicate, the natural definition for the complement would be to swap the values tt and ff. This means that the complement of (A, B) is (B, A).

Bearing in mind that for a partial solid object (A, B), the open sets A and B respectively capture the interior and the exterior of the solid, we can deduce the definition of Boolean operators on partial solids:

$$\begin{aligned} (A_1, B_1) \cup (A_2, B_2) &= (A_1 \cup A_2, B_1 \cap B_2) \\ (A_1, B_1) \cap (A_2, B_2) &= (A_1 \cap A_2, B_1 \cup B_2). \end{aligned}$$

One can likewise define the m-ary union and the m-ary intersection of partial solids. Note that, given two partial solids representing adjacent boxes, their union would not represent the set-theoretic union of the boxes, as illustrated in the figure.



In order to define the Boolean operations as maps on the domain of geometric objects, we need to define the product of two dcpo's. Given dcpo's $(A, \sqsubseteq)$ and $(B, \sqsubseteq)$, we define their product as $(A \times B, \sqsubseteq)$ where the Cartesian product $A \times B$ is given by

$$A \times B = \{(a, b) \mid a \in A, b \in B\}$$

and the partial order $\sqsubseteq$ on $A \times B$, is defined componentwise i.e. $(a, b) \sqsubseteq (a', b')$ iff $a \sqsubseteq a'$ and $b \sqsubseteq b'$.

Exercise 6.15 (i) Check that for dcpo's A and B the partial order $(A \times B, \sqsubseteq)$ is a dcpo.

(ii) Show that if A, B and C are dcpo's, then a map $f : A \times B \to C$ is continuous iff it is continuous in each component separately, i.e. iff for each $a \in A$ the map $b \mapsto f(a, b) : B \to C$ and for each $b \in B$ the map $a \mapsto f(a, b) : A \to C$ are continuous.

(iii) Define the product of n dcpo's A_i $(i = 1, \ldots, n)$, and show that it is a dcpo.

(v) Show that if A_i $(i = 1, \ldots, m)$ and B are dcpo's, then a map $f : A_1 \times A_2 \times \ldots \times A_m \to B$ is continuous iff it is continuous in each component separately, i.e. iff for each $i = 1, \ldots, m$, given $a_j \in A_j$ for $j \neq i$, the map $a_i \mapsto f(a_1, a_2, \ldots, a_m) : A_i \to B$ is continuous.

We can now express the binary and m-ary Boolean operations as functions on the solid domain.

Theorem 6.16 *The following are continuous:*

(i) Complementation map $\neg : \mathbf{S}\mathbb{R} \to \mathbf{S}\mathbb{R}$ *with* $\neg(A, B) = (B, A)$.

(ii) Binary union $- \cup - : \mathbf{S}\mathbb{R}^n \times \mathbf{S}\mathbb{R}^n \to \mathbf{S}\mathbb{R}^n$.

(iii) Binary intersection $- \cap - : \mathbf{S}\mathbb{R}^n \times \mathbf{S}\mathbb{R}^n \to \mathbf{S}\mathbb{R}^n$.

Proof (i) This is easy.

(ii) It is easy to check that $- \cup -$ is monotone in each component separately. Assume then that $(A, B) \in \mathbf{S}\mathbb{R}^n$ and let $((A_i, B_i))_{i \in I}$ be a directed set of geometric objects. We have:

$$(A, B) \cup \bigsqcup_{i \in I} (A_i, B_i) = (A, B) \cup (\bigcup_{i \in I} A_i, \bigcup_{i \in I} B_i)$$

$$= (A \cup \bigcup_{i \in I} A_i, B \cap \bigcup_{i \in I} B_i) = (\bigcup_{i \in I} A \cup A_i, \bigcup_{i \in I} B \cap B_i)$$

$$= \bigsqcup_{i \in I} (A \cup A_i, B \cap B_i).$$

(iii) This is dual to (ii).□

6.6 The convex hull revisited

We will now describe an algorithm in our domain-theoretic framework to compute the convex hull of a finite number of points in the plane in the context of the solid domain. It will be able to approximate the convex hull of m points in the plane by providing approximations from inside and outside to to the convex hull by rational polygons. The algorithm easily extends to higher dimensions.

Assume we have m points in the plane. Each of these points is approximated by a shrinking nested sequence of rational rectangles; at each finite stage of computation we have approximations to the m points by m rational rectangles. For these m rational rectangles we obtain a partial geometric object with an interior open rational polygon, which is contained in the interior of the convex hull of the m points, and an exterior open rational polygon, which is contained in the exterior of the convex hull of the m points.

The union of the interior (respectively, the exterior) open rational polygons obtained for all finite stages of computation gives the interior (respectively, the exterior) of the convex hull of the m points.

Formally, we define a map

$$C_m : (\mathbf{I}\mathbb{R}^2)^m \to \mathbf{S}\mathbb{R}^2,$$

where $\mathbf{I}\mathbb{R}^2$ is the domain of the planar rectangles, the collection of all rectangles of the plane partially ordered by reverse inclusion. Let $\mathcal{C}(\mathbb{R}^2)$ be the collection of non-empty bounded and closed subsets of $\mathbb{R}^2$ with the Hausdorff metric and let

$$H_m : (\mathbb{R}^2)^m \to \mathcal{C}(\mathbb{R}^2)$$

be the classical function which sends any m-tuple of planar points to its convex hull regarded as a closed bounded subset of the plane.

Let $(R_1, R_2, \cdots, R_m) \in (\mathbf{I}\mathbb{R}^2)^m$ be an m-tuple of rectangles. Each rectangle R_i has four vertices denoted, anti-clockwise starting with the bottom left corner, by R_i^1, R_i^2, R_i^3 and R_i^4.

