L11: Algebraic Path Problems with applications to Internet Routing
Lectures 12, 13, 14

Timothy G. Griffin

timothy.griffin@cl.cam.ac.uk
Computer Laboratory
University of Cambridge, UK

Michaelmas Term, 2014

The plan

- Lecture 12: “functions on arcs”
- Lecture 13: Global vs. Local Optimality
- Lecture 14: A simple model of a “fixed” BGP
- Lecture 15: Proof of convergence of iteration for (some) non-distributed algebras ...
- Lecture 16: ... the proof continues

Path Weight with functions on arcs?

For graph $G = (V,\ E)$, and arc path $p = (u_0,\ u_1)(u_1,\ u_2)\cdots(u_{k-1},\ u_k)$.

Functions on arcs: two natural ways to do this...

Weight function $w : E \rightarrow (S \rightarrow S)$. Let $f_j = w(u_{j-1}, u_j)$.

$$
\begin{array}{rclcl}
w_a^L(p) & = & f_1(f_2(\cdots f_k(a)\cdots)) & = & (f_1 \circ f_2 \circ \cdots \circ f_k)(a) \\
w_a^R(p) & = & f_k(f_{k-1}(\cdots f_1(a)\cdots)) & = & (f_k \circ f_{k-1} \circ \cdots \circ f_1)(a)
\end{array}
$$

How can we "make this work" for path problems?

Algebra of Monoid Endomorphisms (AME) (See Gondran and Minoux 2008)

Let $(S,\ \oplus,\ \overline{0})$ be a commutative monoid.

$(S,\ \oplus,\ F \subseteq S \to S,\ \overline{0})$ is an algebra of monoid endomorphisms (AME) if

- $\forall f \in F,\ f(\overline{0}) = \overline{0}$
- $\forall f \in F,\ \forall b, c \in S,\ f(b \oplus c) = f(b) \oplus f(c)$

I will declare these as optional

- $\forall f, g \in F,\ f \circ g \in F$ (closed)
- $\exists i \in F,\ \forall s \in S,\ i(s) = s$
- $\exists \omega \in F,\ \forall n \in N,\ \omega(n) = \overline{0}$

Note: as with semirings, we may have to drop some of these axioms in order to model Internet routing ...

So why do we want AMEs?

Each (closed with ω and i) AME can be viewed as a semiring of functions. Suppose $(S, \oplus, F, \overline{0})$ is an algebra of monoid endomorphisms. We can turn it into a semiring

$$\mathbb{F} = (F, \hat{\oplus}, \circ, \omega, i)$$

where $(f \hat{\oplus} g)(a) = f(a) \oplus g(a)$ and $(f \circ g)(a) = f(g(a))$.

But functions are hard to work with....

- All algorithms need to check equality over elements of a semiring
- $f = g$ means $\forall a \in S, f(a) = g(a)$
- S can be very large, or infinite ....

How do we represent a set of functions $F \subseteq S \to S$?

Assume we a set L and a function

$$\rhd \in L \to (S \to S).$$

We normally write $l \rhd s$ rather than $\rhd(l)(s)$. We think of $l \in L$ as the index for a function $f_l(s) = l \rhd s$. In this way $(L, \rhd)$ can be used to represent the set of functions

$$F = \{f_l = \lambda s.(l \rhd s) \mid l \in L\}.$$

Indexed Algebra of Monoid Endomorphisms (IAME)

Let $(S,\ \oplus,\ \overline{0})$ be a commutative and idempotent monoid.

A (left) IAME $(S,\ L,\ \oplus,\ \rhd,\ \overline{0})$

- $\rhd \in L \to (S \to S)$
- $\forall l \in L,\ l \rhd \overline{0} = \overline{0}$
- $\exists l \in L,\ \forall s \in S,\ l \rhd s = s$
- $\exists l \in L,\ \forall s \in S,\ l \rhd s = \overline{0}$
- $\forall l \in L,\ \forall n, m \in S,\ l \rhd (n \oplus m) = (l \rhd n) \oplus (l \rhd m)$

When we need closure? Not very often! If needed, it would be

$$\forall l_1, l_2 \in L, \exists l_3 \in L, \forall s \in S, l_3 \rhd s = l_1 \rhd (l_2 \rhd s)$$

