L11: Algebraic Path Problems with applications to Internet Routing
Lectures 11, 12

Timothy G. Griffin

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

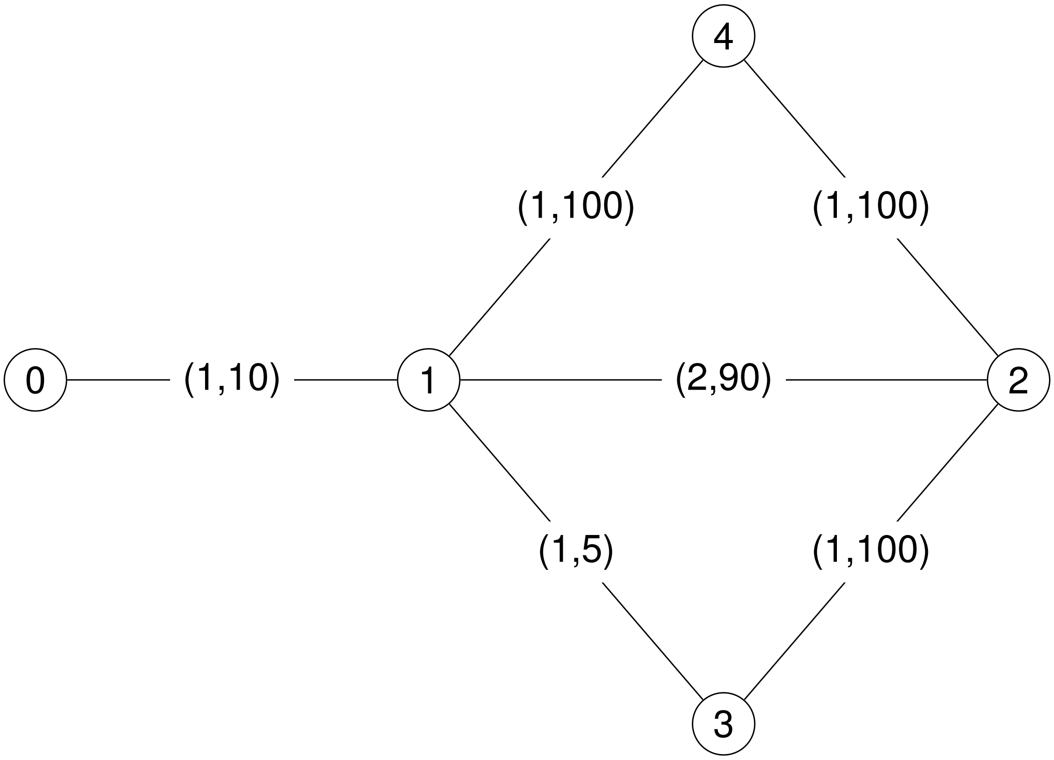
Michaelmas Term, 2016

Widest shortest-paths

- Metric of the form (d, b), where d is distance $(\min, +)$ and b is capacity $(\max, \min)$.
- Metrics are compared lexicographically, with distance considered first.
- Such things are found in the vast literature on Quality-of-Service (QoS) metrics for Internet routing.

$$\text{wsp} = \text{sp} \vec{\times} \text{bw}$$

Widest shortest-paths



Weights are globally optimal (we have a semiring)

Widest shortest-path weights computed by Dijkstra and Bellman-Ford

$$\mathbf{R} \quad = \quad \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} \quad 0 \quad & \quad 1 \quad & \quad 2 \quad & \quad 3 \quad & \quad 4 \quad \end{array} \\ \left[\begin{array}{ccccc} (0, \top) & (1, 10) & (3, 10) & (2, 5) & (2, 10) \\ (1, 10) & (0, \top) & (2, 100) & (1, 5) & (1, 100) \\ (3, 10) & (2, 100) & (0, \top) & (1, 100) & (1, 100) \\ (2, 5) & (1, 5) & (1, 100) & (0, \top) & (2, 100) \\ (2, 10) & (1, 100) & (1, 100) & (2, 100) & (0, \top) \end{array} \right] \end{array}$$

But what about the paths themselves?

