A Hierarchy of Boundary-based Shape Descriptors

Richard Meathrel and **Antony Galton**
School of Engineering and Computer Science
University of Exeter
Exeter EX4 4PT, UK
E-mail: {R.C.Meathrel, A.P.Galton}@ex.ac.uk

Abstract

In this paper we extend previous work on the boundary-based approach to describing shape, by deriving an unbounded hierarchy of "atomic" shape descriptors (called tokens) based on tangent bearing and its successive derivatives, and incorporating angle and cusp curve features. Both open and closed curves have token-string descriptions at all levels in the hierarchy. We provide a pair of compatibility matrices for generating transition tables for any level, from which level-specific token ordering graphs that encode basic string syntax can be systematically constructed.

1 Introduction

In the development of conceptual tools for spatial representation and reasoning, the category of shape has proved to be one of the most problematic areas. Some important earlier work, representative of the boundary-based approach to shape, is exemplified by the contour codons of Hoffman and Richards [1982] and the extremum primitives of Leyton [1988]. These two approaches, although differing in motivation and detail, both used the idea of characterising the shape of an outline by means of a string of tokens, recording salient curvature-based features encountered during a traversal of the outline. A more recent approach, using yet another set of primitives, is that of [Galton and Meathrel, 1999]. In [Meathrel and Galton, 2000], we attempted to define the boundary-based approach to shape in a more general and systematic way. By considering variations in curvature as a starting point, we derived two sets of *atomic tokens* for describing curves, and presented *token ordering graphs* for verifying the syntax of atomic curve descriptions.

The present paper extends the work described in [Meathrel and Galton, 2000] in three significant respects: (i) by systematically investigating the structure of sets of atomic curvature-based tokens, resulting in an unbounded hierarchy of atomic tokens, (ii) by incorporating kink points (angles and cusps) into the framework of tokens, and (iii) by providing a pair of *compatibility matrices*, and an algorithm, for generating transition tables for any level in the hierarchy, from which level-specific token ordering graphs can be constructed.

2 Deriving the hierarchy of atomic tokens

At each point on a curve, the tangent to the curve at the point is either defined or undefined. The rate of change of the tangent bearing b with distance along a curve gives us curvature (the first derivative of tangent bearing with respect to arc-length). We can also consider the rate of change of curvature (c'), which is the second derivative of tangent bearing. For each curve we can think of there being an infinite number of associated plots. Our hierarchy of descriptors is based on a discretisation of b and its derivatives. For tangent bearing we are interested in whether b is defined (D) or undefined (U), so we use the quantity space $\{D, U\}$. Points where the tangent bearing is undefined correspond to angles and cusps, and may be referred to as *kink points*. For all of the derivatives of tangent bearing (i.e., $c, c', c'', \ldots$), we use the quantity space $\{+, 0, -, U\}$, since a derivative may be positive, zero, negative, or undefined.

2.1 Curve states

Associated with each point P on a curve is a sequence of qualitative values, representing b, c, c', etc. The *complete curve state* at P corresponds to an infinite sequence of values. We write $\partial^k x$ to denote the kth component of the curve state x. Not all component sequences give rise to valid curve states. If a component has a value other than U, then the value of the next component is unconstrained. If, however, a component *does* have the value U, then all subsequent components must also be U. More formally, a curve state x must satisfy the following constraint:

$$(\forall k)(\, \partial^k x = U \rightarrow \partial^{k+1} x = U\,)$$

A *partial curve state* at P is any initial n-tuple of the complete curve state. So if $x = \langle D, + \rangle$, for example, then $\partial^1 x = D$, $\partial^2 x = +$, and x is a partial curve state that is assigned to points on a curve where the tangent bearing is defined and the curvature is positive.

We refer to a partial curve state with n components as a level-n state. Given the complete set of states for some level k, it is straightforward to generate the set of states for level $k{+}1$. Each state x, at level k, generates a set of level-$k{+}1$ states, $\mathcal{S}$, as follows (where $y = \partial^k x$):

$$\mathcal{S} = \begin{cases} \{\, \langle \ldots, y, U \rangle \,\} & \text{if } y = U \\ \{\, \langle \ldots, y, + \rangle, \langle \ldots, y, 0 \rangle, & \text{otherwise} \\ \qquad \langle \ldots, y, - \rangle, \langle \ldots, y, U \rangle \,\} & \end{cases}$$