IAME of Matrices

Given a left IAME $(S,\ L,\ \oplus,\ \rhd,\ \overline{0})$ define the left IAME of matrices

$$(\mathbb{M}_n(S),\ \mathbb{M}_n(L),\ \oplus,\ \rhd,\ \mathbf{J}).$$

For all i, j we have $\mathbf{J}(i, j) = \overline{0}$. For $\mathbf{A} \in \mathbb{M}_n(L)$ and $\mathbf{B}, \mathbf{C} \in \mathbb{M}_n(S)$ define

$$(\mathbf{B} \oplus \mathbf{C})(i,\ j) \quad = \quad \mathbf{B}(i,\ j) \oplus \mathbf{C}(i,\ j)$$

$$(\mathbf{A} \rhd \mathbf{B})(i,\ j) \quad = \quad \bigoplus_{1 \leq q \leq n} \mathbf{A}(i,\ q) \rhd \mathbf{B}(q,\ j)$$

Solving (some) equations. Left version here ...

We will be interested in solving for **L** equations of the form

$$\mathbf{L} = (\mathbf{A} \rhd \mathbf{L}) \oplus \mathbf{B}$$

Let

$$\begin{aligned} \mathbf{A} \rhd^0 \mathbf{B} &= \mathbf{B} \\ \mathbf{A} \rhd^{k+1} \mathbf{B} &= \mathbf{A} \rhd (\mathbf{A} \rhd^k \mathbf{B}) \end{aligned}$$

and

$$\mathbf{A} \rhd^{(k)} \mathbf{B} = \mathbf{A} \rhd^0 \mathbf{B} \oplus \mathbf{A} \rhd^1 \mathbf{B} \oplus \mathbf{A} \rhd^2 \mathbf{B} \oplus \cdots \oplus \mathbf{A} \rhd^k \mathbf{B}$$

$$\mathbf{A} \rhd^* \mathbf{B} = \mathbf{A} \rhd^0 \mathbf{B} \oplus \mathbf{A} \rhd^1 \mathbf{B} \oplus \mathbf{A} \rhd^2 \mathbf{B} \oplus \cdots \oplus \mathbf{A} \rhd^k \mathbf{B} \oplus \cdots$$

Definition (q stability)

If there exists a q such that for all **B**, $\mathbf{A} \rhd^q \mathbf{B} = \mathbf{A} \rhd^{q+1} \mathbf{B}$, then **A** is *$q$-stable*. Therefore, $\mathbf{A} \rhd^* \mathbf{B} = \mathbf{A} \rhd^q \mathbf{B}$.

Key results (again)

Theorm 11.1

If $\mathbf{A}$ is q-stable, then $\mathbf{L} = \mathbf{A} \rhd^{*} (\mathbf{B})$ solves the equation

$$\mathbf{L} = (\mathbf{A} \rhd \mathbf{L}) \ \oplus \ \mathbf{B}.$$

Theorem 11.2

If $\mathbf{A}$ is q-stable, then $\mathbf{L} = \mathbf{A} \rhd^{*} (\mathbf{B})$ solves the equation

$$\mathbf{L} = (\mathbf{A} \rhd \mathbf{L}) \ \oplus \ \mathbf{B}.$$

Something familiar : Lexicographic product of AMEs

$$(S,\ L_S,\ \oplus_S,\ \rhd_S)\vec{\times}(T,\ L_T,\ \oplus_T,\ \rhd_T) = (S\times T,\ \ L_S\times L_T,\ \oplus_S\vec{\times}\oplus_T,\ \rhd_S\times\rhd_T)$$

Theorem 11.3

$$\mathrm{D}(S \vec{\times} T) \iff \mathrm{D}(S) \wedge \mathrm{D}(T) \wedge (\mathrm{C}(S) \vee \mathrm{K}(T))$$

Where

Property	Definition
D	$\forall a, b, f,\ \ f(a \oplus b) = f(a) \oplus f(b)$
C	$\forall a, b, f,\ \ f(a) = f(b) \implies a = b$
K	$\forall a, b, f,\ \ f(a) = f(b)$

