Lexicographic products in metarouting

Alexander J. T. Gurney
Computer Laboratory
University of Cambridge
Email: Alexander.Gurney@cl.cam.ac.uk

Timothy G. Griffin
Computer Laboratory
University of Cambridge
Email: Timothy.Griffin@cl.cam.ac.uk

***Abstract*—**

Routing protocols often keep track of multiple route metrics, where some metrics are more important than others. Route selection is then based on *lexicographic comparison*: the most important attribute of each route is considered first, and if this does not give enough information to decide which route is better, the next attribute is considered; and so on.

We investigate protocols that find globally optimal paths and protocols that find only locally optimal paths. In each case we characterize exactly when lexicographic products can be used to define well-behaved routing protocols. We apply our results to protocols that can partition a network into distinct administrative regions, such as OSPF areas and BGP autonomous systems. We show that in some cases this type of local autonomy is fully compatible with global optimality.

I. Introduction

We can think of an idealized routing protocol as comprised of three distinct components,

$$\begin{aligned}\text{routing protocol} \quad = \quad & \text{routing language} \\ & + \text{routing algorithm} \\ & + \text{proof,}\end{aligned}$$

where the protocol's *routing language* is used to configure a network and the *routing algorithm* is used to compute solutions to network configurations specified using the routing language.

The proof component is some type of argument that the protocol behaves correctly. Proving correctness for even simple protocols is often extremely challenging. The very least we should hope for is that no matter how a network is configured, a protocol will compute a network routing for that topology in a finite amount of time, and that routing should not contain any forwarding loops. The proof component of a protocol may be informal, or may not exist at all, as is the case with the provably incorrect Border Gateway Protocol (BGP) [16], [19].

In addition to correctness, there remains the important issue of what type of routing solutions are computed. *Globally optimal* routing seeks to find best paths over all possible paths, whereas *locally optimal* routing seeks to find the best paths at each node given only the best paths of adjacent nodes. João Sobrinho has shown that path-vector algorithms — such as BGP's — can find locally optimal routing solutions even when no globally optimal solution exists [23]. This requires that the application of routing policy always increases the cost of a path. On the other hand, if policy application respects the preference ordering of paths (that is, it is *monotone*), then Sobrinho has shown [22] that globally optimal solutions can be computed with generalizations of Dijkstra's algorithm (in Section III we show that this is closely related to results in generalized pathfinding with semirings [3], [10], [11]).

Griffin and Sobrinho [14] proposed *metarouting* as a means of defining routing protocols in a high-level and declarative manner. Metarouting is based on using a *metalanguage* to specify routing languages. The goal is to design an expressive metalanguage so that the algebraic properties required by algorithms can be derived automatically from a metalanguage specification, in much the same way that types are derived in programming languages.

A special algebraic construct, scoped product, was introduced in [14] to model BGP's partitioning of the network into autonomous systems. In Section II we show how scoped product can be *defined* in terms of *lexicographic* products together with a few very simple operators. Furthermore, we generalize the scoped product and use it to model other types of policy partitioning, including OSPF-like areas. Characterizing exactly when such partitioning is compatible with global or local optima reduces to the characterization of lexicographic products — the main topic of the this paper.

Metarouting as defined in [14] was based on Sobrinho's routing algebras [23]. Before turning to our study of lexicographic constructs, we first revisit the mathematical foundations of metarouting in Section III. Here we take a strictly *algebraic* approach and base metarouting on what we call the *quadrant model of algebraic routing* [13]. This framework includes Sobrinho's routing algebras as a special case. We do this because the new model is more expressive and it firmly locates metarouting within the context of the literature on semirings and their extensions [3], [10], [11].

In Section IV we show exactly what properties are required in order to ensure either global or local optima when using lexicographic products. In order to carry this out in full generality, we define and explore a lexicographic product operator on semigroups.

Section V applies our results to the policy partioning described in Section II. We show that in some cases this type of local autonomy is fully compatible with global optimality.

1-4244-1588-8/07/$25.00 ©2007 IEEE

II. A MODEL OF POLICY PARTITIONS

Several existing protocols partition networks into distinct administrative regions and behave differently within a region and between regions. OSPF and IS-IS have *areas*, while BGP has *autonomous systems* (ASes). Here we show how we can use lexicographic constructs, together with a few other simple operations, to *define algebraically* various types of administrative network partitioning. The importance of an algebraic definition is that the complexity of these operations is shifted to the *routing language* so the generic routing algorithms can be kept as simple as possible.

We will illustrate this using *order transforms*, which will be further discussed in Section III. First, we establish a few basic definitions.

A *preorder* is a relation that is reflexive and transitive. Preordered sets will typically be written as $(S, \lesssim_S)$. Some additional properties a preorder might have are as follows:

reflexive	$x \lesssim_S x$
transitive	$x \lesssim_S y \wedge y \lesssim_S z \implies x \lesssim_S z$
antisymmetric	$x \lesssim_S y \wedge y \lesssim_S x \implies x = y$
full	$x \lesssim_S y \vee y \lesssim_S x$

in each case for all x, y and z in S. Special kinds of preorder include partial orders (which are antisymmetric), preference relations (which are full) and total orders (which are full and antisymmetric).

For a preorder $\lesssim$, we use the following symbols for various useful derived relations:

$$\begin{aligned} x \sim y &\iff \quad x \lesssim y \ \wedge \quad y \lesssim x \\ x < y &\iff \quad x \lesssim y \ \wedge \neg(y \lesssim x) \\ x \# y &\iff \neg(x \lesssim y) \wedge \neg(y \lesssim x) \end{aligned}$$

For ordered sets, the lexicographic product is well understood. Let $(S, \lesssim_S)$ and $(T, \lesssim_T)$ be preordered sets. Then their *lexicographic product* is $(S \times T, \lesssim)$, where

$$\begin{aligned} (s_1, t_1) \lesssim (s_2, t_2) \iff \quad & s_1 <_S s_2 \\ & \vee (s_1 \sim_S s_2 \wedge t_1 \lesssim_T t_2). \end{aligned}$$

Note the use of '$\sim_S$' rather than '$=$' on the right hand side: we respect the ordering of equivalent elements in S. The lexicographic product models an order where S is more important than T: the second order is used only to break ties arising from the first.

Now, order transforms are algebraic structures of the form

$$(S, \lesssim_S, F_S),$$

where S is a set of weights, $\lesssim_S$ is a preorder over S, and F_S is a subset of the functions mapping S to S.

The network topology is modeled by a directed graph, where each directed arc (i, j) is associated with a function $f_{(i,j)}$ in F, and a node *originates* values from S. The weight of a path

$$p = ((i_1, i_2), (i_2, i_3), \ldots, (i_{k-1}, i_k))$$

will then be

$$w(p) = \left(f_{(i_1,i_2)} \circ f_{(i_2,i_3)} \circ \cdots \circ f_{(i_{k-1},i_k)}\right)(a),$$

where a is originated by the node i_k. We can think of $w(p)$ as the weight of a route that could carry traffic from source node i_1 to destination node i_k. BGP is perhaps the best example of a protocol whose approach to path weight computation is 'functional'.