There are two partial curve states with one component: $\langle D\rangle$ and $\langle U\rangle$. From these two states we can generate the five states of level 2 and then, from those, the fourteen states of level 3:

$$
\begin{array}{lll}
\langle D\rangle & : & \langle D,+\rangle,\ \langle D,0\rangle,\ \langle D,-\rangle,\ \langle D,U\rangle \\
\langle U\rangle & : & \langle U,U\rangle
\end{array}
$$

$$
\begin{array}{lll}
\langle D,+\rangle & : & \langle D,+,+\rangle,\ \langle D,+,0\rangle,\ \langle D,+,-\rangle,\ \langle D,+,U\rangle \\
\langle D,0\rangle & : & \langle D,0,+\rangle,\ \langle D,0,0\rangle,\ \langle D,0,-\rangle,\ \langle D,0,U\rangle \\
\langle D,-\rangle & : & \langle D,-,+\rangle,\ \langle D,-,0\rangle,\ \langle D,-,-\rangle,\ \langle D,-,U\rangle \\
\langle D,U\rangle & : & \langle D,U,U\rangle \\
\langle U,U\rangle & : & \langle U,U,U\rangle
\end{array}
$$

For each point on a curve we can assign a partial curve state of k components, so we could theoretically represent a curve by providing a mapping between curve points and partial curve states. However, such a mapping would be infinite and therefore of no practical use. Because our state components take qualitative values, however, certain states may persist over intervals of curve and, therefore, support mappings that are finite. A curve state may have an interval *and/or* a point interpretation. A state that has an interval interpretation may persist over an interval of curve, and a state that has a point interpretation may hold at a single curve point, without holding on any interval adjoining that point. An *atomic token* identifies a particular interpretation of a partial curve state.

2.2 Interval and point interpretation

For each set of level-k curve states, we obtain a corresponding set of level-k atomic tokens by considering the allowable interpretations of each state. A state may support an interval interpretation, a point interpretation, or both. An atomic token (or "atom") is a particular interpretation of a particular state, and is identified by a *signature* consisting of a sequence of qualitative component values. A signature that is underlined indicates an interval interpretation; a non-underlined signature indicates a point interpretation. The atom $\underline{\texttt{D+}}$, for example, is identified with the interval interpretation of the state $\langle D,+\rangle$, and the atom $\texttt{D+-0}$ is identified with the point interpretation of the state $\langle D,+,-,0\rangle$.

A curve state may hold at a single point *iff* one of its components is either zero or undefined. This is because if all components are defined, then all of them are continuous, and hence the values '$+$' and '$-$' can only hold over intervals. A curve state may persist over an interval *iff* none of its components are undefined and, whenever a component has the value zero, the next component also has the value zero. The following predicates, therefore, can be used to determine the interpretations that are supported by a curve state:

$$
\begin{array}{l}
\textsf{point-interp}(x) \leftrightarrow (\exists k)(\partial^k x = U \lor \partial^k x = 0) \\
\textsf{interval-interp}(x) \leftrightarrow \\
\qquad (\neg\exists k)(\partial^k x = U) \land (\forall k)(\partial^k x = 0 \rightarrow \partial^{k+1} x = 0)
\end{array}
$$

2.3 Atomic hierarchy

For each set of partial curve states (each level) we can use the predicates point-interp and interval-interp to obtain the corresponding set of atoms. The partial curve states and atoms for level 3 are as follows:

$\langle D,+,+\rangle$:	$\underline{\texttt{D++}}$	$\langle D,0,U\rangle$:	$\texttt{D0U}$
$\langle D,+,0\rangle$:	$\underline{\texttt{D+0}}$, $\texttt{D+0}$	$\langle D,-,+\rangle$:	$\underline{\texttt{D-+}}$
$\langle D,+,-\rangle$:	$\underline{\texttt{D+-}}$	$\langle D,-,0\rangle$:	$\underline{\texttt{D-0}}$, $\texttt{D-0}$
$\langle D,+,U\rangle$:	$\texttt{D+U}$	$\langle D,-,-\rangle$:	$\underline{\texttt{D--}}$
$\langle D,0,+\rangle$:	$\texttt{D0+}$	$\langle D,-,U\rangle$:	$\texttt{D-U}$
$\langle D,0,0\rangle$:	$\underline{\texttt{D00}}$, $\texttt{D00}$	$\langle D,U,U\rangle$:	$\texttt{DUU}$
$\langle D,0,-\rangle$:	$\texttt{D0-}$	$\langle U,U,U\rangle$:	$\texttt{UUU}$

The first four levels of the hierarchy of atomic tokens are shown in Figure 1. Each atom in the hierarchy (except the atoms $\underline{\texttt{D}}$ and $\texttt{U}$ at level 1) is a child of a single parent atom at the previous level, i.e., each atom at level k is derivable from one, and only one, atom at level $k-1$. We call parent atoms that have more than one child *expansive*, and those with only one child *non-expansive*. An interval atom is expansive *iff* its last component is either '$+$' or '$-$', otherwise it is non-expansive. A point atom is expansive *iff* its last component is not 'U', otherwise it is non-expansive. At level 1, then, there is just one non-expansive atom, $\texttt{U}$, which propagates through the hierarchy as $\texttt{UU}$, $\texttt{UUU}$, and so on. The other level-1 atom, $\underline{\texttt{D}}$, expands into five atoms at level 2; two of which are non-expansive: $\underline{\texttt{D0}}$ and $\texttt{DU}$.