Something new: Functional Union of AMEs

$$(S,\ L_1,\ \oplus,\ \rhd_1) +_{\mathrm{m}} (S,\ L_2,\ \oplus,\ \rhd_2) = (S,\ L_1 \uplus L_2,\ \oplus,\ \rhd_1 \uplus \rhd_2)$$

Fact

$$\mathrm{D}((S,\ L_1,\ \oplus,\ \rhd_1) +_{\mathrm{m}} (S,\ L_2,\ \oplus,\ \rhd_2))$$
$$\iff$$
$$\mathrm{D}((S,\ L_1,\ \oplus,\ \rhd_1)) \wedge \mathrm{D}((S,\ L_2,\ \oplus,\ \rhd_2))$$

Where

$$\begin{array}{rcl} (\mathrm{inl}(l))\ (\rhd_1 \uplus \rhd_2)\ s & = & l\ \rhd_1\ s \\ (\mathrm{inr}(l))\ (\rhd_1 \uplus \rhd_2)\ s & = & l\ \rhd_2\ s \end{array}$$

Left and Right

$(S, \{R\}, \oplus, \textbf{right})$

$$R \ \textbf{right} \ s = s$$

$(S, S, \oplus, \textbf{left})$

$$s_1 \ \textbf{left} \ s_2 = s_1$$

Facts

The following are always true.

$$\begin{array}{ll} \mathrm{D}((S, \{R\}, \oplus, \textbf{right})) & \\ \mathrm{D}((S, S, \oplus, \textbf{left})) & \text{(assuming } \oplus \text{ is idempotent)} \\ \mathrm{C}((S, \{R\}, \oplus, \textbf{right})) & \\ \mathrm{K}((S, S, \oplus, \textbf{left})) & \end{array}$$

Scoped Product (Think iBGP/eBGP)

$$S \Theta T = (S \vec{\times} \textbf{left}(T)) +_{\mathrm{m}} (\textbf{right}(S) \vec{\times} T)$$

Theorem 11.2

$$\mathrm{D}(S \Theta T) \iff \mathrm{D}(S) \wedge \mathrm{D}(T).$$

$$\begin{aligned}
& \mathrm{D}(S \Theta T) \\
& \mathrm{D}((S \vec{\times} \textbf{left}(T)) +_{\mathrm{m}} (\textbf{right}(S) \vec{\times} T)) \\
\iff & \mathrm{D}(S \vec{\times} \textbf{left}(T)) \wedge \mathrm{D}(\textbf{right}(S) \vec{\times} T) \\
\iff & \mathrm{D}(S) \wedge \mathrm{D}(\textbf{left}(T)) \wedge (\mathrm{C}(S) \vee \mathrm{K}(\textbf{left}(T))) \\
& \quad \wedge \mathrm{D}(\textbf{right}(S)) \wedge \mathrm{D}(T) \wedge (\mathrm{C}(\textbf{right}(S)) \vee \mathrm{K}(T)) \\
\iff & \mathrm{D}(S) \wedge \mathrm{D}(T)
\end{aligned}$$

Lexicographic Products in Metarouting. Alexander Gurney, Timothy G. Griffin. International Conference on Network Protocols (ICNP), 2007.

Scoped Product

$$
\begin{array}{rcl}
S & = & (S,\ L_S,\ \oplus_S,\ \rhd_S) \\
T & = & (T,\ L_T,\ \oplus_T,\ \rhd_T) \\
S \vec{\times}\ \mathbf{left}(T) & = & (S \times T,\ L_S \times T,\ \oplus_S \vec{\times} \oplus_T,\ \rhd_S \times \mathbf{left}) \\
\mathbf{right}(S) \vec{\times}\ T & = & (S \times T,\ \{R\} \times L_T,\ \oplus_S \vec{\times} \oplus_T,\ \mathbf{right} \times \rhd_T) \\
S \Theta T & = & (S \vec{\times}\ \mathbf{left}(T)) +_{\mathrm{m}} (\mathbf{right}(S) \vec{\times}\ T) \\
 & = & (S \times T,\ (L_S \times T) \uplus (\{R\} \times L_T),\ \oplus_S \vec{\times} \oplus_T,\ \rhd)
\end{array}
$$

