Linear Time Fourier Transforms of S_{n-k}-invariant Functions on the Symmetric Group S_n

Michael Clausen
Institute of Computer Science
University of Bonn, Germany
Fraunhofer FKIE, Germany
clausen@cs.uni-bonn.de

Paul Hühne
Institute of Computer Science
University of Bonn, Germany
phuehne@uni-bonn.de

ABSTRACT

This paper introduces new techniques for the efficient computation of discrete Fourier transforms (DFTs) of S_{n-k}-invariant functions on the symmetric group S_n. We uncover diamond- and leaf-rake-like structures in Young's seminormal and orthogonal representations. Combining this with both a multiresolution scheme and an anticipation technique for saving scalar multiplications leads to linear time partial FFTs. Following the inductive version of Young's branching rule we obtain a global FFT that computes a DFT of S_{n-k}-invariant functions on S_n in at most $c_k \cdot [S_n : S_{n-k}]$ scalar multiplications and additions, where c_k denotes a positive constant depending only on k. This run-time, which is linear in $[S_n : S_{n-k}]$, is order optimal and improves Maslen's algorithm. For example, it takes less than one second on a standard notebook to run our FFT algorithm for an S_{n-2}-invariant real-valued function on S_n, $n = 5000$.

CCS CONCEPTS

• **Computing methodologies → Symbolic and algebraic algorithms**;

KEYWORDS

FFT; symmetric group; invariant functions;

1 INTRODUCTION

We consider the problem of efficiently computing Fourier transforms of functions on the symmetric group S_n that are constant on left cosets of the subgroup S_{n-k}. This problem arises, e.g., in spectral approaches to multi-target tracking scenarios in computer vision and robotics [12], in the construction of permutation invariant representations of graphs [13], in a kernel-based framework for solving partial ranking problems [10], and in spectral approaches to solve hard combinatorial optimization problems [9].

1.1 DFTs on finite groups

The theoretical basis of the spectral approaches to the above mentioned applications is the ordinary representation theory of finite groups, in particular Wedderburn's structure theorem. According to this theorem, the vector space $\mathbb{C}G := \{\boldsymbol{a} \mid \boldsymbol{a} : G \to \mathbb{C}\}$ of all complex-valued functions on the finite group G equipped with the convolution of functions, $\boldsymbol{a} * \boldsymbol{b} := (G \ni g \mapsto \sum_{xy=g} \boldsymbol{a}(x)\boldsymbol{b}(y))$, becomes an associative algebra. This so-called group algebra (also referred as the signal domain in applications) is isomorphic to an algebra of block-diagonal matrices (the spectral domain)

$$D = \bigoplus_{j=1}^{c} D_j : \mathbb{C}G \to \bigoplus_{j=1}^{c} \mathbb{C}^{d_j \times d_j}. \tag{1}$$

Here, the number c of blocks equals the class number of G and the projections $D_1, \dots, D_c$ form a complete set of pairwise inequivalent irreducible representations of $\mathbb{C}G$. Every such algebra isomorphism D is called a discrete Fourier transform (DFT) on G. With respect to canonical bases in the signal and spectral domain, each DFT on a group G of order N can be described by an $N \times N$ matrix Δ and the transformation of a function $\boldsymbol{a} : G \to \mathbb{C}$ into the spectral domain boils down to a matrix-vector multiplication $\Delta \cdot (\boldsymbol{a}(g))_{g \in G}$. For example, if G is the cyclic group of order N, then $\Delta = (\omega^{ab})_{0 \leq a,b < N}$, $\omega = \exp(2\pi \mathrm{i}/N)$, is the classical DFT-matrix of size N. For an abelian group there is essentially only one DFT-matrix, whereas for a non-abelian group there are infinitely many DFT-matrices, which might differ regarding computational complexity issues. The FFT-problem for a finite group G is to find a suitable DFT on G that allows a transformation of a signal $\boldsymbol{a} : G \to \mathbb{C}$ into the spectral domain with a small number of arithmetic operations (additions and scalar multiplications). At least in a restricted linear computational model, where only scalars of bounded absolute value are at one's disposal, Baum and Clausen [1] proved a general lower complexity bound of order $N \log N$ for evaluating an arbitrary DFT-matrix of a group G of order N.

1.2 Designing FFTs on finite groups

In the last forty years, FFTs for non-abelian finite groups have been investigated by Baum, Beth, Clausen, Diaconis, Maslen, Rockmore, and Willsky among others. For more information see, e.g., the survey article [15] or Chapter 13 in [2]. Almost all FFT algorithms follow a divide-and-conquer technique and are based on the same efficiency principle: produce intermediate results which can be re-used several times. This principle is realized by using DFTs $D = \bigoplus_j D_j$ which are adapted to a suitable chain $\mathcal{C} = (G = G_n > G_{n-1} > \ldots > G_1)$ of subgroups of G, i.e., each D_j restricted to $\mathbb{C}G_i$ is the direct sum of irreducible representations, $D_j \downarrow \mathbb{C}G_i = \bigoplus_\ell D_{i,j,\ell}$, in addition, equivalent irreducible constituents of $D \downarrow \mathbb{C}G_i$ are equal, i.e., $D_{i,j,\ell} \sim D_{i,j',\ell'}$ implies $D_{i,j,\ell} = D_{i,j',\ell'}$ (but not necessarily $(j, \ell) = (j', \ell')$). $\mathcal{C}$-adapted DFTs always exist and, in addition, one can achieve that all D_j and

ISSAC '17, July 25-28, 2017, Kaiserslautern, Germany

ACM ISBN 978-1-4503-5064-8/17/07…$15.00
http://dx.doi.org/10.1145/3087604.3087628

all $D_{i,j,\ell}$ are unitary representations. For the symmetric group S_n, Alfred Young described DFTs adapted to the chain $C_n := (S_n > S_{n-1} > \ldots > S_1)$ about hundred years ago. Recall that a partition $\alpha = (\alpha_1, \ldots, \alpha_r)$ of n, denoted $\alpha \vdash n$, is a non-increasing sequence of positive integers summing to n. As the conjugacy classes of S_n are parametrized via cycle types by the partitions of n, it is natural to index the irreducible representations of $\mathbb{C}S_n$ by those partitions as well:

$$\boldsymbol{\rho}_n = \bigoplus_{\alpha \vdash n} \boldsymbol{\rho}^\alpha : \mathbb{C}S_n \to \bigoplus_{\alpha \vdash n} \mathbb{C}^{d_\alpha \times d_\alpha}. \tag{2}$$

C_n-adaptedness of $\boldsymbol{\rho}_n$ culminates in Young's branching rule

$$\boldsymbol{\rho}^\alpha \downarrow \mathbb{C}S_{n-1} = \bigoplus_\beta \boldsymbol{\rho}^\beta, \tag{3}$$

where the direct sum is over all $\beta \vdash n-1, \beta \subset \alpha$, in lexicographic order. In fact, there are three closely related variants of $\boldsymbol{\rho}_n$: Young's seminormal representation (YSR) $\boldsymbol{\sigma}_n = \bigoplus_{\alpha \vdash n} \boldsymbol{\sigma}^\alpha$, its contragredient variant (YKR) $\boldsymbol{\kappa}_n = \bigoplus_{\alpha \vdash n} \boldsymbol{\kappa}^\alpha$, and Young's orthogonal representation (YOR) $\boldsymbol{\omega}_n = \bigoplus_{\alpha \vdash n} \boldsymbol{\omega}^\alpha$. See Section 2 for a detailed review of YSR, YKR, and YOR. Clausen [3] proved that C_n-adapted DFTs on S_n can be evaluated with $O(N \log^3 N)$ arithmetic operations, where $N := n! = |S_n|$. Maslen [14] improved this bound to $O(N \log^2 N)$.

1.3 FFTs of S_{n-k}-invariant functions on S_n

What is the state of the art for the computation of C_n-adapted Fourier transforms of S_{n-k}-invariant functions on S_n? Let us start with a trivial lower complexity bound. If $T \subset S_n$ is a transversal of the left cosets of S_{n-k} in S_n, then $(n-k)! \cdot \sum_{g \in T} \boldsymbol{a}(g)$ is among the entities which have to be computed. Thus at least $[S_n : S_{n-k}]$ additions and scalar multiplications are needed. Maslen [14] designed an algorithm that computes the Fourier transform of S_{n-k}-invariant functions on S_n with at most

$$\tfrac{3}{4} \cdot k \cdot (2n-k-1) \cdot [S_n : S_{n-k}]$$

arithmetic operations. Thus Maslen's upper bound comes rather close to the lower bound. For $k = 1$, Maslen's algorithm yields the quadratic upper bound $\frac{3}{2} \cdot (n-1) \cdot n$. Clausen and Kakarala [5] proved for $k = 1$ the linear upper bound $3n - 4$, which is order optimal. In his PhD thesis [7], http://hss.ulb.uni-bonn.de/2016/4535/4535.htm, Hühne designed order optimal algorithms for $k = 2$ and $k = 3$.

1.4 Structure and contributions of the paper

Based on the data structures and algorithms proposed in [4], we design in the present paper for each fixed k and all $n > 2k$ an algorithm that computes a DFT of right S_{n-k}-invariant functions on S_n with at most $c_k \cdot [S_n : S_{n-k}]$ arithmetic operations. Thus this new algorithm is order optimal.

In the remaining part of this subsection we describe our contributions and techniques thereby sketching the structure of the paper. In **Section 2** we present a new description of Young's C_n-adapted DFTs which stresses the block structure of the representing matrices. This block structure will be of great importance for the design of efficient algorithms. Then we give an explicit description of the spectral image of the space

$$\mathbb{C}[S_n/S_{n-k}] := \{\boldsymbol{a} \in \mathbb{C}S_n | \boldsymbol{a}(gh) = \boldsymbol{a}(g), \forall g \in S_n, h \in S_{n-k}\}$$

of all right S_{n-k}-invariant functions on S_n with respect to a C_n-adapted DFT on S_n. Finally we introduce family trees $\mathcal{F}_k^n$ as a data structure reflecting iterated applications of Young's branching rule. Our overall FFT-algorithm consists of linear time local FFTs and a global FFT.

Section 3 uses the local FFTs (described later in Section 4) as black boxes and describes the computations along the family tree $\mathcal{F}_k^n$ to get the spectral image of right S_{n-k}-invariant functions on S_n. An analysis of the global FFT shows that the arithmetic cost is proportional to $[S_n : S_{n-k}]$.

