Pseudo-codes for Graph Search Algorithm

1 Enumeration Algorithm of Fringe-Trees via Sequence Representations

For an acyclic chemical graph $G = (H, \alpha, \beta)$ on n vertices, let $V(H) = \{v_1, v_2, \ldots, v_n\}$ be such that $\deg_H(v_n) = 1$. We say that G is rooted at v_1. Let pred : $[2, n] \to [1, n-1]$ be a bijection such that for $k \in [2, n]$, $v_k v_{\mathrm{pred}(k)} \in E(H)$. We call the alternating sequence $S \triangleq (\alpha(v_1), \beta(v_{\mathrm{pred}(2)}v_2), \alpha(v_2), \ldots, \beta(v_{\mathrm{pred}(n)}v_n), \alpha(v_n))$ the *sequence representation* of G.

Algorithm $\textsc{SeqMap}(\Lambda, \boldsymbol{x}^*, \delta)$
Input: A set Λ of chemical elements,
  a vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\mathrm{co}}, \boldsymbol{x}^*_{\mathrm{in}}, \boldsymbol{x}^*_{\mathrm{ex}}, b)$ with $\boldsymbol{x}^*_{\mathrm{co}} \in \mathbb{Z}^{\Lambda^{\mathrm{co}}}$, $\boldsymbol{x}^*_{\mathtt{t}} \in \mathbb{Z}^{\Lambda^{\mathtt{t}}}$, $\mathtt{t} \in \{\mathrm{in}, \mathrm{ex}\}$, $b \in \mathbb{Z}_+$ and
  an integer δ.
Output: The set of sequence representations of all acyclic graphs G and
  their frequency vectors $\boldsymbol{w} = (\boldsymbol{w}_{\mathrm{co}}, \boldsymbol{w}_{\mathrm{in}}, \boldsymbol{w}_{\mathrm{ex}}, 0)$ such that $\boldsymbol{w} \leq \boldsymbol{x}^*$, degree of root in G is 1, and
  G has $\delta + 1$ vertices,
  where the set of these sequences is stored in a trie.
for each $t = \mathtt{a} \in \Lambda$ **do**
  $\mathrm{Cld}_t := \mathrm{Leaf}_t := \emptyset$;
  for each $\mathtt{b} \in \Lambda$ and $m \in [1, 3]$ such that $\mathrm{val}(\mathtt{a}) \geq m, \mathrm{val}(\mathtt{b}) \geq m$ **do**
    Let $S := (\mathtt{a}, m, \mathtt{b})$; /* Sequence representation of a tree with two vertices */
    if $\textsc{Trie}(m, \mathtt{b}, S, \delta - 1)$ returns a node v_γ and
     a leaf set Leaf_γ **then**
      $\mathrm{Leaf}_t := \mathrm{Leaf}_t \cup \mathrm{Leaf}_\gamma$; $\mathrm{Cld}_t := \mathrm{Cld}_t \cup \{v_\gamma\}$
    endif
  endfor;
  if $\mathrm{Cld}_t \neq \emptyset$ **then**
    Create a new node u_t as the parent of nodes in Cld_t;
    Sort the leaves $u \in \mathrm{Leaf}_t$ in lexicographically descending order
     with respect to $\mathrm{key}(u) = (S_u, \mathtt{a}_u, h_u)$;
    Partition Leaf_t into subsets $\mathrm{Leaf}_t^{(i)}$, $i = 1, 2, \ldots, m_t$ so that $\mathrm{key}(u) = \mathrm{key}(u')$
      if and only if $u, u' \in \mathrm{Leaf}_t^{(i)}$ for some i;
    For each $i = 1, 2, \ldots, m_t$, create a new node $u_{t,i}$ (called a superleaf) to the leaves in $\mathrm{Leaf}_t^{(i)}$
      and define $\mathrm{key}(u_{t,i})$ to be $\mathrm{key}(u) = (S_u, \mathtt{a}_u, h_u)$ for a leaf $u \in \mathrm{Leaf}_t^{(i)}$
  endif;
  Set $\mathrm{S}^{(\delta)}[\boldsymbol{x}^*, t]$ to be the set of sequences $S = \mathrm{key}_1(u_{t,i})$ for all superleaves $u_{t,i}$
endfor;
Output $\{\mathrm{S}^{(\delta)}[\boldsymbol{x}^*, t] \mid t \in \Lambda\}$ as the required set of sequence representation of acyclic graphs, and
  for each $S \in \{\mathrm{S}^{(\delta)}[\boldsymbol{x}^*, t] \mid t \in \Lambda\}$, the frequency vector of the graph
  of which the sequence representation is S.

Recursive Procedure $\textsc{Trie}(h, \mathtt{a}, S, \delta)$
Input: A set Λ of chemical elements,
a vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\rm co}, \boldsymbol{x}^*_{\rm in}, \boldsymbol{x}^*_{\rm ex}, b)$ with $\boldsymbol{x}^*_{\rm co} \in \mathbb{Z}^{\Lambda^{\rm co}}$, $\boldsymbol{x}^*_{\mathtt{t}} \in \mathbb{Z}^{\Lambda^{\mathtt{t}}}$, $\mathtt{t} \in \{\rm in, ex\}$,
$b \in \mathbb{Z}_+$ (global constants),
an integer $h \in [1, 3]$, an element $\mathtt{a} \in \Lambda$,
a sequence representation S, and
an integer $\delta \geq 0$.
Output: The set of sequence representation of graphs G rooted at atom $\mathtt{a}$, with $\delta + 1$ vertices
that can be extended from S, and
frequency vector $\boldsymbol{w}$ of G when $\delta = 0$ and $\boldsymbol{w} \leq \boldsymbol{x}^*$,
where the set of these sequences is stored in a trie.
A trie that stores all sequences of length δ from atom $\mathtt{a}$
with a j-bond ($j \in [1, \mathrm{val}(\mathtt{a}) - h]$);
if $\delta = 0$ **then**
  if the frequency vector of the graph with sequence representation S is at most $\boldsymbol{x}^*$
   where we do not consider the configuration of the edge with root as an end vertex **then**
    Create a new leaf node u with $\mathrm{key}(u) = (S, \mathtt{a}, h)$, return u and a leaf set $\mathrm{Leaf} := \{u\}$
  end if
else
  $\mathrm{Cld} := \mathrm{Leaf} := \emptyset$;
  for each $\mathtt{b} \in \Lambda$ and $m \in [1, 3]$ with $\mathrm{val}(\mathtt{a}) \geq m + h, \mathrm{val}(\mathtt{b}) \geq m$ **do**
    if $\textsc{Trie}(m + h, \mathtt{b}, (S, m, \mathtt{b}), \delta - 1)$ returns a node v and
     a leaf set Leaf_v **then**
      $\mathrm{Cld} := \mathrm{Cld} \cup \{v\}$; $\mathrm{Leaf} := \mathrm{Leaf} \cup \mathrm{Leaf}_v$
    endif
  endfor;
  if $\mathrm{Cld} = \emptyset$ **then**
    Return `empty`
  endif
endif.

1.1 Generating All Fringe Trees

We enumerate all possible 2-fringe-trees rooted at vertices with label $\mathtt{a}$ in Λ, under a given resource vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\rm co}, \boldsymbol{x}^*_{\rm in}, \boldsymbol{x}^*_{\rm ex}, b)$.

FRINGETREEWEIGHTVECTORS($\mathtt{a}$)

Input: A vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\rm co}, \boldsymbol{x}^*_{\rm in}, \boldsymbol{x}^*_{\rm ex}, b)$ with $\boldsymbol{x}^*_{\rm co} \in \mathbb{Z}^{\Lambda^{\rm co}}$, $\boldsymbol{x}^*_{\mathtt{t}} \in \mathbb{Z}^{\Lambda^{\mathtt{t}}}$, $\mathtt{t} \in \{\rm in, ex\}$,
two non-negative integers b and $h \leq 2 (= \rho)$, an element $\mathtt{a} \in \Lambda$ and an integer $g \geq 1$.

Output: The sets $\mathrm{V}^{(0)}_{\rm end}(\mathtt{a}, d, m; \boldsymbol{x}^*)$ (resp., $\mathrm{V}^{(0)}_{\rm inl}(\mathtt{a}, d, m; \boldsymbol{x}^*)$,
$\mathrm{V}^{(0)}_{\rm co+2}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$ and $\mathrm{V}^{(0)}_{\rm co+3}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$)
$d \in [1, \mathrm{val}(\mathtt{a}) - 1]$ (resp., $d \in [0, \mathrm{val}(\mathtt{a}) - 2]$ and $d \in [0, \mathrm{val}(\mathtt{a}) - 3]$) and
$m \in [d, \mathrm{val}(\mathtt{a}) - 1]$ (resp., $m \in [d, \mathrm{val}(\mathtt{a}) - 2]$ and $m \in [d, \mathrm{val}(\mathtt{a}) - 3]$)
and for each vector $\boldsymbol{w}$ in these sets, the set $\mathcal{T}_{\boldsymbol{w}}$ of all trees $T_{\boldsymbol{w}}$ with frequency vector $\boldsymbol{w}$ and the size $n(\boldsymbol{w})$ of $\mathcal{T}_{\boldsymbol{w}}$.