We can now define the lexicographic product of order transforms as

$$(S, \lesssim_S, F_S) \vec{\times} (T, \lesssim_T, F_T) := (S \times T, \lesssim, F),$$

where $\lesssim$ is the lexicographic product of $\lesssim_S$ and $\lesssim_T$, and F is the set $\{(f, g) \mid f \in F_S, g \in F_T\}$ of functions over $(S \times T) \to (S \times T)$, where $(f, g)(s, t) := (f(s), g(t))$.

We define two other constructions on order transforms,

$$\begin{aligned} \mathsf{right}((S, \lesssim, F)) &:= (S, \lesssim, \{\mathrm{id}\}), \\ \mathsf{left}((S, \lesssim, F)) &:= (S, \lesssim, \{\kappa_b \mid b \in S\}), \end{aligned}$$

where κ_b is the constant function $\kappa_b(a) := b$. In terms of routing, we can see that $\mathsf{left}(T)$ is similar to the *local preference* attribute of BGP — the last link completely determines the value. On the other hand, $\mathsf{right}(T)$ is similar to the *origin* attribute of BGP — once a value is originated it can only be copied by the identity function.

If we have two order transforms on the same preordered set we can define their *disjoint function union* as

$$(S, \lesssim, F) + (S, \lesssim, G) := (S, \lesssim, F + G),$$

where

$$F + G \quad := \quad (\{1\} \times F) \cup (\{2\} \times G).$$

The application of these functions is as if the tags did not exist:

$$\begin{aligned} (1, f)(a) &:= f(a), \\ (2, g)(a) &:= g(a). \end{aligned}$$

This disjoint union allows us to combine two distinct sets of functions that represent very different types of transformations.

We are now ready to define our first partioning operations, which will model the kind of partitioning seen in BGP. Suppose we have two order transforms,

$$\begin{aligned} S &\equiv (S, \lesssim_S, F), \\ T &\equiv (T, \lesssim_T, G). \end{aligned}$$

We define their *scoped product* as

$$S \ominus T \quad := \quad (S \vec{\times} \mathsf{left}(T)) + (\mathsf{right}(S) \vec{\times} T).$$

This structure is similar to the lexicographic product of S and T — at each node of the graph we have weights of the form (a, b), and such weights are compared using the lexicographic order. However, there are two kinds of arcs — inter-region arcs labelled with functions of the form $(1,\ (f,\ \kappa_c))$, and intra-region arcs labelled with functions of the form $(2,\ (\text{id},\ g))$. The inter-region functions change the first (most significant) component of the pair, while *originating a new second component*. The intra-region functions simply copy the "inter-region information" contained in the first component, while transforming the second component according to T.

h	$h(a,\ b)$
$(1,\ (f,\ \kappa_c))$	$(f(a),\ c)$
$(2,\ (\text{id},\ g))$	$(a,\ g(b))$

This is not the only way of combining these operators to achieve a useful result. The policy partitioning

$$S \,\Delta\, T := (S \vec{\times} T) + (\mathsf{right}(S) \vec{\times} T),$$

results in the table

h	$h(a,\ b)$
$(1,\ (f,\ g))$	$(f(a),\ g(c))$
$(2,\ (\text{id},\ g))$	$(a,\ g(b))$

This is similar to the use of areas in OSPF.

Many other partitioning schemes can be described using this language. Once the interaction of these fundamental operators with key properties is fully understood, we will automatically also know when the policy partition operators can be used in algorithms finding local or global optima.

The properties of interest are well-established, although the precise definitions have varied depending on the mathematical model employed. We will describe them as they appear for order transforms, with a more complete account in Section IV.

Ultimately, we want to know if an order transform is *increasing* or *monotonic*[1]. The 'increasing' property relates to finding local optima, whereas 'monotonicity' allows global optima to be computed. This is not a distinction between algorithms: although we could use a Dijkstra-like method for finding global optima in the monotonic case, other methods could just as well be used instead.

Let $(S, \lesssim_S, F_S)$ be an order transform. Then define

$$\mathrm{I}(S) \iff \forall a \in S, f \in F_S : a \neq \top \implies a <_S f(a)$$
$$\mathrm{M}(S) \iff \forall a, b \in S, f \in F_S : a \lesssim_S b \implies f(a) \lesssim_S f(b)$$

to characterise when S is increasing (I) or monotonic (M) respectively. It can be difficult to prove whether or not these hold for a given order transform, but by giving a language for these structures we allow such properties to be inferred straightforwardly.

[1] The reader should be alerted to the fact that we have adopted more standard order-theoretic terminology that is not the same as that used by Sobrinho [14], [22], [23]. Those works use the term isotonicity for what we are calling monotonicity, and use the term monotonicity for what we are calling non-decreasing.

The original metarouting paper only treated locally optimal routing. The only results were sufficient conditions for local optimality, specifically (using the new notation):

$$\begin{aligned} \mathrm{I}(S) \wedge \mathrm{T}(S) &\implies \mathrm{ND}(S) \\ \mathrm{ND}(S) \wedge \mathrm{ND}(T) &\implies \mathrm{ND}(S \vec{\times} T) \\ \mathrm{I}(S) \vee (\mathrm{ND}(S) \wedge \mathrm{I}(T)) &\implies \mathrm{I}(S \vec{\times} T) \end{aligned}$$

where ND is a non-strict form of I, given by

$$\mathrm{ND}(S) \iff \forall a \in S, f \in F_S : a \lesssim_S f(a),$$

and T is

$$\mathrm{T}(S) \iff \forall f \in F_S : f(\top) = \top.$$

The second of these conditions was also previously shown by Gouda and Schneider [12]. They addressed monotonicity as well, giving a sufficient condition for $S \vec{\times} T$ to be monotonic; we will discuss conditions like this in Section IV.

In this paper we seek an exact characterization of the properties required of lexicographic products for both globally and locally optimal routing. This means that our conditions will be both necessary and sufficient. As such, not only are we able to capture a wider range of examples than before, but we also enable reasoning about the *absence* of important properties. Then, if an algebra fails to meet the required standards, we will be able to deduce exactly which components are at fault, and in what way. We anticipate that this will be useful in the design of routing languages.

III. The Quadrants Model

This section briefly reviews the quadrants model of algebraic routing. Further details can be found in [13].

A *semigroup* is a set together with an associative binary operation. We will usually write semigroups as $(S, \oplus_S)$ or $(S, \otimes_S)$. The operation may have an identity (which is unique if it exists), denoted α or $\alpha_{\oplus_S}$, or an absorbing element, denoted ω or $\omega_{\oplus_S}$; these satisfy

$$\alpha \oplus_S s = s = s \oplus_S \alpha \qquad \omega \oplus_S s = \omega = s \oplus_S \omega$$

for all s in S. A semigroup with identity is a *monoid*.

There are two approaches to weight computation in the literature, which we will refer to as *algebraic* and *functional*. The functional approach is described in Section II. In the algebraic approach we use a semigroup, $(S, \otimes_S)$, and to each directed arc (i, j) in a network graph we associate an arc weight $w(i, j)$ in S. The weight of a path $p = i_1, i_2, i_3, \ldots, i_k$ is then calculated as

$$w(p) = w(i_1, i_2) \otimes_S w(i_2, i_3) \otimes_S \cdots \otimes_S w(i_{k-1}, i_k),$$

where the empty path is usually given the weight $\alpha_\otimes$, the identity element for $\otimes$.

