Mathematics & Design 98 (ed. J. Barallo), pp. 105-114, The University of the Basque Country, San Sebastian, Spain, 1998.

AN ALGEBRAIC APPROACH TO COMPARING REPRESENTATIONS

Rudi Stouffs
Architecture and CAAD, Swiss Federal Institute of Technology Zurich, Switzerland

Ramesh Krishnamurti
Department of Architecture, Carnegie Mellon University, Pittsburgh, USA

Abstract: This work is based on the recognition that there will always be a need for different representations of the same entity, albeit a building or building part, a shape or other complex attribute. This exigency ensues, formally, to define the relations between alternative representations, in order to support translation and identify where exact translation is possible, and to define coverage of different representations. We present an abstraction of representations to model sorts that allows us to define algebraic operations on sorts and recognize algebraic relationships between sorts. This approach provides us with a method for the analysis of representations, and the comparison of their coverage.

1. Introduction

As a mathematical abstraction, a model can be described by a system of mathematical equations, the solution(s) to which specify the model with respect to a chosen universe. For example, a line may be expressed as an infinite set of points, a color as an entity in some color space. Similar models, such as different lines, can be described by similar systems of equations. Generally, the variations between such similar systems can be reduced to a few numeric values within each system of equations. By parametrizing these numeric values and, possibly, specifying additional equations to constrain these parameter values where appropriate, we arrive at a single system of equations expressing an indefinite number of similar models. We denote such a set a *sort*. For example, points and lines each are a sort and so are triangles and squares. Sorts are not limited to geometrical objects, colors are a sort and so do other attributes define sorts. A sort constitutes the basic entity for our formalism; we rely on the fact that for each practical representation we can define an appropriate model sort. We explore a representational description for sorts that illustrates this correspondence between representations and sorts. We consider algebraic operations on sorts, the results of which define new sorts. We posit that the organization of a representational schema, as defined by its representational structures, corresponds to an equivalent algebraic organization.

2. A Representation for Sorts

For any sort, we consider a system of equations over a set of parameters. Such a system is derived from the equations of any individual of the sort, by introducing additional parameters that generalize these equations over all models of the sort, and equations that serve to bound the collection of individuals that constitute the sort. For example, consider a quadrilateral formed by four line segments with their respective equations. When parametrizing the endpoint coordinates, additional equations become necessary to constrain the endpoints to coincide two by two. Thus, within the description of a sort, we distinguish equations that characterize any individual and equations that specify the collection of individuals. We denote the former *characteristic equations* and the latter *instance equations*.

Likewise, we distinguish two types of parameters. *Characteristic parameters* serve to specify an individual within a chosen universe, e.g., a line in Euclidean space. *Instance parameters* serve to specify an individual within a sort, e.g., a particular line in a sort of lines. For example, in the coordinate sort with characteristic equation $x = x_c$, the characteristic parameter x specifies the coordinate as an entity on a cartesian axis, and the instance parameter x_c specifies the particular entity, the one for which x equals the given value of x_c. It follows that characteristic equations are defined over both characteristic and instance parameters, while only instance parameters occur in instance equations.

We consider the following four components in the representation of a sort: a set of instance parameters whose values specify an individual of the sort; a set of characteristic parameters whose value ranges specify an individual within a chosen universe; a system of characteristic equations over both the instance and characteristic parameters, that represents an individual given a set of values for the instance parameters; and a system of instance equations over the instance parameters only, that specifies the collection of individuals that define the sort.

Notationally, for a sort a, with a system of characteristic equations $\mathbf{A_c}$, a system of instance equations $\mathbf{A_i}$, the set of characteristic parameters $A^c = \{ a^c_1, \ldots, a^c_m \}$ and the set of instance parameters $A^i = \{ a^i_1, \ldots, a^i_m \}$, we write

$$a = \left\| \begin{matrix} \mathbf{A_c} & A^c \\ \mathbf{A_i} & A^i \end{matrix} \right\| \qquad [1]$$

and use a to denote both the sort of individuals, semantically, and its representation, syntactically.

An individual of this sort is specified by the characteristic equations and a tuple of values for the instance parameters. These values necessarily adhere to the instance equations of the sort.