Step 1: Enumerate all fringe-trees T rooted at vertex v_r such that
  $\alpha(v_r) = \mathtt{a}$, the height is 2, (resp., at most 2)
  the degree $d_{\rm root}$ of v_r is 1 (i.e., v_r has exactly one child v_c) with
  $\boldsymbol{f}(\gamma^{\rm ex}) \leq \boldsymbol{x}^*_{\rm ex}$
  /* Using recursive algorithm SEQMAP to enumerate these */
  Let $\mathcal{T} = \{(T_i, k_i, d_i, \boldsymbol{w}^i_{\rm co}, \boldsymbol{w}^i_{\rm in}, \boldsymbol{w}^i_{\rm ex}) \mid i = 1, 2, \ldots, q\}$ denote the resulting set of fringe-trees,
  where T_i denotes the i-th tree (say, generated as the i-th solution),
  k_i denotes the multiplicity of edge $v_r v_c$,
  d_i denotes the degree of child v_c, $\boldsymbol{w}^i_{\rm in} = \boldsymbol{f}_{\rm in}(T_i)$, and
  $\boldsymbol{w}^i_{\rm ex} = \boldsymbol{f}_{\rm ex}(T_i) - \mathbf{1}_\gamma$ for $\gamma = (\mathtt{a}1, \mathtt{b}d_c, k_i)$ and $\alpha(v_c) = \mathtt{b}$;

Step 2: Enumerate all fringe-trees T with $d_{\rm root} \in [1, 2, 3]$ as follows:
  $\mathrm{W}[\mathtt{a}, d, m] := \emptyset$ for $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$, $m \in [d, \mathrm{val}(a) - d]$;
  Let $\mathrm{dg}^+ := 1$ (resp., $\mathrm{dg}^+ := 2$, and $\mathrm{dg}^+ := 3$);
  for each $i \in [1, q]$ **do**
    if $|V(T_i)| \leq 4$, $\boldsymbol{w}^i_{\rm ex} + \mathbf{1}_{\gamma(i)} \leq \boldsymbol{x}^*_{\rm ex}$ holds for $\gamma(i) := (\mathtt{a}\{\mathrm{dg}^+ + 1\}, \mathtt{b}d_i, k_i)$ **then**
     /* Also test if the height of the tree T_i is exactly equal to 2 (resp., h) while
       constructing $\mathrm{V}^{(0)}_{\rm end}(a, d, m; \boldsymbol{x}^*)$ (resp., $\mathrm{V}^{(0)}_{\rm co+2}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$ and $\mathrm{V}^{(0)}_{\rm co+3}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$) */
      Let $\boldsymbol{w} := (\boldsymbol{w}^i_{\rm co}, \boldsymbol{w}^i_{\rm in}, \boldsymbol{w}^i_{\rm ex} + \mathbf{1}_{\gamma(i)}, 0)$;
      if $\boldsymbol{w} \in \mathrm{W}[\mathtt{a}, 1, k_i]$ **then** $n_{\boldsymbol{w}} := n_{\boldsymbol{w}} + 1$
       if $|\mathcal{T}_{\boldsymbol{w}}| < g$ **then** $\mathcal{T}_{\boldsymbol{w}} := \mathcal{T}_{\boldsymbol{w}} \cup \{T_i\}$
      else $\mathrm{W}[\mathtt{a}, 1, k_i] := \mathrm{W}[\mathtt{a}, 1, k_i] \cup \{\boldsymbol{w}\}$; $\mathcal{T}_{\boldsymbol{w}} := \{T_i\}$; $n_{\boldsymbol{w}} := 1$ **endif**
    endif;
    for each $j \in [i, q]$ **do**
      if $k_i + k_j \leq \mathrm{val}(\mathtt{a}) - \mathrm{dg}^+$ **then**
        for each $h \in [j, q]$ **do**
          Let $\mathtt{b}_i, \mathtt{b}_j, \mathtt{b}_k$ be the labels of the child of the roots of T_i, T_j, T_k, respectively;
          $\gamma(i) := (\mathtt{a}\{\mathrm{dg}^+ + 3\}, \mathtt{b}_i d_i, k_i)$; $\gamma(j) := (\mathtt{a}\{\mathrm{dg}^+ + 3\}, \mathtt{b}_j d_j, k_j)$; $\gamma(i) := (\mathtt{a}\{\mathrm{dg}^+ + 3\}, \mathtt{b}_h d_h, k_h)$;
          if $k_i + k_j + k_h \leq \mathrm{val}(\mathtt{a}) - \mathrm{dg}^+$ (i.e., $k_i = k_j = k_h = 1$ and $\mathrm{val}(\mathtt{a}) = 4$),
            $\boldsymbol{w}^i_{\rm ex} + \boldsymbol{w}^j_{\rm ex} + \boldsymbol{w}^h_{\rm ex} + \mathbf{1}_{\gamma(i)} + \mathbf{1}_{\gamma(j)} + \mathbf{1}_{\gamma(h)} \leq \boldsymbol{x}^*_{\rm ex}$
            and $|V(T_i)| + |V(T_j)| + |V(T_h)| - 2 \leq 8$ **then**
             /* Also test if the height of at least one tree T_i, T_j, T_h is exactly equal to 2 while

constructing $\mathrm{V}^{(0)}_{\mathrm{end}}(a, d, m; \boldsymbol{x}^*)$ */
$\boldsymbol{w} := (\boldsymbol{w}^i_{\mathrm{co}} + \boldsymbol{w}^j_{\mathrm{co}} + \boldsymbol{w}^h_{\mathrm{co}}, \boldsymbol{w}^i_{\mathrm{in}} + \boldsymbol{w}^j_{\mathrm{in}} + \boldsymbol{w}^h_{\mathrm{in}}, \boldsymbol{w}^i_{\mathrm{ex}} + \boldsymbol{w}^j_{\mathrm{ex}} + \boldsymbol{w}^h_{\mathrm{ex}} + \mathbf{1}_{\gamma(i)} + \mathbf{1}_{\gamma(j)} + \mathbf{1}_{\gamma(h)}, 0)$;
Let T be the tree obtained by identifying the roots of T_i, T_j, and T_h;
$m := k_i + k_j + k_h$;
if $\boldsymbol{w} \in \mathrm{W}[\mathtt{a}, 3, m]$ **then** $n_{\boldsymbol{w}} := n_{\boldsymbol{w}} + 1$
if $|\mathcal{T}_{\boldsymbol{w}}| < g$ **then** $\mathcal{T}_{\boldsymbol{w}} := \mathcal{T}_{\boldsymbol{w}} \cup \{T\}$
else
$\mathrm{W}[\mathtt{a}, 3, m] := \mathrm{W}[\mathtt{a}, 3, m] \cup \{\boldsymbol{w}\}$; $\mathcal{T}_{\boldsymbol{w}} := \{T\}$; $n_{\boldsymbol{w}} := 1$;
endif
endif
endfor;
$\gamma(i) := (\mathtt{a}\{\mathrm{dg}^+ + 2\}, \mathtt{b}_i d_i, k_i)$; $\gamma(j) := (\mathtt{a}(\mathrm{dg}^+ + 2), \mathtt{b}_j d_j, k_j)$;
if $|V(T_i)| + |V(T_j)| - 1 \leq 6$,
$\boldsymbol{w}^i_{\mathrm{ex}} + \boldsymbol{w}^j_{\mathrm{ex}} + \mathbf{1}_{\gamma(i)} + \mathbf{1}_{\gamma(j)} \leq \boldsymbol{x}^*_{\mathrm{ex}}$ **then**
/* Also test if the height of at least one tree T_i, T_j is exactly equal to 2 (resp., h) while constructing $\mathrm{V}^{(0)}_{\mathrm{end}}(a, d, m; \boldsymbol{x}^*)$ (resp., $\mathrm{V}^{(0)}_{\mathrm{co+2}}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$) */
$\boldsymbol{w} := (\boldsymbol{w}^i_{\mathrm{co}} + \boldsymbol{w}^j_{\mathrm{co}}, \boldsymbol{w}^i_{\mathrm{in}} + \boldsymbol{w}^j_{\mathrm{in}}, \boldsymbol{w}^i_{\mathrm{ex}} + \boldsymbol{w}^j_{\mathrm{ex}} + \mathbf{1}_{\gamma(i)} + \mathbf{1}_{\gamma(j)}, 0)$;
Let T be the tree obtained by identifying the roots of T_i and T_j;
$m := k_i + k_j$;
if $w \in \mathrm{W}[\mathtt{a}, 2, m]$ **then** $n_{\boldsymbol{w}} := n_{\boldsymbol{w}} + 1$
if $|\mathcal{T}_{\boldsymbol{w}}| < g$ **then** $\mathcal{T}_{\boldsymbol{w}} := \mathcal{T}_{\boldsymbol{w}} \cup \{T\}$
else
$\mathrm{W}[\mathtt{a}, 2, m] := \mathrm{W}[\mathtt{a}, 2, m] \cup \{\boldsymbol{w}\}$; $\mathcal{T}_{\boldsymbol{w}} := \{T\}$; $n_{\boldsymbol{w}} := 1$
endif
endif
endif
endfor
endfor;
/* It remains to calculate the set $\mathrm{V}^{(0)}_{\mathrm{inl}}(\mathtt{a}, 0, 0; \boldsymbol{x}^*)$, $\mathrm{V}^{(0)}_{\mathrm{co+2}}(\mathtt{a}, 0, 0, h; \boldsymbol{x}^*)$ and $\mathrm{V}^{(0)}_{\mathrm{co+3}}(\mathtt{a}, 0, 0, h; \boldsymbol{x}^*)$ */
Let T be a singleton vertex labeled $\mathtt{a}$;
$\mathrm{W}[\mathtt{a}, 0, 0] := \{\boldsymbol{w} = (\mathbf{0}, \mathbf{0}, \mathbf{0}, 0)\}$; $\mathcal{T}_{\boldsymbol{w}} := \{T\}$; $n_{\boldsymbol{w}} := 1$;
Output $\mathrm{W}[\mathtt{a}, d, m]$ as $\mathrm{V}^{(0)}_{\mathrm{end}}(\mathtt{a}, d, m; \boldsymbol{x}^*)$ (resp., $\mathrm{V}^{(0)}_{\mathrm{inl}}(\mathtt{a}, d, m; \boldsymbol{x}^*)$, $\mathrm{V}^{(0)}_{\mathrm{co+2}}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$ and $\mathrm{V}^{(0)}_{\mathrm{co+3}}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$), for each $\boldsymbol{w} \in \mathrm{W}[\mathtt{a}, d, m]$, $\mathcal{T}_{\boldsymbol{w}}$, and $n_{\boldsymbol{w}}$.

