Discrete stratified Morse theory for 2-dimensional simplicial complexes

Jeferson León Zapata Nieto

Universidad Nacional de Colombia
Facultad de Ciencias, Escuela de Matemáticas
Medellín, Colombia
2020

Teoría de Morse discreta para complejos simpliciales 2-dimensionales

Jeferson León Zapata Nieto

Universidad Nacional de Colombia
Facultad de Ciencias, Escuela de Matemáticas
Medellín, Colombia
2020

Discrete stratified Morse theory for 2-dimensional simplicial complexes

THESIS FOR THE DEGREE OF MASTER IN SCIENCES

Jeferson Zapata

Universidad Nacional de Colombia–Sede Medellín

Academic advisor:

Edgar Ramos[1]

[1]Escuela de Matemáticas
Facultad de Ciencias
Universidad Nacional de Colombia
Medellín, Colombia

Submitted: 30/04/2020

Teoría de Morse discreta para complejos simpliciales 2-dimensionales

TESIS PARA EL TÍTULO DE MAGÍSTER EN CIENCIAS

Jeferson Zapata

Universidad Nacional de Colombia–Sede Medellín

Director:

Edgar Ramos[1]

[1]Escuela de Matemáticas
Facultad de Ciencias
Universidad Nacional de Colombia
Medellín, Colombia

Enviado: 30/04/2020

Discrete stratified Morse theory for 2-dimensional simplicial complexes

Short abstract:

The main objective of this thesis is to analyze a generalization of Morse's theory in the case of stratified spaces. The content is divided into three main parts. In the first part we present the background of the classical Morse theory, the discrete Morse theory of Forman and the stratification of a certain type of topological spaces. In the second part we describe the basic concepts in classical complexity and parameterized complexity. In the last part we analyze two main topics: Lewiner's algorithm for 2-simplicial complexes and the analysis of the complexity of the problem of finding Morse functions in the case of parameterized complexity.

Keywords: Simplicial complex, Morse function, complexity, acyclic matching.

Teoría de Morse discreta para complejos simpliciales 2-dimensionales

Short abstract:
El objetivo principal de esta tesis es analizar una generalización de la teoría de Morse en el caso de espacios estratificados. El contenido se divide en tres partes principales. En la primera parte presentamos los antecedentes de la teoría de Morse clásica, la teoría de Morse discreta de Forman y la estratificación de un cierto tipo de espacios topológicos. En la segunda parte describimos los conceptos básicos en complejidad clásica y complejidad parametrizada. En la última parte analizamos dos temas principales: el algoritmo de Lewiner para complejos 2-simpliciales y el análisis de la complejidad del problema de encontrar funciones de Morse en el caso de la complejidad parametrizada.

Keywords: Complejo simplicial, función de Morse, complejidad parametrizada, apareamiento acíclico.

Dedicated to everybody.

CONTENTS

1 Smooth and Discrete Morse Theory **1**
1.1 SMT: Smooth Morse Theory 2
1.2 DMT: Discrete Morse Theory 9

2 Stratification of a topological space **15**
2.1 Local cohomology of CW complexes 18
2.2 Structure of the strata 22
2.3 Description and complexity of the algorithm 23

3 Parameterized Complexity **25**
3.1 About Classical Complexity 26
3.2 About Parameterized Complexity 33

4 Optimal Morse Matchings **43**
4.1 Morse matchings 44
4.2 Lewiner's algorithm 46
4.3 Hardness of Optimal Morse matchings 50

5 Parameterized complexity of DMT **55**

Bibliography **65**

INTRODUCTION

In the first half of the 20th century Marston Morse developed the fundamental ideas of what is now known as Morse theory ([Bro30]). Given a function on a manifold, this tool allows a description of its topology, as well as its homology groups and many other important characteristics. Inspired by this work, Robin Forman formulated the discrete Morse theory ([For98]), a combinatorial version of Morse theory intended for application in discrete structures such as simplicial complexes or more generally in CW-complexes. The definition, although quite simple, is very powerful and as a discrete analog, allows us to use the concepts of Morse theory in a more computational sense. Specifically, the problem of finding a discrete gradient can be reduced to computing an acyclic mathing, which is a well known problem and allows us to use graph theory to solve the above problem.

In the case of combinatorial 2-manifolds Lewiner provides a linear algorithm to build an optimal discrete Morse function in the sense that it has the minimum number of critical cells ([LLT03a]). Also in the same paper, Lewiner offers an extension of its algorithm to the case of general 2-dimensional simplicial complexes but without offering any guarantee of optimality in this case. It is therefore interesting to ask whether applying the Lewiner algorithm to each of the strata of a stratified Morse function can be extended to a global Morse function in the entire simplicial complex with the minimum number of critical points.

In this report we study the problem of finding a Morse matching from a computational point of view. In particular, we will concentrate in the problem of computing a Morse matching for 2-dimensional simplicial manifolds and for general 2-dimensional simplicial complexes. We are mainly interested in the complexity of the aforementioned problem and the features that make this problem inherently difficult as it has been proven to be inside the class of NP-complete, even for the case of having a pure 2-dimensional simplicial complex. In this light, we analyze the problem replacing the classical concept of complexity and using parameterized complexity.

We assume some familiarity with complexity theory, algebraic topology and basic algorithmic concepts. However, we have included some basics in classic complexity theory in order to introduce the concept of *Parameterized Complexity*.

This work is organized as follows:

Chapter 1 gives a short review of Morse theory from the basic definitions in both the discrete version and the classical version. The most important results in this section are the Morse lemma that tells us how we can describe locally the properties of what will be a critical point. Next, we talk about the concept of flow in a manifold to finally present the main result in the handling decomposition. In the discrete framework, we start with the basic definitions to quickly move to the concept of discrete vector fields and conclude with the relationship between collapsibility and discrete Morse gradients.

In the second chapter, the main concept is the stratification of a space. First we provide an example of what we are pursuing and the possible complications in the construction of an stratification. From the algorithm presented in [Nan19], we give the necessary tools for the construction of a stratification for a given cell complex. In our particular case, we focus on capturing the components of the two-dimensional strata. Finally, we give a description of the complexity of the algorithm that will be necessary for the later analysis in the last chapter.

Chapter three is devoted to the study of computational complexity theory. We start with some concepts in classic complexity theory with some examples of the complexity classes and Cook's theorem. For the second part of this chapter we introduce the classes in parameterized complexity and the parameterized version of Cook's theorem.

In the fourth chapter, we describe the problem of computing an acyclic matching as the combinatorial equivalent of the Morse gradient. Then we present the algorithm [LLT03a] to find optimal Morse functions for the case of 2-dimensional manifold. After that, we analyze the difficulty of this problem for the case of non manifolds.

In the last chapter, we study the problem in the parameterized case using the relation of collapsibility and Morse gradient presented in the previous chapters. Finally, we analyze the same problem using a different parameter given by the stratification of the 2-dimensional complex.

1

Smooth and Discrete Morse Theory

"There is a fundamental error in separating the parts from the whole, the mistake of atomizing what should not be atomized. Unity and complementarity constitute reality." Heisenberg

Contents

1.1 SMT: Smooth Morse Theory 2
1.2 DMT: Discrete Morse Theory 9

1.1 SMT: Smooth Morse Theory

In this section we focus on the main concepts in smooth Morse theory. Mostly we follow the results in [Mil73] and [MHSS02]. Before we start with the theory, we introduce some useful notation. For a smooth manifold M, we will write M^m to denote that M has dimension m. All the maps will be smooth functions. Given a smooth map $f : M \to N$ we will denote its differential as $df : TM \to TN$, where TM is the tangent bundle and the differential at a point $p \in M$ will be denoted by $d_p f : T_p M \to T_{f(p)} N$.

By definition, a *vector field* X on M is an assignment $X : M \to TM$ that assigns to each point $p \in M$ a tangent vector $X_p \in T_p M$. Here, a tangent vector $X_p \in T_p M$ is a considered linear map $X_p : C^\infty(M) \to \mathbb{R}$ satisfying the Leibniz's rule $X_p(fg) = X_p(f)g(p) + f(p)X_p(g)$. So any vector field maps any $f \in C^\infty(M)$ to a function Xf on M defined by $Xf(p) = X_p f$. We denote by $XY(f)$ the composition of vectors fields X and Y such that at point p it satisfies $(XY)_p(f) = X_p(Y(f))$.

At each point p the commutator $[X, Y]_p := X_p(Yf) - Y_p(Xf)$ is the Lie Bracket of two vector fields X and Y.

Definition 1.1.1 (Critical point) *Given a smooth function $f : M^m \longrightarrow N^n$, a point $p \in M$ is critical if its differential $d_p f$ does not have full rank, i.e.* $\operatorname{Im} d_p f < \min(m, n)$.

Since we are mainly interested in real valued functions, we have that a point p is critical if and only if $d_p f = 0$.

Definition 1.1.2 *The Hessian of $f : M \longrightarrow \mathbb{R}$ at a critical point x is the symmetric bilinear map*

$$\begin{aligned} H_{f,p} : T_p M \times T_p M &\longrightarrow \mathbb{R} \\ (x, y) &\longmapsto X_p(Y(f)) \end{aligned}$$

where X and Y are the vector fields such that $X_p = x$ and $Y_p = y$.

The Hessian will help us to describe the behavior of the function f in a neighborhood of the point p. In particular, it will tell us how "flat" is the graph of the function around a neighborhood of p, or more specifically if its Gaussian curvature is zero. In Morse theory the Hessian is needed in order to define the index at a given point and avoid some degenerate cases. Before this definition we need to show some properties of the Hessian.

We recall here that the directional derivative $X(f)$ of f along a vector field X is given by $(Xf)(p) = X_p(f) = d_p f(X_p)$. the following Lemma give a proof of the well definition of the bracket function $[\bullet,\bullet]_p : X_p \times X_p \to X_p$.

Lemma 1.1.3 *$[X, Y]_p(f)$ is a tangent vector at point p.*

Proof: From the definition we can see $[X, Y]_p$ is a linear map. Let's verify that this commutator satisfies the Leibniz law.

$$\begin{aligned}[X,Y]_p(fg) &= (XY)_p(fg) - (YX)_p(fg)\\ &= X_p(Y(fg)) - Y_p(X(fg))\\ &= X_p((Yf)g + f(Yg)) - Y_p((Xf)g + f(Xg)\end{aligned}$$

and expanding these terms we have

$$X_p((Yf)g + f(Yg)) = X(Yf)(p)g(p) + Yf(p)Xg(p) + Xf(p)Yg(p) + f(p)X(Yg)(p)$$

and

$$-Y_p((Xf)g + f(Xg) = Y(Xf)(p)g(p) - Xf(p)Yg(p) - Yf(p)Xg(p) - f(p)Y(Xg)(p)$$

Regrouping the terms we get

$$[X,Y]_p(fg) = [X,Y]_p(f)g(p) + f(p)[X,Y]_p(g).$$

$\diamondsuit$

Lemma 1.1.4 *$H_{f,p}$ is a well-defined symmetric bilinear map.*

Proof: Let $f : M \to \mathbb{R}$ be a smooth function, X, Y vector fields on M and p a critical point of f. On the first hand, suppose we have vector fields $\tilde{X}, \tilde{Y}$ such that $X_p = \tilde{X}$ and $Y_p = \tilde{Y}$. So we have $(X - \tilde{X})_p f = 0$. In particular,

$$\begin{aligned}(XY)_p(f) - (\tilde{X}Y)_p(f) &= (X - \tilde{X})_p Yf = 0\\ (XY)_p(f) - (X\tilde{Y})_p(f) &= X_p(Y - \tilde{Y})f = 0.\end{aligned}$$

Finally, we have $X_p(Yf) - Y_p(Xf) = [X,Y]_p f = d_p f([X,Y]_p) = 0$, as p is a critical point. $\diamondsuit$

A consequence of the previous lemma is that we can see the Hessian as a symmetric matrix, thereby we define the *index of the Hessian* as the number of negative eigenvalues of the matrix associated to it. A critical point p of a smooth function $f : M \to \mathbb{R}$ is *non-degenerate* if the null space of $H_{f,p}$ is 0. If it has positive dimension then p is *degenerate*. With all of this, now we can define the class of functions we are interested in.

Definition 1.1.5 (Morse function) *A Morse function is a smooth function $f : M \to \mathbb{R}$ so that each critical point $p \in M$ is non-degenerate.*

The set of critical points of a Morse function f will be denoted as $\mathrm{Cr}\, f$, and the set of critical points of index i will be denoted as $\mathrm{Cr}_i\, f$.

Choosing local coordinates $(x^1, \ldots, x^n)$ on a chart $U \subset M$, a point $p \in M$ is critical for f if $(\partial f / \partial x^i)(p) = 0$ for all i and it is non-degenerate if the matrix of second derivatives

$$(H_{f,p})_{ij} = \frac{\partial^2 f}{\partial x^i \partial x^j}.$$

is non-singular. Since our matrix is symmetric, it has an eigenbasis and we can find coordinates such that the Hessian is a diagonal matrix with nonzero values in its diagonal. With this idea we can describe the neighborhood of the point p in a convenient way. This is stated in the next two results.

Lemma 1.1.6 *Let f be a smooth function on a convex neighborhood of the origin in $\mathbb{R}^n$ with $f(0) = 0$, then there are smooth functions g_i such that*

$$f(x) = f(x_1, \ldots, x_n) = \sum_{i=1}^{n} x^i g_i(x).$$

Proof: We can rewrite $f(x)$ as follows,

$$f(x) = f(x) - f(0) = \int_0^1 \frac{df(tx)}{dt} dt$$

Now applying the chain rule

$$f(x) = \int_0^1 \sum_{i=0}^{n} x^i \frac{\partial f}{\partial x^i}(x) dt = \sum_{i=0}^{n} x^i \int_0^1 \frac{\partial f}{\partial x^i}(x) dt.$$

$\diamondsuit$

Lemma 1.1.7 (Morse lemma) *For any non-degenerate critical point p of a smooth function $f : M \to \mathbb{R}$ there exists a neighborhood U of p with local coordinates*

$(x^1,\dots,x^n)$ centered at p so that in these coordinates f is given by

$$f(x)=f(p)-\sum_{i=1}^{\lambda(p)}(x^i)^2+\sum_{i=\lambda(p)+1}^{n}(x^i)^2.$$

Proof: First we choose some coordinates with center at p and consider the function $\tilde{f}=f-f(p)$. Using the previous lemma, we have

$$f(x^1,\dots,x^n)=\begin{cases}0 & \text{for } x=0\\ \sum_{i=1}^n x^i g_i(x) & \text{for } x\neq 0.\end{cases}$$

Since 0 is a critical point for $\tilde{f}$ then $g_i(0)=(\partial f)(\partial x^i)(0)=0$. Therefore, we can write g_i as a combination of some functions h_{ij} as follows

$$g_i(x)=\sum_{j=1}^{n}x^j h_{ij}(x)$$

and such that $h_{ij}(0)=(\partial^2 f)(\partial x^i\partial x^j)$. Now we proceed by induction and assume we have

$$\tilde{f}=\pm(x^1)^2\pm\cdots\pm(x^{r-1})^2+\sum_{i,j\geq r}x^i x^j h_{ij}(x).$$

Since the Hessian is non-degenerate, we can find entries $h_{k\ell}(0)\neq 0$ with $k,\ell\geq 0$. Now we can define coordinates u^i such that

$$\begin{cases}u^i=x^i \text{ for } i\neq k,\ell\\ u^i=x^k+x^\ell \text{ for } i=k\\ u^i=x^k-x^\ell \text{ for } i=\ell.\end{cases}$$

Redefine h_{ij} as $\tilde{h}_{ij}$. These coordinates satisfies $4x^k x^\ell=(y^k)^2-(y^\ell)^2$, and $\tilde{h}_{kk}(0),\tilde{h}_{\ell\ell}(0)$ are non-zero. Assume $h_{rr}\neq 0$ (otherwise we can permutate the indices as needed). Let $g(u)=\sqrt{|\tilde{h}_{rr}(u)|}$, define new coordinates v^i as follows,

$$\begin{cases}v^i=u^i \text{ for } i\neq r\\ v^r=g(u)\Big(u^r+\sum_{i>r}u^i\frac{\tilde{h}_{ir}(u)}{\tilde{h}_{rr}(u)}\Big) \text{ for } i=r.\end{cases}$$

Then we have

$$(v^r)^2=|\tilde{h}_{rr}|(u^r)^2+\frac{\tilde{h_{rr}}}{\tilde{h}_{rr}^2}\left(\sum_{i>r}u^i\tilde{h}_{ir}\right)^2+2\frac{\tilde{h}_{rr}}{\tilde{h}_{rr}}\sum_{i>r}u^r u^i\tilde{h}_{ir}.$$

And therefore we have

$$\tilde{f}(v) = \left(\sum_{i<r} \pm (v^i)^2\right) \pm (v^r)^2 + \sum_{i,j>r} v^i v^j \tilde{h}_{ij} - \frac{1}{h_{rr}} \left(\sum_{i>r} u^i \tilde{h}_{ir}\right)^2.$$

Thus, inductively we can apply this diagonalisation for all the coordinates v^i getting

$$f(v) = f(p) + \sum \pm (v^i)^2.$$

◇

The following results are not needed in our analysis, but they are helpful to give us some geometrical intuition about the relationship between the topology of the manifold and its critical points. Detailed proofs can be found in [Nic11].