Four optimal paths of weight $(3, 10)$.

$$\begin{array}{lcl} \mathbf{P}_{\text{optimal}}(0,2) & = & \{(0,1,2),\ (0,1,4,2)\} \\ \mathbf{P}_{\text{optimal}}(2,0) & = & \{(2,1,0),\ (2,4,1,0)\} \end{array}$$

There are standard ways to extend Bellman-Ford and Dijkstra to compute paths (or the associated next hops).

Do these extended algorithms find all optimal paths?

Surprise!

Four optimal paths of weight $(3, 10)$

$$\begin{array}{lcl} \mathbf{P}_{\text{optimal}}(0,2) & = & \{(0,1,2),\ (0,1,4,2)\} \\ \mathbf{P}_{\text{optimal}}(2,0) & = & \{(2,1,0),\ (2,4,1,0)\} \end{array}$$

Paths computed by (extended) Dijkstra

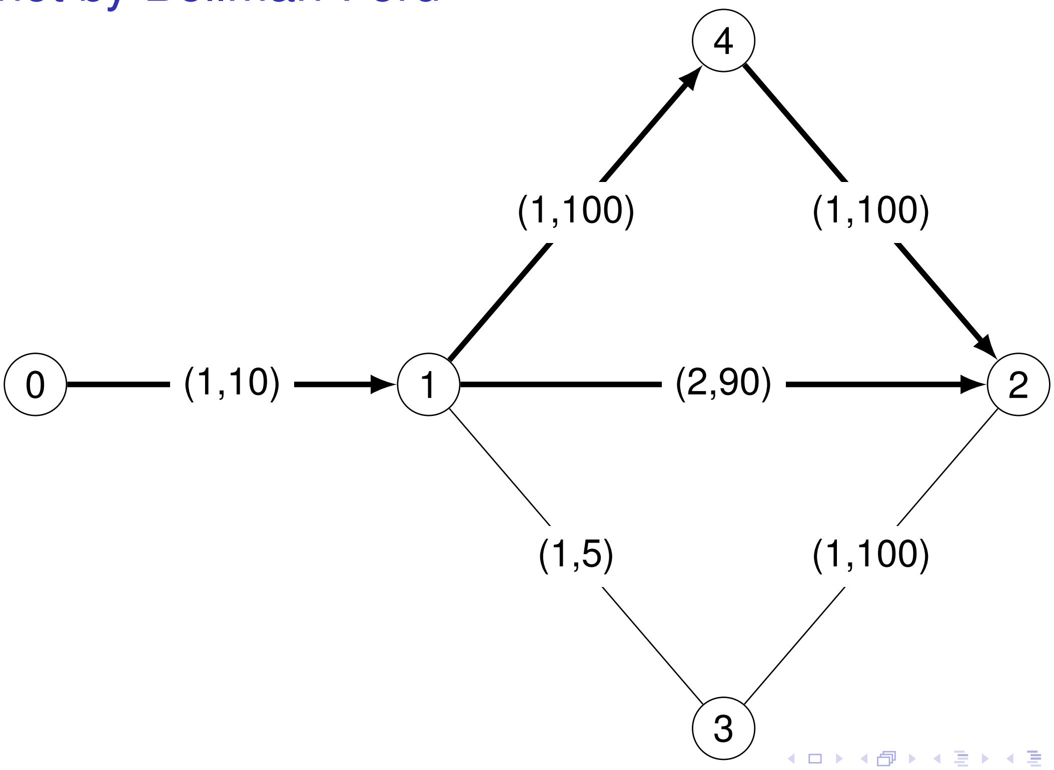
$$\begin{array}{lcl} \mathbf{P}_{\text{Dijkstra}}(0,2) & = & \{(0,1,2),\ (0,1,4,2)\} \\ \mathbf{P}_{\text{Dijkstra}}(2,0) & = & \{(2,4,1,0)\} \end{array}$$

Notice that 0's paths cannot both be implemented with next-hop forwarding since $\mathbf{P}_{\text{Dijkstra}}(1,2) = \{(1,4,2)\}$.