We define

$$C_m((R_1, \cdots, R_m)) = (I_m((R_1, \cdots, R_m)), E_m((R_1, \cdots, R_m)))$$

where the *interior convex hull* is defined by

$$I_m((R_1, \cdots, R_m)) = (\bigcap_{y_i \in R_i \quad i=1,\cdots,m} H(y_1, \cdots, y_m))^\circ$$

and the *exterior convex hull* is defined by

$$E_m((R_1, \cdots, R_m)) = (\bigcup_{y_i \in R_i \; i=1,\cdots,m} H(y_1, \cdots, y_m))^c.$$

We will now show that $I_m((R_1, \cdots, R_m))$ and $E_m((R_1, \cdots, R_m))$ can be obtained by two simple algorithms.

In the following proof, we use the simple property that the convex hull of a finite set of points is precisely the intersection of all half-planes containing these points. Let $x = (R_1, \cdots, R_m)$.

Proposition 6.17 *We have the following finite algorithms to compute the interior and exterior convex hulls:*

$$I_m(x) = (\bigcap_{1\le j\le 4} H_m(R_i^j)_{i=1}^m)^\circ.$$

$$E_m(x) = (H_{4m}((R_i^1, R_i^2, R_i^3, R_i^4))_{i=1}^m)^c$$

Proof That $\bigcap_{y_i\in R_i} H(y_1, \cdots, y_m) \subseteq \bigcap_{1\le j\le 4} H_m(R_i^j)_{i=1}^m$ is clear. To show the converse relation, we note that any half plane containing a point from each of the rectangles will contain $\bigcap_{1\le j\le 4} H_m(R_i^j)_{i=1}^m$ and thus $\bigcap_{y_i\in R_i} H(y_1, \cdots, y_m) \supseteq \bigcap_{1\le j\le 4} H_m(R_i^j)_{i=1}^m$.

To show the second identity, we first note that $\bigcup_{y_i\in R_i} H(y_1, \cdots, y_m) \supseteq H_{4m}((R_i^1, R_i^2, R_i^3, R_i^4))_{i=1}^m$ holds as each point of the boundary of the latter is a point of one of the convex hulls in the union. Furthermore, $\bigcup_{y_i\in R_i} H(y_1, \cdots, y_m) \subseteq H_{4m}((R_i^1, R_i^2, R_i^3, R_i^4))_{i=1}^m$ since every convex hull in the union is contained in $H_{4m}((R_i^1, R_i^2, R_i^3, R_i^4))_{i=1}^m$.□

In words, $E_m(x)$ is the complement of the convex hull of the $4m$ vertices of all rectangles, whereas $I_m(x)$ is the interior of the intersection of the 4 convex hulls of the bottom left, bottom right, top right and top left vertices. Since the intersection of convex sets is convex, $I_m(x)$ as well as $E_m(x)$ are both convex open rational polygons.

Using the above proposition one can show the following theorem whose proof is non-trivial and is omitted.

Theorem 6.18 *The map $C_m : (\mathbf{I}\mathbb{R}^2)^m \to \mathbf{S}\mathbb{R}^2$ is Scott continuous.* □

Using the above theorem we can approximate the convex hull of m points $(a_1, a_2, \cdots, a_m)$ in the plane from inside and outside by rational convex hulls as follows. Suppose, for each $j = 1, \cdots, m$, $\{a_j\} = \bigcap_{i\ge 0} R_{ji}$ where $(R_{ji})_{i\ge 0}$, for each $j = 1, \cdots, m$, is a shrinking sequence of nested rectangles. Then, by the above theorem, the interior and the exterior of the convex hull is given by

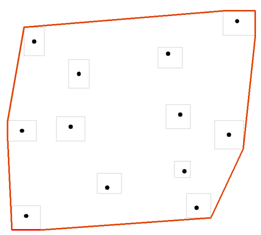
$$\begin{aligned} &C_m(\{a_1\}, \{a_2\}, \cdots, \{a_m\}) \\ &= \bigsqcup_{i\ge 0}(I_m(R_{1i}, \cdots, R_{mi}), E_m(R_{1i}, \cdots, R_{mi})) \\ &= (\bigcup_{i\ge 0} I_m(R_{1i}, \cdots, R_{mi}), \bigcup_{i\ge 0} E_m(R_{1i}, \cdots, R_{mi})). \end{aligned}$$

But by Proposition 6.17, for each $i \ge 0$, the interior and the exterior convex hulls $I_m(R_{1i}, \cdots, R_{mi})$ and $E_m(R_{1i}, \cdots, R_{mi})$ can be computed in finite time, in fact in $m \log m$ time.

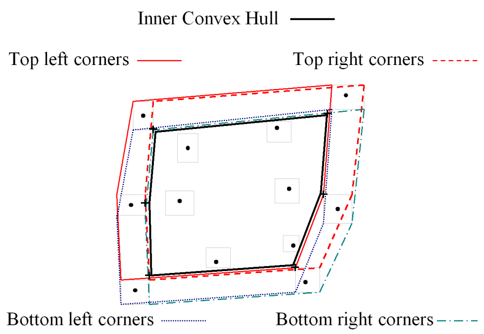
Example 6.19 Consider the points of the plane in the figure below, each approximated at every stage of computation by a rational rectangle.

At every stage of computation, we compute the outer (or exterior) and the inner (or interior) convex hulls. The outer convex hull will consist of points which are definitely outside the convex hull of the planer points and the inner convex hull will consist of points which are definitely in the interior of the convex hull of the points. The union of the inner convex hulls and that of the outer convex hulls give us respectively the interior and the exterior of the convex hull of the points.

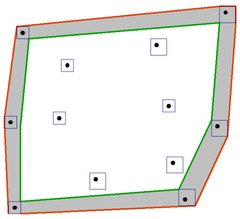
The outer convex hull of the rectangles is obtained by taking the convex hull of all the vertices of the rectangles as in the figure below. Since these vertices have rational coordinates, we can accurately compute their convex hull, a rational polygon.



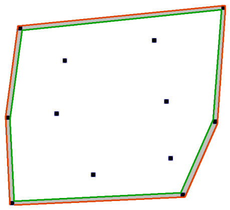
To compute the inner convex hull, we obtain the four convex hulls of the top left, top right, bottom right and bottom left corners and then find their intersection as in the figure below.



