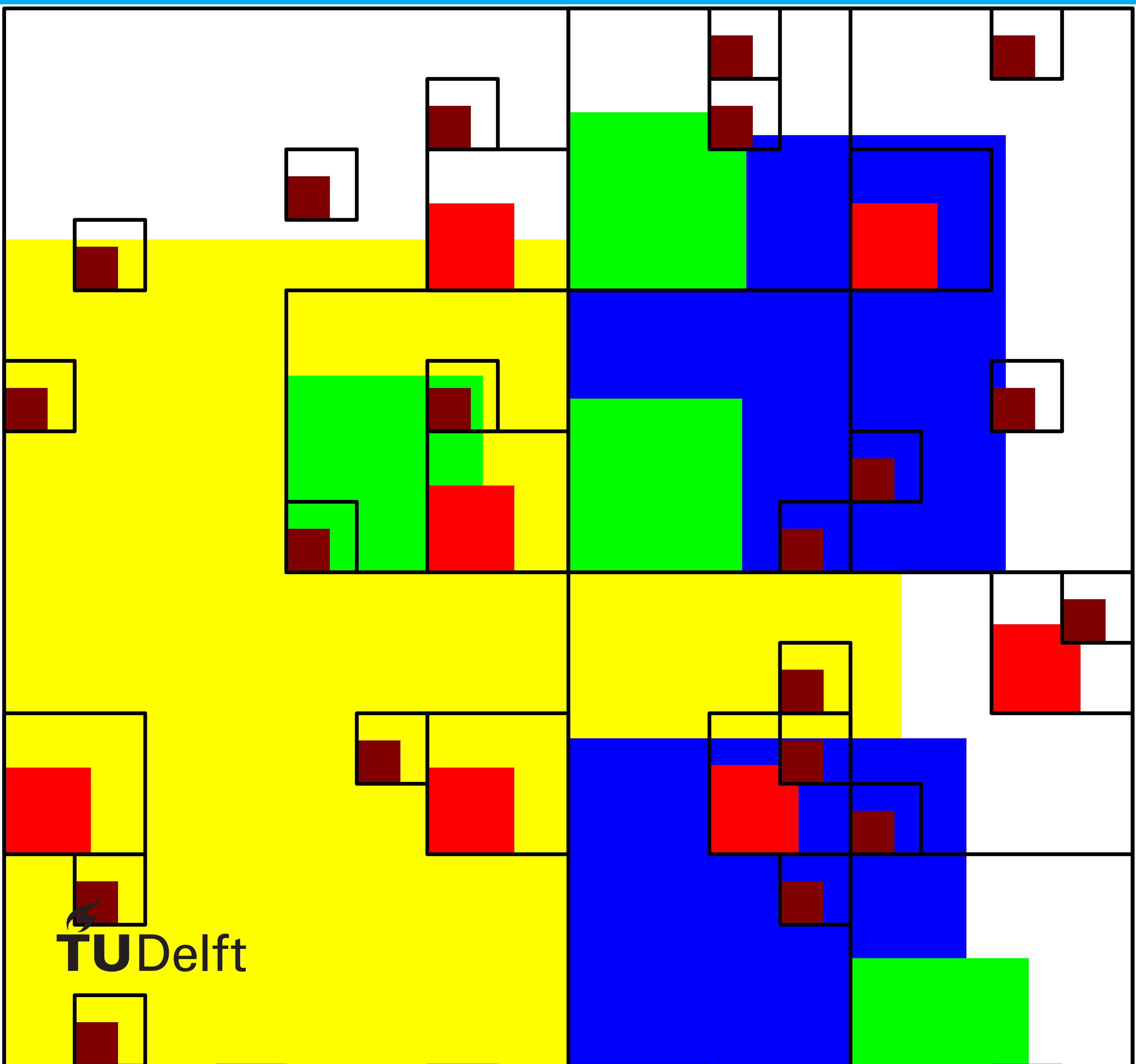
A constructive algorithm to prove the equivalence of the Carleson and sparse condition

E. A. Honig



A constructive algorithm to prove the equivalence of the Carleson and sparse condition

by

E. A. Honig

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended publicly on Friday November 1, 2024 at 13:45.

Student number:	5357306	
Project duration:	September 2, 2024 – November 1, 2024	
Thesis committee:	Dr. ir. E. Lorist,	TU Delft, supervisor
	Prof. dr. D.C. Gijswijt,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Cover picture by Lerner and Nazarov [20, Figure 11]

TU Delft

Laymen’s summary

A *set* is a group of different items. These items can be any object such as apples, numbers, or squares and are called *elements*. We write $Q = \{1, 2, 3\}$, where Q is the set, and 1, 2 and 3 are the elements in Q. We can also have a set that contains other sets. This is called a *collection* of sets. We write $\mathscr{F} = \{P, Q, R\}$, where $\mathscr{F}$ is the collection, and P, Q and R are the sets in the collection. Items can appear in more than one set. If P and Q are two sets, and 1 is both an element of P and of Q, we say that 1 is in the *overlap* of P and Q. Another way to explain overlap, is by noting that a set can also contain all points in the area of a rectangle. We can for example have a collection $\mathscr{F}$ that contains the sets P, Q and R that describe the squares in Figure 1. The overlap between Q_1 and Q_2 is then the area that is both in Q_1 and Q_2.

In some fields of mathematics, it can be useful to describe the amount of overlap there is between all the sets in a collection $\mathscr{F}$. In other words, we want to describe how many elements appear in more than one set that is an element of $\mathscr{F}$. Sparse and Carleson constants are two numbers that can be used to describe the overlap within collections of sets. The closer these constants are to 1, the closer the sets in $\mathscr{F}$ are to having no overlap at all. In this thesis, I will prove that the the two notions are actually equivalent. Although a proof of this equivalence has already been given by other researchers, I will use a lot less difficult theory in my proof than they have. I will also describe a way that the sparse and Carleson constants can be approximated if they are unknown.

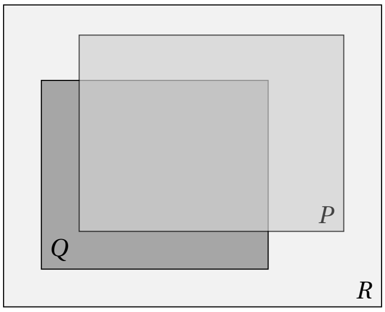


Figure 1: The collection $\mathscr{F}$ of rectangles P, Q and R.

Abstract

Let (S, Σ, μ) be a divisible measure space. Let $\mathscr{F}$ be a collection of subsets of S that are in Σ. For some applications it can be useful to describe the overlap between the sets in $\mathscr{F}$. The sparse and Carleson constant both describe this overlap in a different way. The closer both of these constants are to 1, the closer the sets in $\mathscr{F}$ are to being pairwise disjoint. It has been shown that the sparse and Carleson condition are actually equivalent: we always have that $\mathscr{F}$ is Λ-Carleson if and only if $\mathscr{F}$ is Λ^{-1}-sparse. Proving that a Λ^{-1}-sparse collection is Λ-Carleson is quite simple, but proving that every Λ-Carleson collection also is Λ^{-1}-sparse turns out to be much harder. Previous proofs of the fact that Λ-Carleson are Λ^{-1}-sparse, such as the one by Hänninen [14] and Rey [25], have all relied on difficult theory. There is also no known method to exactly find the sets $\{E_Q\}_{Q\in\mathscr{F}}$ that we need to satisfy the sparse condition. Rey is able to approximate these sets, but his algorithm has a logarithmic loss that can only be removed when imposing geometric restrictions.

In this paper I will give a proof of the equivalence of the sparse and Carleson condition for any finite collection $\mathscr{F}$ that relies only on basic set and optimisation theory. This proof can be extended to infinite collections $\mathscr{F}$ with only a minimal restriction. Asides from proving the equivalence, I also describe an algorithm that can find the sets $\{E_Q\}_{Q\in\mathscr{F}}$ which we need to satisfy the sparse condition if $\mathscr{F}$ is finite and Carleson with respect to a divisible measure μ. Finally, I will describe an algorithm that we can use to find the Carleson constant of a finite collection $\mathscr{F}$ if it is unknown.

Contents

1
Introduction

The concepts of sparse and Carleson collections have been widely applied in harmonic analysis. They know many applications [23], for example due to Lerner [18] [19] and Hytönen [15]. The usual definitions of sparse and Carleson collections are as follows.

Definition 1. *Let (S, Σ, μ) be a measure space. Let $\mathscr{F}$ be a collection of subsets of S that are in Σ such that $\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)$ is finite. Let $0 < \eta < 1$. $\mathscr{F}$ is called η-sparse with respect to measure μ if for every $Q \in \mathscr{F}$ there exists a subset $E_Q \subseteq Q$ such that $\mu\left(E_Q\right) \geq \eta\mu(Q)$ and such that the sets $\{E_Q\}_{Q\in\mathscr{F}}$ are pairwise disjoint. The largest possible η for which these sets exist is called the sparse constant of $\mathscr{F}$.*

Definition 2. *Let (S, Σ, μ) be a measure space. Let $p\mathscr{F}$ be a collection of subsets of S that are in Σ such that $\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)$ is finite. $\mathscr{F}$ is called Λ-Carleson with respect to measure μ if for every subcollection $\mathscr{A} \subseteq \mathscr{F}$ we have:*

$$\sum_{Q\in\mathscr{A}} \mu(Q) \leq \Lambda\mu\left(\bigcup_{Q\in\mathscr{A}} Q\right).$$

The smallest possible Λ for which this inequality holds is called the Carleson constant of $\mathscr{F}$.

Note that we call $\mathscr{F}$ a collection. We use this term to denote an unordered sequence of sets that allows for repeated elements. Also note that from Definition 2 it follows that a Carleson collection $\mathscr{F}$ must be countable, because otherwise we cannot sum over each $Q \in \mathscr{F}$.

Intuitively we can view both the sparse and Carleson constants as quantifiers on how much a collection overlaps. The closer the sparse and Carleson constants are to 1, the closer the sets in $\mathscr{F}$ are to being pairwise disjoint. It is easily shown that an η-sparse collection is η^{-1}-Carleson, simply because

$$\sum_{Q\in\mathscr{F}} \mu(Q) \leq \frac{1}{\eta} \sum_{Q\in\mathscr{F}} \mu\left(E_Q\right) \leq \frac{1}{\eta}\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right), \tag{1.1}$$

where the last inequality holds because all the sets in $\{E_Q\}_{Q\in\mathscr{F}}$ are pairwise disjoint. On the other hand, showing that a Λ-Carleson collection is Λ^{-1}-sparse proves to be more difficult.

It must be noted that in some cases, Carleson collections may not be sparse. Problems arise when working with atomic measures, as these cannot be divided [14]. Take for example the case of a Dirac measure δ_x at the point x. If we have $Q_1 = Q_2 = \{x\}$, then $\{Q_1, Q_2\}$ is 2-Carleson, but cannot be sparse as we will either have $\mu\left(E_{Q_1}\right) = 0$ or $\mu\left(E_{Q_2}\right) = 0$ as these sets should be pairwise disjoint. Rey [25, Definition 1] resolves this problem by proposing an alternative definition for sparseness. This definition is as follows.

Definition 3. *Let (S, Σ, μ) be a measure space. Let $\mathscr{F}$ be a collection of subsets of S that are in Σ such that $\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)$ is finite. $\mathscr{F}$ is called η-sparse with respect to measure μ if for every $Q \in \mathscr{F}$ there exist non-negative functions $\phi_Q \geq 0$ such that:*

(S1) $\int_Q \phi_Q \geq \eta\mu(Q)$

(S2) $\sum_{Q\in\mathscr{F}} \phi_Q \leq 1$.

The largest possible η for which these inequalities hold is called the sparse constant of $\mathscr{F}$.

Note that this definition is only a slight generalisation of Definition 1. If we take $\phi_Q = \mathbb{1}_{E_Q}$, the definitions are actually equivalent. We also can use a convexity argument like the one given by Dor [10, Lemma 2.3] to show that this definition is equivalent with Definition 1 in the case that μ has no point masses.

It has been shown that in general, a Λ-Carleson collection of sets is Λ^{-1}-sparse. In the case of dyadic cubes this was proven both by Verbitsky [28, Corollary 2] and Lerner and Nazarov [20, Lemma 6.3]. Verbitsky gave a dual reformulation of the Carleson condition as a certain estimate [28, Theorem 4], and combined this with the characterisation of such general estimates in terms of the existence of pairwise disjoint sets by Dor [10, Proposition 2.2], which is based on functional and convex analysis. Lerner and Nazarov avoided relying on functional and convex analysis by giving a constructive proof using the strong nestedness property of dyadic intervals. I will go a bit more into depth on why dyadic cubes are so special in Section 3.3.1.

Sadly, the simpler approach of Lerner and Nazarov cannot be generalised. Carleson's counterexample [5] [27, §7] shows that it already fails in the case of dyadic rectangles[1]. Proofs for the fact that a more general Λ-Carleson collection of general sets is Λ^{-1}-sparse have all relied on difficult theory. Hänninen [14, Theorem 1.3] generalised the approach of Verbitsky and Dor to prove the existence of the sets $\{E_Q\}_{Q\in\mathscr{F}}$ that satisfy the Λ^{-1}-sparse condition, but he relies on the choice axiom and requires advanced theory to achieve this, such as the Hahn-Banach's separation theorem. Furthermore, he does not specify how the sets $\{E_Q\}_{Q\in\mathscr{F}}$ can be found or what they look like. Barron later gives a more geometric proof in [3], but also is unclear about how the sets $\{E_Q\}_{Q\in\mathscr{F}}$ can be found.

Rey [25, Algorithm 1] does describe an algorithm that is able to construct the sets $\{E_Q\}_{Q\in\mathscr{F}}$ for any Carleson collection $\mathscr{F}$. However, this algorithm has a logarithmic loss. The result is better when some geometric structure is imposed [25, Algorithm 2], but even then the algorithm yields sets that are only optimal up to an absolute constant.

In this bachelor's thesis, I will prove the equivalence of the sparse and Carleson condition for a collection $\mathscr{F}$ of general sets with minimal restrictions, relying only on basic optimisation and set theory. I will also describe an algorithm that can construct sets $\{E_Q\}_{Q\in\mathscr{F}}$ for a finite collection $\mathscr{F}$ of general sets that are Λ-Carleson with respect to a divisible measure μ without any loss.

In order to give the proof and construct the algorithm, I will transform a Λ-Carleson collection $\mathscr{F}$ into a graph, and then use a MAX FLOW algorithm to find the maximum flow in this graph. MAX FLOW algorithms are widely used in optimisation theory to solve a variety of problems. They are able to find the maximum amount of flow that is able to pass from a source to a destination (usually called a sink) through a network in which each route has a maximum capacity. The algorithms are widely used in transportation problems [26], but also know many other applications [12, Chapter 11]. A general introduction to MAX FLOW algorithms can be found in [6] and [12, Chapter 10]. Each algorithm comes with different advantages and complexities.

Although from an analytic point of view the complexity of the algorithm is not very relevant, it might be helpful to know how fast the $\{E_Q\}_{Q\in\mathscr{F}}$ can be found if applications require us to find these sets. It will also become useful in Chapter 5, where I will construct a similar algorithm as in Chapter 3 to approximate the Carleson constant. It will be interesting to see whether this algorithm is faster than simply checking the size of the sum of each subcollection $\mathscr{A} \subseteq \mathscr{F}$ and see if they comply with Definition 2. Orlin's algorithm [22], and KRT's algorithm (King, Rao and Tarjan) [16] are some of the fastest MAX FLOW algorithms available. They are able to find the maximum flow in $O(|V||E|)$ time, where $|V|$ is the amount of vertices in the graph, and $|E|$ the amount of edges. However, since these algorithms can be complicated to implement, simpler algorithms with a higher complexity, such as Dinitz's algorithm $\left(O\left(|V|^2|E|\right)\right)$ [9] and the Edmonds-Karp algorithm $\left(O\left(|V||E|^2\right)\right)$ [11] are still widely used as well.

I will use the maximum flow in the graph based on a Λ-Carleson collection $\mathscr{F}$ to prove that it is also Λ^{-1}-sparse. To do this, I will use the complementary slackness of Max-flow and Min-cut problems. This complementary slackness is a consequence of the Max-flow Min-cut Theorem [7] [12, §10.3], which states that when we have used a MAX FLOW algorithm to find the maximum flow in a graph, we can divide the vertices in two groups such that no more flow can go through the edges that connect vertices in the first group to those in the second group.

In Chapter 4 I will try to extend the proof from Chapter 2 so that it also works for infinite Λ-Carleson collections. To make this generalisation, we need generalisations of graph theory for graphs with an infinite amount of vertices. This is because we will see that an infinite collection $\mathscr{F}$ will transform into a graph with

[1] This will also be explained further in Section 3.3.1.

an infinite amount of vertices and edges. We will see that this amount could even be uncountable, despite the fact that $\mathscr{F}$ is countable by definition. Fortunately, research is available on infinite graphs. They have actually been a part of graph theory from the very beginning. König [17] already talked about infinite graphs in one of the first books on graph theory. Folkman and Fulkerson [13, Theorem 3] discussed flows in infinite graphs, and generalised the Max-flow Min-cut theorem for locally finite graphs. Locally finite graphs are graphs where each vertex only has a finite amount of neighbours. Aharoni et al. [2, Conjecture 1.2] later propose a Conjecture that states that in any infinite graph there exist an orthogonal pair of a flow and a cut. They manage to prove this for countable graphs, which are graphs with a countable amount of vertices and edges. This proof was later simplified by Lochbihler [21, Theorem 1]. I will use this generalisation of the Max-flow Min-cut theorem to generalise the proof in Chapter 2 to prove the equivalence of the Carleson and sparse condition for a countable collection $\mathscr{F}$ that transforms into a graph with a countable amount of vertices and edges. I will show that this is the case whenever the sets $Q \in \mathscr{F}$ overlap only countably many times. Sadly I will not be able to actually construct the sets $\{E_Q\}_{Q\in\mathscr{F}}$ because no algorithm is known that can find the values of the maximum flow in an infinite graph in finite time, and the values of this flow are crucial in the algorithm that we used for finite collections in Chapter 3.

