Pathfinding through Congruences

Alexander J. T. Gurney and Timothy G. Griffin

Computer Laboratory, University of Cambridge, UK

Abstract. Congruences of path algebras are useful in the definition and analysis of pathfinding problems, since properties of an algebra can be related to properties of its quotient. We show that this relationship can apply even when the algebraic objects involved satisfy weaker forms of the semiring or path algebra axioms. This is useful, since it is just these algebras and their quotients which we need to analyze pathfinding problems characterized by the need to obtain multiple paths even when path preferences are inconsistent, and paths can be filtered out arbitrarily, as in Internet routing.

1 Introduction

For finding optimal paths in graphs and networks, there is a standard theory grounded in linear algebra [2], [3], [4], [11], [15].

But for certain kinds of pathfinding, including some which are important for Internet routing, it seems to be difficult to take advantage of this theory. These situations are problematic because the information sought may not be a single path, because the criteria for path quality may not result in the existence of an optimal solution, or because the routing algorithms are implemented in a distributed and asynchronous fashion. All of these are difficult to incorporate into the theoretical model.

Nonetheless, in recent years there has been an effort to bring the algebraic theory up to speed with the strange and diverse nature of Internet routing. The mathematical language is ultimately not too different, in terms of the signatures of the algebraic objects involved: we will always need some way to compare paths, and some way to compose them out of arcs. The difference comes in the axioms and derived properties that these structures might have. Whereas conditions such as distributivity have historically been assumed for rings and semirings, our new structures may lack distributivity but instead be endowed with other helpful properties [5]. Analogous methods can then be used to treat their structure theory, and in particular the way that important properties are derived compositionally. From a theoretician's perspective, this demonstrates that the unusual features of Internet pathfinding are not so unusual after all, since they are amenable to similar correctness analysis as in the familiar case.

This paper is about the use of congruences as a definitional tool for these new routing algebras. Of course, congruences and quotients are part of the standard abstract algebraic apparatus for familiar structures; the theory of varieties

yields profound insights into how equational properties relate to algebraic constructions. The surprise for our structures is that our most important property is in fact inequational, but relates well with quotient constructions even so.

In particular, we apply congruences to practical problems including the finding of multiple paths, in the presence of filtering, all the while in a world where path preferences do not follow the usual semiring model, but instead satisfy alternative stability criteria.

Much of this material derives from the first author's doctoral thesis [8].

2 Internet Pathfinding in the Abstract

We first summarize the algebraic approach to the analysis of Internet routing, and relate it to the particular features of that problem which differ from more conventional pathfinding.

From a theory perspective, the main issue with interdomain routing is that it is not in fact solving a shortest path problem, and so the usual mathematical apparatus cannot be applied [6]. The use of semirings and related structures for finding best paths in a labelled graph is well understood. The general pattern is that a semiring $(S, \oplus, \otimes)$ can be used to encode path preferences: the elements of S label the arcs of the graph; these are composed with the binary operator $\otimes$ to form path weights, and the best paths emerge when alternatives are summarized with the $\oplus$ operator.

Algorithms based on this pattern solve the best-path problem by computing the closure A^* of the adjacency matrix A of the graph: a matrix whose entries come from S and where matrix addition and multiplication are defined in terms of the operators of S. The facts that the (i, j) entry of

$$A^* = I + A + A^2 + A^3 + \cdots \tag{1}$$

contains the weight of the optimal path from node i to node j, and that this matrix can be computed in finite time, depend on algebraic properties of S. In particular, a distributive law is required for the two operators:

$$\forall a, b, c \in S : c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b). \tag{2}$$

Commonly-considered semirings for this purpose include $(\mathbb{N}, \min, +)$ for computing shortest paths, $(\mathbb{N}, \max, \min)$ for computing widest paths, and $(\mathbb{N}, +, \times)$ for counting paths.

The distributive law encodes the idea that the path choice made by a node (between paths a and b) will be compatible with that made by a neighbour (between $c \otimes a$ and $c \otimes b$). If the first node always behaves as the neighbour would want, then it is algorithmically acceptable for it to make that choice greedily. Therefore, the algorithms of Dijkstra or Bellman-Ford really do compute the best paths, while avoiding the need to enumerate all paths.

Unfortunately, this compatibility of preferences does not hold for Internet pathfinding, where nodes may be controlled by entities in commercial conflict.

In such a situation, it may be that when some node chooses path a over b, its neighbour would have preferred it to take b instead, since $c \otimes b$ could be better than $c \otimes a$. Due to the requirements of hop-by-hop forwarding, a node has no option but to endure its neighbours' choices.