Any atom is either an interval (I) or a point (P), and either expansive (E) or non-expansive (N), giving us four kinds of atoms: IE, IN, PE, and PN. Each IE atom yields five children: two IE atoms, one IN atom, one PE atom, and one PN atom. Each PE atom yields four children: three PE atoms and one PN atom. The non-expansive atoms, IN and PN, yield a single IN atom and a single PN atom, respectively.

The recursive equations for calculating the numbers of each kind of atom at level k, and the actual figures for levels 1 to 6, are as follows:

$$
\begin{array}{rcl}
\mathrm{IE}(k) & = & 2 \times \mathrm{IE}(k-1) \\
\mathrm{IN}(k) & = & \mathrm{IN}(k-1) + \mathrm{IE}(k-1) \\
\mathrm{PE}(k) & = & 3 \times \mathrm{PE}(k-1) + \mathrm{IE}(k-1) \\
\mathrm{PN}(k) & = & \mathrm{PN}(k-1) + \mathrm{PE}(k-1) + \mathrm{IE}(k-1) \\
\mathrm{Total}(k) & = & \mathrm{IE}(k) + \mathrm{IN}(k) + \mathrm{PE}(k) + \mathrm{PN}(k)
\end{array}
$$

Level	IE	IN	PE	PN	Total
1	1	0	0	1	2
2	2	1	1	2	6
3	4	3	5	5	17
4	8	7	19	14	48
5	16	15	65	41	137
6	32	31	211	122	396

2.4 Atom labelling

Annotating curves with signatures and describing curves with strings of signatures is not ideal. Therefore, for notational convenience, we assign each atom at level two or above a suitable descriptive label, according to the following convention[1]: the label begins with 'P', 'Z', 'N', or 'U', depending

[1] At level 1, $\underline{\texttt{D}}$ is given the label '$\underline{\mathsf{D}}$' and $\texttt{U}$ is given the label '$\mathsf{U_b}$', where 'b' signifies that the tangent bearing component is undefined.

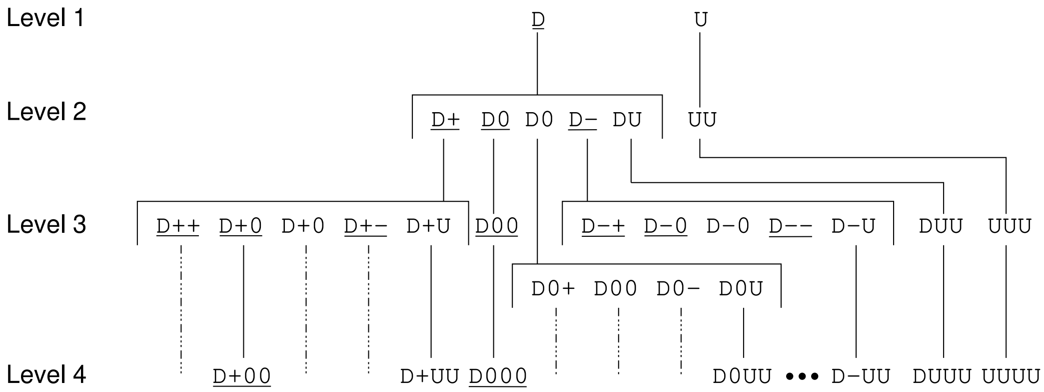

Figure 1: The first four levels of the atomic hierarchy

on whether the curvature component is positive, zero, negative, or undefined. The remaining components $(c', c'', \ldots)$ are denoted by a superscripted string, whose elements are values from the set $\{+, 0, -, U\}$. The label is underlined if the atom it is symbolising is an interval. To distinguish between the two distinct kinds of atoms where the curvature component is U, we use the label '$\mathsf{U_b}$' for atoms where the tangent bearing component is also U, and '$\mathsf{U_c}$' for atoms where the tangent bearing is D. Using the new notation, the first three sets of atomic tokens are written as follows:

Level 1 : $\underline{\mathsf{D}}, \mathsf{U_b}$

Level 2 : $\underline{\mathsf{P}}, \underline{\mathsf{Z}}, \mathsf{Z}, \underline{\mathsf{N}}, \mathsf{U_c}, \mathsf{U_b}$

Level 3 : $\underline{\mathsf{P}^+}, \underline{\mathsf{P}^0}, \mathsf{P}^0, \underline{\mathsf{P}^-}, \mathsf{P}^\mathsf{U}, \mathsf{Z}^+, \underline{\mathsf{Z}^0}, \mathsf{Z}^0, \mathsf{Z}^-, \mathsf{Z}^\mathsf{U}, \underline{\mathsf{N}^+}, \underline{\mathsf{N}^0}, \mathsf{N}^0, \underline{\mathsf{N}^-}, \mathsf{N}^\mathsf{U}, \mathsf{U_c}, \mathsf{U_b}$

