Silesian J. Pure Appl. Math.
vol. 10 (2020), 103–120

Vita OLSHEVSKA

Department of Mathematics, National University of Kyiv-Mohyla Academy,
Kyiv, Ukraine

ALGORITHMS FOR COMPUTATIONS WITH SYLOW 2-SUBGROUPS OF SYMETRIC GROUPS

Abstract. An algorithm to transform n-levels labeled binary rooted trees into elements from Sylow 2-subgroups of symmetric groups of degree 2^n is described. The inverse algorithm that on input permutation from a Sylow 2-subgroup of symmetric groups of degree 2^n finds a labeled tree is presented. An algorithm for multiplication of labeled trees that correspond to the multiplication of permutations from the Sylow 2-subgroup is introduced. The complexity and correctness of these algorithms are studied.

1. Introduction

Sylow p-subgroups of symmetric groups were described by Leo Kaluzhnin in terms of wreath products of cyclic groups in [4]. He proposed the representation of elements of these groups as tables, i.e. ordered sets of polynomials of a certain form (see e.g. [3] and [7] for details). Ju. Dmitruk described the algebraic structure of Sylow 2-subgroups of symmetric groups in [1]. The minimal generating sets and

2010 Mathematics Subject Classification: 20B30, 20B35, 05C25.
Keywords: Sylow 2-subgroup, permutation, binary rooted trees, algorithm.
Corresponding author: V. Olshevska (v.olshevska@ukma.edu.ua).
Received: 29.11.2020.
This work was partially supported by Yuchymenko Family Doctoral School Endowement of the International Charity Fund of Kyiv-Mohyla Academy for development of Yuchymenko Family Doctoral School at the National University of Kyiv-Mohyla Academy.

Cayley graphs of a Sylow p-subgroup (p — prime) of the symmetric group S_{p^n} were characterized by A. Slupik and V. Sushchansky in [8].

Fix a positive integer $n > 1$. Denote by $Syl_2(S_{2^n})$ a Sylow 2-subgroup of the symmetric group S_{2^n}. The group $Syl_2(S_{2^n})$ is isomorphic to the n times iterated wreath product of cyclic groups or order 2, i.e.

$$Syl_2(S_{2^n}) \cong \underbrace{\mathbb{Z}_2 \wr \ldots \wr \mathbb{Z}_2}_{n \text{ times}}.$$

B. Pawlik in [5] using the polynomial representation of elements described the action of $Syl_2(S_{2^n})$ on a set of of minimal generated sets of this group.

Another useful representation of wreath products of permutation groups is in terms of automorphisms of rooted trees. In particular, one can view elements of Sylow p-subgroups of the symmetric group S_{p^n} as so called portraits of such automorphisms, i.e. labeled regular rooted trees (see [2] for details). Since tree-like data structures are convenient and efficient for computations this leads to the natural direction of developing algorithms for computations with Sylow p-subgroup of symmetric groups using this representation. The basic case $p = 2$ deserves special attention due to the binary nature of the data involved.

The goal of this paper is to describe and analyze algorithms for computations with Sylow 2-subgroups of symmetric groups. We present algorithms that transform elements of $Syl_2(S_{2^n})$ into labeled binary rooted trees and vise versa. We discuss correctness and complexity of these algorithms. An algorithm that describes the multiplication of two labeled binary trees is presented. It is shown that using this algorithm the product of corresponding permutations from $Syl_2(S_{2^n})$ can be calculated.

2. Preliminaries

A tree T is called *rooted tree* if we fix one vertex v_0 that is called *the root.* A rooted tree is called *binary tree* if the degree of the root v_0 is equal 2 and the degrees of others vertices (except leaves) are equal 3. *The distance between the vertices* v_i and v_j is equal to the length of the shortest path between them. If the distance between the root v_0 and the vertex v is j, then a vertex v is called a *vertex of the jth level.* Denote by T_n a binary rooted tree with n levels. Denote by V the set of vertices of tree [2, 6].

A binary rooted tree, which has label 0 or 1 on all vertices from level 0 to level $(n-1)$, is called *labeled tree*. Denote the set of all such n-level trees by $LT_{2,n}$. Note that its cardinality is $|LT_{2,n}| = 2^{2^n-1}$.

Let D is a tree from the set $LT_{2,n}$. We numerate all vertices of all levels. And let i is a number of vertex v on level j. In this case we say that a pair (j,i) is *coordinates of the vertex v of a tree D*, $i \in \{1,\dots,2^j\}$, $j \in \{0,\dots,(n-1)\}$.

Denote $Coord(D)$ is *a set of coordinates* of all vertices of tree D.

Define mapping $c : V(T_n) \to Coord(T_n)$ by the rule

$$c(v) = (j,i), \text{ if } (j,i) \text{ is a pair of coordinates of vertex } v \text{ on the tree } T_n.$$

Assume that $(j,i) < (k,r)$ if $j < k$ or $j = k$ with $i < r$.

We also say that $v < w$ if $c(v) < c(w)$.

Denote the next sets

1. $OC(D) = \{(j,i) \in Coord(D)|$ exists vertex with pair of coordinates (j,i) labeled by 1 in the tree D $\}$.

2. $OV(D) = \{v \in V(T_n)|$ pair of coordinates (j,i) of vertex v belongs to $OC(D)\}$.

We consider next *operations on a set of all vertices* of the tree D:

1. *switch*(D,v) ="to switch two sub-trees of the tree D, for which vertex v is a root";

2. *switch*(D,j,i) ="to switch two sub-trees of the tree D, for which some vertex v with a pair of coordinates (j,i) is a root".