Our main contribution is in **Section 4**. Here we design the local FFTs, which are based on diamond- and leaf-rake-like structures closely related to the fact that for the irreducible character χ^β of S_{n-1} induction ($\uparrow$) and restriction ($\downarrow$) nearly commute:

$$\chi^\beta \uparrow S_n \downarrow S_{n-1} = \chi^\beta + (\chi^\beta \downarrow S_{n-2} \uparrow S_{n-1}).$$

This equation, easily deduced from what we call β's 3-generation house, is the source of a reduction technique. The corresponding diamond and leaf-rake computations are described both informally and formally in Section 4. To supersede the diamond computations, we use weighted local FFTs. Combining these weighted local FFTs with a multiresolution scheme results in linear time weighted local FFTs.

2 YOUNG'S ADAPTED DFTS

We assume familiarity with basic concepts of algebraic complexity theory and group representation theory. For detailed accounts, see, e.g., [2, 17]. For a finite group G let $\mathbb{C}G = \{\boldsymbol{a} \mid \boldsymbol{a} : G \to \mathbb{C}\}$ denote its group algebra over $\mathbb{C}$. As usual, we write a function $\boldsymbol{a} : G \to \mathbb{C}$ as a formal sum (with the group element g standing also for its indicator function) $\boldsymbol{a} = \sum_{x \in G} \boldsymbol{a}(x)x =: \sum_{x \in G} \boldsymbol{a}_x x$. Then the multiplication (convolution) in $\mathbb{C}G$ reads as follows

$$\boldsymbol{a} * \boldsymbol{b} = (\sum_{x \in G} \boldsymbol{a}_x x) * (\sum_{y \in G} \boldsymbol{b}_y y) = \sum_{g \in G} (\sum_{xy = g} \boldsymbol{a}_x \boldsymbol{b}_y) g.$$

We describe for all $\alpha \vdash n$ the C_n-adapted irreducible representations $\boldsymbol{\rho}^\alpha \in \{\boldsymbol{\sigma}^\alpha, \boldsymbol{\kappa}^\alpha, \boldsymbol{\omega}^\alpha\}$. For more details see [8].

We identify $\alpha = (\alpha_1, \ldots, \alpha_r) \vdash n$ with its corresponding Young diagram $\bigcup_{i=1}^r \{(i,1), \ldots, (i, \alpha_i)\}$. A standard Young tableau (SYT) of shape α is a bijection $T : \alpha \to [1, n]$ such that the entries are increasing from left to right in each row of T and increasing down each column. SYT^α denotes the set of all SYTs T of shape $|T| := \alpha$.

Iterating Young's branching rule (3) for $\alpha \vdash n$, the α-last letter sequence tree arises quite naturally. The leaves of this ordered tree, consisting of all elements in SYT^α in LLS-order, serve as row and column indices of the representing matrices. Non-leaf nodes will be called SYT tails. Figure 1 shows the α-LLS-tree for $\alpha = (3, 2) \vdash 5$ together with the LLS-orderings on the various levels.

In general, level ℓ of the α-LLS-tree, $\alpha \vdash n$, is a coding of the irreducible constituents of $\boldsymbol{\rho}^\alpha \downarrow \mathbb{C}S_{n-\ell}$. In our example, we obtain for $\ell = 2$: $\boldsymbol{\rho}^{(3,2)} \downarrow \mathbb{C}S_3 = \boldsymbol{\rho}^{(2,1)} \oplus \boldsymbol{\rho}^{(2,1)} \oplus \boldsymbol{\rho}^{(3)}$. Level n tells us that $d_\alpha := \mathrm{degree}(\boldsymbol{\rho}^\alpha) = |\mathrm{SYT}^\alpha|$. This number is given by the celebrated Frame-Robinson-Thrall hook-length formula: $d_\alpha = n!/\prod_{(i,j) \in \alpha} h_{i,j}^\alpha$, where the hook-length $h_{i,j}^\alpha$ is the number of cells $(a, b) \in \alpha$ such that ($a = i$ and $b \geq j$) or ($b = j$ and $a \geq i$). Hook-lengths equal to 1 indicate corner cells. Deleting from α one corner cell yields a partition β of $n-1$ contained in α, which will be called an α-child, denoted $\beta \sqsubset \alpha$ or $\alpha \sqsupset \beta$. By $\alpha^\downarrow$ we denote the set of all α-children. We order the α-children lexicographically. In turn, the partitions α with $\alpha \sqsupset \beta$ will be called the β-parents; $\beta^\uparrow$ denotes the

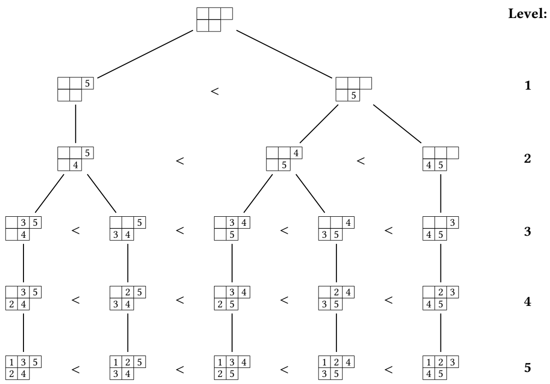

Figure 1: LLS-tree for $\alpha = (3, 2)$

set of all β-parents. Similarly, deleting the cell with the entry n from $T \in \mathrm{SYT}^\alpha$ yields a SYT S. S is called the child of T and T is an S-parent (notation $S \sqsubset T$ or $T \sqsupset S$). Note that in contrast to most partitions, each SYT T has exactly one child but always several parents.

For a partition α we denote by $\mathbf{0}^\alpha$ the zero vector in $\mathbb{C}^\alpha := \mathbb{C}^{d_\alpha}$. For partitions λ and μ of n we denote by $\mathbb{C}^{\lambda\times\mu}$ the space of all $d_\lambda \times d_\mu$ matrices over $\mathbb{C}$. $\mathbf{I}^{\alpha\times\alpha}$ denotes the unit matrix and $\mathbf{0}^{\alpha\times\alpha}$ the zero matrix in $\mathbb{C}^{\alpha\times\alpha}$.

S_n is generated by the transpositions $t_2, \ldots, t_n$, where $t_i := (i-1, i)$. Thanks to Young's branching rule (3), we only need to specify for <u>all</u> n and <u>all</u> $\alpha \vdash n$ the matrices $\boldsymbol{\rho}^\alpha(t_n)$. (E.g., if $\alpha \vdash n$, then $\boldsymbol{\rho}^\alpha(t_{n-1}) = \bigoplus_{\beta\in\alpha\downarrow} \boldsymbol{\rho}^\beta(t_{n-1})$.) We will use the second level of the α-LLS-tree to stress the block structure of the representing matrices. Each SYT tail of that level has only two entries: n and $n-1$. Deleting n and then $n-1$ yields a chain ($\alpha \sqsupset \beta \sqsupset \gamma$). We will identify each SYT tail with the corresponding chain of partitions. Suppose the α-LLS-tree has exactly z elements in its second level. Let $\mathbf{i} := (\alpha \sqsupset \beta^i \sqsupset \gamma^i)$ denote the ith element of the second level. Then $\mathbf{1} < \mathbf{2} < \ldots < \mathbf{z}$ and we can write the representation matrices in block form: $\boldsymbol{\sigma}^\alpha(t_n) = (\Sigma_{\mathbf{ij}})$, $\boldsymbol{\kappa}^\alpha(t_n) = (\mathbf{K}_{\mathbf{ij}})$, and $\boldsymbol{\omega}^\alpha(t_n) = (\Omega_{\mathbf{ij}})$, with suitable $d_{\gamma^i} \times d_{\gamma^j}$ matrices $\Sigma_{\mathbf{ij}}$, $\mathbf{K}_{\mathbf{ij}}$, and $\Omega_{\mathbf{ij}}$.

Theorem 2.1. With this notation the following holds.

(1) $\Sigma_{\mathbf{ij}} = \mathbf{K}_{\mathbf{ij}} = \Omega_{\mathbf{ij}}$ is the zero matrix iff $\gamma^i \neq \gamma^j$.

(2) If $\gamma^i = \gamma^j$, then $\Sigma_{\mathbf{ij}}$, $\mathbf{K}_{\mathbf{ij}}$, and $\Omega_{\mathbf{ij}}$ are nonzero scalar multiples of the unit matrix $\mathbf{I} := \mathbf{I}^{\gamma^i\times\gamma^i}$. More precisely:

- **(R)** If $\mathbf{i}$ is an R-chain, i.e., $n-1$ and n are in the same row of $\mathbf{i}$, then $\Sigma_{\mathbf{ii}} = \mathbf{K}_{\mathbf{ii}} = \Omega_{\mathbf{ii}} = \mathbf{I}$.
- **(C)** If $\mathbf{i}$ is a C-chain, i.e., $n-1$ and n are in the same column of $\mathbf{i}$, then $\Sigma_{\mathbf{ii}} = \mathbf{K}_{\mathbf{ii}} = \Omega_{\mathbf{ii}} = -\mathbf{I}$.
- **(A)** If $\mathbf{i} < \mathbf{j}$ and if $\mathbf{i}$ and $\mathbf{j}$ have $n-1$ and n at positions in $\{(a,b),(c,d)\}$, then $(\mathbf{i},\mathbf{j})$ is called an axial pairing with axial distance $\xi := |a-c|+|b-d|$. Let $q_\xi := \sqrt{\xi^2-1}/\xi$. Then $q_\xi^2 = 1-\xi^{-2}$ and

$$\begin{bmatrix} \Sigma_{\mathbf{ii}} & \Sigma_{\mathbf{ij}} \\ \Sigma_{\mathbf{ji}} & \Sigma_{\mathbf{jj}} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{\mathbf{ii}} & \mathbf{K}_{\mathbf{ji}} \\ \mathbf{K}_{\mathbf{ij}} & \mathbf{K}_{\mathbf{jj}} \end{bmatrix} = \begin{bmatrix} \xi^{-1}\cdot\mathbf{I} & q_\xi^2\cdot\mathbf{I} \\ 1\cdot\mathbf{I} & -\xi^{-1}\cdot\mathbf{I} \end{bmatrix}$$

$$\begin{bmatrix} \Omega_{\mathbf{ii}} & \Omega_{\mathbf{ij}} \\ \Omega_{\mathbf{ji}} & \Omega_{\mathbf{jj}} \end{bmatrix} = \begin{bmatrix} \xi^{-1}\cdot\mathbf{I} & q_\xi\cdot\mathbf{I} \\ q_\xi\cdot\mathbf{I} & -\xi^{-1}\cdot\mathbf{I} \end{bmatrix}.$$

Proof. See [8]. Our claim concerning the block-structure follows from the observation that given an axial pairing $(\mathbf{i},\mathbf{j})$ the transposition $t_n = (n-1, n)$ yields an LLS-order preserving bijection between the leaves of the LLS-subtree corresponding to $\mathbf{i}$ and the leaves corresponding to $\mathbf{j}$. □

Note that $\alpha = (n)$ corresponds to the trivial representation of $\mathbb{C}S_n$. The next result, see [13, 16], describes the $\boldsymbol{\rho}_n$-image of the left ideal $\mathbb{C}[S_n/S_{n-k}] = \mathbb{C}S_n * \sum_{h\in S_{n-k}} h$.