2

Equivalence of the Carleson and sparse condition for finite collections of sets

In this chapter I will prove that the Carleson and sparse condition are equivalent for a finite collection of sets $\mathscr{F}$. As explained in the Introduction, it is easily shown that η-sparse collections are η^{-1}-Carleson. I will thus only need to prove the converse, which is expressed in the following theorem.

Theorem 2.0.1. *Let (S, Σ, μ) be a measure space. Let $\mathscr{F}$ be a finite collection of subsets of S that are in Σ such that $\mu\left(\bigcup_{Q \in \mathscr{F}} Q\right)$ is finite. If $\mathscr{F}$ is Λ-Carleson, then $\mathscr{F}$ is Λ^{-1}-sparse.*

Let $\mathscr{F}$ be as in Theorem 2.0.1. To prove the theorem, we will transform $\mathscr{F}$ into a graph. We will then find the maximum flow in this graph and introduce a function ϕ_Q that is based on this flow. We will show that this function complies with Rey's definition of sparseness (Definition 3) as stated in the Introduction, and thus that $\mathscr{F}$ is indeed Λ^{-1}-sparse.

2.1. Representing the collection as a graph

As mentioned above, we want to transform a Λ-collection $\mathscr{F}$ into a weighted and directed graph $G(V, E, c)$. Here, c is a capacity function $c : E \mapsto \mathbb{R}_{\geq 0}$. Note that the method in this section will also work for a collection $\mathscr{F}$ with a countably infinite amount of elements. Most definitions and lemmas in this sections will thus not require for $\mathscr{F}$ to be finite. Although we will not need this in this chapter, this will become useful in Chapter 4 where we proof the equivalence of the Carleson and sparse conditions for a collection with countably many elements.

We start by defining the following property of each subcollection $\mathscr{A} \subseteq \mathscr{F}$.

Definition 4. *Let (S, Σ, μ) be a measure space and let $\mathscr{F}$ be a collection of sets in Σ. For each subcollection $\mathscr{A} \subseteq \mathscr{F}$, we define the area $A_{\mathscr{A}}$ of the subcollection as*

$$A_{\mathscr{A}} := \left(\bigcap_{Q \in \mathscr{A}} Q \right) \cap \left(\bigcap_{Q \in \mathscr{F} \setminus \mathscr{A}} Q^c \right).$$

The area of $\mathscr{A}$ represents the space that is in all the sets in $\mathscr{A}$, but not in any of the sets that are not in $\mathscr{A}$. Two important properties of the area of the subcollections are described in the following lemma.

Lemma 2.1.1. *Let (S, Σ, μ) be a measure space and let $\mathscr{F}$ be a collection of sets that are in Σ. Define $A_{\mathscr{A}}$ for each subcollection $\mathscr{A} \subseteq \mathscr{F}$ as in Definition 4. We then have that*

1. *The sets $\{A_{\mathscr{A}}\}_{\mathscr{A} \subseteq \mathscr{F}}$ are pairwise disjoint. In other words, we have $A_{\mathscr{A}_1} \cap A_{\mathscr{A}_2} = \emptyset$ for any $\mathscr{A}_1, \mathscr{A}_2 \in \mathscr{F}$ such that $\mathscr{A}_1 \neq \mathscr{A}_2$.*

2. *For each $\mathscr{A} \subseteq \mathscr{F}$, we have that $\bigcup_{Q \in \mathscr{A}} \bigcup_{\mathscr{A}' \ni Q} A_{\mathscr{A}'} = \bigcup_{Q \in \mathscr{A}} Q$.*

Proof. **(1)** We give a proof by contradiction. Suppose there is an element x such that $x \in A_{\mathscr{A}_1}$ and $x \in A_{\mathscr{A}_2}$ with $\mathscr{A}_1, \mathscr{A}_2 \subseteq \mathscr{F}$ and $\mathscr{A}_1 \neq \mathscr{A}_2$. We then have

$$x \in \left(\bigcap_{Q \in \mathscr{A}_1} Q \right) \cap \left(\bigcap_{Q \in \mathscr{F} \setminus \mathscr{A}_1} Q^c \right) \text{ and } x \in \left(\bigcap_{Q \in \mathscr{A}_2} Q \right) \cap \left(\bigcap_{Q \in \mathscr{F} \setminus \mathscr{A}_2} Q^c \right). \tag{2.1}$$

Without loss of generality, we can assume there is at least one set $Q_2 \in \mathscr{A}_2$ such that $Q_2 \notin \mathscr{A}_1$. The first part of (2.1) then tells us that $x \in Q_2^c$, whilst the second part tells us that $x \in Q_2$. This is a contradiction, so we must have that $A_{\mathscr{A}_1}$ and $A_{\mathscr{A}_2}$ are disjoint.

(2) For any $\mathscr{A} \subseteq \mathscr{F}$, let $x \in \bigcup_{Q \in \mathscr{A}} \bigcup_{\mathscr{A}' \ni Q} A_{\mathscr{A}'}$ be arbitrary. Then $x \in A_{\mathscr{A}'}$ for some $\mathscr{A}' \subseteq \mathscr{F}$ which contains a set $Q' \in \mathscr{A}'$ for which we also have $Q' \in \mathscr{A}$. Definition 4 tells us that we must then have $A_{\mathscr{A}'} \subseteq Q'$. Because we have $Q' \in \mathscr{A}$ we can conclude that $x \in A_{\mathscr{A}'} \subseteq Q' \subseteq \bigcup_{Q \in \mathscr{A}} Q$. We thus have $\bigcup_{Q \in \mathscr{A}} \bigcup_{\mathscr{A}' \ni Q} A_{\mathscr{A}'} \subseteq \bigcup_{Q \in \mathscr{A}} Q$. For the converse we let $y \in \bigcup_{Q \in \mathscr{A}} Q$ be arbitrary. There then exists a $Q' \in \mathscr{A}$ such that $y \in Q'$. Let $\mathscr{A}_y$ denote the collection containing all the sets $Q \in \mathscr{F}$ such that $y \in Q$. By Definition 4 we have $y \in A_{\mathscr{A}_y}$. We also have $Q' \in \mathscr{A}_y$, because $y \in Q'$. Recall that $Q' \in \mathscr{A}$ by assumption. We therefore have $y \in A_{\mathscr{A}_y} \subseteq \bigcup_{Q \in \mathscr{A}} \bigcup_{\mathscr{A}' \ni Q} A_{\mathscr{A}'}$, which concludes the proof. □

We are now ready to start constructing our graph. Starting with our vertex set V. We see that we add three kinds of vertices to our graph.

(V1) We add a vertex for the source ⊕ and sink ⊖.

(V2) For any $\mathscr{A} \subseteq \mathscr{F}$, such that $\mu(A_{\mathscr{A}}) > 0$, we add a vertex denoted by $v_{\mathscr{A}}$.

(V3) For each set $Q \in \mathscr{F}$ we add a vertex denoted by v_Q.

Note that for a $Q \in \mathscr{F}$ we will have separate vertices for the subcollection $\mathscr{A} = \{Q\}$ and for the set Q if $\mu\left(A_{\{Q\}}\right) > 0$. Asides from the vertices, we will also add the following edges to our graph. We will denote edges by $(\mathscr{A}, Q)$ instead of $(v_{\mathscr{A}}, v_Q)$ for readability.

(E1) For any $\mathscr{A} \subseteq \mathscr{F}$ such that $\mu(A_{\mathscr{A}}) > 0$, we add an edge $(\oplus, \mathscr{A})$ with capacity $c(\oplus, \mathscr{A}) = \mu(A_{\mathscr{A}})$.

(E2) For any $\mathscr{A} \subseteq \mathscr{F}$ such that $\mu(A_{\mathscr{A}}) > 0$, we add an edge $(\mathscr{A}, Q)$ for each $Q \in \mathscr{A}$ with capacity $c(\mathscr{A}, Q) = \mu(A_{\mathscr{A}})$.

(E3) For each $Q \in \mathscr{F}$, we add an edge $(Q, \ominus)$ with capacity $(Q, \ominus) = \Lambda^{-1}\mu(Q)$.

We can demonstrate the transformation with an example. Let us look at the collection of three rectangles $\mathscr{F} = \{Q_1, Q_2, Q_3\}$ shown in Figure 2.1a. The size of the sets and their overlap can be found in Table 2.1b. The transformation is done in five steps.

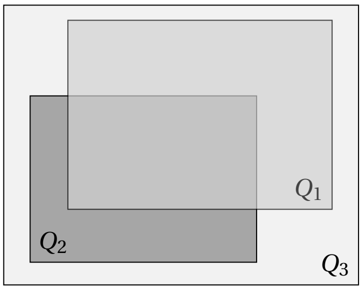
1. We first need to calculate the measure of the area of each subcollection. To do this, we use the values in Table 2.1b. The results can be found in Table 2.1c.

2. We then create the vertices. We start with vertices (V1) for the source and the sink. We then add vertices (V2) for all subcollections $\mathscr{A} \subseteq \mathscr{F}$ such that $\mu(A_{\mathscr{A}}) > 0$. We finally add vertices (V3) for the sets Q_1, Q_2 and Q_3.

3. We then add the edges (E1) from ⊕ to each vertex representing a subcollection. We give the edges a capacity that equals the measure of the area of said subcollection.

4. We add edges (E2) from the vertices representing a subcollection to the vertices representing a set in said subcollection. Each edge gets a capacity that equals the measure of the area of the subcollection from which it departs.

5. Finally, we add the edges (E3) between the vertices Q_1, Q_2 and Q_3, and ⊖. The weight of the edge from each Q_i to ⊖ equals $\Lambda^{-1}\mu(Q_i)$. We can find Λ using Definition 2. We calculate the Carleson constant for each subcollection $\mathscr{A} \subseteq \mathscr{F}$. These constants can be found in Table 2.1c. The Carleson constant of $\mathscr{F}$ is then the largest of these values, which is 2.



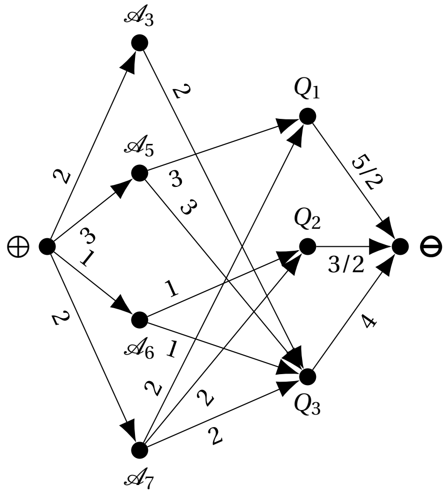
(a) The collection $\mathscr{F}$ of rectangles Q_1, Q_2 and Q_3.

Measures of the sets $\mathbf{Q} \in \mathscr{F}$ and their intersection
$\mu(Q_1) = 5$
$\mu(Q_2) = 3$
$\mu(Q_3) = 8$
$\mu(Q_1 \cap Q_2 \cap Q_3) = 2$

(b) The measures of different the sets of $\bigcup_{\mathbf{Q} \in \mathscr{F}} \mathbf{Q}$

Collection $\mathscr{A} \subseteq \mathscr{F}$	**$\mu(A_{\mathscr{A}})$**	**$\Lambda_{\mathscr{A}}$**
$\mathscr{A}_1 = \{Q_1\}$	$\mu(Q_1 \setminus (Q_2 \cup Q_3)) = 0$	1
$\mathscr{A}_2 = \{Q_2\}$	$\mu(Q_2 \setminus (Q_1 \cup Q_3)) = 0$	1
$\mathscr{A}_3 = \{Q_3\}$	$\mu(Q_3 \setminus (Q_1 \cup Q_2)) = 2$	1
$\mathscr{A}_4 = \{Q_1, Q_2\}$	$\mu((Q_1 \cap Q_2) \setminus Q_3) = 0$	4/3
$\mathscr{A}_5 = \{Q_1, Q_3\}$	$\mu((Q_1 \cap Q_3) \setminus Q_2) = 3$	13/8
$\mathscr{A}_6 = \{Q_2, Q_3\}$	$\mu((Q_2 \cap Q_3) \setminus Q_1) = 1$	11/8
$\mathscr{A}_7 = \{Q_1, Q_2, Q_3\}$	$\mu(Q_1 \cap Q_2 \cap Q_3) = 2$	2

(c) The possible subcollections $\mathscr{A} \subseteq \mathscr{F}$, the measures of their areas $\mu(\mathscr{A})$, and their Carleson constant $\Lambda_{\mathscr{A}}$. See Definitions 4 and 2.



(d) The complete graph G. The capacities of the edges follow from Table 2.1c and the fact that $\mathscr{F}$ is 2-Carleson.

Figure 2.1: Transforming a collection $\mathscr{F}$ into a graph.

2.2. Using the maximum flow in the graph to show that $\mathscr{F}$ is Λ^{-1}-sparse

Suppose that we transform a finite collection $\mathscr{F}$ as in Theorem 2.0.1 into a graph $G = (V, E, c)$ as described in the previous section. We will then end up with a graph with a finite amount of edges and vertices, because the amount of sets $Q \in \mathscr{F}$ and subcollections $\mathscr{A} \in \mathscr{F}$ are finite. The capacities of all the edges will also be finite, because for each edge (u, v) we have that $c(u, v) = \mu(A_{\mathscr{A}}) \leq \mu(Q)$ or $c(u, v) = \Lambda^{-1}\mu(Q) \leq \mu(Q)$ for some $Q \in \mathscr{F}$, and $\mu(Q) \leq \mu\left(\bigcup_{Q \in \mathscr{F}} Q\right)$, which is finite. Knowing these facts, we can use a MAX FLOW algorithm to find the maximum flow $f : E \mapsto \mathbb{R}_{\geq 0}$ in the graph G based on a finite $\mathscr{F}$. Any algorithm that is able to find the maximum flow in a finite, weighted, directed graph should work. A couple of options are mentioned in the Introduction. Depending on one's goal, one can either choose an algorithm that is easy to implement or one with a low time complexity. Time complexity will be discussed further in Section 3.3.

We give the following definitions that we will use to describe the flow in the graph.

Definition 5. *Let $G = (V, E, c)$ be a graph. We define $c_{\text{out}}(v)$ and $f_{\text{out}}(v)$ as the outgoing capacity and the outgoing flow of a vertex $v \in V$. In other words, we have that*

$$c_{\text{out}}(v) = \sum_{u \in V} c(v, u) \text{ and } f_{\text{out}}(v) = \sum_{u \in V} f(v, u).$$

We can define the incoming capacity and flow $c_{\text{in}}(v)$ and $f_{\text{in}}(v)$ in a similar way:

$$c_{\text{in}}(v) = \sum_{u \in V} c(u, v) \text{ and } f_{\text{in}}(v) = \sum_{u \in V} f(u, v).$$

Definition 6. *An edge (u, v) in a graph $G = (V, E, c)$ is saturated by a flow $f : E \mapsto \mathbb{R}_{\geq 0}$ if we have that:*

$$f(u, v) = c(u, v).$$

We now state the following lemma which will gives us a bit more information on what the capacities in our graph are.

Lemma 2.2.1. *Let (S, Σ, μ) be a measure space and let $\mathscr{F}$ be a Λ-Carleson collection of sets that are in Σ. Suppose we transform this collection into a graph $G = (V, E, c)$ according to the method described in Section 2.1. Let $\mathscr{A} \subseteq \mathscr{F}$ be arbitrary. The total amount of flow that can flow into vertices v_Q with $Q \in \mathscr{A}$ equals*

$$\sum_{Q \in \mathscr{A}} c_{\text{in}}(Q) = \sum_{\substack{\mathscr{A}' \subseteq \mathscr{F}: \\ \mathscr{A}' \cap \mathscr{A} \neq \emptyset}} \mu(A_{\mathscr{A}'}) = \mu\left(\bigcup_{Q \in \mathscr{A}} Q\right).$$

Proof. Let us transform $\mathscr{F}$ into a graph $G = (V, E, c)$ according to the method described in Section 2.1. All paths P are of the form

$$P = \oplus \to v_{\mathscr{A}_P} \to v_{Q_P} \to \ominus. \tag{2.2}$$

By (E2) we know that a path goes through a vertex representing a $Q \in \mathscr{A}$ if and only if it goes through a vertex representing a subcollection $\mathscr{A}' \subseteq \mathscr{F}$ which contains a $Q \in \mathscr{F}$ that is also in $\mathscr{A}$. By (E1) and (E2) we have that in each path P

$$c(\oplus, \mathscr{A}_P) = c(\mathscr{A}_P, Q_P). \tag{2.3}$$

So, all the flow that can come into a vertex $v_{\mathscr{A}'}$ with $\mathscr{A}' \subseteq \mathscr{F}$ such that $\mathscr{A}' \cap \mathscr{A} \neq \emptyset$ can also reach a vertex v_Q with $Q \in \mathscr{A} \cap \mathscr{A}'$. We thus have

$$\sum_{Q \in \mathscr{A}} c_{\text{in}}(Q) = \sum_{\substack{\mathscr{A}' \subseteq \mathscr{F}: \\ \mathscr{A}' \cap \mathscr{A} \neq \emptyset}} c(\oplus, v_{\mathscr{A}'}) = \sum_{\substack{\mathscr{A}' \subseteq \mathscr{F}: \\ \mathscr{A}' \cap \mathscr{A} \neq \emptyset}} \mu(A_{\mathscr{A}'}). \tag{2.4}$$

Now, by the first part of Lemma 2.1.1 we have that the sets $\{A_{\mathscr{A}}\}_{\mathscr{A} \subseteq \mathscr{F}}$ are pairwise disjoint. Putting this together with the second part of this lemma we have

$$\sum_{\substack{\mathscr{A}' \subseteq \mathscr{F}: \\ \mathscr{A}' \cap \mathscr{A} \neq \emptyset}} \mu(A_{\mathscr{A}'}) = \mu\left(\bigcup_{Q \in \mathscr{A}} \bigcup_{\mathscr{A}' \ni Q} A_{\mathscr{A}'}\right) = \mu\left(\bigcup_{Q \in \mathscr{A}} Q\right), \tag{2.5}$$

which finally gives us

$$\sum_{Q \in \mathscr{A}} c_{\text{in}}(Q) = \mu\left(\bigcup_{Q \in \mathscr{A}} Q\right), \tag{2.6}$$

which concludes the proof. □

We will use the Max-flow Min-cut Theorem [7] and the complementary slackness of MAX FLOW and MIN CUT problems, which is a direct consequence of this theorem, to give a proof of Theorem 2.0.1 based the maximum flow f. I will therefore state both here [1].