Remarkably, optimal paths can still be computed in this setting, in an efficient manner—as long as we change our definition of optimality. We no longer require a path assignment that is a global optimum, but only a Nash equilibrium, meaning a state from which no node has any incentive to change its current choice of path. A state X in Nash equilibrium can be characterised as a fixed point of

$$X \mapsto AX + I. \tag{3}$$

This operation entails taking each (i, j)-path in X, extending them along the arcs represented in A, and then choosing the best (according to the criteria encoded in the algebra S); the addition of the identity matrix I ensures that the empty path from each node to itself will always be present in the solution. So for a fixed point, we have $X = AX + I$, meaning that when the extension and choice is carried out, each path is the same as it was before. In game-theoretic terms, no (i, j) has any incentive to deviate from its assigned path in X, assuming a game where the only choice is among the paths made available by neighbours. Notably, this equation is the same one which characterises global optimality for shortest paths, if S is the shortest-path semiring. In the wider context, it still represents an optimum: but a local optimum rather than a global one.

To compute such an equilibrium, we simply use the same matrix iteration as in the shortest-path case, with the exception that the underlying algebra is *not* a semiring obeying the distributive law. While this iteration is not guaranteed to terminate in the absence of distributivity, there are other correctness conditions that are sufficient, and are also consistent with the nature of Internet routing. One such is the *strict inflationary* property

$$\forall a, c \in S : a = a \oplus (c \otimes a) \neq c \otimes a, \tag{4}$$

combined with a finite support condition. Even if the distributive law does not hold for a given semiring, this law ensures the existence of a unique fixed point, to which the iterative algorithm converges after finitely many steps, from any starting state [8].

The finite support condition mentioned above is essential for the 'finitely many steps' part of the result. Without this, the possibility remains that the iteration could continue forever, converging towards a state that could never actually be reached. In path computation, it is enough to restrict our set of paths to a some finite subset of the set of all paths in the given graph: so whenever a path arises in the dynamic computation that is outside this set, it should be excluded from consideration. A reasonable choice would be the set of all simple paths in the graph. In terms of weights rather than paths, the condition is that there be only finitely many permitted path weights. Exactly how this kind of condition can be achieved is one of the major topics of this paper, and is explored in Sections 4.1 and 4.2. In brief, the idea is to go from a

possibly-infinite semiring-like structure to a finite one, by taking a congruence that identifies all forbidden paths. The convergence theorem can then be stated with the simple precondition that the given algebraic structure be finite.

A further wrinkle is that the semiring multiplication needs to be replaced with function application, if we are to be capable of expressing the diversity of Internet routing configurations. So rather than dealing with semiring-like structures, we are in fact going to use either *order transforms* $(S, \preceq, F)$ or *semigroup transforms* $(S, \oplus, F)$, where S and $\oplus$ are as before; $\preceq$ is a preorder on S; and F is a set of functions from S to S. These respectively generalize ordered semigroups (the semigroup $\otimes$ being replaced by F) and algebras of monoid endomorphisms (except that our functions need not be endomorphisms). In calculations, the functions F are attached to arcs whereas values in S are originated at nodes: path weights are calculated by applying the functions in order to the starting value. The weights can then be compared with $\preceq$ or summarized by $\oplus$, as appropriate. The analogous *strict inflationary* properties here are

$$\forall a \in S, f \in F : \quad a \prec f(a) \tag{5}$$

$$\forall a \in S, f \in F : \quad a = a \oplus f(a) \neq f(a) \tag{6}$$

respectively.

The reason for using these functions is to permit a wide range of possibilities for how path weights can be derived from arc weights. In routing protocols, a multiplicity of attributes are associated with each route: these are calculated in potentially very complex ways, to allow network operators to exercise fine-grained control over the eventual degree of preference each path will receive.

Some options for how functions in F could operate on route data include:

- Adding a numeric arc weight to the path weight.
- Applying 'bottleneck' bandwidth to the bandwidth of a path.
- Adding a node identifier to a list or set.
- Adding a node identifier, *but* also eliminating the path from consideration if that identifier was already present.
- Adding 'community' tags to remotely influence route choice.
- Importing a route from one routing protocol to another, translating route attributes as appropriate.
- Marking routes based on the business relationship between the systems at either end of the arc.

The elements of S may also be *sets* of paths, or other structured collections. In this case, the functions in F apply to the entire set: they can do any of the above operations on a per-path basis, but can also operate on the set as a whole. For example, the set could be reduced to a single best member path, in some way; and that method need not be the same everywhere in the graph.

In structuring the algebraic theory, we have to consider this complexity, and ideally find ways of making it not matter. This involves the development of constructions, that are justified both theoretically and practically, for building algebras from simpler components. With a good choice of constructions, the task

of deciding whether a particular algebra has the required correctness conditions should not be too difficult at any stage. This, we believe, is the case for our congruence-based constructions, which are theoretically pleasant, have a good computational interpretation, and are useful for several problems which arise in the modelling of Internet routing.

3 Congruences

The notions of congruence and of quotient are critical to the structure theory of many abstract algebraic objects, including semigroups and semirings. The general picture is that a congruence is an equivalence relation that is compatible with the operations of the object; this makes it possible to lift those operations to deal with equivalence classes rather than elements, thus forming the quotient algebra [7].

