# 1 Indexings as Proxies for Partitions

**Definition 1.1.** *Let $V = \{v_1, v_2, \ldots, v_n\}$ be a set of $n$ vertices. An indexing of $V$ is a function $\varphi : V \to \{1, \ldots, n\} = [n]$ that associates every vertex with a number from 1 to $n$.*

The overall idea is to use this concept of indexings to simplify and speed up operations on partitions. Every vertex is mapped to a number (index) which indicates the set in the partition we want this vertex to be a part of. If multiple vertices are mapped to the same index, they will be part of the same set in the partition. Since there are at maximum $n$ sets in a partition of $V$ (every vertex is assigned its own set, i.e. $\varphi$ is bijective), we restrict to indices between $1$ and $n$. Note that every indexing can be stored in a linear amount of memory dependent on the size of $V$.

**Definition 1.2.** *Let $\varphi$ be an indexing of $V$. The partition induced by $\varphi$ is defined as*

$$\Pi(\varphi) = \{\varphi^{-1}(\varphi(v)) : v \in V\}, \tag{1.1}$$

*where $\varphi^{-1}(\varphi(v)) = \{w \in V : \varphi(w) = \varphi(v)\}$.*

Based on this definition, we are able to obtain some immediate results. First, we are able to use indexings as representations for partitions, i.e. for every partition, there is some indexing that yields that very partition (Lemma 1.1). And two, we obtain a criteria that makes it possible to check when two vertices share the same set in the partition (Lemma 1.2).

**Lemma 1.1.** *A set $\Pi$ is a partition of $V$ if and only if there exists an indexing of $V$ that induces $\Pi$.*

**Lemma 1.2.** *Let $\varphi$ be an indexing of $V$. For two vertices $v_1, v_2 \in V$, we get $\varphi(v_1) = \varphi(v_2)$ if and only if $v_1$ and $v_2$ are part of the same set in $\Pi(\varphi)$.*

Since we want to use indexings in order to define transformations on partitions, we are interested in the question when two indexings are "equal", in the sense that they induce the same partition. This is characterized in part by Theorem 1.3.

**Theorem 1.3.** *If $\varphi, \varphi'$ are two indexings of $V$, then the following statements are equivalent:*

*(a)* $\Pi(\varphi) = \Pi(\varphi')$

*(b) there is a bijection $\lambda : \mathrm{image}(\varphi) \to \mathrm{image}(\varphi')$[1] such that for all vertices $v$, $\lambda(\varphi(v)) = \varphi'(v)$*

*(c) for all vertices $v, w$, $\varphi(w) = \varphi(v)$ if and only if $\varphi'(w) = \varphi'(v)$*

If the task is to determine whether two induced partitions $\Pi(\varphi)$, $\Pi(\varphi')$ for given indexings $\varphi, \varphi'$ of $V$ are equal, the result of Theorem 1.3 might come in handy: instead of explicitly computing the resulting partitions and checking if both sets are equal, one can simply determine whether a fitting bijection $\lambda$ exists, which is arguably easier. In fact, one can construct $\lambda$ as in the first part of the proof of Lemma 1.3 and then check whether the result is a bijection or not.

# 2 Move-Operation on Indexings

We will now define the "move"-operation on indexings, which transforms an indexing into another indexing by changing the assigned index of a vertex. The effect this has on the induced partition is that the respective vertex is "taken" from its original set[2] and then "put into" some

[1] $\mathrm{image}(\varphi)$ means the image of $\varphi$, i.e. $\mathrm{image}(\varphi) = \{\varphi(v)\}_{v \in V}$.
[2] i.e. the set $\varphi^{-1}(\varphi(v))$

other set[3]. This is then defined as $\mathrm{move} : [n]^V \times V \times [n] \to [n]^V$ with

$$\mathrm{move}(\varphi, v, k)(u) = \begin{cases} \varphi(u) & u \neq v, \\ k & u = v \end{cases} \tag{2.1}$$

for every vertex $u$. The operation takes as input an indexing $\varphi$, a vertex $v$ and new index $k$ and outputs a new indexing that is essentially the same as $\varphi$, with the only difference being that $v$ is mapped to $k$ instead of whatever it was mapped to before.