Theorem 2.2. Let $\alpha \vdash n$ and $\boldsymbol{a} \in \mathbb{C}[S_n/S_{n-k}]$. For $T \in \mathrm{SYT}^\alpha$ let $\boldsymbol{a}^T$ denote the Tth column of $\boldsymbol{\rho}^\alpha(\boldsymbol{a})$. Then $\boldsymbol{a}^T = \mathbf{0}^\alpha$, unless $1, \ldots, n-k$ are in the first row of T.

For $n \geq 2k$, the number $a(k)$ of SYTs with n cells having the letters $1, \ldots, n-k$ in the first row, is independent of n. It is well-known that $a(k) = \sum_{\ell=0}^k \binom{k}{\ell} \sum_{\lambda\vdash\ell} d_\lambda$. For example, the values of $a(k)$, for $k \in [0,7]$, read as follows: $(1, 2, 5, 14, 43, 142, 499, 1850)$. For more information consult A005425 in [18].

A SYT T with n cells has Yamanouchi symbol $\langle i_1 \ldots i_n\rangle$ if the letter ℓ is in the i_ℓth row of T. We identify T with its Yamanouchi symbol. Theorem 2.2 suggests to construct $\mathcal{F}_k^n$, the family tree of $\langle 1^{n-k}\rangle$ up to the k-th generation. This tree reflects the iterated branching rule $\boldsymbol{\rho}^{(n-k)} \uparrow \mathbb{C}S_{n-k+1} \uparrow \mathbb{C}S_{n-k+2} \uparrow \ldots \uparrow \mathbb{C}S_n$. Figure 2 shows $\mathcal{F}_3^n$. (We have suppressed the common prefix 1^{n-3}.)

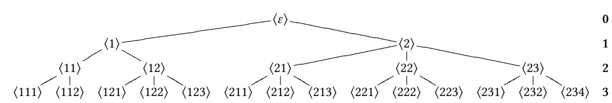

Figure 2: Family tree $\mathcal{F}_3^n$ with level indices

Our FFT-algorithm will be based on those family trees. The root $\langle\varepsilon\rangle \equiv \langle 1^{n-k}\rangle$ in $\mathcal{F}_k^n$ takes the input $\boldsymbol{a} \in \mathbb{C}[S_n/S_{n-k}]$ and the computation will proceed along $\mathcal{F}_k^n$. The leaves of this tree are the output nodes. If $T = \langle i_1 \ldots i_n\rangle \in \mathrm{SYT}^\alpha$ is one of the leaves, then the corresponding output is

$$\boxed{\boldsymbol{a}^T = \boldsymbol{a}^{\langle i_1\ldots i_n\rangle} := T\text{th column of } \boldsymbol{\rho}^\alpha(\boldsymbol{a})}.$$

3 FROM LOCAL TO GLOBAL FFTS

In this section we will describe the computations along $\mathcal{F}_k^n$. Let $\boldsymbol{a} \in \mathbb{C}[S_n/S_{n-k}]$. We decompose $\boldsymbol{a}$ at various levels. These decompositions are based on left coset decompositions $S_n = \bigsqcup_{j_0=1}^n g_{j_0,n}S_{n-1} = \bigsqcup_{j_0=1}^n \bigsqcup_{j_1=1}^{n-1} g_{j_0,n}g_{j_1,n-1}S_{n-2} = \ldots$, where $g_{j,m}$ denotes the cycle $(j, j+1, \ldots, m)$ and $\bigsqcup$ means disjoint union. For $p \leq k$ and $\boldsymbol{j} = (j_0, \ldots, j_{p-1}) \in \boldsymbol{J}_p^n := \prod_{i=0}^{p-1}[1, n-i]$ put $g_{\boldsymbol{j}} := g_{j_0,n} \cdots g_{j_{p-1},n-p+1}$. (Note that $|\boldsymbol{J}_p^n| = (n)_p := \prod_{0\leq i<p}(n-i)$ is a falling factorial.) Then

$$S_n = \bigsqcup_{\boldsymbol{j}\in\boldsymbol{J}_p^n} g_{\boldsymbol{j}}S_{n-p} \quad\text{and}\quad \boldsymbol{a} = \sum_{\boldsymbol{j}\in\boldsymbol{J}_p^n} g_{\boldsymbol{j}}\boldsymbol{a}_{\boldsymbol{j}},$$

where $\boldsymbol{a}_{\boldsymbol{j}} = \sum_{h\in S_{n-p}} \boldsymbol{a}(g_{\boldsymbol{j}}h)h \in \mathbb{C}[S_{n-p}/S_{n-k}]$. Note that for $p = k$, each $\boldsymbol{a}_{\boldsymbol{j}}$ is the constant function $S_{n-k} \ni h \mapsto \boldsymbol{a}(g_{\boldsymbol{j}})$. These constants $\boldsymbol{a}(g_{\boldsymbol{j}})$, $\boldsymbol{j} \in \boldsymbol{J}_k^n$, are the inputs. For $p < k$ we obtain the recurrence

$$\boldsymbol{a}_{\boldsymbol{j}} = \sum_{j_p=1}^{n-p} g_{j_p,n-p}\boldsymbol{a}_{\boldsymbol{j},j_p}, \tag{4}$$

with $\boldsymbol{a}_{j,j_p} \in \mathbb{C}[S_{n-p-1}/S_{n-k}]$. Let us consider the case $p = 1$ in more detail. Here, $\boldsymbol{a} = \sum_{j=1}^{n} g_{j,n}\boldsymbol{a}_j$ with $\boldsymbol{a}_j \in \mathbb{C}[S_{n-1}/S_{n-k}]$. If $T \in \mathrm{SYT}^{\alpha}$ is one of the leaves in $\mathcal{F}_k^n$, then $\boldsymbol{a}^T = \sum_{j=1}^{n} \boldsymbol{\rho}^{\alpha}(g_{j,n})\boldsymbol{a}_j^T$. Thus $\boldsymbol{a}_j^T$ is the Tth column in $\boldsymbol{\rho}^{\alpha}(\boldsymbol{a}_j) = \bigoplus_{\beta\in\alpha\downarrow} \boldsymbol{\rho}^{\beta}(\boldsymbol{a}_j)$. To describe the additional structure of $\boldsymbol{a}_j^T$, we need some preparations. If $\lambda < \mu < \ldots < \nu$ are all the α-children in lexicographic order, then the λ-block of $\boldsymbol{x} \in \mathbb{C}^{\alpha}$ is the vector in $\mathbb{C}^{\lambda}$ consisting of the first d_λ components, the μ-block consists of the next d_μ components, and finally, the last d_ν components of $\boldsymbol{x}$ form the ν-block of $\boldsymbol{x}$.

Definition 3.1. Let β be a child of the partition α.

(1) If $m \geq d_\alpha$, then the *projection operator* $\mathbf{P}^{\alpha} : \mathbb{C}^m \to \mathbb{C}^{\alpha}$ maps every $\boldsymbol{x} \in \mathbb{C}^m$ to the vector of the first d_α components of $\boldsymbol{x}$.

(2) The *projection operator* $\mathbf{P}^{\alpha}_{\beta} : \mathbb{C}^{\alpha} \to \mathbb{C}^{\beta}$ maps $\boldsymbol{x} \in \mathbb{C}^{\alpha}$ to the β-block of $\boldsymbol{x}$.

(3) The *embedding operator* $\mathbf{E}^{\alpha}_{\beta} : \mathbb{C}^{\beta} \to \mathbb{C}^{\alpha}$ applied to $\boldsymbol{y} \in \mathbb{C}^{\beta}$ replaces the β-block of the zero vector $\mathbf{0}^{\alpha}$ by $\boldsymbol{y}$, all other blocks of $\mathbf{0}^{\alpha}$ remain zero.

(4) Every chain of partitions $\beta \supseteq \gamma \supset \delta$, $\beta \vdash n-1$, $\delta \vdash m-1$ defines a *cancellation operator* $\mathbb{C}^{\beta\times n} \ni \boldsymbol{X} \mapsto \boldsymbol{X}^{\gamma}_{\delta} \in \mathbb{C}^{\delta\times m}$ as follows. If $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, then $\boldsymbol{X}^{\gamma}_{\delta} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m)$, where $\boldsymbol{y}_j := \mathbf{P}^{\gamma}_{\delta}\mathbf{P}^{\gamma}\boldsymbol{x}_j$.

Example 3.2. If $\alpha = (3,2,1)$, then $(2,2,1) < (3,1,1) < (3,2)$ are all α-children in lexicographic order. Let $\boldsymbol{x} \in \mathbb{C}^{(3,2,1)}$, $\boldsymbol{y} \in \mathbb{C}^{(3,1,1)}$, and $\boldsymbol{X} = (\boldsymbol{X}_{i,j}) \in \mathbb{C}^{(3,2)\times 6}$. Then $\boldsymbol{X}^{(3,1)}_{(2,1)} = (\boldsymbol{X}_{i,j})_{i\in[1,2],j\in[1,4]}$ and

$$\boldsymbol{x} = \left[\begin{array}{c} \mathbf{P}^{(3,2,1)}_{(2,2,1)}\boldsymbol{x} \\ \hline \mathbf{P}^{(3,2,1)}_{(3,1,1)}\boldsymbol{x} \\ \hline \mathbf{P}^{(3,2,1)}_{(3,2)}\boldsymbol{x} \end{array}\right], \quad \mathbf{E}^{(3,2,1)}_{(3,1,1)}\boldsymbol{y} = \left[\begin{array}{c} \mathbf{0}^{(2,2,1)} \\ \hline \boldsymbol{y} \\ \hline \mathbf{0}^{(3,2)} \end{array}\right].$$

The usefulness of the embedding operator is shown by the following. (We keep the above notation and continue with the case $p = 1$.)