The second row $a = (a_1, a_2, \dots, a_{2^n})$ of permutation $\pi = \begin{pmatrix} 1 & 2 & \dots & 2^n \\ a_1 & a_2 & \dots & a_{2^n} \end{pmatrix}$ is called *a block of elements*. Permutation π is called 2-*separated* if we can do the next steps.

1. At first, we divide the block a into 2 sub-blocks with the same length: $u_1 = (a_1, \dots, a_{2^{n-1}})$ and $u_2 = (a_{2^{n-1}+1}, \dots, a_{2^n})$. Then we check if every element of u_1 is greater (or less) than every element of u_2.

2. If step 1 holds, then we repeat process and divide blocks u_1 and u_2 into sub-blocks $u_{1,1}$, $u_{1,2}$ and $u_{2,1}$ $u_{2,2}$. After that we check the value of elements between corresponding blocks. And so on until we get sub-blocks that contain only one element.

From the definition of the wreath product [9] follows that any 2-separated permutation is an element of wreath product $\underbrace{S_2 \wr \ldots \wr S_2}_{n \text{ times}}$. Thus, these permutations are elements of the Sylow 2-subgroup of the group S_{2^n} according to [7].

3. Transformation a tree from $LT_{2,n}$ to permutation from $Syl_2 S_{2^n}$

Consider an algorithm (see Algorithm 1) of obtaining a permutation from $Syl_2 S_{2^n}$ based on a tree from $LT_{2,n}$. Input is a set of coordinates of vertices labeled by 1 of some tree D. Output is a final vector $a = (\pi(1), \pi(2), \ldots, \pi(2^n))$.

Algorithm 1: Algorithm of transformation a tree into a permutation

Input: $OC(D)$ be a set of coordinates of all vertices labeled by 1 of a tree D.

Output: $(a_{i_1}, a_{i_2}, \cdots a_{i_{2^n}})$ is the second row of permutation.

```
(a_1, a_2, ··· a_{2^n}) = (1, 2, ··· , 2^n) ;
for (j, i) ∈ (OC(D), <) do
    m := 2^{n−j−1} ( is count of elements in one block) ;
    for l := 1 to m do
        b := a_{(2i−2)m+l};
        a_{(2i−2)m+l} := a_{(2i−1)m+l};
        a_{(2i−1)m+l} := b;
```

Example 1. Consider a tree $D \in LT_{2,4}$ where the 2nd vertex of the 1st level labeled by 1. Obtain permutation, which a tree D sets (see Fig. 1).

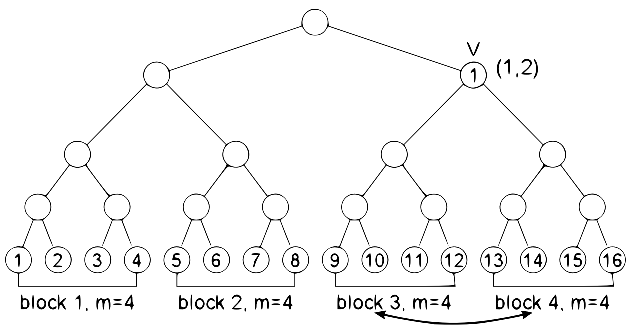

Fig. 1. 4-levels labeled tree and elements of the vector a

From the Fig. 1 we have: $(a_1\, a_2\, a_3\, \dots\, a_{16}) = (1, 2, 3, \dots, 16)$, $j = 1$, $i = 2$, $n = 4$. Then we obtain

- $m = 2^{n-j-1} = 4$ is a length of a block;
- switch $2i - 1 = 3$rd and $2i = 4$th blocks:
  $a_{(2i-2)m+1} = a_9$ switch with $a_{(2i-1)m+1} = a_{13}$,
  $a_{(2i-2)m+2} = a_{10}$ switch with $a_{(2i-1)m+2} = a_{14}$,
  $a_{(2i-2)m+3} = a_{11}$ switch with $a_{(2i-1)m+3} = a_{15}$,
  $a_{(2i-2)m+4} = a_{12}$ switch with $a_{(2i-1)m+4} = a_{16}$.

As result, we get the permutation

$$\begin{pmatrix} 1 & \dots & 8 & 9 & \dots & 12 & 13 & 14 & 15 & 16 \\ 1 & \dots & 8 & 13 & \dots & 16 & 9 & 10 & 11 & 12 \end{pmatrix}.$$

Theorem 2. *The algorithm of transformation tree into a permutation is correct.*

Proof. Note that initial row $(1, 2, \dots, 2^n)$ is 2-separated. Also note, that for every pair of coordinates $(j, i) \in Coord(D)$ blocks $u_1 := (a_{(2i-2)m+1}, \dots, a_{(2i-2)m+m})$ and $u_2 := (a_{(2i-1)m+1}, \dots, a_{(2i-1)m+m})$ also are 2-separated. And after switch among themselves, block $u = (a_{(2i-2)m+1}, \dots, a_{(2i-2)m+m}, a_{(2i-1)m+1}, \dots, a_{(2i-1)m+m})$ is still 2-separated. So, every transformation on pair of coordinates $(j, i) \in OC(D)$ does not change 2-separated property of row a. As result, we obtain that the row a is defined by 2-separated permutation π. $\square$

Theorem 3. *The complexity of transformation algorithm of a tree from $LT_{2,n}$ into a permutation from $Syl_2(S_{2^n})$ is equal to $O(n \cdot 2^n)$.*

Proof. Assume that every vertex v of the tree D is labeled by 1, $c(v) \in OC(D)$. In this case $OC(D)$ has maximum cardinality. Then we need to do 2^{n-j-1} switches of corresponding elements of vector a for every vertex v with coordinates (j, i).