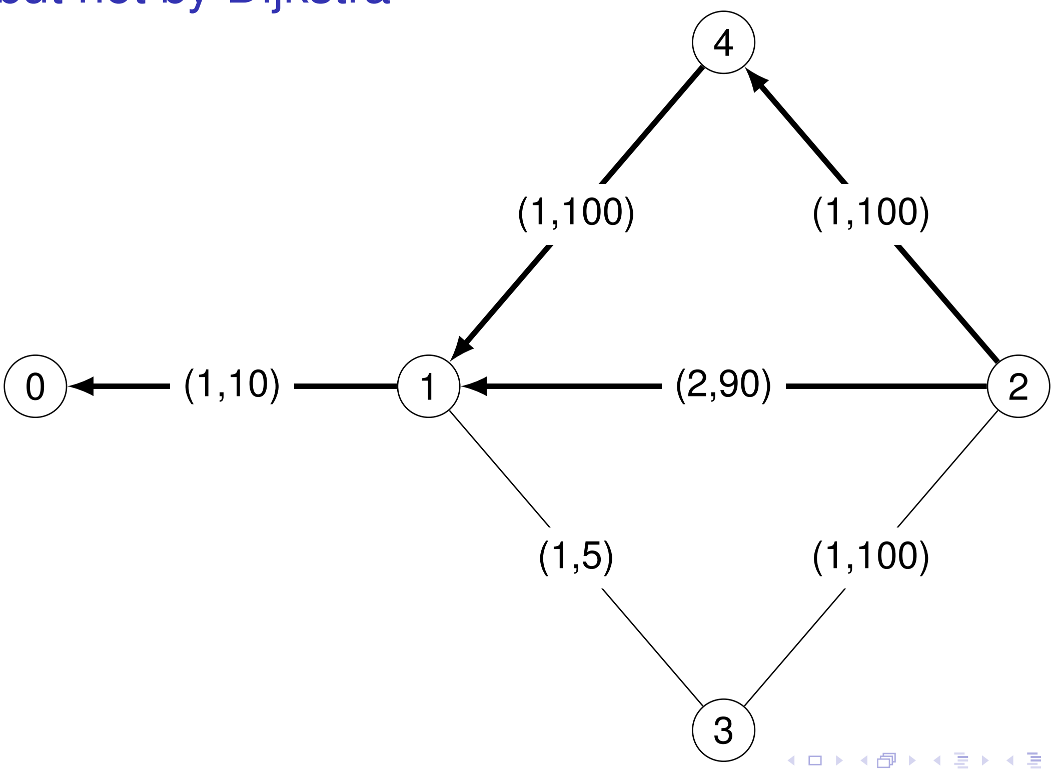
Paths computed by (extended) distributed Bellman-Ford

$$\begin{array}{lcl} \mathbf{P}_{\text{Bellman}}(0,2) & = & \{(0,1,4,2)\} \\ \mathbf{P}_{\text{Bellman}}(2,0) & = & \{(2,1,0),\ (2,4,1,0)\} \end{array}$$

Optimal paths from 0 to 2. Computed by Dijkstra but not by Bellman-Ford



Optimal paths from 2 to 1. Computed by Bellman-Ford but not by Dijkstra



Observations

For distributed Bellman-Ford

$$\begin{array}{rcl} \underline{\text{next-hop-paths}}(\mathbf{A}) & = & \underline{\text{computed-paths}}(\mathbf{A}) \\ & \subseteq & \underline{\text{optimal-paths}}(\mathbf{A}) \end{array}$$

For Dijkstra's algorithm

$$\begin{array}{rcl} \underline{\text{next-hop-paths}}(\mathbf{A}) & \subseteq & \underline{\text{computed-paths}}(\mathbf{A}) \\ & \subseteq & \underline{\text{optimal-paths}}(\mathbf{A}) \end{array}$$

How can we understand this (algebaically)?

