NOTES ON "GRAPH NEURAL NETWORKS ARE DYNAMIC PROGRAMMERS"

IORDAN GANEV

1. Introduction

These are notes on the technical aspects of the paper "Graph Neural Networks are Dynamic Programmers" [DV22]. We formulate integral transforms using bags and lists. The update step of a dynamic programming algorithm can be interpreted as an integral transform; a prominent example is the Bellman–Ford algorithm. Furthermore, the message-passing step of a graph neural network can also be stated in terms of a certain integral transform.

2. Bags and lists

2.1. **Bags.** Let R a set. We denote by $\texttt{bag}(R)$ the free commutative monoid on R. Equivalently, $\texttt{bag}(R)$ is the set of all finite formal linear combinations of elements of r with coefficients in the natural numbers $\mathbb{N} = \{0, 1, 2, \dots\}$. With this interpretation in mind, we write an element of $\texttt{bag}(R)$ as a formal sum: $\sum_{r \in R} n_r r$, where only finitely many of the coefficients $n_r \in \mathbb{N}$ are nonzero. Yet another characterization of $\texttt{bag}(R)$ is as the set of functions $R \to \mathbb{N}$ where only finitely many elements of R map to a non-zero natural number; that is, the set of finitely supported functions. The zero element of $\texttt{bag}(R)$ is the empty bag, which corresponds to the constantly zero function. As an endofunctor of the category of sets, $\texttt{bag}$ is a monad under a form of concatentation, with singleton bags forming the unit. Algebras for $\texttt{bag}$ are commutative monoids in the category of sets.

2.2. **Lists.** We denote by $\texttt{list}(R)$ the free monoid on R. As a set, $\texttt{list}(R) = 1 + R + R^2 + \dots$, where '+' denotes disjoint union, $R^n = R \times R \times \cdots \times R$ is the n-fold product of R with itself, and 1 is the one-element set corresponding to the empty list [] (which is also the unit of $\texttt{list}(R)$). We write an element of $\texttt{list}(R)$ as $[r_1, r_2, \dots, r_n]$ where the r_i are not necessarily distinct elements of R. As an endofunctor of the category of sets, $\texttt{list}$ is a monad under concatentation, with the unit given by singleton lists. Algebras for $\texttt{list}$ are monoids in the category of sets.

2.3. **Bag and lists combined.** There is a natural transformation of monads $\texttt{list} \to \texttt{bag}$ which counts up the multiplicities of elements in a list. Pulling back algebras along this natural transformation of monads forgets the commutative structure on a commutative monoid. There is also a natural transformation:

$$\lambda : \texttt{list} \circ \texttt{bag} \to \texttt{bag} \circ \texttt{list}$$

defined as taking a list of bags $\left[\sum n_r^{(1)} r, \sum n_r^{(2)} r, \ldots, \sum n_r^{(\ell)} r\right]$ to the bag of lists where the coefficient of a list $[r_1, \ldots, r_\ell]$ is the product $\prod_{i=1}^{\ell} n_{r_i}^{(i)}$, that is, the product over $i = 1, \ldots, \ell$ of the number of times $n_{r_i}^{(i)}$ that the element r_i appears in the i-th coordinate of the list of bags. In this notation, lists of length different from ℓ have coefficient zero.

Remark 2.1. There is no inverse map $\texttt{bag} \circ \texttt{list} \to \texttt{list} \circ \texttt{bag}$ in general.