In many cases, there are multiple ways of moving one vertex to different indices while still inducing the same partition afterwards. For example, if $v$ is an arbitrary vertex, $\varphi$ is an indexing of $V$ with $\varphi^{-1}(k_1) = \cdots = \varphi^{-1}(k_m) = \emptyset$ and $k_1, \ldots, k_m$ are pairwise different, then moving $v$ to $k_1, \ldots, k_m$ yields that while $\mathrm{move}(\varphi, v, k_1), \ldots, \mathrm{move}(\varphi, v, k_m)$ are all different indexings, their induced partitions are the same: $\Pi(\mathrm{move}(\varphi, v, k_1)) = \cdots = \Pi(\mathrm{move}(\varphi, v, k_m))$.
Based on this observation we are interested in an efficient way of enumerating all possible "moves" of a vertex with respect to the induced partitions without enumerating too much or having to double-check whether two move-operations induce the same partition. Hence, consider algorithm 1, which aims at finding a solution to this problem.

**Algorithm 1:** Move-Enumeration

**Input:** Set of vertices $V$ with indexing $\varphi$
**Result:** Sequence of indexings $\varphi_1, \varphi_2, \ldots, \varphi_m$

Set $<$ to be some linear order on $V$ ;
Set $\mathcal{N} := \{1, \ldots, n\} \backslash \mathrm{image}(\varphi)$ ;
**forall** *vertices* $v \in V$ **do**
  Set $U_v := \varphi^{-1}(\varphi(v))$ ;
  **forall** $k \in \mathrm{image}(\varphi) \backslash \{\varphi(v)\}$ **do**
    **if** $|U_v| > 1$ *or* $|\varphi^{-1}(k)| > 1$ **then**
      enumerate $\mathrm{move}(\varphi, v, k)$ ;
    **else if** $U_v = \{v\}$ *and* $\varphi^{-1}(k) = \{u\}$ *and* $v < u$ **then**
      enumerate $\mathrm{move}(\varphi, v, k)$ ;
  **if** $\mathcal{N} \neq \emptyset$ **then**
    **if** $|U_v| > 2$ *or* $U_v = \{v, u\}$ *and* $v < u$ **then**
      Let $k \in \mathcal{N}$ ;
      enumerate $\mathrm{move}(\varphi, v, k)$ ;

We want to check three important properties: one, algorithm 1 only enumerates indexings which yield partitions distinct from all induced partitions of the other indexings (the "not too much"-part, see Lemma 2.1), two, every partition that is induced when moving one vertex in $\varphi$ to a different set is induced by some indexing in $\varphi_1, \ldots, \varphi_m$ (Lemma 2.2) and three, the partition that is induced by $\mathrm{move}(\varphi, v, k)$ for any vertex $v$ and number $k$ is not the same that is induced by $\varphi$ (Lemma 2.3).

**Lemma 2.1** (Pairwise Distinctiveness). *Let $\varphi$ be an indexing of $V$ and $\varphi_1, \ldots, \varphi_m$ be a sequence generated by algorithm 1 on input $V$ and $\varphi$. Then for all pairwise distinct $i, j \in \{1, \ldots, m\}$, we have $\Pi(\varphi_i) \neq \Pi(\varphi_j)$.*

**Lemma 2.2** (Completeness). *Let $\varphi$ be an indexing of $V$ and $\varphi_1, \ldots, \varphi_m$ be the sequence that is generated by algorithm 1 on input $V$ and $\varphi$. For all vertices $v$ and $k \in \{1, \ldots, n\}$ such that $\Pi(\mathrm{move}(\varphi, v, k)) \neq \Pi(\varphi)$, there is $i \in \{1, \ldots, m\}$ such that $\Pi(\mathrm{move}(\varphi, v, k)) = \Pi(\varphi_i)$.*