LEMMA 3.3. If $S \in \mathrm{SYT}^{\beta}$ is the child of $T \in \mathrm{SYT}^{\alpha}$, then

$$\boxed{\boldsymbol{a}_j^T = \mathbf{E}^{\alpha}_{\beta}\boldsymbol{a}_j^S} \quad \text{and} \quad \boxed{\boldsymbol{a}^T = \textstyle\sum_{j=1}^{n} \boldsymbol{\rho}^{\alpha}(g_{j,n})\mathbf{E}^{\alpha}_{\beta}\boldsymbol{a}_j^S}.$$

PROOF. Each level of the α-LLS tree corresponds to a partitioning of the set $[1, d_\alpha]$ of row and column indices of $\boldsymbol{\rho}^{\alpha}(\boldsymbol{a})$ into subintervals. Removing in the α-LLS tree its root and in all SYT tails the cell containing n, results in an ordered forest describing the block and index structure of the equation $\boldsymbol{\rho}^{\alpha} \downarrow \mathbb{C}S_{n-1} = \bigoplus_{\beta\in\alpha\downarrow} \boldsymbol{\rho}^{\beta}$: the forest's roots describe the block structure and the forest's leaves correspond to the row and column indices at the S_{n-1}-level. In particular, by removing n, T is replaced by its child S. □

More generally, if $L \in \mathrm{SYT}^{\lambda}$ is a non-leaf node of the family tree $\mathcal{F}_k^n$ and $M \in \mathrm{SYT}^{\mu}$, $\mu \vdash m = n-p$, is a parent of L, i.e., $L \subset M$, then for all $\boldsymbol{j} \in \boldsymbol{J}_p^n$

$$\boldsymbol{a}_{\boldsymbol{j}}^M = \sum_{j_p=1}^{m} \boldsymbol{\rho}^{\mu}(g_{j_p,m})\mathbf{E}^{\mu}_{\lambda}\boldsymbol{a}^L_{\boldsymbol{j},j_p}. \tag{5}$$

The crucial point is that all the $\boldsymbol{a}^L_{\boldsymbol{j},j_p}$ on the right hand side remain, if we replace M by any other parent of L. We will view the $\boldsymbol{a}^L_{\boldsymbol{j},j_p}$ as common inputs. This observation gives rise to the following fundamental definition.

Definition 3.4. Let $\beta \vdash n-1$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{C}^{\beta}$. For all $\alpha \in \beta^{\uparrow}$ define $\boldsymbol{s}^{\alpha}_{\beta}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) := \sum_{j=1}^{n} \boldsymbol{\rho}^{\alpha}(g_{j,n})\mathbf{E}^{\alpha}_{\beta}\boldsymbol{x}_j$.

With this terminology, Equation (5) reads as follows:

$$\boxed{\boldsymbol{a}_{\boldsymbol{j}}^M = \boldsymbol{s}^{\mu}_{\lambda}(\boldsymbol{a}^L_{\boldsymbol{j},1}, \ldots, \boldsymbol{a}^L_{\boldsymbol{j},m})}. \tag{6}$$

In Section 4 a family of local FFT-algorithms $\mathbf{FFT}_\beta$, $\beta \vdash n-1$, will be designed. $\mathbf{FFT}_\beta$ expects n input vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{C}^{\beta}$ and outputs $\boldsymbol{s}^{\alpha}_{\beta}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ for every $\alpha \in \beta^{\uparrow}$. In Section 4 we will prove the following.

THEOREM 3.5. Let $\beta = (n-k, \mu) \vdash n-1$, $\mu \vdash k-1$, and $n-k \geq \mu_1$. Then the arithmetic cost of running $\mathbf{FFT}_\beta$ is of order $[S_n : S_{n-k}] = (n)_k$.

With the help of these local FFT-algorithms we will now show how to compute the $\boldsymbol{\rho}_n$-image of $\boldsymbol{a} \in \mathbb{C}[S_n/S_{n-k}]$.

The global FFT along $\mathcal{F}_k^n$

1. **Input**: $\boldsymbol{a} \in \mathbb{C}[S_n/S_{n-k}]$
2. **Initialization:**
3. **for** $\boldsymbol{j} \in \boldsymbol{J}_k^n$ compute $\boldsymbol{a}_{\boldsymbol{j}}^{\langle 1^{n-k}\rangle} = (n-k)! \cdot \boldsymbol{a}(g_{\boldsymbol{j}})$
4. **Recursion:**
5. **for** $\ell = 0$ to $k-1$ **do**
6.   **for** $L \in \mathrm{SYT}^{\lambda}$ at level ℓ of $\mathcal{F}_k^n$ **do**
7.   **for** $\boldsymbol{j} \in \boldsymbol{J}^n_{k-\ell-1}$ **do**
8. $\mathbf{FFT}_\lambda(\boldsymbol{a}^L_{\boldsymbol{j},1}, \ldots, \boldsymbol{a}^L_{\boldsymbol{j},n-k+\ell+1})$
9.   **end for**
10.   **Local outputs**:
11. $\boldsymbol{a}^M_{\boldsymbol{j}}$, **for all** $M \supset L$, $\boldsymbol{j} \in \boldsymbol{J}^n_{k-\ell-1}$
12.   **end for**
13. **end for**
14. **Outputs:**
15. $\boldsymbol{a}^T$, **for all** T at level k of $\mathcal{F}_k^n$

THEOREM 3.6. Let $k \geq 1$. There is a positive constant c_k such that every C_n-adapted Fourier transform of S_{n-k}-invariant functions on S_n can be computed with at most $c_k \cdot [S_n : S_{n-k}]$ arithmetic operations.

PROOF. cost(Line 3) $= (n)_k$. Line 8: As $\lambda_1 \geq n-k$, we see by Theorem 3.5 that cost($\mathbf{FFT}_\lambda$) $= O((n-k+\ell+1)_{\ell+1})$. Moreover, by Line 7, $\mathbf{FFT}_\lambda$ has to be called $(n)_{k-\ell-1}$ times. As $(n-k+\ell+1)_{\ell+1} \cdot (n)_{k-\ell-1} = (n)_k$, we see that this rough upper bound is independent of the shape of L. Thus $O((n)_k \cdot (1 + \sum_{\ell=0}^{k-1} a(\ell)))$ is an upper bound for the overall cost, where $a(\ell)$ is the ℓth number in A005425. □

4 ON THE DESIGN OF LOCAL FFTS

In this section we design for each $\beta \vdash n-1$ an algorithm, $\mathbf{FFT}_\beta$, which on input $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathbb{C}^{\beta\times n}$ computes $\boldsymbol{s}^{\alpha}_{\beta}(\boldsymbol{X})$, for all β-parents α. Before going into technical details, we would like to illustrate the rationale behind this linear time procedure. (To be as concrete as possible, we use from now on $\boldsymbol{\kappa}_n$ instead of $\boldsymbol{\rho}_n$.)

4.1 Diamond and leaf-rake computations

Our fast algorithm is based on diamond- and leaf-rake-like structures closely related to the fact that for the irreducible character χ^β of S_{n-1} induction ($\uparrow$) and restriction ($\downarrow$) nearly commute:

$$\chi^\beta \uparrow S_n \downarrow S_{n-1} = \chi^\beta + (\chi^\beta \downarrow S_{n-2} \uparrow S_{n-1}). \tag{7}$$

The validity of (7) as well as the diamond-like structures are easily seen by considering β's 3-generation house, consisting of $\beta^\uparrow$ (top level), $\beta^\downarrow$ (ground level), and $\beta^{\uparrow\downarrow} := \bigcup_{\alpha\in\beta^\uparrow} \alpha^\downarrow \stackrel{!}{=} \beta^{\downarrow\uparrow} := \bigcup_{\delta\in\beta^\downarrow} \delta^\uparrow$ (mid level), see Figure 3.

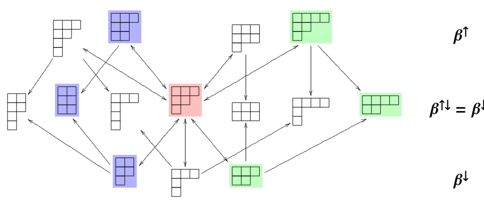

Figure 3: The 3-generation house of $\beta = (3, 2, 1)$

Every $\gamma \in \beta^{\uparrow\downarrow}$, different from β, defines a diamond, which consists of $\alpha = \beta \cup \gamma \vdash n$ in the top level, $\delta = \beta \cap \gamma \vdash n-2$ in the ground level, and β and γ in the mid level, see also [14]. If γ is lexicographically smaller than β, then these four partitions form a left diamond, otherwise a right diamond (w.r.t. β). Figure 3 shows a **l**eft diamond (red & b**l**ue), and a **r**ight diamond (red & g**r**een). Every diamond has a weight depending on $\boldsymbol{\rho}_n$. For $\boldsymbol{\rho}_n = \boldsymbol{\kappa}_n$ we obtain the following. Left diamonds have always the weight 1, whereas the right diamond defined by $\beta < \gamma$ has the weight $1 - \xi^{-2}$, where ξ denotes the axial distance of β and γ, which is the diameter of the symmetric difference of β and γ in the 1-norm. Figure 4 shows a left and a right diamond with labeled directed edges (recall Definition 3.1). On input $\boldsymbol{X} \in \mathbb{C}^{\beta\times n}$, the path over the hill, $\beta \to \alpha \to \gamma$, will be interpreted as the question: How does the γ-block of $\boldsymbol{s}^\alpha_\beta(\boldsymbol{X})$ looks like? The path across the valley, $\beta \to \delta \to \gamma$, will give the answer: project the first $n-1$ inputs via $\mathbf{P}^\beta_\delta$, then evaluate $\boldsymbol{s}^\gamma_\delta$ at $\boldsymbol{X}^\beta_\delta = (\mathbf{P}^\beta_\delta \boldsymbol{x}_1, \ldots, \mathbf{P}^\beta_\delta \boldsymbol{x}_{n-1})$ and, finally, multiply with the diamond's weight.