Definition 1.1.8 *A function $f : M \to \mathbb{R}$ is exhaustive if for all $a \in \mathbb{R}$ the set $f^{-1}(-\infty, a]$ is compact.*

Lemma 1.1.9 *Any smooth manifold admits exhaustive Morse functions.*

Often we use the concept of *gradient* to study some properties of Morse functions, but this implies the choice of a metric. Thus, we can fix a metric and use the gradient defined by such metric or, in a more general way, we can make use of so-called 'gradient-like' vector fields.

Given a vector field X on M, for any point $p \in M$ we have a curve $\gamma(t)$ given by the differential equation

$$\frac{d\gamma}{dt}(t) = X_{\gamma(t)} \quad \text{s.t.} \quad \gamma(0) = p.$$

This is the general initial value problem and the next proposition asserts this curve effectively exists.

Proposition 1.1.10 *Let X be a vector field and $x \in M$.Then there exists $\delta > 0$, a neighborhood U of x in M, and a unique smooth map $\Psi : U \times (-\delta, \delta) \to M$ which satisfies*

$$\frac{\partial}{\partial t}\Psi(y,t) = X_{\Psi(y,t)}$$
$$\Psi(y,0) = y$$

for all $y \in U$ and $t \in (-\delta, \delta)$. For each $t \in (-\delta, \delta)$, the map $\Psi_t : U \to M$ defined by $\Psi_t(y) = \Psi(y,t)$ is a local diffeomorphism, and

$$\Psi_t \circ \Psi_s = \Psi_{t+s}$$

whenever both sides are defined.

The previous proposition incorporates three aspects:

- The solutions exist;
- they are unique;
- the solutions depend in a smooth way on the initial point $p \in M$.

The maps Ψ_t are called the (local) flow of X for time t and they form a "local group":

Definition 1.1.11 *A flow is a smooth group action of* $(\mathbb{R}, +)$ *on the smooth manifold* M.

With these maps we can define a vector field by

$$X_p(f) := \lim_{h \to 0} \frac{f(\Psi_h(p)) - f(p)}{h}.$$

The curves that are solutions to $\gamma'(t) = X_p(f)$ where X generates the flow are called *flow lines.*

Proposition 1.1.12 *Given a vector field* $X \in M$ *vanishing outside a compact set* $K \subset M$*, then there is a unique flow generated by* X.

Proof: Taking a open cover of our set K and using the previous proposition, we can construct a flow pasting neighborhoods of a radius suitably small. $\diamondsuit$

We will need the next result that follows immediately from the previous Proposition.

Corollary 1.1.13 *Given a compact manifold* M*, then any vector field on* M *produces a flow.*

Definition 1.1.14 *A gradient-like vector field for a Morse function* $f : M^m \to \mathbb{R}$ *is a vector field* X *so that* $X(f) > 0$ *and there exists a neighborhood* U_p *of a critical point* p *such that there are coordinates* (x^i) *in* U_p *satisfying*

$$X(f) = -2 \sum_{i=1}^{\lambda(p)} x^i \frac{\partial}{\partial x^i} + 2 \sum_{i=\lambda(p)+1}^{m} x^i \frac{\partial}{\partial x^i}.$$

Gradient-like vector fields can be obtained by choosing a Riemannian metric g that is given by $\sum_{i=1}^{m}(dx^i)^2$ in a Morse neighborhood of each critical point. Then the gradient with respect to this metric, $\nabla_g f$, is a gradient-like vector field.

Definition 1.1.15 *Given a Morse function $f : M^m \to \mathbb{R}$ and a gradient-like vector field X, let Ψ_t be the downward gradient flow, i.e., the flow induced by $-\nabla X$. The stable manifold and the unstable manifold of a critical point p are the subsets*

$$M_s(p) = \left\{ q \in M \,\middle|\, \lim_{t\to\infty} \Psi_t(q) = p \right\}$$

and

$$M_u(p) = \left\{ q \in M \,\middle|\, \lim_{t\to-\infty} \Psi_t(q) = p \right\}$$

respectively.

Definition 1.1.16 *Let M^m be a manifold with boundary and H be a manifold diffeomorphic to D^m. Let $A \subset \partial H$ and $B \subset \partial M^m$ be $(m-1)$-submanifolds of the boundaries, both diffeomorphic to $S^{i-1} \times D^{m-i} \subset \mathbb{R}^m$ via diffeomorphisms that can be extended into collars of A and B. Let $M' = (M \sqcup H)/(A \equiv B)$ be the space obtained by gluing H to M using the identification provided by the diffeomorphisms. Then M' is the result of attaching a smooth i-handle to M.*

A *handle decomposition* of a manifold M is given by a sequence of handles H_i so that $M = \bigcup_i H_i$ with the handles attached in order. If there are h_i handles of index i in a given handle decomposition of M, the sequence $(h_0, \dots, h_m)$ is called the *handle vector* of this decomposition.

Denote the *sublevel set* as ${}^aM := f^{-1}(-\infty, a]$. Similarly, ${}^b_aM := f^{-1}[a,b]$ and ${}_aM := f^{-1}[a, +\infty)$.

For the handle decomposition structure we have the next three results.

Theorem 1.1.17 *Suppose $f : M^m \to \mathbb{R}$ is a Morse function and $a < b \in \mathbb{R}$ two distinct values so that ${}^b_aM$ is compact and contains no critical points of f. Then ${}^bM$ is diffeomorphic to ${}^aM$ and moreover ${}^aM$ is a deformation retract of ${}^bM$.*

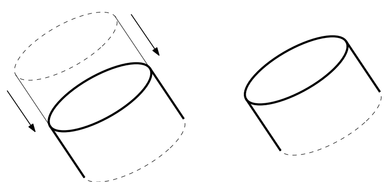

Figure 1.1: Deformation retract between b^M and a^M.

Theorem 1.1.18 *Let $f : M^m \to \mathbb{R}$ be a Morse function and p a critical point with $f(p) = c$ and index λ so that for some $\epsilon > 0$, the set ${}_{c-\epsilon}^{c+\epsilon}M$ is compact and contains no critical points other than p. Then ${}^{c+\epsilon}M$ has the same homotopy type as ${}^{c-\epsilon}M$ with λ-cell e^λ attached; moreover ${}^{c-\epsilon}M \cup e^\lambda$ is a deformation retract of ${}^{c+\epsilon}M$.*

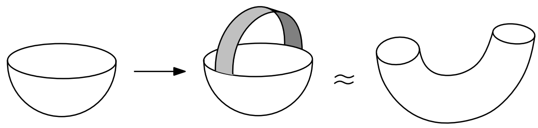

Figure 1.2: We can decompose the lower part of a torus as a disk an a 1-dimensional cell attached to it.

Theorem 1.1.19 *Suppose $f : M \to \mathbb{R}$ is an exhaustive Morse function, then M is homotopy equivalent to a CW complex with one λ-cell for each index λ critical point.*

1.2 DMT: Discrete Morse Theory

In this section, we're going to talk about the discrete version of Morse theory as it was presented by Robin Forman [For98]. In particular we are interested in the relationship of critical points and critical cells in the discrete case, as well as what would be the gradient flow for a discrete Morse function.

A cell σ of a CW-complex is a *regular face* of τ if the characteristic map of τ restricted to the inverse image of the interior of σ is a homeomorphism and the closure of this inverse image is a closed p-ball. A CW complex is called regular if every face is regular. The CW complex representing the n-dimensional sphere with one 0-dimensional cell and one n-dimensional cell is an example of a non-regular face since the attaching for the boundary of the n-cell is not a homeomorphism. But we can construct a regular CW-complex for the n-dimensional sphere attaching the boundary of the i-cell to an already constructed $i-1$-dimensional sphere, so there are exactly two cells of each dimension up to n. Every simplicial complex is a regular CW complex. We have particular interest in simplicial structures where we can apply most of this theory in a computational fashion.

It is well known that in a complex, we can find a partial order induced by the relation of being a face. For two cells σ, τ, we will write $\sigma < \tau$ is σ is a proper face of τ, and $\sigma \leq \tau$ if $\sigma < \tau$ or $\sigma = \tau$. If the cell σ has dimension p, sometimes it is useful to denote it by $\sigma^{(p)}$.

Lemma 1.2.1 *Given cells $\nu < \sigma < \tau$ with ν a regular cell of σ and σ a regular cell of τ. Then there is a cell $\tilde{\sigma}^{(p)} \neq \sigma$ so that $\nu < \tilde{\sigma} < \tau$.*

Proof: We prove it for codimension 1 and the general case follows inductively. Let's proceed by constradiction. First we note that ν is totally contanined in the boundary of σ and σ is the only p dimensional cell containing ν, otherwise we should have the cell $\tilde{\sigma}$ contradicting our assumption. For other hand we know that $\partial^2\tau = 0$. Therefore, the boundary of σ is empty and thus it can not be homeomorphic to the restriction of the characteristic map. $\diamondsuit$

Next we have the main definition in the discrete setting of Morse theory.

Definition 1.2.2 (Forman conditions) *A discrete Morse function on a CW complex M is a function $f : M \longrightarrow \mathbb{R}$ so that any irregular face σ of τ satisfies $f(\sigma) < f(\tau)$. In addition, for any cell $\sigma^{(p)}$ we require*

- $\left|\tau^{(p+1)} > \sigma : f(\tau) \leq f(\sigma)\right| \leq 1$
- $\left|\nu^{(p-1)} < \sigma : f(\nu) \geq f(\sigma)\right| \leq 1$

We can think this definition as an assignation in the cells such that we tend to increase both in the dimension and the value of assigned to the cell. We will denote the sets above as $M^+_{f_\sigma}$ and $M^-_{f_\sigma}$ respectively, omitting the f where the chosen function is clear.

Definition 1.2.3 *A critical cell for a discrete Morse function is a cell $\sigma^{(p)}$ so that*

- $\left|M^+_\sigma\right| = 0.$
- $\left|M^-_\sigma\right| = 0.$

The *index* of σ is defined to be its dimension.

Lemma 1.2.4 *Suppose f is a discrete Morse function on M. For a non-critical cell $\sigma^{(p)}$ exactly one of M^+_σ and M^-_σ contains a cell, the other is empty.*

Proof: If $\dim M = 0$, there is nothing to prove. Now suppose the dimension is positive and take a cell $\tau^{(p+1)}$ coface of σ such that $f(\tau) \leq f(\sigma)$. By the conditions in Definition 1.2.2 and the fact that our complex is regular, we have that for any other face $\tilde{\sigma}$ of τ we must have $f(\tilde{\sigma}) < f(\tau) \leq f(\sigma)$. Now, let's take $\nu^{(p-1)}$ face of σ and suppose $f(\nu) \geq f(\sigma)$. By the previous Lemma, there is another face $\nu < \sigma' < \tau$ and such face must satisfy $f(\nu) < f(\sigma')$ by the second rule in Definition 1.2.2. With the previous inequality, we have $f(\sigma) < f(\sigma') < f(\sigma)$. Thus, τ cannot have a value greater than ν. $\diamondsuit$

Definition 1.2.3 together with the previous lemma tells us that non-critical cells come in pairs respect the discrete function f. With this observation we can define the discrete analog to what we had as flow in the smooth setting. We write M_p to denote the set of p-dimensional cells of M.

Definition 1.2.5 *Given a discrete Morse function f on M, the discrete gradient of f is a set of maps of cells $V_f : M_p \longrightarrow M_{p+1} \cup \{0\}$ (one for each p) so that if $\tau^{(p+1)} > \sigma^{(p)}$ has $f(\sigma) \geq f(\tau)$ then $V(\sigma) = \tau$ and $V(\sigma) = \tau$ and $V(\sigma) = 0$ otherwise.*

As said previously, the properties of a Morse function f defined over the complex M can be reinterpreted in terms of the unique Morse pairs we described in Lemma 1.2.4. We combine what we just discussed about discrete vector fields in the next definition to generalize the previous one.

Definition 1.2.6 *Let M be a cell complex. A discrete vector field is a linear map $V : M \to M \cup \{0\}$ so that*

- $V(M_p) \subset M_{p+1} \cup \{0\}$
- $V(\sigma) = 0$ *if σ is an irregular face of $V(\sigma)$*
- $V(\sigma) = 0$ *if $\sigma \in \operatorname{Im} V$*
- *Given $\sigma^{(p)}$, $\left|\left\{ \nu^{(p-1)} : V(\nu) = \sigma \right\}\right| \leq 1$.*

These two definitions may seem very similar at first glance. The second one is a bit more general and therefore there could be cases where we have discrete vector fields that are not gradient vector fields. However, from the first definition it is clear that every discrete gradient is a discrete vector field (since we defined our complex to be regular). Let's explore this in more detail.

Definition 1.2.7 *A **V-path** is a sequence of cells $\sigma_0^{(p)}, \sigma_1^{(p)}, \dots, \sigma_n^{(p)}$ so that if $V(\sigma_i) = \tau \neq 0$, then $\sigma_i \neq \sigma_{i+1}$ and $\sigma_{i+1} < \tau$. If $V(\sigma_i) = 0$ then $i = n$. A path is called **closed** if $\sigma_n = \sigma_0$ and $n \neq 0$. The length of the V-path $\sigma_0^{(p)}, \sigma_1^{(p)}, \dots, \sigma_n^{(p)}$ is n.*

Lemma 1.2.8 *Let f be a discrete Morse function and $(\sigma_0, \dots, \sigma_n)$ be a $V-$path in the gradient induced by f. Then $f(\sigma_{i+1}) < f(\sigma_i)$ for all $i \in \{0, \dots, n\}$.*

Proof: For any σ_i in the sequence we have a unique coface $\tau > \sigma$ such that $f(\sigma) \geq f(\tau)$, then

$$f(\sigma') < f(\tau) = f(V(\sigma_i)) \leq f(\sigma_i)$$

for any $\sigma' < V(\sigma_i)$. In particular, $f(\sigma_{i+1}) < f(\sigma_i)$. ◇

The previous lemma asserts that for every gradient induced by a Morse function f is such that every V-path is decreasing. Moreover, we are moving along the path flowing in the "negative" direction respect the gradient from any cell to a critical point of same dimension. The next result is important to characterize the discrete gradients as we state in the later Theorem.

Corollary 1.2.9 *Given a discrete Morse function f there are no closed V-paths in the gradient induced by f.*

Theorem 1.2.10 (Forman theorem) *A discrete vector field V if the discrete gradient for some discrete Morse function f if and only if V has no closed V-paths.*

Proof: Suppose we have a closed path $(\sigma_0, \ldots, \sigma_n = \sigma_0)$, adding the respective $V(\sigma_s)$'s we have

$$\sigma_0 < \tau_0 > \ldots < \tau_{n-1} > \sigma_n = \sigma_0$$

Also, since $V(\sigma)$ is the only coface with value less than or equal to $f(\sigma)$, we have

$$f(\sigma_0) \geq f(\tau_0) > \ldots \geq f(\tau_{n-1}) > f(\sigma_n) = f(\sigma_0)$$

which is a contradiction. So we can not have closed paths.

Conversely, suppose V is a vector field with no closed paths. We construct a explicit Morse function f with the given vector field. We proceed inductively building a function f_k and gradient V_k for the k-dimensional skeleton M_k of the complex M. For the first inductive step, we set $f_0(v) = 0$ and $V_0(v) = 0$ for every vertex in M_0.

Now, suppose we have f_{p-1} with gradient V_{p-1} and define the V-path distance

$$d(\sigma) \coloneqq \max\{r : \exists(\sigma, \sigma_1, \ldots, \sigma_r) \text{ such that } V(\sigma_r) = 0\}$$

and let $D \coloneqq \max\{d(\sigma) : \sigma \text{ is a } (p-1)\text{-cell}\}$. So we can define the function f_p as follows

$$f_p(\sigma^{(q)}) = \begin{cases} f_{p-1}(\sigma^{(q)}) & \text{if } q < p-1 \\ f_{p-1}(\sigma^{(q)}) + \frac{d(\sigma^{(q)})}{2D+1} & \text{if } q = p-1 \end{cases}$$

and for p-dimensional cells

$$f_p(\sigma^{(p)}) = \begin{cases} p & \text{if } V(\sigma^{(p)}) = 0 \\ f_p(\nu) & \text{if } V(\nu) = \sigma^{(p)} \end{cases}$$

Note that the V-path distance is well defined because we do not have closed paths. Also, our function is defined for every cell in M_p. Moreover, since $f(\nu)$ is at most

$p-1/2$, for every p-dimensional cell σ there is at most one cell ν that is face of σ and has f-value greater or equal than $f(\sigma)$. And if ν' is another face of σ we have the V-path $(\nu, \nu', \dots, \sigma_r)$ and then $d(\nu) > d(\nu')$. Due to the fact that there are not $p+1$ cells, the second condition of the Morse functions is trivially satisfied.

If σ is a $p-1$-dimensional cell and ν is a face of it, we have $f_p(\nu) = f_{p-1}(\nu)$ and $f_p(\sigma) > f_{p-1}(\sigma)$. Thereby, if $f_{p-1}(\nu) \geq f_{p-1}(\sigma)$ then $V(\nu) = \sigma$ and we conclude as before that ν is the only face of σ with f-values greater or equal than $F(\sigma)$.

Finally, if we have a face relation $\sigma > \nu$ where $\dim(\sigma) < p-1$, by construction we have $f_p(\sigma) = f_{p-1}(\sigma)$ and $f_p(\nu) = f_{p-1}(\nu)$, so the Forman conditions are satisfied as they are for f_{p-1}. Therefore, f is a Morse function.