2.5 Kink points

At each level in the hierarchy there exists a single kink-point atom labelled $\mathsf{U_b}$, each component of which takes the value U. Such atoms represent points on a curve where the tangent bearing (and therefore all of its derivatives) are undefined, and correspond perceptually to angles and cusps. Clearly, we would like to be able to distinguish between different kinds of kink points, otherwise we are losing important curve information. In particular, the two most relevant aspects of a kink point we would like to preserve are its orientation (whether it is inward-pointing or outward-pointing) and its type, i.e., whether it is an angle or a cusp. In order to distinguish between the different kinds of kink points, we introduce the following *kink tokens* into our representation: $\mathsf{U}_<$ and $\mathsf{U}_>$ (for inward and outward pointing angles, respectively), and $\mathsf{U}_\prec$ and $\mathsf{U}_\succ$ (for inward and outward pointing cusps, respectively). These tokens have the same status as $\mathsf{U_b}$, in that they can be thought of as appearing at every level in the hierarchy. Note, however, that kink tokens are not *atomic*, since the process by which the atomic tokens are derived does not produce them. Instead, we need to introduce them explicitly.

3 Atomic description

A curve is described by a string of atomic tokens taken from a particular level in the hierarchy. By prefixing descriptions with a symbol indicating curve type, both open and closed curves can be represented. We use '$\frown$' for open curves and '$\bigcirc$' for closed ones. Our convention for relating changes in tangent bearing to curvature is that a clockwise change in tangent bearing indicates positive curvature, while an anticlockwise change indicates negative curvature. Thus positive curvature is associated with convex curve segments and negative curvature with concave segments. A curve can be traversed in one of two directions. For a closed curve, the direction is that which preserves the intended figure/ground relationship. For open curves, consistent description dictates that one of the two directions is chosen as the "default" one.

Shown in Figure 2 are two curves that have been annotated with level-3 atoms. Note that $\underline{\mathsf{P}^0}$ refers to a convex circular arc (constant positive curvature over an interval), whereas P^0 refers to a positive curvature extremum; and analogously with $\underline{\mathsf{N}^0}$ and N^0. Given a closed curve there will be, in general, a number of different token strings that describe it. However, we can easily transform one string into another by cyclically shifting it by a certain amount. In this sense, all of the strings may be considered equivalent. Any one of sixteen level-3 strings may be used to describe the closed curve in Figure 2, two of which are obtained by either starting at the $\mathsf{U}_<$ or starting at the uppermost $\underline{\mathsf{Z}^0}$:

$$\bigcirc\, \mathsf{U}_< \, \underline{\mathsf{P}^+}\, \mathsf{P}^0\, \underline{\mathsf{P}^-}\, \mathsf{Z}^-\, \underline{\mathsf{N}^-}\, \mathsf{N}^0\, \underline{\mathsf{N}^+}\, \mathsf{U}_\succ\, \underline{\mathsf{Z}^0}\, \mathsf{U_c}\, \underline{\mathsf{P}^0}\, \mathsf{U}_>\, \underline{\mathsf{Z}^0}\, \mathsf{U}_\prec\, \underline{\mathsf{P}^0}$$

$$\bigcirc\, \underline{\mathsf{Z}^0}\, \mathsf{U_c}\, \underline{\mathsf{P}^0}\, \mathsf{U}_>\, \underline{\mathsf{Z}^0}\, \mathsf{U}_\prec\, \underline{\mathsf{P}^0}\, \mathsf{U}_<\, \underline{\mathsf{P}^+}\, \mathsf{P}^0\, \underline{\mathsf{P}^-}\, \mathsf{Z}^-\, \underline{\mathsf{N}^-}\, \mathsf{N}^0\, \underline{\mathsf{N}^+}\, \mathsf{U}_\succ$$

The importance of specifying a particular direction for open curves is illustrated by the open curve in Figure 2, where a reversal of direction yields the description $\frown \underline{\mathsf{Z}^0}\, \mathsf{U}_<\, \underline{\mathsf{N}^-}$. Note, however, that under a system where "mirror-reflected" curves are considered equivalent, the direction chosen for traversal is unimportant.

3.1 Levels of description

A curve has a description at every atomic level. The most coarse-grained description is at level 1, where only the atoms $\underline{\mathsf{D}}$ and $\mathsf{U_b}$ are available. By moving down the hierarchy (and incorporating more qualitative components) we get finer-grained descriptions that reflect the increase in discriminatory power that extra components provide.