2 Computing Frequency Vectors of End-Subtrees

For an integer $h \geq 1$, element $\mathtt{a} \in \Lambda$, integers $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$, and $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$ we give a procedure to compute the set $\mathrm{V}^{(h)}_{\mathrm{end}}(\mathtt{a}, d, m; \boldsymbol{x}^*)$.

ComputeEndSubtreeOne$(\mathtt{a}, d, m, h)$

Input: Element $\mathtt{a} \in \Lambda$, integer $d \in [1, \mathrm{val}(a) - 1]$, $m \in [d, \mathrm{val}(a) - 1]$, $h \geq 1$.
/* Global data: A vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\mathrm{co}}, \boldsymbol{x}^*_{\mathrm{in}}, \boldsymbol{x}^*_{\mathrm{ex}}, b)$ with $\boldsymbol{x}^*_{\mathrm{co}} \in \mathbb{Z}^{\Lambda^{\mathrm{co}}}$, $\boldsymbol{x}^*_{\mathtt{t}} \in \mathbb{Z}^{\Lambda^{\mathtt{t}}}$, $\mathtt{t} \in \{\mathrm{in}, \mathrm{ex}\}$,

a non-negative integer b, the collection
$\mathcal{V}_{\rm inl}$ vector sets $\mathrm{V}_{\rm inl}(\mathtt{a}, d-1, m_{\mathtt{a}}; \boldsymbol{x}^*)$, $m_{\mathtt{a}} \in [d-1, \mathrm{val}(\mathtt{a})-2]$
$\mathcal{V}_{\rm end}^{(h-1)}$ of vector sets $\mathrm{V}_{\rm end}^{(h-1)}(\mathtt{a}_1, d_1, m_1; \boldsymbol{x}^*)$, $\mathtt{a}_1 \in \Lambda$, $d_1 \in [1, \mathrm{val}(\mathtt{a}_1)-1]$,
$m_1 \in [d_1, \mathrm{val}(\mathtt{a}_1)-1]$. */

Output: The set $\mathrm{V}_{\rm end}^{(h)}(\mathtt{a}, d, m; \boldsymbol{x}^*)$, where we store these vectors in a trie.
W := $\emptyset$;
for each triplet $(\mathtt{b}, d_{\mathtt{b}}, m_{\mathtt{b}})$ **do**
  for each triplet $(\mathtt{a}, d-1, m_{\mathtt{a}})$ **do**
    for each $\boldsymbol{y}^{\mathtt{b}} = (\boldsymbol{y}^{\mathtt{b}}_{\rm co}, \boldsymbol{y}^{\mathtt{b}}_{\rm in}, \boldsymbol{y}^{\mathtt{b}}_{\rm ex}, 0) \in \mathrm{V}_{\rm end}^{(h-1)}(\mathtt{b}, d_{\mathtt{b}}, m_{\mathtt{b}}; \boldsymbol{x}^*)$ **do**
      for each $m' \in [1, 3]$ such that
        - $\gamma^{\rm in} = (\mathtt{a}\{d+1\}, \mathtt{b}\{d_{\mathtt{b}}+1\}, m') \in \Gamma^{\rm in}$ and
        - $m_{\mathtt{a}} + m' = m, m_{\mathtt{a}} + m' + 1 \leq \mathrm{val}(\mathtt{a})$ and $m' + m_{\mathtt{b}} \leq \mathrm{val}(\mathtt{b})$ **do**
        for each $\boldsymbol{y}^{\mathtt{a}} = (\boldsymbol{y}^{\mathtt{a}}_{\rm co}, \boldsymbol{y}^{\mathtt{a}}_{\rm in}, \boldsymbol{y}^{\mathtt{a}}_{\rm ex}, 0) \in \mathrm{V}_{\rm inl}^{(0)}(\mathtt{a}, d-1, m_{\mathtt{a}}; \boldsymbol{x}^*)$ **do**
          $\boldsymbol{y}_{\rm in} := \boldsymbol{y}^{\mathtt{a}}_{\rm in} + \boldsymbol{y}^{\mathtt{b}}_{\rm in} + \mathbf{1}_{\gamma^{\rm in}}$;
          $\boldsymbol{y}_{\rm ex} := \boldsymbol{y}^{\mathtt{a}}_{\rm ex} + \boldsymbol{y}^{\mathtt{b}}_{\rm ex}$; $\boldsymbol{y} := (\boldsymbol{y}_{\rm co}, \boldsymbol{y}_{\rm in}, \boldsymbol{y}_{\rm ex}, 0)$;
          if $\boldsymbol{y} \leq \boldsymbol{x}^*$ **then**
            W := W $\cup \{\boldsymbol{y}\}$;
          end if
        end for
      end for
    end for
  end for
end for;
Output W as $\mathrm{V}_{\rm end}^{(h)}(\mathtt{a}, d, m; \boldsymbol{x}^*)$.

3 Generating Frequency Vectors of Rooted Core-subtrees

For an integer $h \geq 1$, element $\mathtt{a} \in \Lambda$, integers $\Delta \in [2, 3]$, $d \in [1, \mathrm{val}(\mathtt{a}) - \Delta]$, and $m \in [d, \mathrm{val}(\mathtt{a})-1]$ we give a procedure to compute the set $\mathrm{V}_{{\rm co}+\Delta}^{(0)}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$.

ComputCoreSubtreeOne$(\mathtt{a}, d, m, h)$

Input: Element $\mathtt{a} \in \Lambda$, integer $d \in [1, \mathrm{val}(a) - \Delta]$, $m \in [d, \mathrm{val}(a)-1]$, $h \geq 1$.
/* Global data: A vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\rm co}, \boldsymbol{x}^*_{\rm in}, \boldsymbol{x}^*_{\rm ex}, b)$ with $\boldsymbol{x}^*_{\rm co} \in \mathbb{Z}^{\Lambda^{\rm co}}$, $\boldsymbol{x}^*_{\mathtt{t}} \in \mathbb{Z}^{\Lambda^{\mathtt{t}}}$, $\mathtt{t} \in \{\mathrm{in}, \mathrm{ex}\}$,
a non-negative integer b, the collection
$\mathcal{V}_{{\rm co}+\Delta+1}^{(0)}$ vector sets $\mathrm{V}_{{\rm co}+\Delta+1}^{(0)}(\mathtt{a}, d-1, m_{\mathtt{a}}, p; \boldsymbol{x}^*)$, $m_{\mathtt{a}} \in [d-1, \mathrm{val}(\mathtt{a}) - \Delta - 1]$,
$p \in [0, 2(=\rho)]$
$\mathcal{V}_{\rm end}^{(h-2-1)}$ of vector sets $\mathrm{V}_{\rm end}^{(h-2-1)}(\mathtt{a}_1, d_1, m_1; \boldsymbol{x}^*)$, $\mathtt{a}_1 \in \Lambda$, $d_1 \in [1, \mathrm{val}(\mathtt{a}_1)-1]$,
$m_1 \in [d_1, \mathrm{val}(\mathtt{a}_1)-1]$, integer $g \geq 1$. */