Between regions ($\lambda \in L_S$)

$$\mathrm{inl}(\lambda,\ t_2) \rhd (s,\ t_1) = (\lambda \rhd_S s,\ t_2)$$

Within regions ($\lambda \in L_T$)

$$\mathrm{inr}(R,\ \lambda) \rhd (s,\ t) = (s,\ \lambda \rhd_T t)$$

Recall (Lecture 1)

The Algorithm to Algebra (A2A) method

$$\left(\begin{array}{c}\text{original metric}\\ +\\ \text{complex algorithm}\end{array}\right) \rightarrow \left(\begin{array}{c}\text{modified metric}\\ +\\ \text{generic algorithm}\end{array}\right)$$

Punch Line

A2A attempts to shift complexity from an algorithm to the metric, which is captured in an algebraic structure — the algebraic properties of that structure will determine what kind of solution is obtained (global or local optima).

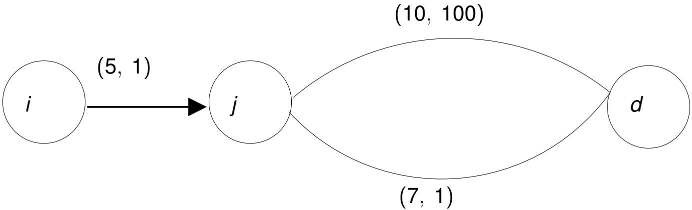
Recall puzzle from Lecture 1

name	S	$\oplus$,	$\otimes$	$\overline{0}$	$\overline{1}$
$\mathrm{min\_plus}$	$\mathbb{N}$	min	$+$		0
$\mathrm{max\_min}$	$\mathbb{N}$	max	min	0	

name	LD	LC	LK
$\mathrm{min\_plus}$	Yes	Yes	No
$\mathrm{max\_min}$	Yes	No	No

name	definition	LD
Widest Shortest-paths	$\mathrm{min\_plus} \vec{\times} \mathrm{max\_min}$	Yes
Shortest Widest-paths	$\mathrm{max\_min} \vec{\times} \mathrm{min\_plus}$	No

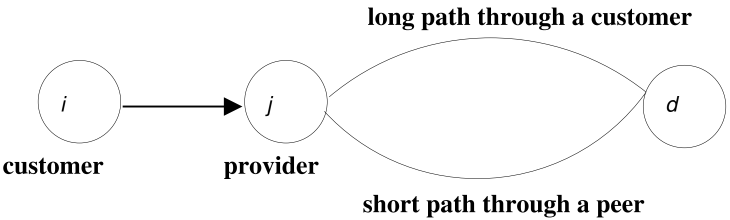
Shortest widest paths



- node j prefers $(10, 100)$ over $(7, 1)$.
- node i prefers $(5, 2)$ over $(5, 101)$.

$$(5, 1) \otimes ((10, 100) \oplus (7, 1)) = (5, 1) \otimes (10, 100) = (5, 101)$$
$$((5, 1) \otimes (10, 101)) \oplus ((5, 1) \otimes (7, 1)) = (5, 101) \oplus (5, 2) = (5, 2)$$

Something similar from inter-domain routing in the global Internet



- j prefers long path though one of its customers
- i prefers the shorter path

Solving (some) equations

If $\mathbf{A}^*$ exists , then $\mathbf{L} = \mathbf{A}^*$ solves the equation

$$\mathbf{L} = \mathbf{A}\mathbf{L} \oplus \mathbf{I}$$

and $\mathbf{R} = \mathbf{A}^*$ solves the equation

$$\mathbf{R} = \mathbf{R}\mathbf{A} \oplus \mathbf{I}.$$

Towards a “non classical” theory of algebraic path problems ...

If we weaken the axioms of the semiring (drop distributivity, for example), could it be that we can find examples where $\mathbf{A}^*$, $\mathbf{L}$, and $\mathbf{R}$ exist, but are all distinct?

Health warning : matrix multiplication over structures lacking distributivity is not associative!