With more accurate input data about the planer points, the boundaries of the inner and outer convex hulls get closer to each other as in the next two figures.



In the limit, the inner and outer convex hulls will be simply the interior and the exterior of the convex hull of the planer points.



Since we work completely with rational arithmetic, we will not encounter any round-off errors and, since comparison of rational numbers is decidable, we will not get inconsistencies.

7 New cpo's from old

We have seen that if D and E are cpo's so is $[D \to E]$. We will now see more examples of constructing new cpo's.

7.1 Product

If D and E are cpo's, the *product* $D \times E$ consists of pairs (d, e) with $d \in D$ and $e \in E$ with the pointwise ordering, i.e. $(d, e) \sqsubseteq (d', e')$ iff $d \sqsubseteq d'$ and $e \sqsubseteq e'$. This is then a cpo with $\perp_{D\times E} = (\perp_D, \perp_E)$ and lubs of chains are obtained componentwise, i.e. the lub of

$$(d_0, e_0) \sqsubseteq (d_1, e_1) \sqsubseteq (d_2, e_2) \sqsubseteq \cdots$$

is simply $(\bigsqcup_i d_i, \bigsqcup_i e_i)$. There are two *projection* maps $\pi_1 : D \times E \to D$ and $\pi_2 : D \times E \to E$, defined by $\pi_1((d, e)) = d$ and $\pi_2((d, e)) = e$. If $f : D_1 \to D_2$ and $g : E_1 \to E_2$ are continuous, then we have a continuous map

$$f \times g : D_1 \times E_1 \to D_2 \times E_2$$

defined by $(f \times g)(d, e) = (f(d), g(e))$.

Exercise 7.1 Show that π_1, π_2 and $f \times g$ are indeed continuous.

7.2 Smash product

Given cpo's D and E, sometimes we want to identify $(d, \perp_E)$ $(d \in D)$ and also $(\perp_D, e)$ $(e \in E)$ with the new bottom. For cpo's D and E we define the *smash product*

$$D \otimes E =$$
$$\{(d, e) \in D \times E \mid d \neq \perp_D, e \neq \perp_E\} \cup \{\perp_{D\otimes E}\}$$

with the pointwise ordering and least element $\perp_{D\otimes E}$. Define the maps:

$$\begin{array}{llll} \mathsf{smash}: & D \times E & \to D \otimes E & \\ & (d, e) & \mapsto (d, e) & \text{if } d \neq \perp, e \neq \perp \\ & & \mapsto \perp_{D\otimes E} & \text{otherwise} \end{array}$$

and

$$\begin{array}{lll} \mathsf{unsmash}: & D \otimes E & \to D \times E \\ & (d, e) & \mapsto (d, e) \\ & \perp_{D\otimes E} & \mapsto (\perp, \perp). \end{array}$$

Now given $f : D_1 \to D_2$ and $g : E_1 \to E_2$ we have the map

$$f \otimes g : D_1 \otimes E_1 \to D_2 \otimes E_2$$

defined by $f \otimes g = \mathsf{smash} \circ (f \times g) \circ \mathsf{unsmash}$.

Exercise 7.2 Show that smash, $\mathsf{unsmash}$ and $f \otimes g$ are all continuous maps.

7.3 Coalesced sum

The *coalesced* sum $D \oplus E$ of cpo's D and E is given by the set

$$((D - \{\perp_D\}) \times \{l\} \cup ((E - \{\perp_E\}) \times \{r\}) \cup \{\perp_{D\oplus E}\}$$

where $\perp_{D\oplus E}$ is the new bottom element and $(D - \{\perp_D\}) \times \{l\}$ and $(E - \{\perp_E\}) \times \{r\}$ inherit the partial orders from D and E respectively.

We have the strict functions $\mathsf{inl} : D \to D \oplus E$ and $\mathsf{inr} : E \to D \oplus E$ defined by

$$\mathsf{inl}(x) = \begin{cases} (x, l) & \text{if } x \neq \perp, \\ \perp_{D\oplus E} & \text{if } x = \perp; \end{cases}$$

$$\mathsf{inr}(x) = \begin{cases} (x, r) & \text{if } x \neq \perp, \\ \perp_{D\oplus E} & \text{if } x = \perp. \end{cases}$$

Given continuous maps

$$f : D_1 \to D_2 \qquad g : E_1 \to E_2$$

we have a continuous map:

$$\begin{array}{llll} f \oplus g : & D_1 \oplus E_1 & \to D_2 \oplus E_2 & \\ & (d, l) & \mapsto (f(d), l) & \text{if } f(d) \neq \perp \\ & & \mapsto \perp & \text{if } f(d) = \perp \\ & (e, r) & \mapsto (g(e), r) & \text{if } g(e) \neq \perp \\ & & \mapsto \perp & \text{if } g(e) = \perp \\ & \perp & \mapsto \perp. & \end{array}$$

7.4 Lifting

We sometimes want to add a new bottom to a cpo. For a cpo D, define its *lift*

$$D_\perp = D \times \{0\} \cup \{\perp\}$$

where $\perp$ is a new bottom and we stipulate

$$\perp \sqsubseteq (d, 0) \text{ for all } d \in D$$

and

$$(d, 0) \sqsubseteq (d', 0) \text{ iff } d \sqsubseteq d'.$$

Given $f : D \to E$ we have a continuous map:

$$\begin{array}{rrcl} f_\perp : & D_\perp & \to & E_\perp \\ & (d,0) & \mapsto & (f(d),0) \\ & \perp & \mapsto & \perp. \end{array}$$

7.5 Disjoint sum

The *disjoint sum* $D + E$ of two cpo's D and E is given by $D + E = D_\perp \oplus E_\perp$. For two continuous functions

$$f : D_1 \to D_2 \qquad g : E_1 \to E_2$$

we have the continuous map

$$f + g : D_1 + E_1 \to D_2 + E_2$$

defined by $f + g = f_\perp \oplus g_\perp$.