[3]i.e. the set $\varphi^{-1}(k)$, where $k$ is some number between 1 and $n$

**Lemma 2.3.** *Let $\varphi$ be an indexing of $V$ and $\varphi_1, \ldots, \varphi_m$ be the sequence that is generated by algorithm 1 on input $V$ and $\varphi$. Then $\Pi(\varphi) \neq \Pi(\varphi_i)$ for all $i \in \{1, \ldots, m\}$.*

**Corollary 2.3.1.** *For an indexing $\psi$ of $V$, the following statements are equivalent:*

- $\psi = \text{move}(\varphi, v, k)$ *and* $\Pi(\psi) \neq \Pi(\varphi)$ *for* $v \in V, 1 \leq k \leq n$
- $\Pi(\psi) \in \{\Pi(\varphi_1), \ldots, \Pi(\varphi_m)\}$.

*Proof.* From top to bottom, simply apply Lemma 2.2. For the direction from bottom to top, apply Lemma 2.3. □

# A Proofs for Section 1 (Indexings as Proxies for Partitions)

**Lemma 1.1.** *A set $\Pi$ is a partition of $V$ if and only if there exists an indexing of $V$ that induces $\Pi$.*

*Proof.* First, we will show the direction from left to right. Let $\Pi = \{U_1, \dots, U_m\}$, $1 \leq m \leq n$, be a partition of $V$. For all vertices $v$ with corresponding set $U_i$ in $\Pi$ (of which there is exactly one), we define $\varphi(v) = i$. For a $j \in \{1, \dots, m\}$ we then obtain $\varphi^{-1}(j) = U_j$. But then $\Pi(\varphi) = \Pi$, i.e. $\Pi$ is induced by $\varphi$.
The other direction requires us to prove that every indexing induces a partition of $V$. I.e., for an indexing $\varphi$ of $V$ with $\Pi(\varphi) = \Pi$, we need to check the following requirements:

1. $\emptyset \notin \Pi$: This is obvious from (1.1), since every element $\varphi^{-1}(\varphi(v)) \in \Pi$ contains at least $v$.
2. $\bigcup_{U \in \Pi} U = V$: "$\subseteq$" is obvious. For "$\supseteq$", take a vertex $v$. Then, $v \in \varphi^{-1}(\varphi(v))$ and since $\varphi^{-1}(\varphi(v)) \in \Pi(\varphi)$, we obtain $v \in \bigcup_{U \in \Pi} U$.
3. if $U_1, U_2 \in \Pi$ and $U_1 \neq U_2$, then $U_1 \cap U_2 = \emptyset$: Take two $U_1, U_2 \in \Pi$ with $U_1 \neq U_2$ such that $U_1 = \varphi^{-1}(\varphi(w))$ and $U_2 = \varphi^{-1}(\varphi(u))$ for two vertices $w$ and $u$. Assume for a contradiction that $U_1 \cap U_2 \neq \emptyset$, i.e. there exists $v \in U_1 \cap U_2$. But then $v \in \varphi^{-1}(\varphi(w))$ and $v \in \varphi^{-1}(\varphi(u))$. Thus, $\varphi(v) = \varphi(w) = \varphi(u)$, which cannot be the case, since that would imply $U_1 = U_2$.