Theorem 2.2.2 (Max-flow Min-cut)**.** *Let $G = (V, E, c)$ be a weighted and directed graph with finitely many edges and vertices and with finite capacities. The value x of the maximum flow that can flow through G is equal to the value of a minimum cut y in G,*

Theorem 2.2.3 (Complementary slackness of Max-flow Min-cut)**.** *Let $G = (V, E, c)$ be a weighted and directed graph with finitely many edges and vertices and with finite capacities. Then a flow f is maximal and a cut $V_1 \subset V$ is minimal if and only if the following two statements are true.*

(M1) $f(u, v) = c(u, v)$ for all $(u, v) \in E$ with $u \in V_1$ and $v \in V \setminus V_1$.

(M2) $f(u, v) = 0$ for all $(u, v) \in E$ with $u \in V \setminus V_1$ and $v \in V_1$.

For a proof of these theorems the reader is referred to [7] [12, §10.3] and to the theory about complementary slackness in [1, Chapter 6]. We now state and prove a lemma which will be crucial for the proof of Theorem 2.0.1.

Lemma 2.2.4. *Let (S, Σ, μ) be a measure space and let $\mathscr{F}$ be a finite Λ-Carleson collection of sets that are in Σ. Suppose we transform this collection into a graph $G = (V, E, c)$ according to the method described in Section 2.1. Let f denote a maximum flow in this graph. There will be no $Q \in \mathscr{F}$ for which the edge $(Q, \ominus)$ is not saturated by f.*

[1] Formulation based on [1, Theorem 9.1 and 9.2]

Proof. Theorem 2.2.3 tells us that the vertices V in the graph G can be partitioned into V_1 and V_2 where

- $\oplus \in V_1$,
- $\ominus \in V_2$,
- The flow f saturates the edges from V_1 to V_2,
- The flow f through edges from V_2 to V_1 is 0.

We will use this to show that there is no $Q \in \mathscr{F}$ for which the edge $(Q, \ominus)$ is not saturated. Start by letting $\mathscr{F}_1 := \{Q \in \mathscr{F} : v_Q \in V_1\}$ and $\mathscr{F}_2 := \{Q \in \mathscr{F} : v_Q \in V_2\}$. Note that this means $\mathscr{F}_1 \cup \mathscr{F}_2 = \mathscr{F}$. Recall that all paths in G are of the form (2.2). We see that if an edge $(Q, \ominus)$ is not saturated by the maximum flow, we have that $Q \in \mathscr{F}_2$. Let us study such a set Q, which we will call Q_2.

Because the flow f is maximal, we must have that at least one edge in each path is saturated by f. Seeing that the flow does not saturate the edge $(Q_2, \ominus)$, we must have one of the following situations for each path through Q_2.

1. The edge $(\mathscr{A}, Q_2)$ is saturated by f. This means that $f(\mathscr{A}, Q_2) = \mu(A_{\mathscr{A}})$, which means that $\sum_{Q \in \mathscr{F}_2} f(\mathscr{A}, Q) = \mu(A_{\mathscr{A}})$.
2. Not the edge $(\mathscr{A}, Q_2)$, but edge $(\oplus, \mathscr{A})$ has reached maximum capacity. This means that $\mathscr{A} \in V_2$. Consequence (M2) from Theorem 2.2.3 then tells us that all Q for which $f(\mathscr{A}, Q) > 0$ are also in V_2, and therefore in $\mathscr{F}_2$. In other words, all the outgoing flow of $v_{\mathscr{A}}$ goes towards a vertex v_Q with $Q \in \mathscr{F}_2$. We also have that the total outgoing flow of $v_{\mathscr{A}}$ equals $c_{\text{in}}(\mathscr{A}) = \mu(A_{\mathscr{A}})$, because $(\oplus, \mathscr{A})$ is saturated. We can conclude that $\sum_{Q \in \mathscr{F}_2} f(\mathscr{A}, Q) = \mu(A_{\mathscr{A}})$.

We see that for each subcollection $\mathscr{A} \subseteq \mathscr{F}$ for which there exists an edge $(\mathscr{A}, Q)$ with $Q \in \mathscr{F}_2$, which is any $\mathscr{A} \subseteq \mathscr{F}$ such that $\mathscr{A} \cap \mathscr{F}_2 \neq \emptyset$, the following expression holds

$$\sum_{Q \in \mathscr{F}_2} f(\mathscr{A}, Q) = \mu(A_{\mathscr{A}}), \tag{2.7}$$

which means we have

$$\sum_{\substack{\mathscr{A} \subseteq \mathscr{F}: \\ \mathscr{A} \cap \mathscr{F}_2 \neq \emptyset}} \mu(A_{\mathscr{A}}) = \sum_{\substack{\mathscr{A} \subseteq \mathscr{F}: \\ \mathscr{A} \cap \mathscr{F}_2 \neq \emptyset}} \sum_{Q \in \mathscr{F}_2} f(\mathscr{A}, Q) = \sum_{Q \in \mathscr{F}_2} \sum_{\mathscr{A} \subseteq \mathscr{F}} f(\mathscr{A}, Q) = \sum_{Q \in \mathscr{F}_2} f_{\text{in}}(Q). \tag{2.8}$$

The middle equality holds because $f(\mathscr{A}, Q) = 0$ if $Q \notin \mathscr{A}$, and no $Q \in \mathscr{F}_2$ is in an $\mathscr{A}$ such that $\mathscr{A} \cap \mathscr{F}_2 = \emptyset$. By Lemma 2.2.1 we also have

$$\sum_{\substack{\mathscr{A} \subseteq \mathscr{F}: \\ \mathscr{A} \cap \mathscr{F}_2 \neq \emptyset}} \mu(A_{\mathscr{A}}) = \mu\left(\bigcup_{Q \in \mathscr{F}_2} Q\right). \tag{2.9}$$

This means we have

$$\mu\left(\bigcup_{Q \in \mathscr{F}_2} Q\right) = \sum_{Q \in \mathscr{F}_2} f_{\text{in}}(Q). \tag{2.10}$$

Now, a proof by contradiction can be given to show that there is no $Q \in \mathscr{F}$ such that the edge $(Q, \ominus)$ is not saturated. Suppose there exists at least one set $Q_0 \in \mathscr{F}_2$, such that the edge $(Q_0, \ominus)$ is not saturated by the maximum flow f. We will then have

$$\sum_{Q \in \mathscr{F}_2} \mu(Q) = \sum_{Q \in \mathscr{F}_2 \setminus \{Q_0\}} \mu(Q) + \mu(Q_0), \tag{2.11}$$

as we would with any collection $\mathscr{A}$ such that $Q \in \mathscr{A}$. Now recall that by (E3) each vertex $v_Q \in (V3)$ only has one outgoing edge $(Q, \ominus)$, meaning that for every $Q \in \mathscr{F}_2$ we have that

$$f_{\text{in}}(Q) = f_{\text{out}}(Q) \leq c(Q, \ominus) = \Lambda^{-1}\mu(Q). \tag{2.12}$$

For Q_0 we even have that the inequality in this expression is strict because $(Q_0, \ominus)$ is not saturated. Knowing these inequalities, (2.11) can be rewritten as

$$\sum_{Q \in \mathscr{F}_2} \mu(Q) > \sum_{Q \in \mathscr{F}_2} \Lambda f_{\text{in}}(Q) = \Lambda\mu\left(\bigcup_{Q \in \mathscr{F}_2} Q\right). \tag{2.13}$$

This expression is in contradiction with the Λ-Carleson condition, which means that all the edges between a vertex representing a set $Q \in \mathscr{F}$ and $\ominus$ are saturated. □

We are now finally ready to prove the main theorem of this Chapter.

Proof of Theorem 2.0.1. Transform $\mathscr{F}$ into a graph $G = (V, E, c)$ using the method described in Section 2.1. Let f be the maximum flow in this graph. We will prove that $\mathscr{F}$ is Λ^{-1}-sparse using Definition 3 with a function ϕ_Q that depends on the flow f. We set

$$\phi_Q = \mathbb{1}_Q \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_{\mathscr{A}})>0}} \mathbb{1}_{A_{\mathscr{A}}} \frac{f(\mathscr{A}, Q)}{\mu(A_{\mathscr{A}})}. \tag{2.14}$$

We first show that this function complies with condition (S2) from Definition 3. To do this, let $x \in \bigcup_{Q\in\mathscr{F}} Q$ be arbitrary and let $\mathscr{A}_x$ denote the subcollection $\mathscr{A}_x \subseteq \mathscr{F}$ of all sets $Q \in \mathscr{F}$ for which we have $\mu(A_{\mathscr{A}}) > 0$ and $x \in Q$. Note that by Definition 4 and Lemma 2.1.1 this implies $x \in A_{\mathscr{A}_x}$ and $x \notin A_{\mathscr{A}}$ for any other $\mathscr{A} \subseteq \mathscr{F}$. We have

$$\sum_{Q\in\mathscr{F}} \phi_Q(x) = \sum_{Q\in\mathscr{F}} \mathbb{1}_Q(x) \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_{\mathscr{A}})>0}} \mathbb{1}_{A_{\mathscr{A}}}(x) \frac{f(\mathscr{A}, Q)}{\mu(A_{\mathscr{A}})} = \sum_{Q\in\mathscr{A}_x} \frac{f(\mathscr{A}_x, Q)}{\mu\left(A_{\mathscr{A}_x}\right)} \le \frac{f_{\text{out}}(\mathscr{A}_x)}{\mu\left(A_{\mathscr{A}_x}\right)} \le \frac{c_{\text{in}}(\mathscr{A}_x)}{\mu\left(A_{\mathscr{A}_x}\right)} = 1, \tag{2.15}$$

because by (E1), $c_{\text{in}}(\mathscr{A}_x) = \mu(\mathscr{A}_x)$.

It is now left to prove that for a Λ-Carleson collection $\mathscr{F}$, this function also complies with (S1) from Definition 3 for $\eta = \Lambda^{-1}$. Let, $A_1 := \bigcup_{\mathscr{A}\subseteq\mathscr{F}:\mu(A_{\mathscr{A}})>0} A_{\mathscr{A}}$ and $A_2 := \bigcup_{\mathscr{A}\subseteq\mathscr{F}:\mu(A_{\mathscr{A}})=0} A_{\mathscr{A}}$. We then have that

$$\int_Q \phi_Q = \int_Q \mathbb{1}_Q \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_{\mathscr{A}})>0}} \mathbb{1}_{A_{\mathscr{A}}} \frac{f(\mathscr{A}, Q)}{\mu(A_{\mathscr{A}})} = \int_{A_1} \mathbb{1}_Q \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_{\mathscr{A}})>0}} \mathbb{1}_{A_{\mathscr{A}}}(x) \frac{f(\mathscr{A}, Q)}{\mu(A_{\mathscr{A}})} + \int_{A_2} \mathbb{1}_Q \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_{\mathscr{A}})>0}} \mathbb{1}_{A_{\mathscr{A}}} \frac{f(\mathscr{A}, Q)}{\mu(A_{\mathscr{A}})}, \tag{2.16}$$

because $A_1 \cup A_2 = \bigcup_{\mathscr{A}\in\mathscr{F}} A_{\mathscr{A}} = \bigcup_{Q\in\mathscr{F}} Q$ and A_1 and A_2 are pairwise disjoint (both these facts hold by Lemma 2.1.1). We clearly have that

$$\int_{A_2} \mathbb{1}_Q \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_{\mathscr{A}})>0}} \mathbb{1}_{A_{\mathscr{A}}} \frac{f(\mathscr{A}, Q)}{\mu(A_{\mathscr{A}})} = 0, \tag{2.17}$$

because for any $x \in \mathscr{A}_2$ we have that $x \notin \{\mathscr{A} \subseteq \mathscr{F} : \mu(A_{\mathscr{A}}) > 0\}$, meaning that the indicator in the summation will always equal 0. We therefore have that

$$\int_Q \phi_Q = \int_{A_1} \mathbb{1}_Q \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_{\mathscr{A}})>0}} \mathbb{1}_{A_{\mathscr{A}}}(x) \frac{f(\mathscr{A}, Q)}{\mu(A_{\mathscr{A}})} = \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_{\mathscr{A}})>0,\\ Q\in\mathscr{A}}} \int_{A_{\mathscr{A}}} \sum_{\substack{\mathscr{A}'\subseteq\mathscr{F}\\ \mu(A_{\mathscr{A}'}>0)}} \mathbb{1}_{A'_{\mathscr{A}}} \frac{f(\mathscr{A}', Q)}{\mu\left(A'_{\mathscr{A}}\right)}, \tag{2.18}$$

because by Lemma 2.1.1 we have that $Q = \bigcup_{\mathscr{A}\subseteq\mathscr{F}:Q\in\mathscr{F}} A_{\mathscr{A}}$ and that the sets $\{A_{\mathscr{A}}\}_{\mathscr{A}\subseteq\mathscr{F}}$ are pairwise disjoint. The fact that these sets are pairwise disjoint also means that we have $\mathbb{1}_{A_{\mathscr{A}}}(x) = 0$ for any $x \in A_{\mathscr{A}'}$ with $\mathscr{A}' \neq \mathscr{A}$. We see that the expression above equals

$$\sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_{\mathscr{A}})>0,\\ Q\in\mathscr{A}}} \int_{A_{\mathscr{A}}} \frac{f(\mathscr{A}, Q)}{\mu(A_{\mathscr{A}})} = \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_{\mathscr{A}})>0,\\ Q\in\mathscr{A}}} f(\mathscr{A}, Q), \tag{2.19}$$

which equals the total incoming flow $f_{\text{in}}(Q)$ of v_Q because by (E2) there are no edges $(\mathscr{A}, Q)$ for an $\mathscr{A}$ such that $Q \notin \mathscr{A}$, meaning there cannot come any flow from a vertex that does not represent a subcollection that contains Q. We can conclude that

$$\int_Q \phi_Q = f_{\text{in}}(Q) = f_{\text{out}}(Q), \tag{2.20}$$

which means that ϕ_Q complies with (S1) for $\eta = \Lambda^{-1}$ if we have that $f_{\text{out}}(Q) \ge \Lambda^{-1}\mu(Q)$ for each $Q \in \mathscr{F}$. By (E3) this is true whenever the maximum flow f saturates the edges $(Q, \ominus)$ for each $Q \in \mathscr{F}$, which by Lemma 2.2.4 is the case in a graph G based on a Λ-Carleson collection $\mathscr{F}$. □

3

An algorithm to find sets $\{E_Q\}_{Q\in\mathscr{F}}$ that satisfy the sparse condition

In this chapter I will present an algorithm that takes as input a finite collection $\mathscr{F}$ that is Λ-Carleson with respect to a divisible measure μ, and then constructs sets $\{E_Q\}_{Q\in\mathscr{F}}$ that satisfy the Λ^{-1}-sparse condition. Note that the measure μ must be divisible, because otherwise the sets $\{E_Q\}_{Q\in\mathscr{F}}$ might not exist, as explained in the introduction. The algorithm will use the maximum flow in the graph based on $\mathscr{F}$ according to the method described in Section 2.1, to distribute the available space amongst the sets $\{E_Q\}_{Q\in\mathscr{F}}$. One could say that we let the available space 'flow' to the different E_Q so that we end up with sets $\{E_Q\}_{Q\in\mathscr{F}}$ of the right size.

3.1. Constructing the algorithm

Let (S,Σ,μ) be a divisible measure space. Let $\mathscr{F}$ be a finite collection of sets in Σ that is Λ-Carleson with respect to μ such that $\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)$ is finite. Let $G=(V,E,c)$ be the graph based on $\mathscr{F}$ according to the method described in Section 2.1, and let f be the maximum flow in this graph. We will use the following algorithm to divide the available space in $\bigcup_{Q\in\mathscr{F}} Q$ amongst the sets $\{E_Q\}_{Q\in\mathscr{F}}$.

Algorithm 1: An algorithm to find sets $\{E_Q\}_{Q\in\mathscr{F}}$ for a finite collection $\mathscr{F}$ that is Λ-Carleson with respect to a divisible measure μ.

Data: A finite collection $\mathscr{F}$ that is Λ-Carleson with respect to a divisible measure μ.

begin
  Transform $\mathscr{F}$ into a graph according to the method described in Section 2.1.
  Run a MAX FLOW algorithm to find the maximum flow $f: E \mapsto \mathbb{R}_{\geq 0}$ in G.
  Set $E_Q = \emptyset$ for every $Q \in \mathscr{F}$.
  Set $A_{\mathscr{A}_{used}} = \emptyset$ for every $\mathscr{A} \subseteq \mathscr{F}$.
end
for $Q \in \mathscr{F}$ **do**
  for $v_{\mathscr{A}} \in (V2)$ **do**
    if $f(\mathscr{A}, Q) > 0$ **then**
      Select an arbitrary part $P_{\mathscr{A},Q} \subseteq A_{\mathscr{A}} \setminus A_{\mathscr{A}_{used}}$ such that $\mu\left(P_{\mathscr{A},Q}\right) = f(\mathscr{A}, Q)$.
      Let $A^*_{\mathscr{A}_{used}} = A_{\mathscr{A}_{used}}$.
      Set $A_{\mathscr{A}_{used}} = A^*_{\mathscr{A}_{used}} \cup P_{\mathscr{A},Q}$.
    else
      Set $P_{\mathscr{A},Q} = \emptyset$.
    end
  end
  Set $E_Q = \bigcup_{\mathscr{A}\subseteq\mathscr{F}: v_{\mathscr{A}}\in(V2)} P_{\mathscr{A},Q}$.
end

Result: The sets $\{E_Q\}_{Q\in\mathscr{F}}$.