In practical data descriptions, a particular representation corresponds exactly to this tuple of values, where the representational structure is an expression of the instance parameters of the corresponding sort. On the other hand, the system of characteristic equations, as uniquely defined for the entire sort, is maintained within the application that interprets the data description. The specific sort is commonly referenced by a label, a code or by the use of delimiters. For example, a representational structure for a line segment may consist of a tuple of four coordinate values of the line segment’s endpoints. A parenthesized structure for this tuple, the term “line segment” or similar, or a corresponding numeric or symbolic code ensures proper interpretation.

3. Algebraic Operations on Sorts

Complex representational schemata can be treated as compositions of other representational structures. Similarly, compositions of sorts are sorts and can be defined by algebraic operations. We consider the following operations on sorts *a* and *b*.

operation	**notation**	**definition**
subsort	$a \leq b$	every individual of *a* is an individual of *b*
sum	$a + b$	the sort of all individuals from *a* and *b*
product	$a \cdot b$	the sort of all individuals that belong to both *a* and *b*
difference	$a - b$	the sort of all individuals that belong to *a* but not to *b*
cartesian product	$a \times b$	the sort of all 2-tuples of individuals, one of which belongs to *a* and the other to *b*

Additionally, we define the *power sum* $a^{\oplus}$ as the sort of all tuples of individuals of *a*. This operation may be considered as an infinite composition of the operations of sum and cartesian product over the same sort. We write $a^{\oplus} = a + a \times a + a \times a \times a + a \times a \times a \times a + \ldots = \sum_{n=1}^{\infty} \prod_{m=1}^{n} a$, or, if we denote $a^1 = a$ and $a^n = a \times a^{n-1}$, then, $a^{\oplus} = \sum_{n=1}^{\infty} a^n$.

We define an empty sort, denoted 0, which contains no individuals. The empty sort is a subsort of every sort. It is also the result of the operation of product on two sorts that do not share any individuals. 0 is the neutral element for sum, i.e., $a + 0 = a = 0 + a$, $\forall\ a$. It follows that $a \cdot 0 = 0 = 0 \cdot a$, $a - 0 = a$ and $0 - a = 0$. We define $a \times 0 = 0 = 0 \times a$ and, thus, $0^{\oplus} = 0$.

The partial order relationship underlying the algebraic operations serves to define a subsumption relationship on representational structures that facilitates the comparison of representational structures in terms of scope and issues of translation. For example, the sum of two sorts has both operands as a part. Similarly, the product of two sorts forms a part of both sorts, while the operations of product and difference define a classification of a sort with respect to another sort into disjoint parts.

Examples

The operation of sum applied to the sorts of points and lines results in the sort that contains all points as well as all lines, i.e., an individual of this sort is either a point or a line. Similarly, the operation of cartesian product applied to these sorts results in the sort of all 2-tuples containing exactly one point and one line. These tuples specify the individuals of the sort. Considering the representation of a line segment by its two endpoints, the corresponding sort can be derived from the cartesian product of two point sorts. Similarly, the sort of triangles is the result of two cartesian products on sorts of line segments. The sort of all polygons, independent of the number of line segments, is defined from the sort of line segments by the operation of power sum.

4. Composing Sorts

A composite sort can always be represented as a syntactical composition of its operands' representation. In order to compare sorts successfully, a composite representation must be reduced whenever possible into a unique representation as specified in [1]. The reductions depend on the relationships between the respective parameter sets and their semantics. Parameters with different meanings must be distinguished by name; conversely, those with identical meaning must be identified into a single parameter. For example, the cartesian coordinates x and y have different meanings, while, by convention, each has the same meaning for all geometric equations within the same cartesian space.

We distinguish two aspects of a parameter's meaning: its *signature* and *domain*. The signature of a parameter expresses its (unique) relationship to the universe or space it operates in. This signature is assigned to a parameter by definition or convention. For example, the parameter x denotes the abscissa on the X-axis within a cartesian space. By the same convention, y denotes the ordinate on the Y-axis in this cartesian space. In a three-dimensional color universe, we can define three parameters r, g and b (for red, green and blue), each with their specific signature. Alternatively, we can define three different parameters h, s and i (for hue, saturation and intensity) that specify the same color universe. The dependencies between r, g and b on one hand, and h, s and i on the other hand, form a part of their signatures and must be taken into account when identifying or distinguishing parameters. The domain of a parameter specifies its extent within the universe or space it is defined, as independent of this parameter. As such, domains can be combined under cartesian product into multi-dimensional domains corresponding to sets of parameters. For example, the domain of parameter x is the X-axis, the set $\{x, y, z\}$ defines the three-dimensional cartesian space, $\{r, g, b\}$ defines a three-dimensional color space, and the domain of their union constitutes a six-dimensional universe of space and color. Note that the sets of color parameters $\{r, g, b\}$ and $\{h, s, i\}$ define the same domain, so do the sets of cartesian and polar coordinates.