Output: The set $\mathrm{V}_{{\rm co}+\Delta}^{(0)}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$, where we store vectors $\mathrm{V}_{{\rm co}+\Delta}^{(0)}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$,
in a trie.
W := $\emptyset$;
for each triplet $(\mathtt{b}, d_{\mathtt{b}}, m_{\mathtt{b}})$ **do**

for each triplet $(\mathtt{a}, d-1, m_{\mathtt{a}}, p)$ **do**
for each $\boldsymbol{y}^{\mathtt{b}} = (\boldsymbol{y}^{\mathtt{b}}_{\mathrm{co}}, \boldsymbol{y}^{\mathtt{b}}_{\mathrm{in}}, \boldsymbol{y}^{\mathtt{b}}_{\mathrm{ex}}, 0) \in \mathrm{V}^{(h-2-1)}_{\mathrm{end}}(\mathtt{b}, d_{\mathtt{b}}, m_{\mathtt{b}}; \boldsymbol{x}^*)$ **do**
for each $m' \in [1,3]$ such that
- $\gamma^{\mathrm{in}} = (\mathtt{a}\{d+\Delta\}, \mathtt{b}\{d_{\mathtt{b}}+1\}, m') \in \Gamma^{\mathrm{in}}$ and
- $m_{\mathtt{a}} + m' = m, m_{\mathtt{a}} + m' + \Delta \leq \mathrm{val}(\mathtt{a})$ and $m' + m_{\mathtt{b}} \leq \mathrm{val}(\mathtt{b})$ **do**
for each $\boldsymbol{w}^{\mathtt{a}} = (\boldsymbol{w}^{\mathtt{a}}_{\mathrm{co}}, \boldsymbol{w}^{\mathtt{a}}_{\mathrm{in}}, \boldsymbol{w}^{\mathtt{a}}_{\mathrm{ex}}, 0) \in \mathrm{W}^{(0)}_{\mathrm{inl}}(\mathtt{a}, d-1, m_{\mathtt{a}}, p; \boldsymbol{x}^*)$ **do**
$\boldsymbol{w}_{\mathrm{in}} := \boldsymbol{w}^{\mathtt{a}}_{\mathrm{in}} + \boldsymbol{y}^{\mathtt{b}}_{\mathrm{in}} + \mathbf{1}_{\gamma^{\mathrm{in}}}$;
$\boldsymbol{w}_{\mathrm{ex}} := \boldsymbol{w}^{\mathtt{a}}_{\mathrm{ex}} + \boldsymbol{y}^{\mathtt{b}}_{\mathrm{ex}}$; $\boldsymbol{y} := (\boldsymbol{y}_{\mathrm{co}}, \boldsymbol{y}_{\mathrm{in}}, \boldsymbol{y}_{\mathrm{ex}}, 1)$;
if $\boldsymbol{y} \leq \boldsymbol{x}^*$ **then**
$\mathrm{W} := \mathrm{W} \cup \{\boldsymbol{y}\}$;
end if
end for
end for
end for
end for
end for;
Output W as $\mathrm{V}^{(0)}_{\mathrm{co}+\Delta}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$.

4 Computing DAG Representation for v-Components

DAGRepresentationVertex$(\mathtt{a}_v, d_v, m_v, t, \Delta_v, \boldsymbol{x}^*)$

Input: /* Global data: A vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\mathrm{co}}, \boldsymbol{x}^*_{\mathrm{in}}, \boldsymbol{x}^*_{\mathrm{ex}}, b)$ with $\boldsymbol{x}^*_{\mathrm{co}} \in \mathbb{Z}^{\Lambda^{\mathrm{co}}}$, $\boldsymbol{x}^*_{\mathtt{t}} \in \mathbb{Z}^{\Lambda^{\mathtt{t}}}$, $\mathtt{t} \in \{\mathrm{in}, \mathrm{ex}\}$,
a non-negative integer b,
integers t, $\Delta_v \in [2,3]$, element $\mathtt{a}_v \in \Lambda$,
integers $d_v \in [0, \mathrm{val}(\mathtt{a}_v) - \Delta_v - 1]$, $m_v \in [d_v, \mathrm{val}(\mathtt{a}_v) - \Delta_v - 1]$,
the collection $\mathcal{V}^{(0)}_{\mathrm{inl}}$ vector sets $\mathrm{V}^{(0)}_{\mathrm{inl}}(\mathtt{a}, d, m; \boldsymbol{x}^*)$,
an integer $t \geq 0$,
$\mathcal{V}^{(h)}_{\mathrm{end}}$ of vector sets $\mathrm{V}^{(h)}_{\mathrm{end}}(\mathtt{a}_1, d_1, m_1; \boldsymbol{x}^*)$, $\mathtt{a}_1 \in \Lambda$, $d_1 \in [1, \mathrm{val}(\mathtt{a}_1) - 1]$,
$m_1 \in [d_1, \mathrm{val}(\mathtt{a}_1) - 1]$, $1 \leq h \leq t$,
the collection $\mathcal{V}^{(0)}_{\mathrm{end}}$ of sets $\mathrm{V}^{(0)}_{\mathrm{end}}(\mathtt{a}_1, d_1, m_1; \boldsymbol{x}^*)$, $\mathtt{a}_1 \in \Lambda$, $d_1 \in [1, \mathrm{val}(\mathtt{a}_1) - 1]$,
the collection $\mathcal{V}^{(0)}_{\mathrm{co}+(\Delta_v+1)}$ of sets $\mathrm{V}^{(0)}_{\mathrm{co}+(\Delta_v+1)}(\mathtt{a}_v, d_v - 1, m'', p; \boldsymbol{x}^*)$ for $p \leq 2$. */
Output: A vertex-labeled and edge-labeled DAG representation.
$F := \emptyset$;
$G := (N, A)$; $A := \emptyset$; $N := \emptyset$;
for each $\boldsymbol{w} \in \mathrm{V}^{(0)}_{\mathrm{co}+(\Delta_v+1)}(\mathtt{a}_v, d_v - 1, m', p; \boldsymbol{x}^*)$ for each possible (m', p) **do**
for each $\boldsymbol{y} \in \mathrm{V}^{(t)}_{\mathrm{end}}(\mathtt{a}_1, d_1, m_1; \boldsymbol{x}^*)$ for each possible $(\mathtt{a}_1, d_1, m_1)$ **do**
if there exists $\gamma := (\mathtt{a}_v\{d_v + \Delta_v\}, \mathtt{a}_1\{d_1 + 1\}, m_v - m') \in \Gamma^{\mathrm{co}}$
such that $\boldsymbol{y} + \boldsymbol{w} + \mathbf{1}_\gamma = \boldsymbol{x}^*$ **then**
$N := N \cup \{(\boldsymbol{x}^*, t+1; \mathtt{a}_v, d_v, m_v)\}$;
$N := N \cup \{(\boldsymbol{y}, t; \mathtt{a}_1, d_1, m_1)\}$;
$A := A \cup \{a_{\boldsymbol{x}^*\boldsymbol{y}}\}$ and

let the label of the arc $a_{\boldsymbol{x}^*\boldsymbol{y}}$ to be $(\boldsymbol{w}, m_v - m')$;
end if
end for
end for;
for each $\ell \in (t, \ldots, 1)$ **do**
$G'' := (N'', A'') :=$ DAGSublayer$(\mathcal{V}_{\mathrm{end}}^{(\ell-1)}, G, \ell - 1, \mathcal{V}_{\mathrm{inl}}^{(0)})$;
$N := N \cup N''; A := A_2 \cup A''$
end for;
Output G as DAG representations and the set F of feasible pairs of v-component.

DAGSublayer$(\mathcal{V}, G, \ell, \mathcal{V}')$

Input: A family $\mathcal{V}$ of set of vectors of trees with root label $\mathtt{a}_1$, degree d_1 and multiplicity m_1, $G = (N, A)$,
a family of $\mathcal{V}'$ vector sets of fringe-trees,
ℓ (the height of the layer that we add in G at this stage).
Output: A DAG G' that is a super-graph of G.
$G' := G$;
for each $\boldsymbol{y}_1 \in \mathcal{V}$ **do**
for each $\boldsymbol{w} \in \mathcal{V}'$ **do**
if there exists $\gamma \in \Gamma^{\mathrm{in}}$ and some $\boldsymbol{y}_2 \in N$ such that
$\boldsymbol{y}_i, i = 1, 2$ are feasible, i.e., $\boldsymbol{y}_1 + \boldsymbol{w} + 1_\gamma = \boldsymbol{y}_2$ **then**
if $\boldsymbol{y}_1 \notin N$ **then** $N := N \cup \{(\boldsymbol{y}_1, \ell; \mathtt{a}_1, d_1, m_1)\}$;
$A := A \cup \{a_{\boldsymbol{y}_2\boldsymbol{y}_1}\}$ and
label the arc from $\boldsymbol{y}_2$ to $\boldsymbol{y}_1$ by $(\boldsymbol{w}, m)$,
where m is the bond multiplicity in γ
end if
end for
end for;
Output G' as a required DAG.