Turning to weight summarization (finding 'best paths'), the literature again contains two distinct approaches, which we will refer to as *ordered* and *algebraic*. In the ordered approach,

weight computation	weight summarization: **algebraic**	weight summarization: **ordered**
algebraic	Bisemigroups $(S, \oplus_S, \otimes_S)$ Semirings [3], [10], [11] Nondistributive semirings [17], [18]	Order Semigroups $(S, \lesssim_S, \otimes_S)$ Ordered semigroups [1], [9], [15] QoS algebras [21]
functional	Semigroup Transforms $(S, \oplus_S, F_S)$ Monoid endomorphisms [10], [11]	Order Transforms $(S, \lesssim_S, F_S)$ Sobrinho structures [14], [23].

Fig. 1. The 'quadrants model' of algebraic routing

we are given a pre-ordered set $(S, \lesssim_S)$, and we use $\lesssim_S$ to select minimal (most preferred) elements of S out of some subset provided to us. In contrast, the algebraic approach uses a semigroup $(S, \oplus_S)$, and from two weights a and b in S we compute the new weight $a \oplus_S b$. This might coincide with a or b, or it might be a new element entirely. This formulation is used for path counting and related problems: $(S, \oplus_S)$ might for example be $(\mathbb{N}, +)$.

Figure 1 presents the four ways we can combine the algebraic and ordered approaches to weight summarization with the algebraic and functional approaches to weight computation. We discuss each in more detail.

The upper left quadrant contains *bisemigroups*, which have the form $(S, \oplus_S, \otimes_S)$. Semirings [3], [10], [11] are included in this class: these are bisemigroups in which $\otimes_S$ distributes over $\oplus_S$, the $\oplus_S$ operation is commutative, and there is an identity $\alpha_\oplus$. Some formulations of the semiring concept also require that $\alpha_\oplus$ should be an absorbing element for $\otimes_S$, or that $\otimes_S$ should have an identity $\alpha_\otimes$. We however do not insist on any of these conditions—they will be inferred as required. Thus, the class of bisemigroups includes nondistributive semirings [17], [18]. Sobrinho has shown that On the whole we expect BGP-like interdomain routing to be non-distributive [23]. However, some components of interdomain routing can be isolated and modeled with distributive algebras [2].

Important examples from this quadrant include the bisemigroups $(\mathbb{N}, \min, +)$ and $(\mathbb{N}, \max, \min)$, which can be used for finding shortest-distance and greatest-bandwidth paths respectively. We also have $(\mathbb{N}, +, \times)$, for counting the total number of paths.

Moving to the lower left, we have *semigroup transforms*, in which the $\otimes_S$ of a bisemigroup is generalized to a set of functions acting on a semigroup. This means that we can transform weights in essentially arbitrary ways: rather than having all weight computations being of the form $s_1 \otimes_S s_2$, we can restrict ourselves to a more limited set of transformations, or expand to take in any function at all on S. In [11] the functions are required to be homomorphisms ($f(s_1 \oplus_S s_2) = f(s_1) \oplus_S f(s_2)$), a condition which is the equivalent of requiring an order to be monotonic. Here, we permit our functions to be other than homomorphisms, but we will infer whether they are or not.

For every bisemigroup $(S, \oplus_S, \otimes_S)$, there is a corresponding semigroup transform $(S, \oplus_S, F_S)$, where F_S is the set $\{\lambda y.(x \otimes_S y) \mid x \in S\}$. Other ways of making new semigroup transforms from old include the local preference, origin preference, and disjoint function union operators from Section II.

The upper right quadrant of Figure 1 contains the *order semigroups*, which have the form $(S, \lesssim_S, \otimes_S)$. An important subclass of these are the ordered semigroups, which have been studied extensively [1], [9], [15]. For ordered semigroups, $\otimes_S$ is required to be *monotonic* with respect to $\lesssim_S$: that is, if $a \lesssim_S b$, then $c \otimes_S a \lesssim_S c \otimes_S b$ and $a \otimes_S c \lesssim_S b \otimes_S c$. Sobrinho [21] studied such structures (with total orders) in the context of Internet routing. In our framework, we require only that $\lesssim_S$ be a preorder. In keeping with our design principle of not requiring anything which can be inferred, we do not enforce monotonicity (which is why we call these structures 'order semigroups' rather than 'ordered semigroups').

Some useful order semigroups are $(\mathbb{N}, \leq, +)$ for shortest distances, $(\mathbb{N}, \geq, \min)$ for greatest bandwidths, and for most reliable paths, $([0,1], \geq, \times)$. Sobrinho showed that

$$\mathrm{M}((\mathbb{N}, \leq, +) \vec{\times} (\mathbb{N}, \geq, \min)) \text{ and } \neg\mathrm{M}((\mathbb{N}, \geq, \min) \vec{\times} (\mathbb{N}, \leq, +)),$$

a good illustration of the difficulties involved in constructing lexicographic products. Simply knowing these facts does not illuminate why it is that monotonicity fails in the second case, and so it is unclear how the situation can be resolved if we intend to select routes by bandwidth first, then delay. In our approach, monotonicity and other properties of the product are deduced from properties of the components, and our emphasis on complete characterisation of properties allows the reasons for failure to be straightforwardly identified.

In the lower right quadrant of Figure 1, we have *order transforms*, structures of the form $(S, \lesssim_S, F_S)$, as previously discussed in Section II. Order transforms include Sobrinho's routing algebras [23] as a special case. Sobrinho algebras (as defined in [14]) have the form

$$(S,\ \preceq,\ L,\ \bullet),$$

where $\preceq$ is a *preference relation* over S (that is, a full preorder), L is a set of *labels*, and $\bullet$ is a function mapping $L \times S$ to S. As an order transform, this is $(S, \preceq, F_L)$ with $F_L = \{g_\lambda \mid \lambda \in L\}$, where $g_\lambda(a) = \lambda \bullet a$. Thus we can think of the pair $(L, \bullet)$ as a means of *indexing* the set of functions F_L. In addition to this slightly higher level of abstraction, we do not insist that $\lesssim_S$ be total.

As with semigroup transforms and bisemigroups, we can construct order transforms from order semigroups via the 'Cayley' map: $(S, \lesssim_S, \otimes_S)$ becomes $(S, \lesssim_S, F_S)$ where $F_S = \{\lambda y.(x \otimes_S y) \mid x \in S\}$.

There are also translations between the left and the right halves of the table: between bisemigroups and order semigroups, and between semigroup transforms and order transforms. We do this by synthesising an order from a semigroup operation, or vice versa. For the former, we use the *natural order*. Each semigroup has two such associated orders, resulting from interpreting the semigroup operation as a greatest lower bound or least upper bound, and deducing from this a partial order [5]. Given a semigroup $(S, \oplus_S)$, these orders are given by

$$\begin{aligned} s_1 \lesssim_S^{\mathrm{L}} s_2 &\iff s_1 = s_1 \oplus_S s_2 \\ s_2 \lesssim_S^{\mathrm{R}} s_2 &\iff s_2 = s_1 \oplus_S s_2. \end{aligned}$$

For idempotent and commutative semigroups, it is straightforward to deduce that these are always partial orders, and that they are dual to one another. Using other kinds of semigroup may not result in orders with such desirable properties.