Definition 1. *An equivalence relation $\sim$ on semigroup $(S, \oplus)$ is a* congruence *if*

$$a \sim b \implies (a \oplus c) \sim (b \oplus c) \wedge (c \oplus a) \sim (c \oplus b)$$

Definition 2. *If $(S, \oplus)$ is a semigroup and $\sim$ is a congruence on S, then the quotient $(S/\!\sim, \oplus/\!\sim)$ is also a semigroup. Here, $S/\!\sim$ is the set of $\sim$-classes. If the class of a is denoted by $[a]$ then the operation of the new semigroup is*

$$[a] \oplus/\!\sim [b] = [a \oplus b]$$

The fact that $\sim$ is a congruence makes this operation well-defined and associative.

The point of these congruences is that in many cases, properties of $S/\!\sim$ can be related to properties of S. This is important for understanding Internet routing from an algebraic perspective: it would be convenient if our key correctness properties were preserved under taking congruences, and if congruences turned out to be useful for modelling certain details of Internet pathfinding problems.

Unfortunately, the inequality in our strict inflationary condition means that a quotient algebra is not guaranteed to have that property, even if the starting algebra did. However, there are some important cases where we are able to use congruences to define new algebras with this property being preserved.

We see this most clearly when considering multipath routing; that is, the idea that for each source and destination we want to find as many good paths as possible, as opposed to a single best path. Algebraically, we just need to choose S to contain not path weights, but sets of path weights, and lift the other operations in the obvious way. We prefer to think of this process as a *construction* on S, because that allows us to examine the relationship between the single-path and multipath cases.

There are other ways of dealing with the presence of multiple best paths. One could also use a conventional single-path algorithm, with some rule for discriminating between otherwise equivalent paths. For example, either the oldest or the

newest path seen could be selected; though these methods introduce undesirable nondeterminism into the path selection process, making the correctness much less tractable to analyze algebraically. Alternatively, a partial order on paths could be linearized to a total order: this amounts to the introduction of some deterministic tiebreaking method. But this does not suffice even for conventional shortest-path finding, since we can construct order transforms which have one of the required correctness properties (monotonicity), but where no linearization has this property ([8], Theorem 3.1).

In the end, the most serious criticism of any of these ideas is that for some purposes, we want to receive multiple routes. Trying to force the use of a single-path algorithm would be inappropriate: a case of solving the wrong problem. The failure of these strategies should make use of our construction more attractive, provided that it does have the right algebraic properties. So we now need to understand how to define algebras that make use of this idea, and how these behave in terms of the properties we need for correctness.

In the case of an order transform $(S, \preceq, F)$, we want to derive the *algebra of minimal sets* of S, written $\mathbf{minset}(S)$. The elements will be subsets of S under the condition that everything in a set is either equivalent or incomparable under $\preceq$. We can also define a lifted version of F, and endow this structure with a semilattice join operation (or an equivalent partial order). This amounts to a free distributive lattice construction. In other words, to obtain $\mathbf{minset}(S)$ we

1. form the power set of S, which is a distributive lattice under inclusion,
2. take a quotient of this lattice by a congruence derived from $\preceq$; this yields the required order or binary operator,
3. and define lifted versions of the functions in F.

Later, we will vary the second step to obtain other useful constructions. These first two steps, taken together, result in the formation of the distributive lattice corresponding to upper sets in S, as in the representation theorem of Birkhoff [1].

Theorem 1 (Birkhoff's theorem). *A finite distributive lattice is isomorphic to the lattice of upper sets of the partial order of its meet-prime elements.*

We quote the theorem in this form (using meet-prime rather than join-prime elements, and upper sets rather than lower sets) because it is the most directly applicable to our purpose, given the conventional interpretation of path preference where $a \prec b$ means that a is *more* preferred than b.

If $(S, \preceq)$ is a partial order, then we can form a corresponding free distributive lattice, whose elements are the upper sets of S, and where the order is the subset order. If the partial order, moreover, has no infinite descending chain, then an equivalent construction takes all sets of the form

$$A = \min(A) \tag{7}$$

where

$$\min(A) = \{x \in A \mid \forall y \in A : \neg(y \prec x)\}. \tag{8}$$

The equivalence comes from the fact that this min operation determines the same congruence as the taking of upper sets [8].

We end up with the same distributive lattice (up to isomorphism). The difference is that using min is a more natural representation of sets for our path problems: $\min(A)$ will (for us) always be a finite set, and its elements have an obvious interpretation as the 'best' things in A. As an alternative, use of a well-quasi-order would guarantee that $\min(A)$ was always a finite set, nonempty unless A were empty [9]. This is because well-quasi-orders, in addition to the lack of infinite descending chains, also lack infinite antichains.

The fact that we have a distributive lattice allows us to deduce immediately that certain computationally useful facts are true of min. In particular, we have

$$\min(A) = \min(\min(A)) \tag{9}$$

$$\min(A \cup B) = \min(\min(A) \cup B) \tag{10}$$

for all A and B. These will influence the implementation of algorithms, by allowing applications of min to be elided in some circumstances.