This completes the proof. □

**Lemma 1.2.** *Let $\varphi$ be an indexing of $V$. For two vertices $v_1, v_2 \in V$, we get $\varphi(v_1) = \varphi(v_2)$ if and only if $v_1$ and $v_2$ are part of the same set in $\Pi(\varphi)$.*

*Proof.* In order to make things more readable, set $U_i = \varphi^{-1}(\varphi(v_i))$ for a vertex $v_i$ in context of this proof.
For the direction from left to right, take $v_1, v_2 \in V$ such that $\varphi(v_1) = \varphi(v_2)$. Then $v_1, v_2 \in U_1 = U_2 \in \Pi(\varphi)$, i.e. they are part of the same set.
For the other direction, we will show the contraposition. Take $v_1, v_2 \in V$ such that $\varphi(v_1) \neq \varphi(v_2)$. Clearly, $v_1 \in U_1$ and $v_1 \notin U_2$ as well as $v_2 \in U_2$ and $v_2 \notin U_1$. Therefore $U_1 \neq U_2$. Since by Lemma 1.1, $\Pi(\varphi)$ is a partition and $U_1$ and $U_2$ are distinct sets in $\Pi(\varphi)$, $U_1$ is the only set that contains $v_1$ and vice versa for $U_2$ and $v_2$. Thus, $v_1$ and $v_2$ cannot be part of the same set in $\Pi(\varphi)$. □

**Theorem 1.3.** *If $\varphi, \varphi'$ are two indexings of $V$, then the following statements are equivalent:*

*(a)* $\Pi(\varphi) = \Pi(\varphi')$

*(b) there is a bijection $\lambda : \mathrm{image}(\varphi) \to \mathrm{image}(\varphi')$[4] such that for all vertices $v$, $\lambda(\varphi(v)) = \varphi'(v)$*

*(c) for all vertices $v, w$, $\varphi(w) = \varphi(v)$ if and only if $\varphi'(w) = \varphi'(v)$*

*Proof.* We will start with the direction from (a) to (b). Let $\varphi$, $\varphi'$ be two indexings of $V$ with $\Pi(\varphi) = \Pi(\varphi')$. We define $\lambda : \mathrm{image}(\varphi) \to \mathrm{image}(\varphi')$ as follows. For all $k \in \mathrm{image}(\varphi)$, associate a vertex $v$ such that $\varphi(v) = k$. Then define $\lambda(k) = \varphi'(v)$. It remains to show that $\lambda$ fulfills the requirements in (b):

[4] $\mathrm{image}(\varphi)$ means the image of $\varphi$, i.e. $\mathrm{image}(\varphi) = \{\varphi(v)\}_{v \in V}$.

1. $\lambda$ is bijective: Since every element in $\text{image}(\varphi)$ corresponds to a set in $\Pi(\varphi)$, and vice versa for $\text{image}(\varphi')$ and $\Pi(\varphi')$, we get $|\text{image}(\varphi)| = |\Pi(\varphi)| = |\Pi(\varphi')| = |\text{image}(\varphi')|$. Since $\text{image}(\varphi)$ and $\text{image}(\varphi')$ are also finite, it suffices to show injectivity of $\lambda$ in order to prove bijectivity. Assume for a contradiction that $\lambda$ is not injective, i.e. there are $v_1, v_2 \in V, \varphi(v_1) \neq \varphi(v_2)$, such that $\lambda(\varphi(v_1)) = \lambda(\varphi(v_2)) = \varphi'(v_1) = \varphi'(v_2)$. Thus, by application of Lemma 1.2, there must be a set in $\Pi(\varphi')$ that contains $v_1$ and $v_2$, while there is no set in $\Pi(\varphi)$ that has both vertices in it. But then $\Pi(\varphi) \neq \Pi(\varphi')$. Contradiction.
2. For all vertices $v$, $\lambda(\varphi(v)) = \varphi'(v)$: Take an arbitrary vertex $v$ and let $\varphi(v) = k \in \text{image}(\varphi)$. Let $\bar{v}$ be the vertex that was previously associated with $k$ in the definition of $\lambda$, i.e. the vertex for which $\varphi(\bar{v}) = k = \varphi(v)$ and $\lambda(\varphi(\bar{v})) = \varphi'(\bar{v})$ holds. Assume for a contradiction that $\varphi'(v) \neq \varphi'(\bar{v})$. By Lemma 1.2, this yields that there is no set in $\Pi(\varphi')$ that contains $v$ and $\bar{v}$ at once, while there is one in $\Pi(\varphi)$. But then again $\Pi(\varphi) \neq \Pi(\varphi')$, i.e. a contradiction. Thus, $\lambda(\varphi(v)) = \lambda(\varphi(\bar{v})) = \varphi'(\bar{v}) = \varphi'(v)$.