We can use the natural order concept to define maps between the quadrants: from bisemigroups to order semigroups, and from semigroup transforms to order transforms. In each case, the mapping acts on the 'weight summarization' part of the structure, deducing explicit preferences from the semigroup summarizing operator:

$$\begin{aligned} \mathrm{NO}^{\mathrm{L}}(S, \oplus_S, \otimes_S) &:= (S, \lesssim_S^{\mathrm{L}}, \otimes_S) \\ \mathrm{NO}^{\mathrm{R}}(S, \oplus_S, \otimes_S) &:= (S, \lesssim_S^{\mathrm{R}}, \otimes_S) \\ \mathrm{NO}^{\mathrm{L}}(S, \oplus_S, F_S) &:= (S, \lesssim_S^{\mathrm{L}}, F_S) \\ \mathrm{NO}^{\mathrm{R}}(S, \oplus_S, F_S) &:= (S, \lesssim_S^{\mathrm{R}}, F_S). \end{aligned}$$

The reverse direction, synthesising a semigroup operation from an order, is more subtle. We might choose to define the operation as taking the least upper bound, or the greatest lower bound, with respect to the order; but unfortunately not all orders possess such bounds. So instead we define it over sets of elements:

$$\begin{aligned} \oplus \colon 2^S \times 2^S &\to 2^S \\ A \oplus B &:= \min\nolimits_{\lesssim}(A \cup B). \end{aligned}$$

There are several ways of extending this order-to-semigroup map to one involving our quadrants model. For example, we can turn order transforms into semigroup transforms via the map

$$(S, \lesssim_S, F_S) \mapsto (S', \oplus, \{f' \mid f \in F_S\})$$

where $S' = \{A \in 2^S \mid \min_{\lesssim}(A) = A\}$ and $f'(A) = \min_{\lesssim}(\{f(a) \mid a \in A\})$.

We can combine this with the Cayley map, to turn order semigroups into semigroup transforms. This capability is important, because it means we can take results about semigroup transforms and apply them to order semigroups—in particular, we know from [11] that for finding global optima with semigroup transforms, we need all of the functions to be homomorphisms. This is more general than the results in [22].

IV. Exact property rules

Our main results in this section are based on a theorem by Saitô [20]. His theorem is for order semigroups, in the special case when the order is total. We will extend this to all four quadrants. In each case, the theorem has substantially the same structure, but some of the properties needed are different.

Write $\mathrm{M}(S)$, $\mathrm{N}(S)$ and $\mathrm{C}(S)$ to represent the following properties or an order semigroup $(S, \lesssim_S, \otimes_S)$:

$$\begin{aligned} \mathrm{M}(S) &\iff \forall a, b, c \in S : a \lesssim_S b \implies c \otimes a \lesssim_S c \otimes b \\ \mathrm{N}(S) &\iff \forall a, b, c \in S : c \otimes_S a = c \otimes_S b \implies a = b \\ \mathrm{C}(S) &\iff \forall a, b, c \in S : c \otimes_S a = c \otimes_S b. \end{aligned}$$

These are the 'left' versions of the assertions that S is respectively monotonic, *cancellative*, or *condensed*; the 'right' versions have the operands of $\otimes_S$ reversed. We will see that properties resembling these are relevant for our other structures.

Theorem 1 (Saitô): Let S and T be order semigroups whose orders are total. Then

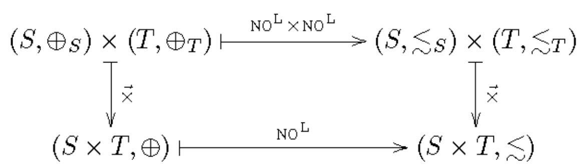
$$\mathrm{M}(S \vec{\times} T) \iff \mathrm{M}(S) \wedge \mathrm{M}(T) \wedge (\mathrm{N}(S) \vee \mathrm{C}(T)).$$

Note that we have one property applying to the first operand, and as an alternative, another property applying to the second. So monotonic algebras can be built that contain arbitrary components, as long as there is another component that does have the appropriate property.

We would like to extend this theorem to cover not only all order semigroups, but each of the other structures in our quadrants model. This will mean that we will be able to construct lexicographic products and prove results about their monotonicity in any model, not just one. First, we will need to define a lexicographic product for semigroups, which will be extended to provide our definition for bisemigroups and semigroup transforms.

A. Lexicographic products of semigroups

Although the lexicographic product may be a well-known operation on ordered sets, far less attention has been paid to how it might be defined over semigroups. As a guide to making our definition in a way that is analogous to the ordered set version, we will use the natural orders. That is, if we have two monoids, take their natural orders, and form the lexicographic product of those orders, we should get the same order as if we had taken the natural order of the lexicographic product of the two monoids. As a diagram, we want



to commute. We will investigate the problem keeping the left natural order in mind, but our definition will turn out to work with the right version as well. We will carry on using the '$\vec{\times}$' symbol to refer to lexicographic products, in all cases.

Let $(S, \oplus_S)$ and $(T, \oplus_T)$ be commutative and idempotent semigroups, with T having an identity. The requirements of commutativity and idempotence are not too onerous: they hold for all of our 'weight summarization' examples. To see why these conditions might be reasonable, consider how the semigroup operation would be used in an algorithm: to combine route information from multiple sources into a single weight. Our conditions enforce that the combining operation must yield the same result regardless of the order in which its operands are given, and it must not matter if any of them are repeated. Another way of thinking about this is that it means that the expression $\bigoplus_{s \in A} s$ is well-defined for a set A of weights.

Now, the carrier set of $S \vec{\times} T$ must be the product $S \times T$; we need to define the monoid operation $\oplus$. If $s_1 \oplus_S s_2 = s_1 \neq s_2$, then we certainly want $(s_1, t_1) \oplus (s_2, t_2) = (s_1, t_1)$, selecting the first operand. Likewise, if $s_1 \oplus_S s_2 = s_2 \neq s_1$ then (s_2, t_2) ought to be chosen instead. Suppose that $s_1 \oplus_S s_2 = s_1 = s_2$: in terms of the left natural order, this means that s_1 and s_2 are equivalent, and we should rely on the T component to make the choice. So we can set

$$(s_1, t_1) \oplus (s_2, t_2) = (s_1, t_1 \oplus_T t_2)$$

in this case. The final case to consider is that the combination of s_1 and s_2 might be neither s_1 nor s_2, but some third element of S. What should appear in the T component of the result? None of t_1, t_2, or $t_1 \oplus_T t_2$ yields the desired natural order. In fact, there is a fourth alternative: the identity element of T. We now have

$$\begin{aligned}&(s_1, t_1) \oplus (s_2, t_2)\\ &\qquad = \begin{cases} (s_1, t_1 \oplus_T t_2) & s_1 = s_2 \\ (s_1, t_1) & s_1 = s_1 \oplus_S s_2 \neq s_2 \\ (s_2, t_2) & s_1 \neq s_1 \oplus_S s_2 = s_2 \\ (s_1 \oplus_S s_2, \alpha_T) & \text{otherwise.} \end{cases}\end{aligned}$$

This can be summarised as

$$(s_1, t_1) \oplus (s_2, t_2) := (s, [s = s_1] t_1 \oplus_T [s = s_2] t_2)$$

where $s = s_1 \oplus_S s_2$, and the notation $[P]x$ for a predicate P is defined as

$$[P]x := \begin{cases} x & P \text{ is true} \\ \alpha_T & P \text{ is false.} \end{cases}$$

If T is not a monoid, we can still use the case-by-case definition, provided that the final case does not occur. This happens when S is *selective*, meaning that $s_1 \oplus_S s_2$ is always equal to s_1 or s_2. So $S \vec{\times} T$ will be defined whenever S is selective, or T is a monoid.

