Effective and sequential definition by cases on the reals via infinite signed-digit numerals

http:/www.elsevier.nl/locate/entcs/volume13.html

Martín Escardó

Appsem Workshop, Pisa, 18th September 1998

In the Appsem specific objectives, one finds

"**I: Domain theory and real number computation**

[...] Recently, there has been some definite progress on these problems. [...] This language, however, is at this stage of theoretical interest only, as even basic arithmetic operations make use of parallel facilities. It is clearly desirable and necessary to find a sequential version of this language, [...]."

This talk deals with such sequentiality issues, particularly regarding the problem that the (in)equality relations are undecidable.

What I'll tell you

1. Computing on the reals.
2. Undecidability of the (in)equality relations.
3. A parallel solution to this problem.
4. Statement of a sequential solution.
5. Lexicographical normalization.
6. Construction of the sequential solution.

Computing on the reals via signed-digit numerals

The introduction of decimal notation, centuries ago, was a breakthrough regarding efficient approximate computation by hand.

Perhaps surprising fact (Brouwer 1920):

> Potentially infinite decimal numerals are not suitable for effective (exact) real number computation.

For example, addition is not computable in decimal notation.

One uses the fact that computable functions have to be of *finite character*:

> Finitely many *output* digits depend only on finitely many *input* digits.

Addition is not computable in decimal notation

It suffices to show that multiplication by 3 is not computable.

Suppose that, at some stage, the scanned part of the input is "$0 \cdot 3$".

The first output digit can be either 0 or 1.

If we eventually see an input digit

< 3, the first output digit has to be 0,

> 3, the first output digit has to be 1.

While the next scanned digit is 3, we cannot decide the first output digit.

If the input happens to be $1/3$, we can never decide the first output digit.

Q.E.D.

Brouwer's solution

Use base $2/3$ with digits 0 and 1, e.g.

$$[\![1001]\!] = 1 \times \left(\frac{2}{3}\right)^3 + 0 \times \left(\frac{2}{3}\right)^2 + 0 \times \left(\frac{2}{3}\right)^1 + 1 \times \left(\frac{2}{3}\right)^0$$

(Also Turing 1937b)

In what sense is this a solution?

Other solutions

Nested sequences of closed intervals with rational end-points.

Cauchy sequences of rationals with fixed rate of convergence.

Continued fractions.

...

All equivalent

In the sense that we can effectively translate between them.

The solution considered in this talk

Similar to Brouwer's.

Wiedmer 1980

Use base 2 with digits $\bar{1}$, 0 and 1.

$$[\![1\bar{1}01]\!] = 1 \times 2^3 - 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

(Leslie 1817, Cauchy 1840, Avizienis 1964, Di Gianantonio 1993 ...)

For simplicity and w.l.o.g., we'll only consider *fractional* infinite signed-digit binary expansions:

$$[\![\alpha]\!] = \sum_{i \geq 0} \alpha_i \times 2^{-(i+1)} \in [-1, 1],$$

for every $\alpha = \alpha_0 \alpha_1 \alpha_2 \cdots \in \{\bar{1}, 0, 1\}^\omega$,
where we omit redundant prefix "$0.$"

Definition ϕ *realizes* f iff

$$\begin{array}{ccc}
\{\bar{1},0,1\}^\omega & \xrightarrow{\phi} & \{\bar{1},0,1\}^\omega \\
{\scriptstyle [\![-]\!]}\Big\downarrow & & \Big\downarrow{\scriptstyle [\![-]\!]} \\
[-1,1] & \xrightarrow[f]{} & [-1,1].
\end{array}$$

Definition f is *computable* iff it has a computable realizer ϕ.

Recall

A necessary condition for ϕ being computable is that it is of finite character.

Topological point of view

$$
\begin{array}{ccc}
\{\bar{1},0,1\}^\omega & \xrightarrow{\phi} & \{\bar{1},0,1\}^\omega \\
{\scriptstyle [\![-]\!]}\downarrow & & \downarrow{\scriptstyle [\![-]\!]} \\
[-1,1] & \xrightarrow[f]{} & [-1,1].
\end{array}
$$

Endow the set $\{\bar{1},0,1\}$ with the discrete topology and $\{\bar{1},0,1\}^\omega$ with the product topology.

Facts

1. ϕ is of finite character iff it is continuous.

2. The denotation function $[\![-]\!]$ is a topological quotient map.

   $\Longrightarrow$ Every computable f is continuous.