It is clear that any system of characteristic equations can have only one characteristic parameter of the same signature, and any parameter can have only one meaning within any system of equations. When combining characteristic equations, a proper identification or, otherwise, distinction of the respective parameters according to their signature is needed to ensure the validity of these properties. Every instance parameter must have a signature that is also assigned to one of the characteristic parameters, though multiple instance parameters may share the same signature, and not every characteristic parameter must have corresponding instance parameters. Thus, the set of instance parameters defines a subdomain with respect to the set of characteristic parameters. We define the domain of a sort as the domain of its set of characteristic parameters.

Domains play an important role when composing sorts. Specifically, whether a unique representation exists for a composite sort depends on the relationship between the domains of the operand sorts. If no unique representation exists, then, we specify the syntactical composition of the operand representations under sum or cartesian product as a representation for the sort. In this case, each partial representation has its own domain, which is a subdomain of the overall domain. We distinguish between *exclusive* and *inclusive* subdomains. In a composition under sum, each subdomain specifies a subset of individuals of the sort, that is, any individual exists exclusively in one of the subdomains. We denote such subdomains as exclusive. In a composition under cartesian product, instead, each subdomain specifies a component of all individuals. We denote such subdomains as inclusive. In general, subdomains may form a hierarchy of exclusions and inclusions. Otherwise, when a unique representation exists for a sort, we say it has a *unique* domain. Note that exclusive subdomains can be identical. For example, an individual of the sort of lines and circles is either a line or a circle, and there is no single system of equations that represents both lines and circles as simultaneous solutions.

5. Subsorts under Unique Domains

Let *a* and *b* be sorts with unique domains, $a = \left\| \begin{array}{ll} \mathbf{A_c} & A^c \\ \mathbf{A_i} & A^i \end{array} \right\|$ and $b = \left\| \begin{array}{ll} \mathbf{B_c} & B^c \\ \mathbf{B_i} & B^i \end{array} \right\|$.

Consider the product of *a* and *b*, $a \cdot b$. Since the system of characteristic equations is the same for all individuals of *a*, respectively, *b*, it follows that $a \cdot b \neq 0$ only if *a*, *b* and $a \cdot b$ all have an equivalent system of characteristic equations. This implies identical characteristic parameters and domains for *a* and *b*. However, the instance parameters must not be identical, but only *identifiable*, as the following example illustrates. Consider the sort of quadrilaterals and its subsort of squares. Both representations share equivalent characteristic equations expressing these figures as four connected line segments. In two dimensions, a general quadrilateral requires eight instance parameters, representing the coordinates of the four corner points. On the other hand, a square requires as little as four instance parameters, e.g., representing the coordinates of any two corner points. The characteristic equations are identical in all other respects. We can also consider characteristic equations for squares using eight distinct instance parameters to represent the coordinates of all four corner points, while defining instance equations that relate these eight parameters and specify equal length line segments and identical angles between consecutive sets of corner points.

Thus, if the domains are identical, the instance parameters identifiable and, subsequently, the characteristic equations are identical, then, we denote these equations as *equivalent*. Note that degenerate cases also exist, where certain characteristic equations become either identical or trivial as a result of the instance equations specified on these. We will not consider such cases here. We write $\mathbf{A_c} \equiv \mathbf{B_c}$ to denote the equivalency of systems of characteristic equations $\mathbf{A_c}$ and $\mathbf{B_c}$. We adopt the same notation when identifying instance parameters, e.g., $A^i \equiv B^i$.