7.6 Function space

We have already defined $[D \to E]$. Given continuous functions

$$f : D_2 \to D_1 \qquad g : E_1 \to E_2$$

we have a continuous function:

$$\begin{array}{rcl} [f \to g] : [D_1 \to E_1] & \to & [D_2 \to E_2] \\ h & \mapsto & g \circ h \circ f \end{array}$$

Note that f is in the direction opposite to g.

7.7 Categories

The proper framework to study these constructors is provided by the notion of *categories*. A category $\mathbf{C}$ consists of a collection, $Obj_{\mathbf{C}}$, of *objects* and a collection, $Mor_{\mathbf{C}}$, of morphisms (or arrows) $f : A \to B$ between the objects. Each object A has an *identity* morphism $1_A : A \to A$, and for each pair of morphisms

$$f : A \to B \qquad g : B \to C$$

there is a *composition* $g \circ f : A \to C$ satisfying:

(i) $f \circ 1_A = f$ for any $f : A \to B$ and $1_A \circ g = g$ for any $g : C \to A$.

(ii) $f \circ (g \circ h) = (f \circ g) \circ h$ whenever the compositions are defined, i.e. composition is associative.

We give a few examples of categories:

category	objects	morphisms
Set	sets	maps
Poset	partial orders	monotone maps
CPO	cpo's	continuous maps
$\mathbf{CPO_s}$	cpo's	strict continuous maps

Any partial order $(D, \sqsubseteq)$ can be considered as a category: The objects are the elements of D and for any $a, b \in D$, we assign a unique morphism from a to b iff $a \sqsubseteq b$.

The *product* $\mathbf{C} \times \mathbf{C}'$ of two categories $\mathbf{C}$ and $\mathbf{C}'$ has collection of objects $Obj_{\mathbf{C}} \times Obj_{\mathbf{C}'}$ and collection of morphisms $Mor_{\mathbf{C}} \times Mor_{\mathbf{C}'}$, with $(f, f') : (A, A') \to (B, B')$ if $f : A \to B$ and $g : A' \to B'$. The identity morphism on (A, A') is $(1_A, 1_{A'})$ and composition of morphisms is obtained componentwise.

The *opposite*, $\mathbf{C}^{op}$, of a category $\mathbf{C}$ has the same objects as $\mathbf{C}$ and also the same morphisms but considered in the opposite direction, i.e. any morphism $f : A \to B$ of $\mathbf{C}$ gives a morphism $f^{op} : B \to A$. If we denote composition in $\mathbf{C}$ as before by $- \circ -$ and in $\mathbf{C}^{op}$ by $- * -$, then we have

$$g^{op} * f^{op} = (f \circ g)^{op}.$$

A *functor* $F : \mathbf{C} \to \mathbf{C}'$ from the category $\mathbf{C}$ to the category $\mathbf{C}'$ is given by two mappings

$$F : Obj_{\mathbf{C}} \to Obj_{\mathbf{C}'} \qquad F : Mor_{\mathbf{C}} \to Mor_{\mathbf{C}'}$$

with any morphism $f : A \to B$ in $\mathbf{C}$ mapped to a morphism $F(f) : F(A) \to F(B)$ in $\mathbf{C}'$ such that the identity morphisms and the composition of morphisms are preserved, i.e.

- $F(1_A) = 1_{F(A)}$ for all $A \in Obj_{\mathbf{C}}$
- $F(g \circ f) = F(g) \circ F(f)$ for all composable $f, g \in Mor_{\mathbf{C}}$.

All constructors we have defined for cpo's are examples of functors:

$$- \times - : \mathbf{CPO} \times \mathbf{CPO} \to \mathbf{CPO}.$$

$$- \otimes - : \mathbf{CPO} \times \mathbf{CPO} \to \mathbf{CPO}.$$

$$- \oplus - : \mathbf{CPO} \times \mathbf{CPO} \to \mathbf{CPO}.$$

$$(-)_\perp : \mathbf{CPO} \to \mathbf{CPO}.$$

$$- + - : \mathbf{CPO} \times \mathbf{CPO} \to \mathbf{CPO}.$$

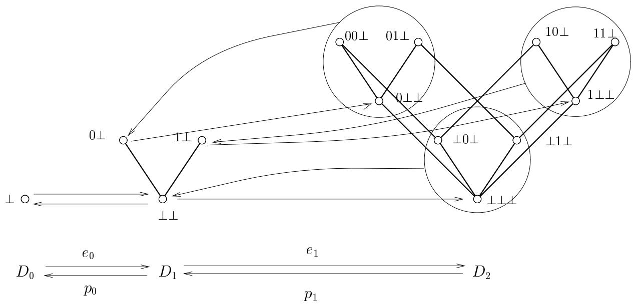
$$[- \to -] : \mathbf{CPO}^{op} \times \mathbf{CPO} \to \mathbf{CPO}.$$

8 Domain equations

Two objects A and B in a category are said to be *isomorphic*, denoted $A \cong B$, if there are morphisms $f : A \to B$ and $g : B \to A$ such that $g \circ f = 1_A$ and $f \circ g = 1_B$.

Let F : CPO $\to$ CPO be a functor. We want to solve the recursive equation $F(D) \cong D$ for D. We call this a domain equation.

Consider, for example, the domain equation $D \cong \Sigma_\perp \times D$, where $\Sigma = \{0, 1\}$. Let us try to solve this by mimicking the way we found the least fixed point of a continuous function $f : A \to A$, on a cpo A, by taking the lub, $\bigsqcup_i f^i(\perp)$, of the chain of iterates of f on the least element of A. The analogue of $\perp$ here is the cpo $\{\perp\}$ with only one element namely $\perp$.



Notice that D_i is *embedded* by e_i in D_{i+1} and each element d of D_{i+1} is *projected* to its best approximation $p_i(d)$ in D_i. Note also that the pairs (e_i, p_i) satisfy:

$$p_i \circ e_i = 1_{D_i} \qquad e_i \circ p_i \sqsubseteq 1_{D_{i+1}}.$$

Definition 8.1

Given cpo's D and E, we say that a continuous function $e : D \to E$ is an *embedding* of D into E if there exists a continuous function, called a *projection*, $p : E \to D$ such that $p \circ e = 1_D$ and $e \circ p \sqsubseteq 1_E$. We write $(e, p) : D \triangleleft E$.