We see that the idea of the algorithm is that if there is a flow of value $f(\mathscr{A},Q)$ from a subcollection $\mathscr{A}$ to a set Q, an arbitrary part of the area $A_{\mathscr{A}}$ of size $f(\mathscr{A},Q)$ becomes part of E_Q. Now, we need to be sure that this algorithm actually works. This is expressed in the following proposition.

Proposition 3.1.1. *Let (S,Σ,μ) be a divisible measure space. Let $\mathscr{F}$ be a finite collection of sets in Σ that is Λ-Carleson with respect to μ such that $\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)$ is finite. When running Algorithm 1 with $\mathscr{F}$ as input, each step in the algorithm is possible and the algorithm will finish in finite time.*

Proof. Clearly the initialisation is possible, as the method described in Section 2.1 will transform any finite collection $\mathscr{F}$ into a finite weighted and directed graph $G=(V,E,c)$ with a source and a sink. As explained in Section 2.2, can always find the maximum flow f in a finite weighted and directed graph in finite time with one of the algorithms mentioned in the Introduction. Lines 4 and 5 are possible and finite because we are just initialising variables.

We now get to the nested for-loops. These loops will only run finitely many times, because by assumption there are only finitely many $Q\in\mathscr{F}$, and the amount of vertices in (V2) is at most the total amount of subcollections $\mathscr{A}\subseteq\mathscr{F}$, which equals[1] $2^{|\mathscr{F}|}$, which is finite. Because we are only assigning variables in these for-loops, we know that each iteration will also finish in finite time. We do need to check if the actions within the loop are always possible. Particularly, we need to check whether we can always choose a suitable $P_{\mathscr{A},Q}$ in line 10. Note that because μ is divisible, we can always choose a part $P_{\mathscr{A},Q}\subseteq A_{\mathscr{A}}\setminus A_{\mathscr{A}_{used}}$ such that $\mu\left(P_{\mathscr{A},Q}\right)=f(\mathscr{A},Q)$, as long as $\mu\left(P_{\mathscr{A},Q}\right)=f(\mathscr{A},Q)\leq\mu\left(A_{\mathscr{A}}\setminus A_{\mathscr{A}_{used}}\right)$. We will prove by induction that this is the case in each iteration of the for-loop in line 7. Let Q_i with $i=1,\ldots,|\mathscr{F}|$ denote the set that is considered in the i^{th} iteration of this for-loop. Let $\mathscr{A}$ such that $v_{\mathscr{A}}\in(V2)$ be arbitrary. Let $P_{\mathscr{A},Q_i}$ denote the $P_{\mathscr{A},Q}$ that is constructed in the for-loop of Q_i. Note that by (E1) we have $c_{\text{in}}(\mathscr{A})=c(\ominus,\mathscr{A})=\mu(A_{\mathscr{A}})$, and that we have

$$\sum_{Q\in\mathscr{F}} f(\mathscr{A},Q)\leq f_{\text{out}}(\mathscr{A})\leq c_{\text{in}}(\mathscr{A})=\mu(A_{\mathscr{A}}). \tag{3.1}$$

Now, in the first iteration of the for-loop $A_{\mathscr{A}_{used}}=\emptyset$. It then follows directly that for Q_1 we have

$$\mu\left(P_{\mathscr{A},Q_1}\right)=f(\mathscr{A},Q_1)\leq\mu(A_{\mathscr{A}})=\mu\left(A_{\mathscr{A}}\setminus A_{\mathscr{A}_{used}}\right). \tag{3.2}$$

Now, let $n=2,\ldots,|\mathscr{F}|$ be arbitrary. Suppose that we had enough space for each Q_i with $i<n$. Line 12 tells us that when the for-loop in line 7 runs for Q_n, we have that $A_{\mathscr{A}_{used}}=\bigcup_{i=1}^{n-1}P_{\mathscr{A},Q_i}$. Combining this with (3.1) we see that we have

$$\begin{aligned}\mu\left(P_{\mathscr{A},Q_n}\right)=f(\mathscr{A},Q_n)\leq\mu(A_{\mathscr{A}})-\sum_{i=1}^{n-1}f(\mathscr{A},Q_i)&=\mu(A_{\mathscr{A}})-\sum_{i=1}^{n-1}\mu\left(P_{\mathscr{A},Q_i}\right)\\ &\leq\mu(A_{\mathscr{A}})-\mu\left(\bigcup_{i=1}^{n-1}P_{\mathscr{A},Q_i}\right)\leq\mu\left(A_{\mathscr{A}}\setminus A_{\mathscr{A}_{used}}\right).\end{aligned} \tag{3.3}$$

This means there is enough space in $A_{\mathscr{A}}$ for each $P_{\mathscr{A},Q}$ in every run of the nested for-loop in line 7. The other lines within the for-loops assign variables based on each $P_{\mathscr{A},Q}$. Because these sets exist, we know that the actions in these lines are possible. We conclude that each step in the algorithm is possible. □

Now before formally proving that Algorithm 1 not only works, but also yields the correct result, I will try to give some intuition on why it does. In other words, I will try to explain why the algorithm yields sets $\{E_Q\}_{Q\in\mathscr{F}}$ of the right size that do not overlap. Remember that the idea of the algorithm is that we let the available space flow towards the different sets in $\{E_Q\}_{Q\in\mathscr{F}}$ through the graph G based on $\mathscr{F}$. The space in each $A_{\mathscr{A}}$ flows through the vertex $v_{\mathscr{A}}$ to a set $E_Q\in\{E_Q\}_{Q\in\mathscr{F}}$.

Letting $\mathscr{A}=\mathscr{F}$ in the second part of Lemma 2.1.1 shows us that the total capacity of the edges (E1) coming out of the source equals the measure of the unions of all the sets in $\mathscr{F}$. This ensures that the total amount of space that is able to flow through the graph is the same as the amount of space that is in $\bigcup_{Q\in\mathscr{F}} Q$. Edges (E1) also make sure that the amount of space that can flow out of $v_{\mathscr{A}}$ equals exactly the amount of space that is in $A_{\mathscr{A}}$. Because the $\{A_{\mathscr{A}}\}_{A\subseteq\mathscr{F}}$ are disjoint by Lemma 2.1.1, the space that can flow towards a set $E_Q\in\{E_Q\}_{Q\in\mathscr{F}}$ through a vertex $v_{\mathscr{A}}$, cannot also flow towards the same or another set in $\{E_Q\}_{Q\in\mathscr{F}}$ through a vertex $v_{\mathscr{A}'}$ that represents a different subcollection $\mathscr{A}'$. The space in each set in $\{A_{\mathscr{A}}\}_{A\subseteq\mathscr{F}}$ can thus only be assigned to exactly one $E_Q\in\{E_Q\}_{Q\in\mathscr{F}}$, which ensures that the sets $\{E_Q\}_{Q\in\mathscr{F}}$ are pairwise disjoint.

[1]We use the notation $|\mathscr{F}|$ to denote the amount of elements in $\mathscr{F}$

Edges (E2) make sure that the space $A_{\mathscr{A}}$ can only flow to an $E_Q \in \{E_Q\}_{Q\in\mathscr{F}}$ such that $Q \in \mathscr{A}$, which means that the space that flows from $v_{\mathscr{A}}$ to v_Q, which is a subset of $A_{\mathscr{A}}$, is also a subset of Q, because $A_{\mathscr{A}} \subseteq Q$ by definition. This ensures that we always have $E_Q \subseteq Q$. If in part 2 of Lemma 2.1.1, we let $\mathscr{A} = \{Q\}$ for any $Q \in \mathscr{F}$, we see that (E2) also ensures that a set $E_Q \in \{E_Q\}_{Q\in\mathscr{F}}$ can get exactly the amount of space that is in the corresponding Q, meaning that this set E_Q can consist of any arbitrary part of Q. Finally, edges (E3) ensure that each $E_Q \in \{E_Q\}_{Q\in\mathscr{F}}$ does not get any more space than what is needed. This is important as the space could be used for another set $E_{Q'} \in \{E_Q\}_{Q\in\mathscr{F}}$.

We now give a proposition that formally shows that the sets $\{E_Q\}_{Q\in\mathscr{F}}$ found with Algorithm 1 are eligible candidates for sparse subsets of $\mathscr{F}$.

Proposition 3.1.2. *Let (S,Σ,μ) be a divisible measure space. Let $\mathscr{F}$ be a finite Λ-Carleson collection of sets that are in Σ such that $\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)$ is finite. The sets $\{E_Q\}_{Q\in\mathscr{F}}$ that result from running Algorithm 1 with $\mathscr{F}$ as input will meet the following two requirements.*

1. *For each $Q \in \mathscr{F}$, we have $E_Q \subseteq Q$.*

2. *The sets in $\{E_Q\}_{Q\in\mathscr{F}}$ are pairwise disjoint.*

Proof. **(1)** Let $Q \in \mathscr{F}$ be arbitrary. Line 17 from Algorithm 1 tells us that $E_Q = \bigcup_{\mathscr{A}\subseteq\mathscr{F}} P_{\mathscr{A},Q}$. From the if-statement in line 9 we know that $P_{\mathscr{A},Q} \neq \emptyset$ only if $f(\mathscr{A},Q) > 0$. We therefore have that

$$E_Q = \bigcup_{\substack{\mathscr{A}\subseteq\mathscr{F}: \\ f(\mathscr{A},Q)>0}} P_{\mathscr{A},Q}. \tag{3.4}$$

Because f describes a feasible flow, we must have $f(\mathscr{A},Q) \leq c(\mathscr{A},Q)$. By (E2) we know that $c(\mathscr{A},Q) > 0$ only if $Q \in \mathscr{A}$, which by Definition 4 means that $A_{\mathscr{A}} \subseteq Q$. Line 10 tells us that if $f(\mathscr{A},Q) > 0$, we have that $P_{\mathscr{A},Q} \subseteq A_{\mathscr{A}}$. So, for any $P_{\mathscr{A},Q}$ with $\mathscr{A}$ such that $f(\mathscr{A},Q) > 0$, we have $P_{\mathscr{A},Q} \subseteq Q$, which together with Equation (3.4) concludes the proof.

(2) We give a poof by contradiction. Suppose that there exists an $x \in \bigcup_{Q\in\mathscr{F}} Q$ such that $x \in E_{Q_1}$ and $x \in E_{Q_2}$, with $Q_1, Q_2 \in \mathscr{F}$, $Q_1 \neq Q_2$. Note that as a consequence of the first part of Lemma 2.1.1 we have that $x \in A_{\mathscr{A}}$ for exactly one $\mathscr{A} \subseteq \mathscr{F}$. Call this subcollection $\mathscr{A}'$. Without loss of generality, suppose that we first run the for-loop in line 7 for Q_1. For x to be in E_{Q_1} we must have $x \in P_{\mathscr{A},Q_1}$ for some $\mathscr{A} \subseteq \mathscr{F}$. As $P_{\mathscr{A},Q_1} \subseteq A_{\mathscr{A}}$ for each $\mathscr{A}$, we must have $x \in P_{\mathscr{A}',Q_1}$, which is only possible if $f(\mathscr{A}',Q_1) > 0$, which means the algorithm runs the if-statement that starts in line 9. We see that this means that $P_{\mathscr{A}'}$ becomes a subset of $A_{\mathscr{A}'_{used}}$, which means we now have $x \in A_{\mathscr{A}'_{used}}$. Now let us see what happens when the algorithm later runs the for-loop for Q_2. Because $x \in E_{Q_2}$, we must have $x \in P_{\mathscr{A},Q_2}$ for some $\mathscr{A} \subseteq \mathscr{F}$. By the same reasoning as for Q_1 we must have $x \in P_{\mathscr{A}',Q_2}$ for Q_2. Let us see if this is possible. We start at line 9. We must have $f(\mathscr{A}',Q_2) > 0$, because otherwise we have $P_{\mathscr{A},Q_2} = \emptyset$. We run the if-statement that starts in line 9. Line 10 tells us that we have $P_{\mathscr{A}',Q_2} \subseteq A_{\mathscr{A}'} \setminus A_{\mathscr{A}'_{used}}$. But, as $x \in E_{Q_1}$, we have $x \in A_{\mathscr{A}'_{used}}$, meaning we cannot possibly have $x \in P_{\mathscr{A}',Q_2}$ for Q_2. This is a contradiction and concludes the proof. □

3.2. Proof that the result from the algorithm satisfies the sparse condition

We will now show that the sets $\{E_Q\}_{Q\in\mathscr{F}}$ that result from Algorithm 1 with as input a finite collection $\mathscr{F}$ that is Λ-Carleson with respect to a divisible measure μ, are large enough to comply with Definition 1 for $\eta = \Lambda^{-1}$.

Theorem 3.2.1. *Let (S,Σ,μ) be a divisible measure space. Let $\mathscr{F}$ be a finite Λ-Carleson collection of sets that are in Σ such that $\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)$ is finite. Suppose we run Algorithm 1 with $\mathscr{F}$ as input. Then, the resulting sets $\{E_Q\}_{Q\in\mathscr{F}}$ satisfy Definition 1 for $\eta = \Lambda^{-1}$.*

Proof. By line 17 in Algorithm 1 we have that for each $Q \in \mathscr{F}$ we have $E_Q = \bigcup_{\mathscr{A}\subseteq\mathscr{F}} P_{\mathscr{A},Q}$. Now note that the sets $\{P_{\mathscr{A},Q}\}_{Q\in\mathscr{F},\mathscr{A}\subseteq\mathscr{F}}$ are pairwise disjoint. Firstly because by line 10, we clearly have

$$P_{\mathscr{A}_1,Q} \cap P_{\mathscr{A}_2,Q} \subseteq A_{\mathscr{A}_1} \cap A_{\mathscr{A}_2} = \emptyset \tag{3.5}$$

for $\mathscr{A}_1, \mathscr{A}_2 \subseteq \mathscr{F}$ with $\mathscr{A}_1 \neq \mathscr{A}_2$. Secondly, from the proof of part (2) of Proposition 3.1.2 it also follows that

$$P_{\mathscr{A},Q_1} \cap P_{\mathscr{A},Q_2} = \emptyset \tag{3.6}$$

for $Q_1, Q_2 \in \mathscr{F}$ with $Q_1 \neq Q_2$. We conclude that the sets $\{P_{\mathscr{A},Q}\}_{Q\in\mathscr{F},\mathscr{A}\subseteq\mathscr{F}}$ are indeed pairwise disjoint. We also know that for each $Q \in \mathscr{F}$, we have $\mu\left(P_{\mathscr{A},Q}\right) = f(\mathscr{A}, Q)$. So,

$$\mu\left(E_Q\right) = \sum_{\mathscr{A}\subseteq\mathscr{F}} \mu\left(P_{\mathscr{A}}\right) = \sum_{\mathscr{A}\subseteq\mathscr{F}} f(\mathscr{A}, Q) = f_{\text{in}}(Q) = f_{\text{out}}(Q). \tag{3.7}$$

So, to have $\mu\left(E_Q\right) \geq \Lambda^{-1}\mu(Q)$ for each $Q \in \mathscr{F}$, we must have $f_{\text{out}}(Q) \geq \Lambda^{-1}\mu(Q)$ for each $Q \in \mathscr{F}$, which by (E3) is true whenever the edges $(Q, \ominus)$ are saturated for each $Q \in \mathscr{F}$. Lemma 2.2.4 tells us that this is the case in our graph, which concludes the proof. □

3.3. The time complexity of the algorithm

Now that we have an algorithm that is able to find the sets $\{E_Q\}_{Q\in\mathscr{F}}$, it is an interesting question to ask whether the algorithm does this efficiently. We want to find out what the time complexity of the algorithm is as a function of properties of the Λ-Carleson family $\mathscr{F}$ that we give as input. We get a better understanding of the time complexity of Algorithm 1 through the following theorem.

Theorem 3.3.1. *Let (S,Σ,μ) be a divisible measure space. Let $\mathscr{F}$ be a finite Λ-Carleson collection of sets that are in Σ such that $\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)$ is finite. Suppose we run Algorithm 1 with $\mathscr{F}$ as input. Then, Algorithm 1 will finish $O\left(|V|^3\right)$*[2] *time, where $|V|$ is the amount of vertices in the graph that is created in line 2 of the algorithm.*

Proof. Let us look at what happens in lines 2 to 5 of Algorithm 1. In the proof of Proposition 3.1.2 we called this part the initialisation of the algorithm, but we will see that most the work actually already happens here. We will thus show that this initialisation already takes $O\left(|V|^3\right)$ time.