It is not hard to note that V is the gradient vector induced by f, this is because in the k-dimensional step we added the respective pairs $V(\sigma^{(k-1)})^{(k)}_{\tau}$ as we can see in the definition of f_p. ♢

We move to a classic concept in computational topology and talk about a definition we will need in future chapters.

First, given a regular cell complex M, a *free face* σ in M is a cell such that it has a unique coface. An *elementary collapse* is the operation of removing a pair (σ, τ) where σ is a free face and τ is its unique coface. We represent this collapse as $M \searrow_{\sigma} M'$. We say M collapses to a complex N, if there exists a sequence of elementary collapses $\left(M \searrow_{\sigma_1} M^1 \searrow_{\sigma_2} \cdots \searrow_{\sigma_n} M^n = N\right)$. Sometimes it will be convenient to omit the sequence of elementary collapses and write simply $M \searrow N$.

Definition 1.2.11 (Collapsible complex) *A cell complex is collapsible if it collapses to a single vertex.*

Hereunder, we have a nice relation between collapsability and Morse functions.

Lemma 1.2.12 *The vector field V defined by a sequence of collapses*

$$M^0 \searrow_{\sigma_1} M^1 \searrow_{\sigma_2} \cdots \searrow_{\sigma_n} M^n$$

is a discrete Morse gradient.

Proof: We define the pairs $V(\sigma) = \tau$ such that (σ, τ) is a pair in the sequence of collapses. Note that for any cell σ not removed in the collapsing process we have $V(\sigma) = 0$. Also we note that once a cell is removed, it can't be removed again. In other words, we cannot have closed V-paths. Applying Theorem 1.2.10 we obtain the result. ♢

The problems of collapsability and finding a Morse gradient are strongly related as it is stated in the next theorem and in the next chapters.

Theorem 1.2.13 *Given a discrete gradient V on an d-complex M, let M' be the subcomplex obtained by removing all critical d-cells from M. Then $M' \searrow M''$ where M'' is a subcomplex of M such that* $\dim(M'') < \dim(M)$.

Proof: Let f be a function with the gradient field given in the hypothesis. Since our complex is finite, we can perturb the f-values adding or subtracting a small value ϵ such that $f \pm \epsilon$ is injective and the gradient pairs are not modified. In particular, the critical cells are still unpaired. Then, we can order the gradient pairs and thus giving an order in the sequence of elementary collapses we will perform.
Now, remove every critical cell in M to get M'. We can start collapsing the pair $(\sigma^{(k)}, \tau^{(k+1)})$ such that $f(\sigma)$ is the maximum value. To see this, note that for every V-path starting at σ we have that $f(\sigma) > f(\sigma')$ for any σ' in M'. On the second hand, τ is the only coface of σ because otherwise we could have a pair (σ', τ') such that $f(\sigma') \geq f(\tau') > f(\sigma) \geq f(\tau)$, a contradiction to the maximality of $f(\sigma)$. We can repeat this process until there is not more collapsable pairs. Note that M'', the finall complex after the whole sequence of collapses, should have strictly lower dimension than M since we have removed every k cell, either because it is a critical cell or it was in a collapsible pair. ♢

Corollary 1.2.14 *Given a discrete gradient on a d-complex M, we can reduce M to its critical* 0*-cells.*

As we will see later, the previous corollary is the key to analyze the complexity of computing a Morse gradient. It is worth pointing out the following theorems are the equivalent discrete versions of theorems in the smooth setting ([For98]).

For the sublevel set in the discrete case we use $M(a) := f^{-1}(-\infty, a]$. Note that this is subcomplex of the cell complex M.

Theorem 1.2.15 *Suppose M is a CW complex with a discrete Morse function f satisfying the discrete Morse hypothesis and $a < b$ are numbers so that no critical cell of f have value in $[a, b]$; then $M(b) \searrow M(a)$.*

Theorem 1.2.16 *Suppose M is a CW complex with a discrete Morse function satisfying the discrete Morse hypothesis and $a < b$ are such that there is a single critical point, $\sigma^{(p)}$, with value in $[a, b]$. Then $M(b)$ is homotopy equivalent to $M(a) \cup_{\partial e^p} e^p$, that is $M(a)$ with a p-cell attached along its boundary.*

2

STRATIFICATION OF A TOPOLOGICAL SPACE

In the previous chapter we dealt with the fundamental concepts of Morse theory both in the smooth case and in the discrete case. Now we want to go into the most important concept of this work: stratification of a topological space. In particular, we want to investigate a manner to split a topological space into connected components that are manifolds such that in these components we can apply what we already know about Morse theory, specifically, we can talk about the complexity of computing a Morse matching in a stratified space.

First, we start with a definition and a particular example to illustrate what we are pursuing.

Definition 2.0.1 *A subset F of a partially ordered set or poset* $(P, \leq)$ *is a filter if the following conditions hold:*

1. *F is nonempty.*

2. *F is nonempty. For every x, y in F, there is some element z in F such that* $z \leq x$ *and* $z \leq y$.

3. *For every x in F and y in P,* $x \leq y$ *implies that y is in F.*

A stratification of a topological space X *is a filtration by closed subsets*

$$\emptyset = X_{-1} \subseteq X_0 \subseteq X_1 \subseteq X_2 \subseteq \cdots \subseteq X_{n-1} \subseteq X_n = X,$$

such that $X_i - X_{i-1}$ *is a* i*-manifold for each* i. *The set* $X - X_{i-1}$ *is called the i-stratum and its components are the dimension i pieces of* X.

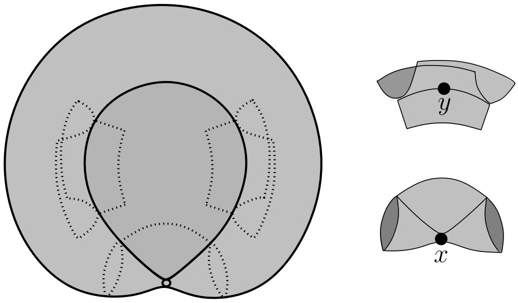

Figure 2.1: Non-manifold space with two kinds of singularities.

We note that the filtration in the definition of the stratified space is not unique. However, there is a natural coarsest filtration which consists of the components in the partition of X defined by calling points x and y equivalent if there exist neighborhoods of x and y and a homeomorphism between these neighborhoods that maps x to y and an algorithm to identify such coarsest stratification can be found in [Nan19]. We will present the basic steps for the construction of the coarsest stratification as we only need to identify the components in the second dimensional strata.

Once we have a stratification of the space we can proceed to find a Morse function for each strata.

Definition 2.0.2 (Cohomological stratification) *An n-dimensional cohomological stratification of a locally compact Hausdorff topological space* X *is an ascending sequence of closed subspaces*

$$\varnothing = \mathrm{X}_{-1} \subset \mathrm{X}_0 \subset \cdots \subset \mathrm{X}_{n-1} \subset \mathrm{X}_n = \mathrm{X},$$

where the connected components of each difference $(\mathrm{X}_d - \mathrm{X}_{d-1})$*, called* d*-dimensional strata, are required to obey the following two axioms.*

• *Frontier axiom: if a stratum* σ *intersects the closure of another stratum* τ*, then in fact* σ *is completely contained in the closure of* τ *and* $\dim \tau \geq \dim \sigma$ *(with equality occurring if and only if* $\tau = \sigma$*). This relation, denoted by* $\tau \succeq \sigma$ *henceforth, forms a partial order on the set of all strata.*

• *Link axiom: for each* d*-stratum* σ*, there exists an* $(n-d-1)$*-dimensional stratified space* $\mathrm{L} = \mathrm{L}(\sigma)$*, called the link of* σ*:*

$$\varnothing = \mathrm{L}_{-1} \subset \mathrm{L}_0 \subset \cdots \subset \mathrm{L}_{n-d-1} = \mathrm{L},$$

so that every open neighborhood around a point p in σ admits a basic open subneighborhood $\mathrm{U}_p \subset \mathrm{X}$ with the following structure. The intersections $\mathrm{U}_p \cap \mathrm{X}_i$ are empty for $i < d$, while for $d \le i \le n$ there are stratified quism of compactly supported singular cochain complexes (of R-modules):

$$C_c^\bullet(\mathscr{C} L_{i-d-1} \times \mathbb{R}^d) \xrightarrow{\simeq} C_c^\bullet(U_p \cap X_i) \tag{2.1}$$

where $\mathscr{C}\mathrm{L}_\bullet$ denotes the open cone on $\mathrm{L}_\bullet$.

By quism or quasi-isomorphism here we mean a cochain map which induces isomorphisms on all cohomology groups. The additional requirement of the maps from 2.1 being stratified means precisely that they behave in three separate ways: they should be filtered, stratum-preserving and refinable as explained below.

1. They should respect (contravariant maps induced by) the inclusions $\mathrm{L}_\bullet \subset \mathrm{L}_{\bullet+1}$ in their domains and $\mathrm{X}_\bullet \subset \mathrm{X}_{\bullet+1}$ their codomains so that the following squares commute:

$$\begin{array}{ccc} C_c^\bullet(\mathscr{C} L_{i-d-1} \times \mathbb{R}^d) & \longrightarrow & C_c^\bullet(U_p \cap X_i) \\ \uparrow & & \uparrow \\ C_c^\bullet(\mathscr{C} L_{i-d} \times \mathbb{R}^d) & \longrightarrow & C_c^\bullet(U_p \cap X_{i+1}) \end{array}$$

2. They should preserve strata in the sense that there exist surjective set-maps

$$\Phi_i : \{(i-d-1)-\text{strata of L}\} \twoheadrightarrow \{i-\text{strata } \tau \succeq \sigma \text{ of X}\}$$

so that for each i-stratum $\tau \succeq \sigma$, the cochain maps from (2) restrict to quasi-isomorphisms

$$C_c^\bullet\left(\mathscr{C}\Phi_i^{-1}(\tau) \times \mathbb{R}^d\right) \xrightarrow{\simeq} C_c^\bullet\left(U_p \cap \tau\right).$$

3. They should refine to smaller basic neighborhoods V_p so that the following triangle of cochain maps commutes:

$$\begin{array}{ccc} C_c^\bullet(\mathscr{C} L_{i-d-1} \times \mathbb{R}^d) & \xrightarrow{\simeq} & C_c^\bullet(U_p \cap X_i) \\ & \searrow^{\simeq} & \uparrow \\ & & C_c^\bullet(V_p \cap X_{i+1}) \end{array}$$

Here the vertical map is induced by the inclusion $\mathrm{V}_p \subset \mathrm{U}_p$, and since the triangle commutes, this map is forced to also be a quasi-isomorphism.

The *skeletal stratification* of a finite-dimensional regular CW complex X is defined as follows.

Writing the face partial order among cells as $\geq$, we recall that each cell y of X has

- an open star $\mathbf{st}(y)$ containing all cells x which satisfy $x \geq y$, and
- a link $\mathbf{lk}(y)$ containing all cells x that share a co-face but no face with y.

The d-dimensional skeletal strata are the d-cells of X, and the frontier partial order coincides with $\geq$. Since $\mathbf{st}(y)$ of a d-cell y is simultaneously homeomorphic to all sufficiently small neighborhoods around points in y and to $\mathscr{C}\mathbf{lk}(y) \times \mathbb{R}^d$, the link L$(y)$ in the sense of the former definition is precisely $\mathbf{lk}(y)$.

We call one stratification a *coarsening* of another whenever each stratum of the former is a union of strata of the latter. All stratifications encountered henceforth will be coarsenings of the skeletal stratification for a fixed finite-dimensional regular CW complex X.

Definition 2.0.3 *The canonical stratification of a finite-dimensional regular CW complex X is the coarsest stratification of X whose strata are all unions of cells.*

The following result is a direct consequence of the frontier axiom from Definition 2.0.2 for cells lying in top strata.

Proposition 2.0.4 *Let X be a regular CW complex of dimension n equipped with any stratification coarser than its skeletal stratification. If a cell y of X lies in an n-dimensional stratum σ, then so must every cell x which satisfies $x \geq y$.*

Proof: Let τ be the unique coarse stratum containing x. Since y lies in the boundary of x by assumption, the closure of τ intersects σ non-trivially at y. The desired conclusion now follows from the axiom of the frontier in Definition 2.0.2 and the fact that there are no strata of dimension exceeding n, since $\dim \tau \geq \dim \sigma = n$ forces $\tau = \sigma$. ♢

Thus, membership of cells in top-dimensional canonical strata is upward-closed with respect to the face partial order.

2.1 Local cohomology of CW complexes

We start this section with the foloowing definition.

Definition 2.1.1 (Localization of a poset) *Let $(\boldsymbol{P}, \geq)$ be a poset and $\Sigma = \{(x_\bullet \geq y_\bullet)\}$ a subset of its relations which is closed under composition. The localization of $\boldsymbol{P}$ about Σ is a category $\boldsymbol{P}[\Sigma^{-1}]$ whose objects are precisely the elements of $\boldsymbol{P}$, while morphisms from p to q are given by equivalence classes of finite (but arbitrarily long) Σ-zigzags of order relations in P*

$$p \geq y_0 \leq x_0 \geq \cdots \geq y_k \leq x_k \geq q,$$

where

1. *only relations in Σ and equalities can point backward (i.e., $\leq$),*
2. *composition is given by concatenation, and*
3. *the trivial zigzag $p = p$ represents the identity of each element p.*

The equivalence between Σ-zigzags is generated by the transitive closure of the following basic relations. Two such zigzags are related horizontally or vertically as follows:

Horizontally if one is obtained from the other by removal of intermediate equalities:

$$\left(\cdots \leq x \geq y = y \geq x' \leq \cdots\right) \sim \left(\cdots \leq x \geq x' \leq \cdots\right),$$
$$\left(\cdots \geq y \leq x = x \leq y' \geq \cdots\right) \sim \left(\cdots \geq y \leq y' \geq \cdots\right),$$

or vertically, if they form the rows in a grid:

$$\begin{array}{ccccccccccccc} p & \geq & y_0 & \leq & x_0 & \geq & \cdots & \geq & y_k & \leq & x_k & \geq & q \\ \shortparallel & & \vee\!\mid & & \vee\!\mid & & \vee\!\mid & & \vee\!\mid & & \vee\!\mid & & \shortparallel \\ p & \geq & y_0 & \leq & x_0 & \geq & \cdots & \geq & y_k & \leq & x_k & \geq & q \end{array}$$

where all vertical face relations lie in Σ.

Definition 2.1.2 *A cellular cosheaf over X is a functor $\boldsymbol{F} : \mathbf{Fc}(\mathrm{X}) \to \boldsymbol{A}$, such that $\boldsymbol{A}$ is an Abelian category. Thus, it assigns to each cell x an $\boldsymbol{A}$-object $\boldsymbol{F}(x)$ and to each face relation $x \geq y$ an $\boldsymbol{A}$-morphism $\boldsymbol{F}(x \geq y) : \boldsymbol{F}(x) \to \boldsymbol{F}(y)$ so that*

- *$\boldsymbol{F}(x = x)$ is the identity on $\boldsymbol{F}(x)$ for each cell x, and*
- *$\boldsymbol{F}(y \geq z) \circ \boldsymbol{F}(x \geq y)$ equals $\boldsymbol{F}(x \geq z)$ across any triple of cells $x \geq y \geq z$.*

We call $\boldsymbol{F}(x)$ the stalk of $\boldsymbol{F}$ at x, and call $\boldsymbol{F}(x \geq y)$ the extension map of $\boldsymbol{F}$ at $x \geq y$.

A morphism of cellular cosheaves $\eta : \boldsymbol{F} \to \boldsymbol{G}$ over X is a natural transformation; it assigns to each cell x an $\boldsymbol{A}$-morphism $\eta_x : \boldsymbol{F}(x) \to \boldsymbol{G}(x)$ so that for each $x \geq y$ the *relevant square in $\boldsymbol{A}$ commutes:*

$$\begin{array}{ccc} \boldsymbol{F}(x) & \xrightarrow{\boldsymbol{F}(x\geq y)} & \boldsymbol{F}(y) \\ \downarrow \eta_x & & \downarrow \eta_y \\ \boldsymbol{G}(x) & \xrightarrow{\boldsymbol{G}(x\geq y)} & \boldsymbol{G}(y) \end{array}$$

Cosheaf morphisms may be composed stalkwise, and there is always a zero morphism $0 : \boldsymbol{F} \to \boldsymbol{G}$ which (as one might expect) assigns the zero map $0_x : \boldsymbol{F}(x) \to \boldsymbol{G}(x)$ in $\boldsymbol{A}$ to each cell x of X.

Definition 2.1.3 (Complex of cellular cosheaves) *A lower-bounded complex of cellular cosheaves $\boldsymbol{F}^\bullet$ (over X, taking values in $\boldsymbol{A}$) is a sequence of $\boldsymbol{A}$-valued cellular cosheaves over X connected by cosheaf morphisms:*

$$\boldsymbol{F}^0 \xrightarrow{\eta^0} \boldsymbol{F}^1 \xrightarrow{\eta^1} \boldsymbol{F}^2 \xrightarrow{\eta^2} \cdots$$

so that every successive composition $\eta^{\bullet+1} \circ \eta^\bullet$ yields the zero morphism.

