

Fixed Points in Metric Semantics

J.W. de Bakker

Abstract

We present a brief tutorial on the use of metric spaces in semantics, with special attention for the role of fixed points.

In traditional semantic modelling, one is primarily interested in the I/O behaviour of a program. In more recent developments, phenomena such as parallel processing and reactive systems have gained increased importance. For these kinds of notions, the I/O model is no longer suitable, and one is led to work with *full histories* of the computations. A further key observation is that these histories may be -at least in principle- *infinite*: there is no reason to assume *a priori* that all computations have to terminate. This situation induces a natural definition of a metric on the space of computations: two computations have distance 2^{-n} whenever the first difference in their behaviour appears after n steps. (This metric is in fact a slight variation on the classical Baire (ultra)metric.) The notion of behaviour referred to here is still generic: it may be instantiated as consisting of *sequences* of actions or states, or of *tree structures* over such entities (further possibilities exist as well but are not discussed here).

Once the Baire metric has been introduced, the scene is set for fruitful use of a range of classical metric (or topological) tools. The Baire distance can be extended to the Hausdorff distance on *sets* of entities, thus allowing the modelling of a spectrum of *nondeterministic* notions. In general, we want to work with sets of meanings where all relevant limits are present, leading to the use of *closed* subsets and of *complete* spaces. A large proportion of the programming notions considered induces in a natural way the notion of a (first) *computational step* - to be followed by the remainder of the computation. In many cases, this will have as counterpart that the semantic function which is to be associated with the notion concerned enjoys the property of being *contractive*. Here we recall the classical Banach fixed point theorem: each contractive endofunction on a complete metric space has a unique fixed point. This result is pervasive in all of metric semantics, both as definitional principle and as a proof principle. As to the latter: if one wants to show

$x = y$, look for some contractive f which has both x and y as fixed point. One further topological notion is to be added to the above: often, in some computational setting a suitable *finiteness* condition is imposed. Topologically, this may take the form of a *compactness* condition. In our metric framework, we shall rely on the fact that a set is compact whenever it is the limit of a sequence of finite sets.

Equipped with the tool set of elementary metric topology, we may start our semantic design: Classically, the first instance where fixed points are applied is in the modelling of recursive procedures, and in metric semantics this holds as well. However, there are many more examples. Firstly, the various semantic operators corresponding to the respective language constructs - such as sequential composition, choice, parallel composition, etc. - have to be defined for both *finite and infinite* operands. A simple definition by induction on their length is not feasible. Rather, we now define these operators as fixed points of higher order mappings. Furthermore, the denotational meaning function itself - say D , from language L to domain of meanings P - may be defined as (unique) fixed point of a higher-order mapping $\Psi : (L \text{ to } P) \text{ to } (L \text{ to } P)$. Again, contractivity of Ψ has to be established; an essential condition for this is that all recursion is *guarded*.

Maybe less standard is the role of fixed points in operational semantics. As usual, we start from a suitable transition system. Based on this, we define an associated operational semantics O

: L to P , essentially by collecting all successive labels from the induced transition sequences, and by catering for the infinite case by another application of the unique fixed point principle. More specifically, we associate a higher-order mapping $\mathbf{Phi} : (L \text{ to } P) \text{ to } (L \text{ to } P)$ with the transition system, and show that we may put

$O = \text{fix}(\mathbf{Phi})$. This provides us with a means to establish that

$O = D$, viz. by verifying that

$D = \mathbf{Phi}(D)$.

We add a side remark on the role of topology here: Transition systems may satisfy two finiteness conditions (viz. of being *finitely branching* or *image finite*). An important result of metric semantics is the following: the operational semantics associated with a finitely branching (image finite) transition system is a compact (closed) set.

We conclude the list of basic building blocks for metric semantics with a few words on the metric approach to domain equations. For F a mapping from the class of complete metric spaces to itself, we may solve the isometry P congruent to $F(P)$ - for F satisfying rather mild conditions - either purely in a metric setting (see [ABBR95] for a recent treatment), or by categorical techniques (first described in [AR89]) which are a variation on the classical cpo-methods. Depending on the semantic setting, we shall encounter functors — built in terms of, e.g., product, disjoint union, power space (of closed or compact subsets), the function space of all nonexpansive functions, the 'halving' functor which leaves the space untouched and halves the metric, and a few more. (Recently, a new perspective on these constructions has been obtained by a systematic use of (final) coalgebras, described e.g. in [RT94].)

In [BV96a], we have gathered comprehensive evidence that metric semantics is both a powerful and a convenient tool: For 27 languages, including both traditional concepts (such as assignment, iteration and recursion, locality) and more modern ones (process creation, action refinement, parallel objects, the rendez-vous construct), we have designed operational and denotational models, and established precise results on their relationships. Throughout, uniqueness of fixed points turns out to be the primary proof principle.
