Toward an Implementation of Jackendoff's Paths

Daniel Barkalow

December 12, 2001

Abstract

Jackendoff [3] gives a compelling argument for a particular model of the physical world based on Paths, continuous functions giving positions. In this paper, I refine his idea of positions and describe a computational model for Paths and related concepts. The resulting model permits a straightforward implementation in data structures.

1 Place vs. Location

The locations of objects are not conceptualized as volumes (the space actually taken up by the object, that is), but are rather as points– the location of the center of mass or center of volume of the object, for instance, or some other key point. That is why a path or location used by one object is suitable for use by a differently-shaped object, so long as the objects are normal[1]. On the other hand, what prepositional phrases of location refer to seems to be volumes (or spaces of lower dimension, sometimes). The semantics of the connection seems to be inclusion: if a cup is "on the table", one presumes it to be at a specific spot on the table; only with certain special prepositions would an object be taken to fill the entire space.

Therefore, it is necessary to have two such concepts: one which is normally of no size, and is used for the exact locations of objects[2], the other of which refers to the set of possible locations which fit certain spatial criteria. I will use [LOCATION] to refer to the former, and [PLACE] (following Jackendoff) for the latter, and I consider Paths to be parameterized Locations.

A Place, first of all, refers to a volume, surface, line, or point. Its semantics is to restrict Locations which must fit it to those within the specified region. In addition, since support, contact, and attachment seem to be conceptualized as being part of a Location[3], these are also possible attributes of a Place.

[1] That is, rivers and highways take different sorts of locations from pens and cups. But an ant can sit where an elephant was without causing semantic problems.

[2] Not that it **specifies** the exact location; however, it symbolized whatever that location turns out to be. While we might not know where an object is, it is perfectly natural to build other concepts off of the location as a symbolic entity.

[3] "on", for example, seems to specify nothing more or less than support: it doesn't imply "above" ("The smoke detector is on the ceiling"), attachment ("The book is on the table"),

While most Locations are semantically single points, there are some which are for objects such as a quantity of water, or a swarm of bees, or a set of telephone poles. These objects can be thought of as being located in a region rather than at a point; the distinctive feature is that the extent is not a property of the object, since it is formless and of variable density, and so it may be considered a property of the location of the object.

Note that these Locations are still different from Places, though, because they are more specific. The place "all along the road" does not specify exactly where each telephone pole is, but only a fact about all of these points, where as the Location represents the positions. Thus, "distributivity" is, like support, an attribute of Places which specifies a corresponding attribute of the Locations.

2 Parameterized vs. Extruded Concepts

Jackendoff refers to Marr's generalized cones in support of his directional or parameterized concepts. I feel that these ideas are somewhat farther apart than he suggests.

First of all, they are very different in implication. When dealing with a path, one does not have the feeling of a uniform solid; each parameter value refers to a single resulting Location. For an object, its primary aspect (on the level of 3D shape) is the total space it takes up; the cross-section at a particular spot does not stand out as a Concept of independent interest. An object may therefore be represented in multiple ways by choosing different axes, while a Path cannot.

Secondly, to the extent that "slices" of each of these are allowed to overlap with each other, the effect is very different. If different slices of an generalized cone would overlap, the only sensible interpretation is either that one prevails or that they must be of the same material and be simply redundant. If, however, an object has different parts in a given location at different values along a path, the repetition is perfectly natural, and is not at all to say that those parts should collide. For example, in "The water spilled all over the floor", the water sensibly has a three-dimensional location at each point in time, and then has a fourth dimension corresponding to the time. The motion, which is the reason for a Path in the first place, would not be conveyed by the three-dimensional space that the water takes up over the course of the event.

Therefore, I will refer only to Paths and concepts like them as "parameterized", and call the other sort "extruded". Time spans and Paths are parameterized, and objects and Events are normally extruded, although objects such as highways which are built on parameterized concepts retain access to the parameterized underlying concept.

I consider a parameterized concept to be a mapping from values in a parameter space to concepts of the underlying category. A parameterized concept must specify the space its values will come from, since, as we will see, parameter spaces are not, in general, the same.

---

or even contact ("The stack of blocks is on the table", where only the bottom block is in contact, buy all are supported by the table through only insignificant intermediaries).

3 The Parameter Space

For extruded concepts, the parameter space is not all that interesting: it is just a continuous range between the two ends. For parameterized concepts, however, the parameter space is much more complicated, with a number of important properties.

3.1 Requirements

It is here that we get sp-binding. It must be possible to have two parameterized concepts such that there are pairs of underlying concepts, one from each, for each parameter value. For example, the construction of motion Events requires a Path and a Time-Span such that the two form a parameterized relation between Locations and Times.

People can forget things and fail to specify information in an interesting way. It is relatively easy, using multiple sentences, to specify a path with two regions, each of which has ordered Locations, but where the ordering of the two sections is unspecified. For example, if you say that you passed a bank on your way home, and then discuss the bank, and then mention a store that you also passed, the listener gets no information about which you passed first. Furthermore, it is easy to remember all of the significant spots along a route that you follow frequently without being about to remember in what order the landmarks appear. Thus, the path and the parameter space have to allow for a partial order of information.

In addition to the information about individual Locations of a Path, we must have information about whole regions of a Path. For example, in "The ball rolled around under the table", the entire Path (that is, every Location on it) is under the table. It is similarly important to be able to specify such information about parts of Paths. This is different from simply using only Locations which fit the requirement, because, since Paths are continuous, it is impossible to determine with a finite number of checks any information about a segment. Furthermore, there are properties, such as "straight", which do not apply to the individual Locations, but only to segments.