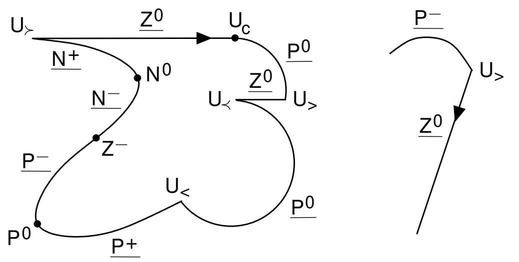

Figure 2: Example curves labelled with level-3 atoms

Given the description of a curve at some level k, we can derive all of the coarser-grained descriptions for the curve, i.e., from the description at level 1 through to the description at level $k-1$. If X denotes the description of a curve $\mathcal{C}$, at level $k > 1$, then the following straightforward three-step procedure yields the description of $\mathcal{C}$ at level $k-1$:

1. Replace each atom of X with its parent.
2. Iteratively replace substrings of the form xx (where x is a single atom) with x, until there are no more substrings of the form xx.
3. If $\mathcal{C}$ is a closed curve, and the string resulting from the previous step is of length greater than two, and begins and ends with atoms of the same type, then remove the last atom in the string.

By applying the procedure to the ellipse shown in Figure 3, we get the level-2 description $\bigcirc\,\underline{\mathsf{P}}$, by re-applying the procedure with $\bigcirc\,\underline{\mathsf{P}}$ as input, we get the most coarse-grained description for an ellipse: $\bigcirc\,\underline{\mathsf{D}}$.

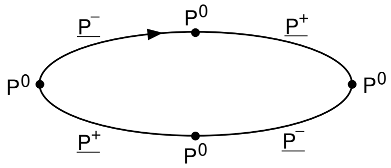

Figure 3: An ellipse described at level 3

Although we can derive all coarser-grained descriptions for a curve given its description at level $k > 1$, it is not possible, in general, to derive any of its finer-grained descriptions. The occasion when it *is* possible to derive a finer-grained description is when the description given consists entirely of atoms that are non-expansive, in which case *all* finer-grained descriptions are accessible. Consider the description of a square at level 2: $\bigcirc\,\underline{\mathsf{Z}}\,\mathsf{U}_>\,\underline{\mathsf{Z}}\,\mathsf{U}_>\,\underline{\mathsf{Z}}\,\mathsf{U}_>\,\underline{\mathsf{Z}}\,\mathsf{U}_>$. Because both $\underline{\mathsf{Z}}$ and $\mathsf{U}_>$ are non-expansive, we can get the "finer-grained" description at level 3 by replacing each atom in the string with its single child atom. The level-5 description, for example, is $\bigcirc\,\underline{\mathsf{Z}^{000}}\,\mathsf{U}_>\,\underline{\mathsf{Z}^{000}}\,\mathsf{U}_>\,\underline{\mathsf{Z}^{000}}\,\mathsf{U}_>\,\underline{\mathsf{Z}^{000}}\,\mathsf{U}_>$. Another example would be a shape that has the level-3 description $\frown\,\underline{\mathsf{P}^0}$ (i.e., a convex circular arc segment). For most shapes it would appear that there exists some minimal value of k such that its description at level k contains only non-expansive atoms. For the square, $k = 2$; descriptions at levels $k > 2$ are unnecessary, leading to no increase in "qualitative precision". For a circle, we need to go to level 3 to achieve this. In this sense, a circle might be regarded as a more complex shape than a square.

4 Token ordering graphs

A curve is described by stringing together those atoms that correspond to the "atomic" qualitative boundary features of the curve. Clearly, some features can occur together and some cannot. A token ordering graph (TOG), of the kind introduced in [Meathrel and Galton, 2000], encodes the constraints that determine the basic string syntax for a set of atoms. In this section, we show how the transition tables that underlie such graphs can be systematically constructed for each level of the atomic hierarchy.

Given a set of atoms, a TOG is a way of pictorially representing the two complementary transition tables that specify which atoms may follow which other atoms in a string. The *interval-interval* ("I-I") table tells us which interval atoms can follow which other interval atoms. The *interval-point-interval* ("I-P-I") table tells us which point atoms can occur in between each pair of interval atoms. It is the construction of these tables, then, that we are primarily interested in. The I-I and I-P-I tables for level 2 are given later, in Figure 4.

4.1 Compatibility matrices

We make use of two *compatibility matrices* when constructing I-I and I-P-I tables. First, consider the construction of an I-I table for a given level in the hierarchy. Each cell $(\text{row } x, \text{column } y)$ in an I-I table contains a tick *iff* the interval atom y can directly follow the interval atom x. Given interval atoms x and y, at level n, we can determine whether ot not y can directly follow x by making use of the matrix given in Table 1. The matrix tells us whether or not the kth qualitative components of the atoms x and y are compatible.