Essentially the same construction can be carried out if $(S, \preceq)$ is only a preorder. We again obtain 'minimal sets' of elements of S, and a join operator

$$A \oplus B = \min(A \cup B). \tag{11}$$

The functions f in F are lifted to

$$f(A) = \min\{f(a) \mid a \in A\}; \tag{12}$$

note the use of min to put the result set into canonical form. This completes the construction of $\mathbf{minset}(S)$ for an order transform S: the resulting structure is suitable for use in path algorithms. Sets of paths are combined, via $\oplus$, by finding the best paths out of either set; the f functions now operate on every path in the given set, and only the best paths are allowed to remain.

This idea of *canonicalization* is central to our understanding of congruence-based constructions. Beginning with min, we can derive an equivalence relation

$$A \sim B \iff \min(A) = \min(B) \tag{13}$$

on subsets of S, and so obtain the appropriate distributive lattice by a quotient of the free lattice. The point is that the min operator is a natural one from the perspective of pathfinding algorithms, but it is not the only choice. In general, whenever we have a way of putting elements of S into a canonical form, we would like to be able to derive a congruence so that a version of the above construction can be applied. This is not always possible, but there are sufficient conditions on the canonicalization function which ensure that the derived equivalence relation is a congruence. In fact they are the same as the properties of min from above.

Definition 3. *If $(S, \oplus)$ is a semigroup and r is a function from S to S such that*

1. $r(a) = r(r(a))$
2. $r(a \oplus b) = r(r(a) \oplus b) = r(a \oplus r(b))$

for all a and b in S, then r is a reduction *[13],[14].*

In the case of a monoid, the first of these axioms is not needed, since we already have $r(a \oplus 1) = r(r(a) \oplus 1)$ from the second axiom, where 1 is the identity for $\oplus$. Similarly, the second axiom can be simplified to a single equality in the case of a commutative semigroup.

A function on a semiring is called a reduction if it is a reduction with respect to both of the semiring operations. Similarly, a reduction on a semigroup transform $(S, \oplus, F)$ is a function r from S to itself, such that r is a reduction on $(S, \oplus)$ and

$$r(f(a)) = r(f(r(a))) \tag{14}$$

for all a in S and f in F. (This replaces the second axiom from Definition 3, for the multiplicative part of the structure.)

A reduction might also be an endomorphism on a semigroup (and similarly, on a semiring), if it additionally satisfies

$$r(a \oplus b) = r(a) \oplus r(b) \tag{15}$$

for all a and b in the carrier set. Furthermore, not every endomorphism of a semigroup will be a reduction, since not all endomorphisms are idempotent.

The min operation with respect to a preorder $(S, \preceq)$ is a reduction on the semigroup $(2^S, \cup)$. Note, however, that it is not a homomorphism. For any function f on S, and any $A \subseteq S$, we also have

$$\min\{f(x) \mid x \in A\} = \min\{f(x) \mid x \in \min(A)\} \tag{16}$$

which demonstrates that min is always a semigroup transform on $(2^S, \cup, F)$, no matter which set of functions F is used.

We now show that a canonicalization or reduction operation defines a congruence, and that conversely every congruence can be used to define a reduction. This also demonstrates that although endomorphisms are not generally reductions, it is always possible to find a reduction that generates the same congruence as a given endomorphism.

Lemma 1. *For any reduction r on $(S, \oplus)$, define a relation $\sim_r$ on S by*

$$a \sim_r b \stackrel{\text{def}}{\iff} r(a) = r(b).$$

This $\sim_r$ is a congruence.

Proof. This is obviously an equivalence relation. To prove that it is a congruence, suppose that $a \sim_r b$, so that $r(a) = r(b)$. Then

$$r(a \oplus c) = r(r(a) \oplus c) = r(r(b) \oplus c) = r(b \oplus c)$$

and likewise for $r(c \oplus a) = r(c \oplus b)$. Hence $\sim_r$ is indeed a congruence. □

We can also produce a reduction from a congruence. In fact, there will typically be many choices of reduction for a different congruence. Between S and $S/\sim$ there is a homomorphism $\rho^\natural$ called the *natural map*, taking each element of S to its $\sim$-equivalence class. If we choose a function going in the other direction, taking each equivalence class to some representative element within the class, then the composition of these two functions will be a reduction. The choice of representatives means that there may be multiple reduction functions, although they all correspond to the same congruence and define the same equivalence classes.

Lemma 2. *Let $(S, \oplus)$ be a semigroup, $\sim$ a congruence, and $\rho^\natural$ the natural map. If $\theta : S/\sim \longrightarrow S$ is such that $\rho^\natural \circ \theta = id$, then $\theta \circ \rho^\natural$ is a reduction; and $\sim$ is equal to $\sim_{\theta\circ\rho^\natural}$.*

Proof. Note that the condition $\rho^\natural \circ \theta = \mathrm{id}$ simply expresses that the representative for a class should be an element of that class. There is always at least one such θ, because there can be no empty classes. This condition also provides that θ must be one-to-one, for if $\theta(P)$ and $\theta(Q)$ are equal, then $(\rho^\natural \circ \theta)(P)$ and $(\rho^\natural \circ \theta)(Q)$ must also be equal; and then $P = Q$.