2.4. **Semirings.** The natural transformation λ induces a natural transformation:

$$M : \texttt{list} \circ \texttt{bag} \circ \texttt{list} \circ \texttt{bag} \longrightarrow \texttt{list} \circ \texttt{bag}$$

where one applies λ to the middle copy of $\texttt{bag} \circ \texttt{list}$, followed by the monadic operation on each of `list` and on `bag`. The natural transformation M defines a monad structure on $\texttt{list} \circ \texttt{bag}$, with unit given by singleton lists of singleton bags. Algebras for the monad $\texttt{list} \circ \texttt{bag}$ are precisely semirings[1]. Examples of semirings include:

- Any ring is a semiring.
- There are two semiring structures on the set $\{0, 1\}$:

$$(\{0,1\}\ ,\ \mathsf{OR}\ ,\ \mathsf{AND}), \qquad (\{0,1\}\ ,\ \mathsf{XOR}\ ,\ \mathsf{AND}),$$

  where we list the 'addition' operation first and the 'multiplication' operation second. The XOR version is isomorphic to the field with two elements, $\mathbb{F}_2$.
- If R is a semiring, then the set of n by n matrices $\mathrm{Mat}_n(R)$ with entries in R is also a semiring.
- The semiring of tropical real numbers is defined as $(\mathbb{R} \cup \{\infty\}, \min, +)$. Here the 'addition' operation is min with unit ∞, while the 'multiplication' operation is $+$ with unit 0. The distributive law is: $a + \min(b, c) = \min(a + b, a + c)$. (Although both operations are commutative, note that $\min(a, b + c) \neq \min(a, b) + \min(a, c)$ in general.)
- Similary, we have the dual (or max) tropical real numbers: $(\mathbb{R} \cup \{-\infty\}, \max, +)$.
- The power set $\mathcal{P}(X)$ of a set is a semiring under intersection and union, which are both commutative and play symmetric roles since both distributive laws hold:

$$A \cap (B \cup C) = (A \cup B) \cap (A \cup C) \qquad A \cup (B \cap C) = (A \cap B) \cup (A \cap C)$$

  where A, B, and C are subsets of X. The unit of union is X and the unit of intersection is the empty set $\emptyset$.

[1]Recall that a semiring is a ring without the assumption of additive inverses.

3. Pushforwards

3.1. **Bag pushforward.** Let $f : Y \to Z$ be a map between finite sets. Given a function $\alpha : Y \to R$ to a (possibly infinite) set R, we have the 'bag pushforward':

$$f_*\alpha : Z \to \mathtt{bag}(R), \qquad\qquad z \mapsto \sum_{y \in f^{-1}(z)} \alpha(y)$$

where we regard $\alpha(y)$ as an element of $\mathtt{bag}(R)$ via the canonical embedding $R \hookrightarrow \mathtt{bag}(R)$. Note that the coefficient of r in $f_*\alpha(z)$ is the cardinality of the intersection $\alpha^{-1}(r) \cap f^{-1}(z)$ in Y. Writing $[-,-]$ for function spaces, we have $f_* : [Y, R] \to [Z, \mathtt{bag}(R)]$.

3.2. **List pushforward.** Let $f : X \to Y$ be a map between finite sets. Suppose $(X, \leq)$ is a total order on X. Such an order induces a map:

$$\mathsf{ord} : \mathcal{P}(X) \to \mathtt{list}(X)$$

that orders the elements of any subset of X. Given a function $\alpha : X \to R$ to a (possibly infinite) set R, we have the 'list pushforward':

$$f_*^{(\mathtt{list})}\alpha : Y \to \mathtt{list}(R), \qquad y \mapsto \mathtt{list}(\alpha) \circ \mathsf{ord}\left(f^{-1}(y)\right)$$

In other words, we apply α element-wise to the fiber over y, regarded as an ordered list using the total order on X. Thus, we have a map $f_*^{(\mathtt{list})} : [X, R] \to [Y, \mathtt{list}(R)]$. Note that applying the forgetful map $\mathsf{forget} : \mathtt{list} \to \mathtt{bag}$ allows us to recover the bag pushforward from the list pushforward:

$$\mathsf{forget}_R \circ f_*^{(\mathtt{list})}\alpha = f_*\alpha$$

3.3. **Integral transform.** Now consider a diagram of finite sets:

$$W \xleftarrow{h} X \xrightarrow{f} Y \xrightarrow{g} Z.$$

Call this diagram D, and fix a total order $(X, \leq)$ on X. Pulling back along h, applying the list pushforward along f, and them applying the bag pushforward along g yields the integral transform determined by D:

$$\mathcal{F}_D = g_* \circ f_*^{(\mathtt{list})} \circ h^* : [W, R] \longrightarrow [Z, \mathtt{bag} \circ \mathtt{list}(R)]$$