Every complex of cosheaves $\boldsymbol{F}^\bullet$ on X may be reinterpreted as a single cosheaf which takes values in the category $\mathbf{Ch}(\boldsymbol{A})$ of lower-bounded cochain complexes in $\boldsymbol{A}$—consider the collection of commuting squares in $\boldsymbol{A}$ that are assigned to face relations $x \geq y \geq z \geq \cdots$ of X:

$$\begin{array}{ccccccc} \boldsymbol{F}^0(x) & \longrightarrow & \boldsymbol{F}^1(x) & \longrightarrow & \boldsymbol{F}^2(x) & \longrightarrow & \cdots \\ \downarrow & & \downarrow & & \downarrow & & \\ \boldsymbol{F}^0(y) & \longrightarrow & \boldsymbol{F}^1(y) & \longrightarrow & \boldsymbol{F}^2(y) & \longrightarrow & \cdots \\ \downarrow & & \downarrow & & \downarrow & & \\ \boldsymbol{F}^0(z) & \longrightarrow & \boldsymbol{F}^1(z) & \longrightarrow & \boldsymbol{F}^2(z) & \longrightarrow & \cdots \\ \downarrow & & \downarrow & & \downarrow & & \\ \vdots & & \vdots & & \vdots & & \end{array}$$

The horizontal slices of this diagram confirm that a cochain complex in $\boldsymbol{A}$ is allocated to each cell, while vertical arrows between these slices prescribe the desired extension maps. Taking cohomology horizontally, one therefore obtains an ordinary cellular cosheaf of (graded) $\boldsymbol{A}$-objects over X. We call this the cohomology of $\boldsymbol{F}^\bullet$ and denote it by $H^\bullet\boldsymbol{F}$.

Let X be a finite-dimensional regular CW complex with face poset $\mathbf{Fc}(X)$ and R a fixed non-trivial commutative ring with identity 1_R. We write $\mathbf{Mod}(R)$ to denote the category of R-modules, and $\mathbf{Ch}(R)$—rather than the cumbersome $\mathbf{Ch}(\mathbf{Mod}(R))$—to indicate the category of cochain complexes of R-modules indexed by the nonnegative natural numbers.

Definition 2.1.4 *The local cohomology $\boldsymbol{L}^\bullet$ of X is a complex of cosheaves*

over X taking values in $\mathbf{Mod}(R)$, prescribed by the following data.

1. *For each dimension $d \geq 0$ and cell x of X, the cosheaf $\boldsymbol{L}^d$ has as its stalk $\boldsymbol{L}^d(x)$ the free R-module with basis*

$$\{z \in \mathbf{Fc}(\mathrm{X}) \mid z \geq x \text{ and } \dim z = d\}.$$

   When $x \geq y$, the extension $\boldsymbol{L}^d(x \geq y) : \boldsymbol{L}^d(x) \hookrightarrow \boldsymbol{L}^d(y)$ is determined by the obvious inclusion of basis cells.

2. *The cosheaf morphism β^d assigns to each cell x the map $\beta^d_x : \boldsymbol{L}^d(x) \to \boldsymbol{L}^{d+1}(x)$ defined by (linearly extending) the following action on basis cells. For each d-cell $z \geq x$, we have*

$$\beta^d_x(z) = \sum_w \langle w, z\rangle_R \cdot w,$$

   where the sum is taken over all $(d+1)$-cells $w \geq x$, and $\langle w, z\rangle_R$ is the R-valued degree of the attaching map in X from the boundary of w onto z. (Since we have assumed that X is regular, this number $\langle w, z\rangle_R$ takes values in $\{0, \pm 1_R\}$ for all cells w and z.)

It is not hard to show that $\beta^{d+1} \circ \beta^d$ is always zero and that the $\beta^\bullet_x$'s commute with all relevant extension maps. In light of the discussion following Definition 2.1.4, we will shift perspective and regard $\boldsymbol{L}^\bullet$ as a single cosheaf valued in $\mathbf{Ch}(R)$. In this setting, the stalk $\boldsymbol{L}^\bullet(x)$ lying over each cell x is the entire cochain complex

$$\boldsymbol{L}^0(x) \xrightarrow{\beta^0_x} \boldsymbol{L}^1(x) \xrightarrow{\beta^1_x} \boldsymbol{L}^2(x) \xrightarrow{\beta^2_x} \cdots$$

and $H^\bullet \boldsymbol{L}(x)$ is the compactly supported cohomology of x's open star in X. By excision, one may describe $H^\bullet \boldsymbol{L}(x)$ as the ordinary relative cohomology of the pair $\left(\overline{\mathbf{st}}(x), \partial\overline{\mathbf{st}}(x)\right)$ where $\overline{\mathbf{st}}(x)$ is the closure of x's open star in X and ∂ denotes the topological boundary:

$$\partial\overline{\mathbf{st}}(x) = \overline{\mathbf{st}}(x) - \mathbf{st}(x).$$

When y is a zero-dimensional cell, we have $\partial\overline{\mathbf{st}}(y) = \mathbf{lk}(y)$.

Definition 2.1.5 *For each $d \geq 0$, we write $R[d]^\bullet$ to indicate the special cochain complex in* **Ch**(R) *which is trivial except for a single copy of R in the dth position:*

$$0 \to \cdots \to 0 \to R \to 0 \to \cdots$$

Thus, $H^\bullet R[d]$ is the cohomology of small neighborhoods in d-dimensional R-cohomology manifolds, as in (1).

By Definition 2.0.2, if a cell x lies in some top-dimensional stratum of X, then we must have an isomorphism $H^\bullet \boldsymbol{L}(x) \simeq H^\bullet R[\dim \mathrm{X}]$. For cells of high dimension, this requirement has strong consequences.

Proposition 2.1.6 *If $n = \dim \mathrm{X}$, then every n-cell x has $\boldsymbol{L}^\bullet(x) = R[n]^\bullet$. And moreover, every $(n-1)$-cell y with $H^\bullet \boldsymbol{L}(y) \simeq H^\bullet R[n]$ lies in the boundary of exactly two n-cells.*

2.2 Structure of the strata

Throughout this section, X denotes an n-dimensional regular CW complex, R is a fixed non-trivial commutative ring with identity, and $\boldsymbol{L}^\bullet : \mathbf{Fc}(\mathrm{X}) \to \mathbf{Ch}(R)$ is the local cohomology complex of X from Definition 2.1.4. Consider the collection of face relations $(x \geq y)$ in **Fc**(X) sent by $\boldsymbol{L}^\bullet$ to quisms, i.e.,

$$E_0 = \{(x \geq y) \text{ in } \mathbf{Fc}(\mathrm{X}) \mid H^\bullet \boldsymbol{L}(x \geq y) \text{ is an isomorphism,}\}$$

and define the subcollection

$$W_0 = \{(x \geq y) \text{ in } E_0 \mid \ H^\bullet \boldsymbol{L}(y) \simeq H^\bullet R[n], \text{ and } \\ (x' \geq y) \in E_0 \text{ for all } x' \geq y \text{ in } \mathbf{Fc}(\mathrm{X})\}.$$

In other words, W_0 is what remains of E_0 when we impose

- *dimensionality*: retain only those face relations in which both cells have the local cohomology (isomorphic to that) of a top-dimensional cell, and
- *upward-closure*: remove $(x \geq y)$ if there exists some $x' \geq y$ for which $\boldsymbol{L}^\bullet(x' \geq y)$ is not a quism.

It is clear that if W_0 contains both $(x \geq y)$ and $(y \geq z)$, then it also contains $(x \geq z)$.

Definition 2.2.1 *The category $\boldsymbol{S}_0$ is the localization of the face poset* $\mathbf{Fc}(X)$ *about* W_0.

Recall that $\boldsymbol{S}_0$ has the cells of X as objects, while its morphisms are equivalence classes of W_0-zigzags. And, there is a canonical functor $\mathbf{Fc}(X) \to \boldsymbol{S}_0$ which is universal with respect to rendering all the face relations from W_0 invertible.

Proposition 2.2.2 *Two cells $w \neq z$ are isomorphic in $\boldsymbol{S}_0$ if and only if there is a W_0-zigzag*

$$w \geq y_0 \leq x_0 \geq \cdots \geq y_k \leq x_k \geq z$$

whose last relation $(x_k \geq z)$ lies in W_0.

Here is the main result of this section.

Proposition 2.2.3 *Two n-dimensional cells lie in the same canonical n-stratum of* X *if and only if they are isomorphic in $\boldsymbol{S}_0$.*

Finally, note that if an arbitrary cell y lies in a (not necessarily canonical) n-stratum σ of X, then there must exist an n-cell $w \geq y$ lying in σ by Proposition 2.7; otherwise, we arrive at the contradiction $H^n\boldsymbol{L}(y) = 0$. Thus, we have the following consequence of Proposition 4.3.

Corollary 2.2.4 *The canonical n-strata of X correspond bijectively with isomorphism classes of its n-cells in $\boldsymbol{S}_0$.*

2.3 Description and complexity of the algorithm

The first subroutine of the algorithm is UPSET has input (P, u, v) where P is the face poset of the complex X and elements $u, v \in P$ and its output is the subposet with all elements $\geq v$ but not $\geq u$. In the particular instance inside the main algorithm the output is precisely the subposet of $\mathbf{Fc}(X)$ containing the cells which lie in the difference of the open stars $\mathbf{st}(v) - \mathbf{st}(u)$.

The second subroutine is COHOM which accepts a poset as input and returns the sequence of R-modules corresponding to the cohomology of the associated cochain complex.

The main algorithm STRATCAST will accept the face poset of the complex X and will assign the value *codim(x)* to each cell in such a way that x will lie in a canonical stratum of dimension n-*codim(x)*. In particular, the algorithm will assign 0 to each cell lying in a canonical 2-stratum and −1 to the ramaining cells. Finally, identify the strata consists in simply identify the connected components in the poset of a fixed dimension.

Algorithm 1: StratCast(**Fc**(X))

Input: Face poset of finite cell complex X;
for w *in* **Fc**(X) **do**
    $h^\bullet(ws) \leftarrow$ Cohom(UpSet(**Fc**(X), $\emptyset$, w))
end
for $(x >_1 y)$ *in* **Fc**(x) *with* $\ell(x >_1 y) = -1$ **do**
    $c^\bullet(x, y) \leftarrow$ Cohom(UpSet(**Fc**(X), y, x));
    if $c^\bullet(x, y)$ *is trivial* **then**
        $\ell(x >_1 y) \leftarrow 0$
    end
end
codim(z) ← 0 to each n-cell z in **Fc**(X);
for i*-cell* u *in* **Fc**(X) *with* i *in* $(n-1, \ldots, 1, 0$ **do**
    if $h^n(u) \simeq \mathbb{R}$ *and all other* $h^\bullet(u) = 0$ **then**
        if codim(v) = 0 *and* $\ell(v >_1 u) = 0$ *for all* $v >_1 u$ *in* **Fc**(X) **then**
            codim ← 0
        end
    end
end
Output: Function **Fc**(X) → $\mathbb{N}$

3

Parameterized Complexity

"Do the difficult things while they are easy and do the great things while they are small. A journey of a thousand miles must begin with a single step." Lao-Tse

Contents

3.1 About Classical Complexity 26
3.2 About Parameterized Complexity 33

Certain computational problems are solvable in principle, but the solutions require so much time or space that they can not be used in practice. Such problems are called intractable. In the next chapters we analyze the problem of Morse matching from the classical definition of complexity and we will find that this problem is NP-complete even for the simplified case of pure 2-dimensional complexes. Having shown this, a question arises naturally: What makes the problem computationally difficult? A first attempt to try to answer this question may be to restrict the problem according to a certain characteristic of it, for example we could restrict the number of 2-dimensional cells that need to be removed in order to "erase" the complex. From the intuition we could affirm that the less 2-dimensional faces the more tractable the problem. This is not the case as we will see in this chapter. Additionally, we can use other features of the problem such as the so-called "width metrics" of graphs, e.g., cutwidth, bandwidth, or pathwidth of G. So, choosing a characteristic and a value k such that the number of apparitions of this feature is restricted by k we can ask how well can we do when k is fixed? This is the question we will address in the particular case of Morse matching. By setting a parameter in the problem we can go back to the complexity analysis this time from the point of view of the parameterized complexity. In this chapter we want to present the most basic concepts of parameterized complexity theory. We begin by presenting the idea of parameterized problem and fixed-parameterized tractability. Since this is a broad and relatively advanced theory, we will illustrate the definitions with examples of concrete problems without going into exhaustive details. For a better exposition of the topic we refer to [DF12] and [FG06]. For the classical part, we will follow mainly [Sip13].

3.1 About Classical Complexity

We will fix an alphabet, which will usually be taken as $\Sigma = \{0,1\}$. The strings obtainable from Σ will be denoted by Σ^*. A subset $L \subseteq \Sigma^*$ is called a *language*.

Definition 3.1.1 (Deterministic Turing Machine) *A **Deterministic Turing Machine** (*DTM*)* is a 7-tuple with the form $(Q,\Sigma,\Gamma,\delta,q_0,q_{\text{accept}},q_{\text{reject}})$, where Q,Σ,Γ are all finite sets and

1. Q is the set of states,
2. Σ is the input alphabet not containing the **blank symbol** $\sqcup$,
3. Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$,

4. $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$ is the transition function,

5. $q_0 \in Q$ is the start state,

6. $q_{\text{accept}} \in Q$ is the accept state, and

7. $q_{\text{reject}} \in Q$ is the reject state, where $q_{\text{reject}} \neq q_{\text{accept}}$.

A nondeterministic Turing machine (NTM) is defined in the expected way. At any point in a computation, the machine may proceed according to several possibilities. The formal definition for the NTM coincides with the definition of the DTM except by the transition function. The transition function for a NTM has the form

$$\delta : Q \times \Gamma \longrightarrow \mathscr{P}(Q \times \Gamma \times \{L, R\})$$

The computation of a nondeterministic Turing machine is a tree whose branches correspond to different possibilities for the machine. If some branch of the computation leads to the accept state, the machine accepts its input.

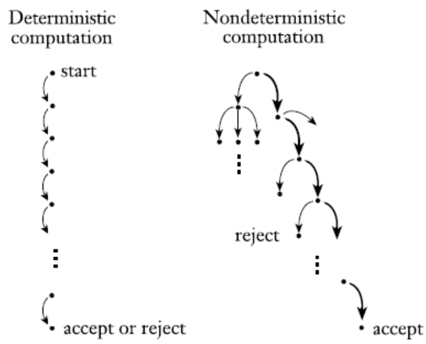

Figure 3.1: Deterministic and nondeterministic trees.

As a Turing machine computes, changes occur in the current state, the current tape contents, and the current head location. A setting of these three items is called a configuration of the Turing machine.

Here we formalize our intuitive understanding of the way that a Turing machine computes. Say that configuration C_1 yields configuration C_2 if the Turing machine can legally go from C_1 to C_2 in a single step.

A Turing machine M accepts input ω if a sequence of configurations $C_1, C_2, \ldots, C_k$ exists, where

- C_1 is the start configuration of M on input ω,

- each C_i yields C_{i+1}, and
- C_k is an accepting configuration.

The collection of strings that M accepts is the language of M, or the language recognized by M, denoted $L(M)$. A language is called *Turing-recognizable* if some Turing machine *recognizes* it.

A Turing machine M can fail to accept an input by entering the q_{reject} state and rejecting, or by looping. Sometimes distinguishing a machine that is looping from one that is merely taking a long time is difficult. For this reason, we prefer Turing machines that halt on all inputs; such machines never loop. These machines are called *deciders* because they always make a decision to accept or reject. A decider that recognizes some language also is said to *decide* that language. Call a language Turing-decidable or simply *decidable* if some Turing machine decides it.

The problem of deciding on a language is one of the most important philosophical issues of computational theory. When we have a problem in front of us, almost always the first step is to try to approach it from different perspectives in order to find a solution...but, what if this solution doesn't even exist? A very famous example of this is Hilbert's tenth problem whose answer ended up being: Unsolvable.

A more basic example of an undecidable problem is the problem of determining whether a Turing machine accepts a given input string. We describe this problem as follows

$$\mathrm{A}_{\mathsf{TM}} = \{\langle M, w\rangle | M \text{ is a } \mathsf{TM} \text{ and } M \text{ accepts } w\}.$$

The demonstration that this language cannot be decided by any Turing Machine uses Cantor's diagonalization argument and the fact that the number of Turing machines is countable while the number of languages in $\{0,1\}$ is not countable.

Another very important example in the problem of decidability is the so called the halting problem:

$$\text{HALT}_{\mathsf{TM}} = \{\langle M, w\rangle | \text{ is a } \mathsf{TM} \text{ and } M \text{ halts on input } w\}.$$

To prove that the halting problem is undecidable, one of the most fundamental tools in complexity theory is used: reducibility [Sip13]. For instance, if we suppose that there is a Turing machine such that the stop problem is decidable then we can prove that A_{TM} would also be so, which is a direct contradiction with what we have already said, and therefore we can conclude that $\text{HALT}_{\mathsf{TM}}$ is undecidable. It is generally easier to prove undecidability using a reduction to a known problem than to prove it directly.

The notion of reducing one problem to another may be defined formally in one of several ways. The choice of which one to use depends on the application. Our choice is a simple type of reducibility called *mapping reducibility*. Roughly speaking, being able to reduce problem A to problem B by using a mapping reducibility means that a computable function exists that converts instances of problem A to instances of problem B. If we have such a conversion function, called a reduction, we can solve A with a solver for B. The reason is that any instance of A can be solved by first using the reduction to convert it to an instance of B and then applying the solver for B. We can formalize the notion of reducibility as follows.