5 Counting Paths and Graphs

CountPathsGraphs($G = (N, A)$)

Input: A DAG $G = (N, A)$.
Output: For each source $\boldsymbol{v} \in N$, the number of paths and graphs
  that can be obtained from $\boldsymbol{v}$ to sinks in G.
Let N_h denote the set of vertices $\boldsymbol{v} \in N$ such that $\mathrm{ht}(\boldsymbol{v}) = h, h \in [0, \mathrm{ht}(G)]$
$p(\boldsymbol{v}) := 1$ for each $\boldsymbol{v} \in N_0$;
$p(\boldsymbol{v}) := 0$, $n(\boldsymbol{v}) := 0$ for each $\boldsymbol{v} \in N \backslash N_0$;
if the nodes in N_0 correspond to vectors of fringe trees **then**
  for each $\boldsymbol{v} \in N_0$ **do**
    $n(\boldsymbol{v}) := n_{\boldsymbol{v}}$, where $n_{\boldsymbol{v}}$ is the number of fringe trees with vector $\boldsymbol{v}$
  end for
else
  for each $\boldsymbol{v} \in N_0$ **do**
    Compute DAG $G_{\boldsymbol{v}}$ for $\boldsymbol{v}$;
    $n(\boldsymbol{v}) :=$ the number of graphs obtained by CountPathsGraphs($G_{\boldsymbol{v}}$)
  end for
end if;
for each $h \in [0, \mathrm{ht}(G) - 1]$ **do**
  for each $\boldsymbol{v} \in N_h$ **do**
    Let $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k \in N_{h+1}$ such that $\boldsymbol{v}_i\boldsymbol{v} \in A$ with $\boldsymbol{v}_i = \boldsymbol{v} + \boldsymbol{x}_i + \gamma$;
    for each $i \in [1, k]$ **do**
      if $\boldsymbol{x}_i$ corresponds to a fringe tree **then**
        $n(\boldsymbol{x}_i) := n_{\boldsymbol{x}_i}$; $x(i) := 1$
      else
        Compute DAG $G_{\boldsymbol{x}_i}$ for $\boldsymbol{x}_i$;
        $n(\boldsymbol{x}_i) :=$ the number of graphs obtained by CountPathsGraphs($G_{\boldsymbol{x}_i}$);
        $x(i) :=$ the number of paths obtained by CountPathsGraphs($G_{\boldsymbol{x}_i}$);
      end if;
      $n(\boldsymbol{v}_i) := n(\boldsymbol{x}_i)n(\boldsymbol{v}) + n(\boldsymbol{v}_i)$; $p(\boldsymbol{v}_i) := x(i)p(\boldsymbol{v}) + p(\boldsymbol{v}_i)$
    end for
  end for
end for;
Output $p(\boldsymbol{v})$ and $n(\boldsymbol{v})$, for each $\boldsymbol{v} \in N_{\mathrm{ht(G)}}$, as the number of paths and graphs obtained
  from source $\boldsymbol{v}$ to sinks in G, respectively.

- The number of all paths and graphs obtained from sources to sinks in $G = (N, A)$ are $\sum_{\boldsymbol{v} \in N_{\mathrm{ht(G)}}} p(\boldsymbol{v})$ and $\sum_{\boldsymbol{v} \in N_{\mathrm{ht(G)}}} n(\boldsymbol{v})$, respectively, where $p(\boldsymbol{v})$ and $n(\boldsymbol{v})$ is the number of paths and graphs obtained by CountPathsGraphs($G = (N, A)$).
- Let $G = (N, A)$ and $G' = (N', A')$ denote the two DAGs for core part for a base vertex e and $V_{\mathrm{pair}}(e)$ denote the set of feasible vector pairs. Then the number of all paths and graphs that

correspond to e-component are $\sum_{\substack{(\boldsymbol{z},\boldsymbol{z}')\in V_{\text{pair}}(e) \\ \boldsymbol{z}\in N, \boldsymbol{z}'\in N'}} p(\boldsymbol{z})p(\boldsymbol{z}')$ and $\sum_{\substack{(\boldsymbol{z},\boldsymbol{z}')\in V_{\text{pair}}(e) \\ \boldsymbol{z}\in N, \boldsymbol{z}'\in N'}} n(\boldsymbol{z})n(\boldsymbol{z}')$, respectively, where $p(\boldsymbol{z})$, $p(\boldsymbol{z}')$ and $n(\boldsymbol{z})$, $n(\boldsymbol{z}')$ are the number of paths and graphs from $\boldsymbol{z}$ and $\boldsymbol{z}'$ to sinks in G and G', respectively.

6 Priority-based Enumeration of Paths in DAG

EnumPaths(DAG)

Input: A rooted vertex-labeled and edge-labeled DAG $G = (N, A)$
and the number $p(\boldsymbol{v})$ of paths from each node $\boldsymbol{v} \in N$ to sinks.
Output: All directed paths from sources to sinks by traversing the DAG w.r.t. the values $p(\boldsymbol{v})$.
We consider G a rooted DAG with a virtual root r that is adjacent with all source vertices;
We consider dfs ordering on the vertices of G starting from root with index 0 and
traverse G in left-right ordering on the children of each vertex;
$\mathcal{P} := \emptyset$;
Let $Q_i :=$ the set of dfs label of all children of the vertex with dfs label i, $i \in |N|$;
if $|Q_1| = 0$ **then**
$\mathcal{P} := \mathcal{P} \cup \{1\}$
else
while $Q_1 \neq \emptyset$ **do**
Let i be the smallest integer in Q_1 and
the node $\boldsymbol{v}_i$ with dfs label i has maximum value $p(\boldsymbol{v}_i)$
among all other nodes with label in Q_1;
Let $\boldsymbol{v}_1$, and $\boldsymbol{v}_i$ be the label of vertices with dfs label 1 and i, respectively, and
the label of arc between $\boldsymbol{v}_1$ and $\boldsymbol{v}_i$ is $(\boldsymbol{x}, m)$
$P := ((\boldsymbol{v}_1, \boldsymbol{v}_i, \boldsymbol{x}, m))$;
$\mathcal{P}' :=$ PathRecursion$(P, i, \mathcal{P}, G)$; $Q_1 := Q_1 \setminus \{i\}$; $\mathcal{P} := \mathcal{P} \cup \mathcal{P}'$
end while
end if;
Output $\mathcal{P}$ as the required family of paths.

PathRecursion$(P, i, \mathcal{P}, G)$

Input: A DAG $G = (N, A)$ with dfs ordering, a path P,
a family of paths $\mathcal{P}$ an integer $i \in [2, |N|]$ and
the number $p(\boldsymbol{v})$ of paths from each node $\boldsymbol{v} \in N$ to sinks.
Output: Family of paths in G that can be extended from P
by traversing the DAG w.r.t. the values $p(\boldsymbol{v})$.
$\mathcal{P}' := \emptyset$;
Let $Q_i :=$ the set of dfs label of all children of the vertex with dfs label i;
if $|Q_i| = 0$ **then** $\mathcal{P}' := \mathcal{P}' \cup \{P\}$;
else

while $Q_i \neq \emptyset$ **do**
Let j be the smallest integer in Q_i and
the node $\boldsymbol{v}_j$ with dfs label j has maximum value $p(\boldsymbol{v}_j)$
among all other nodes with label in Q_j;
Let $\boldsymbol{v}_i$, and $\boldsymbol{v}_j$ be the labels of the vertices with dfs label i and j, respectively, and
the label of arc between $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ is $(\boldsymbol{x}, m)$
$P' := P \oplus ((\boldsymbol{v}_i, \boldsymbol{v}_j, \boldsymbol{x}, m))$; /* sequence concatenation */
$\mathcal{P}'' :=$ PathRecursion$(P', j, \mathcal{P}', G)$; $Q_i := Q_i \setminus \{j\}$; $\mathcal{P}' := \mathcal{P}' \cup \mathcal{P}''$
end while
end if;
Output $\mathcal{P}'$ as the required family of extended paths.

7 A Complete Algorithm to Compute Target v-components

We briefly summarize how to use the procedures described thus far to obtain an algorithm. Our global constants are vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\rm co}, \boldsymbol{x}^*_{\rm in}, \boldsymbol{x}^*_{\rm ex}, b)$ with $\boldsymbol{x}^*_{\rm co} \in \mathbb{Z}^{\Lambda^{\rm co}}$, $\boldsymbol{x}^*_{\tt t} \in \mathbb{Z}^{\Lambda^{\tt t}}$, $\tt t \in \{\rm in, ex\}$, a non-negative integer b, integer $\Delta_v \in [2, 3]$, element $\mathtt{a}_v \in \Lambda$,
integers $d_v \in [0, \mathrm{val}(\mathtt{a}_v) - \Delta_v - 1]$, $m_v \in [d_v, \mathrm{val}(\mathtt{a}_v) - \Delta_v - 1]$.

