First, let us discuss the consequences of the Theorems 1.7 and 1.8 to the construction of Level $\ell+1$.

Let $P = P', [x,B], [u,A]$ be a shortest strongly simple path from s to $[u,A]$ of length $\ell+1$ with the property

$$\text{level}([x,B]) < \ell.$$

Consider the last node $[v,B]$ on P' such that

$$|P_1, [v,B]| = \text{level}([v,B]$$

where $P' = P_1, [v,B], [w,A], P_2$.

Then Theorem 1.7 implies

$$\text{DOM}([w,A]) = \text{DOM}([w,B]) = [u,B].$$

Furthermore, since P is strongly simple

$$[u,B] \notin P_1, [v,B]$$

$$\Rightarrow \text{DOM}([v,B]) \neq [u,B].$$

On the other side, let $([v,B],[w,A]) \in E_M$ such that

i) $\text{DOM}([v,B]) \neq \text{DOM}([w,A]) = \text{DOM}([w,B]) = [u,B]$

ii) $\text{level}([v,B]) + \text{level}([w,B]) - \text{level}([u,B]) = \ell.$

Then Theorem 1.8 implies

1) There exists a shortest strongly simple path P_1 from s to $[v,B]$ with $[u,B] \notin P_1$.

2) Let P_1 be any shortest strongly simple path from s to $[v,B]$ with $[u,B] \notin P_1$ and let

$$Q = Q_1, [u,B], Q_2, [w,B]$$

be any shortest strongly simple path from s to $[w,B]$. Then

$$P = P_1, [w,A], r(Q_2), [u,A]$$

is a strongly simple path from s to $[u,A]$ of length $l+1$.

04.06.

Note that Theorem 1.8 does not assume that $[v,B]$ is the last node on P such that the length of the subpath from s to $[v,B]$ on P is equal level$([v,B])$.

Hence, as long as $DOM([v,B]) \neq DOM([u,A]) = DOM([w,B])$, the Theorem 1.8 can be applied with respect to the edge $([v,B],[w,A])$ and $DOM([w,A])$. This will be have some consequences with respect to an efficient implementation of MBFS.

On the other side, Theorem 1.7 states that on all shortest strongly simple paths

$$P = P', [x,B], [u,A]$$

from s to $[u,A]$ with $level([x,B]) < level([u,A])-1$ an edge characterized by Theorem 1.7 exists such that this edge can be used for the correct construction of the layered graph $\overline{G}_M$.

Note that an edge $([v,B],[w,A])$ can be an edge characterized by Theorem 1.7 at most once. By the definition of $DOM([w,A])$, after the termination of Phase $level([u,A])$, the node $[u,A]$ would be on all considered shortest strongly simple paths R which contains the edge $([v,B],[w,A])$ and the length of the subpath from s to $[u,A]$ of R would be equal $level([u,A])$. Hence, the node $[v,B]$ cannot be the last node on R such that the length of the subpath from s to $[v,B]$ of R is equal $level([v,B])$.

Note that the DOM-values are updated after the end of Part 2 of a phase. This means that all DOM-values used during Part 2 of Phase $l+1$ are the same as the values at the beginning of Phase $l+1$.

Altogether, Phase $l+1$ can be performed in the following way:

Phase $\ell+1$:

The phase separates into two parts.

Part 1:

For all nodes $[v,B]$ such that level$([v,B]) = \ell$ consider all edges $([v,B],[w,A]) \in E_M$.

- The node $[w,A]$ is inserted into Level $\ell+1$ and the edge $([v,B],[w,A])$ is added to $\overline{E}_M$

$$\Longleftrightarrow$$

$$\text{level}([w,A]) > \ell \text{ and } \text{DOM}([v,B]) \neq [w,B].$$

Part 2:

Consider all edges $([v,B],[w,A]) \in E_M$ such that

i) $\text{DOM}([v,B]) \neq \text{DOM}([w,A]) = \text{DOM}([w,B])$

(Assume $[u,B] = \text{DOM}([w,A])$.)

ii) $\text{level}([v,B]) + \text{level}([w,B]) - \text{level}([u,B]) = \ell$

iii) The edge $([v,B],[w,A])$ has been not considered during Part 2 of a previous phase.

An edge satisfying i) – iii) <u>fulfills the edge properties for Phase $\ell+1$.</u>

For all such edges $([v,B],[w,A]) \in E_M$,

- insert the node $[u, A]$ where $[u, B] = \mathrm{DOM}([w, A])$ into Level $l+1$ and the <u>indexed edge</u>

$$([v, B], [w, A])_{[u, A]}$$

into $\overline{E_M}$.

The index $[u, A]$ indicates that the edge $([v, B], [w, A])$ is used on the shortest strongly <u>simple</u> paths from s to $[u, A]$ characterized by Theorem 1.8.

Theorem 1.8 $\Rightarrow$

All subpaths from $[w, A]$ to $[u, A]$ of such a path can easily be constructed by taking the backpath $r(S)$ of the corresponding strongly simple path $\underline{S} = [u, B], \ldots, [w, B]$ in the layered graph G_M.