Proposition 8.2

If $(e, p) : D \triangleleft E$ and $(e', p') : D \triangleleft E$ are two embedding-projection pairs between D and E, then $e \sqsubseteq e'$ iff $p \sqsupseteq p'$.

Proof

$e \sqsubseteq e' \;\Rightarrow\; p' = p \circ e \circ p' \sqsubseteq p \circ e' \circ p' \sqsubseteq p$

$p \sqsupseteq p' \;\Rightarrow\; e = e \circ p' \circ e' \sqsubseteq e \circ p \circ e' \sqsubseteq e' \square$

It follows that any embedding corresponds to a unique projection. It is easy to check that e and p are strict, p is onto and e is one-to-one and preserves existing lubs of arbitrary subsets. Because of the strictness, there is a unique embedding of the one point cpo $\{\perp\}$ to any cpo E.

Proposition 8.3

Let $(e, p) : D \triangleleft E$. Then

$$b \in E \;\Rightarrow\; p(b) = \bigsqcup \{a \in D \mid e(a) \sqsubseteq b\}.$$

Proof If $e(a) \sqsubseteq b$, then $a = p \circ e(a) \sqsubseteq p(b)$. On the other hand $e \circ p(b) \sqsubseteq b$. Therefore, $p(b)$ is both an upper bound and an element of the set $\{a \in D \mid e(a) \sqsubseteq b\}$.$\square$

This means that $b \in E$ is mapped to its best approximation in D. Now assume we have an "approximation sequence":

$$T : \; D_0 \triangleleft D_1 \triangleleft D_2 \triangleleft D_3 \triangleleft \cdots$$

with $(e_n, p_n) : D_n \triangleleft D_{n+1}$. How do we obtain the limiting object? Let D_∞ be the set of infinite sequences $\langle x_n \rangle_n = \langle x_0, x_1, x_2, \cdots \rangle$ with $x_n \in D_n$ and $p_n(x_{n+1}) = x_n$ for all $n \geq 0$. D_∞ is partially ordered componentwise, i.e.

$$\langle x_0, x_1, x_2 \cdots \rangle \sqsubseteq \langle x'_0, x'_1, x'_2 \cdots \rangle$$

iff $x_n \sqsubseteq x'_n$ for all $n \geq 0$. $(D_\infty, \sqsubseteq)$ is a cpo, in which the least upper bound of a chain is obtained componentwise, i.e. if $\langle y^i \rangle_{i \geq 0}$ is a chain in D_∞, then $\bigsqcup_i y^i = \langle \bigsqcup_i y^i_0, \bigsqcup_i y^i_1, \bigsqcup_i y^i_2, \cdots \rangle$.

We will justify that D_∞ is the "limiting object" of the "approximation sequence" T. (For simplicity, we avoid categorical notions.)

For $n \geq 0$, define $(E_n, P_n) : D_n \triangleleft D_\infty$ by

$$P_n(\langle x_0, x_1, x_2, \cdots \rangle) = x_n$$

$$E_n(x) = \langle f_{0n}(x), f_{1n}(x), f_{2n}(x), \cdots \rangle$$

where $f_{ij} : D_j \to D_i$ is defined by

$$f_{ij} = \begin{cases} p_i \circ p_{i+1} \circ \cdots \circ p_{j-1} & \text{if } i < j \\ 1_{D_i} & \text{if } i = j \\ e_{i-1} \circ e_{i-2} \circ \cdots \circ e_j & \text{if } i > j. \end{cases}$$

Exercise 8.4 Check that for each $n \geq 0$, (E_n, P_n) is an embedding-projection pair and that we have

$$E_{n+1} \circ e_n = E_n \qquad p_n \circ P_{n+1} = P_n.$$

This means that we have, for each $n \geq 0$, the following pair of commutative diagrams:

$$\begin{tikzcd}
\cdots\cdots & D_n \arrow[r, shift right, "e_n"'] \arrow[d, shift left, "E_n"] & D_{n+1} \arrow[l, shift right, "p_n"'] \arrow[dl, shift left, "E_{n+1}"] & \cdots\cdots \\
& D_\infty \arrow[u, shift left, "P_n"] \arrow[ur, shift left, "P_{n+1}"] & &
\end{tikzcd}$$

Therefore, each D_n is an approximation to D_∞, in other words D_∞ is an "upper bound" for the approximating chain.

We now have to show that D_∞ is the "least upper bound". Suppose A is a cpo which is another "upper bound" for the "approximating chain", i.e. $(\widetilde{E_n}, \widetilde{P_n}) : D_n \lhd A$ such that the following two diagrams commute for each $n \geq 0$:

$$\begin{tikzcd}[column sep=large, row sep=huge]
\cdots\cdots & D_n \arrow[r, shift right, "e_n"'] \arrow[d, shift left, "\widetilde{E_n}"] & D_{n+1} \arrow[l, shift right, "p_n"'] \arrow[dl, shift left, "\widetilde{E}_{n+1}"] & \cdots\cdots \\
& A \arrow[u, shift left, "\widetilde{P}_n"] \arrow[ur, shift left, "\widetilde{P}_{n+1}"] & &
\end{tikzcd}$$

Define $E : D_\infty \to A$ by

$$E(\langle x_0, x_1, \cdots \rangle) = \bigsqcup_n \widetilde{E_n}(x_n).$$

Note that

$$\widetilde{E_n} = \widetilde{E}_{n+1} \circ e_n \Rightarrow \widetilde{E_n} \circ p_n = \widetilde{E}_{n+1} \circ e_n \circ p_n \sqsubseteq \widetilde{E}_{n+1} \Rightarrow$$
$$\widetilde{E_n} \circ p_n(x_{n+1}) \sqsubseteq \widetilde{E}_{n+1}(x_{n+1}) \Rightarrow \widetilde{E_n}(x_n) \sqsubseteq \widetilde{E}_{n+1}(x_{n+1})$$

and hence E is well-defined.