This concludes this direction. For the direction from (b) to (c), let $\lambda$ be a bijection between the images of two indexings $\varphi, \varphi'$ of $V$ that fulfills the requirements in (b). Let $w, v$ be two arbitrary vertices; then

$$\begin{aligned} \varphi(w) = \varphi(v) \quad & \text{iff} \quad \lambda(\varphi(w)) = \lambda(\varphi(v)) && (*) \\ & \text{iff} \quad \varphi'(w) = \varphi'(v) && (**) \end{aligned}$$

The first identity $(*)$ works since $\lambda$ is a bijection and the second $(**)$ since $\lambda(\varphi(w)) = \varphi'(w)$ for all vertices $w$. The remainder (c) to (a) is relatively trivial: If we assume premise (c), then

$$\begin{aligned} U \in \Pi(\varphi) \quad & \text{iff} \quad U = \varphi^{-1}(\varphi(v)) \text{ for a vertex } v \\ & \text{iff} \quad U = \{w \in V : \varphi(w) = \varphi(v)\} \text{ for a vertex } v \\ & \text{iff} \quad U = \{w \in V : \varphi'(w) = \varphi'(v)\} \text{ for a vertex } v \\ & \text{iff} \quad U = \varphi'^{-1}(\varphi'(v)) \text{ for a vertex } v \\ & \text{iff} \quad U \in \Pi(\varphi'), \end{aligned}$$

which shows $\Pi(\varphi) = \Pi(\varphi')$. □

# B Proofs for Section 2 (Move-Operation on Indexings)

**Lemma 2.1** (Pairwise Distinctiveness). *Let $\varphi$ be an indexing of $V$ and $\varphi_1, \ldots, \varphi_m$ be a sequence generated by algorithm 1 on input $V$ and $\varphi$. Then for all pairwise distinct $i, j \in \{1, \ldots, m\}$, we have $\Pi(\varphi_i) \neq \Pi(\varphi_j)$.*

*Proof.* First note that $\varphi_1, \ldots, \varphi_m$ are indeed results of a move-operation on $\varphi$. This can be checked by simple introspection on algorithm 1. Therefore, we can write $\varphi_i = \text{move}(\varphi, w_i, k_i)$ for all $i = 1, \ldots, m$, with $w_i \in V$ and $k_i \in \{1, \ldots, n\}$, i.e.: $\varphi_i$ moves vertex $w_i$ to index $k_i$. Secondly, note that every pair $(v, k) \in V \times \{1, \ldots, n\}$ is regarded at most once. This means that for every two different indexings $\varphi_i, \varphi_j$, either $w_i \neq w_j$ or $k_i \neq k_j$ holds. Now, pick $i, j \in \{1, \ldots, m\}$ such that $i \neq j$. The rest of the proof can be done via case distinction on all possible conditions under which a move-operation could be executed in the algorithm:

1. $w_i = w_j = w$. Since $i$ and $j$ are pairwise distinct, $k_i \neq k_j$ must hold. For $k_i$, we have either $k_i \in \mathcal{N}$ (line 12) or $k_i \in \text{image}(\varphi) \backslash \{\varphi(w)\}$ (line 6 and 8):

(a) If $k_i \in \mathcal{N}$: Then $\varphi_i(w) = k_i$ only for $w$ (since there is no vertex $v$ that is mapped to $k_i$ in $\varphi$). Also $k_j \notin \mathcal{N}$, since there is at maximum one $k \in \mathcal{N}$ for which $\mathrm{move}(\varphi, w, k)$ is enumerated. Thus, $k_j \in \mathrm{image}(\varphi)\backslash\{\varphi(w)\}$, i.e. there is a vertex $u$, $u \neq w$, with $\varphi(u) = k_j \neq \varphi(w)$. But then we have $\varphi_j(w) = k_j = \varphi_j(u)$ and $\varphi_i(u) = k_j \neq k_i = \varphi_i(w)$ at the same time, which implies $\Pi(\varphi_i) \neq \Pi(\varphi_j)$ by Theorem 1.3.