CompleteAlgorithmVertex(Global constants: $\mathtt{a}_v, d_v, m_v, \Delta_v, \boldsymbol{x}^*_v$, core height)

Let $\ell := |\Gamma^{\rm in}| + 2$;
$t :=$ core height $- 3$;
Compute $\mathrm{V}^{(0)}_{\mathrm{co}+\Delta_v}(\mathtt{a}_v, d_v, m_v, h; \boldsymbol{x}^*_v)$ for a fixed $(\mathtt{a}_v, d_v, m_v, \Delta_v)$,
and for each $h \leq \ell$ if $\ell \leq 2$ and $\boldsymbol{x}^*_v(\mathtt{bc}) = 0$;
Compute $\mathrm{V}^{(0)}_{\mathrm{co}+(\Delta_v+1)}(\mathtt{a}_v, d_v, m, h; \boldsymbol{x}^*_v)$ for a fixed $(\mathtt{a}_v, d_v, \Delta_v)$,
for each $m \in [d_v - 1, \mathrm{val}(\mathtt{a}_v) - \Delta_v - 1]$, $h \leq 2$ if $\ell > 2$ and $\boldsymbol{x}^*_v(\mathtt{bc}) = 1$;
Compute $\mathrm{V}^{(0)}_{\mathrm{end}}(\mathtt{a}, d, m; \boldsymbol{x}^*_v)$ for each $\mathtt{a} \in \Lambda$, $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$,
$m \in [d, \mathrm{val}(\mathtt{a}) - 1]$ if $\ell > 2$ and $\boldsymbol{x}^*_v(\mathtt{bc}) = 1$;
Compute $\mathrm{V}^{(0)}_{\mathrm{inl}}(\mathtt{a}, d, m; \boldsymbol{x}^*_v)$ for each $\mathtt{a} \in \Lambda$, $d \in [0, \mathrm{val}(\mathtt{a}) - 2]$,
$m \in [d, \mathrm{val}(\mathtt{a}) - 2]$ if $\ell > 2$ and $\boldsymbol{x}^*_v(\mathtt{bc}) = 1$;
Compute $\mathrm{V}^{(h)}_{\mathrm{end}}(\mathtt{b}, d', m'; \boldsymbol{x}^*_v)$ for each $\mathtt{b} \in \Lambda$, $d' \in [1, \mathrm{val}(\mathtt{b}) - 1]$,
$m' \in [d', \mathrm{val}(\mathtt{b}) - 1]$, $1 \leq h \leq t$, if $\ell > 2$ and $\boldsymbol{x}^*_v(\mathtt{bc}) = 1$;
Compute the DAG G representation of $\boldsymbol{x}^*_v$;
Enumerate the set $\mathcal{P}$ from sources to leaves in G;
for each path P in G **do**
Let $P := ((\boldsymbol{x}^*, \boldsymbol{y}_h, \boldsymbol{w}_h, m_h), (\boldsymbol{y}_h, \boldsymbol{y}_{h-1}, \boldsymbol{w}_{h-1}, m_{h-1}), \ldots, (\boldsymbol{y}_1, \boldsymbol{y}_0, \boldsymbol{w}_0, m_0))$;
where $\boldsymbol{w}_h \in \mathrm{V}^{(\delta_1)}_{\mathrm{co}+(\Delta_v+1)}$, $\boldsymbol{w}_{h-1}, \ldots, \boldsymbol{w}_1 \in \mathrm{V}^{(0)}_{\mathrm{inl}}$, $\boldsymbol{w}'_0 \in \mathrm{V}^{(0)}_{\mathrm{end}}$, $h = t$;
Get a target v-component by using the trees corresponding to
$\boldsymbol{w}_h, \boldsymbol{w}_{h-1}, \ldots, \boldsymbol{w}_0,$;
Get the number of v-components obtained by the path P
$n(\boldsymbol{w}_h) \times \cdots \times n(\boldsymbol{w}_0)$, where $n(\boldsymbol{w}_h), \ldots, n(\boldsymbol{w}_0)$, are the number of trees with vector
$\boldsymbol{w}_h, \ldots, \boldsymbol{w}_0$, respectively

end for.

8 Generation of Frequency Vectors of Bi-rooted Core-subtrees

For an integer $h \in [h_1, h_2]$, elements $\mathtt{a}, \mathtt{a}^e \in \Lambda$, integers $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$, $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$, $\Delta^e \in [1, \mathrm{val}(\mathtt{a}^e) - 1]$, $m^e \leq \mathrm{val}(\mathtt{a}^e) - \Delta^e$, and $q \geq 1$, we give a procedure to compute the set $\mathrm{V}^{(q)}_{\mathrm{co}+1,\Delta^e}(\mathtt{a}, d, m, \mathtt{a}^e, 1, m^e, h; \boldsymbol{x}^*)$.

COMPUTEBIROOTEDCORESUBTREE$(\mathtt{a}, d, m, \mathtt{a}^e, 1, m^e, h, q)$

Input: An integer $h \geq 0$, elements $\mathtt{a}, \mathtt{a}^e \in \Lambda$, integers $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$, $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$,
$\Delta^e \in [1, \mathrm{val}(\mathtt{a}^e) - 1]$, $m^e \leq \mathrm{val}(\mathtt{a}^e) - \Delta^e$, and $q \geq 1$.
/* Global data: A vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\mathrm{co}}, \boldsymbol{x}^*_{\mathrm{in}}, \boldsymbol{x}^*_{\mathrm{ex}}, b)$ with $\boldsymbol{x}^*_{\mathrm{co}} \in \mathbb{Z}^{\Lambda^{\mathrm{co}}}$, $\boldsymbol{x}^*_{\mathtt{t}} \in \mathbb{Z}^{\Lambda^{\mathtt{t}}}$, $\mathtt{t} \in \{\mathrm{in}, \mathrm{ex}\}$,
a non-negative integer b, the collection
$\mathcal{V}^{(0)}_{\mathrm{co}+2}$ vector sets $\mathrm{V}^{(0)}_{\mathrm{co}+2}(\mathtt{a}, d-1, m_{\mathtt{a}}, p; \boldsymbol{x}^*)$, $m_{\mathtt{a}} \in [d-1, \mathrm{val}(\mathtt{a}) - \Delta - 1]$, $p \in [0, h]$,
for $q \geq 2$, $\mathcal{V}^{(q-1)}_{\mathrm{end}}$ of vector sets $\mathrm{V}^{(q-1)}_{\mathrm{co}+1,\Delta^e}(\mathtt{b}, d', m', \mathtt{a}^e, 1, m^e, h'; \boldsymbol{x}^*)$,
$\mathtt{b} \in \Lambda$, $d' \in [1, \mathrm{val}(\mathtt{b}) - 1]$, $m' \in [d', \mathrm{val}(\mathtt{b}) - 1]$, $h' \in [0, h]$, integer $g \geq 1$. */
Output: The set $\mathrm{V}^{(q)}_{\mathrm{co}+1,\Delta^e}(\mathtt{a}, d, m, \mathtt{a}^e, 1, m^e, h; \boldsymbol{x}^*)$, where we store these vectors in a trie.

```
W := ∅;
for each triplet (a, d − 1, m_a, p) do
  if q = 1 then
    if p = h and val(a) ≥ m_a + m^e then
      for each w^a ∈ V^(0)_{co+2}(a, d − 1, m_a, p; x*) do
        γ^co := (ad, a^e 1, m^e); y := y^a + 1_{γ^co}
        if γ^co ∈ Γ^co and y ≤ x* then
          if y ∈ V then
            V := V ∪ {y}
          end if
        end if
      end for
    end if
  else /* q > 1 */
    for each triplet (b, d_b, m_b, h') do
      for each y^b ∈ V^(q−1)_{co+1,Δ^e}(b, d_b, m_b, a^e, 1, m^e, h'; x*) do
        for each m' ∈ [1, 3] such that
          - γ^co := (ad, b{d_b + 1}, m') ∈ Γ^co and
          - m_a + m' = m, m_a + m' + 1 ≤ val(a), m' + m_b ≤ val(b),
          - h = max{p, h'} and
          - y := y_a + y_b + 1_{γ^co} ≤ x* do
          if y ∈ V then
            V := V ∪ {y};
          end if
        end for
```

end for
end for
end if
end for;
Output W as $\mathrm{V}^{(q)}_{\mathrm{co}+1,\Delta^e}(\mathtt{a}, d, m, \mathtt{a}^e, 1, m^e, h; \boldsymbol{x}^*)$.