Explicitly, given a function $\alpha : W \to R$ and an element $z \in Z$, the coefficient of the list $[r_1, \ldots, r_n]$ in $\mathcal{F}_D(\alpha)(z)$ is the number of times we have an equality:

$$[r_1, \ldots, r_n] = \left[\alpha \circ h\left(x_1^{(y)}\right), \ldots, \alpha \circ h\left(x_n^{(y)}\right)\right]$$

as y runs over the fiber $g^{-1}(z)$ over z and $\left[x_1^{(y)}, \ldots, x_n^{(y)}\right] = \mathsf{ord}(f^{-1}(y))$ is the ordering of the fiber of f over y.

4. Directed graphs

Throughout these notes, we let $G = (V, E)$ be a connected directed graph, and let $s, t : E \to V$ be the source and target maps. We implicitly choose a total order on each of V and E. This gives us a total order on any ordered coproduct of these sets. As far as the author of these notes can tell, the details of this total order is not particularly relevant for the topics at hand, except for the implementation of the algorithms in code. Unless specified otherwise, we assume there are no loops (edges with $s(e) = t(e)$) and there are no double edges (distinct edges e_1 and e_2 with $s(e_1) = s(e_2)$ and $t(e_1) = t(e_2)$). We will comment at various points on loosening these assumptions, or replacing them with other assumptions.

5. Bellman–Ford

5.1. **Algorithm.** The Bellman–Ford algorithm takes a weighted directed graph with a designated source vertex, and computes the lowest-weight paths from the source vertex to each of the other vertices. Specifically, let $w : E \to \mathbb{R}$ be the edge weights, which can be negative[2]. Let v_0 be the designated source vertex and initialize a distance function $d_0 : V \to \mathbb{R} \cup \{\infty\}$ as:

$$d_0(v) = \begin{cases} 0 & \text{if } v = v_0 \\ \infty & \text{otherwise} \end{cases}$$

We update the distance function by repeatedly running over all vertices and applying the following rule at each vertex:

$$d_{t+1}(v) = \min(d_t(v)\ ,\ \min_{e:u\to v}(d_t(u) + w(e))) \tag{5.1}$$

We implicitly choose a total order on the set of vertices in order to loop over all vertices. Using the observation that any simple path has length at most $|V| - 1$, one can show that the algorithm converges after at most $|V| - 1$ iterations over all the vertices. At a given vertex v, we perform $O(|t^{-1}(v)|)$ operations; since $\sum_v |t^{-1}(v)| = |E|$, each iteration across all vertices has time complexity $O(|E|)$. We conclude that the time complexity of the full algorithm is $O(|V||E|)$.

5.2. **Integral transform.** One can realize the Bellman–Ford update rule in Equation 5.1 as an integral transform using the diagram below:

$$D = \begin{tikzcd} V + E + E \arrow[r, "\mathrm{id}+\langle \mathrm{id},\mathrm{id}\rangle"] \arrow[d, "\langle \mathrm{id},s\rangle+\mathrm{id}"'] & V + E \arrow[d, "\langle \mathrm{id},t\rangle"] \\ V + E & V \end{tikzcd}$$

[2]Dijkstra's algorithm is faster than Bellman–Ford when all weights are non-negative.

To see this, first observe that a function $\alpha = \langle d, w \rangle : V + E \to R$ consists of a function $d : V \to R$ on the vertices and a function $w : E \to R$ on the edges. The integral transform of $\alpha = \langle d, w \rangle$, along this diagram is:

$$\begin{aligned} \mathcal{F}_D(d,w) : V &\to \mathtt{bag} \circ \mathtt{list}(R) \\ v &\mapsto [d(v)] + \sum_{e:u\to v} [d(u), w(e)] \end{aligned}$$

where the sum is over all vertices u with an edge to v.

