VNIVERSIDAD NACIONAL AVTONOMA DE MEXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

A STUDY OF THE WAIT-FREE SOLVABILITY OF WEAK SYMMETRY BREAKING AND RENAMING

T E S I S

QUE PARA OBTENER EL GRADO DE:

DOCTOR EN CIENCIAS (COMPUTACIÓN)

P R E S E N T A:

ARMANDO CASTAÑEDA ROJANO

DIRECTOR DE TESIS

DR. SERGIO RAJSBAUM GORODEZKY

México, D.F. **2010.**

Agradecimientos

Quiero agradecer a Sergio Rajsbaum por motivarme y brindarme su apoyo incondicional durante el desarrollo de esta tesis.

Quiero agradecer a Javier Bracho, Eli Gafni, José Galaviz, Maurice Herlihy, Armando Ronquillo, Jorge Urrutia y Carlos Velarde por sus observaciones y comentarios que enriquecieron este trabajo.

Quiero agradecer a mi familia y amigos por apoyarme siempre que los he necesitado.

Quiero agradecer al Consejo Nacional de Ciencia y Tecnología (CONACYT) por apoyarme económicamente durante mis estudios.

Abstract

In the *M-renaming* task $n + 1$ processes start with unique input names taken from a large space $1, \ldots, N$, and must choose unique output names taken from a smaller name space $1, \ldots, M$, $n < M < N$. To rule out trivial solutions, a renaming protocol must be *anonymous*: The value chosen by a process can depend on its input name and on the execution, but not on its specific process id. In the *weak symmetry breaking* (WSB) task $n + 1$ processes start with unique input names taken from a large space $1, \ldots, N$, and must choose binary output values, 0 or 1. It is required that not all processes decide either 0 or 1. As for renaming, a WSB protocol must be anonymous. Prior research has proved that WSB and $2n$-renaming are equivalent tasks.

Various protocols that solve M-renaming for $M \geq 2n + 1$ in the asynchronous wait-free read/write shared memory model have been presented, where *wait-free* means that in every execution, each non-faulty process decides an output value, regardless of delays and failures. Also, several proofs of a lower bound stating that no such protocol exists when $M < 2n + 1$ have been published. All these proofs use the topological approach to distributed computing; some of them are based on algebraic topology and the others use a mixture of combinatorial and algebraic topology techniques.

This thesis is a detailed study of the solvability of the WSB task in a wait-free setting, explaining the relation between WSB and renaming. The study is done both from a combinatorial and algebraic topology perspective. The more concrete combinatorial perspective allows readers with no background in topology to understand the results, while the more abstract algebraic topology perspective shows that this very powerful and mature mathematical branch, is a natural framework to study WSB. Some of the results it contains are known, but this is the first time they are presented in detail.

The first result of this thesis is that for some values of n, WSB is not solvable in the asynchronous wait-free read/write shared memory model. The proof is fully combinatorial. Since WSB and $2n$-renaming are equivalent, this impossibility result gives the first, fully combinatorial renaming lower bound proof stating that M-renaming is not wait-free solvable if $M < 2n + 1$. Also, the thesis shows that, for the other values of n, WSB is indeed solvable, i.e., there exists a protocol that solves WSB in the asynchronous wait-free read/write shared memory model. This result and the equivalence between WSB and $2n$-renaming imply that the previous renaming lower bound proofs are incorrect.

More precisely, the main result in this thesis states that there exists a wait-free WSB protocol for $n+1$ processes if and only if the integers in the set $\{\binom{n+1}{i+1} | 0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$ are relatively prime. For example, such a protocol exists for $n = 5, 9, 11, 13, 14$, and does not exist for the other values smaller than 14.

Contents

List of Figures

Chapter 1

Introduction

A *distributed system* consists of autonomous computational devices, called *processes*, that communicate with each other using a medium, usually by sending messages throughout a fully connected network or by writing in and reading from a shared memory. The processes can be *synchronous* or *asynchronous*. In the former, the processes execute in lock-step manner all steps of computation, while in the latter there is no restriction on the relatively speed of processes. Moreover, the processes are prone to fail. The failures range from simple crashes, where a faulty process does not execute any more steps of computation, to Byzantine failures, where a faulty processes can exhibit any behavior.

A *distributed task*, or simply *task*, is an input/output relation between collections of processes' inputs and outputs. Roughly speaking, given an input configuration for the processes, the task defines all valid output values. For example, in the *consensus* task each process proposes a value and it is required that all non-faulty processes decide a unique value among all proposed values. Therefore, if all processes propose the same value v then v must be elected as a decision by each non-faulty process.

A breakthrough result in distributed computing was proved in 1983 (journal version [39]) by Fischer, Lynch and Paterson. They showed that it is impossible to deterministically achieve consensus in a completely asynchronous system in which just one process can fail, and even if the failure is of the mildest form of just crashing. As mentioned by Attiya et al. [7] in 1987, due to this, and stronger versions of this result [35], it became a widely held folk assumption that it is impossible to solve any non-trivial task that requires coordination by processes, in an asynchronous environment with failures. Today we know that there are infinitely many such tasks [18], but the renaming task was hence proposed [7] and became the first example of a non-trivial task that could be solved the presence of asynchrony and crash failures.

1.1 The Renaming Task

Let us suppose we want to design a protocol for autonomous airplane control (this example was used by Nir Shavit in his Godel Prize presentation in 2004). There is a large number of airplanes identified by their flight numbers, of which $n+1$ are heading to pick distinct altitudes once they are close to each other, as in Figure 1.1. How many altitudes are needed to solve this problem? Obviously at least $n+1$, one for each airplane. In presence of reliable communication and synchrony, $n+1$ altitudes are enough: the airplanes exchange their flight numbers, sort them locally, and the

i-th altitude is assigned to the i-th flight number. However, in presence of failures and asynchrony, an airplane cannot wait indefinitely to hear from the other airplanes. For this environment, consider the following protocol for two airplanes, A and B. After A and B send to each other their flight numbers, there are essentially three possible scenarios:

1. A does not hear about B (maybe because of delays) but B hears about A.
2. B does not hear about A but A hears about B.
3. A and B hear about each other.

The difficulty comes from the fact that in scenario 1, B does not know if A heard him or not; namely, for B, the situation could be either 1 or 3. A symmetric situation occurs for A with scenarios 2 and 3. The graph in Figure 1.2 represents these uncertainties.

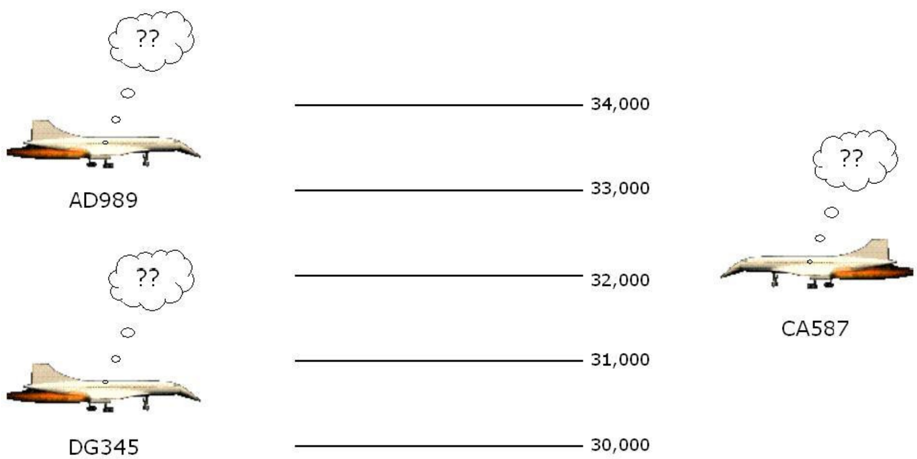

Figure 1.1: Autonomous airplane control.

For the protocol, let us assume that whenever an airplane fails to hear about any other airplane (either because of failures or because no other plane is approaching), it picks altitude 1. This assumption can be made without loss of generality because the space of flight numbers is big, hence for every protocol there exist two airplanes such that each one of them picks the same altitude when it does not hear about the other. Thus, A and B pick altitude 1 in scenarios 1 and 2 respectively. In scenario 3, B is aware that A may not have heard its flight number, and to avoid conflicts, picks say, altitude 2. Then A has no choice but to pick altitude 3 in scenario 2, whenever it gets B's flight number. The previous protocol solves the airplane control problem using three distinct altitudes for two airplanes, in an environment where it is possible to avoid a 4-th scenario: both A and B participate but they don't hear from each other. Such an environment is a shared memory where processes (instead of airplanes) communicate via read/write registers. In Figure 1.2, that 4-th scenario would be represented as an edge closing a cycle.

Can we solve the problem with two altitudes? The previous arguments show that the answer is no, if the protocol consists of just one communication exchange. It is possible to show that the answer is still no, no matter how many communication rounds are tried, because it is known since 1988 [17] (the conference version of [18]) that the uncertainty graph is always connected between the two endpoints representing the situation where A and B do not hear from each other. In those extremes, A and B still need to pick altitude 1, and again it would be impossible for them to pick only altitudes 1 or 2 in the internal nodes, because an edge with equal decisions in its endpoints would be unavoidable.

Considering the airplane control example, how many altitudes are needed in general for $n+1$ airplanes? More formally, in the M*-renaming* task [8] $n+1$ processes, $p_0, p_1, \ldots, p_n$, start with unique input names taken from a large space $1, \ldots, N$, and must choose unique output names taken from a smaller name space $1, \ldots, M$, $n < M < N$. Clearly, renaming is trivial if process p_i picks output name $i+1$. To rule out trivial solutions, a protocol must be *anonymous*: the value chosen by a process can depend on its input name and on the execution, but not on its specific process id.

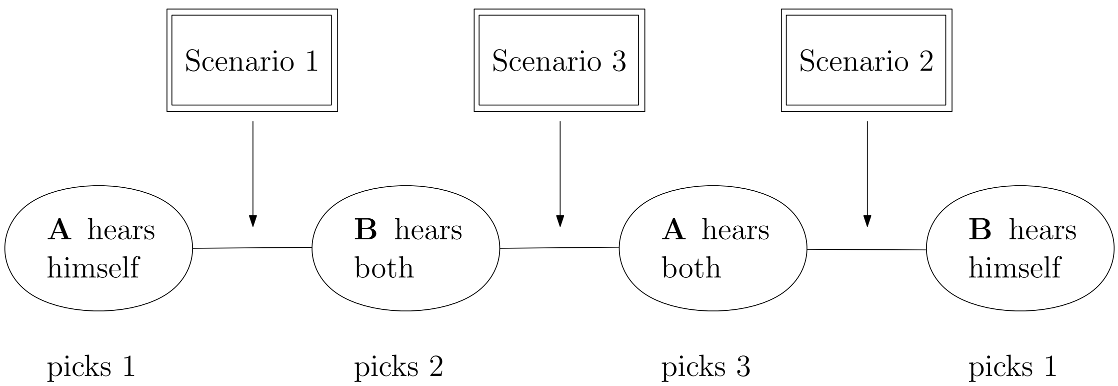

Figure 1.2: Uncertainty graph for 2 airplanes and one communication exchange.