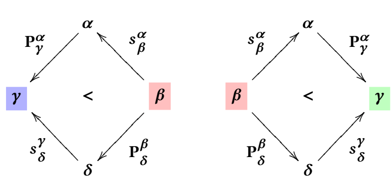

Figure 4: Left & right diamond computations

But how does the β-block of $\boldsymbol{s}^\alpha_\beta(\boldsymbol{X})$ looks like? It turns out that this block is obtained as follows: for all β-children δ evaluate $\boldsymbol{s}^\beta_\delta$ at $\boldsymbol{X}^\beta_\delta$, then form an α-specific weighted sum of all these items and finally add $\boldsymbol{x}_n$. This is illustrated by the following leaf-rake-like structure (we suppose that $\lambda < \ldots < \mu < \ldots < \nu$ are all β-children):

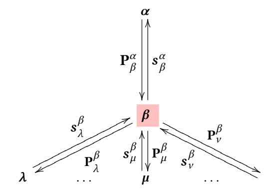

Figure 5: Leaf-rake computations

The leaf-rakes corresponding to the various β-parents have all the prongs $\boldsymbol{s}^\beta_\delta(\boldsymbol{X}^\beta_\delta)$, $\delta \in \beta^\downarrow$, in common, only their weights vary with the sticks $\mathbf{P}^\alpha_\beta \boldsymbol{s}^\alpha_\beta(\boldsymbol{X})$, $\alpha \in \beta^\uparrow$. The above discussion also shows that (up to diamond and leaf-rake computations) calling $\mathbf{FFT}_\beta$ reduces to calling all $\mathbf{FFT}_\delta$, $\delta \in \beta^\downarrow$. Now we come to the more technical part.

Lemma 4.1. Let $\alpha \vdash n$ and let β, γ be two different α-children.

(1) $\mathbf{P}^\alpha_\beta \mathbf{E}^\alpha_\beta$ is the identity on $\mathbb{C}^\beta$, whereas $\mathbf{P}^\alpha_\beta \mathbf{E}^\alpha_\gamma : \mathbb{C}^\gamma \to \mathbb{C}^\beta$ is the zero operator.

(2) Let $\boldsymbol{x} \in \mathbb{C}^\alpha$. Then $\boldsymbol{x} = \sum_{\beta\in\alpha^\downarrow} \mathbf{E}^\alpha_\beta \mathbf{P}^\alpha_\beta \boldsymbol{x}$.

(3) Let $\boldsymbol{B} = \bigoplus_{\beta\in\alpha^\downarrow} \boldsymbol{B}_\beta \in \bigoplus_{\beta\in\alpha^\downarrow} \mathbb{C}^{\beta\times\beta}$ be a block diagonal matrix, whose blocks are ordered lexicographically. Then $\boldsymbol{B} = \sum_{\beta\in\alpha^\downarrow} \mathbf{E}^\alpha_\beta \boldsymbol{B}_\beta \mathbf{P}^\alpha_\beta$.

Proof. **(1), (2)**: obvious. **(3)**: For $\boldsymbol{x} \in \mathbb{C}^\alpha$, $\boldsymbol{B}_\beta \mathbf{P}^\alpha_\beta \boldsymbol{x}$ is the β-block of $\boldsymbol{B}\boldsymbol{x}$. By (2), $\boldsymbol{B}\boldsymbol{x} = \sum_{\beta\in\alpha^\downarrow} \mathbf{E}^\alpha_\beta \boldsymbol{B}_\beta \mathbf{P}^\alpha_\beta \boldsymbol{x}$. As this is true for all $\boldsymbol{x} \in \mathbb{C}^\alpha$, our claim follows. □

We come back to Theorem 2.1 and its notation. Suppose β is a child of $\alpha \vdash n$ and $\boldsymbol{x} \in \mathbb{C}^\beta$. Put $\widetilde{\boldsymbol{x}} := \mathbf{E}^\alpha_\beta \boldsymbol{x}$. We subdivide $\widetilde{\boldsymbol{x}}$ with respect to the second level of the α-LLS tree: $\widetilde{\boldsymbol{x}}_{\mathbf{1}}, \ldots, \widetilde{\boldsymbol{x}}_{\mathbf{z}}$, where $\mathbf{i} = (\alpha \supset \beta^i \supset \gamma^i)$. Let $p = 1 + \sum_{\gamma\in\alpha^\downarrow:\gamma<\beta} |\gamma^\downarrow|$ and $q = \sum_{\gamma\in\alpha^\downarrow:\gamma\le\beta} |\gamma^\downarrow|$. Then $\beta^{p-1} < \beta = \beta^p = \ldots = \beta^q < \beta^{q+1}$. Moreover, $\gamma^p < \ldots < \gamma^q$. Thus $\widetilde{\boldsymbol{x}}_{\mathbf{j}} = \mathbf{0}$, for all indices $j \notin [p, q]$. In addition, if $(\mathbf{i}, \mathbf{j})$ is an axial pairing, then at most one of the indices i, j can belong to the interval $[p, q]$. Thus if $i \in [p, q]$, then $j > q$, which forces $\widetilde{\boldsymbol{x}}_{\mathbf{j}} = \mathbf{0}$. On the other hand, if $j \in [p, q]$, then $i < p$, hence $\widetilde{\boldsymbol{x}}_{\mathbf{i}} = \mathbf{0}$.

Corollary 4.2. Let $\alpha \vdash n$, $\beta \in \alpha^\downarrow$. Let $\boldsymbol{x} \in \mathbb{C}^\beta$ and put $\widetilde{\boldsymbol{x}} := \mathbf{E}^\alpha_\beta \boldsymbol{x}$. Let $\mathbf{K} = (\mathbf{K}_{\mathbf{ij}}) := \boldsymbol{\kappa}^\alpha(t_n)$, $\widetilde{\boldsymbol{y}} = (\widetilde{\boldsymbol{y}}_{\mathbf{i}}) = \mathbf{K} \cdot \widetilde{\boldsymbol{x}}$.

(1) If γ^i is not contained in β, then $\widetilde{\boldsymbol{y}}_{\mathbf{i}} = \mathbf{0}^{\gamma^i}$.

(2) If $\mathbf{i}$ is an R-chain and $\beta^i = \beta$, then $\widetilde{\boldsymbol{y}}_{\mathbf{i}} = \widetilde{\boldsymbol{x}}_{\mathbf{i}}$.

(3) If $\mathbf{i}$ is a C-chain and $\beta^i = \beta$, then $\widetilde{\boldsymbol{y}}_{\mathbf{i}} = -\widetilde{\boldsymbol{x}}_{\mathbf{i}}$.

(4) Let $(\mathbf{i}, \mathbf{j})$ be an axial pairing with axial distance ξ. If $i \in [p, q]$ then $\widetilde{\boldsymbol{y}}_{\mathbf{i}} = \xi^{-1} \cdot \widetilde{\boldsymbol{x}}_{\mathbf{i}}$ and $\widetilde{\boldsymbol{y}}_{\mathbf{j}} = (1 - \xi^{-2}) \cdot \widetilde{\boldsymbol{x}}_{\mathbf{i}}$. If $j \in [p, q]$, then $\widetilde{\boldsymbol{y}}_{\mathbf{i}} = \widetilde{\boldsymbol{x}}_{\mathbf{j}}$ and $\widetilde{\boldsymbol{y}}_{\mathbf{j}} = -\xi^{-1} \cdot \widetilde{\boldsymbol{x}}_{\mathbf{j}}$.

We reformulate Corollary 4.2. Note that $\widetilde{\boldsymbol{y}}_i = \mathbf{P}^{\beta^i}_{\gamma^i} \mathbf{P}^\alpha_{\beta^i} \widetilde{\boldsymbol{y}}$.

Corollary 4.3. Let β, γ denote two different α-children with corresponding axial distance $\xi_{\beta,\gamma}$ and $\delta := \beta \cap \gamma$. Let $\boldsymbol{x} \in \mathbb{C}^\beta$ and $\boldsymbol{K} := \boldsymbol{\kappa}^\alpha(t_n)$. Then the level-2-blocks of $\boldsymbol{y} := \boldsymbol{K}\mathbf{E}^\alpha_\beta \boldsymbol{x}$ satisfy:

(1) $\mathbf{P}^\kappa_\varepsilon \mathbf{P}^\alpha_\kappa \boldsymbol{y} = \mathbf{0}^\varepsilon$, if $\alpha \supset \kappa \supset \varepsilon$ and ε is not contained in β.

(2) $\mathbf{P}^\gamma_\delta \mathbf{P}^\alpha_\gamma \boldsymbol{y} = \mathbf{P}^\beta_\delta \boldsymbol{x}$, if $\gamma < \beta$.

(3) $\mathbf{P}^{\gamma}_{\delta}\mathbf{P}^{\alpha}_{\gamma}\boldsymbol{y} = (1-\xi^{-2}_{\beta,\gamma})\cdot\mathbf{P}^{\beta}_{\delta}\boldsymbol{x}$, if $\gamma > \beta$.

(4) $\mathbf{P}^{\beta}_{\varepsilon}\mathbf{P}^{\alpha}_{\beta}\boldsymbol{y} = \text{⅄}^{\alpha}_{\beta,\varepsilon}\mathbf{P}^{\beta}_{\varepsilon}\boldsymbol{x}$, where

$$\text{⅄}^{\alpha}_{\beta,\varepsilon} = \begin{cases} +1 & \text{if } \alpha \supset \beta \supset \varepsilon \text{ is an R-chain} \\ -1 & \text{if } \alpha \supset \beta \supset \varepsilon \text{ is a C-chain} \\ \xi^{-1}_{\beta,\gamma} & \text{if } \varepsilon = \delta \text{ and } \gamma > \beta \\ -\xi^{-1}_{\beta,\gamma} & \text{if } \varepsilon = \delta \text{ and } \gamma < \beta. \end{cases}$$

Now we will discuss the diamond and leaf-rake computations.

Theorem 4.4. Let $\beta \vdash n-1$, $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathbb{C}^{\beta\times n}$.

(1) If $\gamma \in \beta^{\downarrow\uparrow}$ is different from β, $\xi_{\beta,\gamma}$ the corresponding axial distance, $\alpha := \beta\cup\gamma$, and $\delta := \beta\cap\gamma$, then the γ-block of $\boldsymbol{s}^{\alpha}_{\beta}(X)$ satisfies

$$\boxed{\mathbf{P}^{\alpha}_{\gamma}\boldsymbol{s}^{\alpha}_{\beta}(X) = \diamond_{\beta,\gamma}\cdot\boldsymbol{s}^{\gamma}_{\delta}(X^{\beta}_{\delta})},$$

$\diamond_{\beta,\gamma} := 1$, if $\gamma < \beta$, $\diamond_{\beta,\gamma} := 1-(\xi_{\beta,\gamma})^{-2}$, if $\gamma > \beta$.