Based on the fact that level j contains 2^j vertices, $0 \le j \le (n - 1)$, we get

$$\sum_{j=0}^{n-1} 2^j \cdot 2^{n-j-1} = \sum_{j=0}^{n-1} 2^{n-1} = n \cdot 2^{n-1}.$$

So, we have $O(n \cdot 2^{n-1}) = O(n \cdot 2^n)$. $\square$

4. Transformation a permutation π from $Syl_2(S_{2^n})$ into a tree from $LT_{2,n}$

Consider an algorithm (see Algorithm 2) that finds a corresponding tree $D \in LT_{2,m}$ for a permutation $\pi \in Syl_2(S_{2^n})$. Input is $a = (\pi(1), \pi(2), \ldots, \pi(2^n))$. Output is a set of coordinates whose vertices are labeled by 1.

Algorithm 2: Transformation algorithm of a permutation into a tree

```
Input: (a_1, a_2, ..., a_{2^n}) is the second row of 2-separated permutation π,
       π ∈ Syl_2(S_{2^n}).
Output: OC(D).
OC(D) := ∅;
for j := 0 to n − 1 do
    m := 2^{n−j−1} (length of block);
    for i := 1 to 2^j do
        if a_{(2i−2)m+1} > a_{(2i−1)m+1} then
            OC(D) := OC(D) ⋃ {(j, i)}
```

Example 4. Let $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 4 & 2 & 1 & 6 & 5 & 7 & 8 \end{pmatrix} \in S_8$. Consider the corresponding vector

$$(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8) = (3, 4, 2, 1, 6, 5, 7, 8).$$

Define the set $OC(D) = \emptyset$. According to the Algorithm 2 we have (see Table 1):

Table 1

j	m	Blocks	i	$a_{(2i-2)m+1}$	$a_{(2i-1)m+1}$	Comparing	$OC(D)$
0	4	3 4 2 1 | 6 5 7 8	1	$a_1 = 3$	$a_5 = 6$	$3 \not> 6$	$\emptyset$
1	2	3 4 | 2 1 | 6 5 | 7 8	1	$a_1 = 3$	$a_3 = 2$	$3 > 2$	$\{(1,1)\}$
			2	$a_5 = 6$	$a_7 = 7$	$5 \not> 7$	$\{(1,1)\}$
2	1	3| 4| 2| 1| 6| 5| 7| 8	1	$a_1 = 3$	$a_2 = 4$	$3 \not> 4$	$\{(1,1)\}$
			2	$a_3 = 2$	$a_4 = 1$	$2 > 1$	$\{(1,1),(2,2)\}$
			3	$a_5 = 6$	$a_6 = 5$	$6 > 5$	$\{(1,1),(2,2),(2,3)\}$
			4	$a_7 = 7$	$a_8 = 8$	$7 \not> 8$	$\{(1,1),(2,2),(2,3)\}$

As result, we obtain: $OC(D) = \{(1,1), (2,2), (2,3)\}$. So, we have the next tree (see Fig. 2):

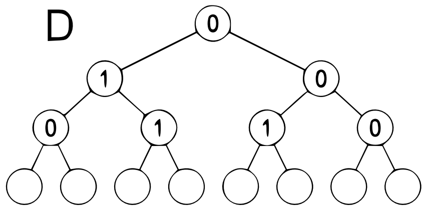

Fig. 2. The tree D as result of Algorithm 2

Theorem 5. *The complexity of the transformation algorithm of a permutation from $Syl_2(S_{2^n})$ into a tree from $LT_{2,n}$ is equal to $O(2^n)$.*

Proof. There are two loops in algorithm: first is over j (required for defining level of tree) and second is over i (required for defining index of vertex on level). All vertices of tree are considered in this loops. Inside the loops we have one operation. So, the complexity of algorithm is $O(2^n - 1) = O(2^n)$. $\square$

Define mapping $\psi : LT_{2,n} \to Syl_2(S_{2^n})$ that is determined by the Algorithm 1. Similar, define mapping $\tau : Syl_2(S_{2^n}) \to LT_{2,n}$, that is determined by the Algorithm 2.

Theorem 6. *Mappings ψ and τ are inverse to each other and are bijections between $LT_{2,n}$ and $Syl_2(S_{2^n})$.*

Proof. We need to show that $\tau(\psi) = id$.

Let π be some 2-separated permutation and $D = \tau(\pi)$ be a tree from $LT_{2,n}$, obtained from π by Algorithm 2. Then

$$(j, i) \in OC(\tau(\pi)) \text{ if and only if } a_{(2i-2)m+1} > a_{(2i-1)m+1} \text{ in permutation } \pi. \quad (1)$$

Let D be some tree from $LT_{2,n}$ and $\pi = \psi(D)$ be 2-separated permutation, obtained by Algorithm 1. As Algorithm 1 started from ordered row $(1, 2, \ldots, 2^n)$,

$$(j, i) \in OC(D) \text{ if and only if } a_{(2i-2)m+1} > a_{(2i-1)m+1} \text{ in permutation } \psi(D). \quad (2)$$

Then from (2) we have

$$(j, i) \in OC(D) \text{ if and only if in } \psi(D): \ a_{(2i-2)m+1} > a_{(2i-1)m+1}.$$

So, (1) implies that $(j, i) \in OC(\tau(\psi(D)))$. Therefore, $\tau(\psi) = id$.

The proof of $\psi(\tau) = id$ is similar. $\square$

5. Multiplication of trees from $LT_{2,n}$

Consider the multiplication algorithm (see Algorithm 3) for two trees $D_1, D_2 \in LT_{2,n}$. Let a vertex v labeled by 1 and $c(v) = (j,i) \in OC(D_1)$. Let w be a vertex of the tree D_2 and $c(w) = (j,i) \in OC(D_2)$. In steps 3-4 we run the operation $switch(D_2, j, i)$ and change a label of vertex w to the opposite.