Now, $\theta \circ \rho^\natural$ satisfies the axioms for a reduction. Firstly, it is idempotent:

$$(\theta \circ \rho^\natural)^2 = \theta \circ (\rho^\natural \circ \theta) \circ \rho^\natural = \theta \circ \rho^\natural.$$

The second reduction axiom is also fulfilled

$$\begin{aligned}
(\theta \circ \rho^\natural)(a \oplus b) &= \theta(\rho^\natural(a) \oplus \rho^\natural(b)) && \text{since } \rho^\natural \text{ is a homomorphism} \\
&= \theta(\rho^\natural(\theta(\rho^\natural(a))) \oplus \rho^\natural(b)) && \text{since } \rho^\natural \circ \theta = \mathrm{id} \\
&= (\theta \circ \rho^\natural)((\theta \circ \rho^\natural)(a) \oplus b) && \text{since } \rho^\natural \text{ is a homomorphism.}
\end{aligned}$$

and symmetrically for the second equality.

Furthermore, the congruence derived from this reduction is $\sim$ again:

$$\begin{aligned}
a \sim_{\theta\circ\rho^\natural} b &\iff \theta(\rho^\natural(a)) = \theta(\rho^\natural(b)) \\
&\iff \rho^\natural(a) = \rho^\natural(b) && \text{since } \theta \text{ is one-to-one} \\
&\iff a \sim b && \text{by definition of the natural map.}
\end{aligned}$$

Hence for any congruence there is at least one equivalent reduction. □

We can therefore choose to represent any reduction r as a pair $(\sim, \theta)$, since this is enough to determine all of the values of the function. The interpretation of reductions in terms of congruences is helpful because it clarifies the true role of a reduction as well as often being more algebraically useful. A reduction is not an arbitrary transformation that fulfils some unusual axioms, but instead arises as the combination of a congruence—to say which distinctions between elements are being ignored—and a choice of representative element from each equivalence class. In some contexts, the reduction function may be the more natural way of

thinking about the operations being modelled. This justifies using the reduction idea in the first place, as opposed to making use of congruences throughout. The use of a functional viewpoint rather than a relational one may be more natural from the point of view of implementing a routing protocol, because it provides a direct answer to the question of how to deal with route data. On the other hand, the algebraic theory associated with congruences is much more extensive, which suggests that they should be the preferred representation when trying to prove facts about these algebraic structures.

In terms of algebraic constructions, the picture is that for a given *reduction* on one of our algebraic objects we can define the corresponding *congruence* and therefore the *quotient*.

Specifically, for a given $(S, \oplus, F)$ and reduction $r : S \longrightarrow S$ we can define the quotient S/r as follows.

1. The carrier consists of r-equivalence classes of elements of S; we can choose the canonical representative of each class to be a fixed point of r.
2. The semigroup operation is given by $\rho^\natural(a) \oplus/r \rho^\natural(b) = \rho^\natural(a \oplus b)$.
3. The functions in F are lifted: $f(\rho^\natural(a)) = \rho^\natural(f(a))$.

This can be verified to be a semigroup transform. The **minset** construction is clearly a special case, where r is min, S is a set of sets, and $\oplus$ is set union.

4 Applications in Routing

Aside from the use of min-like operations, our main application of congruences is in the handling of pathfinding errors. In practical situations, it is often not enough to have an algorithm simply throw its hands up and declare that no suitable solution exists. Instead, we would like to retain detailed information about what kinds of errors occurred. For example, in interdomain routing there are several reasons why a path might be considered erroneous:

- The same node is visited more than once.
- The path is intended to be filtered out.
- The path violates known economic relationships between networks.
- The path is too long (exceeding a maximum size for routing announcements).
- The origin is unexpected (given neighbours are only anticipated to advertise certain addresses).
- Route data is otherwise malformed.

Any or all of these could be true of a given path.

We believe that from a correctness point of view, it is not enough to sweep all of these under the implementers' rug. Many of the anomalies we observe in Internet routing today can be traced back to the handling of erroneous routes. Error handling is an integral part of the path selection process, and must be dealt with in the algebraic model, just as we deal with ordinary, non-erroneous routes. If not, then the correctness result we obtain is merely 'As long as nothing bad happens, protocol convergence is guaranteed', whereas we would prefer to

be in a position to make stronger statements about the resilience of the routing system even in the presence of errors.

A reduction operation is a suitable way to begin encoding error-handling. These functions are all about putting route data into a canonical form: this includes mapping certain routes to error values. In an algebra which includes such values, less preferred than 'ordinary' routes, we obtain the desired behaviour automatically. Erroneous routes are removed from consideration, since they cannot ever be more preferred than a non-erroneous route. Information about the error can still be propagated through the algorithm, enabling diagnosis, but this propagation is suppressed if an alternative route exists. All of this is totally compatible with multipath routing, via **minset** and related constructions.

The safety of these schemes depends on the interaction between

- the nature of path preferences;
- the operations extending paths; and
- the reduction function.