(2) The β-block of $\boldsymbol{s}^{\alpha}_{\beta}(X)$ satisfies

$$\boxed{\mathbf{P}^{\alpha}_{\beta}\boldsymbol{s}^{\alpha}_{\beta}(X) = \sum_{\varepsilon\in\beta^{\downarrow}} \text{⅄}^{\alpha}_{\beta,\varepsilon}\boldsymbol{s}^{\beta}_{\varepsilon}(X^{\beta}_{\varepsilon}) + \boldsymbol{x}_n},$$

where $\text{⅄}^{\alpha}_{\beta,\varepsilon}$ are the weights from Corollary 4.3(4).

Proof. For $j < n$ put $\boldsymbol{y}_j := \boldsymbol{\kappa}^{\alpha}(t_n)\mathbf{E}^{\alpha}_{\beta}\boldsymbol{x}_j$.

(1) As $\mathbf{P}^{\alpha}_{\gamma}\mathbf{E}^{\alpha}_{\lambda} : \mathbb{C}^{\lambda} \to \mathbb{C}^{\gamma}$ is the zero operator for $\gamma \neq \lambda$ whereas $\mathbf{P}^{\alpha}_{\gamma}\mathbf{E}^{\alpha}_{\gamma}$ is the identity on $\mathbb{C}^{\gamma}$, we get by Lemma 4.1 $\mathbf{P}^{\alpha}_{\gamma}\boldsymbol{\kappa}^{\alpha}(g_{j,n-1})\boldsymbol{y}_j = \mathbf{P}^{\alpha}_{\gamma}\sum_{\lambda\in\alpha^{\downarrow}}\mathbf{E}^{\alpha}_{\lambda}\boldsymbol{\kappa}^{\lambda}(g_{j,n-1})\mathbf{P}^{\alpha}_{\lambda}\boldsymbol{y}_j$. Thus

$$\mathbf{P}^{\alpha}_{\gamma}\boldsymbol{\kappa}^{\alpha}(g_{j,n-1})\boldsymbol{y}_j = \boldsymbol{\kappa}^{\gamma}(g_{j,n-1})\mathbf{P}^{\alpha}_{\gamma}\boldsymbol{y}_j. \tag{8}$$

Combining this with $\mathbf{P}^{\alpha}_{\gamma}\mathbf{E}^{\alpha}_{\beta}\boldsymbol{x}_n = \mathbf{0}^{\gamma}$ yields

$$\begin{aligned}
\mathbf{P}^{\alpha}_{\gamma}\boldsymbol{s}^{\alpha}_{\beta}(X) &= \mathbf{P}^{\alpha}_{\gamma}\sum_{j=1}^{n-1}\boldsymbol{\kappa}^{\alpha}(g_{j,n})\mathbf{E}^{\alpha}_{\beta}\boldsymbol{x}_j = \\
\sum_{j=1}^{n-1}\mathbf{P}^{\alpha}_{\gamma}\boldsymbol{\kappa}^{\alpha}(g_{j,n-1})\boldsymbol{y}_j &\overset{(8)}{=} \sum_{j=1}^{n-1}\boldsymbol{\kappa}^{\gamma}(g_{j,n-1})\mathbf{P}^{\alpha}_{\gamma}\boldsymbol{y}_j \\
&= \sum_{j=1}^{n-1}\boldsymbol{\kappa}^{\gamma}(g_{j,n-1})\sum_{\lambda\in\gamma^{\downarrow}}\mathbf{E}^{\gamma}_{\lambda}\mathbf{P}^{\gamma}_{\lambda}\mathbf{P}^{\alpha}_{\gamma}\boldsymbol{y}_j \\
\text{by Cor. 4.3(1)} &= \sum_{j=1}^{n-1}\boldsymbol{\kappa}^{\gamma}(g_{j,n-1})\mathbf{E}^{\gamma}_{\delta}\mathbf{P}^{\gamma}_{\delta}\mathbf{P}^{\alpha}_{\gamma}\boldsymbol{y}_j \\
\text{by Cor. 4.3(2+3)} &= \diamond_{\beta,\gamma}\sum_{j=1}^{n-1}\boldsymbol{\kappa}^{\gamma}(g_{j,n-1})\mathbf{E}^{\gamma}_{\delta}\mathbf{P}^{\beta}_{\delta}\boldsymbol{x}_j \\
&= \diamond_{\beta,\gamma}\cdot\boldsymbol{s}^{\gamma}_{\delta}(X^{\beta}_{\delta}).
\end{aligned}$$

This proves our first statement.

(2) Note that

$$\mathbf{P}^{\alpha}_{\beta}\boldsymbol{y}_j = \sum_{\varepsilon\in\beta^{\downarrow}}\mathbf{E}^{\beta}_{\varepsilon}\mathbf{P}^{\beta}_{\varepsilon}\mathbf{P}^{\alpha}_{\beta}\boldsymbol{y}_j. \tag{9}$$

As $\mathbf{P}^{\alpha}_{\beta}\mathbf{E}^{\alpha}_{\beta}\boldsymbol{x}_n = \boldsymbol{x}_n$ we get

$$\begin{aligned}
\mathbf{P}^{\alpha}_{\beta}\boldsymbol{s}^{\alpha}_{\beta}(X) &= \sum_{j=1}^{n}\mathbf{P}^{\alpha}_{\beta}\boldsymbol{\kappa}^{\alpha}(g_{j,n})\mathbf{E}^{\alpha}_{\beta}\boldsymbol{x}_j = \boldsymbol{x}_n + \sum_{j<n}\mathbf{P}^{\alpha}_{\beta}\boldsymbol{\kappa}^{\alpha}(g_{j,n-1})\boldsymbol{y}_j \\
&\overset{(8)}{=} \boldsymbol{x}_n + \sum_{j<n}\boldsymbol{\kappa}^{\beta}(g_{j,n-1})\mathbf{P}^{\alpha}_{\beta}\boldsymbol{y}_j \\
&\overset{(9)}{=} \boldsymbol{x}_n + \sum_{j<n}\boldsymbol{\kappa}^{\beta}(g_{j,n-1})\sum_{\varepsilon\in\beta^{\downarrow}}\mathbf{E}^{\beta}_{\varepsilon}(\mathbf{P}^{\beta}_{\varepsilon}\mathbf{P}^{\alpha}_{\beta}\boldsymbol{y}_j) \\
&\overset{(*)}{=} \boldsymbol{x}_n + \sum_{\varepsilon\in\beta^{\downarrow}}\text{⅄}^{\alpha}_{\beta,\varepsilon}\sum_{j<n}\boldsymbol{\kappa}^{\beta}(g_{j,n-1})\mathbf{E}^{\beta}_{\varepsilon}\mathbf{P}^{\beta}_{\varepsilon}\boldsymbol{x}_j \\
&= \boldsymbol{x}_n + \sum_{\varepsilon\in\beta^{\downarrow}}\text{⅄}^{\alpha}_{\beta,\varepsilon}\boldsymbol{s}^{\beta}_{\varepsilon}(X^{\beta}_{\varepsilon}),
\end{aligned}$$

where $(*)$ means by Cor. 4.3(4). This proves statement (2). □

4.2 Weighted local FFTs and multiresolution

To supersede the diamond computations we anticipate them by integrating them into the output behavior. To this end we will work with weighted local FFTs $\mathbf{FFT}^{\boldsymbol{w}}_{\beta}$. More precisely, $\mathbf{FFT}^{\boldsymbol{w}}_{\beta}$ expects as inputs a matrix $X \in \mathbb{C}^{\beta\times n}$ and a weight function $\boldsymbol{w}_{\beta} : \beta^{\uparrow} \to (0;1] := \{w \in \mathbb{R} \mid 0 < w \leq 1\}$. $\mathbf{FFT}^{\boldsymbol{w}}_{\beta}[X, \boldsymbol{w}_{\beta}]^{\alpha} := \boldsymbol{w}_{\beta}(\alpha)\cdot\boldsymbol{s}^{\alpha}_{\beta}(X)$ is the α-output, $\alpha \in \beta^{\uparrow}$.

Inspired by Theorem 4.4 and [11], our weighted local FFT-algorithm $\mathbf{FFT}^{\boldsymbol{w}}_{\beta}$ will follow β's multiresolution tree, illustrated in Figure 6 for $\beta = (5,2,1)$. The rightmost tree in Figure 6 and the matrix below indicate those parts of the original input matrix $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = X_{0,0}$ which will serve as inputs for smaller FFTs. Put $\boldsymbol{x}'_{n-i} := \mathbf{P}^{\beta_{i,0}}\boldsymbol{x}_{n-i}$. (More details will follow after the example.)

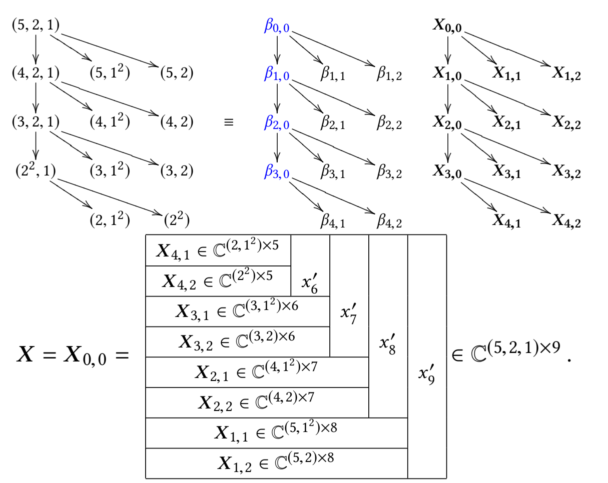

Figure 6: Multiresolution tree for $\beta = (5,2,1)$ and corresponding partition of original input matrix X

$\mathbf{FFT}^{\boldsymbol{w}}_{\beta}$ works in a bottom-up manner. Figure 7 illustrates this for $\beta = (5,2,1)$, where $\mathbf{FFT}^{\boldsymbol{w}}_{i,j} := \mathbf{FFT}^{\boldsymbol{w}}_{\beta_{i,j}}$.

After this illustrating example, we carry on with the more technical part.