We see that in line 2 we transform $\mathscr{F}$ into a graph $G = (V, E, c)$ according to the method described in Section 2.1. This happens in $O(|V| + |E|)$ time, as this is the amount of variables we need to assign. In line 3 we use a MAX FLOW algorithm to find the maximum flow in the graph we just created. As mentioned in the Introduction, the best available algorithms can find the maximum flow in $O(|V||E|)$ time. As $|V|, |E| > 2$, we have that $|V| + |E| < |V||E|$. This means that transforming $\mathscr{F}$ into a graph takes less than $O(|V||E|)$ time, meaning lines 2 and 3 together happen in $O(|V||E|)$ time. We clearly have $|V| = O(|V|)$. As for the edges, we have at most one edge between each vertex, so we have $|E| = O\left(|V|^2\right)$. We conclude that the time it takes to find the maximum flow equals

$$O(|V||E|) = O\left(|V||V|^2\right) = O\left(|V|^3\right). \tag{3.8}$$

Now, to check whether the time complexity of the algorithm is not actually larger than $O\left(|V|^3\right)$, we need to check that the rest of the algorithm runs in less time. Setting $E_Q = \emptyset$ for every $Q \in \mathscr{F}$ (line 4) and setting $A_{\mathscr{A}_{used}} = \emptyset$ for each $\mathscr{A}$ such that $v_{\mathscr{A}} \in (V2)$ (line 5), clearly happens in $O(|(V3)|) + O(|(V2)|) = O(|V|)$ time as there are $|(V2)| + |(V3)|$ variables to be initialised. We then have the nested for-loop. The for-loop that starts in line 7 will run $|\mathscr{F}| = O(|V|)$ times, and the for-loop in line 8 will run $|(V2)| = O(|V|)$ times. The actions within these lines all happen in $O(1)$ time, as it only concerns assigning some variables. This means that nested for-loop will take $O\left(|V|^2\right)$ time. We conclude that the complexity of the entire algorithm is

$$O\left(|V|^3\right) + O(|V|) + O\left(|V|^2\right) = O\left(|V|^3\right). \tag{3.9}$$

□

We see that the time complexity of the algorithm depends on the number of vertices V in the graph $G = (V, E, c)$ that is based on the collection $\mathscr{F}$ that is given as input. We saw in Section 2.1 that we have

$$V = (V1) + (V2) + (V3). \tag{3.10}$$

We see that for any $\mathscr{F}$, we have that $|(V1)| = 2$. Also, we always have $|(V3)| = |\mathscr{F}|$. We cannot really impose any restriction on $\mathscr{F}$ to reduce these numbers, except limit the number of sets in $\mathscr{F}$. We see something different however for $|(V2)|$, which equals the number of subcollections $\mathscr{A} \subseteq \mathscr{F}$ such that $A_{\mathscr{A}} \neq \emptyset$. The number of vertices in this set can widely vary between different Carleson collections $\mathscr{F}$, even if they contain the same number of elements. The complexity of Algorithm 1 thus depends on the amount of overlap that we have in $\mathscr{F}$. We see in the following corollary that if we do not put any restrictions on the overlap between the sets in $\mathscr{F}$, the time complexity can become very large.

[2]We will see that the time complexity of the algorithm depends on the time complexity of the MAX FLOW algorithm that is used in line 3, it should be noted that if any faster MAX FLOW algorithm were to be used, the efficiency of the algorithm improves as well.

Corollary 3.3.2. *Let $\mathscr{F}$ be a collection as in Theorem 3.3.1 on which we put no restriction except that it must be Λ-Carleson. The running time of Algorithm 1 with $\mathscr{F}$ as input will be at most of order $O\left(2^{3|\mathscr{F}|}\right)$.*

Proof. We know from Theorem 3.3.1 that the time complexity of Algorithm 1 is $O\left(|V|^3\right)$. Now, because we do not put any restrictions on $\mathscr{F}$, the only upper bound for $|(V2)|$ we have is that $|(V2)| \leq |P(\mathscr{F})|$, as for any $\mathscr{A} \subseteq \mathscr{F}$ we could have that $A_{\mathscr{A}} \neq \emptyset$. This means that at worst we have $O(|(V2)|) = O\left(2^{|\mathscr{F}|}\right)$. Now we clearly have that $|(V1)| = 2 \leq O\left(2^{|\mathscr{F}|}\right)$ and that $|(V3)| = |\mathscr{F}| \leq O\left(2^{|\mathscr{F}|}\right)$. We thus have that

$$|V| = |(V1)| + |(V2)| + |(V3)| \leq O\left(2^{|\mathscr{F}|}\right). \tag{3.11}$$

We conclude that the time complexity of Algorithm 1 with $\mathscr{F}$ as input is at most

$$O\left(|V|^3\right) = O\left(\left(2^{|\mathscr{F}|}\right)^3\right) = O\left(2^{3|\mathscr{F}|}\right) \tag{3.12}$$

time. □

It should be noted that a time complexity of $O\left(2^{3|\mathscr{F}|}\right)$ time is very large, which means that the algorithm is not very efficient.

To improve the time complexity, we can try to put some restriction on $\mathscr{F}$ to reduce the amount of sets $\mathscr{A} \subseteq \mathscr{F}$ such that $mA_{\mathscr{A}} > 0$. In the next section we will see that the time complexity already greatly reduces in the case of dyadic cubes.

3.3.1. The complexity of the algorithm for a Λ-Carleson collection $\mathscr{F}$ of dyadic cubes

I already briefly mentioned dyadic cubes in the Introduction. We saw that Lerner and Nazarov [20] have been able to give a constructive proof of the equivalence of the Carleson and sparse condition for Λ-Carleson collection $\mathscr{F}$ of dyadic cubes. A collection $\mathscr{F}$ of dyadic cubes is a subcollection of cubes in a dyadic lattice $\mathscr{D}$, which is defined as follows[3].

Definition 7. *A dyadic lattice $\mathscr{D}$ in $\mathbb{R}^n$ is any collection of cubes such that*

(D1) If $Q \in \mathscr{D}$, then each child of Q is in $\mathscr{D}$ as well. A child of Q is any of the 2^n cubes obtained by partitioning Q by n hyperplanes parallel to the faces of Q that divide each edge into two equal parts.

(D2) Every 2 cubes $Q_1, Q_2 \in \mathscr{D}$ have a common ancestor. This means there exists a $Q \in \mathscr{D}$ such that Q_1 and Q_2 are both children of Q.

(D3) For every compact set $K \subset \mathbb{R}^n$, there exists a cube $Q \in \mathscr{D}$ containing K.

Proving that a Λ-Carleson collection $\mathscr{F}$ of dyadic cubes is also Λ^{-1}-sparse is easier because with dyadic cubes, the Carleson condition becomes local. This means that for a collection $\mathscr{F}$ of dyadic cubes, the Carleson condition as described in Definition 2 is equivalent to requiring that

$$\sum_{\substack{Q' \in \mathscr{F} \\ Q' \subseteq Q}} \mu\left(Q'\right) \leq \Lambda \mu(Q), \tag{3.13}$$

for every $Q \in \mathscr{F}$. Because of this property, if $\mathscr{F}$ is finite, we can construct the sets $\{E_Q\}_{Q \in \mathscr{F}}$ inductively, starting with the smallest cubes, and then working our way up.

Sadly, this locality does not hold in many other cases. As mentioned before, Carleson's counterexample [5] [27, Chapter 7] shows that it already fails in the case of dyadic rectangles. Because of this, construction the sets $\{E_Q\}_{Q \in \mathscr{F}}$ for a collection $\mathscr{F}$ of dyadic rectangles already is a lot harder than for dyadic cubes.

Seeing that the case of dyadic cubes is much simpler, we can check whether Algorithm 1 might have a lower time complexity if we take a Λ-Carleson collection $\mathscr{F}$ of dyadic cubes as input. Note that this will be quite useful, as more techniques are evolving to transfer results form the dyadic setting to the continuous setting [23, Chapter 1], for example because the dyadic maximal function controls the maximal function as a consequence of the Three Lattice Theorem [20, Theorem 3.1]. We will show that a collection of dyadic cubes the algorithm indeed becomes a lot more efficient, using the following lemma.

[3] For more information on dyadic cubes, see [20].

Lemma 3.3.3. *Let $\mathscr{D}$ be a dyadic lattice in $\mathbb{R}^n$, and let $\mathscr{F} \subseteq \mathscr{D}$ be a finite collection of dyadic cubes. Suppose we transform this collection into a graph $G = (V, E, c)$ as in Section 2.1. The amount of vertices in (V2) will not exceed the amount of cubes in $\mathscr{F}$.*

Proof. Recall that $|(V2)| = \left|\{\mathscr{A} \subseteq \mathscr{F} : \mu(A_{\mathscr{A}}) \geq 0\}\right|$. We will show that there is a one to one relation between the $Q \in \mathscr{F}$ and the $\mathscr{A} \subseteq \mathscr{F}$ such that we might have $A_{\mathscr{A}} \neq \emptyset$, which proves the lemma.

Now, because $\mathscr{F}$ is a family of dyadic cubes, Definition 7 tells us that for any $Q_1, Q_2 \in \mathscr{F}$ we either have $Q_1 \cap Q_2 = \emptyset$, $Q_1 \subseteq Q_2$ or $Q_2 \subseteq Q_1$. Suppose $\mathscr{F}$ contains N cubes Q. Randomly order each cube $Q \in \mathscr{F}$ as $Q_1, Q_2, ..., Q_N$, and for each $n = 1, ..., N$, let $\mathscr{A}_n = \left\{Q \in \mathscr{F} : Q_n \subseteq Q\right\}$. We will show that for each $\mathscr{A} \subseteq \mathscr{F}$ that is not in $\{A_n\}_{n\in\{1,...,N\}}$, we have that $A_{\mathscr{A}} = \emptyset$. Let $\mathscr{A}'$ be any such $\mathscr{A} \subseteq \mathscr{F}$ that is not in $\{A_n\}_{n\in\{1,...,N\}}$. We must then have a Q_a and a Q_b with $a, b \in \{1, ..., N\}$, such that Q_a is not a subset of Q_b and that Q_b is not a subset of Q_a. Otherwise we would have $\mathscr{A}' = \mathscr{A}_a$ or $\mathscr{A}' = \mathscr{A}_b$. Because Q_a and Q_b are both members of dyadic family $\mathscr{F}$, this means that we have $Q_a \cap Q_b = \emptyset$. We then have by Definition 4 that

$$A_{\mathscr{A}'} = \left(\bigcap_{Q\in\mathscr{A}'} Q\right) \cap \left(\bigcap_{Q\in\mathscr{F}\setminus\mathscr{A}'} Q^c\right) \subseteq Q_a \cap Q_b = \emptyset. \tag{3.14}$$

We conclude that only sets $\{A_n\}_{n\in\{1,...,N\}}$ can have an $A_{\mathscr{A}} \neq \emptyset$, which is as many as there are cubes in $\mathscr{F}$. □

We can use this lemma to show that Algorithm 1 actually has a polynomial time complexity when the input is a family $\mathscr{F}$ of dyadic cubes.

Corollary 3.3.4. *Let $\mathscr{F}$ be a finite family of dyadic cubes that is Λ-Carleson. If we run Algorithm 1 with $\mathscr{F}$ as input, the algorithm will finish in $O\left(|\mathscr{F}|^3\right)$ time.*

Proof. By Theorem 3.3.1 we know that Algorithm 1 has a time complexity of $O\left(|V|^3\right)$. By Lemma 3.3.3 we have that for a family $\mathscr{F}$ of dyadic cubes, $|(V2)| = |\mathscr{F}|$. This means that

$$|V| = |(V1)| + |(V2)| + |(V3)| \leq 2 + |\mathscr{F}| + |\mathscr{F}| = O(|\mathscr{F}|) \tag{3.15}$$

Combining these to facts yields the result that the algorithm finishes in $O\left(|\mathscr{F}|^3\right)$ time. □

4

Equivalence of Carleson and sparse condition for countable collections of sets

In this chapter we will study whether it is possible to generalise the proof of Chapter 2 so that it also holds for infinite collections of sets. It can easily be seen that if we try to transform an infinite collection $\mathscr{F}$ into a graph as in Section 2.1, we will end up with a graph with infinitely many vertices and edges. Seeing that the proof in Chapter 2 relies on theory about the maximum flow in this graph, we will have to look at theory about flow in infinite graphs to see whether an extension to infinite collections is possible.

4.1. Max-flow Min-cut for infinite graphs

If we look back at the proof for Theorem 2.0.1, we see that we transformed a Λ-Carleson collection $\mathscr{F}$ into a graph, and then used the maximum flow in this graph to prove that this collection $\mathscr{F}$ was also Λ^{-1}-sparse. The Max-flow Min-cut theorem (Theorem 2.2.2) played an important role in this proof. We will see if something similar is possible for an infinite collection $\mathscr{F}$. Note that the summation in Definition 2 tells us that we only need to consider countable collections of sets.

If we transform a countably infinite collection $\mathscr{F}$ into a graph $G = (V, E, c)$ in the same way we did in Section 2.1, (V3) and (E3) tell us that we will get a graph with an infinite amount of vertices and edges. By (V2), (E1) and (E2), the amount of vertices and edges might even be uncountable. This is because we could have $|(V2)| = |P(\mathscr{F})|$, and by Cantor's theorem [4] [24, Theorem 2.1], $P(\mathscr{F})$ is uncountable for a countably infinite collection $\mathscr{F}$.

As mentioned above, the Max-flow Min-Cut Theorem (Theorem 2.2.2) played a crucial role in the proof of the equivalence of the Carleson and sparse condition for finite collections. Luckily, this theorem has been generalised for infinite graphs [2, Conjecture 1.2] [21, Theorem 1], albeit only for graphs with a countable amount of edges. I will paraphrase the generalised version from [21, Theorem 1] below[1].

Theorem 4.1.1. *Let $G(V, E, c)$ be a weighted and directed graph with countably many edges. There exists a flow f and a cut $V_1 \subseteq V$ such that the following two statements are true.*

(IM1) $f(u, v) = c(u, v)$ *for all* $(u, v) \in E$ *with* $u \in V_1$ *and* $v \in V \setminus V_1$.

(IM2) $f(u, v) = 0$ *for all* $(u, v) \in E$ *with* $u \in V \setminus V_1$ *and* $v \in V_1$.

For the proof I refer the reader to [2, Section 6] or to [21, Section 4.2]. Do note that the fact that this theorem only holds for graphs with a countable amount of edges could be a problem because (E1) and (E2) make it so that a countable collection might still transform into a graph with uncountably many edges, as mentioned above. We must thus impose certain restrictions on our collection so that the graph will only contain countably many edges. In the following lemma we will see a restriction we can put on $\mathscr{F}$ to assure that we only have countably many edges.

Lemma 4.1.2. *Let (S, Σ, μ) be a measure space. Let $\mathscr{F}$ be a countable collection of sets that are in Σ, for which we only have countably many instances where sets in $\mathscr{F}$ overlap. In other words, we have only countably many*

[1] I paraphrased the theorem so that the notation and structure is similar to Theorem 2.2.3.

subcollections $\mathscr{A} \subseteq \mathscr{F}$ such that $\mu(A_{\mathscr{A}}) > 0$. If we transform $\mathscr{F}$ into a graph $G = (V, E, c)$ according to the method described in Section 2.1, the number of edges in G will be countable.

Proof. Note that it will suffice to prove whether the amount of vertices is countable. Indeed, the set of edges equals

$$E = \bigcup_{v \in V} E_v, \tag{4.1}$$

where we use E_v to denote the total amount of incoming and outgoing edges of vertex v. In Section 2.1 we see that by construction there can at most be one edge between each vertex. This means that for each $v \in V$, we have that $|E_v| \leq |V|$. This is countable if $|V|$ is countable. If this is the case, we have that (4.1) is a countable union of countable sets, which is again countable as a consequence of the Countable Axiom of Choice [8, Proposition 317]. We see that if $|V|$ is countable, $|E|$ is countable.

We will now check whether the amount of vertices is countable. Recall from Section 2.1 that the set V consists of three subsets, (V1), (V2) and (V3). It will suffice to prove that each of these subsets is countable because this implies that V is the countable union of countable sets which is countable by the Countable Axiom of Choice mentioned above. It is clear that (V1) is countable as $|(V1)| = 2$, which is countable. (V2) is countable because because there is only countably many instances where sets $Q \in \mathscr{F}$ overlap, which means there are only countably many $\mathscr{A} \subseteq \mathscr{F}$ such that $\mu(A_{\mathscr{A}}) > 0$. Finally, (V3) is countable because $\mathscr{F}$ is countable. So we can conclude that V is countable. □

Note that a countable collection $\mathscr{F}$ of dyadic cubes will also transform into a graph with a countable amount of edges, because as we have shown in Lemma 3.3.3, there is a one-to-one correspondence between the sets in $\mathscr{F}$ and the subcollections $\mathscr{A} \subseteq \mathscr{F}$ for which we could have $\mu(A_{\mathscr{A}}) > \emptyset$. This means there is only countably many $\mathscr{A} \subseteq \mathscr{F}$ such that $\mu(A_{\mathscr{A}}) > 0$.

4.2. The proof for infinite collections

With Lemma 3.3.3 we have found a category of infinite collections $\mathscr{F}$ that transform into a graph G for which the Max-flow Min-cut Theorem (Theorem 4.1.1) holds. This means that we know that there exists a maximum flow f in G and a corresponding minimum cut. Although we will not be able to find this flow, we can use the fact that it exists to generalise the proof from Chapter 2 and prove the following theorem.

Theorem 4.2.1. *Let $\mathscr{F}$ be a countably infinite collection as in Lemma 4.1.2 such that $\mu\left(\bigcup_{Q \in \mathscr{F}} Q\right)$ is finite. If $\mathscr{F}$ is Λ-Carleson, then $\mathscr{F}$ is Λ^{-1}-sparse.*

Proof. Transform $\mathscr{F}$ into a graph $G = (V, E, c)$ with the method described in Section 2.1. By Lemma 4.1.2 this graph only has countably many vertices and edges. Let f denote the maximum flow in this graph, which exists by Theorem 4.1.1. We will prove that $\mathscr{F}$ is Λ^{-1}-sparse using Definition 3. Recall that we did the same in the proof of Theorem 2.0.1. We will now do the same, so we let ϕ_Q be

$$\phi_Q^{\infty} = \mathbb{1}_Q \sum_{\substack{\mathscr{A} \subseteq \mathscr{F}: \\ \mu(A_{\mathscr{A}}) > 0}} \mathbb{1}_{A_{\mathscr{A}}} \frac{f(\mathscr{A}, Q)}{\mu(A_{\mathscr{A}})}, \tag{4.2}$$

The summation only goes over countably many elements because by assumption there is only countably many subcollections $\mathscr{A} \subseteq \mathscr{F}$ such that $\mu(A_{\mathscr{A}}) > 0$. Note that we do not need to know the exact values that f maps to in order to use this function to prove that $\mathscr{F}$ is Λ^{-1}-sparse. We only need to show that it complies with condition (S1) and (S2) from Definition 3.