In the remainder of this section, we examine some simple examples of how the language of reductions and congruences can be used to prove required safety properties.

4.1 Forbidden Paths

Presentations of pathfinding algorithms tend to focus on computation of path *weight*, as opposed to returning the identity of each path. In the case of Dijkstra's algorithm, for example, a simple modification allows the recording of path information alongside weight information: this path information is not used while the algorithm is running, but is an additional output. But in our context, the degree of preference associated with a path depends upon the identity of that path—the nodes and arcs that make it up. In particular, we need to exclude, explicitly, paths that are not *simple*, whereas for conventional shortest path problems, this happens automatically. So we will, by default, want to include path information as part of the algebra.

Other paths may also be forbidden, even if they are simple. Network operators are able to make essentially arbitrary decisions about which paths will be unacceptable to them: in protocol implementations, they can be excluded from consideration as soon as they are received. This is equally the case in a multipath context.

Both of these cases can be handled by defining appropriate reductions. The obvious alternative would be to modify each algorithm to have the required behaviour, rather than seeking to encode this within the algebra. The problem with this idea is that it breaks the relationship with the theory of pathfinding based on linear algebra: if this link is not maintained, then we can no longer take advantage of existing theory in understanding the termination or efficiency of algorithms. In terms of convergence proof, our experience has been that it is a great help to make the algorithm as generic as possible, eliminating special cases by putting them into the algebra instead.

The general principle is to define a reduction which will eliminate forbidden paths, by mapping them onto a greatest element. This mirrors the conventional shortest-path model, where non-existent paths are given 'infinite' length. Because any path that is actually present will have finite length, these infinities will only persist in the algorithm if there is no path connecting the nodes in question. Equally, our forbidden paths will be worse than any permitted path, regardless of any of their other merits.

If $(S, \oplus, F)$ is a semigroup transform, with $\oplus$ commutative and having identity 0, and E is a subset of S containing 0, then define a function r_E on S by

$$r_E(x) = \begin{cases} x & x \notin E \\ 0 & x \in E. \end{cases} \tag{17}$$

For this to be a reduction, it is required that E satisfies the property

$$\forall e \in E, x \in S : (x \in E \wedge e \oplus x \in E) \vee (x = e \oplus x). \tag{18}$$

It is then possible to define a new structure based on r_E. This criterion makes operational sense. It states, in effect, that the forbidden paths have to be worse than the non-forbidden paths: if x does not emerge from $e \oplus x$, then all of e, x and $e \oplus x$ are in the error set. So if we forbid a certain path, then we also have to forbid any path for which it is a prefix: once in the error set, we cannot get out.

Definition 4. *Let* $\mathbf{err}(S, E)$ *be the semigroup transform* $(S_E, \oplus_E, F_E)$*, where*

- S_E *consists of those elements of* X *for which* $r_E(x) = x$*,*
- $x \oplus_E y$ *is* $r_E(x \oplus y)$*, and*
- F_E *consists of functions* f_E *for each* f *in* F*, and* $f_E(x) = r_E(f(x))$*.*

This $\oplus_E$ can be verified to be associative, since r_E is a reduction. The other properties of $\mathbf{err}(S, E)$ will depend on the choice of S and E.

We have reduced the error set E to a single element in the quotient. Anything in E is mapped to 0, the topmost element of the order; consequently, forbidden paths will be excluded from consideration, in favour of non-forbidden paths of any quality. This mapping is associated with each arc; operationally speaking, this means that on import or export, the forbidden paths are removed from the candidate set.

The congruence associated with such a reduction is related to the notion of a Rees congruence on a semigroup. A subset E of $(S, \oplus)$ is an *ideal* if

$$\forall x \in S, e \in E : (x \oplus e \in E) \wedge (e \oplus x \in E). \tag{19}$$

For a given ideal E, the relation

$$x \sim_E y \overset{\text{def}}{\iff} x = y \vee (x \in E \wedge y \in E) \tag{20}$$

is a congruence, called the *Rees congruence* with respect to E [7]. In the case of our r_E, the congruence may not be a Rees congruence because E may not

satisfy (19). This is in line with our general principle of not enforcing conditions which can be inferred: the definition of $\mathbf{err}(S, E)$ makes sense even when E is not an ideal, though it may not have desirable properties.

The relationship between reductions and congruences suggests that other representations of $\mathbf{err}(S, E)$ are possible. Specifically, we could preserve some information about the forbidden path, rather than limiting the available data to merely 'an error occurred'. As long as the correct rules are followed for F and $\oplus$, no difficulty arises. That is, we have to ensure that whatever representation we choose is equivalent under r to the semigroup transform $\mathbf{err}(S, E)$ above. Instead of mapping everything in E to a single 0, we could have many possibilities, drawn from a subset A of S. This will be acceptable if A is an upper set of S, and if r maps elements of A to elements of A. The correctness argument is the same, but the resulting solution state is perhaps more informative than previously, in the case when the only available path from i to j was forbidden.