Now we specialize our coefficients to the semiring of tropical real numbers $R = (\mathbb{R} \cup \{\infty\}, \min, +)$. Let $\mu : \mathtt{bag} \circ \mathtt{list}(R) \to R$ be the map induced by the semiring structure, so that we take sums of lists and minima of bags. Then we have:

$$\mu \circ \mathcal{F}_D(d,w)(v) = \min(d(v)\ ,\ \min_{e:u\to v}(d(u) + w(e)))$$

for any function $d : V \to \mathbb{R}$. In this way, we recover the Bellman–Ford update rule of Equation 5.1 as an integral transform.

5.3. **Vertex costs.** Suppose we have a cost $c : V \to R$ for visiting each vertex. The variation of the Bellman–Ford algorithm that takes these costs into account has update rule:

$$d_{t+1}(v) = \min(d_t(v)\ ,\ c(v) + \min_{e:u\to v}(d_t(u) + w(e))) \tag{5.2}$$

We can realize this update rule via the following diagram:

$$D = \begin{array}{ccc} V + E + E + E & \xrightarrow{\mathrm{id}+\langle \mathrm{id},\mathrm{id},\mathrm{id}\rangle} & V + E \\ {\scriptstyle \langle \mathrm{id},s\rangle + t + \mathrm{id}}\downarrow & & \downarrow{\scriptstyle \langle \mathrm{id},t\rangle} \\ V + V + E & & V \end{array}$$

A function $\langle d, c, w \rangle : V + V + E \to R$ consists of two functions $d, c : V \to R$ on the vertices and a function $w : E \to R$ on the edges. The integral transform of $\langle d, c, w \rangle$ along this diagram is:

$$\begin{aligned} \mathcal{F}_D(d,c,w) : V &\to \mathtt{bag} \circ \mathtt{list}(R) \\ v &\mapsto [d(v)] + \sum_{e:u\to v} [d(u), c(v), w(e)] \end{aligned}$$

where the sum is over all vertices u with an edge to v. Taking R to be the tropical real numbers and using the distributive law, we recover the update rule in Equation 5.2:

$$\mu \circ \mathcal{F}_D(d,c,w) = \min(d(v)\ ,\ c(v) + \min_{e:u\to v}(d(u) + w(e)))$$

5.4. **Variation.** Consider the following diagram, which appears in the original paper [DV22]:

$$D = \quad \begin{array}{ccc} V+E+V+E & \xrightarrow{\langle \mathrm{id},\mathrm{id}\rangle} & V+E \\ {\scriptstyle \langle \mathrm{id}, s\rangle + \mathrm{id}}\big\downarrow & & \big\downarrow{\scriptstyle \langle \mathrm{id}, t\rangle} \\ V+V+E & & V \end{array}$$

A function $\langle d, b, w\rangle : V + V + E \to R$ consists of two functions $d, b : V \to R$ on the vertices and a function $w : E \to R$ on the edges. The integral transform of $\langle d, b, w\rangle$ along this diagram is:

$$\begin{aligned} \mathcal{F}_D(d, b, w) : V &\to \mathtt{bag} \circ \mathtt{list}(R) \\ v &\mapsto [d(v), b(v)] + \sum_{e : u \to v} [d(u), w(e)] \end{aligned}$$

where the sum is over all vertices u with an edge to v. Taking R to be the tropical real numbers, we arrive at an update rule:

$$\mu \circ \mathcal{F}_D(d, b, w)(v) = \min(d(v) + b(v)\ ,\ \min_{e:u\to v}(d(u) + w(e)))$$

If b is constantly zero, this update rule matches the original one in Equation 5.1. The interpretation of this update rule for $b \neq 0$ eludes the author of these notes.