In addition to the construction of the $(l+1)$st level, MBFS has to update some $\mathrm{DOM}([v, X])$.

After the definition of Level $l+1$, $\mathrm{DOM}([u, A])$ for all nodes $[u, A]$ with $\mathrm{level}([u, A]) = l+1$ has to be defined. For doing this, let

$$\mathrm{DOM}^i([v, X]) := \begin{cases} \mathrm{DOM}([v, X]) & \text{if } i = 0 \\ \mathrm{DOM}(\mathrm{DOM}^{i-1}([v, X])) & \text{if } i > 0 \end{cases}$$

and

$$DOM^*([v,X]) := \bigcup_{i \geq 0} DOM^i([v,X]).$$

Let

$$Pred([u,A]) := \{ [v,B] \mid ([v,B],[u,A]) \in \overline{E}_M \text{ or } ([v,B],[w,A])_{[u,A]} \in \overline{E}_M \text{ for some } [w,A] \in V' \}.$$

Note that $Pred([u,A])$ contains exactly those nodes $[v,B]$ which are the first node of those edges and those indexed edges which are added to $\overline{E}_M$ because of the definition of $level([u,A])$.

Let $DOM(Pred([u,A]))$ denotes that node $x \in V'$ such that

a) $x \in \bigcap_{[v,B] \in Pred([u,A])} DOM^*([v,B])$

and

b) $level(y) \leq level(x)$ for all $y \in V' \setminus \{x\}$ which satisfy a).

Note that node x exists since $s \in \bigcap_{[v,B] \in Pred([u,A])} DOM^*([v,B])$.

By the definition of $DOM([u,A])$ and by Lemma 1.6, at the beginning of Phase $level([u,A]) + 1$ there holds

$$\mathrm{DOM}([u,A]) = \mathrm{DOM}(\mathrm{Pred}([u,A])).$$

MBFS updates $\mathrm{DOM}([v,X])$, $[u,X] \in V'$ as follows:

- After the definition of level$([u,B])$

$$\mathrm{DOM}([u,B]) := \mathrm{DOM}([v,A])$$

where $[v,A]$ is the unique node with $([v,A],[u,B]) \in E_M$ (i.e., $(v,u) \in M$).

- After the definition of level$([u,A])$

$$\mathrm{DOM}([v,X]) := \mathrm{DOM}([u,B])$$

for all $[v,X]$ with $\mathrm{DOM}([v,X]) = [u,B]$.

- After the definition of Level $l+1$

$$\mathrm{DOM}([u,A]) := \mathrm{DOM}(\mathrm{Pred}([u,A]))$$

for all $[u,A]$ with level$([u,A]) = l+1$.

Two questions have to be answered.

1. How to compute $\mathrm{DOM}(\mathrm{Pred}([u,A]))$?
2. How to compute the edges which fulfill the edge properties for Phase $l+1$?

The computation of DOM(Pred([u,A])

Our goal is to compute all these values in time $O(m+n)$. For doing this, the following data structure is very useful:

The DOM-tree DT is a rooted tree which is defined inductively with respect to the performance of the algorithm MBFS.

i) At the beginning, the DOM-tree DT is only the root s.

ii) After the definition of $level([v,x])$, $[v,x] \in V'$, the leaf $[v,x]$ is added to DT as a son of the node $DOM([v,x])$.

~~For~~ the computation of $DOM(Pred([u,A]))$, the following operation on a rooted tree is needed:

Let T be a rooted tree and x, y be two nodes of T. The nearest common ancestor $nca(x,y)$ of x and y in T is the common ancestor of x and y with largest depth. Let $\{x_1, x_2, \ldots, x_t\}$ be a set of nodes of T. The nearest common ancestor $nca(x_1, x_2, \ldots, x_t)$ of $x_1, x_2, \ldots, x_t$ is the common ancestor of $x_1, x_2, \ldots, x_t$ with largest depth.

Let $nca(Pred([u,A])) = z$. By the definition of $DOM(Pred([u,A]))$ and Lemma 1.6 we obtain

$$DOM(Pred([u,A])) = \begin{cases} [x,B] & \text{if } z = [x,B] \text{ and } level([x,A]) \text{ is not defined} \\ DOM(z) & \text{otherwise} \end{cases}$$

Note $level([x,A])$ not defined $\Rightarrow DOM([x,B]) = [x,B]$.

Note that $nca(Pred([u,A]))$ can be computed with $|Pred([u,A])| - 1$ nca-operations on two nodes in DT. Hence, we need a data structure which supports the following operations:

1) $nca(x,y)$: Return the nearest common ancestor of x and y in DT.

2) $add\_leaf(x,y)$: Add a new leaf y as a son of x to DT.

<u>H. N. Gabow</u>, Data structures for Weighted Matching and Nearest Common Ancestors with Linking, 1st SODA, 1990, 434 – 443.

shows how to perform k add_leaf- and nca-operations in $O(k)$ time.

$\Rightarrow$

DOM(Pred([u,A])) can be computed in $O(|\mathrm{Pred}([u,A])|$ time plus the time needed for the computation of DOM(z).