E is continuous: For a chain $\langle x^i \rangle_{i \geq 0}$ in D_∞:

$$E\left(\bigsqcup_i x^i\right) = \bigsqcup_n \widetilde{E}_n\left(\bigsqcup_i x^i_n\right) = \bigsqcup_n \bigsqcup_i \widetilde{E}_n(x^i_n) =$$
$$\bigsqcup_i \bigsqcup_n \widetilde{E}_n(x^i_n) = \bigsqcup_i E(x^i).$$

Also define $P : A \to D_\infty$ by $P(y) = \langle \widetilde{P_n}(y) \rangle_n$. P is continuous since each P_n is continuous.

Proposition 8.5 $(E, P) : D_\infty \lhd A$.

Proof We have

$$E \circ P(y) = E(\langle \widetilde{P}_n(y) \rangle_n) = \bigsqcup_n \widetilde{E}_n \circ \widetilde{P}_n(y) \sqsubseteq y.$$

Also for $n > m$, we have

$$\widetilde{P}_m \circ \widetilde{E}_n(x_n) = f_{mn}(x_n) = x_m.$$

Hence, for any element $\langle x_n \rangle_n \in D_\infty$ we get

$$\widetilde{P}_m\left(\bigsqcup_n \widetilde{E}_n(x_n)\right) = x_m.$$

So

$$P \circ E\ (\langle x_m \rangle_m) = P\left(\bigsqcup_n \widetilde{E}_n(x_n)\right) =$$
$$\left\langle \widetilde{P}_m\left(\bigsqcup_n \widetilde{E}_n(x_n)\right)\right\rangle_m = \langle x_m \rangle_m \qquad \square$$

Exercise 8.6 Show that $(E, P) : D_\infty \lhd A$ is the unique embedding-projection pair which makes the following diagrams commute for all $n \geq 0$:

$$\begin{tikzcd}[column sep=large, row sep=huge]
D_n \arrow[r, shift right, "E_n"'] \arrow[d, shift left, "\widetilde{E}_n"] & D_\infty \arrow[l, shift right, "P_n"'] \arrow[dl, shift left, "E"] \\
A \arrow[u, shift left, "\widetilde{P}_n"] \arrow[ur, shift left, "P"] &
\end{tikzcd}$$

Conclude that D_∞ is "the least upper bound".

In the terminology of category theory, D_∞ is a *direct limit* or *colimit* of the approximation chain in the category of cpo's and embedding-projection pairs, denoted by $\mathbf{CPO}^{ep}$.

Exercise 8.7 Show that any two direct limits of a chain of embedding-projection pairs are isomorphic.

Note the analogy with the lubs of chains in cpo's. We further use this analogy to solve domain equations. We say a functor

$$F : \mathbf{CPO} \to \mathbf{CPO}$$

is *ω-continuous* if it preserves the pointwise ordering of morphisms in each function space and the "lubs" of "approximation chains" in $\mathbf{CPO}^{ep}$. Then, similar to the least fixed point theorem for cpo's, we deduce that for any continuous functor the "lub", D_∞, of

$$1 \xrightarrow{\ !\ } F(1) \xrightarrow{F(!)} F^2(1) \xrightarrow{F^2(!)} F^3(1) \xrightarrow{F^3(!)} \cdots$$

where $1 = \{\bot\}$ is the one point cpo and $! : 1 \to F(1)$ is the unique embedding-projection pair from 1 to $F(1)$, satisfies $F(D_\infty) \cong D_\infty$.

Using the following result, it is easy to check the ω-continuity of functors:

Proposition 8.8
If F : $\mathbf{CPO} \to \mathbf{CPO}$ is continuous on function spaces, then F : $\mathbf{CPO}^{ep} \to \mathbf{CPO}^{ep}$ is ω-continuous.

Proof See Plotkin's notes, page 43.$\square$

We cannot use this result for the function space functor

$$[- \to -] : \mathbf{CPO}^{op} \times \mathbf{CPO} \to \mathbf{CPO}$$

as it stands. However, we can define a functor

$$[- \to^{ep} -] : \mathbf{CPO}^{ep} \times \mathbf{CPO}^{ep} \to \mathbf{CPO}^{ep}$$

which is defined on objects as $[- \to -]$ and on morphisms $(e, p) : D \to E$ and $(e', p') : D' \to E'$ by

$$[(e, p) \to^{ep} (e', p')] =$$
$$([p \to e'], [e \to p']) : [D \to D'] \to [E \to E']$$

We check that $([p \to e'], [e \to p'])$ is an embedding-projection pair. We have

$$[e \to p'] \circ [p \to e'] = [p \circ e \to p' \circ e'] = [1_D \to 1_{D'}] = 1_{[D \to D']}$$
$$[p \to e'] \circ [e \to p'] = [e \circ p \to e' \circ p'] \sqsubseteq [1_E \to 1_{E'}] = 1_{[E \to E']}$$

as required.

We can now define a class of functors for which we can solve the domain equation. Note that for any functors $F : \mathbf{C} \to \mathbf{C}'$ and $G : \mathbf{C}' \to \mathbf{C}''$, the composition $G \circ F : \mathbf{C} \to \mathbf{C}''$ is a functor. For any category $\mathbf{C}$ there is a *diagonal functor* $\Delta : \mathbf{C} \to \mathbf{C} \times \mathbf{C}$ given by $\Delta(A) = (A, A)$ for an object A and $\Delta(f) = (f, f) : (A, A) \to (B, B)$ for a morphism $f : A \to B$.