Algorithm 3: Algorithm of tree multiplication

Input: Two trees: D_1 first multiplier, D_2 second multiplier.

Output: A tree D with defined set $OC(D)$.

1 $OC(D) := OC(D_2)$;

2 **for** $(j,i) \in (OC(D_1), <)$ **do**

3 $\quad switch(D, j, i)$;

4 $\quad OC(D) := OC(D) \triangle \{(j,i)\}$;

By $\triangle$ we denote the symmetric difference of sets.

Note that in Algorithm 3 we can replace the ordered set $(OC(D_1), <)$ by the ordered set $(OV(D_1), <)$.

Example 7. Consider trees D_1, D_2 (see Fig. 3). We have

$$OC(D_1) = \{(0,1), (1,1), (2,2), (2,4)\},$$
$$OC(D_2) = \{(1,1), (2,1), (2,3)\}.$$

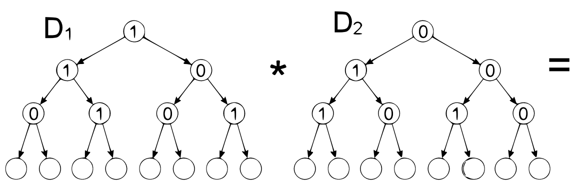

Fig. 3. Input for Algorithm 3

The next tree (see Fig. 4) is the product of the trees D_1, D_2:

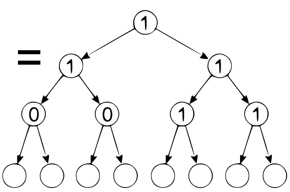

Fig. 4. Output for Algorithm 3

Theorem 8. *The complexity of the multiplication algorithm of the trees from $LT_{2,n}$ is equal to $O(2^n)$.*

Proof. There is a loop of length $|OC(D_1)|$ in the Algorithm 3. The maximum cardinality of the set will be reached if the tree D_1 will have the label 1 on all vertices from 0 to $(n-1)$th level. That means

$$|OC(D_1)| = |Coord(D_1)| = 2^n - 1.$$

Inside the loops we have 2 operations on every pair of coordinates (j, i): $switch(D, j, i)$ and symmetric difference $OC(D) \triangle (j, i)$. Therefore there will be done $2 \cdot (2^n - 1)$ operations.

So, the complexity of the algorithm is $O(2 \cdot (2^n - 1)) = O(2^n)$. □

6. Multiplication of permutations in terms of multiplication of binary labeled rooted trees

6.1. Vertex mapping

Let $v_0, v, w \in V(T_n)$. Denote by $Path_{v_0}(v) := \{v_0, \ldots, v_{j-1}\}$ a path, that is connected the root v_0 with some vertex v of jth level. We say that *a vertex v is under a vertex w (a vertex w is above a vertex v)* if $w \in Path_e(v)$. In this case we will write that $v \succ w$. Remark that if $v \succ w$ then $v > w$.

Definition 9. *Let w be a vertex from $V(T_n)$. Define a mapping $\mathbb{ACT}_w : V(T_n) \to V(T_n)$ for a vertex $v \in V(T_n)$ by the next rule: $\mathbb{ACT}_w(v) = v'$ if and only if v' is an image v after $swithc(T_n, w)$.*

Note that

- if $v \succ w$ and $c(w) = (k, r)$, $c(v) = (j, i)$, then $c(v') = (j, i')$ and

$$i' = \begin{cases} i + 2^{j-k-1}, \text{ if } i \leq (r-1) \cdot 2^{j-k} + 2^{j-k-1} \\ \text{ which means that } v \text{ is in the left branch of a sub-tree with a root } w, \\ i - 2^{j-k-1}, \text{ if } i \geq (r-1) \cdot 2^{j-k} + 2^{j-k-1} + 1 \\ \text{which means that } v\text{is in the right branch of a sub-tree with a root } w, \end{cases}$$

- if $v \not\succ w$ then $\mathbb{ACT}_w(v) = v$.

Remark 10. For every vertex $w \in V(T_n)$ we have

$$\mathbb{ACT}_w^2 = id.$$

Let $A = \{w_1, \ldots, w_t\}$ be an ordered set, $B = \{v_1, \ldots, v_l\}$, $A, B \subset V(T_n)$ and v be some vertex from $V(T_n)$. Denote by

- $\mathbb{ACT}_v(B) := \{\mathbb{ACT}_v(v_1), \ldots, \mathbb{ACT}_v(v_t)\}$.
- $\mathbb{ACT}_A(v) := (\mathbb{ACT}_{w_1} \cdot \ldots \cdot \mathbb{ACT}_{w_t})(v) = \mathbb{ACT}_{w_t}(\ldots(\mathbb{ACT}_{w_1}(v))\ldots)$.

  We will consider that for empty set $A = \emptyset$: $\mathbb{ACT}_A = id$.
- $\mathbb{ACT}_A(B) := \{\mathbb{ACT}_A(v_1), \ldots, \mathbb{ACT}_A(v_l)\}$

Let $D \in LT_{2,n}$, $w \in V(D)$. Note that the set of vertices of the tree $switch(D, w)$ is $\mathbb{ACT}_w(OV(D))$.