The lexicographic product operator $\oplus$ is associative, commutative, and idempotent (assuming this is true of S and T). This helps in building n-ary lexicographic products, as well as for general utility.

Furthermore, the lexicographic product operator $\vec{\times}$ is itself associative. This is an important property, since it makes the construction of n-ary products much simpler. We do need to take account of whether the operands have identities or are selective, in order for the product to be defined.

Theorem 2: Let S_1 through S_n be commutative, idempotent semigroups. Let $1 \leq k \leq n$. If the semigroups S_1 through S_{k-1} are selective, and the semigroups S_{k+1} through S_n are monoids, then the lexicographic product $S_1 \vec{\times} S_2 \vec{\times} \cdots \vec{\times} S_n$ is defined, and is commutative and idempotent.

Note that in the above result we are allowed to have one semigroup, S_k, that is not selective and need not be a monoid. All those preceding it must be selective, and all those following must be monoids.

We now show that the defined product does indeed commute with the natural order, so it is the product that we were seeking.

Theorem 3: Let S and T be commutative and idempotent semigroups, with T being a monoid. Then $\text{NO}^{\text{L}}(S \vec{\times} T) = \text{NO}^{\text{L}}(S) \vec{\times} \text{NO}^{\text{L}}(T)$ and $\text{NO}^{\text{R}}(S \vec{\times} T) = \text{NO}^{\text{R}}(S) \vec{\times} \text{NO}^{\text{R}}(T)$

Proof: The left natural order on $S \vec{\times} T$ is given by

$$\begin{aligned} & (s_1, t_1) \leq (s_2, t_2) \\ \iff & (s_1, t_1) = (s_1, t_1) \oplus (s_2, t_2) \\ \iff & (s_1 = s_1 \oplus_S s_2) \\ & \quad \wedge (t_1 = [s_1 = s_1 \oplus_S s_2] t_1 \oplus_T [s_2 = s_1 \oplus_S s_2] t_2) \\ \iff & (s_1 = s_1 \oplus_S s_2 \neq s_2) \\ & \quad \vee (s_1 = s_2 \wedge t_1 = t_1 \oplus_T t_2). \end{aligned}$$

The last step was reached by splitting the cases $s_2 = s_1 \oplus_S s_2$ and $s_2 \neq s_1 \oplus_S s_2$. So we have

$$(s_1, t_1) \leq (s_2, t_2)$$
$$\iff (s_1 <_S s_2) \vee (s_1 = s_2 \wedge t_1 \leq_T t_2)$$

as required. The proof for NO^{R} is analogous. ■

We now know how to define a lexicographic product for our basic 'weight summarization' structures: ordered sets and semigroups. These definitions can easily be extended to each of the four quadrants, as follows:

$$\begin{array}{lcl}
(S, \oplus_S, \otimes_S) \vec{\times} (T, \oplus_T, \otimes_T) & := & (S \times T, \oplus, \otimes) \\
(S, \oplus_S, F_S) \vec{\times} (T, \oplus_T, F_T) & := & (S \times T, \oplus, F) \\
(S, \lesssim_S, \otimes_S) \vec{\times} (T, \lesssim_T, \otimes_T) & := & (S \times T, \lesssim, \otimes) \\
(S, \lesssim_S, F_S) \vec{\times} (T, \lesssim_T, F_T) & := & (S \times T, \lesssim, F)
\end{array}$$

where

- $\oplus$ is the lexicographic product of $\oplus_S$ and $\otimes_S$,
- $\lesssim$ is the lexicographic product of $\lesssim_S$ and $\lesssim_T$,
- $(s_1, t_1) \otimes (s_2, t_2) = (s_1 \otimes_S s_2, t_1 \otimes_T t_2)$, and
- F is the set $\{(f_S, f_T) \mid f_S \in F_S, f_T \in F_T\}$ with $(f_S, f_T)(s, t) = (f_S(s), f_T(t))$.

B. Global optima

Now, the general theorem is:

Theorem 4: If S and T come from one of the four quadrants, then

$$\text{M}(S \vec{\times} T) \iff \text{M}(S) \wedge \text{M}(T) \wedge (\text{N}(S) \vee \text{C}(T))$$

where the properties M, N and C are as in Figure 2.

For each structure, there is a different family of properties, but these are parallel to one another. They are not just 'moral equivalents', but formal equivalents, because they relate to the standard translations between the four quadrants.

The combination of M and N yields 'strict monotonicity', the requirement that if $a < b$ then $c \otimes a < c \otimes b$. It has previously been shown [12] that if $\text{SM}(S)$ and $\text{M}(T)$, then $\text{M}(S \vec{\times} T)$, where SM is strict monotonicity. In contrast, our theorem gives necessary and sufficient conditions, and so also addresses the case when S is not strictly monotonic, but property C holds for T.

We can combine the left and right versions of this theorem (if they are different) to obtain a result about two-sided monotonicity. This demonstrates the wide variety of possibilities for constructing monotonic structures, and the different patterns of requirements that might be placed on the operands.

Corollary 1: If S and T are both bisemigroups or both order semigroups, then $S \vec{\times} T$ is left- and right-monotonic if and only if both S and T are left- and right-monotonic, and at least one of the following is true:

- $\text{N}^{\text{L}}(S) \wedge \text{N}^{\text{R}}(S)$,
- $\text{N}^{\text{L}}(S) \wedge \text{C}^{\text{R}}(T)$,
- $\text{N}^{\text{R}}(S) \wedge \text{C}^{\text{L}}(T)$, or
- $\text{C}^{\text{L}}(T) \wedge \text{C}^{\text{R}}(T)$.

Here, N^{L} refers to the left version of N for the appropriate structure, N^{R} refers to the right version, and so on.

The same corollary in the case of order semigroups with a total order was also proved by Saitô.

We will first show that our generalization of Saitô's result to all order semigroups is valid. Because we are working with preorders rather than total orders, we need to take account of some new possibilities for the ordering of elements: they may be equivalent, although not equal, or they may be incomparable. So the new C condition requires only that $c \otimes a$ always be equivalent to $c \otimes b$, rather than that they should be equal, as required by the original C. Similarly, the N property has changed to allow cancelling when $c \otimes a$ is equivalent to $c \otimes b$: then it must be that a is equivalent to b, *or* that they are incomparable. The effect of this is to rule out the possibility that $c \otimes a \sim c \otimes b$, but a and b are strictly ordered, a state of affairs which would result in monotonicity being violated for the lexicographic product. Suppose that $s_1 <_S s_2$ and $s_3 \otimes_S s_1 \sim_S s_3 \otimes_S s_2$. Then $(s_1, t_1) < (s_2, t_2)$ lexicographically, but in order to have $(s_3 \otimes_S s_1, t_3 \otimes_T t_1) \lesssim (s_3 \otimes_S s_2, t_3 \otimes_T t_2)$ we need $t_3 \otimes_T t_1 \lesssim_T t_3 \otimes_T t_2$. This will not be true for all t_1, t_2 and t_3 (unless $\text{C}(T)$ holds), so monotonicity fails.