Left-Local Optimality

Say that $\mathbf{L}$ is a left locally-optimal solution when

$$\mathbf{L} = (\mathbf{A} \otimes \mathbf{L}) \oplus \mathbf{I}.$$

That is, for $i \neq j$ we have

$$\mathbf{L}(i,\ j) = \bigoplus_{q \in V} \mathbf{A}(i,\ q) \otimes \mathbf{L}(q,\ j)$$

- $\mathbf{L}(i,\ j)$ is the best possible value given the values $\mathbf{L}(q,\ j)$, for all out-neighbors q of source i.
- Rows $\mathbf{L}(i,\ \_)$ represents **out-trees <u>from</u>** i (think Bellman-Ford).
- Columns $\mathbf{L}(\_,\ i)$ represents **in-trees <u>to</u>** i.
- Works well with hop-by-hop forwarding from i.

Right-Local Optimality

Say that $\mathbf{R}$ is a right locally-optimal solution when

$$\mathbf{R} = (\mathbf{R} \otimes \mathbf{A}) \oplus \mathbf{I}.$$

That is, for $i \neq j$ we have

$$\mathbf{R}(i,\ j) = \bigoplus_{q \in V} \mathbf{R}(i,\ q) \otimes \mathbf{A}(q,\ j)$$

- $\mathbf{R}(i,\ j)$ is the best possible value given the values $\mathbf{R}(q,\ j)$, for all in-neighbors q of destination j.
- Rows $\mathbf{L}(i,\ \_)$ represents **out-trees <u>from</u>** i (think Dijkstra).
- Columns $\mathbf{L}(\_,\ i)$ represents **in-trees <u>to</u>** i.

With and Without Distributivity

With distributivity

For (bounded) semirings, the three optimality problems are essentially the same — locally optimal solutions are globally optimal solutions.

$$\mathbf{A}^* = \mathbf{L} = \mathbf{R}$$

Without distributivity

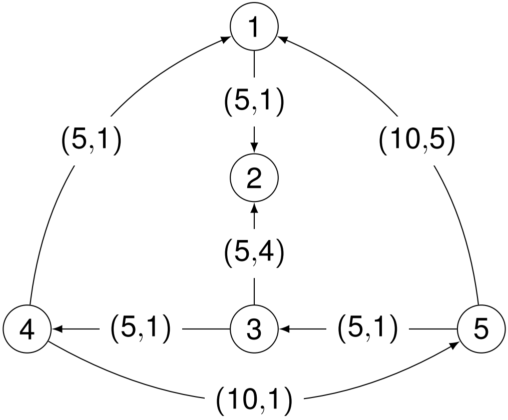
It may be that $\mathbf{A}^*$, $\mathbf{L}$, and $\mathbf{R}$ exists but are all distinct.

Back and Forth

$$\mathbf{L} = (\mathbf{A} \otimes \mathbf{L}) \oplus \mathbf{I} \quad \Longleftrightarrow \quad \mathbf{L}^T = (\mathbf{L}^T \otimes^T \mathbf{A}^T) \oplus \mathbf{I}$$

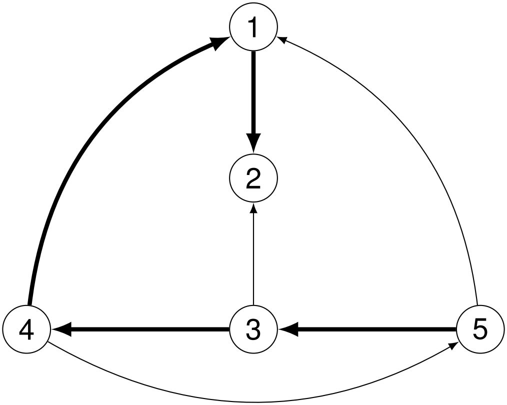
where $\otimes^T$ is matrix multiplication defined with $a \otimes^T b = b \otimes a$

Example

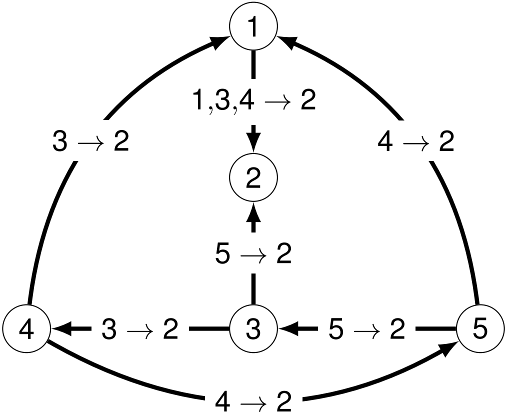


(bandwidth, distance) with lexicographic order (bandwidth first).