The Algorithm to Algebra (A2A) method

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

We can capture path computation with this algebra

$$\text{sp} \vec{\times} \text{bw} \vec{\times} \text{seq}(E)$$

But this algebra is not distributive!

$$\neg\mathbb{LC}(\text{sp} \vec{\times} \text{bw})$$

$$\neg\mathbb{LK}(\text{seq}(E))$$

Towards a non-classical theory of algebraic path finding

We need theory that can accept algebras that violate distributivity.

Global optimality

$$\mathbf{A}^*(i,\ j) = \bigoplus_{p \in P(i,\ j)} w(p),$$

Left local optimality (distributed Bellman-Ford)

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

Right local optimality (Dijkstra's Algorithm)

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

Embrace the fact that all three notions can be distinct.

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 $a \otimes^T b = b \otimes a$

Dijkstra's Algorithm

Classical Dijkstra

Given adjacency matrix **A** over a selective semiring and source vertex $i \in V$, Dijkstra's algorithm will compute $\mathbf{A}^*(i,\ \_)$ such that

$$\mathbf{A}^*(i,\ j) = \bigoplus_{p \in P(i,j)} w_{\mathbf{A}}(p).$$

Non-Classical Dijkstra

If we drop assumptions of distributivity, then given adjacency matrix **A** and source vertex $i \in V$, Dijkstra's algorithm will compute $\mathbf{R}(i,\ \_)$ such that

$$\forall j \in V : \mathbf{R}(i,\ j) = \mathbf{I}(i,j) \oplus \bigoplus_{q \in V} \mathbf{R}(i,\ q) \otimes \mathbf{A}(q,\ j).$$

Routing in Equilibrium, João Luís Sobrinho and Timothy G. Griffin, MTNS 2010.

Dijkstra's algorithm

Input : adjacency matrix **A** and source vertex $i \in V$,
Output : the i-th row of **R**, $\mathbf{R}(i,\ \_)$.

begin
  $S \leftarrow \{i\}$
  $\mathbf{R}(i,\ i) \leftarrow \overline{1}$
  for each $q \in V - \{i\} : \mathbf{R}(i,\ q) \leftarrow \mathbf{A}(i,\ q)$
  while $S \neq V$
    begin
      find $q \in V - S$ such that $\mathbf{R}(i,\ q)$ is $\leqslant^L_{\oplus}$ -minimal
      $S \leftarrow S \cup \{q\}$
      for each $j \in V - S$
        $\mathbf{R}(i,\ j) \leftarrow \mathbf{R}(i,\ j) \oplus (\mathbf{R}(i,\ q) \otimes \mathbf{A}(q,\ j))$
    end
end

Classical proofs of Dijkstra's algorithm (for global optimality) assume

Semiring Axioms

$$
\begin{array}{rcrcl}
\mathbb{AS}(\oplus) & : & a \oplus (b \oplus c) & = & (a \oplus b) \oplus c \\
\mathbb{CM}(\oplus) & : & a \oplus b & = & b \oplus a \\
\mathbb{ID}(\oplus) & : & \overline{0} \oplus a & = & a \\
\mathbb{AS}(\otimes) & : & a \otimes (b \otimes c) & = & (a \otimes b) \otimes c \\
\mathbb{IDL}(\otimes) & : & \overline{1} \otimes a & = & a \\
\mathbb{IDR}(\otimes) & : & a \otimes \overline{1} & = & a \\
\mathbb{ANL}(\otimes) & : & \overline{0} \otimes a & = & \overline{0} \\
\mathbb{ANR}(\otimes) & : & a \otimes \overline{0} & = & \overline{0} \\
\mathbb{LD} & : & a \otimes (b \oplus c) & = & (a \otimes b) \oplus (a \otimes c) \\
\mathbb{RD} & : & (a \oplus b) \otimes c & = & (a \otimes c) \oplus (b \otimes c)
\end{array}
$$

Classical proofs of Dijkstra's algorithm assume

Additional axioms

$$
\begin{array}{rcrcl}
\mathbb{SL}(\oplus) & : & a \oplus b & \in & \{a,\ b\} \\
\mathbb{AN}(\oplus) & : & \overline{1} \oplus a & = & \overline{1}
\end{array}
$$

Note that we can derive right absorption,

$$\mathbb{RA} \quad : \quad a \oplus (a \otimes b) \quad = \quad a$$

and this gives (right) inflationarity, $\forall a, b : a \leqslant a \otimes b$.

$$
\begin{array}{rcl}
a \oplus (a \otimes b) & = & (a \otimes \overline{1}) \oplus (a \otimes b) \\
 & = & a \otimes (\overline{1} \oplus b) \\
 & = & a \otimes \overline{1} \\
 & = & a
\end{array}
$$

What will we assume? Very little!