The representation of Paths generated with functions like "TOWARD" given in Jackendoff [2] seems incorrect to me. While the Paths are nonbounded, they are not like open intervals, with the semantics that the Path gets arbitrarily close to a particular Location, which it still never reaches. In fact, if someone walks "toward the store", they are presumed to stop at some Location, which they do reach. The nonboundedness, like that of a plural or a material, comes because the boundary point, which must sensibly exist, is unspecified and thus free to be specified by something else: a measure phrase on either the path or the time span, for instance. The implication seems to be that the Path goes to the intended goal, but only part of the Path is used.

People have generally considered the parameter space to be discrete; most of the literature that Jackendoff cites[4] (and disagrees with) assumes that Paths

[4] Tenny and Verkuyl, for instance. Borchardt[1], on the other hand, uses segments to good

consist of a finite ordered set of Locations: the "snapshots". While Jackendoff seems to be correct that Paths are conceptually continuous, the natural intuition that there cannot be an infinite amount of information in them, and that there are a set of special slices that are somehow more important must not be ignored. In the case of language, for example, we pick out just a few important examples to mention.

3.2 Representation

All of these properties can be handled by my representation. It is a finite set of parameter values, which are arranged in a partial order. Two values are distinguished as the beginning and the end. There may be values beyond these two, and there may be values incomparable to them. New values may be freely created at any time with any relationships to existing values.

A Path maps each value to a Location, and also contains Places (and other information) attached to ranges of values (that is, comparable pairs a and b), which specify that all of the Locations which values in the range map to fit the Place. This allows the Locations of values in the range to be restricted even if the values have not yet been created.

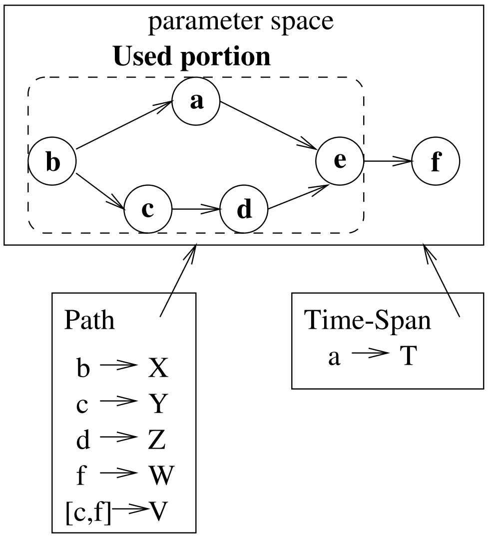

Figure 1: A Path and Time-Span sp-bound for the representation of a GO event. X is the source, W is the intended goal (i.e., TOWARD W), Y and Z are Locations along the Path. T is a time that occurs during the event. The Path from Y on (all the way to W) is in V.

Sp-binding works by simply reusing the same parameter space in multiple

effect, although he is considering only time in this way.

concepts; the "structure" is in the parameter space, not the concepts, and thus is automatically preserved through the binding.

Underspecification of the order of parts of paths follows from the partial order; while the idea is that the order is complete, partial information can be represented and used; furthermore, unimportant ordering information may be simply forgotten out of disuse.

Path segments have as straightforward an attachment to Places as Locations do, and the semantics of this connection is straightforward and easily applied in the representation. Since a Location may be restricted by a Place, the meaning of a segment being restricted by a Place can be easily applied to newly created Locations for new values, simply by restricting the Location for a new value v with all of the Places attached to segments from a to b where $a < v < b$.

TOWARD and AWAY-FROM are easily handled by specifying them as beyond the endpoints. The Path is thus over a longer virtual range, with only part marked as actually used.

While the parameter space is theoretically continuous, it is also "lazy", in that only those values where something important happens actually get represented. The rest of the values are represented only implicitly, by the information stored about the segments that new values may be within.

4 Result

In addition to the category [PLACE], I have a category [LOCATION], which better fits the task of specifying where something is; Places are now taken to be discriminating functions for Locations.

I have a support structure, the parameter space, which is an essential part of the representation of Paths and Time-Spans, and may be shared between multiple concepts to implement sp-binding.

I recast Paths as mappings from values of this parameter space to Locations, thus making Paths essentially different from Places. This representation captures the continuous properties of Paths by applying Places to segments of the Path as well as to Locations on it; this allows a finite representation to contain all of the important information about a continuous concept.

The one thing this Path representation does not do, however, is submit easily to typography. The notation that Jackendoff uses for his concepts is entirely unsuitable for these concepts at this level of detail. Of course, the earlier Path notation is a suitable shorthand for all of the information accessible to language. The work here, however, concerns what is going on beyond what is accessible in a single linguistic expression, and it is possible to use Jackendoff's notation (with a few reasonable extensions) multiple times to capture all of the information, as with multiple sentences.

References

[1] Gary C. Borchardt. Causal reconstruction. A.I. Memo 1403, MIT, February 1993.

[2] Ray Jackendoff. Parts and boundaries. *Cognition*, 41:9–45, 1991.

[3] Ray Jackendoff. The proper treatment of measuring out, telicity, and perhaps even quantification in english. *Natural Language and Linguistic Theory*, 14:305–354, 1996.