Observe that ϕ_Q^{∞} always converges. Let $x \in \bigcup_{Q \in \mathscr{F}} Q$ be arbitrary. Because by (E2) we can only have $f(\mathscr{A}, Q) > 0$ if $Q \in \mathscr{A}$, which by Definition 4 implies that $A_{\mathscr{A}} \subseteq Q$ and therefore $\mu(A_{\mathscr{A}}) \leq \mu(Q)$, we have that

$$\phi_Q^{\infty}(x) = \mathbb{1}_Q(x) \sum_{\substack{\mathscr{A} \subseteq \mathscr{F}: \\ \mu(A_{\mathscr{A}}) > 0}} \mathbb{1}_{A_{\mathscr{A}}}(x) \frac{f(\mathscr{A}, Q)}{\mu(A_{\mathscr{A}})} \leq \sum_{\substack{\mathscr{A} \subseteq \mathscr{F}: \\ \mu(A_{\mathscr{A}}) > 0}} \frac{f(\mathscr{A}, Q)}{\mu(A_{\mathscr{A}})} \leq \sum_{\substack{\mathscr{A} \subseteq \mathscr{F}: \\ \mu(A_{\mathscr{A}}) > 0}} \frac{f(\mathscr{A}, Q)}{\mu(Q)} = \frac{f_{\text{in}}(Q)}{\mu(Q)} \leq \frac{c_{\text{out}}(Q)}{\mu(Q)} = \Lambda^{-1}, \tag{4.3}$$

which is finite.

We can show that $\phi_Q^{\infty}(x)$ complies with (S2) from Definition 3 in the same way we did in the proof of Theorem 2.0.1. For any $x \in \bigcup_{Q \in \mathscr{F}} Q$, let $\mathscr{A}_x$ denote the subcollection $\mathscr{A}_x \subseteq \mathscr{F}$ of all sets $Q \in \mathscr{F}$ for which we have $\mu(A_{\mathscr{A}}) > 0$ and $x \in Q$. We have

$$\sum_{Q \in \mathscr{F}} \phi_Q^{\infty}(x) = \sum_{Q \in \mathscr{F}} \mathbb{1}_Q(x) \sum_{\substack{\mathscr{A} \subseteq \mathscr{F}: \\ \mu(A_{\mathscr{A}}) > 0}} \mathbb{1}_{A_{\mathscr{A}}}(x) \frac{f(\mathscr{A}, Q)}{\mu(A_{\mathscr{A}})} = \sum_{Q \in \mathscr{A}_x} \frac{f(\mathscr{A}_x, Q)}{\mu\left(A_{\mathscr{A}_x}\right)} \leq 1, \tag{4.4}$$

where the last inequality holds for the same reason as in (2.15).

Now to prove that ϕ_Q^∞ satisfies condition (S1) with $\eta = \Lambda^{-1}$ for a Λ-Carleson collection $\mathscr{F}$, note that for each $Q \in \mathscr{F}$ we have

$$\int_Q \phi_Q^\infty = \int_Q \mathbb{1}_Q \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_\mathscr{A})>0}} \mathbb{1}_{A_\mathscr{A}} \frac{f(\mathscr{A},Q)}{\mu(A_\mathscr{A})}, \tag{4.5}$$

We again let $A_1 := \bigcup_{\mathscr{A}\subseteq\mathscr{F}:\mu(A_\mathscr{A})>0} A_\mathscr{A}$. By the same reasoning as in the proof of Theorem 2.0.1, we have that

$$\int_Q^\infty \phi_Q = \int_{A_1} \mathbb{1}_Q(x) \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_\mathscr{A})>0}} \mathbb{1}_{A_\mathscr{A}} \frac{f(\mathscr{A},Q)}{\mu(A_\mathscr{A})} = \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_\mathscr{A})>0,\\ Q\in\mathscr{A}}} \int_{A_\mathscr{A}} \sum_{\substack{A'\subseteq\mathscr{F}:\\ \mu(A'_\mathscr{A})>0}} \mathbb{1}_{A'_\mathscr{A}} \frac{f(\mathscr{A}',Q)}{\mu(A'_\mathscr{A})} = \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_\mathscr{A})>0,\\ Q\in\mathscr{A}}} \int_{A_\mathscr{A}} \frac{f(\mathscr{A},Q)}{\mu(A_\mathscr{A})}, \tag{4.6}$$

where the last two equalities holds because the sets $\{A_\mathscr{A}\}_{\mathscr{A}\subseteq\mathscr{F}}$ are pairwise disjoint (see the proof of Theorem 2.0.1). We conclude that

$$\int_Q \phi_Q^\infty = \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_\mathscr{A})>0,\\ Q\in\mathscr{A}}} \int_{A_\mathscr{A}} \frac{f(\mathscr{A},Q)}{\mu(A_\mathscr{A})} = \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_\mathscr{A})>0,\\ Q\in\mathscr{A}}} f(\mathscr{A},Q) = f_{\text{in}}(Q) = f_{\text{out}}(Q). \tag{4.7}$$

We conclude that ϕ_Q^∞ complies with (S1) if $f_{\text{out}}(Q) = \Lambda^{-1}\mu(Q)$ for each $Q \in \mathscr{F}$, which is the case if the edges $(Q,\ominus)$ are saturated for every $Q \in \mathscr{F}$. We have proven that this is the case for a graph based on a finite Carleson collection. We will now see if this proof also works for a countably infinite collection $\mathscr{F}$, by going through all the steps in this proof, and checking whether they are also possible in the countably infinite case.

The proof of Lemma 2.2.4, starts by partitioning the graph G. Theorem 4.1.1 tells us that because the graph G based on a Λ-Carleson collection $\mathscr{F}$ as described in Lemma 4.1.2 has a countable amount of edges, we can also partition this graph in the same way. That is, we can partition the vertices V in G into V_1 and V_2 such that

- $\oplus \in V_1$,
- $\ominus \in V_2$,
- The flow f saturates the edges from V_1 to V_2,
- The flow f through the edges form V_2 to V_2 is 0,

just like in the finite case. We again define $\mathscr{F}_1 := \{Q \in \mathscr{F} : v_Q \in V_1\}$ and $\mathscr{F}_2 := \{Q \in \mathscr{F} : v_Q \in V_2\}$.

The paths in the infinite graph G also are of the form (2.2). The maximum flow f saturates at least one edge in each path, so by the same reasoning as in the proof of Lemma 2.2.4 we have that for each subcollection $\mathscr{A} \subseteq \mathscr{F}$ for which there exists an edge $(\mathscr{A},Q)$ with $Q \in \mathscr{F}_2$ the following expression holds

$$\sum_{Q\in\mathscr{F}_2} f(\mathscr{A},Q) = \mu(A_\mathscr{A}). \tag{4.8}$$

This means we have a similar result as in (2.8):

$$\sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_\mathscr{A})>0,\\ \mathscr{A}\cap\mathscr{F}_2\neq\emptyset}} \mu(A_\mathscr{A}) = \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_\mathscr{A})>0,\\ \mathscr{A}\cap\mathscr{F}_2\neq\emptyset}} \sum_{Q\in\mathscr{F}_2} f(\mathscr{A},Q) = \sum_{Q\in\mathscr{F}_2} \sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(A_\mathscr{A})>0}} f(\mathscr{A},Q) = \sum_{Q\in\mathscr{F}_2} f_{\text{in}}(Q). \tag{4.9}$$

Note that these summations will always converge, as we are summing flows in the graph which are all on a different path. The sum of these flows must always be smaller than the total incoming capacity of the graph, which equals $\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)$ which is finite. We can also give slight generalisation of the proof in of Lemma 2.2.1 that shows that

$$\sum_{\substack{\mathscr{A}\subseteq\mathscr{F}:\\ \mu(\mathscr{A})>0,\\ \mathscr{A}\cap\mathscr{F}_2\neq\emptyset}} \mu(A_\mathscr{A}) = \mu\left(\bigcup_{Q\in\mathscr{A}} \bigcup_{\substack{\mathscr{A}'\subseteq\mathscr{F}:\\ \mu(\mathscr{A}')>0\\ \mathscr{A}'\cap\mathscr{A}\neq 0}} A_\mathscr{A}\right) = \mu\left(\bigcup_{Q\in\mathscr{A}} \bigcup_{\substack{\mathscr{A}'\subseteq\mathscr{F}:\\ \mathscr{A}'\cap\mathscr{A}\neq 0}} A_\mathscr{A}\right) = \mu\left(\bigcup_{Q\in\mathscr{F}_2} Q\right), \tag{4.10}$$

where the last inequality holds by Lemma 2.1.1. We now see that that

$$\mu\left(\bigcup_{Q\in\mathscr{F}_2} Q\right) = \sum_{Q\in\mathscr{F}_2} f_{\text{in}}(Q). \tag{4.11}$$

We can now again give a proof by contradiction to show that the edges $(Q,\ominus)$ are saturated for every $Q \in \mathscr{F}$. Suppose there is at least one $Q_0 \in \mathscr{F}$ such that the edge $(Q_0,\ominus)$ is not saturated. We can make the same argument as in the proof of Lemma 2.2.4 that shows that

$$\sum_{Q\in\mathscr{F}_2} \mu(Q) > \sum_{Q\in\mathscr{F}_2} \Lambda f_{\text{in}}(Q) = \Lambda\mu\left(\bigcup_{Q\in\mathscr{F}_2} Q\right). \tag{4.12}$$

Note that $\sum_{Q\in\mathscr{F}_2} \Lambda f_{\text{in}}(Q)$ is always finite, as this value will never exceed the total incoming capacity of the graph which is finite. This means that this inequality holds. We see that we have found a contradiction with the Carleson condition. We conclude that the edges $(Q,\ominus)$ are saturated for each $Q \in \mathscr{F}$, and thus that $\mathscr{F}$ must be Λ^{-1}-sparse. □

Note that there are no algorithms known that can actually find the flow f in a graph with a countably infinite amount of edges. We will therefore not be able to actually construct the sets $\{E_Q\}_{Q\in\mathscr{F}}$ with an algorithm similar to Algorithm 1.

5

Approximating the Carleson constant

We saw in Section 3.3 that for Carleson collections $\mathscr{F}$ with enough structure, we can find the sets $\{E_Q\}_{Q\in\mathscr{F}}$ complying with the Λ^{-1}-sparse definition in polynomial time. It will be interesting to study whether we can use a similar algorithm to find the Carleson constant of a collection $\mathscr{F}$ when it is unknown. I will introduce such an algorithm in this section. The idea is that we take a Λ-Carleson collection $\mathscr{F}$ for which the Carleson constant Λ is unknown. We then make an educated guess Λ' for the Carleson constant and transform the collection into a graph as if the constant Λ' were correct and find the maximum flow in this graph. We then use this flow and the complementary minimum cut to come up with a better estimate, and find the maximum flow in the graph based on $\mathscr{F}$ and the new estimate. We keep correcting until we have found the correct Carleson constant. Note that we will restrict ourselves to finite families in this chapter, as we cannot find the maximum flow in an infinite graph (see Chapter 4).

5.1. Correcting a Carleson estimate based on the resulting $\{\mathbf{E_Q}\}_{\mathbf{Q}\in\mathscr{F}}$

To understand how we might use the maximum flow in the graph based on an estimated Carleson constant to correct our guess, we can make use of the following proposition.

Proposition 5.1.1. *Let (S,Σ,μ) be a divisible measure space. Let $\mathscr{F}$ be a finite Λ-Carleson collection of sets in Σ such that $\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)$ is finite. Suppose we use the method described in Section 2.1 to transform $\mathscr{F}$ into a graph $G=(V,E,c)$ as if an estimate $\Lambda'<\Lambda$ were the Carleson constant. That is, we set $c(Q,\ominus)=(\Lambda')^{-1}\mu(Q)$ instead of $c(Q,\ominus)=\Lambda^{-1}\mu(Q)$ for all $Q\in\mathscr{F}$. The maximum flow in this graph will leave at least one edge $(Q,\ominus)$ with $Q\in\mathscr{F}$ unsaturated.*

Proof. Let $\mathscr{F}$ be a finite Λ-Carleson collection, and let $\Lambda'<\Lambda$ be an estimate for Λ that is too small. By Definition 2 this means that

$$\sum_{Q\in\mathscr{A}}\mu(Q)>\Lambda'\mu\left(\bigcup_{Q\in\mathscr{A}}Q\right) \tag{5.1}$$

for some subcollection $\mathscr{A}\subseteq\mathscr{F}$. Call this subcollection $\mathscr{A}'$. Note that we assume $\Lambda'\geq 1$, because $\Lambda\geq 1$ by definition. Now transform $\mathscr{F}$ into a graph $G=(V,E,c)$ according to the method in Section 2.1, but as if Λ' were the Carleson constant. This means that for each $Q\in\mathscr{F}$, we set $c(Q,\ominus)=\Lambda^{-1}\mu(Q)$. (E3) tells us that the total amount of flow that can flow out of vertices v_Q such that $Q\in\mathscr{A}'$ equals

$$\sum_{Q\in\mathscr{A}'}c(Q,\ominus)=\frac{1}{\Lambda'}\sum_{Q\in\mathscr{A}'}\mu(Q). \tag{5.2}$$

By Lemma 2.2.1 we have that the total amount of flow that can flow into vertices v_Q such that $Q\in\mathscr{A}'$ equals

$$c_{\text{in}}(\mathscr{A}')=\mu\left(\bigcup_{Q\in\mathscr{A}'}Q\right). \tag{5.3}$$

The flow coming out of the vertices v_Q with $Q\in\mathscr{A}'$ can thus never be larger than this value. We thus have

$$\sum_{Q\in\mathscr{A}'}f(Q,\ominus)\leq c_{\text{in}}(\mathscr{A}')=\mu\left(\bigcup_{Q\in\mathscr{A}'}Q\right)<\frac{1}{\Lambda'}\sum_{Q\in\mathscr{A}'}\mu(Q)=\sum_{Q\in\mathscr{A}'}c(Q,\ominus). \tag{5.4}$$

We conclude that the flow indeed does not saturate all the edges from a vertex representing a set $Q \in \mathscr{F}$ to $\ominus$. □

5.2. Finding a lower bound for the Carleson constant

Proposition 5.1.1 gives us a way to verify whether an estimate for the Carleson constant Λ is too small, because if our estimate is too small, there will be at least one $Q \in \mathscr{F}$ such that $(Q, \ominus)$ is not saturated. Our approach will therefore be to approximate the Carleson constant from below. To do this, we need to find a lower bound for the Carleson constant. Such a lower bound is easily deducted from Definition 2, which we do in the following proposition.

Proposition 5.2.1. *Let (S, Σ, μ) be a divisible measure space. Let $\mathscr{F}$ be a finite Λ-Carleson collection of sets in Σ such that $\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)$ is finite. For $\mathscr{F}$, we always have*

$$\Lambda \geq \frac{\sum_{Q\in\mathscr{A}} \mu(Q)}{\mu\left(\bigcup_{Q\in\mathscr{A}} Q\right)}, \tag{5.5}$$

for any subcollection $\mathscr{A} \subseteq \mathscr{F}$.

Proof. Let us look at Definition 2, which states that if $\mathscr{F}$ is Λ-Carleson, we have

$$\sum_{Q\in\mathscr{A}} \mu(Q) \leq \Lambda\mu\left(\bigcup_{Q\in\mathscr{A}} Q\right) \tag{5.6}$$

for any subcollection $\mathscr{A} \subseteq \mathscr{F}$. This can be rewritten as

$$\frac{\sum_{Q\in\mathscr{A}} \mu(Q)}{\mu\left(\bigcup_{Q\in\mathscr{A}} Q\right)} \leq \Lambda, \tag{5.7}$$

which concludes the proof. □

Now that we have found a lower bound for the Carleson constant, it would be nice to know some properties of the graph $G = (V, E, c)$ that is based on a Λ-Carleson collection $\mathscr{F}$ and the estimate Λ' that we found with Proposition 5.2.1. We describe some in the following lemma.

Lemma 5.2.2. *Let (S, Σ, μ) be a divisible measure space. Let $\mathscr{F}$ be a finite Λ-Carleson collection of sets in Σ such that $\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)$ is finite. Suppose we use the method described in Section 2.1 to transform this collection into a graph $G = (V, E, c)$ as if the Carlson constant is*

$$\Lambda' = \frac{\sum_{Q\in\mathscr{F}} \mu(Q)}{\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)}, \tag{5.8}$$

instead of the actual Λ. If $\Lambda' \neq \Lambda$, then the following statement holds.

When partitioning V into V_1 and V_2 amongst a minimum cut in the same way as in the proof of Lemma 2.2.4, there is at least one $Q \in \mathscr{F}$ such that $v_Q \in V_1$. Furthermore, let $\mathscr{A} \subseteq \mathscr{F}$ be any subcollection for which we have $f(\mathscr{A}, Q_1) > 0$ with $Q_1 \in \mathscr{F}$. We then have that any $Q \in \mathscr{F}$ for which there exists an edge $(\mathscr{A}, Q)$, that $Q \in V_1$.