6. Knapsack problem

Let $X = \{1, \ldots, n\}$ be a list of items, each of which has a value v_i and a weight w_i. The knapsack problem seeks to select items with maximum total value subject to a constraint on the total weight. More precisely, we seek a maximum of the function:

$$F : \mathcal{P}(X) \to \mathbb{R}, \qquad S \mapsto \sum_{i \in S} v_i$$

subject to $\sum_{i\in S} w_i \leq W_{\max}$, where $W_{\max}$ is the maximum weight capacity of the knapsack. The usual dynamic programming approach involves setting $M(i, r)$ to be the maximum total value of a subset of $\{1, \ldots, i\}$ subject to the condition that the total weight of the subset is less than or equal to r. In symbols, $M(i, r)$ is equal to:

$$\max_{S \in \mathcal{P}(\{1,\ldots,i\})} \left(\sum_{j\in S} v_j \right) \qquad \text{subject to } \sum_{j \in S} w_j \leq r$$

Using the fact that the empty sum has value 0, we initialize the $i = 0$ case as:

$$M(0, r) = \begin{cases} -\infty & \text{if } r < 0 \\ 0 & \text{if } r \geq 0 \end{cases}$$

where the motivation for the choice of $-\infty$ will become clear below. The recurrence is given by:

$$M(i, r) = \max\big(\ M(i-1, r)\ ,\ M(i-1, r - w_i) + v_i\ \big) \tag{6.1}$$

Indeed, given a subset of $\{1, \ldots, i-1\}$, we can either consider it as a subset of $\{1, \ldots, i\}$ without changing the weight, or we add i to the subset. The second option requires decreasing the total weight bound by w_i and increasing the value by v_i. The value of the option with higher value becomes $M(i,r)$.

To realize the recurrence in Equation 6.1 as an integral transform, consider the following augment and skip maps:

$$\begin{aligned} \mathsf{aug} : X \times \mathbb{R} \to X \times \mathbb{R}, \qquad & (i,r) \mapsto (i+1\ ,\ r + w_{i+1}) \\ \mathsf{skip} : X \times \mathbb{R} \to X \times \mathbb{R}, \qquad & (i,r) \mapsto (i+1\ ,\ r) \end{aligned}$$

The first corresponds to augmenting a subset of $\{1,\ldots,i\}$ of weight $\leq r$ by the element $i+1$ at the cost of increasing the upper bound on the weight by w_{i+1}. The second corresponds to considering a subset of $\{1,\ldots,i\}$ of weight $\leq r$ as a subset of $\{1,\ldots,i+1\}$, thus leaving the weight unchanged. Now consider the following diagram:

$$D = \begin{tikzcd} X \times \mathbb{R} + X \times \mathbb{R} + X \times \mathbb{R} \arrow[r, "\langle \mathsf{skip},\mathsf{aug}\rangle + \mathrm{id}"] \arrow[d, "\langle \mathrm{id},\mathrm{id}\rangle + \pi_1"'] & X \times \mathbb{R} + X \times \mathbb{R} \arrow[d, "\langle \mathrm{id},\mathrm{id}\rangle"] \\ X \times \mathbb{R} + X & X \times \mathbb{R} \end{tikzcd}$$

where $\pi_1 : X \times \mathbb{R} \to X$ is the projection. Let R be a set and let $\alpha : X \times \mathbb{R} \to R$ and $\beta : X \to R$ be functions. The integral transform of $\langle \alpha, \beta \rangle : X \times \mathbb{R} + X \to R$ along this diagram is:

$$\begin{aligned} \mathcal{F}_D(\alpha,\beta) : X \times \mathbb{R} &\to \mathtt{bag} \circ \mathtt{list}(R) \\ (i,r) &\mapsto [\ \alpha(i-1,r)\ ]\ +\ [\ \alpha(i-1, r-w_i)\ ,\ \beta(i)\ ] \end{aligned}$$