4.3 Multiresolution trees

Let $\beta = (n-k, \mu)$, $\mu \vdash k-1$ and $n-k \geq \mu_1$. Let $\mu^{\uparrow} = (\lambda^1 < \ldots < \lambda^s)$ and $\mu^{\downarrow} = (\nu^1 < \ldots < \nu^r)$ are all μ-parents and all μ-children. Put $f := n-k-\mu_1$. Then β's multiresolution tree has $f+2$ levels.

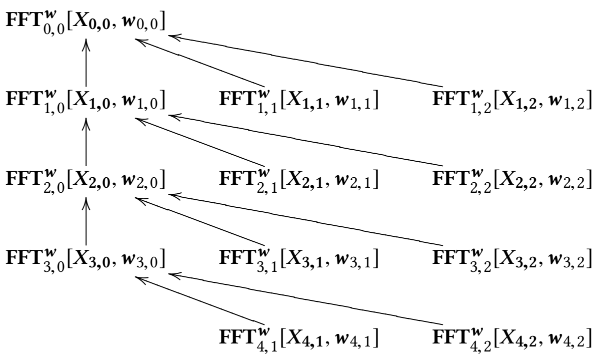

Figure 7: Sketch of $\mathbf{FFT}^{w}_{\beta}$ for $\beta = (5, 2, 1)$

The only partition of level 0 is $\beta_{0,0} := \beta$. For $i \in [1, f]$, level i consists of all $\beta_{i-1,0}$-children: $\beta_{i,0} < \beta_{i,1} < \ldots < \beta_{i,r}$, where $\beta_{i,0} = (n - k - i, \mu)$ and $\beta_{i,j} = (n - k - i + 1, \nu^j)$, $j \in [1, r]$. Finally, level $f + 1$ consists of all $\beta_{f+1,j} = (\mu_1, \nu^j)$, $j \in [1, r]$. (As $(\mu_1 - 1, \mu)$ is not a partition, $\beta_{f+1,0}$ is absent in this level.)

If $X \in \mathbb{C}^{\beta \times n}$ is the input matrix, then $X_{0,0} := X$ and (recall Definition 3.1(4)) $X_{i,j} := X^{\beta_{i-1,0}}_{\beta_{i,j}}$.

4.4 Compatible propagation of weights

We describe the propagation of a weight function $w_\beta : \beta^{\uparrow} \to (0; 1]$ to weight functions for all β-children. For all $\gamma \in \beta^{\uparrow\downarrow}$, $\gamma \neq \beta$, and all $\varepsilon \in \beta^{\downarrow}$ put

$$\boxed{w_{\beta\cap\gamma}(\gamma) := w_\beta(\beta \cup \gamma) \cdot \diamond_{\beta,\gamma}} \quad \text{and} \quad \boxed{w_\varepsilon(\beta) := 1}. \tag{10}$$

In other words, every $\alpha \in \beta^{\uparrow}$ broadcasts its weight $w_\beta(\alpha)$ to all α-children $\gamma \neq \beta$. Moreover, β itself gets the weight 1. If $\alpha = \beta \cup \gamma$, this corresponds to the factor $w_\beta(\beta \cup \gamma)$ in (10). In Figure 8 this broadcasting is indicated by dotted downarrows. To get the final weight of $w_{\beta\cap\gamma}(\gamma)$ one has to multiply $w_\beta(\beta\cup\gamma)$ with the diamond coefficient $\diamond_{\beta,\gamma}$, which is equal to 1, if $\gamma < \beta$ or equal to $q^2_\xi = 1 - \xi^{-2}$, if $\beta < \gamma$ and ξ is the axial distance between these two partitions. As $\beta^{\uparrow\downarrow} = \{\beta\} \sqcup \bigsqcup_{\gamma\in\beta^{\downarrow}}(\gamma^{\uparrow} \setminus \{\beta\})$, (10) defines all the weight functions $w_{1,j}$, $j \in [0, r]$. A slight modification of (10) yields the other weight functions $w_{i,j}$: For a partition $\gamma = (\gamma_1, \gamma_2, \ldots)$ and $i \geq 1$ we define $\gamma^i := (\gamma_1 - i, \gamma_2, \ldots)$, if $\gamma_1 - i \geq \gamma_2$, otherwise γ^i is undefined. Note that $\beta^i = \beta_{i,0}$. Let $i \geq 1$. If for the partitions in (10) $\beta^i, \gamma^i, \varepsilon^i$ are defined, then $w_{\beta^i\cap\gamma^i}(\gamma^i) := w_{\beta^i}(\beta^i \cup \gamma^i) \cdot \diamond_{\beta^i,\gamma^i}$ and $w_{\varepsilon^i}(\beta^i) := 1$ define the remaining weight functions. Furthermore, $w_{\beta^i}(\beta^i \cup \gamma^i) = w_{\beta^1}(\beta^1 \cup \gamma^1)$ and $\diamond_{\beta^i,\gamma^i} = \diamond_{\beta,\gamma}$, if $\gamma < \beta$ or ($\gamma > \beta$ and $\beta_1 = \gamma_1$). If $\beta_1 < \gamma_1$ and ξ denotes the axial distance of β and γ, then $\diamond_{\beta,\gamma} = q^2_\xi = 1 - \xi^{-2}$, whereas $\diamond_{\beta^i,\gamma^i} = q^2_{\xi-i}$. All this is illustrated in Figure 8 for $\beta = (5, 2, 1)$ and $w_\beta = (A, B, C, D)$.

To get, e.g., $w_{1,2}$ and $w_{2,1}$, follow the uparrows starting at $\boxed{52}$ and $\boxed{411}$: $w_{1,2} = (521 \mapsto 1, 53 \mapsto C \cdot q^2_3, 62 \mapsto D \cdot q^2_7)$ and $w_{2,1} = (41^3 \mapsto A, 421 \mapsto 1, 51^2 \mapsto q^2_4)$.

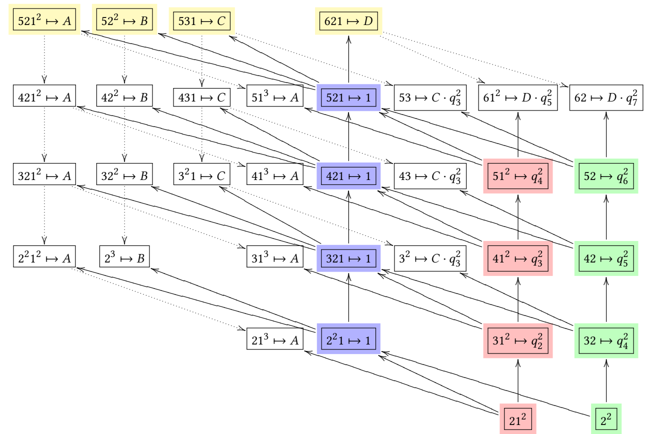

Figure 8: Structure of $w_{i,0}$, $w_{i,1}$, $w_{i,2}$

The next result shows that the above weight propagation is compatible with our recursion scheme.

Theorem 4.5 (weighted form of Thm. 4.4). *Let $\gamma \in \beta^{\uparrow\downarrow}$, $\gamma \neq \beta$, $\alpha := \beta \cup \gamma$, $\delta := \beta \cap \gamma$. Then*

$$\boxed{\mathbf{P}^{\alpha}_{\gamma}\mathbf{FFT}^{w}_{\beta}[X, w_\beta]^{\alpha} = \mathbf{FFT}^{w}_{\delta}[X^{\beta}_{\delta}, w_\delta]^{\gamma}}, \tag{11}$$

and with $\curlywedge^{\alpha}_{\beta} := \sum_{\varepsilon\in\beta^{\downarrow}} \curlywedge^{\alpha}_{\beta,\varepsilon} \cdot \mathbf{FFT}^{w}_{\varepsilon}[X^{\beta}_{\varepsilon}, w_\varepsilon]^{\beta}$

$$\boxed{\mathbf{P}^{\alpha}_{\beta}\mathbf{FFT}^{w}_{\beta}[X, w_\beta]^{\alpha} = \left(\curlywedge^{\alpha}_{\beta} + x_n\right) \cdot w_\beta(\alpha)}. \tag{12}$$

Proof. (11) follows from

$$\begin{aligned}
\mathbf{P}^{\alpha}_{\gamma}\mathbf{FFT}^{w}_{\beta}[X, w_\beta]^{\alpha} &= w_\beta(\alpha) \cdot \mathbf{P}^{\alpha}_{\gamma} s^{\alpha}_{\beta}(X) \\
&= w_\beta(\alpha) \cdot \diamond_{\beta,\gamma} \cdot s^{\gamma}_{\delta}(X^{\beta}_{\delta}) \text{ (by Thm. 4.4)} \\
&= w_\delta(\gamma) \cdot s^{\gamma}_{\delta}(X^{\beta}_{\delta}) \qquad \text{(by Def. (10))} \\
&= \mathbf{FFT}^{w}_{\delta}[X^{\beta}_{\delta}, w_\delta]^{\gamma}.
\end{aligned}$$

(12) is shown in a similar way. □

We describe the analogons of (11) and (12) when $\beta = \beta_{0,0}$ is replaced by $\beta_{i,0}$. For $i \in [1, n - k - \mu_1]$ let $\gamma \in \beta^{\uparrow\downarrow}_{i,0}$, $\gamma \neq \beta_{i,0}$. Then $\beta_{i,0} \cap \gamma = \beta_{i+1,j}$, for a unique $j \in [0, r]$. Let $\alpha := \beta_{i,0} \cup \gamma$. Then

$$\boxed{\mathbf{P}^{\alpha}_{\gamma}\mathbf{FFT}^{w}_{i,0}[X_{i,0}, w_{i,0}]^{\alpha} = \mathbf{FFT}^{w}_{i+1,j}[X_{i+1,j}, w_{i+1,j}]^{\gamma}}. \tag{13}$$

Let $\curlywedge^{\alpha}_{i} := \sum_{j=0}^{r} \curlywedge^{\alpha}_{\beta_{i,0},\beta_{i+1,j}} \cdot \mathbf{FFT}^{w}_{i+1,j}[X_{i+1,j}, w_{i+1,j}]^{\beta_{i,0}}$ and $x'_{n-i} := \mathbf{P}^{\beta_{i,0}} x_{n-i}$. Then

$$\boxed{\mathbf{P}^{\alpha}_{\beta_{i,0}}\mathbf{FFT}^{w}_{i,0}[X_{i,0}, w_{i,0}]^{\alpha} = (\curlywedge^{\alpha}_{i} + x'_{n-i}) \cdot w_{i,0}(\alpha)}. \tag{14}$$

4.5 Weighted local FFTs

Now we can state our local FFT-algorithm at a meta-level. (The unspecified instructions in Line 5 and Line 9 will become clear in a moment.)