	$\partial^k y = +$	$\partial^k y = 0$	$\partial^k y = -$
$\partial^k x = +$	$\top$	$\partial^{k+1} x = -$	$\bot$
$\partial^k x = 0$	$\partial^{k+1} y = +$	$\top$	$\partial^{k+1} y = -$
$\partial^k x = -$	$\bot$	$\partial^{k+1} x = +$	$\top$

Table 1: Compatibility matrix for I-I tables

The kth component of y is compatible with the kth component of x, and $\mathsf{compatible}(k, y, x)$ holds, *iff* the condition in the corresponding cell is satisfied. If $\partial^k x$ and $\partial^k y$ are of opposing signs then they are definitely incompatible (signified by a $\bot$ in the appropriate cell), because a derivative cannot change from positive to negative (or vice versa) without taking the value 0 or U in between. If an interval over which ∂^k has the value 0 is followed immediately by an interval over which ∂^k is positive, then in the latter interval it must initially be increasing, i.e., ∂^{k+1} must be positive. This explains the entry for cell $(\partial^k x = 0, \partial^k y = +)$. Analogous

	$\partial^k y = +$	$\partial^k y = 0$	$\partial^k y = -$
$\partial^k x = +$	$\partial^k p = + \vee \partial^k p = U \vee$ $(\partial^k p = 0 \wedge \partial^{k+1} x = - \wedge \partial^{k+1} y = +)$	$\partial^k p = U \vee$ $(\partial^k p = 0 \wedge \partial^{k+1} x = -)$	$\partial^k p = U \vee$ $(\partial^k p = 0 \wedge \partial^{k+1} x = - \wedge \partial^{k+1} y = -)$
$\partial^k x = 0$	$\partial^k p = U \vee$ $(\partial^k p = 0 \wedge \partial^{k+1} y = +)$	$\partial^{k-1} p = U$	$\partial^k p = U \vee$ $(\partial^k p = 0 \wedge \partial^{k+1} y = -)$
$\partial^k x = -$	$\partial^k p = U \vee$ $(\partial^k p = 0 \wedge \partial^{k+1} x = + \wedge \partial^{k+1} y = +)$	$\partial^k p = U \vee$ $(\partial^k p = 0 \wedge \partial^{k+1} x = +)$	$\partial^k p = - \vee \partial^k p = U \vee$ $(\partial^k p = 0 \wedge \partial^{k+1} x = + \wedge \partial^{k+1} y = -)$

Table 2: Compatibility matrix for I-P-I tables

explanations hold for the other non-trivial entries in the table. Given interval atoms x and y at level n, then, y can directly follow x *iff* $\mathsf{can\text{-}follow}(y, x)$ holds:

$$\mathsf{can\text{-}follow}(y, x) \leftrightarrow \\ y \neq x \ \wedge \ (\forall k)(\, 1 < k \leq n \rightarrow \mathsf{compatible}(k, y, x)\,)$$

In other words, y and x must be distinct interval atoms (to ensure an underlying state change), and the components of y must be compatible with the components of x. Note that, whenever a condition in the matrix refers to an "unavailable" component of an atom α, i.e., when $k{+}1 > n$, the condition *is* satisfied because the value of $\partial^{n+1}\alpha$ is unconstrained (since it is not specified). As an example, consider the level-3 interval atoms $\underline{\mathsf{P}^+}$ and $\underline{\mathsf{P}^0}$, and whether $\underline{\mathsf{P}^0}$ can directly follow $\underline{\mathsf{P}^+}$. We start by comparing the *second* component of each atom, since all interval atoms have the same value (D) for their first component. Because both $\partial^2\, \underline{\mathsf{P}^+}$ and $\partial^2\, \underline{\mathsf{P}^0}$ take the value '+', the compatibility condition is $\top$, and therefore satisfied. The condition of compatibility for the final component pair evaluates to $\partial^4\, \underline{\mathsf{P}^+} = -$, since $\partial^3\, \underline{\mathsf{P}^+} = +$ and $\partial^3\, \underline{\mathsf{P}^0} = 0$. The condition is satisfied because $\underline{\mathsf{P}^+}$ has no fourth component ($\underline{\mathsf{P}^+}$ is a generalisation of a set of child atoms that includes the level-4 atom $\underline{\mathsf{P}^{+-}}$).

Next, consider the construction of an I-P-I table for a given level in the hierarchy. Each cell (row x, column y) in an I-P-I table contains a list of point atoms that can occur in between interval atoms x and y. For I-P-I tables, we make use of the compatibility matrix given in Table 2, which differs from the matrix for I-I tables in that the notion of compatibility concerns *three* components rather than just two. Given a point atom p and interval atoms x and y, at level n, the component $\partial^k p$ is compatible with the components $\partial^k x$ and $\partial^k y$, and $\mathsf{compatible}(k, p, x, y)$ holds, *iff* the condition in the corresponding cell is satisfied. A point atom p, then, can occur in between two interval atoms x and y (after x and before y) *iff* $\mathsf{can\text{-}occur\text{-}between}(p, x, y)$ holds:

$$\mathsf{can\text{-}occur\text{-}between}(p, x, y) \leftrightarrow \\ (\exists a, b \leq n)(\, \partial^a p \neq \partial^a x \ \wedge \ \partial^b p \neq \partial^b y\,) \\ \wedge (\forall k)(\, 1 < k \leq n \rightarrow \mathsf{compatible}(k, p, x, y)\,)$$