Finally, let $R = (\mathbb{R} \cup \{-\infty\}, \max, +)$ be the semiring of (max) tropical numbers, and let $\mu : \mathtt{bag} \circ \mathtt{list}(R) \to R$ be the map induced by the semiring structure, so that we take sums of lists and maxima of bags. Observe that the initialization $M(0,-)$ of M is defined in R. Taking $\mathcal{V} : X \to \mathbb{R}$ to be the value function $i \mapsto v_i$, we have:

$$\mu \circ \mathcal{F}_D(M, \mathcal{V})(i,r) = \max(\ M(i,r)\ ,\ M(i-1, r-w_i) + v_i\ )$$

In this way, we recover the Knapsack recurrence from Equation 6.1 as an integral transform.

7. Graph neural networks: easy version

Consider the following diagram:

$$D = \begin{tikzcd} V + E \arrow[r, "\mathrm{id}"] \arrow[d, "\langle \mathrm{id}, s\rangle"'] & V + E \arrow[d, "\langle \mathrm{id}, t\rangle"] \\ V & V \end{tikzcd}$$

Note that this diagram is obtained from one appearing in relation to the Bellman–Ford algorithm by dropping an extra copy of E on the left side; this reflects the fact that (in this easy version at least) we perform graph convolution on unweighted graphs. The integral transform of a function $\alpha = \alpha_V : V \to R$, along this new diagram is given by:

$$\begin{aligned} \mathcal{F}_D(\alpha) : V &\to \texttt{bag} \circ \texttt{list}(R) \\ v &\mapsto [\alpha(v)] + \sum_{u \to v} [\alpha(u)] \end{aligned}$$

where the sum is over all vertices u with an edge to v. Since all lists are length one, the integral transform factors through $\texttt{bag}(R)$.

Set $R = \mathbb{R}^n$ and $S = \mathbb{R}^m$ to be finite-dimensional vector spaces, and let $K : \mathbb{R}^n \to \mathbb{R}^m$ be a linear map, i.e., a matrix. Recall that any vector space is a commutative group under addition, so we have a sum operation $\texttt{sum} : \texttt{bag}(\mathbb{R}^m) \to \mathbb{R}^m$. Applying $\texttt{bag}(K)$ to $\mathcal{F}_D(\alpha)$ and taking sums we obtain:

$$\texttt{sum} \circ \texttt{bag}(K) \circ \mathcal{F}_D(\alpha)(v) = K(\alpha(v)) + \sum_{u \to v} K(\alpha(u)) \in \mathbb{R}^m \tag{7.1}$$

To be clear, in this last expression, the sum is not a formal sum, but an actual sum in the vector space $\mathbb{R}^m$.

Remark 7.1. If one prefers the convolution in Equation 7.1 not to include $\alpha(v)$, then the simpler diagram $V \xleftarrow{s} E \xrightarrow{\mathrm{id}} E \xrightarrow{t} V$ works.

Remark 7.2. Recall that we assume by default that G has no edge loops and no double edges. This guarantees that Equation 7.1 not to include any $\alpha(u)$ more than once. Also recall that a graph G is reflexive if for every vertex there is an edge e with $s(e) = t(e) = v$. This is the extreme opposite of the no-edge-loop assumption. If this is the case, then one can use the simpler diagram $V \xleftarrow{s} E \xrightarrow{\mathrm{id}} E \xrightarrow{t} V$ to recover Equation 7.1.

8. Graph convolution: message passing

8.1. **Message passing.** We now consider a more sophisticated version of graph convolution. Fix the following two functions:

- Let $\psi : R \times R \to R$ be the message function.
- Let $\phi : R \times R \to R$ be the read-out function.
- Let $\oplus : R \times R \to R$ be a commutative monoid structure on R, with unit $0 \in R$.