$\mathbf{FFT}^{\boldsymbol{w}}_{\boldsymbol{\beta}}$ $(\beta = (n-k,\mu) \vdash n-1, \mu \vdash k-1)$

1. **Input**: $X \in \mathbb{C}^{\beta \times n}$, $\boldsymbol{w}_\beta : \beta^\uparrow \to (0;1]$
2. **Preprocessing:** Compute all $\boldsymbol{w}_{i,j}$
3. **Initialization:**
4. $f := n - k - \mu_1$
5. **for** $j \in [1,r]$ compute $\mathbf{FFT}^{\boldsymbol{w}}_{f+1,j}[X_{f+1,j}, \boldsymbol{w}_{f+1,j}]$
6. Compute $\mathbf{FFT}^{\boldsymbol{w}}_{f,0}[X_{f,0}, \boldsymbol{w}_{f,0}]$ blockwise via (13), (14), and the outputs of Line 5
7. **Recursion:**
8. **for** $i = f$ **downto** 1 **do**
9. Compute $\mathbf{FFT}^{\boldsymbol{w}}_{i,j}[X_{i,j}, \boldsymbol{w}_{i,j}]$, for all $j \in [1,r]$
10. Compute $\mathbf{FFT}^{\boldsymbol{w}}_{i-1,0}[X_{i-1,0}, \boldsymbol{w}_{i-1,0}]$ blockwise via (13), (14), $\mathbf{FFT}^{\boldsymbol{w}}_{i,0}[X_{i,0}, \boldsymbol{w}_{i,0}]$, and the outputs of Line 9
11. **end for**
12. **Output:**
13. $\mathbf{FFT}^{\boldsymbol{w}}_{\beta}[X, \boldsymbol{w}_\beta]^\alpha = \mathbf{FFT}^{\boldsymbol{w}}_{0,0}[X_{0,0}, \boldsymbol{w}_{0,0}]^\alpha$, $\forall \alpha \in \beta^\uparrow$

THEOREM 4.6. For $n > k$ the following statement holds.

> $\boldsymbol{AC}^{\boldsymbol{n}}_{\boldsymbol{k}}$: For all partitions β satisfying $\beta = (n-k,\mu)$, $\mu \vdash k-1$, the arithmetic cost of running $\mathbf{FFT}^{\boldsymbol{w}}_{\boldsymbol{\beta}}$ is of order $[S_n : S_{n-k}]$.

The proof needs some preparations. For $n \geq 2$ and $I \subseteq [0, n-1]$ let $(n)_I := \prod_{i \in I}(n-i)$. Then $(n)_{[0,n-1]} = n!$ and $(n)_{[0,k-1]} = (n)_k = [S_n : S_{n-k}]$ is a falling factorial.

LEMMA 4.7. Let $n \geq 2k$ and $\beta = (n-k,\mu)$, $\mu \vdash k-1$. Define $I(\mu) := [1, k-1+\mu_1] \setminus \bigcup_{j \leq \mu_1}\{k-1+j-\mu'_j\}$, μ' the conjugate partition of μ. Then $|I(\mu)| = k-1$ and

$$d_\beta = \frac{d_\mu}{(k-1)!} \cdot (n)_{I(\mu)} \leq \frac{d_\mu}{(k-1)!} \cdot (n-1)_{k-1}.$$

PROOF. The equality follows from the hook-length formula by a straightforward computation. The inequality results from the fact that $I(\mu)$ is a $(k-1)$-subset of $[1, k-1+\mu_1]$. □

Furthermore, we need the identity (see, e.g., [6], (2.50))

$$\sum_{m=0}^{n-1} (m)_{k-1} = \frac{(n)_k}{k}. \tag{15}$$

PROOF. (of Theorem 4.6) We prove the $\boldsymbol{AC}^{\boldsymbol{n}}_{\boldsymbol{k}}$-statement by induction on k. **Start**: $k=1$. This is true by [5].

Step: $k-1 \to k$. Note that $\mathbf{FFT}^{\boldsymbol{w}}_{i,j}[X_{i,j}, \boldsymbol{w}_{i,j}]$ is an instance of $\boldsymbol{AC}^{\boldsymbol{n-i}}_{\boldsymbol{k-1}}$, if $j \in [1,r]$, while for $j=0$ it is an instance of $\boldsymbol{AC}^{\boldsymbol{n-i}}_{\boldsymbol{k}}$. (This explains the vagueness of Line 5 and Line 9: by induction hypothesis we already know how to compute these local FFTs in an order optimal way.) We will split the analysis of the arithmetic cost into three parts: cost of preprocessing (Line 2), cost of running $\mathbf{FFT}^{\boldsymbol{w}}_{i,j}$ for all relevant $i \geq 1$ and all $j \in [1,r]$, and cost of all leaf-rake computations incurred when computing $\mathbf{FFT}^{\boldsymbol{w}}_{i,0}$. As we use $\mathbf{FFT}^{\boldsymbol{w}}_{\boldsymbol{\beta}}$, no diamond computations incur. Thus cost($\mathbf{FFT}^{\boldsymbol{w}}_{\boldsymbol{\beta}}$) $\leq$ cost(Line 2) + cost($\mathbf{FFT}^{\boldsymbol{w}}_{i \geq 1, j \geq 1}$) + cost($\curlywedge$).

$\boxed{\text{cost(Line 2)}}$ We assume that the input independent entities q^2_ξ are tabulated. Fig. 8 indicates that Line 2 affords $\leq |\{\gamma \in \beta^{\uparrow\downarrow} | \gamma > \beta\}|$ scalar multiplications. Let $|\mu^\downarrow| = r$. Then $r \leq \lfloor(\sqrt{8k-7}-1)/2\rfloor$, see A003056 in [18]. Now the number of those γ is upper bounded by $r + \binom{r+1}{2}$, which is smaller than $k + \sqrt{2k}$.

$\boxed{\text{cost}(\mathbf{FFT}^{\boldsymbol{w}}_{i \geq 1, j \geq 1})}$ At level i we have r calls of instances of $\boldsymbol{AC}^{\boldsymbol{n-i}}_{\boldsymbol{k-1}}$. By the induction hypothesis, these subroutines cause cost of order $r\sum_{i=1}^{n-k-\mu_1+1}(n-i)_{k-1}$, which, by (15), is smaller than $\frac{r}{k} \cdot (n)_k < (n)_k$.

$\boxed{\text{cost}(\curlywedge)}$ In level $i \geq 1$, we have to perform $|\beta^\uparrow_{i-1,0}| \leq 1 + |\mu^\uparrow|$ leaf-rake computations. By Equation (12), each such computation is a linear combination of $1 + |\beta^\downarrow_{i-1,0}| \leq 2 + |\mu^\downarrow|$ vectors in $\mathbb{C}^{\beta_{i-1,0}}$. As $|\mu^\uparrow| = 1 + |\mu^\downarrow|$ we get the upper bound

$$\text{cost}(\curlywedge) \leq 2 \cdot (2 + |\mu^\downarrow|)^2 \sum_{i=0}^{n-k-\mu_1-1} d_{\beta_{i,0}} \leq \frac{2 \cdot (2+|\mu^\downarrow|)^2 \cdot d_\mu}{(k-1)!} \cdot (n)_k,$$

by Lemma 4.7. This proves Theorem 4.6. □

5 CONCLUDING REMARKS

In this paper we designed order-optimal FFTs for computing spectral images of S_{n-k}-invariant functions on the symmetric group S_n. In the context of his PhD, the second author has implemented a variant of our algorithm for $k \leq 3$. For implementation details and run-time tables we refer to [7].

6 ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions.

REFERENCES

[1] U. Baum and M. Clausen. Some lower and upper complexity bounds for generalized Fourier transforms and their inverses. *SIAM J. Comput.*, 20(3):451–459, 1991.

[2] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1997.

[3] M. Clausen. Fast generalized Fourier transforms. *Theor. Comput. Sci.*, 67(1):55–63, 1989.

[4] M. Clausen and U. Baum. Fast Fourier transforms for symmetric groups: theory and implementations. *Math. Comp.*, 61(204):833–847, 1993.

[5] M. Clausen and R. Kakarala. Computing Fourier transforms and convolutions of S_{n-1}-invariant signals on S_n in time linear in n. *Appl. Math. Lett.*, 23(2):183–187, 2010.

[6] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science, second edition.* Addison-Wesley, 1994.

[7] P. C. Hühne. *Beiträge zum Entwurf größenoptimaler schneller Fouriertransformationen auf gewissen homogenen Räumen symmetrischer Gruppen.* PhD thesis, Universität Bonn, 2016.

[8] G. James and A. Kerber. *The Representation Theory of the Symmetric Group.* Encyclopedia of Mathematics and its Applications, 16. Addison-Wesley, 1981.

[9] R. Kondor. A Fourier space algorithm for solving quadratic assignment problems. In *SODA 2010*, pages 1017–1028, 2010.

[10] R. Kondor and M. S. Barbosa. Ranking with kernels in Fourier space. In *COLT 2010*, pages 451–463, 2010.

[11] R. Kondor and W. Dempsey. Multiresolution analysis on the symmetric group. In *NIPS 2012*, pages 1646–1654, 2012.

[12] R. Kondor, A. Howard, and T. Jebara. Multi-object tracking with representations of the symmetric group. In *AISTATS 2007*, pages 211–218, 2007.

[13] R. Kondor, N. Shervashidze, and K. M. Borgwardt. The graphlet spectrum. In *ICML 2009*, pages 529–536, 2009.

[14] D. K. Maslen. The efficient computation of Fourier transforms on the symmetric group. *Math. Comput.*, 67(223):1121–1147, 1998.

[15] D. K. Maslen and D. N. Rockmore. Generalized FFT's – A survey of some recent results. In *Groups and Computation, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, June 7-10, 1995*, pages 183–238, 1995.

[16] D. Rockmore, P. Kostelec, W. Hordijk, and P. F. Stadler. Fast Fourier transforms for fitness landscapes. *Applied and Computational Harmonic Analysis*, 11(1):57–76, 2002.

[17] J.-P. Serre. *Linear Representations of Finite Groups*, volume 42 of *Graduate Texts in Mathematics.* Springer, 1977.

[18] N. Sloane. The on-line encyclopedia of integer sequences. https://oeis.org.