~~Semiring~~ Axioms

$$
\begin{array}{rcrcl}
\mathbb{AS}(\oplus) & : & a \oplus (b \oplus c) & = & (a \oplus b) \oplus c \\
\mathbb{CM}(\oplus) & : & a \oplus b & = & b \oplus a \\
\mathbb{ID}(\oplus) & : & \overline{0} \oplus a & = & a \\
\cancel{\mathbb{AS}(\otimes)} & : & \cancel{a \otimes (b \otimes c)} & \cancel{=} & \cancel{(a \otimes b) \otimes c} \\
\mathbb{IDL}(\otimes) & : & \overline{1} \otimes a & = & a \\
\cancel{\mathbb{IDR}(\otimes)} & : & \cancel{a \otimes \overline{1}} & \cancel{=} & \cancel{a} \\
\cancel{\mathbb{ANL}(\otimes)} & : & \cancel{\overline{0} \otimes a} & \cancel{=} & \cancel{\overline{0}} \\
\cancel{\mathbb{ANR}(\otimes)} & : & \cancel{a \otimes \overline{0}} & \cancel{=} & \cancel{\overline{0}} \\
\cancel{\mathbb{LD}} & : & \cancel{a \otimes (b \oplus c)} & \cancel{=} & \cancel{(a \otimes b) \oplus (a \otimes c)} \\
\cancel{\mathbb{RD}} & : & \cancel{(a \oplus b) \otimes c} & \cancel{=} & \cancel{(a \otimes c) \oplus (b \otimes c)}
\end{array}
$$

What will we assume?

Additional axioms

$$
\begin{array}{rcrcl}
\mathbb{SL}(\oplus) & : & a \oplus b & \in & \{a,\ b\} \\
\mathbb{ANL}(\oplus) & : & \overline{1} \oplus a & = & \overline{1} \\
\mathbb{RA} & : & a \oplus (a \otimes b) & = & a
\end{array}
$$

- Note that we can no longer derive $\mathbb{RA}$, so we must assume it.
- Again, $\mathbb{RA}$ says that $a \leqslant a \otimes b$.
- We don't use $\mathbb{SL}$ explicitly in the proofs, but it is implicit in the algorithm's definition of q_k.
- We do not use $\mathbb{AS}(\oplus)$ and $\mathbb{CM}(\oplus)$ explicitly, but these assumptions are implicit in the use of the "big-$\oplus$" notation.

Under these weaker assumptions ...

Theorem (Sobrinho/Griffin)

Given adjacency matrix $\mathbf{A}$ and source vertex $i \in V$, Dijkstra's algorithm will compute $\mathbf{R}(i, \_)$ such that

$$\forall j \in V : \mathbf{R}(i, j) = \mathbf{I}(i,j) \oplus \bigoplus_{q \in V} \mathbf{R}(i, q) \otimes \mathbf{A}(q, j).$$

That is, it computes one row of the solution for the right equation

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

Dijkstra's algorithm, annotated version

Subscripts make proofs by induction easier ....

begin
  $S_1 \leftarrow \{i\}$
  $\mathbf{R}_1(i, i) \leftarrow \overline{1}$
  for each $q \in V - S_1 : \mathbf{R}_1(i, q) \leftarrow \mathbf{A}(i, q)$
  for each $k = 2, 3, \ldots, \mid V \mid$
    begin
      find $q_k \in V - S_{k-1}$ such that $\mathbf{R}_{k-1}(i, q_k)$ is $\leqslant^L_\oplus$ -minimal
      $S_k \leftarrow S_{k-1} \cup \{q_k\}$
      for each $j \in V - S_k$
        $\mathbf{R}_k(i, j) \leftarrow \mathbf{R}_{k-1}(i, j) \oplus (\mathbf{R}_{k-1}(i, q_k) \otimes \mathbf{A}(q_k, j))$
    end
end