Remark 11. Let $D \in LT_{2,n}$ be some tree and $\pi = \psi(D)$ be some permutation, obtained by Algorithm 1. Then for any $k, 1 \leq k \leq 2^n$ and $v \in V(T_n)$, $c(v) = (n, s)$:

$$\pi(k) = s \text{ if and only if } c(\mathbb{ACT}_{(OV(D),<)}(v)) = (n, k).$$

Theorem 12. *For any trees $D_1, D_2 \in LT_{2,n}$:*

$$OV(D_1 * D_2) = (\,\mathbb{ACT}_{(OV(D_1),<)}(OV(D_2))\,) \bigtriangleup OV(D_1).$$

Proof. Consider steps $2-4$ from multiplication Algorithm 3. This loop can be replaced by two in such way

$$\begin{aligned}
&\textbf{For } v \in (OV(D_1), <):\\
&\quad switch(D, v);\\
&\textbf{For } v \in (OV(D_1), <):\\
&\quad OV(D) := OV(D) \triangle \{v\};
\end{aligned}$$

Then the first loop can be rewritten as follows $OV(D) := \mathbb{ACT}_{(OV(D_1),<)}(OV(D))$. And the second loop can be rewritten as follows $OV(D) := OV(D) \triangle OV(D_1)$.

From notation $D = D_1 * D_2$ and step 1 we have

$$OV(D_1 * D_2) = \mathbb{ACT}_{(OV(D_1),<)}(OV(D_2)) \bigtriangleup OV(D_1).$$

$\square$

6.2. Vertex mapping properties

Lemma 13. *Let $v_1, v_2 \in V(T_n)$ and $v_1 \not\succ v_2$ and $v_2 \not\succ v_1$. Then*

$$\mathbb{ACT}_{v_1} \cdot \mathbb{ACT}_{v_2} = \mathbb{ACT}_{v_2} \cdot \mathbb{ACT}_{v_1}.$$

Proof. Let $t \in V(T_n)$. Consider the next cases:

Case 1. Let $t \succ v_1$. Then for $\mathbb{ACT}_{v_1} t := t'$ we obtain

$$t' \succ v_1, \quad t \not\succ v_2 \quad \text{and} \quad t' \not\succ v_2.$$

Then

$$\mathbb{ACT}_{v_2}(\mathbb{ACT}_{v_1} t) = \mathbb{ACT}_{v_2} t' = t',$$
$$\mathbb{ACT}_{v_1}(\mathbb{ACT}_{v_2} t) = \mathbb{ACT}_{v_1} t = t'.$$

Case 2. Let $t \succ v_2$. Then the proof is similar to case 1.

Case 3. Let $t \not\succ v_1$ and $t \not\succ v_2$. Then we obtain

$$\mathbb{ACT}_{v_2}(\mathbb{ACT}_{v_1} t) = \mathbb{ACT}_{v_2} t = t,$$
$$\mathbb{ACT}_{v_1}(\mathbb{ACT}_{v_2} t) = \mathbb{ACT}_{v_1} t = t.$$

Therefore $\mathbb{ACT}_{v_1} \cdot \mathbb{ACT}_{v_2} = \mathbb{ACT}_{v_2} \cdot \mathbb{ACT}_{v_1}$. □

Lemma 14. *Let $v, v_1 \in V(T_n)$ and $v_1 < v$. Then*

$$\mathbb{ACT}_{\mathbb{ACT}_{v_1}(v)} = \mathbb{ACT}_{v_1} \cdot \mathbb{ACT}_{v} \cdot \mathbb{ACT}_{v_1}. \tag{3}$$

Proof. Case 1. Let $v \not\succ v_1$. Lemma 13 implies that we have

$$\mathbb{ACT}_{v} \cdot \mathbb{ACT}_{v_1} = \mathbb{ACT}_{v_1} \cdot \mathbb{ACT}_{v}. \tag{4}$$

Note that $v \not\succ v_1$, so $\mathbb{ACT}_{v_1}(v) = v$. So, we have

$$\mathbb{ACT}_{\mathbb{ACT}_{v_1}(v)} = \mathbb{ACT}_{v}. \tag{5}$$

From (4) and (5) we have

$$\mathbb{ACT}_{v_1} \cdot \mathbb{ACT}_{v} \cdot \mathbb{ACT}_{v_1} = \mathbb{ACT}_{v_1} \cdot \mathbb{ACT}_{v_1} \cdot \mathbb{ACT}_{v} = id \cdot \mathbb{ACT}_{v} = \mathbb{ACT}_{v} = \mathbb{ACT}_{\mathbb{ACT}_{v_1}(v)}.$$

Case 2. Let $v \succ v_1$. Denote by $v' := \mathbb{ACT}_{v_1}(v)$. Let t be some fixed vertex $V(T_n)$. We consider the next cases.

- Let $t \not\succ v_1$. As result $t \not\succ v$ and $t \not\succ v'$. That's why

$$(\mathbb{ACT}_{v_1} \cdot \mathbb{ACT}_v \cdot \mathbb{ACT}_{v_1})(t) = t = \mathbb{ACT}_{v'}(t).$$

- Let $t \succ v_1$ and $t \not\succ v'$. Then, on the one hand $\mathbb{ACT}_{v'}(t) = t$. Note that $t \not\succ v' = \mathbb{ACT}_{v_1}(v)$. Then, by acting over vertex v_1, we have

$$\mathbb{ACT}_{v_1}(t) \not\succ \mathbb{ACT}_{v_1}(\mathbb{ACT}_{v_1}(v)) = v,$$

so $\mathbb{ACT}_v(\mathbb{ACT}_{v_1}(v)) = \mathbb{ACT}_{v_1}(v)$. So, on the other hand

$$(\mathbb{ACT}_{v_1} \cdot \mathbb{ACT}_v \cdot \mathbb{ACT}_{v_1})(t) = \mathbb{ACT}_{v_1}(\mathbb{ACT}_v(\mathbb{ACT}_{v_1}(t))) = \\ = \mathbb{ACT}_{v_1}(\mathbb{ACT}_{v_1}(t)) = t.$$

- Let $t \succ v_1$ and $t \succ v'$. Let vertices v, v_1 and t have the next coordinates: $c(v) = (k, r), c(v_1) = (k_1, r_1)$ and $c(t) = (k_2, r_2)$, where $k_1 < k < k_2$, $1 \leq r_1 \leq 2^{k_1}$ (see Fig. 5).