The message-passing transformation of $f : V \to R$ using ϕ and ψ is given by:

$$\Gamma_{\phi,\psi}(f) : V \to R, \qquad v \mapsto \psi\left(f(v), \bigoplus_{u \to v} \phi\left(f(u), f(v)\right)\right)$$

Remark 8.1. In practice, we will take $R = \mathbb{R}^k$ as a commutative group under addition.

In order to present message passing as an integral transform, we require two constructions using the functions ϕ and ψ. We provide these constructions in a somewhat general setting before returning to ϕ and ψ.

8.2. **Binary operations.** Let $\kappa : R^2 \to R$ be any function. Select a distinguished element[3] $p \in R$. For $i = 0, 1, \ldots$, we define a map:

$$\kappa_i : R^i \to R$$

as follows. The map κ_0 takes the single element of $R^0 = 1$ to $p \in R$. The map κ_1 is the identity. For $i \geq 2$, we have $\kappa_i([r_1, r_2, \ldots]) = \kappa(a_1, \kappa_{i-1}([r_2, \ldots]))$. Note that $\kappa_2 = \kappa$. Taking the coproduct of the maps κ_i we obtain a map:

$$\tilde{\kappa} : \mathtt{list}(R) \to \coprod_{i=0}^{\infty} R$$

Note that we can compose with the fold map $\coprod_{i=0}^{\infty} R \to R$ to obtain:

$$\text{fold} \circ \tilde{\kappa} : \mathtt{list}(R) \to R$$

We can also apply $\mathtt{bag}(\tilde{\kappa})$, which turns bags of lists into lists of bags:

$$\mathtt{bag}(\tilde{\kappa}) : \mathtt{bag} \circ \mathtt{list}(R) \to \mathtt{bag}\left(\coprod_{i=0}^{\infty} R\right) \simeq \prod_{i=0}^{\infty} \mathtt{bag}(R) = \mathtt{list} \circ \mathtt{bag}(R)$$

Returning to the functions $\phi, \psi : R^2 \to R$, we can combine them to produce a map:

$$A_{\phi,\psi} : \mathtt{bag} \circ \mathtt{list}(R) \to R$$

defined as the following composition, which also invokes the commutative monoid structure $\oplus$ on R:

$$\mathtt{bag} \circ \mathtt{list}(R) \xrightarrow{\mathtt{bag}(\tilde{\phi})} \mathtt{list} \circ \mathtt{bag}(R) \xrightarrow{\mathtt{list}(\oplus)} \mathtt{list}(R) \xrightarrow{\text{fold} \circ \tilde{\psi}} R.$$

8.3. **The integral transform.** Consider the following diagram:

$$D = \begin{array}{ccc} V + E + E & \xrightarrow{\text{id} + \langle \text{id}, \text{id} \rangle} & V + E \\ {\scriptstyle \langle \text{id}, s, t \rangle}\big\downarrow & & \big\downarrow{\scriptstyle \langle \text{id}, t \rangle} \\ V & & V \end{array}$$

Note that this diagram is obtained from the diagram relevant to the Bellman–Ford algorithm by composing the left leg with $\langle \text{id}, t \rangle : V + E \to V$. The integral transform of $\alpha : V \to R$ along D is:

$$\begin{aligned} \mathcal{F}_D(\alpha) : V &\to \mathtt{bag} \circ \mathtt{list}(R) \\ v &\mapsto [\alpha(v)] + \sum_{u \to v} [\alpha(u), \alpha(v)] \end{aligned}$$

where the sum is over all vertices u with an edge to v

Lemma 8.2. *Message passing is the composition of the integral transform along D with the map $A_{\phi,\psi}$:*

$$A_{\phi,\psi} \circ \mathcal{F}_D = \Gamma_{\phi,\psi}$$

[3] In practice, R will have a zero element under a commutative operation and we will take $p = 0$.