9 Computing DAG Representation for e-Components

DAGRepresentationEdge($\mathtt{a}^e_i, m^e_i, \Delta^e_i, \delta_i, h_i \boldsymbol{x}^*$)

Input: /* Global data: A vector $\boldsymbol{x}^* = (\boldsymbol{x}^*_{\mathrm{co}}, \boldsymbol{x}^*_{\mathrm{in}}, \boldsymbol{x}^*_{\mathrm{ex}}, b)$ with $\boldsymbol{x}^*_{\mathrm{co}} \in \mathbb{Z}^{\Lambda^{\mathrm{co}}}$, $\boldsymbol{x}^*_{\mathtt{t}} \in \mathbb{Z}^{\Lambda^{\mathtt{t}}}$, $\mathtt{t} \in \{\mathrm{in}, \mathrm{ex}\}$,
a non-negative integer b,
$\mathtt{a}^e_i \in \Lambda$, integers $\Delta^e_i \in [1, \mathrm{val}(\mathtt{a}^e_i) - 1]$, $m^e_i \leq \mathrm{val}(\mathtt{a}^e_i) - \Delta^e_i$,
the collection $\mathcal{V}^{(0)}_{\mathrm{inl}}$ vector sets $\mathrm{V}^{(0)}_{\mathrm{inl}}(\mathtt{a}, d, m; \boldsymbol{x}^*)$,
integers $\delta_i \geq 0$, $h_i \geq 1$, $i = 1, 2$,
$\mathcal{V}^{(h)}_{\mathrm{end}}$ of vector sets $\mathrm{V}^{(h)}_{\mathrm{end}}(\mathtt{a}_1, d_1, m_1; \boldsymbol{x}^*)$, $\mathtt{a}_1 \in \Lambda$, $d_1 \in [1, \mathrm{val}(\mathtt{a}_1) - 1]$,
$m_1 \in [d_1, \mathrm{val}(\mathtt{a}_1) - 1]$, $1 \leq h \leq \max\{\delta_1, \delta_2\}$,
the collection $\mathcal{V}^{(0)}_{\mathrm{end}}$ of sets $\mathrm{V}^{(0)}_{\mathrm{end}}(\mathtt{a}_1, d_1, m_1; \boldsymbol{x}^*)$, $\mathtt{a}_1 \in \Lambda$, $d_1 \in [1, \mathrm{val}(\mathtt{a}_1) - 1]$,
$m_1 \in [d_1, \mathrm{val}(\mathtt{a}_1) - 1]$,
the collection $\mathcal{V}^{(0)}_{\mathrm{co}+2}$ of sets $\mathrm{V}^{(0)}_{\mathrm{co}+2}(\mathtt{a}, d, m, h; \boldsymbol{x}^*)$ for all possible $\mathtt{a}, d, m$ and $h \leq \max\{h_1, h_2\}$,
$\mathcal{V}^{(0)}_{\mathrm{co}+(\Delta+1)}$ of sets $\mathrm{V}^{(0)}_{\mathrm{co}+(\Delta+1)}(\mathtt{a}, d-1, m'', p; \boldsymbol{x}^*)$ for $p \leq 2$,
for $2 \leq q_i \leq \delta_i, i = 1, 2$, families $\mathcal{V}^{(q_i)}_{\mathrm{end},i}(\mathtt{a}^e_i, m^e_i)$ of vector sets $\mathrm{V}^{(q_i)}_{\mathrm{co}+1,\Delta^e_i}(\mathtt{a}_i, d_i, m_i, \mathtt{a}^e_i, 1, m^e_i, h_i; \boldsymbol{x}^*)$. */
Output: A set of feasible pairs $\boldsymbol{y}_i, i = 1, 2$ of length $\delta_i, i = 1, 2$, respectively,
two vertex-labeled and edge-labeled DAG representation of these feasible pairs of e-component,
and DAG representations of frequency vector of each non-core part of the e-component
with frequency vector $\boldsymbol{x}^*$.
$F := \emptyset$; /* to store feasible pairs for core part */
$G_i := (N_i, A_i)$; $A_i := \emptyset$; $N_i := \emptyset, i = 1, 2$;/* core part */
for each $(\mathtt{a}_i, d_i, m_i), i = 1, 2$
for each $\gamma = (\mathtt{a}_1\{d_1 + 1\}, \mathtt{a}_2\{d_2 + 1\}, m) \in \Gamma^{\mathrm{co}}$ with
$m \in [1, \min\{3, \mathrm{val}(\mathtt{a}_1) - m_1, \mathrm{val}(\mathtt{a}_2) - m_2\}]$ **do**
Let L_1 denote the sorted list of vectors in $\mathrm{V}^{(\delta_1)}_{\mathrm{co}+1,\Delta^e_1}(\mathtt{a}_1, d_1, m_1, \mathtt{a}^e_1, 1, m^e_1, h_1; \boldsymbol{x}^*)$;
Construct the set $\overline{\mathrm{W}} := \{\overline{\boldsymbol{z}} \mid \boldsymbol{z} \in \mathrm{V}^{(\delta_2)}_{\mathrm{co}+1,\Delta^e_2}(\mathtt{a}_2, d_2, m_2, \mathtt{a}^e_2, 1, m^e_2, h_2; \boldsymbol{x}^*)\}$ of the γ-complement vectors;
Sort the vectors in $\overline{\mathrm{W}}$ to obtain a sorted list L_2;
Merge L_1 and L_2 into a single sorted list L_γ of vectors in both lists (as a multiset);
Trace the list L_γ and for each consecutive pair $\boldsymbol{z}^1, \boldsymbol{z}^2$ of vectors with $\boldsymbol{z}^1 = \boldsymbol{z}^2$
$\boldsymbol{y}_1 := \boldsymbol{z}^1, \boldsymbol{y}_2 := \overline{\boldsymbol{z}^2}$ is a feasible pair;
$N_i := N_i \cup \{(\boldsymbol{y}_i, \delta_i; \mathtt{a}_i, d_i, m_i, h_i)$;
Let the label of the arc from $\boldsymbol{y}_1$ to $\boldsymbol{y}_2$ is $(\mathbf{0}, m)$;
$F := F \cup \{(\boldsymbol{y}_1, \boldsymbol{y}_2; \mathbf{0}, m'; \mathtt{a}_1, d_1, m_1, h_1; \mathtt{a}_2, d_2, m_2, h_2)\}$
end for
end for;

$\mathcal{C} := \emptyset$;
/* a set of vectors of rooted core subtrees for which we calculate DAG in second phase */
$G' := G_2$;
for each $\ell \in (\delta_2, \ldots, 1)$ **do**
  $(G'' := (N'', A''), \mathcal{D}) :=$ CoreDAGSublayer$(\mathcal{V}^{(\ell-1)}_{\mathrm{co}+1,2}, G', \ell-1, \mathcal{V}^{(0)}_{\mathrm{co}+2}, h_2)$;
    $N_2 := N_2 \cup N''$; $A_2 := A_2 \cup A''$; $\mathcal{C} := \mathcal{C} \cup \mathcal{D}$
end for;
$G' := G_1$;
for each $\ell \in (\delta_1, \ldots, 1)$ **do**
  $(G'' := (N'', A''), \mathcal{D}) :=$ CoreDAGSublayer$(\mathcal{V}^{(\ell-1)}_{\mathrm{co}+1,1}, G', \ell-1, \mathcal{V}^{(0)}_{\mathrm{co}+2}, h_1)$;
    $N_1 := N_1 \cup N''$; $A_1 := A_1 \cup A''$; $\mathcal{C} := \mathcal{C} \cup \mathcal{D}$
end for;
for each $(\boldsymbol{y}, \mathtt{a}, d, m, t) \in \mathcal{C}$ **do**
  $G''' := (N''', A''')$; $N''' := \{\boldsymbol{y}\}$; $A''' := \emptyset$;
  for each $\ell \in (t-2, \ldots, 1)$ **do**
    if $\ell = t-2$ **then**
      $G^* := (N^*, A^*) :=$ DAGSublayer$(\mathcal{V}^{(0)}_{\mathrm{co}+2}(\ell-1, \boldsymbol{y}), G''', \ell-1, \mathcal{V}^{(0)}_{\mathrm{co}+\Delta+1}(\mathtt{a}, d, m; \boldsymbol{y}))$,
       where $\mathcal{V}^{(0)}_{\mathrm{co}+2}(\ell-1, \boldsymbol{y})$ is a family of vectors of end-subtrees under $\boldsymbol{y}$
       with core height $\ell-1$,
       $\mathcal{V}^{(0)}_{\mathrm{co}+\Delta+1}(\mathtt{a}, d, m; \boldsymbol{y})$ is the family of sets $\mathrm{V}^{(0)}_{\mathrm{co}+\Delta+1}(\mathtt{a}, d, m, p; \boldsymbol{y})$;
       $N''' := N''' \cup N^*$; $A''' := A''' \cup A^*$;
    else /* $\ell < t-2$ */
      $G^* := (N^*, A^*) :=$ DAGSublayer$(\mathcal{V}^{(0)}_{\mathrm{co}+2}(\ell-1, \boldsymbol{y}), G''', \ell-1, \mathcal{V}_{\mathrm{inl}})$,
       where $\mathcal{V}^{(0)}_{\mathrm{co}+2}(\ell-1, \boldsymbol{y})$ is a family of vectors of end-subtrees under $\boldsymbol{y}$
       with core height $\ell-1$;
       $N''' := N''' \cup N^*$; $A''' := A''' \cup A^*$;
    end if
  end for;
  Output $(\boldsymbol{y}, G'''')$
end for;
Output $G_i, i = 1, 2$ as DAG representations and the set F.