4.2 Only Simple Paths

In the multipath setting, a slightly different definition is necessary. We will show an example of how to ensure that only simple paths emerge from the algorithm. The standard algebra of paths is to order them by length: we have a preference relation rather than a semilattice. A variation on the **minset** construction will convert such an algebra into one which can be used in the context of matrix operations.

Let P be the algebra of paths $(N^*, \preceq, C)$, where $p \preceq q$ if and only if $\mid p \mid \leq \mid q \mid$, and C consists of functions c_n for all n in N, which concatenate the node n onto the given path. Let $(S, \leq, F)$ be an order transform, which will be responsible for encoding the path weights.

Now, let E be the subset of $S \times N^*$ consisting of those pairs which contain a non-simple path:

$$\{(s, p) \in S \times N^* \mid p \text{ is not simple}\}. \tag{21}$$

The **err** construction cannot be used directly, since E does not satisfy the required property (18). However, there is a reduction which can be used over subsets of $S \times N^*$. Let r be the function

$$r(A) \stackrel{\text{def}}{=} \min(A \setminus E); \tag{22}$$

where min uses the lexicographic order on $S \times N^*$; this satisfies the reduction axioms. It is also operationally consistent with the view of path filtering wherein forbidden paths are removed first, with best-path selection applied to the remainder [12].

Consequently, a semigroup transform can be constructed where

- the elements are those subsets of $S \times N^*$ which are fixed points of r;
- the operation $\oplus$ is given by $A \oplus B \stackrel{\text{def}}{=} r(A \cup B)$; and

– the functions are pairs (f, c_n) for f in F, where

$$(f, c_n)(A) \stackrel{\text{def}}{=} r(\{(f(s), c_n(p)) \mid (s, p) \in A\}).$$

It can be seen that this algebra implements the simple paths criterion in the case of multipath routing: if during the course of computation a non-simple path is computed, it and its associated S-value will be removed from the candidate set.

It is possible to prove that the restriction to simple paths, together with the strict inflationary condition on S, suffice to ensure algorithmic convergence to a unique fixed point [8]. That is, the straightforward algorithm where every node periodically communicates its best paths to its neighbours, and updates its local best path data based on path information received from neighbours, is guaranteed to terminate; moreover, the final state will be a pure Nash equilibrium in the sense of Section 2, and is unique. Indeed, this convergence is guaranteed from *any* starting state, and so the algorithm can be considered to be self-synchronizing to the extent permitted by the nature of the underlying inter-node communication.

5 Algebraic Correctness in Finite Structures

The distinction between the finite and the infinite is of considerable practical importance in network routing. We have already discussed how convergence in a finite number of steps is greatly to be preferred. Another issue in correctness analysis where this distinction arises is in consideration of finite data domains. We almost invariably use the infinite to approximate the finite, working with an idealized, infinite algebraic structure such as $(\mathbb{N}, \min, +)$ for shortest paths, when the actual reality is that routing protocols only allow the expression of a finite number of distinct path lengths. In the case of the Routing Information Protocol (RIP), this finite number is fifteen [10].

The problem for algebraic analysis is that it is much easier to prove results about the infinite structures; indeed, the corresponding 'theorems' for finite structures may even be false. For example, we know that for the lexicographic product $\mathbf{lexprod}(S, T)$ of two semigroup transforms to be distributive, it suffices for S and T to be individually distributive, and for S to be cancellative, meaning that if $f(a) = f(b)$ then $a = b$, for any f in the function set of S. Addition of integers is a perfectly acceptable cancellative operation. But addition with a finite maximum value is not. On a given graph, our iterative algorithm could fail to reach a global optimum, due to lack of distributivity associated with this upper limit being reached. In particular, the problem would be that some node could be left with the value (∞_S, x) rather than the actual global optimum (∞_S, y), where $y \prec_T x$ according to the order $\preceq_T$ of T, and ∞_S denotes the maximum element of S. This is only a limited form of failure, especially since the termination of the iteration still occurs, but it does seem to undermine the promise of the algebraic method for guaranteeing correctness of pathfinding.

As an aside, the infamous 'counting to infinity' problem of RIP, whereby the protocol could take a long time to adapt to loss of connectivity, is *not* a

product of the handling of 'infinity' within RIP. Rather, it derives from the fact that routing information includes the weight of a path but not its identity, and that it is therefore possible for nodes to adopt cyclic paths without realizing. The cycles grow longer and longer, until the limit of sixteen is reached, this 'infinity' denoting the absence of a usable path. If RIP had a more generous notion of infinity, this problem would in fact be even worse, since convergence to the maximum value would take longer.

Returning to proofs of properties, the use of reductions or congruences can ease the difficulty here as well. We can use our **err** operation as part of a larger construction, and trace the correctness properties through. So for an algebra of the form $\mathbf{err}(\mathbf{lexprod}(S, T), E)$, we would use our theorems about the lexicographic product to derive properties of $\mathbf{lexprod}(S, T)$, and then use our theorems about **err** to derive properties of the whole algebra. The existence of these standard constructions allows many cases to be treated uniformly.