Main Claim, annotated

$$\forall k : 1 \leqslant k \leqslant |\, V \,| \Longrightarrow \forall j \in S_k : \mathbf{R}_k(i,\ j) = \mathbf{I}(i,j) \oplus \bigoplus_{q \in S_k} \mathbf{R}_k(i,\ q) \otimes \mathbf{A}(q,\ j)$$

We will use

Observation 1 (no backtracking) :

$$\forall k : 1 \leqslant k <|\, V \,| \Longrightarrow \forall j \in S_{k+1} : \mathbf{R}_{k+1}(i,\ j) = \mathbf{R}_k(i,\ j)$$

Observation 2 (Dijkstra is “greedy”):

$$\forall k : 1 \leqslant k \leqslant |\, V \,| \Longrightarrow \forall q \in S_k : \forall w \in V - S_k : \mathbf{R}_k(i,\ q) \leqslant \mathbf{R}_k(i,\ w)$$

Observation 3 (Accurate estimates):

$$\forall k : 1 \leqslant k \leqslant |\, V \,| \Longrightarrow \forall w \in V - S_k : \mathbf{R}_k(i,\ w) = \bigoplus_{q \in S_k} \mathbf{R}_k(i,\ q) \otimes \mathbf{A}(q,\ w)$$

Observation 1

$$\forall k : 1 \leqslant k <|\, V \,| \Longrightarrow \forall j \in S_{k+1} : \mathbf{R}_{k+1}(i,\ j) = \mathbf{R}_k(i,\ j)$$

Proof: This is easy to see by inspection of the algorithm. Once a node is put into S its weight never changes again.

The algorithm is "greedy"

Observation 2

$$\forall k : 1 \leqslant k \leqslant\mid V \mid \implies \forall q \in S_k : \forall w \in V - S_k : \mathbf{R}_k(i,\ q) \leqslant \mathbf{R}_k(i,\ w)$$

By induction.
Base : Since $S_1 = \{i\}$ and $\mathbf{R}_1(i,\ i) = \overline{1}$, we need to show that

$$\overline{1} \leqslant \mathbf{A}(i,\ w) \equiv \overline{1} = \overline{1} \oplus \mathbf{A}(i,\ w).$$

This follows from $\mathbb{ANL}(\oplus)$.
Induction: Assume $\forall q \in S_k : \forall w \in V - S_k : \mathbf{R}_k(i,\ q) \leqslant \mathbf{R}_k(i,\ w)$ and show $\forall q \in S_{k+1} : \forall w \in V - S_{k+1} : \mathbf{R}_{k+1}(i,\ q) \leqslant \mathbf{R}_{k+1}(i,\ w)$.
Since $S_{k+1} = S_k \cup \{q_{k+1}\}$, this means showing

$$\begin{array}{ll} (1) & \forall q \in S_k : \forall w \in V - S_{k+1} : \mathbf{R}_{k+1}(i,\ q) \leqslant \mathbf{R}_{k+1}(i,\ w) \\ (2) & \forall w \in V - S_{k+1} : \mathbf{R}_{k+1}(i,\ q_{k+1}) \leqslant \mathbf{R}_{k+1}(i,\ w) \end{array}$$

By Observation 1, showing (1) is the same as

$$\forall q \in S_k : \forall w \in V - S_{k+1} : \mathbf{R}_k(i,\ q) \leqslant \mathbf{R}_{k+1}(i,\ w)$$

which expands to (by definition of $\mathbf{R}_{k+1}(i,\ w)$)

$$\forall q \in S_k : \forall w \in V - S_{k+1} : \mathbf{R}_k(i,\ q) \leqslant \mathbf{R}_k(i,\ w) \oplus (\mathbf{R}_k(i,\ q_{k+1}) \otimes \mathbf{A}(q_{k+1},\ w))$$

But $\mathbf{R}_k(i,\ q) \leqslant \mathbf{R}_k(i,\ w)$ by the induction hypothesis, and
$\mathbf{R}_k(i,\ q) \leqslant (\mathbf{R}_k(i,\ q_{k+1}) \otimes \mathbf{A}(q_{k+1},\ w))$ by the induction hypothesis and $\mathbb{RA}$.
Since $a \leqslant^L_\oplus b \wedge a \leqslant^L_\oplus c \implies a \leqslant^L_\oplus (b \oplus c)$, we are done.