The renaming task has been intensively studied since Attiya et al. proposed it in 1987 [7] (conference version of [8]). An initial motivation was to explore the border between solvable and unsolvable tasks in an asynchronous environment. However, it turns out that renaming is important also for practical reasons, for example, the complexity of some protocols depends on the size of the initial name space, and thus using renaming as a preprocessing stage, the complexity of those protocols can be reduced.

Attiya et al. [8] presented a wait-free $(2n+1)$-renaming protocol in the asynchronous read/write shared memory model with crash failures, where *wait-free* means that in every execution any non-faulty process produces an output value, regardless of delays or failures by other processes. Actually, this protocol was first presented in the message passing model and uses output name space $1, 2, \ldots, n+1+t$, where $t < (n+1)/2$ denotes the number of processes that may crash. However, it can can be extended [16] to the asynchronous wait-free read/write shared memory mode where $t = n$. Attiya et al. [8] also showed that there is no wait-free M-renaming protocol for $M \leq n+2$. However, there was neither protocol nor proof of impossibility for the range $n+2 < M < 2n+1$. Then, Herlihy and Shavit discovered [53] (conference version of [55]) a connection between distributed computing and topology and used it to achieve a lower bound stating that no wait-free M-renaming protocol exists when $M < 2n+1$, i.e., the minimum number of output names needed to solve renaming is equal to double of the number of processes minus one.

1.2 Distributed Computing and Topology

In 1993 Herlihy and Shavit [53] discovered a deep connection between distributed computing and topology, and provided a new perspective on the area; see also Saks and Zaharoglou [70], Borowsky and Gafni [19]. They showed that the executions of any wait-free protocol in the asynchronous read/write shared memory model, can be represented by a *complex*, a topological object, that "has no holes". A complex is made of simplexes, and a *simplex* is a generalization of the notion of a line or triangle for any dimension. In Figure 1.3 a 1-dimensional complex and a 2-dimensional complex are depicted.

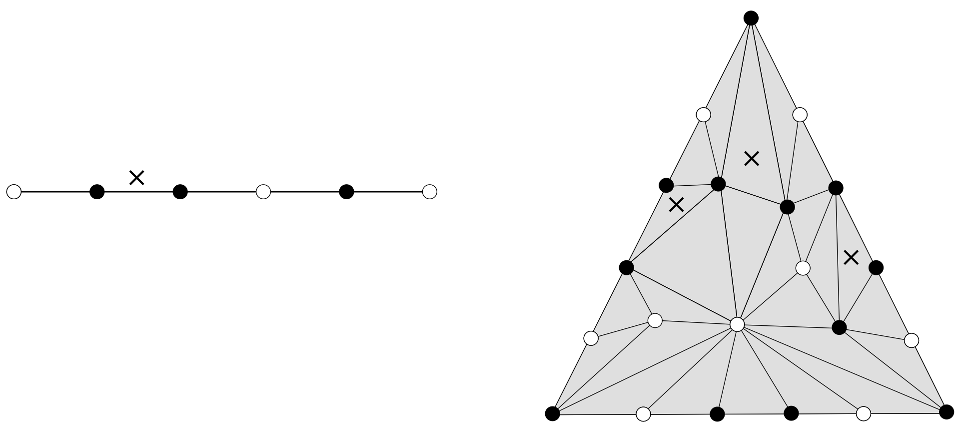

Figure 1.3: Two subdivided simplexes.

The executions of a wait-free protocol for two processes that start with a specific input configuration (assignment of input values), can be represented by a 1-dimensional complex that is a subdivided line. Each edge of the line is a simplex that corresponds to a set of executions that are indistinguishable to the two processes (the local state of a process is the same at the end of any of these executions). The two vertexes of the edge are labeled with the local states of the two processes, respectively. A protocol that executes more steps, will induce a line with a finer subdivision (more edges). The endpoints of the line correspond to executions where a process runs solo, namely, it decides without seeing any value written in the shared memory by the other process. Therefore, the graph in Figure 1.2 is the complex that represents the simple protocol described in Section 1.1. Similarly, the executions of a protocol for three processes, are represented by a complex that is a subdivided triangle, as in Figure 1.3. In general, an n-dimensional complex, which a subdivision of an n-dimensional simplex, is used to represent the executions of a protocol for $n+1$ processes with an input configuration. Moreover, this subdivision is *chromatic* in a sense that the $n+1$ vertexes of each of its n-simplexes, are labeled with the local states of the processes at the end of the execution, respectively

Herlihy and Shavit [55] presented the Asynchronous Computability Theorem and the Anonymous Computability Theorem that fully characterize the tasks that are wait-free solvable in the asynchronous read/write shared memory model. Intuitively, the new insight to distributed computing is that the solvability (or time complexity) of a task, has a topological nature, it depends on whether there exists a complex (representing a protocol) that can be mapped into the task complex (the topological representation of the task), respecting the task's input/output specification.

1.3 The WSB Task

The renaming lower bound proof in [53] was the first of four lower bound proofs [14, 56, 53, 55], all closely related and based on algebraic topology, stating that no wait-free M-renaming protocol exists when $M < 2n + 1$. The second proof appeared in [56], where an algebraic methodology is developed to obtain lower bounds for various tasks. The journal version [55] of [53] includes the third one, and is based on the proof in [56]. The last proof appeared in [14]. That paper aimed to provide a combinatorial version of the renaming lower bound that could be accessible to a reader that is unfamiliar with algebraic topology. However, the crucial step of the proof relies on an algebraic topology lemma in [56]. Thus, the question of a fully combinatorial renaming lower bound proof was left open.

Intuitively, all these proofs rely on proving the following claim: any chromatic subdivision of an n-dimensional simplex, with a binary coloring on its vertexes that is symmetric on the boundary, have at least one monochromatic n-dimensional simplex, i.e., a simplex with the very same color at its vertexes. For example, the reader can easily verify that, in dimension $n = 1$, a line subdivided into an odd number of edges with the same binary color at its ends (its boundary), has at least one monochromatic edge; in the subdivided line of Figure 1.3 there is exactly one monochromatic edge. For dimension $n = 2$, any chromatic subdivision of a triangle with a symmetric binary coloring on the boundary, contains at least one monochromatic triangle; the subdivided triangle in Figure 1.3 has three monochromatic triangles.

The existence of monochromatic simplexes is closely related to the renaming task. Consider a protocol solving the M-renaming task, and the corresponding protocol complex. Each vertex of this complex is labeled with the local state of a process at the end of an execution, hence it has an associated output name. It was observed in [42, 55] that it is convenient to label each vertex of this complex with the parity of its output name. The anonymity requirement for renaming implies that these values induce a binary coloring that is symmetric on the boundary of the protocol complex. Moreover, if $M = 2n$ then no n-dimensional simplex can be monochromatic because there are exactly n even and exactly n odd names within the range $1, \dots, 2n$. This observation motivated to define the weak symmetry breaking task.

In the *weak symmetry breaking* (WSB) task [42] (called *reduced renaming* in [55]), $n+1$ processes, $p_0, p_1, \dots, p_n$, start with unique input names taken from a large space $1, \dots, N$, and must choose binary output values, 0 or 1. It is required that not all processes decide either 0 or 1. Obviously, WSB is trivial if process p_i picks 0 if i is even, and 1 if i is odd. Thus, as for renaming, a WSB protocol must be anonymous. The WSB task is a "weak" version of the *strong symmetry breaking* (SSB) task [19] (called $(n + 1, n)$-set-test-and-set in [19]), in which it is required that that in every execution at least one process decides 0, in addition to the requirement that not all processes decide either 0 or 1.

Similarly as how it is showed in Section 1.1 that two processes cannot solve renaming with 2 output names, it can be showed that two processes cannot solve WSB. Consider first a protocol in which the processes execute just one round of communication. One can assume, without loss of generality, that if a process runs solo, it picks 0. Thus, in Figure 1.2, A and B pick 0 in scenarios 1 and 2, respectively. In scenario 3, B has no choice but to pick 1. Now A in scenario 3 cannot make a good decision: if it decides 0, there is conflict in scenario 2, and if it decides 1, there is a conflict in scenario 3. Therefore, A and B cannot solve WSB in one round of communication. Moreover, no matter how many rounds of communications they do, the protocol complex always is connected, hence a monochromatic edge is unavoidable.

WSB and M-renaming are closely related for the specific value $M = 2n$, namely, a $2n$-renaming protocol gives a solution for WSB: since the range $1, \ldots, 2n$ has exactly n even names and exactly n odd names, each process can decide the parity of the output name received from a $2n$-renaming protocol. Moreover, the other direction is true, i.e., a WSB protocol gives a solution for $2n$-renaming. Generally speaking, using a WSB protocol, the processes can be partitioned into two disjoint groups, S_0 and S_1, each one of size at most n, as not all processes receive the same output value from WSB. It turns out that these two groups can independently solve renaming into two disjoint output name spaces of size $2|S_0| - 1$ and $2|S_1| - 1$, respectively, giving a final output name space of size $2n$, since $|S_0| + |S_1| = n + 1$. Therefore, the WSB and $2n$-renaming are equivalent tasks [42].

1.4 Contributions

This thesis is a detailed study of the solvability of the WSB task in a wait-free setting, explaining the relation between WSB and renaming. The study is done both from a combinatorial and algebraic topology perspective. The more concrete combinatorial perspective allows readers with no background in topology to understand the results, while the more abstract algebraic topology perspective shows that this very powerful and mature mathematical branch, is a natural framework to study WSB. Some of the results it contains are known, but this is the first time they are presented in detail. From a combinatorial topology view, the major contributions are the following two results:

1. For certain non-exceptional values of n, any chromatic subdivision of an n-dimensional simplex, with a symmetric binary coloring, contains at least one monochromatic n-dimensional simplex, where *exceptional* means that the integers in the set $\{\binom{n+1}{i+1} | 0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$ are relatively prime.