We have the following requirements on the sorts *a* and *b* for $a \cdot b = c \neq 0$:

i. the representational domains of *a* and *b* are identical, i.e., $C^c = A^c = B^c$;

ii. the instance parameters of *a* and *b* are identifiable, i.e., $C^i = (A^i \equiv B^i)$;

iii. the characteristic equations of *a* and *b* are equivalent, i.e., $\mathbf{C_c} = (\mathbf{A_c} \equiv \mathbf{B_c})$;

The instance equations of $c = a \cdot b$ are a composition of the instance equations of *a* and *b* under the logical connective $\wedge$ (and). If the instance equations of *a* and *b* exclude each other, then, $a \cdot b = 0$. We write the following reduction rules:

$$\left\| \begin{array}{ll} \mathbf{A_c} & A^c \\ \mathbf{A_i} & A^i \end{array} \right\| \cdot \left\| \begin{array}{ll} \mathbf{B_c} & B^c \\ \mathbf{B_i} & B^i \end{array} \right\| \rightarrow \left\| \begin{array}{ll} \mathbf{A_c} \equiv \mathbf{B_c} & A^c = B^c \\ \mathbf{A_i} \wedge \mathbf{B_i} & A^i \equiv B^i \end{array} \right\| ; \tag{2a}$$

$$\left\| \begin{array}{ll} \mathbf{A_c} & A^c \\ \mathbf{A_i} & A^i \end{array} \right\| \cdot \left\| \begin{array}{ll} \mathbf{B_c} & B^c \\ \mathbf{B_i} & B^i \end{array} \right\| \rightarrow 0, \text{ otherwise.} \tag{2b}$$

The difference of two sorts is dependent on the existence of a common subsort to both sorts. That is, $a - b = a$ if $a \cdot b = 0$. If a and b have identical domains and equivalent characteristic equations, then, the instance equations of c are a composition of the instance equations of a and the negation of the instance equations of b, under the logical connective $\wedge$. The resulting reduction rules are:

$$\left\Vert\begin{array}{ll}\mathbf{A_c} & A^c \\ \mathbf{A_i} & A^i\end{array}\right\Vert - \left\Vert\begin{array}{ll}\mathbf{B_c} & B^c \\ \mathbf{B_i} & B^i\end{array}\right\Vert \rightarrow \left\Vert\begin{array}{ll}\mathbf{A_c} \equiv \mathbf{B_c} & A^c = B^c \\ \mathbf{A_i} \wedge \neg\mathbf{B_i} & A^i \equiv B^i\end{array}\right\Vert ; \tag*{[3a]}$$

$$\left\Vert\begin{array}{ll}\mathbf{A_c} & A^c \\ \mathbf{A_i} & A^i\end{array}\right\Vert - \left\Vert\begin{array}{ll}\mathbf{B_c} & B^c \\ \mathbf{B_i} & B^i\end{array}\right\Vert \rightarrow \left\Vert\begin{array}{ll}\mathbf{A_c} & A^c \\ \mathbf{A_i} & A^i\end{array}\right\Vert \text{, otherwise.} \tag*{[3b]}$$

For the sum of two sorts, both sorts are subsorts. If a and b have identical domains and equivalent characteristic equations, then, the instance equations of c are a composition of the instance equations of a and b under the logical connective $\vee$ (or). Otherwise, we maintain the expression $a + b$ as a representation for c. The single reduction rule is:

$$\left\Vert\begin{array}{ll}\mathbf{A_c} & A^c \\ \mathbf{A_i} & A^i\end{array}\right\Vert + \left\Vert\begin{array}{ll}\mathbf{B_c} & B^c \\ \mathbf{B_i} & B^i\end{array}\right\Vert \rightarrow \left\Vert\begin{array}{ll}\mathbf{A_c} \equiv \mathbf{B_c} & A^c = B^c \\ \mathbf{A_i} \vee \mathbf{B_i} & A^i \equiv B^i\end{array}\right\Vert \tag*{[4]}$$

6. Power-sorts over Unique Domains

Whenever a composition of sorts with unique domains is reduced to a unique representation, the respective instance parameters of the operand sorts must be either identified or distinguished. In the case of cartesian product, we consider each individual of the resulting sort as composed of two components, one from either operand sort. Thus, we must necessarily distinguish the instance parameters. This may be achieved through a renaming of these parameters, if necessary. We distinguish four cases dependent on the relationship between the operands' domains:

i. the domains are identical;

ii. the domains are disjoint;

iii. the domains overlap and there is a classification of the characteristic equations such that the respective subdomains are either identical or disjoint;

iv. the domains overlap and no such classification exists.