By Observation 1, showing (2) is the same as showing

$$\forall w \in V - S_{k+1} : \mathbf{R}_k(i,\ q_{k+1}) \leqslant \mathbf{R}_{k+1}(i,\ w)$$

which expands to

$$\forall w \in V - S_{k+1} : \mathbf{R}_k(i,\ q_{k+1}) \leqslant \mathbf{R}_k(i,\ w) \oplus (\mathbf{R}_k(i,\ q_{k+1}) \otimes \mathbf{A}(q_{k+1},\ w))$$

But $\mathbf{R}_k(i,\ q_{k+1}) \leqslant \mathbf{R}_k(i,\ w)$ since q_{k+1} was chosen to be minimal, and $\mathbf{R}_k(i,\ q_{k+1}) \leqslant (\mathbf{R}_k(i,\ q_{k+1}) \otimes \mathbf{A}(q_{k+1},\ w))$ by $\mathbb{RA}$.
Since $a \leqslant^L_{\oplus} b \wedge a \leqslant^L_{\oplus} c \implies a \leqslant^L_{\oplus} (b \oplus c)$, we are done.

Observation 3

Observation 3

$$\forall k : 1 \leqslant k \leqslant \mid V \mid \implies \forall w \in V - S_k : \mathbf{R}_k(i,\ w) = \bigoplus_{q \in S_k} \mathbf{R}_k(i,\ q) \otimes \mathbf{A}(q,\ w)$$

Proof: By induction:
Base : easy, since

$$\bigoplus_{q \in S_1} \mathbf{R}_1(i,\ q) \otimes \mathbf{A}(q,\ w) = \overline{1} \otimes \mathbf{A}(i,\ w) = \mathbf{A}(i,\ w) = \mathbf{R}_1(i,\ w)$$

Induction step. Assume

$$\forall w \in V - S_k : \mathbf{R}_k(i,\ w) = \bigoplus_{q \in S_k} \mathbf{R}_k(i,\ q) \otimes \mathbf{A}(q,\ w)$$

and show

$$\forall w \in V - S_{k+1} : \mathbf{R}_{k+1}(i,\ w) = \bigoplus_{q \in S_{k+1}} \mathbf{R}_{k+1}(i,\ q) \otimes \mathbf{A}(q,\ w)$$

By Observation 1, and a bit of rewriting, this means we must show

$$\forall w \in V - S_{k+1} : \mathbf{R}_{k+1}(i,\ w) = \mathbf{R}_k(i,\ q_{k+1}) \otimes \mathbf{A}(q_{k+1},\ w) \oplus \bigoplus_{q \in S_k} \mathbf{R}_k(i,\ q) \otimes \mathbf{A}(q$$

Using the induction hypothesis, this becomes

$$\forall w \in V - S_{k+1} : \mathbf{R}_{k+1}(i,\ w) = \mathbf{R}_k(i,\ q_{k+1}) \otimes \mathbf{A}(q_{k+1},\ w) \oplus \mathbf{R}_k(i,\ w)$$

But this is exactly how $\mathbf{R}_{k+1}(i,\ w)$ is computed in the algorithm.