Consider the class of functors

$$F : \mathbf{CPO}^{ep} \to \mathbf{CPO}^{ep}$$

obtained from composing the following functors

$$F ::= F_D \mid ID \mid - \times - \mid - \otimes - \mid - \oplus - \mid (-)_\perp \\ \mid - + - \mid [- \to^{ep} -] \mid \Delta$$

where F_D is the constant functor with D a given cpo, $F_D(X) = D$ and $F_D(f) = 1_D$ for all objects X and morphisms f, and ID is the identity functor with $ID(X) = X$ and $ID(f) = f$.

It is easy to check that each of the functors in the above list is continuous on function spaces. By Proposition 8.8, we have a class of functors which are ω-continuous. We can therefore solve domain equations with these functors. For example, the domain equation $D \cong \Sigma_\perp \times D$ treated earlier in this section is given by the functor $F = (F_{\Sigma_\perp} \times ID) \circ \Delta$.

9 Algebraic cpo's

To construct a computable theory of domains, we need to approximate the elements of a cpo by *finite* information or *finite* elements. This is the basic intuition for an algebraic cpo. The slogan for algebraic domains is that it is always sufficient to work with the finite elements.

Definition 9.1 An element $a \in A$ of a cpo A is *finite* or *compact* if for all chains $\langle d_i \rangle_{i \geq 0}$, whenever $a \sqsubseteq \bigsqcup_i d_i$ then $a \sqsubseteq d_i$ for some $i \geq 0$. The set of finite elements of A is denoted by K_A.

Example 9.2

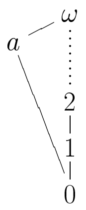
(i) In the cpo $(\mathbb{N} \cup \{\infty\}, \leq)$ the natural numbers are the finite elements.

(ii) In the cpo $\mathsf{Str}(\Sigma)$, the finite elements are precisely the finite strings.

Definition 9.3 A cpo A is *ω-algebraic* if the set, K_A, of finite elements of A, is countable (i.e. denumerable) and for any $a \in A$ there is a chain $\langle d_i \rangle_{i \geq 0}$, $d_i \in K_A$, of finite elements of A, with $a = \bigsqcup_i d_i$. The category of ω-algebraic cpos and continuous functions is denoted by ω-**ALG**.

The cpo $(\mathbb{N} \cup \{\infty\}, \leq)$ is ω-algebraic. If Σ is a countable set (e.g. $\Sigma = \{0, 1\}$), then $\mathsf{Str}(\Sigma)$ is ω-algebraic.

Example 9.4 The cpo $([0, 1], \leq)$, where $[0, 1]$ is the unit interval, is not ω-algebraic as its only finite element is 0. The following is another example of a cpo which is not ω-algebraic:



Exercise 9.5

(i) Show that a finite cpo is ω-algebraic and all its elements are finite.

(ii) Show more generally that if all chains in a countable cpo are eventually constant, then the cpo is ω-algebraic and all its elements are finite.

(iii) Show that the least upper bound $a \sqcup b$ of two finite elements a and b of a domain is finite (if it exists). Show that this is not always true of the greatest lower bound.

(iv) Show that embeddings preserve finiteness. How about projections?

Proposition 9.6
Let D and E be cpo's with $d \in K_D$ and $e \in K_E$. Then the step function $d \searrow e : D \to E$ defined by

$$(d \searrow e)(x) = \begin{cases} e & \text{if } d \sqsubseteq x \\ \perp & \text{otherwise} \end{cases}$$

is a finite element of the function space $[D \to E]$.

Proof It is easily seen that $d \searrow e$ is monotone. Let $\langle x_i \rangle_{i \geq 0}$ be a chain in D. If $d \sqsubseteq \bigsqcup_i x_i$, then by finiteness of d there is some $i \geq 0$ with $d \sqsubseteq x_i$ and hence

$$\textstyle\bigsqcup_i (d \searrow e)(x_i) = e = (d \searrow e)\left(\bigsqcup_i x_i\right).$$

Otherwise if $d \not\sqsubseteq \bigsqcup_i x_i$, then $d \not\sqsubseteq x_i$ for all $i \geq 0$ and we have

$$\textstyle\bigsqcup_i (d \searrow e)(x_i) = \perp = (d \searrow e)\left(\bigsqcup_i x_i\right).$$

Therefore $d \searrow e$ is continuous. Suppose now that $\langle f_i \rangle_{i \geq 0}$ is a chain in $[D \to E]$ with

$$(d \searrow e) \sqsubseteq \textstyle\bigsqcup_i f_i.$$

Then $\bigsqcup_i f_i(d)$ is a chain in E and

$$e = (d \searrow e)(d) \sqsubseteq \textstyle\bigsqcup_i f_i(d)$$

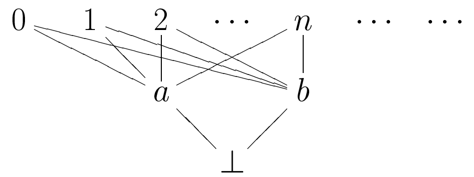
and hence, as e is finite, there is $i \geq 0$ with $e \sqsubseteq f_i(d)$. It follows that $(d \searrow e) \sqsubseteq f_i$, i.e. $d \searrow e$ is finite.$\Box$

9.1 Scott domains

It is easy to check that the constructors $(-)_\perp$, $- \times -$, $- \otimes -$ and $- + -$ all preserve ω-algebraicity; e.g. if A and A' are ω-algebraic, so is $A \times A'$ with

$$K_{A \times A'} = K_A \times K_{A'}.$$

However, the function space constructor does not preserve ω-algebraicity. For example, the function space $[A \to A]$ of the following ω-algebraic cpo A has an uncountable set of finite elements. (why?)



This is a serious shortcoming, as the function space plays a crucial role in semantics. E.g., models of the *untyped λ-calculus* in functional programming are solutions of the equation:

$$X \cong [X \to X] + A.$$

Fortunately, ω-**ALG** has a few subcategories with function space. We study the most important one.

Definition 9.7 A cpo is *bounded complete* if every bounded subset has a lub. A *Scott* domain is a bounded complete ω-algebraic cpo.