In other words, the underlying curve state of p must be distinct from the underlying curve states of x and y, and the qualitative components of p must be compatible with those of x and y. As before, conditions that refer to "unavailable" components are automatically satisfied. To illustrate we shall use the table to determine which point atoms can come between $x = \underline{\mathsf{P}^-}$ and $y = \underline{\mathsf{P}^0}$. We have $\partial^2 x = \partial^2 y = +$ and $\partial^3 x = -, \partial^3 y = 0$. We want to know the possible values for $\partial^1 p$, $\partial^2 p$ and $\partial^3 p$. Since the definition of $\mathsf{can\text{-}occur\text{-}between}$ imposes no constraints on the first level components, $\partial^1 p$ can be either D or U. For $k = 2$, the top-left cell of the matrix states that $\partial^2 p$ may be any of $+$, U, and 0, but 0 is ruled out because it requires that $\partial^3 x = -$ and $\partial^3 y = +$, which contradicts $\partial^3 y = 0$. Moving on to $k = 3$, we consult the middle cell in the bottom row of the matrix. We see that $\partial^3 p$ may be either U or 0, the additional constraints on the latter value being irrelevant in the present case, as $k{+}1 > 3$. Remembering that if any component is U, all subsequent components must be U too, we derive the following candidate component sequences for p:

$$\langle U, U, U\rangle, \langle D, +, U\rangle, \langle D, +, 0\rangle, \langle D, U, U\rangle$$

corresponding to the level-3 atoms $\mathsf{U_b}$, $\mathsf{P^U}$, $\mathsf{P^0}$, and $\mathsf{U_c}$. The first condition of $\mathsf{can\text{-}occur\text{-}between}$ rules out $\mathsf{P^0}$, as its underlying curve state is not distinct from that of y. Thus we conclude that only the point atoms $\mathsf{U_b}$, $\mathsf{P^U}$, and $\mathsf{U_c}$ can occur in between the interval atoms $\underline{\mathsf{P}^-}$ and $\underline{\mathsf{P}^0}$.

4.2 Transition tables

The set of atoms at level k is the union of a set of interval atoms, $\mathcal{I}_k$, and a set of point atoms, $\mathcal{P}_k$, e.g., at level 2 we have $\mathcal{I}_2 = \{\underline{\mathsf{P}}, \underline{\mathsf{Z}}, \underline{\mathsf{N}}\}$ and $\mathcal{P}_2 = \{\mathsf{Z}, \mathsf{U_c}, \mathsf{U_b}\}$. The transition tables at level k, I-I$_k$ and I-P-I$_k$, each contain $|\mathcal{I}_k|^2$ cells, one for each ordered pair of interval atoms. We refer to a cell (row x, column y) in the I-I$_k$ and I-P-I$_k$ tables by writing I-I$_k(x, y)$ and I-P-I$_k(x, y)$, respectively. The following algorithm populates the I-I and I-P-I tables for any level $k > 1$ in the hierarchy[2]:

[2] At level 1, I-I$_1$ $(\underline{\mathsf{D}}, \underline{\mathsf{D}})$ is empty and I-P-I$_1$ $(\underline{\mathsf{D}}, \underline{\mathsf{D}})$ contains $\mathsf{U_b}$.

for each $\langle x, y \rangle \in \mathcal{I}_k \times \mathcal{I}_k$ **do**
  if $\mathsf{can\text{-}follow}(y, x)$ **then** I-I$_k(x, y) \leftarrow \surd$;
  for each $p \in \mathcal{P}_k$ **do**
    if $\mathsf{can\text{-}occur\text{-}between}(p, x, y)$ **then** add p to I-P-I$_k(x, y)$;

Owing to space constraints, only the I-I and I-P-I transition tables for level 2 are given here (see Figure 4). As described in §2.5, we can substitute the single kink-point atom $\mathsf{U_b}$ with a set of kink tokens: $\{\mathsf{U}_<, \mathsf{U}_>, \mathsf{U}_\prec, \mathsf{U}_\succ\}$. In order to do this, we need to replace each $\mathsf{U_b}$ in cell (x, y) of the I-P-I table with the correct subset of kink tokens, as follows: (i) replace $\mathsf{U_b}$ with $\mathsf{U}_<$ and $\mathsf{U}_>$, and (ii) add $\mathsf{U}_\prec$ if $\partial^2 x = +$ or $\partial^2 y = +$, and add $\mathsf{U}_\succ$ if $\partial^2 x = -$ or $\partial^2 y = -$, i.e., inward-pointing cusps must be flanked by an interval of positive curvature and outward-pointing cusps by an interval of negative curvature (cf. [Galton and Meathrel, 1999, §3.2]).