(b) If $k_i \in \mathrm{image}(\varphi)\backslash\{\varphi(w)\}$: Then there is a vertex $u$ with $\varphi(u) \neq \varphi(w)$ such that $\varphi(u) = k_i$. After moving $w$ to $k_i$ in $\varphi_i$, one obtains $\varphi_i(w) = \varphi_i(u) = k_i$, and after moving $w$ to $k_j$ in $\varphi_j$, one gets $\varphi_j(w) = k_j \neq k_i = \varphi_j(u)$. But that implies $\Pi(\varphi_i) \neq \Pi(\varphi_j)$ by Theorem 1.3.

2. $w_i \neq w_j$, and therefore $\varphi_i(w_j) = \varphi(w_j)$ and $\varphi_j(w_i) = \varphi(w_i)$ (i.e., $\varphi_i$ does not change the index of $w_j$ and $\varphi_j$ does not change the index of $w_i$). Again, we make the distinction for the case $k_i \in \mathcal{N}$ and $k_i \in \mathrm{image}(\varphi)\backslash\{\varphi(w)\}$:

   (a) If $k_i \in \mathcal{N}$: since $k_i \notin \mathrm{image}(\varphi)$, $\varphi(w_i) \neq k_i$. Also, at least one of the following cases (see line 10 of the algorithm) must hold:

      i. If $|U_{w_i}| > 2$, i.e. $U_{w_i} = \{w_i, u, v, \dots\}$: At least one of the vertices $u$ and $v$ must be different from $w_j$, since $u \neq v$ and $w_j$ cannot be equal to both of them. Let w.l.o.g. $u \neq w_j$. Then $\varphi_j(w_i) = \varphi(w_i) = \varphi(u) = \varphi_j(u)$ but $\varphi_i(w_i) = k_i \neq \varphi(w_i) = \varphi_i(u)$, i.e. $\Pi(\varphi_i) \neq \Pi(\varphi_j)$ by Theorem 1.3.

      ii. If $U_{w_i} = \{w_i, u\}$ and $w < u$:

         A. $w_j \neq u$: Since $w_i \neq w_j$ and $\varphi(w_i) \neq k_i$, we get $\varphi_i(w_i) \neq \varphi(u) = \varphi_i(u)$ and $\varphi_j(w_i) = \varphi(w_i) = \varphi(u) = \varphi_j(u)$. Then simply apply Theorem 1.3 and obtain $\Pi(\varphi_i) \neq \Pi(\varphi_j)$.

         B. $w_j = u$. Since $u \not< w_i$ and $U_{w_j} = U_{w_i}$, there is no possibility that line 13 is executed for $w_j$. Therefore, $k_j \notin \mathcal{N}$. But then $k_j \in \mathrm{image}(\varphi)\backslash\{\varphi(w_j)\}$, i.e. there is a vertex $v$ such that $k_j = \varphi(v) \neq \varphi(w_j)$ (this implies $v \notin U_{w_i}$). Thus, $\varphi_j(w_j) = \varphi(v) = \varphi_j(v)$ and $\varphi_i(w_j) = \varphi(w_j) \neq \varphi(v) = \varphi_i(v)$. Again, the application of Theorem 1.3 yields $\Pi(\varphi_i) \neq \Pi(\varphi_j)$.