Proof of Main Claim

Main Claim

$$\forall k : 1 \leqslant k \leqslant \mid V \mid \Longrightarrow \forall j \in S_k : \mathbf{R}_k(i,\ j) = \mathbf{I}(i, j) \oplus \bigoplus_{q \in S_k} \mathbf{R}_k(i,\ q) \otimes \mathbf{A}(q,\ j)$$

Proof : By induction on k.
Base case: $S_1 = \{i\}$ and the claim is easy.
Induction: Assume that

$$\forall j \in S_k : \mathbf{R}_k(i,\ j) = \mathbf{I}(i, j) \oplus \bigoplus_{q \in S_k} \mathbf{R}_k(i,\ q) \otimes \mathbf{A}(q,\ j)$$

We must show that

$$\forall j \in S_{k+1} : \mathbf{R}_{k+1}(i,\ j) = \mathbf{I}(i, j) \oplus \bigoplus_{q \in S_{k+1}} \mathbf{R}_{k+1}(i,\ q) \otimes \mathbf{A}(q,\ j)$$

Since $S_{k+1} = S_k \cup \{q_{k+1}\}$, this means we must show

$$(1) \quad \forall j \in S_k : \mathbf{R}_{k+1}(i,\ j) = \mathbf{I}(i,j) \oplus \bigoplus_{q \in S_{k+1}} \mathbf{R}_{k+1}(i,\ q) \otimes \mathbf{A}(q,\ j)$$
$$(2) \quad \mathbf{R}_{k+1}(i,\ q_{k+1}) = \mathbf{I}(i,q_{k+1}) \oplus \bigoplus_{q \in S_{k+1}} \mathbf{R}_{k+1}(i,\ q) \otimes \mathbf{A}(q,\ q_{k+1})$$

By use Observation 1, showing (1) is the same as showing

$$\forall j \in S_k : \mathbf{R}_k(i,\ j) = \mathbf{I}(i,j) \oplus \bigoplus_{q \in S_{k+1}} \mathbf{R}_k(i,\ q) \otimes \mathbf{A}(q,\ j),$$

which is equivalent to

$$\forall j \in S_k : \mathbf{R}_k(i,\ j) = \mathbf{I}(i,j) \oplus (\mathbf{R}_k(i,\ q_{k+1}) \otimes \mathbf{A}(q_{k+1},\ j)) \oplus \bigoplus_{q \in S_k} \mathbf{R}_k(i,\ q) \otimes \mathbf{A}(q,\ j)$$

By the induction hypothesis, this is equivalent to

$$\forall j \in S_k : \mathbf{R}_k(i,\ j) = \mathbf{R}_k(i,\ j) \oplus (\mathbf{R}_k(i,\ q_{k+1}) \otimes \mathbf{A}(q_{k+1},\ j)),$$

Put another way,

$$\forall j \in S_k : \mathbf{R}_k(i,\ j) \leqslant \mathbf{R}_k(i,\ q_{k+1}) \otimes \mathbf{A}(q_{k+1},\ j)$$

By observation 2 we know $\mathbf{R}_k(i,\ j) \leqslant \mathbf{R}_k(i,\ q_{k+1})$, and so

$$\mathbf{R}_k(i,\ j) \leqslant \mathbf{R}_k(i,\ q_{k+1}) \leqslant \mathbf{R}_k(i,\ q_{k+1}) \otimes \mathbf{A}(q_{k+1},\ j)$$

by $\mathbb{RA}$.

To show (2), we use Observation 1 and $\mathbf{I}(i, q_{k+1}) = \overline{0}$ to obtain

$$\mathbf{R}_k(i,\ q_{k+1}) = \bigoplus_{q \in S_{k+1}} \mathbf{R}_k(i,\ q) \otimes \mathbf{A}(q,\ q_{k+1})$$

which, since $\mathbf{A}(q_{k+1},\ q_{k+1}) = \overline{0}$, is the same as

$$\mathbf{R}_k(i,\ q_{k+1}) = \bigoplus_{q \in S_k} \mathbf{R}_k(i,\ q) \otimes \mathbf{A}(q,\ q_{k+1})$$

This then follows directly from Observation 3.

Finding Left Local Solutions?

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

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

where

$$a \otimes^T b = b \otimes a$$

Replace $\mathbb{RA}$ with $\mathbb{LA}$,

$$\mathbb{LA} : \forall a, b : a \leqslant b \otimes a$$