2. For the other exceptional values of n, there exists chromatic subdivisions of n-dimensional simplexes, with a symmetric binary coloring and no monochromatic n-dimensional simplexes.

As Section 1.3 explains, if any chromatic subdivision of an n-dimensional simplex with a symmetric binary coloring contains at least one monochromatic n-dimensional simplex, then no protocol can solve the WSB task. Therefore, result 1 implies that if n is non-exceptional, there is no wait-free WSB protocol, hence there is no wait-free M-renaming protocol for $M < 2n + 1$, as WSB is equivalent to $2n$-renaming. This is the first, fully combinatorial renaming lower bound proof, closing the open question left in [14] (except that the lower bound applies only to non-exceptional values of n).

Result 2 states that if n is exceptional, there exist chromatic subdivided simplexes with a symmetric binary coloring and no monochromatic n-dimensional simplexes. The smallest exceptional value is $n = 5$ (6 processes). An implication of this result is that, while the previous renaming lower bound proofs are correct in dimensions 1 and 2 (as in Figure 1.3), dimension 3 (a subdivided tetrahedron), and dimension 4, they are incorrect in dimension 5. These subdivisions for exceptional n, and the Anonymous Computability Theorem [55], or the Simplex Convergence algorithmic version [21], imply that there exists a wait-free protocol that solves WSB, and thus there exists a wait-free $2n$-renaming protocol, for exceptional n.

More precisely, the main result in this thesis states that there exists a wait-free WSB protocol for $n + 1$ processes if and only if n is exceptional. For example, such a protocol exists for $n = 5, 9, 11, 13, 14$, and does not exist for the other values smaller than 14.

Also, this thesis studies the relation of results 1 and 2 with classic topics in algebraic topology. More precisely, it shows the relation of these results with equivariant maps between chain complexes. Intuitively, a chain complex is an algebraic structure associated with a complex, and an equivariant map is a map that holds certain properties of symmetry. From an algebraic perspective, the solvability of WSB depends on whether there exists an equivariant map from the chain complex of an n-simplex, representing the initial state of the system, to the chain complex that represents the valid outputs for WSB.

A preliminary version of these results was presented in the 27th Annual ACM Symposium on Principles of Distributed Computing (PODC) 2008 [27]. That paper won the best paper student award. The journal version of the impossibility of WSB for non-exceptional n, appears in [28], by invitation as one of the best papers in PODC 2008. The journal version of the WSB protocol for exceptional n, has been submitted for publication. A preliminary version is in [29].

1.5 Organization

Chapter 2 presents a detailed description of the model of computation, the definition of renaming and WSB, and the equivalence between WSB and $2n$-renaming. Chapter 3 shows how the topological approach to distributed computing can be used to reduce the question of solvability of WSB, to a topological question. Chapter 4 presents the main combinatorial tools used in proving two theorems that give the impossibility and possibility results for WSB, both described in Chapters 5 and 6, respectively. Chapter 7 shows where the mistake is in previous renaming lower bounds proofs, and presents the relation of the combinatorial topology results in Chapters 5 and 6, with algebraic topology. There is a large amount of previous work related to renaming and WSB. Chapter 8 presents a panoramic view of most of this work. The aim of this chapter is to give a more concrete idea about the area to the reader. Conclusions and future work appear in Chapter 9. Some proofs appear in Appendixes A and B.

Chapter 2

Model of Computation, Renaming and WSB

This chapter presents the model of computation used in the rest of this thesis, the usual asynchronous wait-free read/write shared memory model. See textbooks [9, 59] for more details about the model. Also it presents the formal definition of renaming and WSB, and proves that WSB and $2n$-renaming are equivalent tasks. In addition, it shows that any protocol that solves either WSB or renaming, can be transformed into a comparison-based protocol that solves the same task. Generally speaking, a comparison-based protocol is restricted to use only comparison operations. Finally, the chapter shows that the comparison-based result is related with the well known Ramsey's Theorem.

2.1 Model of Computation

System and Executions. A *system* consists of $n+1$ *asynchronous* processes with distinct *id*'s in $ID^n = \{0, \ldots, n\}$. A *process* is a deterministic state machine with a set of *local states* S and two subsets of S called *initial states* and *output states*, respectively. The processes communicate by using a shared memory with a finite number of *single-writer multi-reader atomic registers*. No assumption is made regarding the size of the registers, thus we can assume that process with *id* i has a single register labeled i to which it can write its entire state. The process with *id* i has two atomic operations available to it: $write_i(v)$ that writes the value v into the register labeled i, and $read_i(j)$ that returns the current value in the register labeled j. A *step* is performed by a single process, which executes one of its two available operations, *read* or *write*, performs some local computation and then changes its local state.

A *configuration* of the system consists of the local states of the processes and the content of the registers. For a configuration C, let $state_i(C)$ denote the local state of process with *id* i in C. An *initial configuration* is a configuration in which all local states are initial states and all registers are set to $\perp$, a special value that cannot be written by any process. An *output configuration* is a configuration in which all local states are output states.

An *execution* of the system is a, possibly infinite, alternating sequence of configurations and steps $\alpha = C_0, s_0, C_1, s_1, C_2, \ldots$, where C_0 is an initial configuration and C_{k+1} is the result of applying the step s_k to C_k, for $k \geq 0$. The *schedule* of α is the sequence of steps $s_0, s_1, \ldots$. The *view* of a process with *id* i in α, denoted $\alpha|i$, is the sequence of local states $state_i(C_0), state_i(C_1), \ldots$. The *participating set* of an execution is the set of processes that take at least one step in the execution. Two executions are *indistinguishable* for a set of processes if all processes in the set have the same view in both executions.

Protocols and Tasks. The state machine of a process p with *id* i, models a *local protocol* $\mathcal{P}_i$ that determines p's next step. All local protocols are identical, i.e., processes have identical state machines which do not depend on a specific *id*. If the behavior of a local protocol has to depend on its *id*, it must be encoded as part of the input. A *protocol* is a collection $\mathcal{P}$ of local protocols $\mathcal{P}_0, \ldots, \mathcal{P}_n$.

Each process has two distinguished components, *input* and *output*, that allow the system to model decision tasks. The initial states differ only in the value of the input component, and the input component never changes. The output component contains initially $\perp$, and once a process reaches a local state in which a non-$\perp$ value is written in it, the output component never changes. When that happens, we say that the process has *decided*. The output states are the states with non-$\perp$ output values.

A view of a process p in a finite execution α is *final*, if there is a configuration in α in which p decides. Notice that the output of p is not $\perp$ in any extension of α, since the output component cannot be over-written. A final view of p in α is *minimal* if none of its prefixes is final. In other words, the minimal final view of p in α is the prefix of the view of p up to the first configuration in which p decides. A finite execution α is *final* (*minimal final*) if the view of each process is a final (minimal final) view.

A *task* $\triangle$ has a domain I of input values and a domain O of output values, and $\triangle$ specifies the outputs values that the processes can decide, for each input assignment to the processes.

A protocol *solves* a task $\triangle$ if any finite execution α can be extended to an execution α' in which all processes decide valid output values specified by $\triangle$. That is, the outputs in α can be extended to outputs for all processes that are valid for the inputs in α.

In a given execution, a process is *faulty* if it performs a finite number of events and it has not decided. A protocol is *t-resilient* if it tolerates t or fewer failures, i.e., in any execution in which at most t processes fail, the non-faulty processes decide an output value. Only *wait-free* protocols are considered, namely $t = n$. That is, in every execution, each non-faulty process decides an output value, regardless of delays or failures.

A protocol is *full-information* if every process "remembers everything" and writes "everything it knows". Formally, it writes its entire local state each time it executes a write operation. We say that a protocol is in *standard-form* if processes proceed in a sequence of rounds. In a round, each process first executes a write operation and then reads all registers, in some fixed order. Since efficiency is not an issue for this thesis, only full-information and standard-form protocols are considered.

For a full-information and standard-form protocol, the local state of a process p can be represented as nested arrays: the initial state of p is its input, and later states have the form $(reg_0, \ldots, reg_n)$, where reg_i, $0 \leq i \leq n$, is the value p reads from the register labeled i. Therefore, for a full-information and standard-form protocol, the only interesting state transition is if a process decides a value and what value.

Anonymous and Comparison-Based Protocols. A protocol is *anonymous* if the following holds. Let E_1 be any execution of the protocol and π be any permutation of ID^n. Then the execution E_2 obtained by replacing *id* x of each step, by $\pi(x)$, is an execution of the protocol. Therefore, if x decides z in E_1 then $\pi(x)$ decides z in E_2. In other words, the output of a process does not depend on its *id*, it only depends on its input and on the execution. The rest of this thesis focuses on anonymous protocols.

A protocol is *comparison-based* if for any execution E_1 with inputs $i_0 < \ldots < i_k$ to the participating processes, and any valid inputs $j_0 < \ldots < j_k$, the execution E_2 obtained by replacing each occurrence of i_ℓ of each local state by j_ℓ, $1 \leq \ell \leq k$, is an execution of the protocol. Therefore, processes with inputs i_ℓ and j_ℓ in E_1 and E_2, respectively, decide the same output. One can think a comparison-based protocol as a protocol in which processes can only use comparison operations $(<, =, >)$. Observe that a comparison-based protocol is not necessarily anonymous.

Immediate Snapshot Executions. The immediate snapshot executions (ISE) is a subset of all possible executions in the asynchronous wait-free read/write shared memory model. The executions in ISE have a structure that makes easy to analyze them. In addition, ISE captures the power of the model of computation. In Chapter 3 this subset of executions will be used to achieve a solvability condition for WSB.

Formally, an *immediate snapshot* (IS) [19, 20, 70] execution consists of a sequence of rounds. The i-th round is specified by a non-empty set of processes which denotes the processes active in round i. Active processes first perform, one by one, a write operation, and then read all registers. Intuitively, each round consists of a concurrent write by every active process, followed by a concurrent atomic snapshot of the shared memory.

For example, Figure 2.1 shows two IS executions in which each process is active in exactly one round. Notice that the views of p_1 and p_2 are the same in both execution. Moreover, p_0 is the unique process that distinguishes between α_1 and α_2: in α_1, p_0 is not aware that p_2 participates in the execution, while in α_2 it reads from the memory what p_1 and p_2 have written.