   (b) If $k_i \in \mathrm{image}(\varphi)\backslash\{\varphi(w_i)\}$ then there is a vertex $v$ for which $k_i = \varphi(v) \neq \varphi(w_i)$ holds, and one of the following cases applies:

      i. $|U_{w_i}| > 1$, i.e. $U_{w_i} = \{w_i, u, \dots\}$.

         A. If $w_j = u$: then $\varphi_j(w_i) = \varphi(w_i) \neq \varphi(v) = \varphi_j(v)$ and $\varphi_i(w_i) = \varphi(v) = \varphi_i(v)$. Here, Theorem 1.3 can be applied, which results in $\Pi(\varphi_i) \neq \Pi(\varphi_j)$.

         B. If $w_j \neq u$: then $\varphi_j(w_i) = \varphi(w_i) = \varphi(u) = \varphi_j(u)$ and $\varphi_i(w_i) \neq \varphi(w_i) = \varphi(u) = \varphi_i(u)$. Theorem 1.3 can be applied with result $\Pi(\varphi_i) \neq \Pi(\varphi_j)$.

      ii. $|\varphi^{-1}(k_i)| > 1$, i.e. $\varphi^{-1}(k_i) = \{u, v, \dots\}$.

         A. If $w_j = u$: then $\varphi_i(w_i) = \varphi_i(v)$ but $\varphi_j(w_i) \neq \varphi_j(v)$ . In this case, Theorem 1.3 can be applied to obtain $\Pi(\varphi_i) \neq \Pi(\varphi_j)$.

         B. If $w_j \neq u$: then $\varphi_i(w_i) = \varphi_i(u)$ but $\varphi_j(w_i) \neq \varphi_j(u)$. Application of Theorem 1.3 yields $\Pi(\varphi_i) \neq \Pi(\varphi_j)$.

      iii. $U_{w_i} = \{w_i\}$ and $\varphi^{-1}(k_i) = \{u\}$ and $w_i < u$.

A. If $w_j = u$. Then neither line 6 (since $|U_{w_j}| = 1$) nor line 8 are executed for $w_j$ and $k = \varphi(w_i)$ (since $w_j \not< w_i$). Thus, $k_j \neq \varphi(w_i)$ must hold. But then $\varphi_i(w_i) = k_i = \varphi(u) = \varphi(w_j) = \varphi_i(w_j)$ and $\varphi_j(w_i) = \varphi(w_i) \neq k_j = \varphi_j(w_j)$. Application of Theorem 1.3 yields $\Pi(\varphi_i) \neq \Pi(\varphi_j)$.

B. If $w_j \neq u$. Then $\varphi_i(w_i) = \varphi(u) = \varphi_i(u)$ but $\varphi_j(w_i) = \varphi(w_i) \neq \varphi(u) = \varphi_j(u)$. Again, application of Theorem 1.3 gives us $\Pi(\varphi_i) \neq \Pi(\varphi')$.

This finishes the proof, since in all cases $\Pi(\varphi_i) \neq \Pi(\varphi_j)$ holds. □

**Lemma 2.2** (Completeness). *Let $\varphi$ be an indexing of $V$ and $\varphi_1, \dots, \varphi_m$ be the sequence that is generated by algorithm 1 on input $V$ and $\varphi$. For all vertices $v$ and $k \in \{1, \dots, n\}$ such that $\Pi(\mathrm{move}(\varphi, v, k)) \neq \Pi(\varphi)$, there is $i \in \{1, \dots, m\}$ such that $\Pi(\mathrm{move}(\varphi, v, k)) = \Pi(\varphi_i)$.*

*Proof.* Let $\varphi$ be an indexing of $V$ and let $v$ be a vertex and $k \in \{1, \dots, n\}$. We want to show that if $\Pi(\mathrm{move}(\varphi, v, k)) \neq \Pi(\varphi)$, then there is an $i \in \{1, \dots, m\}$ such that $\Pi(\mathrm{move}(\varphi, v, k)) = \Pi(\varphi_i)$. First, note that $\varphi(v) \neq k$ holds, since otherwise this would imply $\Pi(\mathrm{move}(\varphi, v, k)) = \Pi(\varphi)$. The remainder of this proof works with multiple case distinctions, starting with $U_v = \varphi^{-1}(\varphi(v))$:

1. If $U_v = \{v\}$. This directly implies $\varphi^{-1}(k) \neq \emptyset$, since that would mean $\Pi(\mathrm{move}(\varphi, v, k)) = \Pi(\varphi)$. Thus, for $\varphi^{-1}(k)$ there are the following options:
   (a) If $\varphi^{-1}(k) = \{u\}$. In the case $v < u$, line 9 enumerates $\mathrm{move}(\varphi, v, k)$ directly. Otherwise, if $u < v$, line 9 enumerates $\mathrm{move}(\varphi, u, \varphi(v))$, where $\Pi(\mathrm{move}(\varphi, u, \varphi(v)) = \Pi(\mathrm{move}(\varphi, v, k))$.
   (b) If $\varphi^{-1}(k) = \{u, w, \dots\}$. Since $k \in \mathrm{image}(\varphi) \setminus \{\varphi(v)\}$ and $|\varphi^{-1}(k)| > 1$, line 7 in the algorithm can be applied. This then directly enumerates $\mathrm{move}(\varphi, v, k)$.
2. If $U_v = \{v, u, \dots\}$. Since $\varphi$ maps $v$ and $u$ to the same index, there is at least one index in $1, \dots, n$ that is assigned no vertex. But then $\mathcal{N} \neq \emptyset$. Again, for $\varphi^{-1}(k)$ there are the following options:
   (a) If $\varphi^{-1}(k) = \emptyset$.
      i. If $|U_v| > 2$, then line 13 is executed and there is some $\ell \in \mathcal{N}$ for which $\mathrm{move}(\varphi, v, \ell)$ is enumerated. But then $\Pi(\mathrm{move}(\varphi, v, k)) = \Pi(\mathrm{move}(\varphi, v, \ell))$.
      ii. If $U_v = \{v, u\}$.
         A. If $v < u$, then line 13 enumerates $\mathrm{move}(\varphi, v, \ell)$ for $v$ and some $\ell \in \mathcal{N}$. But then $\Pi(\mathrm{move}(\varphi, v, \ell)) = \Pi(\mathrm{move}(\varphi, v, k))$.
         B. If $u < v$, then line 13 enumerates $\mathrm{move}(\varphi, u, \ell)$ for $u$ and some $\ell \in \mathcal{N}$. But then again, $\Pi(\mathrm{move}(\varphi, u, \ell)) = \Pi(\mathrm{move}(\varphi, v, k))$.
   (b) If $\varphi^{-1}(k) = \{w, \dots\}$. Then $k \in \mathrm{image}(\varphi) \setminus \{\varphi(v)\}$ and since $|U_v| > 1$, line 7 is executed. This enumerates $\mathrm{move}(\varphi, v, k)$.

This shows that in all cases, there is some $\varphi_i$ that is enumerated which yields the same partition as $\mathrm{move}(\varphi, v, k)$. □

**Lemma 2.3.** *Let $\varphi$ be an indexing of $V$ and $\varphi_1, \dots, \varphi_m$ be the sequence that is generated by algorithm 1 on input $V$ and $\varphi$. Then $\Pi(\varphi) \neq \Pi(\varphi_i)$ for all $i \in \{1, \dots, m\}$.*

*Proof.* $\Pi(\text{move}(\varphi, v, k)) = \Pi(\varphi)$ if and only if either $k = \varphi(v)$ or if $\varphi^{-1}(\varphi(v)) = \{v\}$, then $k \in \mathcal{N}$. Simple case distinction yields that both cases never happen for any $\varphi_i$, $i \in \{1, \ldots, m\}$. $\square$