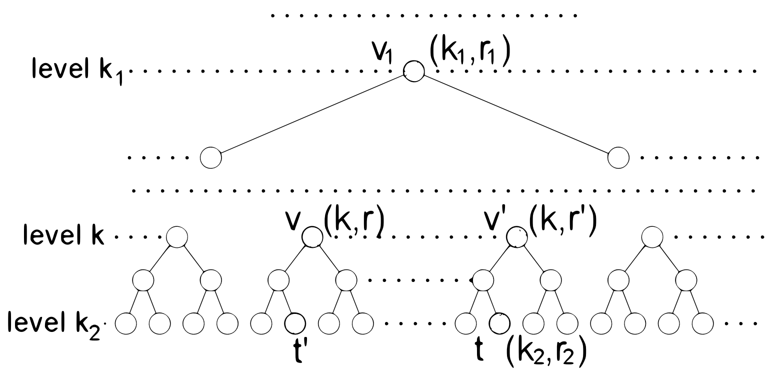

Fig. 5. The location of vertices of the tree T_n

Without loss of generality we can say that

(a) v lies on the left to v'. Then

$$r' = r + 2^{k-k_1-1}, \tag{6}$$

(b) t is in the left branch of a sub-tree with the root v' of the tree T_n. Then

$$(r'-1)\cdot 2^{k_2-k} + 1 \le r_2 \le (r'-1)\cdot 2^{k_2-k} + 2^{k_2-k-1}. \tag{7}$$

From (7) and (b) follows that the left part of equality (3) is equal to

$$c(\mathbb{ACT}_{v'}(t)) = (k_2, r_2 + 2^{k_2-k-1}). \tag{8}$$

From (a) and (b) we have that t is in the right branch of a sub-tree with the root v_1 of the tree T_n. So, the image of t will be in the left branch of the sub-tree with a root v_1. Hence $\mathbb{ACT}_{v_1}(t) = t'$, where $c(t') = (k_2, r_2 - 2^{k_2-k_1-1})$. So, we have

$$(\mathbb{ACT}_{v_1}\cdot\mathbb{ACT}_{v}\cdot\mathbb{ACT}_{v_1})(t) = (\mathbb{ACT}_{v}\cdot\mathbb{ACT}_{v_1})(t'). \tag{9}$$

From (7) and (6) we have:

$$\begin{aligned} r_2 \le (r'-1)\cdot 2^{k_2-k} + 2^{k_2-k-1} = (r-1+2^{k-k_1-1})\cdot 2^{k_2-k} + 2^{k_2-k-1} = \\ = (r-1)2^{k_2-k} + 2^{k_2-k-1} + 2^{\cancel{k}-k_1-1+k_2-\cancel{k}}. \end{aligned}$$

Hence

$$\underbrace{r_2 - 2^{k_2-k_1-1}}_{\substack{\text{second coordinate}\\ \text{of the vertex t}'}} \le (r-1)2^{k_2-k} + 2^{k_2-k-1},$$

which means that t' is in the left branch of a sub-tree with a root v of the tree T_n. Hence $\mathbb{ACT}_v(t') = t''$, where $c(t'') = (k_2, r_2 - 2^{k_2-k_1-1} + 2^{k_2-k-1})$. So, we have

$$(\mathbb{ACT}_{v}\cdot\mathbb{ACT}_{v_1})(t') = \mathbb{ACT}_{v_1}(t''). \tag{10}$$

As $t' \succ v$, $t'' \succ v$. Also, this vertex v is in the left branch of a tree with a root v_1. As result, t'' is also in the left branch of a tree with a root v_1.

That's why the coordinates of image of vertex t'' due to action over v_1 are

$$c(\mathbb{ACT}_{v_1}(t'')) = (k_2, r_2 - \cancel{2^{k_2-k_1-1}} + 2^{k_2-k-1} + \cancel{2^{k_2-k_1-1}}) = \\ = (k_2, r_2 + 2^{k_2-k-1}). \quad (11)$$

From (8), (9), (10) and (11) we obtain

$$\mathbb{ACT}_{\mathbb{ACT}_{v_1}(v)}(t) = (\mathbb{ACT}_{v_1} \cdot \mathbb{ACT}_v \cdot \mathbb{ACT}_{v_1})(t).$$

$\square$

Lemma 15. *Let $A \subset V(T_n)$ be some ordered set and $B, C \subset V(T_n)$. Then*

$$\mathbb{ACT}_A(B \triangle C) = \mathbb{ACT}_A(B) \triangle \mathbb{ACT}_A(C).$$

Proof.

$$\mathbb{ACT}_A(B \triangle C) = \{\mathbb{ACT}_A(v) | v \in B \triangle C\} = \\ = \{\mathbb{ACT}_A(v) | v \in B\} \triangle \{\mathbb{ACT}_A(v) | v \in C\} = \mathbb{ACT}_A(B) \triangle \mathbb{ACT}_A(C).$$

$\square$

Lemma 16. *For every $a, b \in V(T_n)$:*

$$\mathbb{ACT}_a \cdot \mathbb{ACT}_b = \mathbb{ACT}_{(\{\mathbb{ACT}_b(a)\} \triangle \{b\}, <)}. \quad (12)$$