Round :	1	2	3
p_0 :		wr	
p_1 :	wr		
p_2 :			wr

α_1

Round :	1	2	3
p_0 :		wr	
p_1 :	wr		
p_2 :		wr	

α_2

Figure 2.1: Two IS executions.

It is clear that if a protocol that solves a task, solves the same task in the subset ISE. In [20, 22] it is proved that ISE can be wait-free simulated in the asynchronous read/write shared memory model, hence any protocol that solves a task in ISE, wait-free solves the same task in the whole set of executions.

Theorem 2.1.1 *There exists a wait-free protocol that solves a task* Δ *if and only if there exists a wait-free protocol that solves* Δ *in ISE.*

2.2 Renaming and WSB

In the *M-renaming* task [8] the processes start with unique *input names* taken from a large input space $1, \ldots, N$, $N \geq n+1$, and must choose unique *output names* taken from a smaller output space $1, \ldots, M$, $n+1 \leq M < N$. As only anonymous protocols are considered, the trivial solution in which process with *id* i picks output name i, is not allowed.

In the *weak symmetry breaking* (WSB) task [42] the processes start with unique inputs taken from a large input space $1, \ldots, N$, $N \geq n+1$, and the output values are 0 or 1. It is required that in every execution in which all processes decide, at least one process decides 1 and at least one process decides 0. As for renaming, since only anonymous protocols are considered, trivial solutions such as the one in which processes with even *id* decide 0 and processes with odd *id* decide 1, is not allowed.

Attiya et al. present in [7] a comparison-based wait-free protocol, $ABDPR$, that solves $(2n+1)$-renaming in the asynchronous read/write shared memory model. The input name space of $ABDPR$ is $1, \ldots, N$, for any $N \geq n+1$. In what follows, $ABDPR$ is used as a building block for proving various results, however any comparison-based $(2n+1)$-renaming protocol can be used (see for example [20, 48]).

Lemma 2.2.1 *The following two sentences hold:*

- *There exists an anonymous wait-free WSB protocol on input space* $1, \ldots, 2n+1$ *if and only if there exists an anonymous wait-free WSB protocol on input space* $1, \ldots, N$, $N > 2n+1$.
- *There exists an anonymous wait-free M-renaming protocol on input space* $1, \ldots, 2n+1$ *if and only if there exists an anonymous wait-free M-renaming protocol on input space* $1, \ldots, N$, $N > 2n+1$.

Proof: Consider an anonymous wait-free WSB protocol $\mathcal{P}$ on input space $1, \ldots, 2n+1$. Using $ABDPR$, we can get an anonymous wait-free WSB protocol $\mathcal{P}'$ on input space $1, \ldots, N$, $N > 2n+1$: each process first calls $ABDPR$ with its input name as input, then invokes $\mathcal{P}$ using as input the name it receives from $ABDPR$, and finally outputs the name it receives from $\mathcal{P}$. The other direction holds trivially.

The proof for M-renaming is the same.

■

Therefore, we can assume that the input name space $1, 2, \ldots, 2n+1$ is fixed for both renaming and WSB. This assumption helps for proving the equivalence between WSB and $2n$-renaming.

Intuitively, two tasks are equivalent if a solution for one of them can be used to implement the other one, and vice versa. More formally, a task $\mathcal{A}$ *implements* a task $\mathcal{B}$ if there is a wait-free protocol that solves $\mathcal{B}$ in the asynchronous read/write shared memory model that is enriched with objects that solve $\mathcal{A}$. An object *obj* that *solves* a task $\mathcal{A}$ provides a single method **choose**. Each process invokes **choose** at most once, with an input value, and **choose** returns an output value. Object *obj* guarantees that if the collection of input values belong to the domain of input values of $\mathcal{A}$, the collection of output values returned by **choose** is allowed by $\mathcal{A}$. We say $\mathcal{A}$ and $\mathcal{B}$ are *equivalent* If $\mathcal{A}$ implements $\mathcal{B}$ and $\mathcal{B}$ implements $\mathcal{A}$.

Gafni et al. prove in [42] that WSB and $2n$-renaming are equivalent. Figure 2.2 contains an implementation of WSB using a $2n$-renaming object. Each process calls **choose** of the renaming object, R, with its input name as input, and decides the parity of the name received by R.

Renaming R = new Renaming()   %$2n$-renaming object%

WSB$_i$**(**$inName_i$**)**

  $outName_i \leftarrow$ R.choose($inName_i$)

  decide $outName_i \mod 2$

Figure 2.2: From $2n$-renaming to WSB [42].

Lemma 2.2.2 *$2n$-renaming implements WSB.*

Proof: Since WSB and renaming has input space $1, \ldots, 2n+1$, the inputs for R are valid. Also, observe that the range $1, 2, \ldots, 2n$ has exactly n even and n odd values, and hence in every execution in which all $n+1$ processes participate, at least one process decides 1 and at least one process decides 0. The protocol is anonymous and wait-free because R is anonymous and wait-free.

■

Figure 2.3 presents an implementation of $2n$-renaming using a WSB object. The protocol also uses two instances, R_0 and R_1, of the comparison-based protocol $ABDPR$ [7] that solves $(2n+1)$-renaming. $ABDPR$ also has the feature that it is *size-adaptive*: the output name space depends on the number of processes that actually participate in a given execution and not on the total number of processes of the system. The output space for $ABDPR$ is $1, \ldots, 2p-1$, where $p \leq n+1$ is the actual number of processes that participate (Section 8.1.1 contains a more detailed description of this protocol). In the implementation in Figure 2.3, each process first calls the WSB object to decide a group, either 0 or 1. Then each process in group 0 chooses an output name by calling R_0, and each process in group 1 uses R_1 to get an intermediate name and chooses an output name by subtracting that intermediate name from $2(n+1)$.

WSB wsb = new WSB()   %WSB object%

Renaming R_0 = new Renaming()   %size-adaptive $(2p-1)$-renaming object [7]%

Renaming R_1 = new Renaming()   %size-adaptive $(2p-1)$-renaming object [7]%

$2n$**-renaming**$_i$**(**$inName_i$**)**

  $side_i \leftarrow$ wsb.choose($inName_i$)

  if $side_i = 0$ **then**

    decide R_0.choose($inName_i$)

  else

    decide $(2n+1) - R_1$.choose($inName_i$)

Figure 2.3: From WSB to $2n$-renaming [42].

Lemma 2.2.3 *WSB implements $2n$-renaming.*

Proof: The inputs for wsb, R_0 and R_1 are valid because WSB and renaming has input space $1, \ldots, 2n+1$, and the input space for $ABDPR$ is $1, \ldots, N$, $N \geq n+1$. Now, we have that not all processes get 0 and not all get 1 from object wsb. The p processes that get 0, $0 < p < n+1$, choose output names in $1, \ldots, 2p-1$ by calling R_0. The other $n+1-p$ processes use R_1 to get an intermediate name in $1, \ldots, 2(n+1-p)-1$ and then pick a name in $(2n+1) - (2(n+1-p)-1) =$

$2p, \ldots, (2n+1)-1 = 2n$. Since wsb, R_1 and R_2 are anonymous and wait-free, the protocol is anonymous and wait-free.

■

Lemmas 2.2.2 and 2.2.3 imply the equivalence between WSB and $2n$-renaming:

Theorem 2.2.4 *WSB and $2n$-renaming are equivalent.*

The fact that $ABDPR$ is comparison-based implies that any WSB (M-renaming) protocol can be used to obtain a comparison-based WSB (M-renaming) protocol. This result will be useful in achieving a topological solvability condition for WSB, in Chapter 3. Figure 2.4 shows a comparison-based WSB protocol that uses $ABDPR$ and an object that solves WSB. Each process gets an intermediate name by calling an $ABDPR$ implementation, using its input name as input, and then uses that intermediate name as input for the WSB object. This transformation uses the same idea as the one used in [54] for proving that any $2n$-renaming protocol can be transformed into a comparison-based $2n$-renaming protocol.

WSB wsb = new WSB()   %WSB object%
Renaming R = new Renaming()   %comparison-based $(2n+1)$-renaming object [7]%
CB_WSB$_i(inName_i)$
  $outName_i \leftarrow$ R.choose($inName_i$)
  $dec_i \leftarrow$ wsb.choose($outName_i$)
  decide dec_i

Figure 2.4: From WSB to comparison-based WSB.

Lemma 2.2.5 *The following two sentences hold:*

- *There exists an anonymous wait-free WSB protocol if and only if there exists an anonymous wait-free comparison-based WSB protocol.*
- *There exists an anonymous wait-free M-renaming protocol if and only if there exists an anonymous wait-free comparison-based M-renaming protocol.*

Proof: Consider the protocol in Figure 2.4. The protocol is anonymous and wait-free because R and wsb are anonymous and wait-free. Let E_1 be an execution with inputs $i_0 < \ldots < i_k$ to the participating processes, and $j_0 < \ldots < j_k$ be valid inputs. Consider the execution E_2 obtained by replacing each occurrence of i_ℓ by j_ℓ, $0 \leq \ell \leq k$. Since R is comparison-based, for each participating process, the value of $name_i$ is the same in E_1 and E_2, hence dec_i is the same in both executions because wsb is anonymous. The other direction holds trivially.

The proof for M-renaming is the same.

■

Theorem 2.2.4 and Lemma 2.2.5 give the following theorem that summarizes the results of this section.

Theorem 2.2.6 *Any of the following anonymous wait-free protocols can be transformed into any other protocol in the list:*

- *WSB protocol.*
- *comparison-based WSB protocol.*
- $2n$*-renaming protocol.*
- *comparison-based* $2n$*-renaming protocol.*

As we shall see, Chapter 5 proves that, for some values of n, there is no wait-free comparison-based protocol that solves WSB on input space $1, \ldots, 2n+1$. This result, Lemma 2.2.1 and Theorem 2.2.6 imply that, for that values of n, neither there is a wait-free protocol (comparison-based or not) that solves either WSB or $2n$-renaming on input space $1, \ldots, N$, $N \geq 2n+1$.

2.3 Comparison-Based Protocols and Ramsey's Theorem

Lemma 2.2.5 says that given a protocol that solves WSB (M-renaming) on the input space $1, \ldots, 2n+1$, one can construct a comparison-based protocol that solves WSB (M-renaming) on the same input space; and Lemma 2.2.1 implies that the same holds for the input space $1, \ldots, N$, with $N > 2n+1$. This section proves that if N is large enough, any *bounded wait-free* WSB (M-renaming) protocol on input space $1, \ldots, N$, behaves as a comparison-based protocol for an ℓ-subset of inputs, for any $\ell \geq n+1$, where *bounded wait-free* means that in every execution each non-faulty process executes a number of steps bounded some fix number before deciding. Ramsey's theorem is used for proving this result.