First, we remember the concept of computable function. A Turing machine computes a function by starting with the input to the function on the tape and halting with the output of the function on the tape. A function is said to be *computable* if there exists a Turing machine such that this machine halts with just $f(w)$ on its tape when started on any input w.

Definition 3.1.2 *A language A is mapping reducible to a language B, written $A \leq_m B$, if there is a computable function $f : \Sigma^* \to \Sigma^*$, where for every w,*

$$w \in A \Longleftrightarrow f(w) \in B.$$

The function f is called the reduction from A to B.

A mapping reduction of A to B provides a way to convert questions about membership testing in A to membership testing in B. To test whether $w \in A$, we use a reduction f to map w to $f(w)$ and test whether $f(w) \in B$. The term mapping reduction comes from the function or mapping that provides the means of doing the reduction. If one problem is mapping reducible to a second, previously solved problem, we can thereby obtain a solution to the original problem.

Theorem 3.1.3 *If $A \leq_m B$ and B is decidable, then A is decidable.*

Corollary 3.1.4 *If $A \leq_m B$ and A is undecidable, then B is undecidable.*

Even when a problem can be solved and therefore can be solved computationally in principle, it may not be solved in practice if the solution requires a disproportionate amount of time or memory, for example, we can not afford an algorithm that solves a given problem in a number of years when a quick and accurate response is required. To address this part of theory of computation, two main elements are defined: time and the amount of memory. Our goal in this chapter is to present the fundamentals of this theory in the case of temporal complexity.

The number of steps that an algorithm uses on a particular input may depend on several parameters. For instance, if the input is a graph, the number of steps may depend on the number of nodes, the number of edges, and the maximum degree of the graph, or some combination of these and/or other factors. For simplicity, we compute the running time of an algorithm purely as a function of the length of the string representing the input and don't consider any other parameters.

Definition 3.1.5 *Let M be a deterministic Turing machine that halts on all inputs. The running time or time complexity of M is the function $f : \mathbb{N} \to \mathbb{N}$, where $f(n)$ is the maximum number of steps that M uses on any input of length n. If $f(n)$ is the running time of M, we say that M runs in time $f(n)$ and that M is an $f(n)$ time Turing machine. Customarily we use n to represent the length of the input.*

Definition 3.1.6 *Let $t : \mathbb{N} \to \mathbb{R}^+$ be a function. Define the **time complexity class**,* TIME$(t(n))$, *to be the collection of all languages that are decidable by an $O(t(n))$ time Turing machine.*

Let N be a nondeterministic Turing machine that is a decider. The running time of N is the function $f : \mathbb{N} \to \mathbb{N}$, where $f(n)$ is the maximum number of steps that N uses on any branch of its computation on any input of length n.

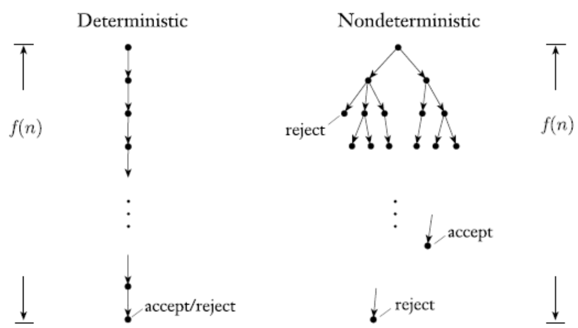

Figure 3.2: Illustration of time in a deterministic and nondeterministic tree.

Theorem 3.1.7 *Let $t(n)$ be a function, where $t(n) \geq n$. Then every $t(n)$ time nondeterministic single-tape Turing machine has an equivalent $2^{O(t(n))}$ time deterministic singletape Turing machine.*

Definition 3.1.8 *P is the class of languages that are decidable in polynomial time on a deterministic single-tape Turing machine. In other words,*

$$P = \bigcup_k \text{TIME}(n^k).$$

The class P plays a central role in our theory and is important because

1. P is invariant for all models of computation that are polynomially equivalent to the deterministic single-tape Turing machine, and
2. P roughly correspond

Definition 3.1.9 *A **verifier** for a language A is an algorithm V, where*

$$A = \{w | V \textit{ accepts } \langle w, c\rangle \textit{ for some string } c\}.$$

We measure the time of a verifier only in terms of the length of w, so a polynomial time verifier runs in polynomial time in the length of w. A language A is *polynomially time verifiable* if it has a polynomial time verifier.

Definition 3.1.10 *NP is the class of languages that have polynomial time verifiers.*

The term NP comes from nondeterministic polynomial time and is derived from an alternative characterization by using nondeterministic polynomial time Turing machines.

Theorem 3.1.11 *A language is in NP iff it is decided by some nondeterministic polynomial time Turing machine.*

We define the nondeterministic time complexity class NTIME($t(n)$) as analogous to the deterministic time complexity class TIME($t(n)$).

Definition 3.1.12

$$\text{NTIME}(t(n)) = \{L | L \textit{ is a language decided by an } O(t(n)) \textit{ time } \mathsf{NTM}\}.$$

Corollary 3.1.13

$$\text{NP} = \bigcup_k \text{NTIME}(n^k).$$

As we have been saying, NP is the class of languages that are solvable in polynomial time on a nondeterministic Turing machine; or, equivalently, it is the class of languages whereby membership in the language can be verified in polynomial time. P is the class of languages where membership can be tested in polynomial time. We summarize this information as follows, where we loosely refer to polynomial time solvable as solvable “quickly.”

P = the class of languages for which membership can be decided quickly.

NP = the class of languages for which membership can be verified quickly.

The question of whether P = NP is one of the greatest unsolved problems in theoretical computer science and contemporary mathematics. If these classes were equal, any polynomially verifiable problem would be polynomially decidable. Most researchers believe that the two classes are not equal because people have invested enormous effort to find polynomial time algorithms for certain problems in NP, without success. Researchers also have tried proving that the classes are unequal, but that would entail showing that no fast algorithm exists to replace brute-force search. Doing so is presently beyond scientific reach. The following figure shows the two possibilities.

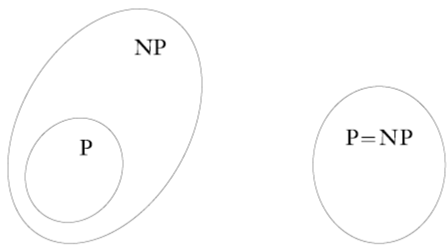

Figure 3.3: P=NP?.

The best deterministic method currently known for deciding languages in NP uses exponential time. In other words, we can prove that

$$\mathrm{NP} \subseteq \mathrm{EXPTIME} = \bigcup_k \mathrm{TIME}\left(2^{n^k}\right),$$

but we don't know whether NP is contained in a smaller deterministic time complexity class.

One important advance on the P versus NP question came in the early 1970s with the work of Stephen Cook and Leonid Levin. They discovered certain problems in NP whose individual complexity is related to that of the entire class. If a polynomial time algorithm exists for any of these problems, all problems in NP would be polynomial time solvable. These problems are called NP-complete.

The first NP-complete problem that we present is called the satisfiability problem. The satisfiability problem is to test whether a Boolean formula is satisfiable. Let

$$\textsc{Sat} = \{\langle\phi\rangle | \phi \text{ is a satisfiable Boolean formula}\}. \tag{3.1}$$

Now we state a theorem that links the complexity of the SAT problem to the complexities of all problems in NP.

Theorem 3.1.14 *SAT $\in P$ iff $P = NP$.*

When problem A reduces to problem B, a solution to B can be used to solve A. Now we define a version of reducibility that takes the efficiency of computation into account. When problem A is efficiently reducible to problem B, an efficient solution to B can be used to solve A efficiently.

Definition 3.1.15 *A function $f : \Sigma^* \rightarrow \Sigma^*$ is a polynomial time computable function if some polynomial time Turing machine M exists that halts with just $f(w)$ on its tape, when started on any input w.*

Definition 3.1.16 *A language A is polynomial time mapping reducible, or simply polynomial time reducible, to language B, written $A \geq_P B$, if a polynomial time computable function $f : \Sigma^* \rightarrow \sigma^*$ exists, where for every ω,*

$$w \in A \Longleftrightarrow f(w) \in B.$$

The function f is called the polynomial time reduction of A to B.

Polynomial time reducibility is the efficient analog to mapping reducibility. As with an ordinary mapping reduction, a polynomial time reduction of A to B provides a way to convert membership testing in A to membership testing in B – but now the conversion is done efficiently. If one language is polynomial time reducible to a language already known to have a polynomial time solution, we obtain a polynomial time solution to the original language, as in the following theorem.

Theorem 3.1.17 *If $A \leq_P B$ and $B \in P$, then $A \in P$.*

Definition 3.1.18 *A language B is **NP-complete** if it satisfies two conditions:*

1. *B is in NP, and*
2. *every A in NP is polynomial time reducible to B.*

Theorem 3.1.19 (Cook-Levin) *SAT is NP-complete.*

3.2 About Parameterized Complexity

Certain computational problems are solvable in principle, but the solutions require so much time or space that they can't be used in practice. Such problems are called intractable. In order to try to classify these problems, we will be considering languages $L \subseteq \Sigma \times \Sigma^*$. We refer to such languages as parameterized languages. If $\langle x, k \rangle$ is in a parameterized language L, we call k the parameter. Usually the parameter will be a positive integer, but it might be a graph or algebraic structure. However, in the interest of readability and with no loss of generality, we will usually identify the domain of the parameter as the natural numbers (in unary) $\mathbb{N}$ and hence consider languages $L \subseteq \Sigma^* \times \mathbb{N}$. For a fixed k, we call $L_k = \{\langle x, k \rangle : \langle x, k \rangle \in L\}$ the k-th slice of L.

As we have seen in the introduction, our main idea is to study languages that are tractable "by the slice." As the reader will recall, being tractable by the slice meant that there is a constant c, independent of k, such that for all k, membership of L_k can be determined in time $O(|x|^c)$.

Definition 3.2.1 *Let Σ be a finite alphabet.*

1. *A parameterization of Σ^* is a mapping $\kappa : \Sigma^* \to \mathbb{N}$ that is polynomial time computable.*

2. *A parameterized problem (over Σ) is a pair (Q, κ) consisting of a set $Q \subseteq \Sigma^*$ of strings over Σ and a parameterization κ of Σ^*.*

If (Q, κ) is a parameterized problem over the alphabet Σ, then we call strings $x \in \Sigma^*$ *instances* of Q and the numbers $\kappa(x)$ the corresponding *parameters*. Usually, we represent a parameterized problem (Q, κ) in the form

PROBLEM
Instance: $x \in \Sigma^*$.
Parameter: $\kappa(x)$.
Question: Decide whether $x \in Q$.

Example 3.2.2 *Let SAT be as in 3.1, we know SAT denote the set of all satisfiable propositional formulas, where propositional formulas are encoded as strings over some finite alphabet Σ. Let $x \in \Sigma^*$ and $\kappa : \Sigma^* \to \mathbb{N}$ be the parameterezation defined by*

$$\kappa(x) = \begin{cases} \textit{number of variables of } x, & \textit{if } x \textit{ is a propositional formula.} \\ 1, & \textit{otherwise.} \end{cases} \tag{3.2}$$

We denote the parameterized problem $(\textsc{Sat}, \kappa)$ by p-SAT. This problem can be represented as

p-**SAT**
Instance: A propositional formula α.
Parameter: Number of variables of α.
Question: Does it exist a truth assignation satisfying α?

Example 3.2.3 *In the previous chapter we talked a bit about the Vertex Cover problem in the non parameterized case. Now suppose we bound the covering by a certain number k, the problem in the parameterized version can be restated as follows*

p-**VERTEX COVER**
Instance: A graph $G = (V, E)$.
Parameter: A positive number k.
Question: Does G have a vertex cover of size $\leq$ k?

Now we define the first class of parameterized problems.

Definition 3.2.4 *Let Σ be a finite alphabet and $\kappa : \Sigma^* \to \mathbb{N}$ a parameterization.*

1. *An algorithm $\mathbb{A}$ with input alphabet Σ is an **fpt-algorithm with respect to κ** if there is a computable function $f : \mathbb{N} \to \mathbb{N}$ and a polynomial $p \in \mathbb{N}_0[X]$ such that for every $x \in \Sigma^*$, the running time of $\mathbb{A}$ on input x is at most*

$$f(\kappa(x)) \cdot p(|x|).$$

2. *A parameterized problem (Q, κ) is **fixed-parameter tractable** if there is an fpt-algorithm with respect to κ that decides Q.*

The motivation for the notion of fixed-parameter tractability is that if the parameter is small then the dependence of the running time of an algorithm on the parameter is not so significant. A fine point of the notion is that it draws a line between running times such as $2^k \cdot n$ on one side and n^k on the other, where n denotes the size of the input and k the parameter. But this may seem ambiguous because the k parameter would allow an exponential size function, so even for $k(1)$ the problem could be intractable. And however, this is not an exclusive problem of the definition of FPT problems, in the case of classical complexity, problems belonging to class P can present times of order n^c with c relatively small (say 15 or 20) and an algorithm with exponential time can have a relatively small base as for example 1.000001^n. Values of n for which the polynomial case is better than in the exponential can lead to time close to the age of the universe! Clearly this is not something we can call "tractable".

Example 3.2.5 *The parameterized satisfiability problem p-SAT is fixed-parameter tractable. Indeed, the obvious brute-force search algorithm decides if a formula α of size m with k variables is satisfiable in time $O(2^k \cdot m)$.*

Back to the Satisfiability problem. Computers are built from electronic devices wired together in a design called a digital circuit. We can also simulate theoretical models, such as Turing machines, with the theoretical counterpart to digital circuits, called Boolean circuits. Two purposes are served by establishing the connection between TMs and Boolean circuits. First, researchers believe that circuits provide a convenient computational model for attacking the P versus NP and related questions. Second, circuits provide an alternative proof of the Cook–Levin theorem that SAT is NP-complete.

Definition 3.2.6 *A Boolean circuit is a collection of gates and inputs connected by wires. Cycles aren't permitted. Gates take three forms: AND gates, OR gates, and NOT gates, as shown schematically in the figure 3.4.*

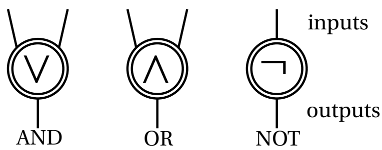

Figure 3.4: Logic gates.

The wires in a Boolean circuit carry the Boolean values 0 and 1. The gates are simple processors that compute the Boolean functions AND, OR, and NOT. The AND function outputs 1 if both of its inputs are 1 and outputs 0 otherwise. The OR function outputs 0 if both of its inputs are 0 and outputs 1 otherwise. The NOT function outputs the opposite of its input; in other words, it outputs a 1 if its input is 0 and a 0 if its input is 1. The inputs are labeled $x_1, \ldots, x_n$. One of the gates is designated the output gate. The figure 3.5 depicts a Boolean circuit for $[(\neg x_1) \vee (x_1 \wedge x_2)] \wedge [\neg[x_2 \wedge x_3]]$.

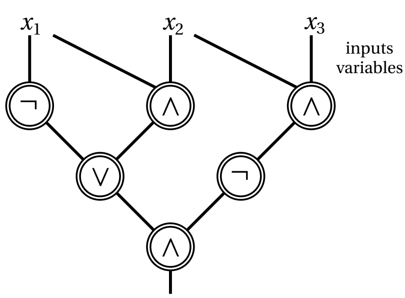

Figure 3.5: Boolean circuit.

A Boolean circuit computes an output value from a setting of the inputs by propagating values along the wires and computing the function associated with the respective gates until the output gate is assigned a value.

We use functions to describe the input/output behavior of Boolean circuits. To a Boolean circuit C with n input variables, we associate a function $f_C : \{0,1\} \to \{0,1\}$, where if C outputs b when its inputs $x_1,\dots,x_n$ are set to $a_1,\dots,a_n$, we write $f_C(a_1,\dots,a_n) = b$. We say that C computes the function f_C. We sometimes consider Boolean circuits that have multiple output gates. A function with k output bits computes a function whose range is $\{0,1\}^k$. We plan to use circuits to test membership in languages once they have been suitably encoded into $\{0,1\}$. One problem that occurs is that any particular circuit can handle only inputs of some fixed length, whereas a language may contain strings of different lengths. So instead of using a single circuit to test language membership, we use an entire family of circuits, one for each input length, to perform this task.

Definition 3.2.7 *A circuit family C is an infinite list of circuits, $(C_0, C_1, C_2, \dots)$, where C_n has n input variables. We say that C decides a language A over* $\{0,1\}$ *if for every string w,*

$$w \in A \text{ iff } C_n(w) = 1,$$

where n is the length of w.

The size of a circuit is the number of gates that it contains. Two circuits are equivalent if they have the same input variables and output the same value on every input assignment. A circuit is size minimal if no smaller circuit is equivalent to it. The problem of minimizing circuits has obvious engineering applications but is very difficult to solve in general. Even the problem of testing whether a particular circuit is minimal does not appear to be solvable in P or in NP. The depth of a circuit is the length (number of wires) of the longest path from an input variable to the output gate.The circuit complexity of a language is the size complexity of a minimal circuit family for that language. The circuit depth complexity of a language is defined similarly, using depth instead of size.The circuit complexity of a language is related to its time complexity. Any language with small time complexity also has small circuit complexity, as the following theorem shows.