Left-locally optimal paths to node 2



Right-locally optimal paths to node 2



(Distributed) Bellman-Ford can compute left-local solutions[1]

$$\begin{aligned} \mathbf{A}^{[0]} &= \mathbf{I} \\ \mathbf{A}^{[k+1]} &= (\mathbf{A} \otimes \mathbf{A}^{k}) \oplus \mathbf{I}, \end{aligned}$$

- Bellman-ford algorithm must be modified to ensure only loop-free paths are inspected.
- $(S, \oplus, \overline{0})$ is a commutative, idempotent, and selective monoid,
- $(S, \otimes, \overline{1})$ is a monoid,
- $\overline{0}$ is the annihilator for $\otimes$,
- $\overline{1}$ is the annihilator for $\oplus$,
- Left strictly inflationarity, L.S.INF : $\forall a, b : a \neq \overline{0} \implies a < a \otimes b$
- Here $a \leq b \equiv a = a \oplus b$.

Convergence to a unique left-local solution is guaranteed. Currently no polynomial bound is known on the number of iterations required.

[1] See dissertation of Alexander Gurney

Sobrinho's encoding of the Gao/Rexford rules for BGP

Additive component uses min with

- 0 is the type of a downstream route,
- 1 is the type of a peer route, and
- 2 is the type of an upstream route.
- ∞ is the type of no route.

Multiplicative component

	0	1	2	∞
0	0	∞	∞	∞
1	1	∞	∞	∞
2	2	2	2	∞
∞	∞	∞	∞	∞

Note that this is not associative! In addition, this models just the "local preference" component of BGP. Not this must be combined with a lexicographic product. Can we improve on this?

Important properties for algebraic structures of the form $(S,\ \oplus,\ F,\ \overline{0},\ \overline{1})$

property	definition
D	$\forall a, b \in S,\ f \in F\ :\ f(a \oplus b) = f(a) \oplus f(b)$
INFL	$\forall a \in S,\ f \in F\ :\ a \leq f(a)$
S.INFL	$\forall a \in S,\ F \in F\ :\ a \neq \overline{0} \implies a < f(a)$
K	$\forall a, b \in S,\ f \in F\ :\ f(a) = f(b) \implies a = b$
$K_{\overline{0}}$	$\forall a, b \in S,\ f \in F\ :\ f(a) = f(b) \implies (a = b \vee f(a) = \overline{0})$
C	$\forall a, b \in S,\ f \in F\ :\ f(a) = f(b)$
$C_{\overline{0}}$	$\forall a, b \in S,\ f \in F\ :\ f(a) \neq f(b) \implies (f(a) = \overline{0} \vee f(b) = \overline{0})$

Stratified Shortest-Paths Metrics

Metrics

$$(s,\ d)\ \text{ or }\ \infty$$

- $s \neq \infty$ is a stratum level in $\{0, 1,\ 2, \ldots,\ m-1\}$,
- d is a "shortest-paths" distance,
- Routing metrics are compared lexicographically

$$(s_1,\ d_1) < (s_2,\ d_2) \iff (s_1 < s_2) \vee (s_1 = s_2 \wedge d_1 < d_2)$$

Stratified Shortest-Paths Policies

Policy has form $(f,\ d)$

$$
\begin{aligned}
(f,\ d)(s,\ d') &= \langle f(s),\ d + d' \rangle \\
(f,\ d)(\infty) &= \infty
\end{aligned}
$$

where

$$
\langle s,\ t \rangle = \begin{cases} \infty & (\text{if } s = \infty) \\ (s,\ t) & (\text{otherwise}) \end{cases}
$$

Constraint on Policies

$$(f,\ d)$$

- Either f is inflationary and $0 < d$,
- or f is strictly inflationary and $0 \leq d$.