3. Every continuous f has a continuous realizer ϕ (projectivity).

   $\Longrightarrow$ Every continuous f is "potentially" computable.

   In decimal notation, not even multiplication by 3 has a continuous realizer.

Undecidability of (in)equality

Non-constant predicates on real numbers are undecidable.

$$\begin{array}{ccc}
(\{\bar{1},0,1\}^\omega)^n & \xrightarrow{\phi} & \{\mathrm{true},\mathrm{false}\} \\
\big\downarrow & & \big\downarrow \\
[-1,1]^n & \xrightarrow[p]{} & \{\mathrm{true},\mathrm{false}\}.
\end{array}$$

Topologically, the set $p^{-1}\{\mathrm{true}\}$ would be clopen, but $[-1,1]^n$ is a connected space.

In particular, the (in)equality relations are undecidable.

(Alternatively, we can see that they are undecidable because otherwise there would be discontinuous computable functions, e.g.

$$\mathrm{sgn}(x) = \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{otherwise.)} \end{cases}$$

Nevertheless

Many definitions by cases consisting of inequalities, such as

$$\min(x, y) \quad = \quad \begin{cases} x & \text{if } x \leqslant y, \\ y & \text{otherwise,} \end{cases}$$

$$|x| \quad = \quad \begin{cases} -x & \text{if } x < 0, \\ x & \text{otherwise,} \end{cases}$$

do produce computable functions.

But

Such functions *cannot* be computed by

- first evaluating the condition,
- then computing the corresponding branch.

A parallel solution

Boehm and Cartwritght 1990

The partial function

$$x <_\perp y = \begin{cases} \text{true} & \text{if } x < y, \\ \perp & \text{if } x = y, \\ \text{false} & \text{if } x > y \end{cases}$$

is known to be computable.

Consider a computable "parallel" conditional such that

$$\mathrm{pif} \perp \mathrm{then}\, x \,\mathrm{else}\, x = x.$$

Then one can implement $|x|$ by

$$|x| = \mathrm{pif}\, x <_\perp 0 \,\mathrm{then} -x \,\mathrm{else}\, x,$$

because in the boundary case one has that

$$|0| = \mathrm{pif} \perp \mathrm{then} -0 \,\mathrm{else}\, 0 = 0.$$

More generally,

If f, g are computable and agree at a computable number x_0, then the function

$$h(x) = \begin{cases} f(x) & \text{if } x \leqslant x_0, \\ g(x) & \text{if } x \geqslant x_0 \end{cases}$$

is computable because

$$h(x) = \operatorname{pif} x <_\perp x_0 \operatorname{then} f(x) \operatorname{else} g(x).$$

The point is that, in the boundary cases,

the branches have to agree if a continuous function is to be defined.

So the parallel conditional always overcomes the fact that computable partial predicates are undefined at boundary cases.

But one would like to avoid parallel facilities.

This is the topic of this talk.

A sequential solution

We'll sketch a the proof of

Theorem

There is a sequentially computable partial function

$$\mathrm{cases} : [-1,1]^4 \rightharpoonup [-1,1]$$

such that

$$\mathrm{cases}(x,t,y,z) = \begin{cases} y & \text{\textit{if} } x < t \text{ \textit{or} } y = z, \\ z & \text{\textit{if} } x > t \text{ \textit{or} } y = z, \end{cases}$$

with domain of definition

$$\{(x,t,y,z) \mid x = t \text{ \textit{implies} } y = z\}.$$

Note

1. This partial function is realizable by a finite deterministic automaton (with some silent moves).
2. Topologically, the above is the largest domain of definition that it can have.

Lexicographical normalization

We order numerals lexicographically:

$\alpha < \beta$ iff

> α and β are of the form $\gamma d \alpha'$ and $\gamma e \beta'$ where γ is a finite sequence and d, e are digits with $d < e$.

Canonical forms for numerals

Every number is denoted by a smallest and by a largest numeral.

But

There are no *effectively determinable* canonical forms.

(These two facts are easily proved by topological methods.)

If $\alpha, \beta \in \{0,1\}^\omega$ are standard numerals then

$\alpha \leqslant \beta$ implies $[\![\alpha]\!] \leqslant [\![\beta]\!]$.

This property fails for signed-digit numerals.

For example, for $\alpha = \bar{1}1^\omega$ and $\beta = 0\bar{1}^\omega$ one has $\alpha < \beta$ but $[\![\alpha]\!] = 0 \nleqslant -1/2 = [\![\beta]\!]$.