Proof. Case 1. If $a < b$ then $\mathbb{ACT}_b(a) = a$ and $(\{a\} \triangle \{b\}, <) = \{a, b\}$. Hence

$$\mathbb{ACT}_{(\{\mathbb{ACT}_b(a)\} \triangle \{b\}, <)} = \mathbb{ACT}_{\{a,b\}} = \mathbb{ACT}_a \cdot \mathbb{ACT}_b.$$

Case 2. If $a = b$ then $\mathbb{ACT}_b(a) = \mathbb{ACT}_a(a) = a$ and $(\{a\} \triangle \{b\}, <) = \emptyset$. Hence

$$\mathbb{ACT}_{(\{\mathbb{ACT}_b(a)\} \triangle \{b\}, <)} = \mathbb{ACT}_\emptyset = id = \mathbb{ACT}_a \cdot \mathbb{ACT}_a.$$

Case 3. If $b < a$ then $b < \mathbb{ACT}_b(a)$. That's why $(\{\mathbb{ACT}_b(a)\} \triangle \{b\}, <) = \{b, \mathbb{ACT}_b(a)\}$. Lemma 14 implies that $\mathbb{ACT}_{\mathbb{ACT}_b(a)} = \mathbb{ACT}_b \cdot \mathbb{ACT}_a \cdot \mathbb{ACT}_b$. Hence

$$\mathbb{ACT}_{(\{\mathbb{ACT}_b(a)\} \triangle \{b\}, <)} = \mathbb{ACT}_{\{b, \mathbb{ACT}_b(a)\}} = \mathbb{ACT}_b \cdot \mathbb{ACT}_{\mathbb{ACT}_b(a)} = \mathbb{ACT}_a \cdot \mathbb{ACT}_b.$$

$\square$

Lemma 17. *Let $A \subset V(T_n)$ be an ordered set by $<$ and $b \in V(T_n)$. Then*

$$\mathbb{ACT}_A \cdot \mathbb{ACT}_b = \mathbb{ACT}_{(\mathbb{ACT}_b(A) \triangle \{b\},<)}.$$

Proof. Let $A = \{a_1, \ldots, a_m\}$ and $1 \leq k \leq m$:

$$a_1 < \ldots < a_k \leq b < a_{k+1} < \ldots a_m. \tag{13}$$

Based on Lemma 16 and equation (13) we have

$$\begin{aligned}
\mathbb{ACT}_A \cdot \mathbb{ACT}_b = \mathbb{ACT}_{a_1} \cdot \ldots \cdot \mathbb{ACT}_{a_m} \cdot \mathbb{ACT}_b = \\
= \mathbb{ACT}_{a_1} \cdot \ldots \cdot \mathbb{ACT}_{(\{\mathbb{ACT}_b(a_m)\} \triangle \{b\},<)} = \\
= \mathbb{ACT}_{a_1} \cdot \ldots \cdot \mathbb{ACT}_{\{b, \mathbb{ACT}_b(a_m)\}} = \mathbb{ACT}_{a_1} \cdot \ldots \cdot \mathbb{ACT}_b \cdot \mathbb{ACT}_{\mathbb{ACT}_b(a_m)} = \ldots \\
\ldots = \mathbb{ACT}_{a_1} \cdot \ldots \cdot \mathbb{ACT}_{a_k} \cdot \mathbb{ACT}_b \cdot \mathbb{ACT}_{\mathbb{ACT}_b(\{a_{k+1}, \ldots, a_m\})}.
\end{aligned}$$

Note that for every $a \in \{a_1, \ldots, a_k\}$: $\mathbb{ACT}_b(a) = a$. Hence the last is equal to

$$\mathbb{ACT}_{\mathbb{ACT}_b(\{a_1, \ldots, a_k\})} \cdot \mathbb{ACT}_b \cdot \mathbb{ACT}_{\mathbb{ACT}_b(\{a_{k+1}, \ldots, a_m\})}. \tag{14}$$

1. If $a_k \neq b$, then (14) is equal to $\mathbb{ACT}_{(\mathbb{ACT}_b(A) \bigcup \{b\},<)}$.

2. If $a_k = b$, then (14) is equal to $\mathbb{ACT}_{(\mathbb{ACT}_b(A) \setminus \{b\},<)}$.

Hence, in general case (14) is equal to $\mathbb{ACT}_{(\mathbb{ACT}_b(A) \triangle \{b\},<)}$. □

Lemma 18. *Let $A, B \subset V(T_n)$ be some ordered sets by $<$. Then*

$$\mathbb{ACT}_A \cdot \mathbb{ACT}_B = \mathbb{ACT}_{(\mathbb{ACT}_B(A) \triangle B,<)}.$$

Proof. Let $B = \{b_1, \ldots, b_l\}$. By Lemma 17 for A and vertices b_1, b_2 we have

$$\begin{aligned}
\mathbb{ACT}_A \cdot \mathbb{ACT}_{b_1} \cdot \mathbb{ACT}_{b_2} = \mathbb{ACT}_{(\mathbb{ACT}_{b_1}(A) \triangle \{b_1\},<)} \cdot \mathbb{ACT}_{b_2} = \\
= \mathbb{ACT}_{(\mathbb{ACT}_{b_2}(\mathbb{ACT}_{b_1}(A) \triangle \{b_1\},<) \triangle \{b_2\},<)} = \\
= \mathbb{ACT}_{(\mathbb{ACT}_{b_2}(\mathbb{ACT}_{b_1}(A) \triangle \{b_1\}) \triangle \{b_2\},<)}.
\end{aligned} \tag{15}$$