CoreDAGSublayer$(\mathcal{V}, G, \ell, \mathcal{V}', h)$

Input: A family $\mathcal{V}'$ of vectors rooted core-subtrees with a root label $\mathtt{a}_1$,
  degree d_1 and multiplicity m_1 and core height $t \le h$, $G = (N, A)$,
    a family $\mathcal{V}$ of vectors of bi-rooted core subtrees with core height at most h and
    ℓ (the height of the layer that we add in G at this stage).
  Output: A DAG G' that is a super-graph of G, and a set of vectors of rooted core subtrees.
  $G' := G$; $\mathcal{C} := \emptyset$;
  for each $\boldsymbol{y}_1 \in \mathcal{V}$ **do**
    for each $\boldsymbol{y}'_1 \in \mathcal{V}'$ **do**
      Let the height of $\boldsymbol{y}_1$ and $\boldsymbol{y}'_1$ be t and t', respectively;

if there exists $\gamma \in \Gamma^{\text{in}}$ and some $\boldsymbol{y}_2 \in N$ such that
$\boldsymbol{y}_i, i = 1, 2$ are feasible, i.e., $\boldsymbol{y}_1 + \boldsymbol{y}'_1 + 1_\gamma = \boldsymbol{y}_2$ and $\max\{t, t'\} = h$ **then**
if $\boldsymbol{y}_1 \notin N$ **then** $N := N \cup \{(\boldsymbol{y}_1, \ell; \mathtt{a}_1, d_1, m_1, t')\}$;
$A := A \cup \{a_{\boldsymbol{y}_2 \boldsymbol{y}_1}\}$ and
label the arc from $\boldsymbol{y}_2$ to $\boldsymbol{y}_1$ by $(\boldsymbol{y}', m)$,
where m is the bond multiplicity in γ;
$\mathcal{C} := \mathcal{C} \cup \{(\boldsymbol{y}'_1, \mathtt{a}_1, d_1, m_1, t')\}$
end if
end for
end for;
Output G' as a required DAG and $\mathcal{C}$ the required set of rooted core subtrees.

10 A Complete Algorithm to Compute Target e-components

We briefly summarize how to use the procedures described thus far to obtain an algorithm. Our global constants are a frequency vector $\boldsymbol{x}^*_{\mathrm{e}}$ of an e-component, two fixed tuples $(\mathtt{a}^e_j, m^e_j, \Delta^e_j), j = 1, 2$ a lower bound $\mathrm{ch}_{\mathrm{LB}}(e)$ and an upper bound $\mathrm{ch}_{\mathrm{UB}}(e)$ on core height, where we take $\rho = 2$.

CompleteAlgorithmEdge(Global constants: $\mathtt{a}^e_j, m^e_j, \Delta^e_j, \boldsymbol{x}^*_e$, core height bounds)

Γ^{in}_e := The set internal edges in $\boldsymbol{x}^*_e$;
Compute $\mathrm{V}^{(0)}_{\mathrm{co}+(\Delta+1)}(\mathtt{a}, d, m, h; \boldsymbol{x}^*_e)$ for each
  $\Delta \in [2, 3]$, $\mathtt{a} \in \Lambda$, $d \in [0, \mathrm{val}(\mathtt{a}) - \Delta]$, $m \in [d, \mathrm{val}(\mathtt{a}) - \Delta]$, $h \in [0, \min\{2, \mathrm{ch}_{\mathrm{UB}}(e)\}]$;
Compute $\mathrm{V}^{(0)}_{\mathrm{end}}(\mathtt{a}, d, m; \boldsymbol{x}^*_e)$ for each $\mathtt{a} \in \Lambda$, $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$, $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$;
Compute $\mathrm{V}^{(0)}_{\mathrm{inl}}(\mathtt{a}, d, m; \boldsymbol{x}^*_e)$ for each $\mathtt{a} \in \Lambda$, $d \in [0, \mathrm{val}(\mathtt{a}) - 2]$, $m \in [d, \mathrm{val}(\mathtt{a}) - 2]$;
Compute $\mathrm{V}^{(h)}_{\mathrm{end}}(\mathtt{a}, d, m; \boldsymbol{x}^*_e)$ for each $\mathtt{a} \in \Lambda$, $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$,
  $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$, $1 \leq h \leq \min\{|\Gamma^{\mathrm{in}}_e| - 1, \mathrm{ch}_{\mathrm{UB}}(e) - 2 - 1\}$ if $\mathrm{ch}_{\mathrm{UB}}(e) > 2$;
Compute $\mathrm{V}^{(0)}_{\mathrm{co}+\Delta}(\mathtt{a}, d, m, h; \boldsymbol{x}^*_v)$ for each $\Delta \in [2, 3]$, $\mathtt{a} \in \Lambda$, $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$,
  $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$, $h \leq \min\{|\Gamma^{\mathrm{in}}_e| + 2, \mathrm{ch}_{\mathrm{UB}}(e)\}$, if $\mathrm{ch}_{\mathrm{UB}}(e) > 2$;
Compute $\mathrm{V}^{(q)}_{\mathrm{co}+1,\Delta^e_j}(\mathtt{a}, d, m, \mathtt{a}^e_j, 1, m^e_j, h; \boldsymbol{x}^*_e)$ for fixed $(\mathtt{a}^e_j, m^e_j, \Delta^e_j)$, $\mathtt{a}, \in \Lambda$,
  integers $d \in [1, \mathrm{val}(\mathtt{a}) - 1]$, $m \in [d, \mathrm{val}(\mathtt{a}) - 1]$, $q = \Delta^e_j, j = 1, 2$;
Compute the set FP of feasible pairs $(\boldsymbol{z}, \boldsymbol{z}')$ such that $\boldsymbol{z} + \boldsymbol{z}' + \mathbf{1}_\gamma = \boldsymbol{x}^*_e$;
Compute the DAG G_1 (resp., G_2) representation of all vectors $\boldsymbol{z}$ (resp., $\boldsymbol{z}'$)
  such that $(\boldsymbol{z}, \boldsymbol{z}') \in$ FG (resp., $\boldsymbol{z}' \in$ FG);
Enumerate the set $\mathcal{P}_1$ (resp., $\mathcal{P}_2$) from sources to sinks in G_1 (resp., G_2);
for each feasible pair $(\boldsymbol{z}, \boldsymbol{z}') \in$ FG **do**
  Let $P := ((\boldsymbol{z}, \boldsymbol{z}_h, \boldsymbol{y}_h, m_h), (\boldsymbol{z}_h, \boldsymbol{z}_{h-1}, \boldsymbol{y}_{h-1}, m_{h-1}), \ldots, (\boldsymbol{z}_1, \boldsymbol{z}_0, \boldsymbol{y}_0, m_0))$;
  $P' := ((\boldsymbol{z}', \boldsymbol{z}'_{h'}, \boldsymbol{y}'_{h'}, m'_{h'}), (\boldsymbol{z}'_{h'}, \boldsymbol{z}'_{h'-1}, \boldsymbol{y}'_{h'-1}, m'_{h'-1}), \ldots, (\boldsymbol{z}'_1, \boldsymbol{z}'_0, \boldsymbol{y}'_0, m'_0))$,
  Compute DAG representation G^i (resp., $(G')^i$ of each $\boldsymbol{y}_i$ (resp., $\boldsymbol{y}'_i$).
  Get a target e-component by using the trees corresponding to
    $\boldsymbol{y}_h, \boldsymbol{y}_{h-1}, \ldots, \boldsymbol{y}_0, \boldsymbol{y}'_{h'}, \ldots, \boldsymbol{y}'_0$;
  Get the number of target e-components obtained by paths P and P' as
    $(n(\boldsymbol{y}_h) \times \cdots \times n(\boldsymbol{y}_0)) \times (n(\boldsymbol{y}'_{h'}) \times \cdots \times n(\boldsymbol{y}'_0))$,
    where $n(\boldsymbol{y}_i)$ (resp., $n(\boldsymbol{y}'_i)$) denote the number of graphs that can be obtained
    from $\boldsymbol{y}_i$ (resp., $\boldsymbol{y}'_i$) as explained in CompleteAlgorithmVertex
end for.