The resulting representations for the four cases are given below. The union of sets of instance parameters always implies a distinguishing of these parameters.

i	**ii**	**iii**	**iv**
$\left\Vert\begin{array}{ll}\mathbf{A_c} \vee \mathbf{B_c} & A^c = B^c \\ \mathbf{A_i} \wedge \mathbf{B_i} & A^i \cup B^i\end{array}\right\Vert$	$\left\Vert\begin{array}{ll}\mathbf{A_c} \wedge \mathbf{B_c} & A^c \cup B^c \\ \mathbf{A_i} \wedge \mathbf{B_i} & A^i \cup B^i\end{array}\right\Vert$	$\left\Vert\begin{array}{cc}(\mathbf{A_c^{\equiv}} \vee \mathbf{B_c^{\equiv}}) \wedge \mathbf{A_c^{\perp}} \wedge \mathbf{B_c^{\perp}} & A^c \cup B^c \\ \mathbf{A_i} \wedge \mathbf{B_i} & A^i \cup B^i\end{array}\right\Vert$	$a \times b$

In case **i** the domains are identical. Therefore, we distinguish the instance parameters and join the respective sets under union, in order to denote the two components of each individual of the resulting sort. This results in two distinct systems of characteristic equations defined over the same characteristic parameters. We then combine these under the connective $\vee$. Because instance parameters must adhere to both systems of instance equations, these are combined under the connective $\wedge$. In case **ii** the domains are disjoint. Again, we join the sets of instance parameters, under union, to denote the two components of each individual. Similarly, we combine both systems of instance equations under the connective $\wedge$. Since the domains are disjoint, we join both sets of characteristic parameters under union and combine the systems of characteristic equations under the connective $\wedge$. The systems are defined over disjoint domains; these must be satisfied simultaneously.

For case **iii**, we consider a classification of the characteristic equations such that the respective subdomains are either identical or disjoint. This allows us to adopt the results of cases **i** and **ii**. As the domains overlap, the sets of characteristic parameters A^c and B^c must have a common subset, i.e., $A^c \cap B^c \neq \emptyset$. Consider a classification of these sets into the subsets $A^c \cap B^c$, $A^c \setminus B^c$ and $B^c \setminus A^c$. If a classification of the characteristic equations over identical and disjoint subdomains exists, then, such a classification must also exist, necessarily, over the domains corresponding to $A^c \cap B^c$, $A^c \setminus B^c$ and $B^c \setminus A^c$. Any other such classification can always be reduced to the above one by combining the appropriate subclasses. Let $\mathbf{A}_\mathbf{c}^{\equiv}$ denote the subsystem of characteristic equations from $\mathbf{A}_\mathbf{c}$ over the characteristic parameters of $A^c \cap B^c$, and similarly for $\mathbf{B}_\mathbf{c}^{\equiv}$ from $\mathbf{B}_\mathbf{c}$. Let $\mathbf{A}_\mathbf{c}^{\perp}$ denote the subsystem of remaining characteristic equations from $\mathbf{A}_\mathbf{c}$, and similarly for $\mathbf{B}_\mathbf{c}^{\perp}$ from $\mathbf{B}_\mathbf{c}$. Then, case **i** specifies the combination of the systems $\mathbf{A}_\mathbf{c}^{\equiv}$ and $\mathbf{B}_\mathbf{c}^{\equiv}$, while case **ii** specifies all other combinations. Thus, the resulting expression for the system of characteristic equations for the resulting sort is $(\mathbf{A}_\mathbf{c}^{\equiv} \vee \mathbf{B}_\mathbf{c}^{\equiv}) \wedge \mathbf{A}_\mathbf{c}^{\perp} \wedge \mathbf{B}_\mathbf{c}^{\perp}$. The corresponding set of characteristic parameters is $(A^c \cap B^c) \cup (A^c \setminus B^c) \cup (B^c \setminus A^c) = A^c \cup B^c$. Similarly, the resulting set of instance parameters is $A^i \cup B^i$ as this is identical for both cases. Also, the resulting system of instance equations is $\mathbf{A}_\mathbf{i} \wedge \mathbf{B}_\mathbf{i}$. Note that both cases **i** and **ii** may be considered special cases of case **iii**. Finally, in case **iv**, we adopt the expression $a \times b$ as a representation for c.