Subsequent chapters do not use this result, however, it is presented to show the relation of comparison-based protocols with Ramsey's theorem because these ideas have been used before [9, 13, 38]. We will use the following version of Ramsey's theorem [49].

Theorem 2.3.1 (Ramsey's theorem) *For all integers k, ℓ and t, there exists an integer $f(k, \ell, t)$ such that for every set S of size at least $f(k, \ell, t)$, and every t-coloring of the k-subsets of S, some ℓ-subset of S has all its k-subsets with the same color.*

Consider a wait-free protocol $\mathcal{P}$ that solves either WSB or M-renaming on input space $1, \ldots, N$, and let S be a set of inputs of size at least $n+1$. We say that $\mathcal{P}$ *is comparison-based for S* if $\mathcal{P}$ is comparison-based in all executions in which the inputs are taken from S. The proof of the following lemma uses ideas of [9, 13, 38].

Theorem 2.3.2 *Let $\mathcal{P}$ be an anonymous bounded wait-free protocol that solves either WSB or M-renaming on input space $1, \ldots, N$, for any $N \geq n+1$. Then, for every $\ell \geq n+1$, there exists a set of inputs of size ℓ such that $\mathcal{P}$ is comparison-based for it.*

Proof: Consider an $(n+1)$-set X of $\mathbb{N}$. Let x_j, $0 \leq j \leq n$, denote the element of X such that it is greater than exactly j elements of X. Let Q be the set of all $(n+1)$-sets of $\mathbb{N}$. For $X, Y \in Q$, we say that $X \to Y$ if for any execution E_1 of $\mathcal{P}$ in which participating processes have inputs in X, the execution E_2 obtained by replacing each occurrence of x_j of each local state by y_j, $0 \leq j \leq n$, is an execution of $\mathcal{P}$ (recall that we assume $\mathcal{P}$ is full-information and standard-form, see Section 2.1).

We define the relation $\sim$ over the set Q as follows: For $X, Y \in Q$, we have that $X \sim Y$ if $X \rightarrow Y$ and $Y \rightarrow X$. In other words, $\mathcal{P}$ behaves as a comparison-based protocol when the inputs are drawn from either X or Y. The relation $\sim$ is an equivalence relation. First, it is reflexive: We have $X \sim X$ because for any execution E_1 with inputs in X, we obtain E_1 by replacing each occurrence of x_j by itself. Second, it is symmetric: If $X \sim Y$ then $X \rightarrow Y$ and $Y \rightarrow X$, thus $Y \sim X$. And third, it is transitive: If $X \sim Y$ and $Y \sim Z$, then $X \rightarrow Y$ and $Y \rightarrow Z$, hence for any execution E_1 with inputs in X, the execution E_2 obtained by replacing x_j by z_j of Z, $0 \leq j \leq n$, is an execution of $\mathcal{P}$, thus $X \rightarrow Z$; similarly we can see that $Z \rightarrow Y$ and $Y \rightarrow X$ imply that $Z \rightarrow X$, and hence $X \sim Z$.

Now, since $\mathcal{P}$ is bounded wait-free, there exists $r \in \mathbb{N}$ such that in every execution, no matter the inputs to the processes, each non-faulty process decides in at most r steps. Also, for each execution there is a finite number of combinations of outputs assignments because the number of output values is finite. Therefore, all $(n+1)$-subsets of inputs are partitioned into a finite number of equivalence classes.

Let t be the number of equivalence classes induced by $\sim$, and ℓ be an integer greater or equal than $n+1$. By Ramsey's theorem, there exists an integer $f(n+1, \ell, t)$ such that for every set S of size $f(n+1, \ell, t)$, and every t-coloring of the $(n+1)$-subsets of S, a ℓ-subset of S has all its $(n+1)$-subsets with the same color. Therefore, there is a ℓ-subset S' of the input space $1, \ldots, f(n+1, \ell, t)$ such that all its $(n+1)$-subsets are in the same equivalence class. By construction of the equivalence classes, $\mathcal{P}$ is comparison-based for S'.

■

With $\ell = 2n+1$, Theorem 2.3.2 says that for any bounded wait-free protocol $\mathcal{P}$ that solves WSB on input space $1, \ldots, N$, for any $N \geq n+1$, $\mathcal{P}$ is comparison-based for a $(2n+1)$-set S of inputs. Using S and the function $rk : \{1, \ldots, 2n+1\} \rightarrow S$ such that $rk(x) < rk(y)$ if and only if $x < y$, we can construct a wait-free comparison-based protocol that solves WSB on input space $1, \ldots, 2n+1$: each process with input in invokes $\mathcal{P}$ with input $rk(in)$ and outputs the value it receives from $\mathcal{P}$. This implies that $\mathcal{P}$ cannot exist for some values of n, since, as mentioned at the end of Section 2.2, Chapter 5 proves that, for some values of n, there is no wait-free comparison-based protocol that solves WSB on input space $1, \ldots, 2n+1$. Observe that this implication is weaker than the one derived in Section 2.2.

Chapter 3

A Topological Solvability Condition for WSB

Herlihy and Shavit [55], Saks and Zaharoglou [70] and Borowsky and Gafni [19] discovered a deep connection between distributed computing and topology, and provided a new perspective on the area. Intuitively, the idea is that for each protocol $\mathcal{P}$ for $n+1$ processes, its executions starting with a given input configuration can be represented as an n-dimensional complex $\mathcal{K}$ with a set of properties $\mathcal{S}$, which is dependent on the model of computation and on the task $\mathcal{T}$ that $\mathcal{P}$ solves. In the asynchronous wait-free read/write shared memory model, $\mathcal{K}$ always is a subdivision of an n-dimensional simplex that represents the initial state of the processes. Surprisingly, Herlihy and Shavit proved that the other direction is true, namely, for each n-dimensional subdivision $\mathcal{K}$ with properties $\mathcal{S}$, there is a wait-free protocol $\mathcal{P}$ for $n+1$ processes that solves $\mathcal{T}$ in the asynchronous read/write shared memory model.

This chapter uses this powerful approach to achieve a necessary and sufficient topological solvability condition for WSB. That is, it presents the set $\mathcal{S}$ of properties that must be satisfied by the complex $\mathcal{K}$ associated to a WSB protocol. Subsequent chapters use this condition to prove that WSB is not wait-free solvable for some values of n, while it is for the other values of n. The chapter closely follows the combinatorial framework developed in [14].

3.1 Combinatorial Topology Preliminaries

This section presents some basic definitions such as simplexes and complexes, and then presents the definition of divided image which will be useful in achieving solvability conditions for WSB.

3.1.1 Complexes, Subdivisions and Orientability

Simplexes and complexes are the main tools used here. Roughly speaking, a simplex is a generalization of the notion of a line or triangle for any dimensions, and a complex is made of simplexes that are "appropriately" glued. For example, Figure 3.1 shows two complexes of dimension 2.

Geometric Simplexes and Complexes. Let $v_0, v_1, \ldots, v_n$ be points in a Euclidean space. The hyperplane spanned by these points consists of all linear combinations $\sum_{i=0}^{n} \lambda_i v_i$, where each $\lambda_i \in \mathbb{R}$ and $\sum_{i=0}^{k} \lambda_i = 1$. The points are in *general position* if any subset of them span a strictly smaller hyperplane. Given $n+1$ points $v_0, v_1, \ldots, v_n$ in general position, the smallest convex set containing them is called an *n-dimensional geometric simplex*, or *n-geometric simplex*, i.e., all points that can be written as a linear combination $\sum_{i=0}^{n} \lambda_i v_i$, where $\lambda_i \in \mathbb{R}$, $\lambda_i \geq 0$ and $\sum_{i=0}^{n} \lambda_i = 1$. The points $v_0, v_1, \ldots, v_n$ are the *vertexes* of the simplex. For example, a 0-simplex is a point, a 1-simplex is a line segment and a 2-simplex is a triangle. Any simplex spanned by a subset of $v_0, v_1, \ldots, v_n$ is a *face* of the geometric simplex spanned by $v_0, v_1, \ldots, v_n$.

A *geometric complex* $\triangle$ is a collection of geometric simplexes in some Euclidean space such that every face of every simplex of $\triangle$ is also a simplex of $\triangle$, and the intersection of any two simplexes of $\triangle$ is also a simplex of $\triangle$.

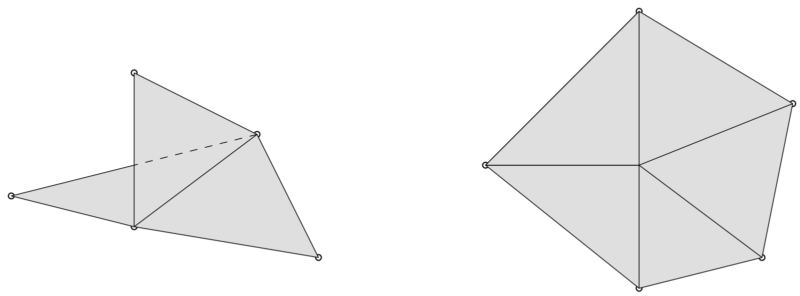

Figure 3.1: Two complexes of dimension 2.

Abstract Simplexes and Complexes. An *abstract simplex*, or just *simplex*, σ is a finite set. The elements of a simplex are its *vertexes*. The dimension of a simplex σ, $dim(\sigma)$, is the number of its vertexes minus 1. If the dimension of σ is n, then we say σ is an *n-simplex* and is denoted by σ^n. If $\sigma = \emptyset$, we say it is the *empty simplex*. Sometimes a 0-simplex $\{v\}$ is just denoted by v. A simplex τ is a *face*, or *$dim(\tau)$-face*, of σ if τ is a subset of σ. If τ is not equal to σ then τ is a *proper face* of σ. Notice that the number of i-faces of an n-simplex, $i \leq n$, is $\binom{n+1}{i+1}$.