We will carry out the proof for the left-monotonicity case. In the following, let $(S, \lesssim_S, \otimes_S)$ and $(T, \lesssim_T, \otimes_T)$ be order semigroups.

Lemma 1: If $\text{M}(S \vec{\times} T)$ then $\text{M}(S)$, $\text{M}(T)$, and either $\text{N}(S)$ or $\text{C}(T)$.

Proof: Suppose that $\text{M}(S \vec{\times} T)$. Then for all (s_1, t_1), (s_2, t_2) and (s_3, t_3) in $S \times T$ we have

$$(s_1, t_1) \lesssim (s_2, t_2) \implies (s_3 \otimes_S s_1, t_3 \otimes_T t_1) \lesssim (s_3 \otimes_T s_2, t_3 \otimes_T t_2).$$

If we set $t_1 = t_2$, and choose any t_3, we obtain

$$s_1 \lesssim_S s_2 \iff (s_1, t_1) \lesssim (s_2, t_2)$$

and

$$s_3 \otimes_S s_1 \lesssim_S s_3 \otimes_S s_2 \iff (s_3 \otimes_S s_1, t_3 \otimes_T t_1) \lesssim (s_3 \otimes_T s_2, t_3 \otimes_T t_2)$$

and so $\text{M}(S)$ is true.

Likewise, set $s_1 = s_2$, and choose any s_3 to deduce $\text{M}(T)$.

Now, suppose that $\text{N}(S)$ is false. Then there exist s_1, s_2 and s_3 such that

$$s_1 <_S s_2 \text{ and } s_3 \otimes_S s_1 \sim_S s_3 \otimes_S s_2.$$

For any t_1 and t_2, we have $(s_1, t_1) \lesssim (s_2, t_2)$. Then by monotonicity of $S \vec{\times} T$, for any t_3, $t_3 \otimes_T t_1 \lesssim_T t_3 \otimes_T t_2$. Hence $t_3 \otimes_T t_1 \sim_T t_3 \otimes_T t_2$ for all t_1, t_2 and t_3, and $\text{C}(T)$.

Theorem: $\text{M}(S \vec{\times} T) \iff \text{M}(S) \wedge \text{M}(T) \wedge (\text{N}(S) \vee \text{C}(T))$

Structure	M	N	C
Bisemigroups (left)	$c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$	$c \otimes a = c \otimes b \implies a = b$	$c \otimes a = c \otimes b$
Bisemigroups (right)	$(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$	$a \otimes c = b \otimes c \implies a = b$	$a \otimes c = b \otimes c$
Semigroup transforms	$f(a \oplus b) = f(a) \oplus f(b)$	$f(a) = f(b) \implies a = b$	$f(a) = f(b)$
Order semigroups (left)	$a \lesssim b \implies (c \otimes a) \lesssim (c \otimes b)$	$c \otimes a \sim c \otimes b \implies a \sim b \vee a \# b$	$c \otimes a \sim c \otimes b$
Order semigroups (right)	$a \lesssim b \implies (a \otimes c) \lesssim (b \otimes c)$	$a \otimes c \sim b \otimes c \implies a \sim b \vee a \# b$	$a \otimes c \sim b \otimes c$
Order transforms	$a \lesssim b \implies f(a) \lesssim f(b)$	$f(a) \sim f(b) \implies a \sim b \vee a \# b$	$f(a) \sim f(b)$

Fig. 2. Properties required for global optima. Each expression is universally quantified over its free variables.

Finally, suppose that $\text{C}(T)$ is false. Then there exist t_1, t_2 and t_3 such that $t_3 \otimes_T t_1$ and $t_3 \otimes_T t_2$ are strictly ordered or incomparable. In any case, we can potentially swap t_1 and t_2 so that

$$\neg(t_3 \otimes_T t_1 \lesssim_T t_3 \otimes_T t_2).$$

From monotonicity of $S \vec{\times} T$ we have

$$\begin{aligned} & s_1 <_S s_2 \vee (s_1 \sim_S s_2 \wedge t_1 \lesssim_T t_2) \\ \implies & s_3 \otimes_S s_1 <_S s_3 \otimes_S s_2 \vee (\text{FALSE}) \end{aligned}$$

and hence $s_1 <_S s_2$ implies $s_3 \otimes_S s_1 <_S s_3 \otimes_S s_2$. Now, if $s_3 \otimes_S s_1 \sim_S s_3 \otimes_S s_2$ then $\neg(s_3 \otimes_S s_1 <_S s_3 \otimes_S s_2)$ and $\neg(s_3 \otimes_S s_2 <_S s_3 \otimes_S s_1)$, which implies that $\neg(s_1 <_S s_2)$ and $\neg(s_2 <_S s_1)$. Therefore either $s_1 \sim_S s_2$ or $s_1 \#_S s_2$, so we have $\text{N}(S)$. ■

Lemma 2: If $\text{M}(S)$, $\text{M}(T)$, and either $\text{N}(S)$ or $\text{C}(T)$, then $\text{M}(S \vec{\times} T)$.

Proof: Suppose that $(s_1, t_1) \lesssim_S (s_2, t_2)$. Then either $s_1 <_S s_2$, or $s_1 \sim_S s_2$ and $t_1 \lesssim_T t_2$. Hence $s_1 \lesssim_S s_2$, and by monotonicity of S we have $s_3 \otimes_S s_1 \lesssim_S s_3 \otimes_S s_2$. If $s_3 \otimes_S s_1 <_S s_3 \otimes_S s_2$ then there is nothing more to prove, so consider the case when $s_3 \otimes_S s_2 \sim_S s_3 \otimes_S s_1$.

If $\text{C}(T)$ is true, then we automatically now have monotonicity of $S \vec{\times} T$, since $(s_3 \otimes_S s_1, t_3 \otimes_T t_1) \lesssim (s_3 \otimes_S s_2, t_3 \otimes_T t_2)$.

If, however, $\text{N}(S)$ is true, then it follows that $s_1 \sim_S s_2$ (or $s_1 \#_S s_2$, but this cannot be the case because we already know that $s_1 \lesssim_S s_2$). Hence $t_1 \lesssim_T t_2$, and since T is monotonic we have $t_3 \otimes_T t_1 \lesssim_T t_3 \otimes_T t_2$. Then $(s_3 \otimes_S s_1, t_3 \otimes_T t_1) \lesssim (s_3 \otimes_S s_2, t_3 \otimes_T t_2)$, and so in this case $S \vec{\times} T$ is also monotonic. ■

C. Local optima

For finding local optima, the properties of interest are 'increasing' and 'nondecreasing'. As with monotonicity, these look slightly different in each of the four quadrants, while still expressing the same general idea: that extensions of paths are (strictly) less preferred than the originals.

In the case of the lexicographic product, our general theorems are as follows.

Theorem 5: If S and T come from one of the four quadrants, then

$$\begin{aligned} \text{ND}(S \vec{\times} T) &\iff \text{I}(S) \vee (\text{ND}(S) \wedge \text{ND}(T)) \\ \text{I}(S \vec{\times} T) &\iff \text{I}(S) \vee (\text{ND}(S) \wedge \text{I}(T)) \end{aligned}$$

where the properties ND and I are as in Figure 3

Note that we always have $\text{I}(S) \implies \text{ND}(S)$.

It is then straightforward to prove when an n-ary lexicographic product is increasing.