A single reduction rule remains for cartesian product of sorts with unique domains:

$$\left\| \begin{matrix} \mathbf{A}_\mathbf{c} & A^c \\ \mathbf{A}_\mathbf{i} & A^i \end{matrix} \right\| \times \left\| \begin{matrix} \mathbf{B}_\mathbf{c} & B^c \\ \mathbf{B}_\mathbf{i} & B^i \end{matrix} \right\| \rightarrow \left\| \begin{matrix} (\mathbf{A}_\mathbf{c}^{\equiv} \vee \mathbf{B}_\mathbf{c}^{\equiv}) \wedge \mathbf{A}_\mathbf{c}^{\perp} \wedge \mathbf{B}_\mathbf{c}^{\perp} & A^c \cup B^c \\ \mathbf{A}_\mathbf{i} \wedge \mathbf{B}_\mathbf{i} & A^i \cup B^i \end{matrix} \right\| \qquad [5]$$

with $\mathbf{A}_\mathbf{c} \equiv \mathbf{A}_\mathbf{c}^{\equiv} \wedge \mathbf{A}_\mathbf{c}^{\perp}$ and $\mathbf{B}_\mathbf{c} \equiv \mathbf{B}_\mathbf{c}^{\equiv} \wedge \mathbf{B}_\mathbf{c}^{\perp}$, while $\mathbf{A}_\mathbf{c}^{\perp} \wedge \mathbf{B}_\mathbf{c}^{\perp}$ is defined over $(A^c \setminus B^c) \cup (B^c \setminus A^c)$ and $\mathbf{A}_\mathbf{c}^{\equiv} \vee \mathbf{B}_\mathbf{c}^{\equiv}$ is defined over $A^c \cap B^c$.

We expand the operation of power sum into an infinite sum of cartesian products: $a^{\oplus} = \sum_{=}^{\infty} a^n$, with $a^1 = a$ and $a^n = a \times a^{n-1}$. $a^n = \left\| \begin{matrix} \mathbf{A}_\mathbf{c} & A^c \\ \mathbf{A}_\mathbf{i} & A^i \end{matrix} \right\|^n$ is a composition under cartesian product over sorts with identical domain; case **i** applies. Let $\bigcup A_m^i$ denote the union of n times A^i, upon distinguishing all parameters. Let $\bigvee_{m=1}^{n} \mathbf{A}_\mathbf{c}^m$ and $\bigwedge_{m=1}^{n} \mathbf{A}_i^m$ denote the respective compositions of characteristic and instance equations. Since the resulting system of characteristic equations is different for each n, no further reduction applies to $a^{\oplus}$. The following reduction rules result:

$$a^{\oplus} \rightarrow \sum_{=}^{\infty} a^{n} \qquad [6]$$

$$\begin{Vmatrix} \mathbf{A}_{\mathbf{c}} & A^{c} \\ \mathbf{A}_{\mathbf{i}} & A^{i} \end{Vmatrix}^{n} \rightarrow \begin{Vmatrix} \bigvee_{m=1}^{n} \mathbf{A}_{\mathbf{c}}^{m} & A^{c} \\ \bigwedge_{m=1}^{n} \mathbf{A}_{i}^{m} & \bigcup_{-}^{n} A_{m}^{i} \end{Vmatrix} \qquad [7]$$

7. A Semi-canonical Form

If no unique representation exists for a sort, we consider a semi-canonical representational form using algebraic sum and cartesian product over at most two levels. The top level is a composition over sum of one or more subsorts, each of which is a composition over cartesian product, at the second level, of subsorts with unique domains. Naturally, either or both levels may be absent depending on the particular sort. In order to achieve this semi-canonical form, we consider additional reduction rules that use distributive properties to reorganize the algebraic operations. Product and difference are distributed until these apply to sorts with unique domains, or are eliminated. Cartesian product distributes over sum, and all three operations of cartesian product, sum and product are associative as well as commutative.

$(a + b) \times c \rightarrow (a \times c) + (b \times c)$ [8]

$(a + b) \cdot c \rightarrow (a \cdot c) + (b \cdot c)$ [9]

$(a + b) - c \rightarrow (a - c) + (b - c)$ [10]

$c - (a + b) \rightarrow (c - a) + (c - b)$ [11]

While the operation of cartesian product distributes over the operation of sum ([8]), the opposite does not hold. As an example, consider the sorts of *points*, *colors* and *labels*. The sort of all points with either a color or a label assigned is identical to the sort of all colored and all labeled points. We write *points* × (*colors* + *labels*) = (*points* × *colors*) + (*points* × *labels*). However, no such derivation holds for the sort of all colored points and all labels, (*points* × *colors*) + *labels*. Similarly, product and difference do not generally distribute over cartesian product. Only in special cases may we consider some form of distribution:

$(a \times b) \cdot (c \times d) \rightarrow (a \cdot c) \times (b \cdot d)$ if $a \cdot c \neq 0$ and $b \cdot d \neq 0$; [12a]