An *abstract complex*, or just *complex*, $\mathcal{K}$ is a set of simplexes closed under containment. The dimension of a complex $\mathcal{K}$ is the maximum dimension of its simplexes. A complex $\mathcal{K}$ of dimension n is called an *n-complex* and is denoted by $\mathcal{K}^n$. The *vertexes of a complex $\mathcal{K}$*, denoted $\mathcal{V}(\mathcal{K})$, is the set consisting of the union of all its 0-simplexes. A complex $\mathcal{L}$ is a *subcomplex* of a complex $\mathcal{K}$ if $\mathcal{L} \subseteq \mathcal{K}$. Consider a complex $\mathcal{K}$ and one of its vertexes v. The *star complex* of v in $\mathcal{K}$, denoted $st(v, \mathcal{K})$, is the complex consisting of all simplexes of $\mathcal{K}$ containing v. For an n-simplex σ, let $\mathcal{M}(\sigma)$ be the complex consisting of σ and all its faces.

The *i-graph* of a complex has a node for each i-simplex of the complex, and an edge between two vertexes if they share an $(i-1)$-face. A complex is *i-connected*[1] if its i-graph is connected, or it consists of a single vertex if $i = 0$. An n-complex is *connected* if it is n-connected. An *i-path* of a complex is a path in its i-graph, namely, a sequence of i-simplexes such that every two consecutive i-simplexes share an $(i-1)$-face. The *length* of an i-path P, denoted $|P|$, is the number of its i-simplexes.

[1]This definition is not equivalent to the usual definition of i-connected (see for example [64]). Intuitively, the usual definition of i-connected means that the complex does not have “holes” of dimension less or equal than i (the homology group of dimension $k \leq i$ is trivial).

An n-complex $\mathcal{K}$ is *i-complete*, $i \leq n$, if each of its j-simplex, $j \leq i$, is a face of at least one of its i-simplexes. Let $\mathcal{K}^n$ be an n-complete complex. If an $(n-1)$-simplex of K^n is a face of exactly one of its n-simplexes, then we call such a simplex *external*, otherwise *internal*. The *boundary* of $\mathcal{K}^n$, $bd(\mathcal{K}^n)$, is the subcomplex induced by the external simplexes of $\mathcal{K}^n$, and all their faces. We say that $\mathcal{K}^n$ is an *n-pseudomanifold* if each of its $(n-1)$-simplex is contained in either one or two n-simplexes. The complex at the left in Figure 3.1 is not a pseudomanifold because one of its 1-simplexes is contained in three 2-simplexes; the complex at the right is a pseudomanifold.

Colorings and Simplicial Maps. A *coloring* of a complex $\mathcal{K}$ is a function from $\mathcal{V}(\mathcal{K})$ to a set of *colors*. The set $ID^n = \{0, \ldots, n\}$ is often used as colors. A *binary coloring* of $\mathcal{K}$ is a coloring with colors $\{0, 1\}$. A coloring of a simplex is *proper* if it gives different values to different vertexes. If a coloring of a simplex gives the same value b to every vertex, we say the simplex is *b-monochromatic* or just *monochromatic*. An n-complex is *chromatic* if it has a coloring that uses $n+1$ colors and each one of its simplexes is properly colored. Subsequent sections often consider complexes with more than one coloring at its vertexes.

Let $\mathcal{K}$ and $\mathcal{L}$ be complexes and δ be a function from $\mathcal{V}(\mathcal{K})$ to $\mathcal{V}(\mathcal{L})$. We say that δ is *simplicial* if for every simplex $\sigma = \{v_0, \ldots, v_n\}$ of $\mathcal{K}$, the set $\tau = \{\delta(v_0), \ldots, \delta(v_n)\}$ is a simplex of $\mathcal{L}$, possibly with $dim(\tau) < dim(\sigma)$. Sometimes it is convenient to extend δ to simplexes: a simplex $\sigma \in \mathcal{K}$ is mapped to a simplex $\delta(\sigma) \in \mathcal{L}$. Hence δ can also be extended to subcomplexes. A simplicial map δ between two colored complexes is *color-preserving* if for every vertex v, $\delta(v)$ has the same color as v. Frequently, for a simplicial map δ, we write $\delta : \mathcal{K} \to \mathcal{L}$ instead of $\delta : \mathcal{V}(\mathcal{K}) \to \mathcal{V}(\mathcal{L})$.

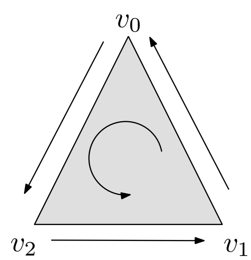

Figure 3.2: An oriented and properly colored simplex.

Orientability. Let $\sigma^n = \{v_0, v_1, \ldots, v_n\}$ be a simplex. An *orientation* of σ^n is a set consisting of a sequence of its vertexes and all even permutations of it. If $n > 0$, these sets fall into two equivalence classes, the sequence $\langle v_0, v_1, \ldots, v_n \rangle$ and all its even permutations, and the sequence $\langle v_1, v_0, \ldots, v_n \rangle$ and all its even permutations. For example, the two possible orientations of a 2-simplex are the clockwise and the counterclockwise directions. An orientation of σ^n, $n > 0$, *induces an orientation* on all of its $(n-1)$-faces: if σ^n is oriented $\langle v_0, v_1, \ldots, v_n \rangle$ then its face, τ, without vertex v_i, gets the orientation $\langle v_0, v_1, \ldots \widehat{v_i} \ldots, v_n \rangle$ if i is even, where circumflex $(\widehat{\ })$ denotes omission; otherwise τ gets the orientation $\langle v_1, v_0, \ldots \widehat{v_i} \ldots, v_n \rangle$. Figure 3.2 contains an oriented 2-simplex and the induced orientation of its 1-faces. For a simplex σ^n with a proper coloring id with colors ID^n, $d = +1$ denotes the *positive* orientation containing the sequence $\langle 0, 1, \ldots, n \rangle$, i.e., the sequence $\langle v_0, v_1, \ldots, v_n \rangle$ of σ^n such that $id(v_i) = i$, $0 \leq i \leq n$, and $d = -1$ denotes the *negative* orientation. If σ^n has orientation d, its $(n-1)$-face without color i gets the induced orientation $(-1)^i d$.

A pseudomanifold $\mathcal{K}^n$ is *orientable* if it is possible to give an orientation to each of its n-simplexes such that if $\sigma_1^n, \sigma_2^n \in \mathcal{K}^n$ share an $(n-1)$-face τ then τ gets opposite induced orientations from σ_1^n and σ_2^n. Such an orientation is a *coherent orientation* of $\mathcal{K}^n$. We say $\mathcal{K}^n$ is *coherently oriented* if it has a coherent orientation.

Lemma 5.12 in [14] gives a necessary and sufficient orientability condition for chromatic pseudomanifolds. Lemma 3.1.1 is a restatement of that lemma using our notation.

Lemma 3.1.1 *A chromatic n-pseudomanifold is orientable if and only if its n-simplexes can be partitioned into two classes, $+1$ and -1, such that every two n-simplexes sharing an $(n-1)$-face belong to distinct classes. In a coherent orientation, each n-simplex in a class d, is d oriented.*

Subdivisions. A geometric complex $\lambda(\triangle)$ is a *subdivision* of a geometric complex $\triangle$ if every simplex of $\lambda(\triangle)$ is contained in a simplex of $\triangle$, and every simplex of $\triangle$ is the union of finitely many simplexes of $\lambda(\triangle)$. Notice that each geometric complex $\triangle$ determines a complex $\mathcal{K}$. We say $\triangle$ is a *geometric realization* of $\mathcal{K}$. A complex $\lambda(\mathcal{K})$ is a *subdivision* of a complex $\mathcal{K}$ if there exists geometric realizations $\lambda(\triangle)$ and $\triangle$ of $\lambda(\mathcal{K})$ and $\mathcal{K}$, such that $\lambda(\triangle)$ is a subdivision of $\triangle$. For each $\sigma \in \lambda(\mathcal{K})$, the *carrier* of σ, denoted $carr(\sigma, \mathcal{K})$, is the unique smallest simplex $\tau \in \mathcal{K}$ such that σ is contained in τ. A chromatic complex $\lambda(\mathcal{K})$ is a *chromatic subdivision* of a chromatic complex $\mathcal{K}$ if each $\sigma \in \lambda(\mathcal{K})$ is properly colored with the colors of $carr(\sigma, \mathcal{K})$.

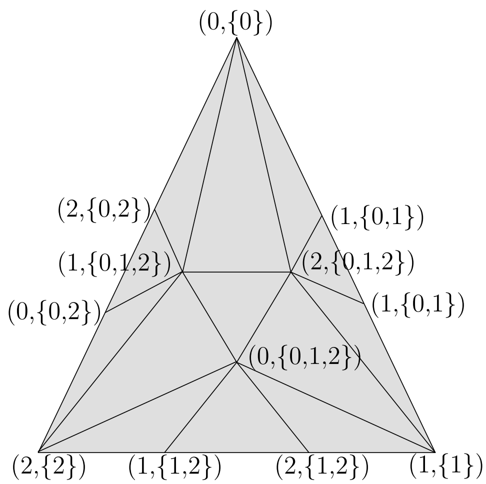

Figure 3.3: The standard chromatic subdivision of dimension 2.

For an n-simplex σ with a proper coloring id that uses colors ID^n, the *standard chromatic subdivision of* σ, denoted $\chi^1(\sigma)$, consists of all simplexes of the form $\{(\ell_0, \sigma_0), \dots, (\ell_i, \sigma_i)\}$, $0 \leq i \leq n$, where $\sigma_0 \subseteq \dots \subseteq \sigma_m$ are faces of σ, $|\{\ell_0, \dots, \ell_m\}| = m + 1$ and $\ell_j \in id(\sigma_j)$, $0 \leq j \leq i$. Figure 3.3 depicts the standard chromatic subdivision of dimension 2. The *k iterated standard chromatic subdivision of* σ, $\chi^k(\sigma)$, is the chromatic subdivision obtained by applying the standard chromatic subdivision over each n-simplex of $\chi^{k-1}(\sigma)$, namely, $\chi^k(\sigma) = \cup_{\tau^n \in \chi^{k-1}(\sigma)} \chi^1(\tau^n)$. Similarly, for an n-complex $\mathcal{K}$, the *k iterated standard chromatic subdivision of* $\mathcal{K}$, $\chi^k(\mathcal{K})$, is $\cup_{\tau^n \in \mathcal{K}} \chi^k(\tau^n)$.

3.1.2 Divided Images

Divided images were introduced and studied in [14] to model the structure of the complex associated to all IS executions of a wait-free protocol. This section briefly reviews the main properties of these combinatorial topology objects.