Sketch of proof. Applying $\texttt{bag}(\tilde{\phi})$ to $\mathcal{F}_D(\alpha)$, we obtain

$$[\alpha(v), \sum_{u\to v} \phi(\alpha(u), \alpha(v))] \in \texttt{list} \circ \texttt{bag}(R).$$

Applying $\texttt{list}(\oplus)$, we obtain

$$[\alpha(v), \bigoplus_{u\to v} \phi(\alpha(u), \alpha(v))] \in \texttt{list}(R).$$

Finally, applying fold $\circ\, \tilde{\psi}$, we obtain the desired result:

$$\psi\left(\alpha(v), \bigoplus_{u\to v} \phi(\alpha(u), \alpha(v))\right) \in R.$$

$\square$

9. Additional integral transforms

In this section, we collect a number of further diagrams from [DV22] compute the corresponding integral transforms. We continue to work with a directed graph $G = (V, E)$.

9.1. Consider the following diagram:

$$D = \begin{array}{ccc} E+E+E+E & \xrightarrow{\langle \mathrm{id},\mathrm{id},\mathrm{id},\mathrm{id}\rangle} & E \\ {\scriptstyle 1+\langle s,t\rangle+\mathrm{id}}\big\downarrow & & \big\downarrow{\scriptstyle t} \\ 1+V+E & & V \end{array}$$

A function $\alpha = \langle \alpha_1, \alpha_V, \alpha_E\rangle : 1+V+E \to R$, consists of a constant α_1, a function $\alpha_V : V \to R$ on the vertices, and a function $\alpha_E : V \to R$ on the edges. The integral transform of α along this diagram is:

$$\begin{aligned} \mathcal{F}_D(\alpha) : V &\to \texttt{bag} \circ \texttt{list}(R) \\ v &\mapsto \sum_{e:u\to v} [\alpha_1\ ,\ \alpha_V(u)\ ,\ \alpha_V(v)\ ,\ \alpha_E(e)] \end{aligned}$$

where the sum is over all incoming edges e at v.

9.2. Consider the following diagram:

$$D = \begin{array}{ccc} 4V^2+7V^3 & \xrightarrow{\text{fold}} & V^2+V^3 \\ \big\downarrow & & \big\downarrow{\scriptstyle \pi_2+\pi_{23}} \\ 1+V+V^2 & & V+V^2 \end{array}$$

where the left vertical map is induced by the combination of the unique map $V^2 \to 1$, the two projection maps $V^2 \to V$, the identity map $V^2 \to V^2$, the unique map $V^3 \to 1$, the three projection maps $V^3 \to V$, and the three projection maps $V^3 \to V^2$. The horizontal map is the obvious fold map built out of the identity maps. The right vertical map is projection onto the second coordinate $V^2 \to V$, combined with projection onto the second two coordinates $V^3 \to V^2$. Observe that this diagram is obtained from the previous one by replacing the edge set E by V^2 (with the two projections π_1 and π_2 corresponding to the source and target maps s and t, respectively), and adding copies of V^3 and V^2. The integral transform of $\alpha = \langle \alpha_1, \alpha_V, \alpha_{V^2} \rangle : 1 + V + V^2 \to R$, along this diagram is:

$$
\begin{aligned}
\mathcal{F}_D(\alpha) : V + V^2 &\to \mathtt{bag} \circ \mathtt{list}(R) \\
v &\mapsto \sum_{u \in V} [\alpha_1 \, , \, \alpha_V(u) \, , \, \alpha_V(v) \, , \, \alpha_{V^2}(u,v)] \\
(a,b) &\mapsto \sum_{u \in V} [\alpha_1 \, , \, \alpha_V(u) \, , \, \alpha_V(a) \, , \, \alpha_V(b) \, , \, \alpha_{V^2}(u,a) \, , \, \alpha_{V^2}(u,b) \, , \, \alpha_{V^2}(a,b)]
\end{aligned}
$$

References

[DV22] Andrew J. Dudzik and Petar Veličković, *Graph Neural Networks are Dynamic Programmers*, Advances in Neural Information Processing Systems **35** (December 2022), 20635–20647.