Moreover, its converse fails for both standard and signed-digit numerals.

For example, for $\alpha = 10^\omega$ and $\beta = 01^\omega$ one has that $[\![\alpha]\!] = [\![\beta]\!] = 1/2$ but $\alpha \nleqslant \beta$.

However

Signed-digit numerals admit a very strong order-normalization property that standard numerals don't.

Definition A pair (α, β) of numerals is $\mathit{order\text{-}normal}$ iff its lexicographical order coincides with its numerical order:

1. $\alpha < \beta$ and $[\![\alpha]\!] < [\![\beta]\!]$, or
2. $\alpha = \beta$ (and $[\![\alpha]\!] = [\![\beta]\!]$), or
3. $\alpha > \beta$ and $[\![\alpha]\!] > [\![\beta]\!]$.

Thus, for order-normal pairs (α, β),

$[\![\alpha]\!] = [\![\beta]\!]$ implies $\alpha = \beta$.

Warning

1. Order-normality is a property of pairs of numerals and not of single numerals.

   In fact, a pair (α, α) is always order-normal.

2. There are $\mathit{distinct}$ order-normal pairs (α, β) and (α', β') denoting the same pair of numbers.

Let $\mathbf{3} = \{\bar{1}, 0, 1\}$.

Theorem

There is a sequentially computable, denotation-preserving, idempotent map

$$\mathrm{norm} : \mathbf{3}^\omega \times \mathbf{3}^\omega \to \mathbf{3}^\omega \times \mathbf{3}^\omega$$

whose fixed-points are order-normal pairs.

That is, if $\mathrm{norm}(\alpha, \beta) = (\alpha', \beta')$ then

1. $([\![\alpha]\!], [\![\beta]\!]) = ([\![\alpha']\!], [\![\beta']\!])$,

2. (α', β') is order-normal, and

3. $\mathrm{norm}(\alpha', \beta') = (\alpha', \beta')$.

Corollary

One can always assume w.l.o.g. that any two numerals that denote the same number are themselves the same.

Proof sketch

We write

$$\alpha = \gamma\alpha'\alpha'' \qquad \beta = \gamma\beta'\beta'',$$

with γ, α', β' initially empty.

Step 1

We transfer digits, one at a time, from α'', β'' to α', β' respectively.

(The size of α' and β' will be at most 3 at any stage.)

If the situation $\alpha' = \bar{1}\bar{1}0$ and $\beta' = 0$ is reached, we can stop because

$$[\![\gamma\bar{1}\bar{1}0\alpha'']\!] < [\![\gamma 0\beta'']\!].$$

Similarly, we can stop if $\alpha' = \bar{1}\bar{1}$ and $\beta' = 00$.

There are two additional symmetric cases, which we omit.

Step 2

If none of the above terminating cases holds, a case analysis shows that it is possible to apply the identities

$$[\![0\bar{1}]\!] = [\![\bar{1}1]\!] \qquad [\![01]\!] = [\![1\bar{1}]\!]$$

to prefixes of α', β' so that α', β' end up sharing the first digit.

This digit is removed from α', β' and transfered to γ.

Then go back to Step 1

The process doesn't terminate iff γ keeps growing for ever.

In this case α and β must denote the same number and $\mathrm{norm}(\alpha, \beta) = (\gamma, \gamma)$.

Q.E.D.

We can now prove the sequential case-analysis theorem.

Recall that

$$\mathrm{cases}(x,t,y,z) = \begin{cases} y & \text{if } x < t \text{ or } y = z, \\ z & \text{if } x > t \text{ or } y = z \end{cases}$$

and that we assume that $x = t$ implies $y = z$.

Algorithm for a sequential realizer:

1. Let $\alpha, \beta, \gamma, \delta$ be numerals denoting the numbers x, t, y, z.

2. We can assume that (α, β) and (γ, δ) are order-normal pairs.

3. We first output the greatest common prefix of γ, δ.

4. If it is infinite, we'll never reach the next stage.

5. Otherwise, we must have that either $\alpha < \beta$ or else $\alpha > \beta$.

6. We output the remainder of γ or δ accordingly, and the algorithm "terminates".

What I've told you

1. Computing on the reals.
2. Undecidability of the (in)equality relations.
3. A parallel solution to this problem.
4. Statement of a sequential solution.
5. Lexicographical normalization.
6. Construction of the sequential solution.

For more details see

`http:/www.elsevier.nl/locate/entcs/volume13.html`

THE END