Proof. Note that Λ' is a lower bound for Λ, meaning that if $\Lambda' \neq \Lambda$, we have $\Lambda' < \Lambda$. Let $G = (V, E, c)$ be the graph based on a Λ-Carleson collection $\mathscr{F}$, as if Λ' were the Carleson constant. We will prove the first part of the statement using a proof by contradiction. We thus suppose that $\Lambda' < \Lambda$ and that there is no $Q \in \mathscr{F}$ such that $v_Q \in V_1$. This means that the maximum flow saturates all paths from $\oplus$ to v_Q for any $Q \in \mathscr{F}$. Recall that all paths P are of the form (2.2). If we want to have no $Q \in \mathscr{F}$ such that $v_Q \in V_1$, we must have that in each path P either the edge $(\oplus, \mathscr{A}_P)$ or the edge $(\mathscr{A}_P, Q_P)$ is saturated. Note that by (E1) and (E2) we have that in each path P

$$c(\oplus, \mathscr{A}_P) = c(\mathscr{A}_P, Q_P). \tag{5.9}$$

Combining this with the fact that we know that each vertex representing a subcollection $\mathscr{A}$ only has one incoming edge whilst it has one or more outgoing edges, we know that

$$f(\oplus, \mathscr{A}_P) \geq f(\mathscr{A}_P, Q_P) \tag{5.10}$$

in each path. This means that $(\mathscr{A}_P, Q_P)$ can only be saturated if $(\oplus, \mathscr{A}_P)$ is saturated as well. Thus for all paths to be saturated before reaching a vertex v_Q, we must have that $(\oplus, \mathscr{A})$ is saturated for each $\mathscr{A} \subseteq \mathscr{F}$ such that $v_{\mathscr{A}} \in V$. We must thus have a total flow out of the source $\oplus$ of value

$$\sum_{\mathscr{A} \subseteq \mathscr{F}} c(\oplus, \mathscr{A}) = \sum_{\mathscr{A} \subseteq \mathscr{F}} \mu(A_{\mathscr{A}}) = \mu\left(\bigcup_{Q \in \mathscr{F}} Q\right), \tag{5.11}$$

where the last equality holds by Lemma 2.2.1. For this to be possible, the total flow in the graph must be of this value. This is only possible if the total outgoing capacity towards the sink is at least of the same value. By (E3) we have that this capacity equals

$$\sum_{Q \in \mathscr{F}} c(Q, \ominus) = \sum_{Q \in \mathscr{F}} \frac{1}{\Lambda'} \mu(Q). \tag{5.12}$$

Additionally, Proposition 5.1.1 tells us that at least one of these edges in not saturated as Λ' is smaller than the actual Carleson constant. This means that for the flow to be feasible, we must have

$$\sum_{\mathscr{A} \subseteq \mathscr{F}} f(\oplus, \mathscr{A}) = \mu\left(\bigcup_{Q \in \mathscr{F}} Q\right) < \frac{1}{\Lambda'} \sum_{Q \in \mathscr{F}} \mu(Q) = \sum_{Q \in \mathscr{F}} c(Q, \ominus). \tag{5.13}$$

But, by (5.8) we have

$$\mu\left(\bigcup_{Q \in \mathscr{F}} Q\right) = \frac{1}{\Lambda'} \sum_{Q \in \mathscr{F}} \mu(Q), \tag{5.14}$$

which gives us our contradiction. We conclude there is at least one $Q \in \mathscr{F}$ such that $v_Q \in V_1$.

Now for the proof of the second statement, let $\mathscr{A}$ be any set $\mathscr{A} \subseteq \mathscr{F}$ such that $f(\mathscr{A}, Q_1) > 0$ for some $Q_1 \in V_1$. By (M2) from Theorem 2.2.3 this means that $\mathscr{A} \in V_1$. Now let Q be any set in $\mathscr{F}$ such that the edge $(\mathscr{A}, Q)$ exists. By (E1) and (E2) we have that $c_{\text{in}}(\mathscr{A}) = c(\mathscr{A}, Q)$. This means that $f(\mathscr{A}, Q) \leq c_{\text{in}} - f(\mathscr{A}, Q_1) < c(\mathscr{A}, Q)$, meaning $(\mathscr{A}, Q)$ is not saturated, which by (M1) means that we have $Q \in V_1$. □

5.3. An algorithm to approximate the Carleson constant

Bringing all the above observations together allows us to construct the following algorithm for approximating the Carleson constant.

Algorithm 2: An algorithm to approximate the Carleson constant of a finite Carleson collection $\mathscr{F}$.

```
    Data: A finite Carleson collection 𝓕.
1   begin
2   |  Set 𝓕' = 𝓕.
3   end
4   while 𝓕' ≠ ∅ do
5   |  Set Λ' = Σ_{Q∈𝓕'} μ(Q) / μ(⋃_{Q∈𝓕'} Q).
6   |  Transform 𝓕' into a graph G according to the method described in Section 2.1 as if Λ' were the
    |    Carleson constant of 𝓕'.
7   |  Run a MAX FLOW algorithm to find the maximum flow f : E ↦ ℝ_{≥0} in G.
8   |  Use the Max-flow Min-cut Theorem to partition the graph into V_1 and V_2 such that the edges
    |    between V_1 and V_2 are saturated. Do this in such a way that v ∈ V_2 if and only if there is an
    |    unsaturated path between v and ⊖.
9   |  Let 𝓕' = {Q : v_Q ∈ V_2}.
10  end
11  return Λ'.
    Result: The Carleson constant Λ of 𝓕.
```

Note that all the steps in this algorithm are possible. Assigning variables is always possible, and we have already shown in Chapter 2 that we can transform $\mathscr{F}$' into a graph and find the maximum flow in this graph because $\mathscr{F}$' is finite. The partition that is made in line 8 is possible because all paths in $\mathscr{F}$' are of the form (2.2).

The maximum flow f always saturates at least one of the edges in each path. If only one edge is saturated, we make the cut in this place. For example, if only the edge $(\mathscr{A}_P, Q_P)$ is saturated for a path P, we get $v_{\mathscr{A}_P} \in V_1$ and $v_{Q_P} \in V_2$. If two edges are saturated in the same path, we make the cut along the edge that is closest to $\ominus$. So, for example, if $(\mathscr{A}_P, Q_P)$ and $(Q_P, \ominus)$ are saturated for a path P, we have that $v_{\mathscr{A}_P} \in V_1$ and $v_{Q_P} \in V_1$. We see that if we make the partition in this way, we always have an unsaturated path from a $v \in V_2$ to $\ominus$. Now, to prove that this algorithm will yield a correct result, we first need the following lemma.

Lemma 5.3.1. *Let (S, Σ, μ) be a divisible measure space. Let $\mathscr{F}$ be a finite Λ-Carleson collection of sets in Σ such that $\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)$ is finite. Suppose we run Algorithm 2 with $\mathscr{F}$ as input. Suppose the while loop in lines 4 to 9 needs to run N times. Let Λ_n denote the value of Λ' in the n^{th} iteration of the while-loop. We then always have that*

$$\Lambda_n \leq \Lambda_{n+1}$$

for $n \in \{1, 2, ..., N-1\}$.

Proof. Suppose we are in the n^{th} iteration of the while loop for any $n \in \{1, ..., N-1\}$. Let $\mathscr{F}_n$ denote $\mathscr{F}'$ in the beginning of the iteration. Let G_n be the graph that is constructed in line 6, let f_n denote the maximum flow found in line 7, and let $V_{1,n}$ and $V_{2,n}$ denote the partitions from line 8. Let $\mathscr{F}_{2,n} := \{Q \in \mathscr{F}_n : v_Q \in V_{2,n}\}$. By line 9 this equals $\mathscr{F}_{n+1}$.

Now if $\Lambda_n = \Lambda$, Theorem 2.0.1 tells us that the edges $(Q, \ominus)$ should be saturated for all $Q \in \mathscr{F}_n$. This would mean that $\mathscr{F}_{n+1} = \mathscr{F}_{2,n} = \emptyset$ which would mean that this n^{th} loop would be the last iteration of the while-loop. But, we assumed that $n \leq N-1$, so we conclude that $\Lambda_n \neq \Lambda$.

If $\Lambda' \neq \Lambda$ we have by (E3) that the total outgoing capacity of the vertices representing a $Q \in \mathscr{F}_{2,n}$ equals

$$\sum_{Q\in\mathscr{F}_{2,n}} c_{n,\text{in}}(Q) = \sum_{Q\in\mathscr{F}_{2,n}} c_n(Q, \ominus) = \frac{1}{\Lambda_n} \sum_{Q\in\mathscr{F}_{2,n}} \mu(Q). \tag{5.15}$$

Because the edges $(Q, \ominus)$ with $Q \in \mathscr{F}_{2,n}$ are not saturated, we must have $f_n(\mathscr{F}_{2,n}) < c_{n,\text{out}}(\mathscr{F}_{2,n})$. Now, by Lemma 2.2.1 we have that the total incoming capacity of paths going through $\mathscr{F}_{2,n}$ equals

$$\sum_{Q\in\mathscr{F}_{2,n}} c_{n,\text{in}}(Q) = \mu\left(\bigcup_{Q\in\mathscr{F}_{2,n}} Q\right). \tag{5.16}$$

Because of the second part of Lemma 5.2.2 we know that no flow that goes through any vertices $v_{\mathscr{A}}$ such that there exists an edge $(\mathscr{A}, Q_2)$ with $Q_2 \in \mathscr{F}_{2,n}$, flows towards a vertex v_Q with $v_Q \in V_{1,n}$. Seeing that the edges $(\mathscr{A}, Q)$ can not be saturated before $(\oplus, \mathscr{A})$ is saturated (see the proof of Lemma 5.2.2) and we are dealing with a maximum flow, we must have either $\sum_{Q\in\mathscr{F}_{2,n}} f_{n,\text{in}}(Q) = \sum_{Q\in\mathscr{F}_{2,n}} c_{n,\text{out}}(Q)$ or $\sum_{Q\in\mathscr{F}_{2,n}} f_{n,\text{in}}(Q) = \sum_{Q\in\mathscr{F}_{2,n}} c_{n,\text{in}}(Q)$. Because the edges between vertices v_Q with $Q \in \mathscr{F}_{2,n}$ and $\ominus$ are not saturated, we must have

$$\sum_{Q\in\mathscr{F}_{2,n}} f_{n,\text{in}}(Q) = \sum_{Q\in\mathscr{F}_{2,n}} c_{n,\text{in}}(Q) = \mu\left(\bigcup_{Q\in\mathscr{F}_{2,n}} Q\right) < \sum_{Q\in\mathscr{F}_{2,n}} c_{n,\text{out}}(Q) = \frac{1}{\Lambda_n} \sum_{Q\in\mathscr{F}_{2,n}} \mu(Q). \tag{5.17}$$

Now, as said before $\mathscr{F}_{2,n} = \mathscr{F}_{n+1}$. When running the while-loop for the (n+1)$^{\text{th}}$ time, we see that we have

$$\Lambda_{n+1} = \frac{\sum_{Q\in\mathscr{F}_{n+1}} \mu(Q)}{\mu\left(\bigcup_{Q\in\mathscr{F}_{n+1}} Q\right)} = \frac{\sum_{Q\in\mathscr{F}_{2,n}} \mu(Q)}{\mu\left(\bigcup_{Q\in\mathscr{F}_{2,n}} Q\right)} > \frac{\sum_{Q\in\mathscr{F}_{2,n}} \mu(Q)}{\frac{1}{\Lambda_n} \sum_{Q\in\mathscr{F}_{2,n}} \mu(Q)} = \Lambda_n, \tag{5.18}$$

which concludes the proof. □

We are now finally ready to prove the main theorem of this chapter:

Theorem 5.3.2. *Let (S, Σ, μ) be a divisible measure space. Let $\mathscr{F}$ be a finite Λ-Carleson collection of sets in Σ such that $\mu\left(\bigcup_{Q\in\mathscr{F}} Q\right)$ is finite. When running Algorithm 2 with $\mathscr{F}$ as input, the result will always be the Carleson constant Λ. To get this result, the while-loop will run at most $N \leq |\mathscr{F}|$ times.*

Proof. In this proof, we again let $\mathscr{F}_n$ with $n \in \{1, ..., N\}$ denote the collection $\mathscr{F}'$ at the beginning of the n^{th} iteration of the while loop, and Λ_n the value of Λ' in the n^{th} iteration of the while loop. Note that Λ_N then denotes the result of the algorithm. We let $V_{1,n}$ and $V_{2,n}$ denote the partitions of $V_n \in G_n$ that is made in line

8, and we set $\mathscr{F}_{n,1} := \{Q \in \mathscr{F}_n : v_Q \in V_{1,n}\}$ and $\mathscr{F}_{2,n} := \{Q \in \mathscr{F}_n : v_Q \in V_{2,n}\}$. Note that we have $\mathscr{F}_{2,n} = \mathscr{F}_{n+1}$ for every $n \in \{1,...,N-1\}$.

We start by proving that the while-loop will run at most $N \leq |\mathscr{F}|$ times. For the n^{th} loop we can either have that $\Lambda_n = \Lambda$ or that $\Lambda_n < \Lambda$ because Λ_n is a lower bound for Λ. By Lemma 5.2.2, if $\Lambda_n < \Lambda$, we have that there is at least one $Q \in \mathscr{F}_n$ such that $Q \in \mathscr{F}_{1,n}$. This means that $|\mathscr{F}_{n+1}| = \left|(\mathscr{F}_n \setminus \mathscr{F}_{1,n})\right| \leq |\mathscr{F}_n| - 1$. This clearly holds for every $n = 1,...,N-1$. We see that $\left|\mathscr{F}'\right|$ decreases by at least 1 for each iteration. We thus have that $\mathscr{F}_n \leq |\mathscr{F}| - (n-1)$. We conclude that we must have $\mathscr{F}_{n+1} = \emptyset$, which is when the algorithm would stop, after at most $|\mathscr{F}|$ iterations.

To prove that this algorithm yields a correct result, we shall first show that the result is not too large, i.e. we show that $\Lambda_N \leq \Lambda$. Note that we have that $\mathscr{F}_n \subseteq \mathscr{F}$ for each $n = 1,...,N$. By Proposition 5.2.1 we then have that

$$\Lambda \geq \frac{\sum_{Q\in\mathscr{F}_n} \mu(Q)}{\mu\left(\bigcup_{Q\in\mathscr{F}_n} Q\right)} = \Lambda_n, \tag{5.19}$$

which proves that each Λ_n is a lower bound for the actual Carleson constant Λ.

We will now prove that the result is not too small, i.e. that $\Lambda_N \geq \Lambda$. Note that by Theorem 2.0.1 this is equivalent to showing that for the estimated sparse constant we have $\Lambda_N^{-1} \leq \Lambda^{-1}$. We prove this by showing that for each $Q \in \mathscr{F}$ we have a ϕ_Q that complies with Definition 3 for $\eta = \Lambda_n^{-1}$. In other words, we have a ϕ_Q that complies with (S2) such that $\int_Q \phi_Q \geq \Lambda_N^{-1}\mu(Q)$. By Definition 3 this means that the actual sparse constant Λ^{-1} cannot be any smaller, because Λ^{-1} must be the largest possible value for which this inequality holds, which means we must have $\Lambda_N^{-1} \leq \Lambda^{-1}$.

Now for $n = 1,...,N$, let

$$\phi_{Q,n} = \sum_{\mathscr{A}\subseteq\mathscr{F}} \mathbb{1}_{A_\mathscr{A}} \frac{f_n(\mathscr{A},Q)}{\mu(A_\mathscr{A})}, \tag{5.20}$$

and let us set

$$\phi_Q = \sum_{n=1}^{N} \mathbb{1}_{\left[\bigcup_{\mathscr{A}\in\mathscr{A}_n} A_\mathscr{A}\right]} \phi_{Q,n}, \tag{5.21}$$

where $\mathscr{A}_n = \{\mathscr{A} \subseteq \mathscr{F}_n : \mathscr{A} \cap \mathscr{F}_{2,n} = \emptyset\}$. Note that the sets $\{\mathscr{A}_n\}_{n\in\{1,...,N\}}$ are pairwise disjoint, because if $\mathscr{A} \in \mathscr{A}_n$ for some $n \in \{1,...,N-1\}$, then $\mathscr{A} \cap \mathscr{F}_{n+1} = \mathscr{A} \cap \mathscr{F}_{2,n} = \emptyset$, meaning $\mathscr{A} \nsubseteq \mathscr{F}_{n+1}$, and therefore $\mathscr{A} \notin \mathscr{A}_{n+1}$. This means that for any $x \in \bigcup_{Q\in\mathscr{F}} Q$ we have that

$$\sum_{Q\in\mathscr{F}} \phi_Q(x) = \sum_{Q\in\mathscr{F}} \sum_{n=1}^{N} \mathbb{1}_{\left[\bigcup_{\mathscr{A}\in\mathscr{A}_n} A_\mathscr{A}\right]} \phi_{Q,n} = \sum_{Q\in\mathscr{A}_x} \phi_{Q,n} \tag{5.22}$$

for some $n \in \{1,...,N-1\}$. We saw in the proof of Theorem 2.0.1 that this expression is bounded by 1 because

$$\sum_{Q\in\mathscr{F}} \phi_Q(x) = \sum_{Q\in\mathscr{A}_x} \phi_{Q,n} \leq \frac{f_{n,\text{out}}(\mathscr{A}_x)}{\mu\left(A_{\mathscr{A}_x}\right)} \leq \frac{c_{n,\text{in}}(\mathscr{A}_x)}{\mu\left(A_{\mathscr{A}_x}\right)} = 1, \tag{5.23}$$

where $\mathscr{A}_x = \{Q \in \mathscr{F} : x \in Q\}$. This means that (S2) is satisfied.