	$\underline{\mathsf{P}}$	$\underline{\mathsf{Z}}$	$\underline{\mathsf{N}}$
$\underline{\mathsf{P}}$		$\surd$	
$\underline{\mathsf{Z}}$	$\surd$		$\surd$
$\underline{\mathsf{N}}$		$\surd$	

	$\underline{\mathsf{P}}$	$\underline{\mathsf{Z}}$	$\underline{\mathsf{N}}$
$\underline{\mathsf{P}}$	Z $\mathsf{U_b}$ $\mathsf{U_c}$	$\mathsf{U_b}$ $\mathsf{U_c}$	Z $\mathsf{U_b}$ $\mathsf{U_c}$
$\underline{\mathsf{Z}}$	$\mathsf{U_b}$ $\mathsf{U_c}$	$\mathsf{U_b}$	$\mathsf{U_b}$ $\mathsf{U_c}$
$\underline{\mathsf{N}}$	Z $\mathsf{U_b}$ $\mathsf{U_c}$	$\mathsf{U_b}$ $\mathsf{U_c}$	Z $\mathsf{U_b}$ $\mathsf{U_c}$

Figure 4: The I-I and I-P-I tables for level 2

4.3 Graph correspondence

A TOG consists of nodes representing atomic tokens and edges representing token ordering constraints. The nodes for interval atoms are shown as rectangles and those for point atoms as circles. Any path through a TOG, which starts and finishes at a node representing an interval atom, yields a syntactically valid string of tokens that is instantiable as an open curve and that may or may not be instantiable as a closed curve. A TOG that encodes the ordering constraints for the atoms of level 2 (incorporating the set of kink tokens) is shown in Figure 5. We have seen that we can construct I-I and I-P-I tables for any level of the atomic hierarchy. From the tables for a particular level, a corresponding TOG can be constructed that visually encodes the constraints given by the table contents. We do not here provide an algorithm for constructing TOGs from I-I and I-P-I tables; details relevant to such an algorithm may be found in [Meathrel and Galton, 2000, §5]. The TOG in Figure 5, then, corresponds to the I-I and I-P-I tables given in Figure 4 (with $\mathsf{U_b}$ replaced with the kink tokens, as described at the end of §4.2). The TOGs in Figures 2 and 4 of [Meathrel and Galton, 2000] correspond, respectively, to the I-I and I-P-I tables of levels 2 and 3 of the atomic hierarchy[3].

[3]With $\mathsf{U_b}$ *not* replaced by the more-specific kink tokens.

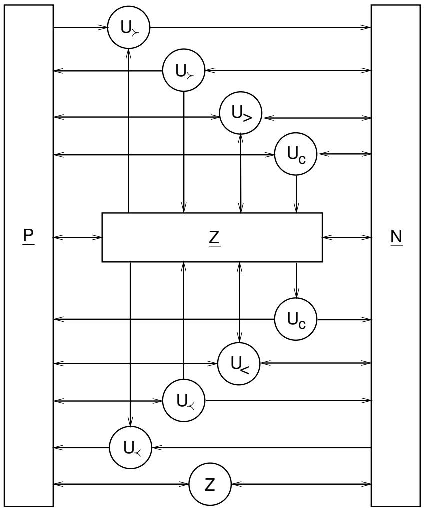

Figure 5: Level-2 TOG with distinct angle and cusp tokens

5 Conclusion

We have generalised the work described in [Meathrel and Galton, 2000], by deriving an unbounded hierarchy of atomic tokens and incorporating distinct kink tokens for inward and outward pointing angles and cusps. Both open and closed curves have qualitative descriptions, consisting of strings of atoms, at every level of the hierarchy, with the most coarse-grained description at level 1. Higher-numbered levels, where more qualitative components are included, possess greater discriminatory power and allow for finer-grained descriptions. We showed how, for each level of the hierarchy, I-I and I-P-I transition tables, encoding the ordering constraints for the set of atoms at that level, can be constructed by a simple algorithm that makes use of a pair of compatibility matrices. From a pair of I-I and I-P-I tables, a corresponding TOG can be constructed that visually encodes their constraints.

In [Meathrel and Galton, 2000], we showed how sequences of atomic tokens (with "contexts") can be used to model boundary-based schemes such as those of Leyton [1988] and Hoffman and Richards [1982]. We believe the theory presented here provides an adequate basis for constructing *any* boundary-based scheme of qualitative shape descriptors.

References

[Galton and Meathrel, 1999] A. Galton and R. Meathrel. Qualitative outline theory. In T. Dean, ed., *Proc. of 16th IJCAI*, pp. 1061–1066. Morgan Kaufmann, Inc., 1999.

[Hoffman and Richards, 1982] D. D. Hoffman and W. A. Richards. Representing smooth plane curves for recognition: implications for figure-ground reversal. In *Proc. of AAAI-82*, pp. 5–8, 1982.

[Leyton, 1988] Michael Leyton. A process-grammar for shape. *Artificial Intelligence*, 34:213–247, 1988.

[Meathrel and Galton, 2000] R. Meathrel and A. Galton. Qualitative representation of planar outlines. In W. Horn, ed., *Proc. of 14th ECAI*, pp. 224–228. IOS Press, 2000.