$(a \times b) \cdot c \rightarrow 0$, otherwise. [12b]

$(a \times b) - (c \times d) \rightarrow (a - c) \times (b - d)$ if $a - c \neq 0$ and $b - d \neq 0$; [13a]

$(a \times b) - c \rightarrow a \times b$ and [13b]

$c - (a \times b) \rightarrow c$, otherwise. [13c]

$(a \times c) + (b \times c) \rightarrow (a + b) \times c$ if $a + b$ has a unique domain, i.e., [4] applies. [14]

Rule [14], the inverse of rule [8], applies only in the special case that $a + b$ has a unique domain, while this domain overlaps with the domain of c, but no classification of the characteristic equations with respective identical and disjoint domains exists. Consider c any three-dimensional geometry. Let a and b be the sorts of rectangles and rhombs in two dimensions. Such combinations of 2D and 3D geometries are commonplace in CAD libraries. The resulting sort is best represented as the cartesian product of the sort of rectangles and rhombs with c. If a or b is a subsort of the other, then, the resulting expression simplifies to $a \times c$ or $b \times c$, respectively.

8. A Part Relation on Sorts

The algebraic operations of sum, difference, product, cartesian product and power sum are closed within the universe of sorts. It can easily be shown that this universe defines an algebra under the operations of sum and difference, based on a subsumption relation, ≤, that is a *partial order* relation. The reduction rules for sorts give us an insight into when one sort is a subsort of another.

Let D denote the universe of sorts. According to the semi-canonical form, let D_0 denote the set of sorts with unique domains, D_1 the set of sorts with inclusive subdomains, and D_2 the set of sorts with exclusive subdomains. D_0, D_1 and D_2 specify a classification of the universe of sorts D. Based on syntactical considerations, we can expect a sort $a \in D_i$ to be a part of a sort $b \in D_j$ only if $i \leq j$. We can find the necessary support for this proposition in the reduction rules for representational expressions. We have that $a \leq b$ if and only if $a + b = b$, $a \cdot b = a$ and $a - b = 0$. The reduction rules for sum ([4] and [14]) give us an insight into the cases when $a + b = b$. Assume $a \in D_i$ and $b \in D_j$. We show that $a + b \in D_k$ with $k \geq max\{i,j\}$ always holds, even after reduction, on condition that a and b are in semi-canonical form. Initially, we have $a + b \in D_2$. Rule [4] reduces this to D_0, but only if both a and b belong to D_0, and similar for rule [14] and D_1. No other computations reduce an expression over sum to either D_0 or D_1. This proves the conjecture, $i \leq j$. It also follows that if $b \in D_0$, necessarily $a \in D_0$, and similar if $b \in D_1$, $a \in D_1$. Thus, for $a \in D_i$ to be a part of $b \in D_j$, j must either be equal to i or equal to 2.

This same result can also be derived from rules [2a], [9] and [12a], which specify the reduction of product in a representational expression. Or, we can use these to determine the exact conditions for $a \leq b$. For example, rule [2a] specifies for subsorts within D_0 that $a \cdot b = a$ if and only if the sorts a and b have identical domains, equivalent characteristic equations, and the instance equations of b form a subsystem of the instance equations of a.

9. Conclusion

Sorts present an abstraction of representations that provide us with an algebraic approach for comparing representations. The goal of this work is to lay the foundation for a multi-way communication system based on sorts, providing the ability to identify when and where exact translation is possible. Stouffs et al (1996) present a vocabulary of data classes, subclasses and subsumption relations for solid representations and consider the coverage of a solid model in terms of the range of possible shape subclasses it can depict. Stouffs and Krishnamurti (1997) present an implementation of sorts with an emphasis on the representation of information for the World-Wide Web.

10. References

Stouffs, R. and Krishnamurti, R.: 1997, Sorts: A concept for representational flexibility, *in* R. Junge (ed.), *CAAD Futures 1997*, Kluwer Academic, Dordrecht, The Netherlands, pp. 553-564.

Stouffs, R., Krishnamurti, R. and Eastman, C.M.: 1996, A formal structure for nonequivalent solid representations, *in* S. Finger, M. Mäntylä and T. Tomiyama (eds.), Proceedings of *IFIP WG 5.2 Workshop on Knowledge Intensive CAD II*, International Federation for Information Processing, Working Group 5.2., pp 269-289.