Theorem 3.2.8 *Let $t : \mathbb{N} \to \mathbb{N}$ be a function, where $t(n) \geq n$. If $A \in \mathrm{TIME}(t(n))$, then A has circuit complexity $O(t^2(n))$.*

This theorem gives an approach to proving that P $\neq$ NP whereby we attempt to show that some language in NP has more than polynomial circuit complexity.

For languages L and L', we recall that $L \leq_T^P L'$ if and only if there is a polynomial-time Oracle Turing Machine[i] Φ such that for all $x \in \Sigma^*$, $x \in L$ iff $\Phi^{L'}(x) = 1$. We also recall that $L \leq_m^P L'$ iff there is a polynomial-time computable function $f : \Sigma^* \to \Sigma^*$, such that for all $x \in \Sigma^*$, $x \in L$ iff $f(x) \in L'$. Karp [Kar72] demonstrated the importance of polynomial-time m-reductions by constructing transformations between many natural combinatorial problems in a wide variety of diverse areas, thereby revealing that the NP completeness phenomenon is widespread. We recall the following polynomial-time m-reduction.

Reductions allow us to partially order languages in terms of their computational complexity. Two languages, L and L', are taken to have the same complexity from the point of view of the given reduction iff they are in the same degree; that is, there is a reduction from L to L' and vice versa. We will need to create reductions which express the fact that two languages have the same parameterized complexity. What is needed are reductions that ensure that if L' is computable in time $f(k)n^c$ "by the slice", then there is a parameterized algorithm for L running in time $g(k)n^c$ "by the slice". After a moment's thought, we realize that we can achieve such reductions only if we allow each slice of L to reduce to a finite number of slices of L'. The easiest way is to reduce, in parameterized polynomial time, the kth slice of L to the k'th slice of L'. This leads us to the working definition of a parameterized problem reduction.

Definition 3.2.9 *We say that L reduces to L' by a standard parameterized m-reduction if there are functions $k \to k'$ and $k \to k''$ from N to N, and a function $\langle x, k\rangle \to x'$ from $\Sigma * \times \mathbb{N} \to \Sigma *$, such that $\langle x, k\rangle \to x'$ is computable in time $k''|x|^c$, and $\langle x, k\rangle \in L$ iff $\langle x', k'\rangle \in L'$.*

In the next part we present the analog of Cook-Levin theorem for the case of parameterized complexity. First we introduce a paremeterized version of the TM acceptance problem.

SHORT TURING MACHINE ACCEPTANCE *Input:* A nondeterministic TM M and a string x. *Parameter:* A positive integer k. *Question:* Does M have a computation path accepting x in $\leq k$ steps?

Now consider a 3CNF formula as a (Boolean) circuit. Thus, a 3CNF formula is considered as a circuit consisting of one input (of unbounded fanout) for each variable, possibly inverters below the variable, and structurally a large AND of small OR's (of size 3) with a single output line (see figure 3.6). We can similarly consider a 4CNF formula to be a large AND of small OR's, where "small" is defined

[i] An Oracle Turing Machine (OTM) is a "black box" which somehow can tell us whether a given Turing machine with a given input eventually halts.

to be 4. More generally, it is convenient to consider the model of a decision circuit consisting of large and small gates with a single output line, and no restriction on the fanout of gates. A gate is called *large* if its fanin exceeds some preagreed bound.

Definition 3.2.10 (Weft) *Let C be a decision circuit. The weft of C is defined to be the maximum number of large gates on any path from the input variables to the output line.*

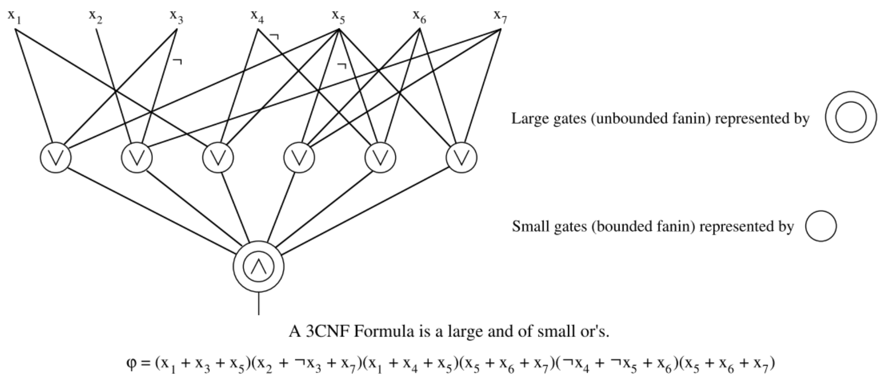

Figure 3.6: 3CNF formula as a circuit.

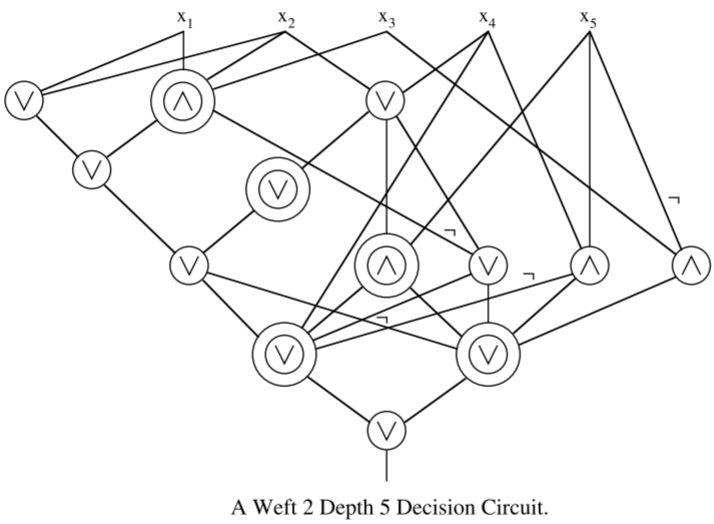

Figure 3.7: a Weft 2 Depth 5 Decision Circuit.

Let $\mathscr{F} = C_1, \ldots, C_n, \ldots$ be a family of decision circuits. Associated with $\mathscr{F}$ there is a basic parameterized language

$$L_{\mathscr{F}} = \{\langle C_i, k\rangle : C_i \text{ has a weight } k \text{ satisfying assignment }\}.$$

For instance, if $\mathscr{F}$ is the family of boolean circuits corresponding to propositional formulas in 3CNF form, then $L_{\mathscr{F}}$ corresponds to WEIGHTED 3CNF SATISFIABILITY. A generalization of the class of circuits corresponding to 3CNF formulas is provided by the WEIGHTED WEFT t DEPTH h CIRCUIT SATISFIABILITY, or WCS(t, h) for short, as follows.

WCS(t,h)
Input: A weft t depth h decision circuit C.
Parameter: A positive integer k.
Question: Does C have a weight k satisfying assignment?

Note that WEIGHTED 3CNF SATISFIABILITY is covered by WEIGHTED WEFT 1 DEPTH 2 CIRCUIT SATISFIABILITY.

We will denote by $L_{\mathscr{F}}(t, h)$ the parameterized language associated with the family of weft t depth h decision circuits.

Definition 3.2.11 (Basic hardness class) *We define a language L to be in the class W[t] if and only if L is fixed-parameter reducible to $L_{\mathscr{F}}(t, h)$ for some h.*

Theorem 3.2.12 (Analog of Cook-Levin's theorem) *The following are complete for W[1]:*

1. *WEIGHTED n-SATISFIABILITY for any fixed $n \geq 2$.*
2. *SHORT TURING MACHINE ACCEPTANCE.*

Definition 3.2.13 *The family of parameterized problems W[1,s] is defined to be those parameterized problems in W[1] reducible to $L_{\mathscr{F}}(s)$ for the family $\mathscr{F}(s)$ of depth 2 weft 1 normalized circuits, with the OR gates on level 1 having fan-in bounded by s.*

Lemma 3.2.14 *Let F$\mathscr{F}$ be a family of weft 1 circuits of depth bounded by a constant h. Then, $L_{\mathscr{F}}$ is reducible (by a standard reduction) to $L_{\mathscr{F}}(s)$ for $s = 2h + 1$, and hence $L_{\mathscr{F}} \in W[1, s]$.*

The W[t]-classes reflect the intrinsic difficulty of solution checking formulas of bounded logical depth. Naturally, the question arises as to what happens if we have no bound on the depth and simply look at parameterized problems of "polynomial size". When we do this we arrive at classes hard for $\bigcup_t W[t]$. Two important classes immediately suggest themselves. These are the classes W[SAT] and W[P] generated by the following problems.

WEIGHTED SATISFIABILITY
Input: A (Boolean) formula X.
Parameter: A positive integer k.
Question: Does X have a weight k satisfying assignment?

WEIGHTED CIRCUIT SATISFIABILITY
Input: A decision circuit C.
Parameter: A positive integer k.
Question: Does C have a weight k satisfying assignment?

Definition 3.2.15 (The W-hierarchy) *We term the union of these W[t] classes together with two other classes $W[SAT] \subseteq W[P]$, the W-hierarchy. Here W[P] denotes the class obtained by having no restriction on depth, i.e. P-size circuits, and W[SAT] denotes the restriction to circuits corresponding to Boolean formulas of P-size. Hence the W-hierarchy is*

$$FPT \subseteq W[1] \subseteq W[2] \subseteq \cdots \subseteq W[P].$$

It is conjectured the containments are proper.

4

Optimal Morse Matchings

"A property belongs esentially to a thing only if that thing would cease to exist without that property." Aristotle

Contents

4.1 Morse matchings **44**

4.2 Lewiner's algorithm **46**

4.3 Hardness of Optimal Morse matchings **50**

Find a discrete vector field on a simplicial complex can be a complicated procedure and the computational cost can be high because we have to verify that all the conditions of a Morse function are satisfied as well as to check that the set of directed paths on the complex are not closed. It is therefore ideal to have an effective method to find such functions. In particular what we want is an algorithm that presents the following characteristics:

- Optimality. This means we want to have as few critical cell as possible, and if possible to get a perfect Morse function.
- Low complexity. Also we would like to have a polynomial-time algorithm or near-polynomial-time.

We discussed these two points further throughout the chapter.

4.1 Morse matchings

Now consider a Morse function from a more graph-theoretical point of view. This interpretation of a Morse function was first noted by [Cha00].

A non-directed graph is a pair $G = (V, E)$ where V is a set of points and E is a set of unordered pairs of V, while a graph is directed if E is a set of ordered pairs of V. We allow pairs with the same element and we call them loops. In a given graph, we can take a special subset of the edges that we define as follows.

Definition 4.1.1 (Matching) *Given a graph $G = (V, E)$, a matching M in G is a set of pairwise non-adjacent edges.*

We are particularly interested in graphs with a structure of a *poset*. In such a graph we can set a direction to every edge according to the order in the poset. In this type of graphs we can define what is known as an acyclic matching. Given a Morse function f on a simplicial complex K, the most relevant information of this function can be encoded in a matching and to which we will call a Morse matching. Hereby, we define a graph structure that simplifies in some way the structure of a simplicial complex using the face relationship.

Definition 4.1.2 (Hasse diagram) *Let K be an abstract simplicial complex, we define the Hasse diagram $\mathscr{H}(K)$ or simply $\mathscr{H}$ as the directed graph where V is the set of simplices in K and a pair (β, α) is in E whenever α is a face of β and* $\dim(\beta) = \dim(\alpha) + 1$. *Also, we define $H_i \subset \mathscr{H}$ as the bipartite subgraph spanned by the set of nodes of $\mathscr{H}$ representing the i- and $i+1$-simplicial complexes. In particular, H_1 is subgraph of the Hasse diagram corresponding to the adjacencies between the 1-simplices and 2-simplices, and we call it the **spine** of the simplicial complex.*

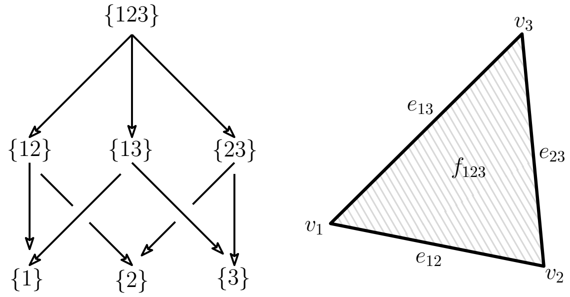

Figure 4.1: Hasse diagram of a 2-simplex.

Given a Hasse diagram $\mathscr{H}$ of a simplicial complex K and a matching $\mathscr{M}$ of this graph, we define the modified Hasse diagram with respect this matching, $\mathscr{H}_{\mathscr{M}}$, as the original digraph $\mathscr{H}$ where we reverse the edges in the matching $\mathscr{M}$. This is illustrated in the figure 4.2.

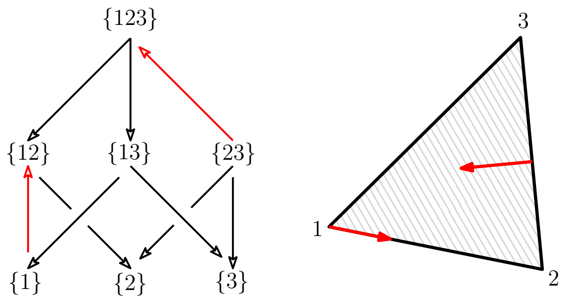

Figure 4.2: Partial acyclic matching.

Note in the original Hasse diagram every directed path goes from a face of greater dimension to a face of lower dimension, this implies there cannot be directed cycles. But once we reverse the edges in the matching we could create such cycles, see figure 4.3. Henceforth, we can conclude there is a relation between discrete vector fields on a simplicial complex K and the acyclic matchings of the respective Hasse diagram. This is stated in the following theorem.

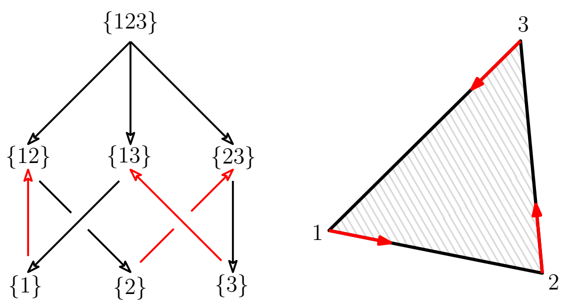

Figure 4.3: Directed cycle after reversing edges in the matching.

Theorem 4.1.3 *A discrete vector field V is the gradient field of a discrete Morse function on a simplicial complex K if and only if the modified Hasse diagram has no directed loops.*

The following theorem is an useful feature we find in a modified Hasse diagram when the Morse matching is not acyclic.

Theorem 4.1.4 *If the modified Hasse diagram has a cycle $\mathscr{C}$, we can find i such that $\mathscr{C} \subset H_i$.*

The poset structure of the Hasse diagram is very useful to us in a large number of cases as it encodes all the information from the simplicial complex into a single graph. In spite of them, many times we want to work with codimension face relations equal to 1, for example we can work directly with the 1-skeleton of the simplicial complex and that clearly is also a graph without need of any additional transformation. On the other hand, if we want to use the relation between edges and faces we will need a different structure, in the 2-dimensional case we can use a dual structure to the 1-skeleton of the complex. First let's remember that a loop in a graph is an edge whose enpoints indicate in the same vertex, and in a graph there are multiple edges if there are two edges or more whose endpoints are the same two vertices.

Definition 4.1.5 (Pseudograph) *A pseudograph is a non-simple graph in which both graph loops and multiple edges are permitted.*

Given a 2-dimensional simplicial complex, we define the dual pseudograph of the complex as the graph where each vertex represents a 2-dimensional face of the complex and two of these vertices are connected by an edge if there is a 1-dimensional face in common between the respective 2-dimensional simplicies, in particular, the cells of the boundary will be represented by loops in the dual pseudograph. This definition can be extended to dimension n, i.e., the vertices will be the n-dimensional faces and the edges the $n-1$-dimensional faces.

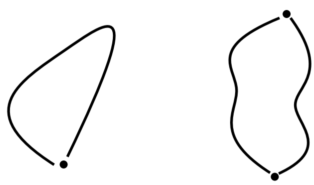

Figure 4.4: Example of loop and multipled edge.

4.2 Lewiner's algorithm

Building a Morse function for the one dimensional case is quite simple. This is similar to the computation of the homology groups of a graph: we only need a spanning tree and contract the graph on that tree to obtain a certain number of cycles and from there obtain the Betti numbers.

Since creating a spanning tree is not computationally expensive at all it is enough to consider a greedy algorithm, we would say that this first case is quite straightforward. If we analyze the two dimensional case we find a new level in the Hasse diagram. We could say that this case is the first non-trivial case and this is supported by the analysis of this problem in the next section. For now we will concentrate on finding a Morse function in the two dimensional case where the simplicial complex is homeomorphic to a manifold.

In this section we present the algorithm to compute optimal Morse functions for the case of 2-dimensional simplicial complexes from [LLT03a]. This algorithm uses a *dual pseudograph* of the simplicial complex and the Poincaré duality to obtain the minimum number of critical cells in the manifold case, that is the case when the simplicial complex is a combinatorial manifold. First we state the Optimality problem:

OPTIMALITY PROBLEM
Instance: A pair (K, n), where K is a 2-dimensional complex and $n \in \mathbb{Z}^+$.
Question: Does it exist a Morse function on K with at most n critical cells?