In the example above, the real issue is that elements like (∞_S, x) do not in fact denote usable paths: even if the value x is acceptable, the ∞_S is certainly not. Therefore, a way forward is to prohibit such elements from occurring in the computation at all. Take the subset $E = \{\infty_S\} \times T \subseteq S \times T$ and form the algebra $\mathbf{err}(\mathbf{lexprod}(S, T), E)$. All of the problematic elements are now identified, meaning that they are no longer barriers to the achieving of a global optimum. We also have a recipe for how to deal with such elements when they crop up in the path computation: map them to a single overall 'infinity' value, effectively by dropping the T component.

It can be shown that an algebra of this form is distributive, if we have a distributivity condition for the appropriate subset of $\mathbf{lexprod}(S, T)$ (see [8], Theorem 5.9 and Appendix A.5). The condition is that

$$(f, g)(s_1, t_1) \oplus (f, g)(s_2, t_2) = (f, g)\left((s_1, t_1) \oplus (s_2, t_t)\right) \tag{23}$$

for all (f, g) in the function set of $\mathbf{lexprod}(S, T)$, and all (s_1, t_1) and (s_2, t_2) in the subset $(S \setminus \{\infty_S\}) \times T$ of $S \times T$.

In this way, the required correctness property can be regained, by a modification to the algebra and the use of reduction- or congruence-based theorems.

6 Conclusion

There is an ongoing effort to provide a sound theoretical foundation for Internet routing. While in many cases this task can be tackled on an ad-hoc basis, by writing new definitions and proofs for each proposed routing scheme, a better approach is to provide a general theory which can address several such models. The existing pathfinding theory based on semirings is a sound starting point, but several adaptations need to be made in order to make it applicable to these practical examples.

This paper has demonstrated that several such alterations are more mathematically rich than might be suspected. The apparently awkward 'min' operation

has been revealed as having a deep connection with lattice theory and with congruences. Related 'reduction' operations are also susceptible to explanation in terms of congruences. We have also shown that these operations are useful in multipath routing, and for more complex scenarios incorporating route filtering.

The examples in this paper are inspired by interdomain routing. There is considerable scope for applying this theory to the design of future routing systems, so that they can be not only flexible, but also provably correct with reference to an underlying optimization problem.

Acknowledgments

This work was supported by grant EP/F002718/1 from the Engineering and Physical Sciences Research Council (EPSRC). The authors would like to thank Georg Struth and Philip Taylor for their helpful comments.

References

1. Birkhoff, G.: Lattice Theory. American Mathematical Society, Providence (1948)
2. Carré, B.A.: Graphs and networks. Oxford University Press, Oxford (1979)
3. Gondran, M., Minoux, M.: Graphs and algorithms. Wiley, Chichester (1984)
4. Gondran, M., Minoux, M.: Graphes, dioïdes et semi-anneaux: Nouveaux modèles et algorithmes. Tec & Doc, Paris (2001)
5. Griffin, T.G., Gurney, A.J.T.: Increasing bisemigroups and algebraic routing, In: Berghammer, R., Möller, B., Struth, G. (eds.) RelMiCS/AKA 2008. LNCS, vol. 4988, pp. 123–137. Springer, Heidelberg (2008)
6. Griffin, T.G., Shepherd, F.B., Wilfong, G.: The stable paths problem and interdomain routing. IEEE/ACM Trans. Netw. 10(2), 232–243 (2002)
7. Grillet, P.A.: Semigroups: An introduction to the structure theory. Monographs and Textbooks in Pure and Applied Mathematics, vol. 193. Marcel Dekker, New York (1995)
8. Gurney, A.J.T.: Construction and verification of routing algebras. PhD thesis, University of Cambridge (2009)
9. Kruskal, J.B.: The theory of well-quasi-ordering: A frequently discovered concept. J. Combin. Theory Ser. A 13(3), 297–305 (1972)
10. Malkin, G.: RIP version 2. RFC 2453 (1998).
11. Rote, G.: Path problems in graphs. In: Tinhofer, G., Mayr, E.W., Noltemeier, H., Syslo, M. (eds.) Computational graph theory. Computing Supplementa, vol. 7, pp. 155–189. Springer, Heidelberg (1990)
12. Wang, Y., Schapira, M., Rexford, J.: Neighbor-specific BGP: More flexible routing policies while improving global stability. In: Douceur, J.R., Greenberg, A.G., Bonald, T., Nieh, J. (eds.) Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems, SIGMETRICS/Performance 2009, pp. 217–228. ACM, New York (2009)
13. Wongseelashote, A.: An algebra for determining all path-values in a network with application to k-shortest-paths problems. Networks 6(4), 307–334 (1976)
14. Wongseelashote, A.: Semirings and path spaces. Discrete Math. 26(1), 55–78 (1979)
15. Zimmermann, U.: Linear and combinatorial optimization in ordered algebraic structures. Annals of Discrete Mathematics, vol. 10. Elsevier North-Holland, Amsterdam (1981)