Since the set B is ordered, then $b_1 < b_2$. Therefore $\mathbb{ACT}_{b_2}(b_1) = b_1$. From this equality, Lemma 15 and (15) we have

$$\mathbb{ACT}_{(\mathbb{ACT}_{b_2}(\mathbb{ACT}_{b_1}(A) \triangle \{b_1\}) \triangle \{b_2\},<)} = \mathbb{ACT}_{(\mathbb{ACT}_{\{b_1,b_2\}}(A) \triangle \{b_1,b_2\},<)}. \tag{16}$$

From (15) and (16) for every $b_3, \ldots, b_l$:

$$\mathbb{ACT}_A \cdot \mathbb{ACT}_B = \mathbb{ACT}_{(\mathbb{ACT}_{\{b_1,b_2\}}(A)\triangle\{b_1,b_2\},<)} \cdot \prod_{k=3}^{l} \mathbb{ACT}_{b_k} = \ldots$$
$$= \mathbb{ACT}_{\mathbb{ACT}_B(A)\triangle B}.$$

□

Corollary 19. *Let $D_1, D_2 \in LT_{2,n}$. Then*

$$\mathbb{ACT}_{(OV(D_1 * D_2),<)} = \mathbb{ACT}_{(OV(D_2),<)} \cdot \mathbb{ACT}_{(OV(D_1),<)}.$$

Proof. Proof directly follows from Theorem 12 and Lemma 18. □

6.3. Isomorphism theorem

Theorem 20. *The mapping ψ is an isomorphism between $LT_{2,n}$ and $Syl_2(S_{2,n})$.*

Proof. First, note that by Theorem 6 ψ is bijection. We need to show that we have the next for any $D_1, D_2 \in LT_{2,n}$:

$$\psi(D_1 * D_2) = \psi(D_1) \cdot \psi(D_2).$$

So, for permutations $\pi_1 := \psi(D_1)$, $\pi_2 := \psi(D_2)$, $\pi := \psi(D_1 * D_2)$ and every number $1 \leq i \leq 2^n$ we need to show the next

$$\pi(i) = (\pi_1\pi_2)(i).$$

The last equality is equivalent to

$$\pi^{-1}(i) = (\pi_1\pi_2)^{-1}(i). \tag{17}$$

Let w be a vertex with coordinates (n, s) and $1 \leq k \leq 2^n$. Then Remark 11 implies that

$$\pi_1(k) = s \text{ if and only if } c(\mathbb{ACT}_{(OV(D_1),<)}(w)) = (n, k), \tag{18}$$
$$\pi_2(k) = s \text{ if and only if } c(\mathbb{ACT}_{(OV(D_2),<)}(w)) = (n, k), \tag{19}$$
$$\pi(k) = s \text{ if and only if } c(\mathbb{ACT}_{(OV(D_1*D_2),<)}(w)) = (n, k). \tag{20}$$

Let the vertex $v \in V(T_n)$ and $c(v) = (n, i)$. We take $k := {\pi_1}^{-1}(i)$ in (18); $k := {\pi_2}^{-1}(i)$ in (19); $k := \pi^{-1}(i)$ in (20). Then (18)-(20) implies that (17) is equivalent to

$$c(\mathbb{ACT}_{(OV(D_1 * D_2),<)}(v)) = (n, \pi^{-1}(i)) = (n, (\pi_1\pi_2)^{-1}(i)) = \\ = (n, (\pi_2^{-1}\pi_1^{-1})(i)) = c((\mathbb{ACT}_{(OV(D_2),<)} \cdot \mathbb{ACT}_{(OV(D_1),<)})(v)).$$

So, equality (17) for any index i is equivalent to the next

$$\mathbb{ACT}_{(OV(D_1 * D_2),<)}(v) = (\mathbb{ACT}_{(OV(D_2),<)} \cdot \mathbb{ACT}_{(OV(D_1),<)})(v), \tag{21}$$

for any $v \in V(T_n)$ on the nth level.

Note that the last equality holds according to Corollary 19. □

Corollary 21. *The mapping τ is an isomorphism between $Syl_2(S_{2^n})$ and $LT_{2,n}$.*

References

1. Dmitruk Ju.V.: *Structure of Sylow two-subgroups of the symmetric group of degree 2^n*. Ukrainian Math. J. **30** (1978), 117–124.
2. Grigorchuk R.I., Nekrashevich V.V., Sushchanskii V.I.: *Automata, dynamical systems, and groups.* MAIK Nauka/Interperiodica Publishing, Moscow 2000, 128–203.
3. Kaluzhnin L.: *Selected Chapters of Group Theory.* Taras Shevchenko National University of Kyiv, Kyiv 1979 (in Russian).
4. Kaloujnine L.: *La structure des p-groupes de Sylow des groupes symétriques finis.* Ann. Sci. Éc. Norm. Supér. **65** (1948), 239–276.
5. Pawlik B.: *The action of Sylow 2-subgroups of symmetric groups on the set of bases and the problem of isomorphism of their Cayley graphs.* Algebra Discrete Math. **21**, no. 2, (2016), 264–281.
6. Diestel R.: *Graph Theory. 5th edition.* Graduate Texts in Mathematics **173**, Springer-Verlag, Berlin 2017.
7. Robinson D.J.S.: *A Course in the Theory of Groups. 2nd edition.* Graduate Texts in Mathematics **80**, Springer-Verlag, New York 1995.

8. Slupik A.J., Sushchansky V.I.: *Minimal generating sets and Cayley graphs of Sylow p-subgroups of finite symmetric groups.* Algebra Discrete Math. no. 4 (2009), 167–184.
9. Sushchanskii V.I., Sikora V.S.: *Operations on Permutation Groups. Theory and Applications. 2nd edition.* Technodruk, Chernivci 2017 (in Ukrainian).