To compute an optimal Morse matching in the complex K we proceed as follows:

Construct a spanning tree T on the dual pseudograph of K. If K has a boundary edge, add one boundary edge of K to T. This edge will be a loop in the dual pseudograph. Define the discrete Morse function f on T as follows:

- Choose a root r in T
- For every vertex in T assign its distance from the root plus the number of vertices in K plus 1
- To every edge of T the maximum value of its two ends

Let G be the complement of T and let U be a spanning tree in G

- Assign to every vertex of G its distance from a chosen root q of U
- Assign to each edge in G the maximum value of its two ends
- For each edge $G \setminus U$ assign a value equal to the number of vertices in K

In the figure 4.5 we can see an implementation example of each of the steps of the algorithm. In the top of the figure we can see a triangulation of the torus and the tree T in the graph dual (green) and the spanning tree U of the 1-skeleton (blue). The only two edges that are not covered by either T or U are displayed in red.

Algorithm 2: Lewiner's Algorithm

Input: Simplicial complex X;
$T \leftarrow$ Spanning tree in X^*;
$U \leftarrow$ Spanning tree in $G \setminus E(T)$;
$r^*(T) \leftarrow \emptyset$;
$r(U) \leftarrow \emptyset$;
if $\partial X \neq \emptyset$ **then**
| $r^*(T) \leftarrow$ 2-cell in the boundary
end
else
| $r^*(T) \leftarrow$ any 2-cell
end
Output: $T, U, (G \setminus (E(T) \cup E(U)))$

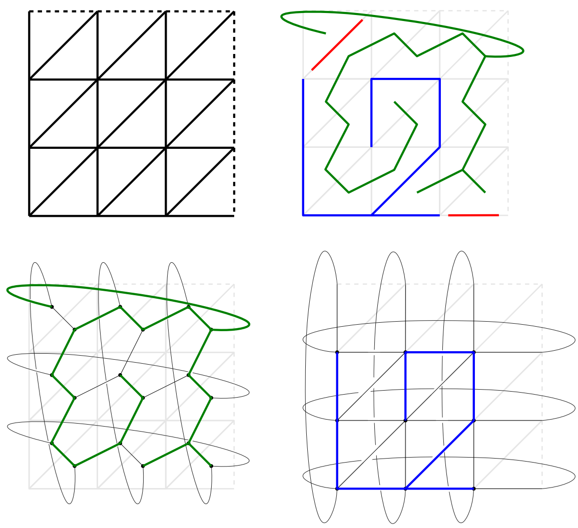

Figure 4.5: Example in a triangulation of the torus.

Proposition 4.2.1 *Algorithm 2 generates a discrete Morse function on X.*

Proof: We prove that the inequalities of the Morse conditions are satisfied. Let $v \in K$ be any vertex, we know such vertex is represented in the tree U since U is a spanning tree in the 1-skeleton of K. For other hand, there is only one path from v to the root q of U. So the only edge incident to v in this path is the unique edge with a value less or equal than the value of v since the values of the vertices increase as we move away from the root and we assign to each edge the greater value of its endpoints. For any other incident edge of v, the value will be strictly greater because the values on the edges are increasing as we move away respect the root q in the tree U or they are represented in the tree T where the values of the simplices are at least $|K_0|+1$, while in U the values are at most $|K_0|$. Thus the conditions are satisfied for the vertices.

For a 2-dimensional face $f \in K$ we have a vertex in the tree T. If e is an edge of f, then it is represented in T or in G. For the case when $e \in G$, its value is at most $|K_0|$ and the value of f is always greater than that value. If $e \in T$, we take the only path from the vertex representing f to the root r of T, if e belongs to this path then its value is exactly equal to the one of f and it will be the unique edge with a value greater or equal than the value of f. If e does not belong to this path, then its value will be equal to a vertex in T with a lower value than f since such vertex will have a lower height in the tree.

The analysis for the edges is straightforward from the two previous parts. $\diamondsuit$

The algorithm takes advantage of the duality of the 2-dimensional complex. We can see the only critical vertex in the complex is the root q in the tree U and the only 2-dimensional critical simplex is the one represented by the root r in T in the case of non-boundary, and for the case of boundary such root is paired with the loop added in the second step of the algorithm taking advantage from the fact that the complex is collapsible to a 1-dimensional complex meaning there should not be critical triangles. So our function is optimum in the 0- an 2-dimensional simplices. Finally, since we know $\chi(K) = c_0(K) - c_0(K) + c_0(K)$ then the number of critical edges need to be optimum to satisfied the Euler characteristic. We can conclude the following result.

Proposition 4.2.2 *The Morse matching obtained with the Algorithm 2 is a Optimal Morse Matching.*

This algorithm is linear in the case of manifolds as the creation of the dual pseudo-graph can be built with any of the known methods (See [Tar83], [LTR$^+$02]) and we can even take advantage of the data structure for the simplicial complex (Hasse diagram, simplex tree). The spanning tree can be constructed using a simple greedy algorithm and once it is saved, each vertex and edge is visited at most once.

Finally we want to emphasize the fact that the algorithm of the section 4.2 is linear by the fact that the simple complex is homeomorphic to a manifold. Then a natural question arises: why does the algorithm fail in the case of non manifolds? There are three cases where the complex can violate the manifold condition (see figure 4.6):

- Dangling edge.
- Singular vertex.

- Non-regular edge.

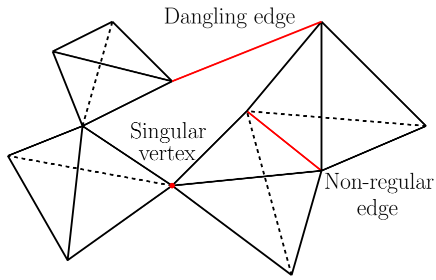

Figure 4.6: Non-manifold cases for 2-dimensional complexes.

In the first two cases the algorithm will process the singularities in the spanning tree of G because each vertex in the complex belongs to this tree and each dangling edge is not incident to any 2-dimensional face and therefore will not be represented in the spanning tree of the dual graph of the complex. Then as in the case of manifold, the algorithm visits each one of these simplices only once and the time is linear again.

The last case is when we have a conflict. This is because having a non-regular edge does not allow us to have a well defined face pseudograph. More spicifically, we must define an edge connecting several nodes representing more than two faces, what not occurs in the 2-dimensional manifold case. This type of edges can be defined in more general structure called a hypergraph. In [LLT03b], the authors proposed an approximation to this problem by stating it in the language of hypergraphs and hyperforests. Although, the exponential part of complexity is avoided using heuristics in the algorithm.

4.3 Hardness of Optimal Morse matchings

The above algorithm is applicable for cases when the simplicial complex is not a manifold and in such case it is where we find the different kinds of singularities previously described. However, the algorithm is not guaranteed to be linear and, moreover, the version of this problem, as a decision problem, turns out to be NP-complete even for the case of pure 2-dimensional simplicial complexes which allows us to claim that the problem is interesting enough for dimension 2. To prove that the problem is NP-complete we will use the usual approximation by a polynomial time reduction to a known NP-complete problem. This analysis follows the argument of [JP06] which is based on the reformulation of the collapsibility problem, which was proven to be strongly NP-complete[i] by [OEG96]:

COLLAPSIBILITY PROBLEM
Instance: A pair (K, n), where K is a pure 2-dimensional simplicial complex embeddable in $\mathbb{R}^3$ and $n \in \mathbb{Z}^+$.
Question: Does it exist a subset of simplices $S \subset K$ with at most n elements such that there exists a sequence of collapses which transform $K \setminus S$ to a 1-dimensional complex?

This problem can be reformulated as a problem in Morse theory as stated in section 4.2 using the proposition 1.2.15. As a consequence we can restate the main theorem in [OEG96] as follows:

Theorem 4.3.1 *Let K be a connected pure 2-dimensional simplicial complex embeddable in $\mathbb{R}^3$ and let k be a nonnegative integer. Then it is strongly NP-complete to decide whether there exists a discrete gradient vector field on K with at most n critical 2-simplices.*

Now we want to construct a polynomial time reduction from the problem of a discrete gradient vector with at most n critical 2-simplices to the problem of finding a discrete gradient vector field with a most m critical simplices in total. In order to do this, let's proceed with the construction given in [JP06].

Given a simplicial complex K with $\dim(K) \geq 2$ and a discrete gradient vector field X on K. Let M be the Morse matching in the Hasse diagram $\mathscr{H}(K)$ induced by the vector field x, consider the subgraph $\Gamma(M)$ of the 1-skeleton of K where we remove all the 1-simplices paired with 2-simplices in the Morse matching. Note all the vertices of the complex$K$ are also in $\Gamma(M)$.

Lemma 4.3.2 $\Gamma(M)$ *is connected.*

Proof: To prove this we proceed by contradiction. Suppose the graph $\Gamma(M)$ is disconnected and consider the set of cut edges C of K respect the connected components of $\Gamma(M)$. C is nonempty since the complex K is defined to be connected.

Let N be a connected component of $\Gamma(M)$, we take an edge $e_1 = \{v_1, v_2\} \in C$ and without loss of generality we suppose $v_1 \in N$. By definition of C, we know there is a face f_1 such that the pair $(e_1, f_1) \in M$. Since f_1 is a triangle, this implies it has another edge different from e_1 out-coming from N, otherwise the face have two

[i] A problem is said to be strongly NP-complete if it is NP-complete even in the case when its input parameters are bound by a polynomial in the length of the input.

edges in N and thus its three vertices are also in N but this contradicts the fact that e_1 has its two endpoints in different connected components. So, if we suppose the face f_1 is $\{v_1, v_2, v_3\}$ and such edge is $e_2 = \{v_1, v_3\}$ we have again a face f_2 such that $(e_2, f_2) \in M$ because e_2 is a cut edge, therefore we have the path e_1, f_1, e_2, f_2. We can proceed inductively with this argument and since the simplicial complex was defined to be finite, there will be a moment when we do not obtain new simplices in our path but this means we will start to repeat one of the simplices in the path obtaining a closed path in the Morse matching. This contradicts the fact that the Morse matching has not cycles. ♢

With the help of the graph $\Gamma(M)$ we can construct a Morse matching M' in the simplicial complex satisfying that the number of critical simplices of M' are not increased respect the critical cells in M, and in particular $c_0(M') = 1$ and $c_2(M') \leq c_2(M)$. Consider a rooted spanning tree T in $\Gamma(M)$ and take the edges outgoing respect the root r. Since the graph $\Gamma(M)$ is connected, T is a spanning tree in the 1-skeleton of K and we can direct the edges in the new matching M' as in the spanning tree T keeping the same direction in any other edges not in T, in this form we have a unique critical 0-simplex. Also note there is not possibility of increasing the number of critical cells since we are only changing the matching in H_0 getting exactly one 0-dimensional critical cell, and because the Morse inequalities we can see this is the less possible number. Thus, we have the following result.

Lemma 4.3.3 *Given a Morse matching M, we can construct a new matching M' optimal on $\Gamma(M)$ such that $c(M') \leq c(M)$, and $c_0(M') = 1$ and $c_2(M') = c_2(M)$.*

With all of this, we know we can find a Morse matching having at most k critical 2-dimensional simplices if and only if there exists a Morse matching with at most c critical simplices in total. We proceed to prove the main theorem of this section:

Theorem 4.3.4 *Given a simplicial complex K and $n \in \mathbb{Z}^+$, it is strongly NP-complete to decide if there exists a Morse matching with at most c critical simplices.*

Proof: We will assume the simplicial complex is a connected pure 2-dimensional complex. It is clear that this implies the problem is NP-complete in the general case.

For one hand, if we have a subset of edges of the Hasse diagram we can check in polynomial time[ii] if such subset is a Morse matching since it is at most as complex as the total sum of outgoing degrees of the vertices in a same H_i (see 4.1.4), hence the problem belongs to NP.

[ii]We could solve the problem even in linear time in the complexity of the graph, see [PP12]

Now suppose (K, n) is the input for the collapsibility problem. If we have a given Morse matching M with at most n 2-dimensional critical nodes in the Hasse diagram, using lemma 4.3.3 we can construct a new matching M' such that $c(M') \leq c(M)$, $c_0(M') = 1$ and $c_2(M') = c_2(M)$. Using the Morse equality we have $\chi(K) = c_2(M') - c_1(M') + 1$ and since $c(M') = c_2(M') + c_1(M') + c_0(M')$. Solving the former equations for $c_2(M')$ and adding both sides we have

$$c_2(M') = c(M') + \chi(K) - 2$$

By hypothesis we know $c_2(M) \leq n$ and since $c_2(M) = c_2(M')$ we obtain

$$n \geq c_2(M) = c_2(M') \leq \frac{1}{2}\big(c(M') + \chi(K)\big) - 1$$

Finally, solving for c(M') we have

$$c(M') \leq 2n + 2 - \chi(K)$$

Define the function $g(n) = 2(n+1) - \chi(K)$. Since the Euler characteristic is polynomial-time computable function[iii], so is $g(n)$.

Conversely, suppose $c(M) \leq g(n)$ for a given Morse matching M on K. Again we construct a Morse matching M' satisfying the same properties as before. The equality $c_2(M') = c(M') + \chi(K) - 2$ applies again and from the fact that $c(M') = c(M) \leq g(n)$ we have

$$c_2(M') = c_2(M) \leq \frac{1}{2}\big(g(n) + \chi(K)\big) - 1$$

Expanding $g(n)$ we get

$$c_2(M') = c_2(M) \leq \frac{1}{2}\big(2n + 2 - \chi(K) + \chi(K)\big) - 1 = n$$

We conclude that there exists a Morse matching with at most n critical 2-dimensional simplices if and only if there exists a Morse matching in K with at most $g(n)$ critical simplices in total. Since $g(n)$ is polynomial-time computable, we have a polynomial-time reduction and this together with theorem 4.3.1 we can conclude what we wanted to show. ◇

Then putting together all that we have done, we have shown that the problem of computing a Morse matching with at most c critical cells in total is polynomial-time reducible to the problem of computing a Morse matching with at most

[iii] The Betti numbers over finite fields can easily be obtained in polynomial time in the size of K, by computing the ranks of the boundary matrices for each dimension, see [EH10] for details.

n 2-dimensional critical simplices, and because this problem is equivalent to the collapsability problem then all this is summed up to show the strongly NP-completeness of the collapsability problem. In [OEG96], the authors use a reduction to the problem of vertex coverage:

VERTEX COVER PROBLEM
Instance: A pair (G, n), where $G = (V, E)$ is any undirected graph and $n \in \mathbb{Z}^+$.
Question: Does it exist a subset of vertices $S \subset V$ with a most n elements such that every edge $e \in E$ incides to at least one vertex $v \in S$?

The construction makes use of a set of gadgets to model the problem of vertex cover as a 2-simplicial complex with a particular structure and we will not attempt to expose that construction here. We invite the reader to review this paper if necessary.

5

PARAMETERIZED COMPLEXITY OF DMT

"Do the difficult things while they are easy and do the great things while they are small. A journey of a thousand miles must begin with a single step." Lao-Tse

From the proof done in [OEG96], we can reduce SETCOVER to ERASABILITY which is an equivalent problem to COLLAPSIBILITY, to show that ERASABILITY is NP-complete.

ERASABILITY PROBLEM
Instance: A pair (K, n), where K is a pure 2-dimensional simplicial complex embeddable in $\mathbb{R}^3$ and $n \in \mathbb{Z}^+$.
Question: Does it exist a subset of simplices $S \subset K$ with at most n elements such that there exists a sequence of collapses which transform $K \setminus S \rightsquigarrow \emptyset$?

Remember the SETCOVER problem is that given a family of subsets S over a set of elements X and a integer k , k-set cover consists of finding a subfamily $T \subseteq S$ of cardinality at most k. This problem is W[2]-hard when parameterized by k ([PM81]). So, it turns out that the reduction approach performed in this paper is a parameterized reduction, and these results can be restated in the language of parameterized complexity as follows.

Proposition 5.0.1 *SET COVER $\leq_{\text{FPT}}$ ERASABILITY, therefore ERASABILITY is W[2]-hard.*

This shows that, if the parameter k is simultaneously bounded in both problems, ERASABILITY is at least as hard as SET COVER. In this section we will determine exactly how much harder ERASABILITY is than SET COVER, which is W[2]-complete. Namely, we will show that ERASABILITY is W[P]-complete in the natural parameter k. This will be done by

i) using a W[P]-complete problem as an oracle to solve an arbitrary instance of ERASABILITY, and

ii) reducing an arbitrary instance of a suitable problem which is known to be W[P]-complete to an instance of ERASABILITY.

To show how hard is ERASABILITY compared to SET COVER in the W-hierarchy we make use of the following problem known to be W[P]-complete.

MINIMUM AXIOM SET
Instance: A finite set S of sentences, and an implication relation R consisting of pairs (U, s) where $U \subseteq S$ and $s \in S$.
Parameter: A positive integer k.
Question: Is there a set $S_0 \subseteq S$ (called an axiom set) with $|S_0| \leq k$ and a positive integer n, for which $S_n = S$, where we define $S_i, 1 \leq i \leq n$, to consist of exactly those $s \in S$ for which either $s \in S_{i-1}$ or there exists a set $U \subseteq S_{i-1}$ such that $(U, s) \in R$?