Definition 3.1.2 ([14, Definition 4.1]) *Let $\mathcal{K}^n$, $\mathcal{L}^n$ be complexes, and ψ a function that assigns to each simplex of $\mathcal{L}^n$ a finite subcomplex of $\mathcal{K}^n$. The complex $\mathcal{K}^n$ is a* divided image *of $\mathcal{L}^n$ under ψ if:*

1. *$\psi(\emptyset) = \emptyset$*
2. *for every $\tau \in \mathcal{K}^n$ there is a simplex $\sigma \in \mathcal{L}^n$ such that $\tau \in \psi(\sigma)$*
3. *for every $\sigma^0 \in \mathcal{L}^n$, $\psi(\sigma^0)$ is a single vertex*
4. *for every $\sigma_1, \sigma_2 \in \mathcal{L}^n$, $\psi(\sigma_1 \cap \sigma_2) = \psi(\sigma_1) \cap \psi(\sigma_2)$*
5. *for every $\sigma \in \mathcal{L}^n$, $\psi(\sigma)$ is a $dim(\sigma)$-pseudomanifold with $bd(\psi(\sigma)) = \psi(bd(\sigma))$*

$\mathcal{K}^n$ is a divided image *of $\mathcal{L}^n$ if for some ψ, $\mathcal{K}^n$ is a divided image of $\mathcal{L}^n$ under ψ.*

In condition 5, $bd(\sigma)$ is a set of simplexes, and $\psi(bd(\sigma))$ is the complex consisting of the union of $\psi(\tau)$, over the simplexes $\tau \in bd(\sigma)$. This notation is often used.

Figure 3.4 depicts a divided image of dimension 2 where $\mathcal{L}^2$ is the complex consisting of a 2-simplex and all its faces, and the arrows show how ψ maps the vertexes of $\mathcal{L}^2$.

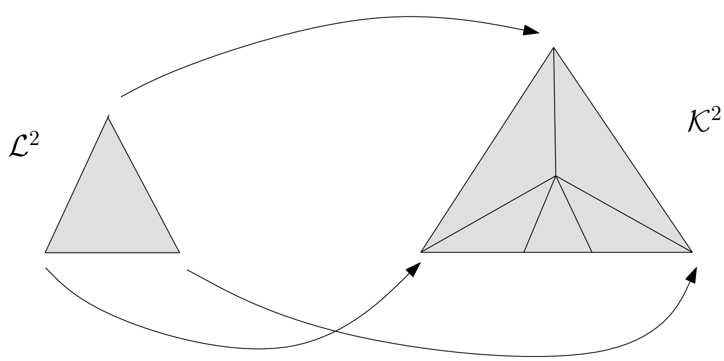

Figure 3.4: A 2-dimensional divided image.

It is worth to notice that a divided image is not necessarily a subdivision, even if it is connected. For example, a torus $\mathcal{L}$ of dimension 2 with a 2-simplex τ removed from it, is a divided image of a 2-simplex σ: $bd(\sigma)$ is mapped to $bd(\tau)$ and σ is mapped to $\mathcal{L}$.

Let $\mathcal{K}^n$ be a divided image of $\mathcal{L}^n$ under ψ. We say that $\mathcal{K}^n$ is *well-connected* if for every i-simplex $\sigma \in \mathcal{L}^n$, if $i \geq 1$ then $\psi(\sigma)$ is i-connected, and if $i \geq 2$ then $bd(\psi(\sigma))$ is $(i-1)$-connected. Similarly, $\mathcal{K}^n$ is *orientable* if for every $\sigma \in \mathcal{L}^n$, $\psi(\sigma)$ is orientable. And $\mathcal{K}^n$ is *coherently oriented* if for every n-simplex $\sigma \in \mathcal{L}^n$, the n-pseudomanifold $\psi(\sigma^n)$ is coherently oriented. Also, we say that $\mathcal{K}^n$ is a *subdivision* if for each simplex $\sigma \in \mathcal{L}^n$, $\psi(\sigma)$ is a subdivision of $\mathcal{M}(\sigma)$.

The *carrier* of a simplex $\tau \in \mathcal{K}^n$, denoted $carr(\tau)$, is the simplex $\sigma \in \mathcal{L}^n$ of smallest dimension such that $\tau \in \psi(\sigma)$. Assume $\mathcal{L}^n$ is chromatic with colors ID^n. The set colors of a simplex $\sigma \in \mathcal{L}^n$ is denoted $id(\sigma)$. We say that $\mathcal{K}^n$ is *chromatic* if every simplex $\tau \in \mathcal{K}^n$ with $dim(\tau) = dim(carr(\tau))$, is properly colored with $id(carr(\tau))$. Figure 3.10 depicts a chromatic divided image of a 2-simplex.

3.2 Modeling Tasks

For a domain of inputs I, an *input complex*, denoted $\mathcal{I}^n$, is the complex consisting of the n-simplexes (subsets with $n+1$ elements) of $ID^n \times I$, and all their faces, whose *id* coloring, the first entry of each pair, is proper. Thus, $\mathcal{I}^n$ is chromatic with respect to *id*. An *output complex*, $\mathcal{O}^n$, over a domain of outputs O, is defined similarly. The meaning of a vertex (i, v) of $\mathcal{I}^n$ ($\mathcal{O}^n$) is that process with *id* i has input (output) v.

A *task* is defined as a triple $\langle \mathcal{I}^n, \mathcal{O}^n, \triangle \rangle$, where $\mathcal{I}^n$ is an input complex, $\mathcal{O}^n$ is an output complex and $\triangle$ is a function that maps each n-simplex of $\mathcal{I}^n$ to a non-empty set of n-simplexes of $\mathcal{O}^n$. We sometimes mention only $\triangle$ when $\mathcal{I}^n$ and $\mathcal{O}^n$ are understood. The simplexes of $\triangle(\sigma^n)$ are the *admissible* output simplexes for the input simplex σ^n, namely, the outputs in $\tau^n \in \triangle(\sigma^n)$ can be the outputs of the processes when they start with inputs σ^n. Observe that $\triangle$ is only defined for n-simplexes, however, it can be extended for simplexes of lower dimensions, since the outputs of an execution, possibly of a proper subset of processes, can be completed to outputs for all processes that are admissible for the inputs of the execution (see Section 2.1). Therefore, $\triangle$ maps each σ^m, $m < n$, to all faces of the simplexes of $\triangle(\sigma^n)$ with same *id*'s as σ^m, for all $\sigma^n \in \mathcal{I}^n$ such that σ^m is a face of σ^n.

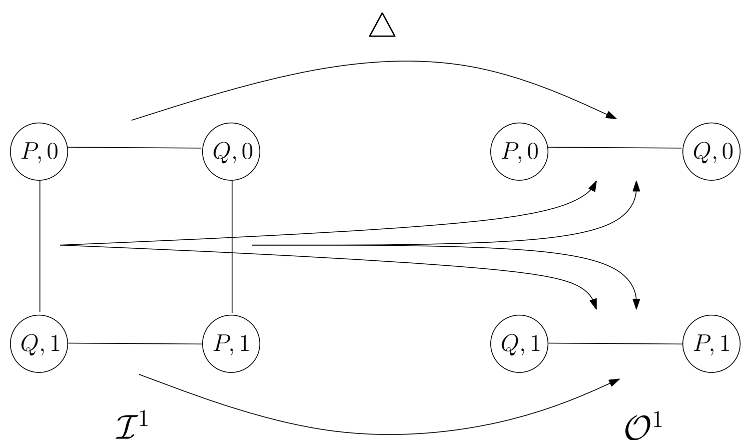

Figure 3.5: The binary consensus task on two processes.

For example, in the binary consensus task each process has input either 0 or 1 and it is required that processes decide a unique output value among the inputs to the processes. Thus, $I = O = \{0, 1\}$. Figure 3.5 depicts the binary consensus task for two processes with *id*'s P and Q. It only shows how $\triangle$ maps the 1-simplexes of $\mathcal{I}^1$, however, $\triangle$ is extended as $\triangle((i, v)) = (i, v)$ for each vertex $(i, v) \in \mathcal{I}^1$.

As another example consider the WSB task. We have $I = \{1, \ldots, 2n+1\}$ (recall the input name space for WSB was fixed, see Section 2.2) and $O = \{0, 1\}$. Input complex $\mathcal{I}^n_{wsb}$ contains every *id* properly colored n-simplex of $ID^n \times \{1, \ldots, 2n+1\}$ such that any pair of its vertexes have distinct input values, as input names in WSB are unique. And $\mathcal{O}^n_{wsb}$ contains every *id* properly colored n-simplex of $ID^n \times \{0, 1\}$ such that it is not monochromatic, considering the output values, as not all processes decide 0 and not all decide 1. Figure 3.6 shows the WSB output complex for three processes. For each $\sigma^m \in \mathcal{I}^n$, $1 \leq m \leq n$, $\triangle(\sigma^m)$ contains every m-simplex of $\mathcal{O}^n$ with same *id* colors as σ^m. Thus, for every $\sigma^n \in \mathcal{I}^n$, $\triangle(\sigma^n) = \mathcal{O}^n_{wsb}$.

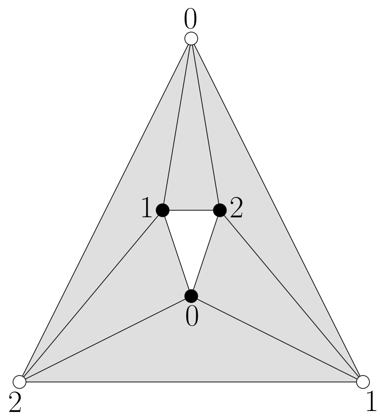

Figure 3.6: WSB output complex for three processes.

The rest of this chapter only considers tasks that hold the following symmetry requirement on inputs and outputs: for every simplex τ of $\mathcal{I}^n$ ($\mathcal{O}^n$) and every permutation π of ID^n, $\mathcal{I}^n$ ($\mathcal{O}^n$) contains the simplex τ' obtained by replacing each *id* i in τ by $\pi(i)$. Observe that the input and output complexes for WSB and renaming hold this property.