Corollary 2: Let S_1 through S_n be from one of the four quadrants. Then $\text{I}(S_1 \vec{\times} \cdots \vec{\times} S_n)$ if and only if there is some k with $1 \leq k \leq n$ such that $\text{I}(S_k)$ and for $j < k$, $\text{ND}(S_j)$.

So a lexicographic product that is increasing has three parts. First, there are zero or more components that are merely nondecreasing. Then, a component that is increasing; and finally, the remaining components need not have any special properties at all. This means that in our lexicographic products, we can use increasing algebras to 'guard' any lower-priority algebras at all, and still be able to compute local optima using these metrics.

We will prove Theorem 5 in the case of semigroup transforms; the other proofs are essentially the same.
Proof: Let $(S, \oplus_S, F_S)$ and $(T, \oplus_T, F_T)$ be semigroup transforms. Now, $S \vec{\times} T$ is nondecreasing if and only if

$$(s, t) = (s, t) \oplus (f(s), g(t))$$

for all s in S, t in T, f in F_S and g in F_T; that is,

$$\begin{aligned} (s, t) = (s \oplus_S f(s), & \\ & [s = s \oplus_S f(s)]t \oplus_T [f(s) = s \oplus_S f(s)]g(t)). \end{aligned}$$

If $\text{I}(S)$ then $s = s \oplus_S f(s) \neq f(s)$, and we have equality without needing to consider T at all. If $\text{ND}(S)$ but not $\text{I}(S)$ then we additionally need $\text{ND}(T)$ so that $t = t \oplus_T g(t)$, for the case when $f(s) = s \oplus_S f(s)$. For $S \vec{\times} T$ to be increasing, we need the same as before except that (s, t) should not equal $(f(s), g(t))$. If $\text{I}(S)$ then we are done, but if we only have $\text{ND}(S)$ we need $\text{I}(T)$ to make t different from $g(t)$. ■

V. Revisiting Policy Partitions

Given the results of the previous section we can now revisit the policy partitioning constructions of Section II, and show

Theorem: $\text{ND}(S \vec{\times} T) \iff \text{I}(S) \vee (\text{ND}(S) \wedge \text{ND}(T))$ and $\text{I}(S \vec{\times} T) \iff \text{I}(S) \vee (\text{ND}(S) \wedge \text{I}(T))$

Structure	ND	I
Bisemigroups (left)	$a = a \oplus (c \otimes a)$	$a = a \oplus (c \otimes a) \neq (c \otimes a)$
Bisemigroups (right)	$a = a \oplus (a \otimes c)$	$a = a \oplus (a \otimes c) \neq (a \otimes c)$
Semigroup transforms	$a = a \oplus f(a)$	$a = a \oplus f(a) \neq f(a)$
Order semigroups (left)	$a \lesssim c \otimes a$	$a \neq \top \implies a < c \otimes a$
Order semigroups (right)	$a \lesssim a \otimes c$	$a \neq \top \implies a < a \otimes c$
Order transforms	$a \lesssim f(a)$	$a \neq \top \implies a < f(a)$

Fig. 3. Properties relating to local optima. Each expression is universally quantified over its free variables.

exactly when they are or are not monotonic, increasing, or nondecreasing.

Theorem 6: If S and T are order transforms, with S having two or more elements and T having two or more equivalence classes, then

$$\begin{aligned} \text{ND}(S \ominus T) &\iff \text{I}(S) \wedge \text{ND}(T), \\ \text{I}(S \ominus T) &\iff \text{I}(S) \wedge \text{I}(T), \\ \text{M}(S \ominus T) &\iff \text{M}(S) \wedge \text{M}(T). \end{aligned}$$

We first remark that it is easy to see that for any order transform S, the following properties hold:

$$\text{ND}(\mathsf{right}(S)),\ \text{M}(\mathsf{left}(S)),\ \text{M}(\mathsf{right}(S)).$$

Unless S has only one element, we also have

$$\neg\text{I}(\mathsf{left}(S)) \text{ and } \neg\text{I}(\mathsf{right}(S)),$$

and unless S consists of a single equivalence class, $\neg\text{ND}(\mathsf{left}(S))$. In addition, for any property P in $\{\text{ND}, \text{I}, \text{M}\}$ it is easy to check that $\text{P}(S) \wedge \text{P}(T) \iff \text{P}(S + T)$.

Theorem 6 then follows easily from the results of the previous section. For example, for the first claim of the theorem we have

$$\begin{aligned} &\text{ND}(S \ominus T) \\ \iff &\text{ND}(S \vec{\times} \mathsf{left}(T)) \wedge \text{ND}(\mathsf{right}(S) \vec{\times} T) \\ \iff &(\text{I}(S) \vee (\text{ND}(S) \wedge \text{ND}(\mathsf{left}(T)))) \\ &\wedge (\text{I}(\mathsf{right}(S)) \vee (\text{ND}(\mathsf{right}(S)) \wedge \text{ND}(T))) \\ \iff &(\text{I}(S) \vee (\text{ND}(S) \wedge \text{FALSE})) \\ &\wedge (\text{FALSE} \vee (\text{TRUE} \wedge \text{ND}(T))) \\ \iff &\text{I}(S) \wedge \text{ND}(T). \end{aligned}$$

What is remarkable about Theorem 6 is that for monotonicity, we only need the two operands to be monotonic. Compare this situation with Theorem 4, where either S or T needs an additional algebraic property. We can see that the use of the scoped product removes this requirement, so there are many more order transforms that can be employed in these products than in lexicographic products.

Recall the bandwidth-delay example: both metrics are monotonic, but the lexicographic product was not, because the property N failed for bandwidths. But a scoped product with these operands *is* monotonic. This works because in the scoped product, changes to the first component always replace the second component at the same time: we never experience the inversion of preferences resulting from the second component becoming significant although it previously was not. So for this example, we can find global optima providing we use a scoped product rather than a simple lexicographic product; and we can always find local optima, since we have ND for bandwidths and I for delays.

In a similar way, we can establish exact conditions for the OSPF-like operator Δ.

Theorem 7: If S and T are order transforms as in Theorem 6, then

$$\begin{aligned} \text{ND}(S \,\Delta\, T) &\iff \text{I}(S) \wedge \text{ND}(T) \\ \text{I}(S \,\Delta\, T) &\iff \text{I}(S) \wedge \text{I}(T) \\ \text{M}(S \,\Delta\, T) &\iff \text{M}(S) \wedge \text{M}(T) \wedge (\text{N}(S) \vee \text{C}(T)) \end{aligned}$$

This variation on the scoped product is therefore more demanding in terms of the restrictions it places on the operands. This can be explained by the observation that the BGP-like operator restricts the *use* of the algebra, by enforcing a strict separation of external and internal functions; but Δ can be used just like an ordinary lexicographic product in addition to its internal-only mode.

VI. Discussion

An alternative to using the lexicographic order would be to combine metrics into a single value according to some fixed formula, as is done for example in EIGRP [4]. For some applications, this may be a better design[2]. Gouda and Schneider [12] have investigated 'additive composite metrics' as a step towards understanding such techniques, and have proved that if both S and T are nondecreasing, then so is the additive combination of S and T. We would like to find similar criteria for other properties, in more complex situations, and with both necessary and sufficient conditions.