Theorem 5.0.2 *MINIMUM AXIOM SET is W[P]-complete.*

Theorem 5.0.3 *ERASABILITY $\leq_{\mathrm{FPT}}$ MINIMUMAXIOMSET, therefore ERASABILITY is in W[P].*

Proof: Let K be a simplicial complex, K_0 the set with the k triangles that we need to erase in K such that $K \setminus K_0 \rightsquigarrow G$ where G is a 1-dimensional subcomplex of K without triangles. Without loss of generality assume there are not free edges in K, otherwise we can collapse the corresponding edge together with the unique triangle containing it and proceed. Define S as the set of triangles in the simplicial complex K and for every subcomplex $K_i \subseteq K$, S_i will denote its set of triangles. For every edge e, we set $S_e := \mathrm{St}(e)$ and define the set of relations $(S_e \setminus \{s\}, s)$ for every triangle $\tau \in \mathrm{St}(e)$.

First we note that for every S_0 satisfying MINIMUMAXIOMSET we have $K \setminus K_0 \rightsquigarrow G$. To see this, let's take a sequence $S_0 \subset S_1 \subset \cdots \subset S_n = S$ and its associated sequence $K_0 \subset K_1 \subset \cdots \subset K_n = K$. Every $s \in S_i \setminus S_{i-1}$ has an associated edge e such that there is a relation $(S_e \setminus \{e\}, s)$ and $S_e \setminus \{s\} \subset S_{i-i}$. For the triangle τ associated to this s we have

$$S_e \setminus \tau \subset K_{i-1}$$

Thus τ must be an external triangle and hence it can be erased. Then, since all the triangles K_0 are already erased in $K \setminus K_0$ it follows $K \setminus K_0 \rightsquigarrow G$.

Now suppose we have given the set $K_0 \subset K$ such that $K \setminus K_0 \rightsquigarrow G$. Since K has no external edges then there exists a triangle τ such that this triangle is external in $K \setminus K_0$ and we can find a relation $(S_e \setminus \{s\}, s)$ with $S_e \setminus \{s\} \subset S_0$ where s is the sentence in S associated to τ. Define S_1 as the union of S_0 and all the relations of the type described previously. Proceeding inductively we get the sequence $S_0 \subset S_1 \subset \cdots \subset S_n = S$ for some n. ♢

To proof that ERASABILITY belongs to the hardest problems in the W[P] class we need to define the gadgets used in the reduction process. The gadgets that we use in the proof correspond to a "nice family" of simplicial complexes corresponding to a subfamily of embeddable 2-dimensional simplicial complexes. It turns out that even for this small class of complexes the problem of ERASABILITY is still hard.

Definition 5.0.4 (Sentence gadget) *Let (S, R, k) be an instance of the problem of MINIMUMAXIOMSET and let $s \in S$ be a sentence. By an s**-gadget** or **sentence gadget** we mean a triangulated 2-dimensional sphere with $2n + m$ punctures as shown in Figure 5.1, where $m = \#\{(U, s) \in R\}$ and $n = \#\{(U, s) \in R : s \in U\}$.*

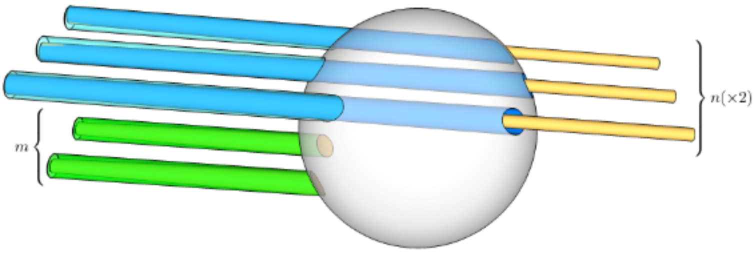

Figure 5.1: Example of sentence gadget.

Definition 5.0.5 (Implication gadget) *Let* $(U, s) \in R$ *be a relation. The* (U, s)-***gadget*** *or* ***implication gadget*** *is a collection of* $|U|+1$ *sentence gadgets for each sentence of* $U \cup \{s\}$ *together with* $2|U|$ *nested tubes as shown in the Figure 5.2 such that*

i) two tubes are attached to two punctures of the u*-gadget for each* $u \in U$*, and*

ii) all $2|U|$ *boundary components at the other side of the tubes are identified at a single puncture of the* s*-gadget.*

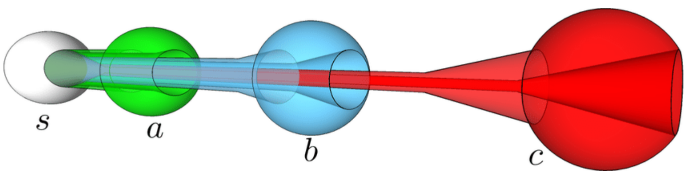

Figure 5.2: Example of an implication gadget with (U, s)-gadget with $U = \{a, b, c\}$, with sentence gadgets $\{a, b, c, s\}$.

Lemma 5.0.6 *An implication gadget can be erased if and only if all sentence gadgets corresponding to sentences in* U *are already erased.*

Proof: First, let (U, s) be the set of relation corresponding to the implication gadget. If we erased all the 2-dimensional spheres representing the sentence gadgets in U, we get only a collection of tubes or cylinders together with the 2-dimensional sphere representing the sentence s. These sets of cylinders can be erased one by one without problem since for each of them we can find a free face. Finally, we can erase the punctured sphere corresponding to s.

Note that each sentence gadget together with the two tubes connected to it is a complex without free faces and in fact it is homeomorphic to the torus (Figure 5.3). Hence, we cannot erase the implication gadget if we have not deleted each of its sentence gadgets. ◊

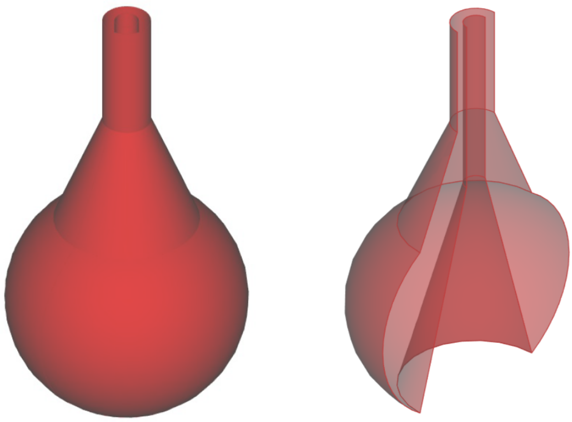

Figure 5.3: A sentence gadget and a view of the inside

We have proven the problem of erasability belongs to the class W[P]. Now we will prove that this problem is as hard as all the problems in this parameterized complexity class. Again, we note the complex that we can construct using the sentence and implication gadgets are 2-dimensional simplicial complexes embedded in $\mathbb{R}^3$.

Theorem 5.0.7 *MINIMUMAXIOMSET* $\leq_{\text{FPT}}$ *ERASABILITY. Therefore ERASABILITY is W[P]-hard.*

Proof: Let (S, R, k) be an instance for the MINIMUMAXIOMSET problem. Let K be the 2-dimensional simplicial complex representing this instance. As we observed previously, this complex K does not have free triangles that we can erase.

Let $S_0 \subset S$ be the set with the sentences in the minimum axiom set. We remove exactly one triangle in the 2-dimensional sphere of the respective sentence gadget and hence we can erase the whole sentence gadget. By Lemma 5.0.6, we can erase the whole implication gadget except possibly the implicated sentence. Iterating this process with every sentence in S_0 we can erase all the implication gadgets and the implicated sentences. Thus, we have erased the whole complex.

Conversely, let K_0 the set of triangles such that $K \setminus K_0 \rightsquigarrow \emptyset$. From Figure 5.3 we notice that if a triangle lies on one of the tubes then the whole sentence gadget can be erased. Hence without loss of generality let's assume the triangles lie only on the spheres. Define S_0 as such set of spheres. Thus, the corresponding set of sentence gadgets is an axiom set of size at most k.

For one hand, every sentence gadget can be constructed using the same number of triangles used in the sentence with most incident tubes. Aditionally, every tube can be a polygonal cylinder with 8 triangles stretched as needed. Since we have a finite number of senteces this complex is well defined. From here, it is not difficult to note that we can construct this complex using a number of 2-simplices that is polynomial in the size of the sentences in S. Therefore, MINIMUMAXIOMSET $\leq_{\text{FPT}}$ ERASABILITY . ♢

With respect to this result, if we want to prove fixed parameter tractability of ERASABILITY , the parameter must be different from the natural parameter.

It is still possible to solve the ERASABILITY and MORSE MATCHING problems efficiently whenever we have some control over a different parameter, specifically it can be shown that these problems are fixed parameter tractable in the treewidth of the spine of the input simplicial complex. The treewidth of the dual graph is a common and useful parameter when working with triangulated manifolds, as it closely interacts with the topology of the underlying manifold. Formally, the treewidth is defined as follows.

Definition 5.0.8 (Treewidth) *A tree decomposition of a graph G is a tree T whose nodes $\{X_i | i \in I\}$ are called bags. Each bag X_i is a subset of nodes of G, and we require that:*

- *node coverage: every node of G is contained in at least one bag X_i,*
- *arc coverage: for each arc of G, some bag X_i contains both its endpoints,*
- *coherence: for all bags X_i, X_j and X_k of T, if X_j lies on the unique simple path from X_i to X_k in T, then $X_i \cap X_k \subseteq X_j$.*

The width of a tree decomposition is defined as $\max|X_i| - 1$*, and the treewidth of G is the minimum width over all tree decompositions.*

For bounded tree width, computing a tree decomposition of a graph $G = (V, E)$ of width $\leq k$ has running time $O(f(k)|V|)$ due to an algorithm by Bodlaender [Bod93].

In [BLPaS16] it is proved that if the spine of the Hasse diagram has a treewidth bounded by a number k, it is used as a parameter, and the erasability problem is FPT tractable and a procedure is described to find the Morse matching in time $O(4^{w^2+w} w^3 \log(w) n)$ where w trewidth of spine of the Hasse diagram and n is the number of nodes in this spine. In fact, the problem of ERASABILITY is restated as a problem to find an alternating cycle-free matching in the spine.

Here we've decided to use a different parameter. We will first study the problem by giving a simple complex stratification and using the number of 1-cells in the

1-dimensional stratum as the parameter. Then we will try to change the parameter by the number of connected components of that strata.

Suppose we have a stratification of a cohomologically stratifiable space. Each of the two strata is guaranteed to be a manifold so we can apply the algorithm 2 fixing the critical points in the 1-dimensional substrata that are contained in the clousure of this 2-dimensional stratum. On the other hand, in the 1-dimensional strata the calculation of a Morse matching can be done by fixing any covering tree with root and taking the remaining edges and the root we obtain all the critical cells. Finally, in the 0-stratum we only have vertices and therefore each of them automatically becomes a critical point. Next we must reconcile the critical cells in the 0- and 1-dimensional strata with the critical cells in the 2-dimensional strata.

Claim. Erasability for pure 2-dimensional complexes is FPT in the max degree of connected components multiplied by the number of components.

We propose an algorithm to compute an optimal Morse matching:

1. Compute the stratification of S of X.
2. Compute an optimal Morse matching M_S for the spine $H_1(S)$ of S (by brute force): Enumerate candidate matchings, for each verify if it is acyclic, and if so count the number of unmatched 2-strata
3. For each $(\tau;\sigma) \in M_S$, obtain a sequence for collapsing the triangles in σ starting from a triangle t_σ adjacent to τ, using a spanning tree of the dual graph of the triangulation of σ.
4. For each 2-stratum σ not paired, obtain a sequence for collapsing the triangles in σ starting with the removal of an arbitrary (critical) triangle t_σ, using a spanning tree of the dual graph of the triangulation of σ.
5. A global sequence of collapses (and hence a matching) is obtained by concatenating the sequences in the 2-strata according to an ordering of the 2-strata corresponding to MS.
6. Compute an optimal Morse matching for the graph resulting from the triangle removals/collapses.

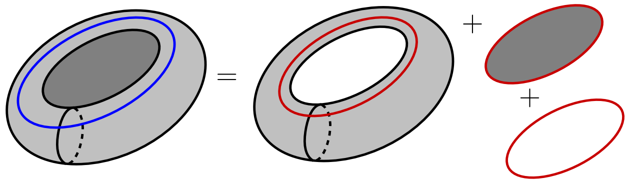

Figure 5.4: A torus with a disk glued to its inner equator. The blue circle is contractible.

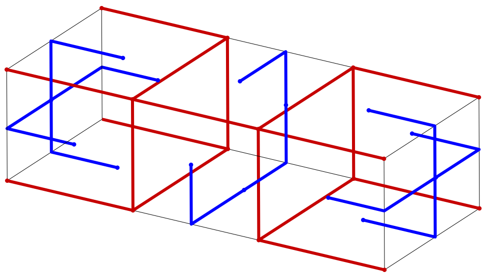

Figure 5.5: Three cubes with their respective dual spanning trees and the global spanning tree U.

For the running time, let s be the size of $H_1(S)$ and n denote the size of the triangulation. Also, Let $s_1; s_2$ be the number of 1- and 2-strata. An upper bound on the number of possible matchings is $2^{s_2} s_1!$: 2^{s_2} is the number of ways of choosing 2-strata to be matched, and $s_1!$ is the number of permutations of the 1-strata, each permutation determines a match of the 1-strata to the 2-strata selected. Verifying if a matching is acyclic and counting unmatched 2-strata take linear time in the number of strata. Thus the total time of step 2 is

$$O(2^s \cdot s!)$$

All the other steps take linear time in n, so the total time is

$$O(2s \cdot s! + n)$$

Acknowledgements

I would like to make a special recognition to all those people who helped me in one way or another to complete this work. In particular, I would like to thank my thesis advisor Edgar Ramos for his patience and help. To Juan Diego Velez for guiding me during my undergraduate and graduate years, for including me in many of his projects and for all the help he gave me.

I thank my friends Laura, Jaime and all those who supported me from the beginning of my career. To my fellow students for the hours of discussion and exchange of ideas, and more importantly, for the many hours of fun during these last 7 years of study. Thanks to all of you.

Jeferson Zapata

Medellín, Colombia, May 2019

Bibliography

[BLPaS16] Benjamin A. Burton, Thomas Lewiner, João Paixão, and Jonathan Spreer. Parameterized complexity of discrete morse theory. *ACM Trans. Math. Softw.*, 42(1), March 2016.

[Bod93] Hans L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, page 226–234, New York, NY, USA, 1993. Association for Computing Machinery.

[Bro30] Arthur B. Brown. Relations between the critical points of a real analytic functions of n independent variables. *American Journal of Mathematics*, 52(2):251–270, 1930.

[Cha00] Manoj Chari. On discrete morse functions and combinatorial decompositions. *Discrete Mathematics*, 217:101–113, 04 2000.

[DF12] Rodney G Downey and Michael Ralph Fellows. *Parameterized complexity*. Springer Science & Business Media, 2012.

[EH10] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. 01 2010.

[FG06] Jörg Flum and Martin Grohe. Parameterized complexity theory. 2006. *Texts Theoret. Comput. Sci. EATCS Ser*, 2006.

[For98] Robin Forman. Morse theory for cell complexes. *Advances in Mathematics*, 134(1):90 – 145, 1998.

[JP06] Michael Joswig and Marc E Pfetsch. Computing optimal morse matchings. *SIAM Journal on Discrete Mathematics*, 20(1):11–25, 2006.

[Kar72] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[LLT03a] T. Lewiner, H. Lopes, and G. Tavares. Optimal discrete morse functions for 2-manifolds. *Computational Geometry*, 26(3):221–233, 2003.

[LLT03b] T. Lewiner, H. Lopes, and G. Tavares. Towards optimality in discrete morse theory. *Experimental Mathematics*, 12(3):271–285, december 2003.

[LTR+02] Hélio Lopes, Geovan Tavares, Jarek Rossignac, Andrzej Szymczak, and Alla Safanova. Edgebreaker: a simple compression for surfaces with handles. In *Proceedings of the seventh ACM symposium on Solid modeling and applications*, pages 289–296, 2002.

[MHSS02] Y. Matsumoto, K. Hudson, American Mathematical Society, and M. Saito. *An Introduction to Morse Theory*. Europe and Central Asia Poverty Reduction and Economic Manag. American Mathematical Society, 2002.

[Mil73] John Milnor. *Morse theory.(AM-51)*, volume 51. Princeton university press, 1973.

[Nan19] Vidit Nanda. Local cohomology and stratification. *Foundations of Computational Mathematics*, pages 1–28, 2019.

[Nic11] L. Nicolaescu. *An Invitation to Morse Theory*. Universitext. Springer New York, 2011.

[OEG96] Omer Eğecioğlu and Teofilo F. Gonzalez. A computationally intractable problem on simplicial complexes. *Computational Geometry*, 6(2):85 – 98, 1996.

[PM81] A. Paz and S. Moran. Non deterministic polynomial optimization problems and their approximations. *Theoretical computer science*, 15(3):251–277, 1981.

[PP12] BS Panda and Dinabandhu Pradhan. Acyclic matchings in subclasses of bipartite graphs. *Discrete Mathematics, Algorithms and Applications*, 4(04):1250050, 2012.

[Sip13] Michael Sipser. *Introduction to the Theory of Computation*. Course Technology, Boston, MA, third edition, 2013.

[Tar83] Robert Endre Tarjan. *Data structures and network algorithms*, volume 44. Siam, 1983.