If we use for the manipulation of the DOM-sets

- disjoint set union

then we can bound the time used for all computations of DOM-values by

- $O(m + n \log n)$ using the weighted union heuristic

or by

- $O(m+n)$ using incremental tree set union.

<u>The computation of the edges which fulfill the edge properties for Phase $l+1$</u>:

Note that for an edge ([v,B], [w,A]) which fulfills the edge property for Phase $l+1$ holds

$$\mathrm{level}([v,B]) > \mathrm{level}([w,A]).$$

The edge is used for the construction of shortest strongly simple paths from s to DOM([w,A]) = [u,B] as described by Theorem 1.8.

By the definition of the DOM-values, for all subsequent phases there holds

$$\mathrm{DOM}([w,A]) = \mathrm{DOM}([w,B]).$$

Theorem 1.8 $\Rightarrow$

As long as

$$\mathrm{DOM}([v,B]) \neq \mathrm{DOM}([w,A])$$

the edge $([v,B],[w,A])$ can be used for the construction of shortest strongly simple paths from s to $[z,A]$ where $[z,B] = \mathrm{DOM}([w,A])$. But we does not need this edge for the construction.

Nevertheless, we can use the fact

$$\mathrm{DOM}([v,B]) \neq \mathrm{DOM}([w,A])$$

for the exclusion of other edges to fulfill the edge properties for a certain phase.

If there is an edge $([v',B],[w',A])$ such that

a) $\mathrm{DOM}([v',B]) \neq \mathrm{DOM}([w',A]) = \mathrm{DOM}([w',B])$ and
b) $\mathrm{DOM}([w',A]) = \mathrm{DOM}([w,A])$ but
c) $\mathrm{level}([v',B]) + \mathrm{level}([w',B]) > \mathrm{level}([v,B]) + \mathrm{level}([w,B])$

then the edge $([v',B],[w',A])$ cannot be used for a shortest strongly simple path from s to $[z,A]$ where $[z,B] = \mathrm{DOM}([w',A])$ since a path which uses the edge $([v,B],[w,A])$ would be shorter.

But it is possible that the edge can be used for a shortest strongly simple path to a node $[y, A]$ where $[y, B]$ is a node between

$$nca([v, B], DOM([w, A])) \text{ and } nca([v', B], DOM([w, A])).$$

For abbreviation let

$$Val([v, B], [w, A]) = level([v, B]) + level([w, B])$$

where $([v, B], [w, A])$ is an edge in E_M.

Note that the edge property for these level$([u, A])$ $[u, B] = DOM([w, A])$ states that at the moment when level$([u, A])$ is defined

$$DOM([v, B]) \neq DOM([w, A]) = DOM([w, B]).$$

This does not imply that this property has to be fulfilled earlier, e.g. at the moment when the edge $([v, B], [w, A])$ is considered by MBFS or at the moment when level$([w, B])$ is defined.

A considered edge $([v, B], [w, A])$ is called <u>dead</u> if we can exclude that $([v, B], [w, A])$ will fulfill the edge properties in the future. Otherwise, the considered edge $([v, B], [w, A])$ is called <u>living</u>.

Our goal is to maintain a data structure which contains the living edges. Note that the first time where $Val([v, B], [w, A])$, $([v, B], [w, A]) \in E_M$ can be computed is

$\min\{ level([v,B])+1, \ level([w,B]) \}$.

We compute the value $Val([v,B],[w,A])$ at the moment when we

- consider the edge $([v,B],[w,A])$

  if $level([v,B])+1 > level([w,B])$

- compute $level([w,B])$

  if $level([v,B])+1 < level([w,B])$.

Question:

At the moment when $Val([v,B],[w,A])$ is computed, what is the first phase such that $([v,B],[w,A])$ fullfills the edge properties for this phase?

At that phase there holds

$$DOM([w,A]) = DOM([w,B])$$

$\Longrightarrow$

$([v,B],[w,A])$ cannot fulfill the edge properties before

Phase $Val([v,B],[w,A]) - level(nca([w,A],[w,B]))+$

Furthermore, at that phase

$DOM([v,B]) \neq DOM([w,A]) = DOM([w,B])$

Hence,

$level(nca([v,B],[w,A])) > level(nca([w,A],[w,B]))$

implies that the edge $([v,B],[w,A])$ is dead.

$\Rightarrow$

$([v,B],[w,A])$ living $\Rightarrow$

$level(nca([v,B],[w,A])) < level(nca([w,A],[w,B]))$.

$\leadsto$

At the moment when $Val([v,B],[w,A])$ is computed, we also compute

- $nca([w,A],[w,B])$ and then $level(nca([w,A],[w,B]))$,
- $nca([v,B],[w,A])$ and then $level(nca([v,B],[w,A]))$.

If $level(nca([w,A],[w,B])) > level(nca([v,B],[w,A]))$ then the edge $([v,B],[w,A])$ is stored in an appropriate data structure.

<u>Question:</u>

What is an appropriate data structure?