Note that the sets $\{\mathscr{F}_{1,n}\}_{n\in\{1,...,N\}}$ are pairwise disjoint. Indeed, let $n, m \in \{1,...,N-1\}$ be arbitrary. Without loss of generality suppose $n < m$. We have $\mathscr{F}_{1,m} \subseteq \mathscr{F}_m \subseteq \mathscr{F}_{n+1} = \mathscr{F}_n \setminus \mathscr{F}_{1,n}$. These relations are due to line 9 in Algorithm 2. Now let $Q \in \mathscr{F}$ be arbitrary. Then $Q \in \mathscr{F}_{1,n_Q}$ for exactly one $n_Q \in \{1,...,N\}$. Observe that

$$\int_Q \phi_Q = \int_Q \sum_{n=1}^{N} \mathbb{1}_{\left[\bigcup_{\mathscr{A}\in\mathscr{A}_n} A_\mathscr{A}\right]} \phi_{Q,n}. \tag{5.24}$$

Note there is no $x \in Q$ such that $x \in \bigcup_{\mathscr{A}\in\mathscr{A}_{n^-}} A_\mathscr{A}$ for an $n^- < n_Q$. Because if there were such an x, we would have $x \in \mathscr{A}^-$ for an $\mathscr{A}^- \in \mathscr{A}_{n^-}$. By Definition 4 this is only possible if $Q \in \mathscr{A}^-$, but by definition of $\mathscr{A}_{n^-}$ we should have $\mathscr{A}^- \cap \mathscr{F}_{n^-+1} = \mathscr{A}^- \cap \mathscr{F}_{n^-,2} = \emptyset$, which would mean $Q \notin \mathscr{F}_{n_Q} \subseteq \mathscr{F}_{n^-+1}$, where we have $\mathscr{F}_{n_Q} \subseteq \mathscr{F}_{n^-+1}$ because $n_Q \geq n^- + 1$. This is a contradiction. We could have an $x \in Q$, such that $x \in A_\mathscr{A}$ with $\mathscr{A} \in \mathscr{A}_{n^+}$ for an $n^+ > n_Q$. However, for such an $\mathscr{A}$ we must have $f_{n^+}(A_\mathscr{A}, Q) = 0$, because $Q \notin \mathscr{F}_n \subseteq \mathscr{F}_{n_Q+1}$ as $Q \in \mathscr{F}_{n_Q} \setminus \mathscr{F}_{n_Q+1}$, which means $\phi_{Q,n^+} = 0$. We conclude that expression (5.24) is equivalent to

$$\int_Q \phi_Q = \int_Q \mathbb{1}_{\left[\bigcup_{\mathscr{A}\in\mathscr{A}_{n_Q}} A_\mathscr{A}\right]} \phi_{Q,n_Q} \tag{5.25}$$

Now note that by Lemma 5.2.2 we have that for an $\mathscr{A} \subseteq \mathscr{F}_{n_Q}$, $f_{n_Q}(\mathscr{A}, Q) > 0$ only if $\mathscr{A} \in \mathscr{A}_{n_Q}$, because if $\mathscr{A} \notin \mathscr{A}_{n_Q}$, we have a $Q_2 \in \mathscr{F}_{n_Q,2}$ for which there exists an edge $(\mathscr{A}, Q_2)$ in G_{n_Q}, which is only possible if $f_{n_Q}(\mathscr{A}, Q) = 0$ because $Q \in \mathscr{F}_{n_Q,1}$. This means we have

$$\phi_{Q,n_Q}(x) = \sum_{\mathscr{A} \subseteq \mathscr{F}} \mathbb{1}_{A_\mathscr{A}} \frac{f_{n_Q}(\mathscr{A}, Q)}{\mu(A_\mathscr{A})} = 0 \tag{5.26}$$

for any $x \in Q \setminus \left(\bigcup_{\mathscr{A} \in \mathscr{A}_{n_Q}} A_\mathscr{A} \right)$. We conclude that we have

$$\int_Q \phi_Q = \int_Q \mathbb{1}_{\left[\bigcup_{\mathscr{A} \in \mathscr{A}_{n_Q}} A_\mathscr{A}\right]} \phi_{Q,n_Q} = \int_Q \phi_{Q,n_Q} = \int_Q \sum_{\mathscr{A} \subseteq \mathscr{F}} \mathbb{1}_{A_\mathscr{A}} \frac{f_{n_Q}(\mathscr{A}, Q)}{\mu(A_\mathscr{A})} = \int_Q \mathbb{1}_Q \sum_{\mathscr{A} \subseteq \mathscr{F}} \mathbb{1}_{A_\mathscr{A}} \frac{f_{n_Q}(\mathscr{A}, Q)}{\mu(A_\mathscr{A})} = f_{n_Q,\text{in}}(Q), \tag{5.27}$$

where the last equality holds by the same reasoning as in in (2.18), (2.19) and (2.20) in the proof of Theorem 2.0.1. Now, because f_{n_Q} saturates the edge $(Q, \ominus)$, in G_{n_Q}, we have

$$f_{n_Q,\text{in}}(Q) = c_{n_Q,\text{out}}(Q) = \Lambda_{n_Q}^{-1}(Q) \geq \Lambda_N^{-1}(Q), \tag{5.28}$$

because $\Lambda_{n_Q} \leq \Lambda_N$ by Lemma 5.3.1. This means that we have

$$\int_Q \phi_Q = \int_Q \phi_{Q,n} \geq \Lambda_N^{-1}(Q) \tag{5.29}$$

and that (S1) with $\eta = \Lambda^{-1}$ is satisfied. □

5.4. The time complexity of the Algorithm

It will be again interesting to check how efficiently we can discover what the Carleson constant is. Note that we can also discover the value of the constant with the help of Definition 2 using the following algorithm.

Algorithm 3: An algorithm to approximate the Carleson constant of a finite collection $\mathscr{F}$ with the help of the definition.

Data: A finite Carleson collection $\mathscr{F}$.
1 **begin**
2   Set $\Lambda = 1$.
3 **end**
4 **for** $\mathscr{A} \subseteq \mathscr{F}$ **do**
5   Set $\Lambda' = \dfrac{\sum_{Q \in \mathscr{A}} \mu(Q)}{\mu\left(\bigcup_{Q \in \mathscr{A}} Q\right)}$.
6   **if** $\Lambda' > \Lambda$ **then**
7   Set $\Lambda = \Lambda'$.
8   **end**
9 **end**
Result: The Carleson constant Λ.

We would like for Algorithm 2 to be more efficient than this one. To see whether this is the case, we first need to know how efficient Algorithm 3 is, which follows from the following proposition.

Proposition 5.4.1. *Let (S, Σ, μ) be a divisible measure space such that $\mu\left(\bigcup_{Q \in \mathscr{F}} Q\right)$ is finite. Let $\mathscr{F}$ be a finite Λ-Carleson collection of sets in Σ. Then Algorithm 3 is able to find the Carleson constant Λ of $\mathscr{F}$ in $O(P(\mathscr{F})) = O\left(2^{|\mathscr{F}|}\right)$ time.*

Proof. The initialisation of Algorithm 3 in line 2 takes $O(1)$ time, as there is only one variable to be initialised. The actions that are performed within the for-loop between line 5 and 7 also take $O(1)$ time because one inequality is checked en then either one or two variables are set. The time complexity of the algorithm now only depends on how many times the for-loop will run. The for-loop will run one time for each $\mathscr{A} \subseteq \mathscr{F}$, meaning it will run $|P(\mathscr{F})|$ times. We conclude that the time complexity of the algorithm equals $O(P(\mathscr{F})) = O\left(2^{|\mathscr{F}|}\right)$. □

The time complexity of Algorithm 2 follows almost directly from Theorem 5.3.2:

Theorem 5.4.2. *Let (S, Σ, μ) be a divisible measure space such that $\mu\left(\bigcup_{Q \in \mathscr{F}} Q\right)$ is finite. Let $\mathscr{F}$ be a finite Λ-Carleson collection of sets in Σ. The time complexity of Algorithm 2 with $\mathscr{F}$ as input is at most $O\left(|\mathscr{F}||V|^3)\right) \leq O\left(|V|^4\right)$.*

Proof. The initialisation in line 2 takes $O(1)$ time, as there is only one variable to be set. By Theorem 5.3.2, the while-loop in lines 4 to 9 runs at most $|\mathscr{F}|$ times. We will now check how long it takes to run one iteration of this while-loop.

In line 5 we set one variable, which takes $O(1)$ time. Then in lines 6 to 7 we again represent a collection $\mathscr{F}$' as a graph and then find the maximum flow in this graph. In the proof of Theorem 3.3.1, we saw that this takes $O\left(|V|^3\right)$ time. In line 8 we partition the graph into V_1 and V_2 according to the maximum flow. This means that for each vertex $v \in V \setminus \{\oplus, \ominus\} = (V2) \cup (V3)$, we need to check whether there is an unsaturated path form this vertex to $\ominus$. For each $v \in (V3)$ we only need to check one path, as these only have one outgoing edge. So, this will take $O(|(V3)|)$, which is $O(|V|)$ by the proof of Theorem 3.3.1 time. For each $v \in (V2)$ we need to check at most $|\mathscr{F}|$ paths, as this is the amount of sets that could be in any $\mathscr{A}$. We can thus check all the paths from a vertex in $|(V2)|$ to $\ominus$ in at most $O(|\mathscr{F}||(V2)|) = O\left(|V|^2\right)$ time. We then finally have line 9, which concerns defining a variable. This can happen in $O(1)$ time. We conclude that one iteration of the while loop takes at most

$$O(1) + O\left(|V|^3\right) + O(|V|) + O\left(|V|^2\right) = O\left(|V|^3\right) \tag{5.30}$$

time. All the while loops together therefore take at most

$$O\left(|\mathscr{F}||V|^3\right) \leq O\left(|V|^4\right) \tag{5.31}$$

time. □

We see that our algorithm has a better time complexity than Algorithm 2 if $|V|$ is not too large. We can put the same restriction on $\mathscr{F}$ as in Lemma 3.3.3 to make sure the algorithm finishes in polynomial time. This is stated in the following Corollary.

Corollary 5.4.3. *Let $\mathscr{F}$ be a finite collection of dyadic cubes that is Λ-Carleson. If we run Algorithm 2 with $\mathscr{F}$ as input, the algorithm will finish in at most $O\left(|\mathscr{F}|^4\right)$ time.*

Proof. By Theorem 5.4.2 we have that Algorithm 2 runs in at most $O\left(|V|^4\right)$ time. The proof of Corollary 3.3.4 shows that for a dyadic collection $\mathscr{F}$, we have $|V| = |F|$. We conclude that the complexity of Algorithm 2 for $\mathscr{F}$ is at most $O\left(|\mathscr{F}|^4\right)$ □

6
Discussion

In this bachelor's thesis I have succeeded in proving the equivalence between the sparse and Carleson condition for a finite Λ-Carleson collection $\mathscr{F}$, and generalised this proof for countably infinite collections with limited overlap. I have also managed to construct an algorithm that can approximate the Carleson constant for a finite collection $\mathscr{F}$ for which this constant is unknown, and an algorithm that can find the sets $\{E_Q\}_{Q\in\mathscr{F}}$ if $\mathscr{F}$ is finite and Λ-Carleson with respect to a divisible measure μ. Both of these algorithms finish in polynomial time for a Λ-Carleson collection $\mathscr{F}$ of dyadic cubes.

The biggest obstacle in writing this thesis was the construction of Algorithm 1 which is able to find the sets $\{E_Q\}_{Q\in\mathscr{F}}$. Originally, my supervisor and I thought we could divide the space in $\bigcup_{Q\in\mathscr{F}} Q$ amongst the set $\{E_Q\}_{Q\in\mathscr{F}}$ by finding the maximum flow in a graph which, asides from the source and the sink vertex, only had vertices for each set $Q \in \mathscr{F}$, and not for each subcollection $\mathscr{A} \subseteq \mathscr{F}$ with $\mu(A_{\mathscr{A}}) > 0$. We quickly discovered that with such a method you run into trouble if there are areas where more than two sets in $\mathscr{F}$ overlap. Luckily we came up with the solution that I have described in this thesis, but for this solution the graph can have many more vertices than the amount of sets in $\mathscr{F}$. Because of this, the time complexity of this new algorithm is much larger than that of the original one, seeing that the time complexity depends on how long it takes to find the maximum flow in the graph, which for a large part depends on the amount of vertices.

We have tried to come up with other, more efficient ways to transform $\mathscr{F}$ into a graph, but sadly without success. For this reason, I expect it might be impossible for an algorithm to find the sets $\{E_Q\}_{Q\in\mathscr{F}}$ satisfying the Λ^{-1}-sparse condition in polynomial time for a general collection $\mathscr{F}$. I have not been able to prove this, but this would be an interesting topic for further research. It would also be interesting to try and prove that the amount of vertices in the graph is limited for any other type of collections except for dyadic cubes, meaning that the algorithm would be more efficient for this category as well.

Bibliography

[1] Karen Aardal, Leo van Iersel, and Remie Janssen. *Lecture notes AM2020/IN4344 Optimization*. Delft University of Technology, Faculty of Electical Engineering, Mathematics and Computer Science, September 2023.

[2] Ron Aharoni, Eli Berger, Agelos Georgakopoulos, Amitai Perlstein, and Philipp Sprüssel. The max-flow min-cut theorem for countable networks. *Journal of Combinatorial Theory, Series B*, 101(1):1–17, 2011. doi: https://doi.org/10.1016/j.jctb.2010.08.002.

[3] Alexander Barron. *Sparse bounds in harmonic analysis and semiperiodic estimates*. PhD thesis, Brown University, 2019.

[4] Georg Cantor. Über eine elementare Frage der Mannigfaltigkeitslehre. *Jahresbreicht der Deutschen Mathematiker-Vereinigung*, 1:75–78, 1891. Translation by James R. Meyer at https://www.jamesrmeyer.com/infinite/cantors-original-1891-proof.

[5] Lennart Carleson. *A counter example for measures bounded on H_p for the bi-disc*. Report No. 7 - 1974. Institut Mittag-Leffler, 1974.

[6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 24. Maximum flow. In *Introduction to Algorithms, Fourth Edition.*, pages 670–693. The MIT Press, 2022.

[7] George Bernard Dantzig and D. R. Fulkerson. *On the Max Flow Min Cut Theorem of Networks*. RAND Corporation, Santa Monica, CA, 1955.

[8] Abhijit Dasgupta. *Set Theory: With an Introduction to Real Point Sets*. SpringerLink : Bücher. Springer New York, 2013. doi: https://doi.org/10.1007/978-1-4614-8854-5.

[9] Yefim Dinitz. Dinitz' algorithm: the original version and Even's version. In *Theoretical Computer Science: Essays in Memory of Shimon Even*, pages 218–240. Springer Berlin Heidelberg, 2006. doi: 10.1007/11685654_10.

[10] Leonard E. Dor. On projections in L_1. *Annals of Mathematics*, 102(3):463–474, 1975. doi: https://doi.org/10.2307/1971039.

[11] Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264, 1972. doi: https://doi.org/10.1145/321694.321699.

[12] Jeff Erickson. *Algorithms*. Jeff Erickson, 2019.

[13] Jon Folkman and Delbert R. Fulkerson. Flows in infinite graphs. *Journal of Combinatorial Theory*, 8(1): 30–44, 1970. doi: https://doi.org/10.1016/S0021-9800(70)80006-0.

[14] Timo S. Hanninen. Equivalence of sparse and Carleson coefficients for general sets. *arXiv eprint 1709.10457*, 2017. doi: https://doi.org/10.48550/arXiv.1709.10457.

[15] Tuomas P. Hytönen. The sharp weighted bound for general calderon-zygmund operators. *arXiv eprint 1007.4330*, 2010. doi: https://doi.org/10.48550/arXiv.1007.4330.

[16] Valerie King, Satish Rao, and Rorbert Tarjan. A faster deterministic maximum flow algorithm. *Journal of Algorithms*, 17(3):447–474, 1994. doi: https://doi.org/10.1006/jagm.1994.1044.

[17] Dennis König. *Theorie der endlichen und unendlichen Graphen*. AMS Chelsea Publishing Series. American Mathematical Society, 2001. doi: https://doi.org/10.1007/BF01792729.

[18] Andrei K. Lerner. A simple proof of the a_2 conjecture. *arXiv eprint 1202.2824*, 2012. doi: https://doi.org/10.48550/arXiv.1202.2824.

[19] Andrei K. Lerner. On pointwise estimates involving sparse operators. *arXiv eprint 1512.07247*, 2015. doi: https://doi.org/10.48550/arXiv.1512.07247.

[20] Andrei K. Lerner and Fedor Nazarov. Intuitive dyadic calculus: the basics. *arXiv eprint 1508.05639*, 2015. doi: https://doi.org/10.48550/arXiv.1508.05639.

[21] Andreas Lochbihler. A mechanized proof of the max-flow min-cut theorem for countable networks. In *12th International Conference on Interactive Theorem Proving (ITP 2021)*, volume 193 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 25:1–25:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi: https://doi.org/10.4230/LIPIcs.ITP.2021.25.

[22] James B. Orlin and Xiao-Yue Gong. A fast max flow algorithm. *arXiv eprint 1910.04848*, 2019. doi: https://doi.org/10.48550/arXiv.1910.04848.

[23] María Cristina Pereyra. Dyadic harmonic analysis and weighted inequalities: the sparse revolution. *arXiv eprint 1812.00850*, 2018. doi: https://doi.org/10.48550/arXiv.1812.00850.

[24] Natarajan Raja. Yet another proof of Cantor's theorem. *Dimensions of Logical Concepts, Coleçao CLE*, 54: 209–217, 2009.

[25] Guillermo Rey. Greedy approximation algorithms for sparse collections. *arXiv eprint 2202.10267*, 2022. doi: https://doi.org/10.48550/arXiv.2202.10267.

[26] Alexander Schrijver. On the history of the transportation and maximum flow problems. *Mathematical Programming*, 91(3):437–445, 2002. doi: 10.1007/s101070100259.

[27] Terence Tao. Dyadic product H^1, BMO, and Carleson's counterexample. Preprint. URL `http://www.math.ucla.edu/~tao/preprints/Expository/product.dvi`.

[28] Igor E. Verbitsky. Imbedding and multiplier theorems for discrete Littlewood-Paley spaces. *Pacific Journal of Mathematics*, 176(2):529 – 556, 1996.