Why?

$$(\text{S.INFL}(S) \vee (\text{INFL}(S) \wedge \text{S.INFL}(T))) \implies \text{S.INFL}(S \vec{\times}_{\overline{0}} T).$$

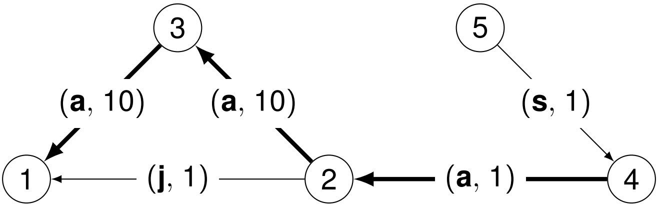
All Inflationary Policy Functions for Three Strata

	0	1	2	D	K_∞	C_∞		0	1	2	D	K_∞	C_∞
a	0	1	2	$\star$	$\star$		**m**	2	1	2			
b	0	1	∞	$\star$	$\star$		**n**	2	1	∞		$\star$	
c	0	2	2	$\star$			**o**	2	2	2	$\star$		$\star$
d	0	2	∞	$\star$	$\star$		**p**	2	2	∞	$\star$		$\star$
e	0	∞	2		$\star$		**q**	2	∞	2			$\star$
f	0	∞	∞	$\star$	$\star$	$\star$	**r**	2	∞	∞	$\star$	$\star$	$\star$
g	1	1	2	$\star$			**s**	∞	1	2		$\star$	
h	1	1	∞	$\star$		$\star$	**t**	∞	1	∞		$\star$	$\star$
i	1	2	2	$\star$			**u**	∞	2	2			$\star$
j	1	2	∞	$\star$	$\star$		**v**	∞	2	∞		$\star$	$\star$
k	1	∞	2		$\star$		**w**	∞	∞	2		$\star$	$\star$
l	1	∞	∞	$\star$	$\star$	$\star$	**x**	∞	∞	∞	$\star$	$\star$	$\star$

Almost shortest paths

	0	1	2	D	K_∞	interpretation
a	0	1	2	$\star$	$\star$	$+0$
j	1	2	∞	$\star$	$\star$	$+1$
r	2	∞	∞	$\star$	$\star$	$+2$
x	∞	∞	∞	$\star$	$\star$	$+3$
b	0	1	∞	$\star$	$\star$	filter 2
e	0	∞	2		$\star$	filter 1
f	0	∞	∞	$\star$	$\star$	filter 1, 2
s	∞	1	2		$\star$	filter 0
t	∞	1	∞		$\star$	filter 0, 2
w	∞	∞	2		$\star$	filter 0, 1

Shortest paths with filters, over **INF**$_3$



Note that the path $5, 4, 2, 1$ with weight $(1,\ 3)$ would be the globally best path from node 5 to node 1. But in this case, poor node 5 is left with no path! The locally optimal solution has $\mathbf{R}(5,\ 1) = \infty$.

Both D and $K_{\overline{0}}$

This makes combined algebra distributive!

	0	1	2
a	0	1	2
b	0	1	∞
d	0	2	∞
f	0	∞	∞
j	1	2	∞
l	1	∞	∞
r	2	∞	∞
x	∞	∞	∞

Why?

$$(\mathrm{D}(S) \wedge \mathrm{D}(T) \wedge \mathrm{K}_{\overline{0}}(S)) \implies \mathrm{D}(S \vec{\times}_{\overline{0}} T)$$

BGP : standard view

- 0 is the type of a downstream route,
- 1 is the type of a peer route, and
- 2 is the type of an upstream route.

	0	1	2
f	0	∞	∞
l	1	∞	∞
o	2	2	2

"Autonomous" policies

	0	1	2	D	K_∞
f	0	∞	∞	$\star$	$\star$
h	1	1	∞	$\star$	
l	1	∞	∞	$\star$	$\star$
o	2	2	2	$\star$	
p	2	2	∞	$\star$	
q	2	∞	2		
r	2	∞	∞	$\star$	$\star$
t	∞	1	∞		$\star$
u	∞	2	2		
v	∞	2	∞		$\star$
w	∞	∞	2		$\star$
x	∞	∞	∞	$\star$	$\star$