Lexicographic decision making has been studied extensively by economists, following Debreu's 1954 discovery that lexi-

[2] In fact, EIGRP can be perceived as using a lexicographic order in addition to its normal metric, since it discards routes with hop count greater than a predefined maximum before considering the 'real' metric.

cographic preferences are not always representable by utility functions [7]. In the economics literature, attention has been given to the question of whether lexicographic preference is a good model of how real decisions are made (for example [6]) which is somewhat relevant to our routing topic. Although several important protocols do make use of such orderings, it is not so obvious whether the traffic engineering intentions of network operators can be fully realized by a strict hierarchy of metrics. A thorough review of the early treatment of lexicographic orders in economics and behavioral science was given by Fishburn [8].

There is a construction due to Szendrei [24] that resembles our lexicographic product of semigroups. The difference comes in the handling of absorbing elements. If S is a commutative idempotent semigroup with absorbing element $\omega_{\oplus_S}$, and T is a commutative idempotent monoid, define $S \vec{\times}_\omega T$ to be the semigroup $((S \setminus \{\omega_{\oplus_S}\} \times T) \cup \{\omega\}, \oplus)$, where $\omega \oplus (s,t) = (s,t) \oplus \omega = \omega$, and $(s_1,t_1) \oplus (s_2,t_2)$ is ω if $s_1 \oplus s_2 = \omega_{\oplus_S}$, and equal to our definition otherwise. So ω is the new absorbing element, and if $s_1 \oplus s_2$ would have yielded the absorber for S, the whole pair is replaced by ω.

The usefulness of this definition is that if we are dealing with finite algebraic structures, some of our properties will necessarily not be true. For example, the N property for semigroup transforms asserts that if $f(a) = f(b)$, then $a = b$. However, in

$$(\{0,\ldots,n\}, \min, \{\lambda x. \min(n, x+y) \mid y \in \{0,\ldots,n\}\})$$

it is quite possible to have $f(a) = f(b) = n$ but $a \neq b$. But this algebra is usable as the first component of a lexicographic product with $\vec{\times}_\omega$, since if n ever arises the entire expression will be reduced to ω.

In this paper, we have not explored the relationship between $\vec{\times}$ and $\vec{\times}_\omega$. It is evident, at least, that care needs to be taken in the definition of possible mixed-mode n-ary lexicographic products, where some of the products use $\vec{\times}$ and some use $\vec{\times}_\omega$. A proper handling of this issue will take into account the necessary distinction between a weight that is least-preferred, and one which represents an error, such as an out-of-range value.

Wongseelashote [25] defined a *reduction* on a semigroup $(V, \circ)$ to be a function $r : 2^V \longrightarrow 2^V$ satisfying

1) $r(\emptyset) = \emptyset$,
2) for all A and B in 2^V, $r(A \cup B) = r(r(A) \cup B)$, and
3) for all A and B in 2^V, $r(A \circ B) = r(r(A) \circ B) = r(A \circ r(B))$.

where $A \circ B = \{a \circ b \mid a \in A, b \in B\}$. For example, min is a reduction on $(\mathbb{N}, +)$: this is what we have been calling 'min-set-map'. We would like to integrate the idea into our framework, because we use this example but do not have a solid account of how it relates to other, similar mappings. The algebraic structure of the set of reductions on a given semigroup is not well understood, so there is a good deal of work to be done in order to give a satisfactory account of min-like operations. We hope that problems like finding k-best paths can be tackled using the reduction idea.

Acknowledgements

We thank the ICNP reviewers, John Billings, Ken Calvert, and Marcelo Fiore for helpful comments on drafts of this paper. A. Gurney is supported by a Doctoral Training Account from the Engineering and Physical Sciences Research Council (EPSRC). T. Griffin is grateful for support under the the Cisco Collaborative Research Initiative.

References

[1] G. Birkhoff. *Lattice Theory, 3rd edition*. Amer. Math. Soc., 1967.
[2] M.-O. Buob, M. Meulle, and J.-L. Lutton. Un modèle de graphe et de dioïde pour le routage interdomaine. In *Colloque Francophone sur l'Ingénierie des Protocoles (CFIP)*, 2006.
[3] B. Carré. *Graphs and Networks*. Oxford University Press, 1979.
[4] Cisco Systems. *Enhanced Interior Gateway Routing Protocol*, 2005. Technology White Paper: document ID 16406.
[5] A. H. Clifford and G. B. Preston. *The algebraic theory of semigroups*, volume 1. American Mathematical Society, Providence, RI, 1961.
[6] A. M. Colman and J. A. Stirk. Singleton bias and lexicographic preferences among equally valued alternatives. *Journal of Economic Behaviour and Organization*, 40:337–351, 1999.
[7] G. Debreu. Representation of a preference ordering by a numerical function. In R. M. Thrall, C. H. Coombs, and R. L. Davis, editors, *Decision Processes*. Wiley, 1954.
[8] P. C. Fishburn. Lexicographic orders, utilities and decision rules: a survey. *Management Science*, 20(11):1442–1471, July 1974. Theory Series.
[9] L. Fuchs. *Partially Ordered Algebraic Systems*. Addison-Wesley, 1963.
[10] M. Gondran and M. Minoux. *Graphs and Algorithms*. Wiley, 1984.
[11] M. Gondran and M. Minoux. *Graphes, dioïdes et semi-anneaux: Nouveaux modéles et algorithmes*. Tec & Doc, 2001.
[12] M. G. Gouda and M. Schneider. Maximizable routing metrics. *IEEE/ACM Transactions on Networking*, 11(4):663–675, August 2003.
[13] T. G. Griffin and A. Gurney. The quadrants model of algebraic routing. Under review, 2007.
[14] T. G. Griffin and J. L. Sobrinho. Metarouting. In *Proc. ACM SIGCOMM*, August 2005.
[15] R. E. Johnson. Free products of ordered semigroups. *Proceedings of the American Mathematical Society*, 19(3):697–700, 1968.
[16] K.Varadhan, R.Govindan, and D. Estrin. Persistent route oscillations in inter-domain routing. *Computer Networks*, 32:1–16, 2000.
[17] T. Lengauer and D. Theune. Efficient algorithms for path problems with general cost criteria. *Lecture Notes in Computer Science*, 510:314–326, 1991.
[18] T. Lengauer and D. Theune. Unstructured path problems and the making of semirings. *Lecture Notes in Computer Science*, 519:189–200, 1991.
[19] Y. Rekhter and T. Li. A Border Gateway Protocol. RFC 1771 (BGP version 4), March 1995.
[20] T. Saitô. Note on the lexicographic product of ordered semigroups. *Proceedings of the Japan Academy*, 46(5):413–416, 1970.
[21] J. L. Sobrinho. Algebra and algorithms for QoS path computation and hop-by-hop. *IEEE/ACM Transactions on Networking*, 10(4):541–550, August 2002.
[22] J. L. Sobrinho. Network routing with path vector protocols: Theory and applications. In *Proc. ACM SIGCOMM*, September 2003.
[23] J. L. Sobrinho. An algebraic theory of dynamic network routing. *IEEE/ACM Transactions on Networking*, 13(5):1160–1173, October 2005.
[24] M. B. Szendrei. On the structure of orthodox semigroups. *Journal of Algebra*, 56(1):71–90, 1979.
[25] A. Wongseelashote. Semirings and path spaces. *Discrete Mathematics*, 26(1):55–78, 1979.