Proposition 9.8 *An ω-algebraic cpo is bounded complete iff every bounded pair of finite elements has a least upper bound.*

Proof The 'only if part' is trivial. For the 'if part', assume the cpo A has lubs of bounded pairs of finite elements and suppose $\langle c_i \rangle_{i \geq 0}$ is a bounded subset of finite elements with $c_i \sqsubseteq a$ for all $i \geq 0$. Define $\langle c'_i \rangle_{i \geq 0}$ inductively by $c'_0 = c_0$ and $c'_{i+1} = c'_i \sqcup c_{i+1}$ for all $i \geq 0$. We check by induction that c'_i is well-defined, finite and below a for all $i \geq 0$. This is clear for $c'_0 = c_0 \sqsubseteq a$. If c'_i is well-defined, finite and below a, then c'_i and c_{i+1} are both finite and bounded by a and therefore have by our assumption a least upper bound c'_{i+1}, which is below a and also finite by Exercise 9.5(iii). Clearly $\langle c'_i \rangle_{i \geq 0}$ is an increasing chain. Let $c = \bigsqcup_{i \geq 0} c'_i$. Then c is an upper bound for c_i (for all $i \geq 0$). On the other hand any upper bound of c_i's is also an upper bound for the chain $\langle c'_i \rangle_{i \geq 0}$ as well. Hence c is the lub of $\langle c_i \rangle_{i \geq 0}$. Suppose $B \subseteq A$ is bounded. Then

$$\{a \in A \mid a \sqsubseteq b \text{ for some } b \in B\} \cap K_A$$

is a bounded subset of finite elements and hence has lub b_0, say. But b_0 is an upper bound of each element of B and is clearly the lub of B.□

Our aim is to show that **S-DOM**, the category of Scott domains and continuous functions is closed under the function space construction.

Proposition 9.9
Suppose D and E are bounded complete cpo's. Then so is $[D \to E]$.

Proof Let $F \subseteq [D \to E]$ be bounded. Then for each $x \in D$, $\{f(x) \mid f \in F\}$ is bounded in E and so has lub $\bigsqcup_{f \in F} f(x)$. Define $(\bigsqcup F)(x) = \bigsqcup_{f \in F} f(x)$. Then $\bigsqcup F$ is continuous since for any chain $\langle x_i \rangle_{i \geq 0}$ in D

$$(\bigsqcup F)(\bigsqcup_i x_i) = \bigsqcup_{f \in F} \bigsqcup_i f(x_i) =$$
$$\bigsqcup_i \bigsqcup_{f \in F} f(x_i) = \bigsqcup_i (\bigsqcup F)(x_i).$$

Clearly $\bigsqcup F$ is the lub of F.□

We need one more result:

Proposition 9.10
Given a cpo A, if the countable set $B \subseteq K_A$ is such that every element of A is the lub of a chain in B, then $B = K_A$ and A is ω-algebraic.

Proof Let $c \in K_A$. Then by assumption there is a chain $\langle x_i \rangle_{i \geq 0}$ in B with $c = \bigsqcup_i x_i$. By finiteness of c there is $i \geq 0$ with $c \sqsubseteq x_i$, i.e. $c = x_i \in B$. Hence $B = K_A$ and A is ω-algebraic.□

Proposition 9.11 *Suppose D and E are Scott domains. Then so is $[D \to E]$.*

Proof We have already proved that $[D \to E]$ is bounded complete. Let $f \in [D \to E]$. We claim that $f = \bigsqcup F$ where

$$F = \{d \searrow e \mid d \in K_D, e \in K_E, e \sqsubseteq f(d)\}.$$

First note that $\bigsqcup F$ exists as $F \subseteq [D \to E]$ is bounded by f. Suppose that $x \in D$ and $e \in K_E$ with $e \sqsubseteq f(x)$. Since D is ω-algebraic there exists an increasing chain of finite elements $\langle d_i \rangle_{i \geq 0}$ with $x = \bigsqcup_i d_i$. By continuity of f we have

$$e \sqsubseteq f(x) = f(\bigsqcup d_i) = \bigsqcup f(d_i).$$

Hence, by finiteness of e, $e \sqsubseteq f(d)$, where $d = d_i$ for some $i \geq 0$. But then $(d \searrow e) \in F$ and $e \sqsubseteq (d \searrow e)(x)$. Since e is an arbitrary finite element below $f(x)$ and since $f(x)$ is the lub of the set of finite elements below it, we conclude that $f(x) \sqsubseteq (\bigsqcup F)(x)$ i.e. $f \sqsubseteq \bigsqcup F$. Hence, $f = \bigsqcup F$. Now consider the set $A \subseteq [D \to E]$ containing all step functions of the form $d \searrow e$ with $d \in K_D$ and $e \in K_E$ and the lubs of their bounded finite subsets. Then $A \subseteq K_{[D \to E]}$ will satisfy the conditions of Proposition 9.10. We conclude that $[D \to E]$ is ω-algebraic and hence a Scott domain.□

It is easy to check that Scott domains are closed under all other constructors we have studied so far, i.e. under $- \times -$, $- \otimes -$, $- \oplus -$, $- + -$, $(-)_\perp$ as well as $[- \to -]$.

In order to solve domain equations in **S-DOM** we need the following fact:

Exercise 9.12
Show that ω-**ALG** and **S-DOM** are closed under direct limits.

It follows that if

$$F : \mathbf{S\text{-}DOM} \to \mathbf{S\text{-}DOM}$$

is any functor which is continuous on function spaces, in particular if F is made up of the functors studied so far, the direct limit D_∞ of the chain

$$1 \xrightarrow{\;!\;} F(1) \xrightarrow{F(!)} F^2(1) \xrightarrow{F^2(!)} F^3(1) \xrightarrow{F^3(!)} \cdots$$

in $\mathbf{S\text{-}DOM}^{ep}$ is a Scott domain and and satisfies the domain equation $D_\infty \cong F(D_\infty)$.

Scott domains can be made *effective* by enumerating their finite elements and defining a *computable* element as the lub of an effective or computable chain of finite elements.