3.3 Modeling Protocols

For any execution α, let $views(\alpha)$ be the set $\{(0, input_0, \alpha|0), \ldots, (n, input_n, \alpha|n)\}$, where $input_i$ is the input for process with *id* i. The *protocol complex*, $\mathcal{P}^n$, of a protocol $\mathcal{P}$ is the complex containing the n-simplex $views(\alpha)$, an all its faces, for each minimal final execution α of $\mathcal{P}$. Notice that $\mathcal{P}^n$ has three coloring: an *id*, an input and a view. Moreover, *id* coloring is chromatic. Since $\mathcal{P}$ is deterministic and wait-free, each process decides in a finite number of steps, implying that $\mathcal{P}^n$ is finite. It is possible that, for two distinct minimal final executions α and α', $views(\alpha) = views(\alpha')$, hence we can think of each n-simplex of $\mathcal{P}^n$ as an equivalence class of executions in which processes "look the same".

Given an input n-simplex σ, $\mathcal{P}^n(\sigma)$ is the subcomplex of $\mathcal{P}^n$ that contains all n-simplexes, and all their faces, corresponding to all minimal final executions of $\mathcal{P}$ in which processes start with inputs σ. Roughly, τ belongs to $\mathcal{P}^n(\sigma)$ if and only if there is an execution α with inputs σ such that the views in τ are the same as the views in α. Moreover, if τ is contained in two n-simplexes σ_1 and σ_2 of $\mathcal{P}^n(\sigma)$, processes in τ cannot distinguish between the corresponding executions for σ_1 and σ_2.

Figure 3.7 shows the protocol complex of a standard-form protocol for two processes with inputs i_0 and i_1. Each process is active in exactly one round. The vertexes of the complex are only colored with *id*'s and views (below each vertex), represented by the content of the memory. Above each

1-simplex it is the schedule of one the executions associated to that simplex. The reader can verify that every execution of the protocol is represented by a simplex of the complex. For example, the execution with schedule $w_0, r_0(0), w_1, r_1(0), r_0(1), r_1(1)$ is represented by the simplex at the center.

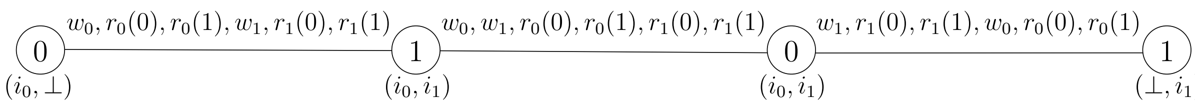

Figure 3.7: Protocol complex of a protocol for two processes.

For some task $\langle \mathcal{I}^n, \mathcal{O}^n, \triangle \rangle$, $\mathcal{P}$ induces a *decision map* $\delta_{\mathcal{P}} : \mathcal{P}^n \to \mathcal{O}^n$ that specifies the output value for every final view of a process. If $\mathcal{P}$ solves $\triangle$ then $\delta_{\mathcal{P}}$ is simplicial, since for each $\tau \in \mathcal{P}^n$, $\delta_{\mathcal{P}}(\tau)$ is an output simplex. Moreover, $\delta_{\mathcal{P}}$ preserves the *id* coloring and $\delta_{\mathcal{P}}(\mathcal{P}^n(\sigma))$ is a complex, for each n-simplex $\sigma \in \mathcal{I}^n$. Using this notation, the operational definition of a protocol solving a task presented in Section 2.1, can be interpreted as Proposition 3.3.1 states. Figure 3.8 shows a graphic description this interpretation.

Proposition 3.3.1 ([14, Proposition 4.5]) *$\mathcal{P}$ solves $\langle \mathcal{I}^n, \mathcal{O}^n, \triangle \rangle$ if and only if $\delta_{\mathcal{P}}(\mathcal{P}^n(\sigma)) \subseteq \triangle(\mathcal{M}(\sigma))$, for every n-simplex $\sigma \in \mathcal{I}^n$. In this case we say that $\delta_{\mathcal{P}}$* agrees *with $\triangle$.*

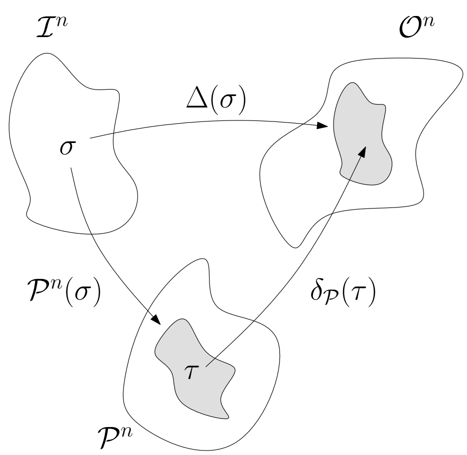

Figure 3.8: A protocol solving a task.

Since only anonymous protocols are considered, which intuitively means that the decision of a process does not depend on its *id*, if two n-simplexes $\sigma, \sigma' \in \mathcal{I}^n$ have the same input, i.e., they differ only by a permutation of ID^n, then $\mathcal{P}^n(\sigma)$ can be obtained from $\mathcal{P}^n(\sigma')$. This implies that $\delta_{\mathcal{P}}$ must be "anonymous" in a sense that $\delta_{\mathcal{P}}(\mathcal{P}^n(\sigma))$ can be obtained from $\delta_{\mathcal{P}}(\mathcal{P}^n(\sigma'))$. These ideas are formalized as follows.

Consider an n-complex $\mathcal{K}$ with a chromatic coloring *id* that uses ID^n. We say that $\mathcal{K}$ is *symmetric* if for any permutation π of ID^n, there is a simplicial bijection $\pi' : \mathcal{K} \to \mathcal{K}$ that *respects* π: for each $u \in \mathcal{K}$, $\pi(id(v)) = id(\mu(v))$. Sometimes π denotes both the map π' and permutation π, relying on the context. It is not hard to verify that the input and output complexes of the tasks considered here (see end of Section 3.2), are symmetric.

Consider two symmetric n-complexes $\mathcal{K}$ and $\mathcal{L}$. Let ξ be a color-preserving simplicial map from $\mathcal{K}$ to $\mathcal{L}$. We say that ξ is *anonymous* if for any permutation π of ID^n, there exist simplicial bijections $\pi_1 : \mathcal{K} \to \mathcal{K}$ and $\pi_2 : \mathcal{L} \to \mathcal{L}$ that respect π, such that $\xi \circ \pi_1 = \pi_2 \circ \xi$.

Therefore, if input simplexes $\sigma, \sigma' \in \mathcal{I}^n$ differ by a permutation $\pi : ID^n \to ID^n$, then there exist simplicial bijections $\pi_1 : \mathcal{P}^n(\sigma) \to \mathcal{P}^n(\sigma')$ and $\pi_2 : \delta_{\mathcal{P}}(\mathcal{P}^n(\sigma)) \to \delta_{\mathcal{P}}(\mathcal{P}^n(\sigma'))$ that respect π, such that $\delta_{\mathcal{P}} \circ \pi_1 = \pi_2 \circ \delta_{\mathcal{P}}$.

3.4 An Anonymous Condition for Wait-Free Solvability

For a protocol $\mathcal{P}$ with protocol complex $\mathcal{P}^n$, its *immediate snapshot complex* (IS complex), denoted $\mathcal{E}^n$, is the subcomplex of $\mathcal{P}^n$ containing all ISE execution of $\mathcal{P}$.

Attiya and Rajsbaum prove in [14] that $\mathcal{E}^n$ is a divided image of $\mathcal{I}^n$, where each input simplex σ is mapped to the subcomplex of $\mathcal{E}^n$ that contains all $dim(\sigma)$-simplexes corresponding to the IS execution of $\mathcal{P}$, in which processes in $id(\sigma)$ start with inputs σ. Clearly, due to the id coloring of $\mathcal{E}^n$ and $\mathcal{I}^n$, $\mathcal{E}^n$ is a chromatic divided image of $\mathcal{I}^n$. Figure 3.9 shows an example of this for a 2-dimensional input simplex σ. The inputs of σ and the views and output values of $\psi(\sigma)$ do not appear. Each process is active in exactly one round in every execution. The corners of $\psi(\sigma)$ correspond to solo executions: a process is active alone in the first round and hence does not see the other two processes. The 1-simplexes on the boundary correspond to all executions in which two processors do not see the third process. And the 2-simplexes represent the executions in which at least one processor sees all processes. In particular, the 2-simplex at the center of $\psi(\sigma)$ corresponds to the execution in which the three processes are active in the same round and thus see each other.

It is also proved in [14] that $\mathcal{E}^n$ is a chromatic, well-connected and orientable divided image of $\mathcal{I}^n$. These properties of $\mathcal{E}^n$ and Proposition 3.3.1 give the following necessary solvability condition of a task.

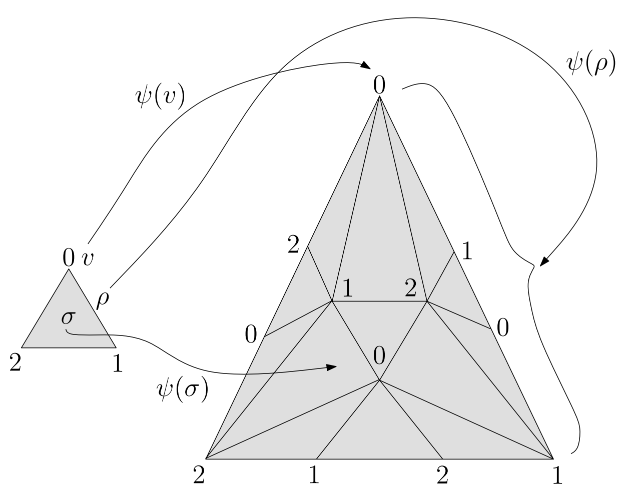

Figure 3.9: IS-complex for three processes.

Notation. From now on, we write *ccodi* as a shorthand for chromatic, well-connected and orientable divided image, and *ccosdi* for ccodi that is also a subdivision.

Theorem 3.4.1 ([14, Theorem 5.14]) *Let $\langle \mathcal{I}^n, \mathcal{O}^n, \triangle \rangle$ be a task. If there is an anonymous wait-free protocol which solves this task, then there is a ccodi of $\mathcal{I}^n$ and there is a color-preserving (on ids), anonymous simplicial map form it to $\mathcal{O}^n$ that agrees with $\triangle$.*

The Anonymous Computability Theorem by Herlihy and Shavit in [55], fully characterizes the tasks that are anonymous wait-free solvable in the asynchronous read/write shared memory model. The following theorem is a restatement in our notation of one direction of the Anonymous Computability Theorem, which gives the opposite direction of Theorem 3.4.1.

Theorem 3.4.2 *For a task $\langle \mathcal{I}^n, \mathcal{O}^n, \triangle \rangle$, if there exists an integer k and a color-preserving (on ids) anonymous simplicial map $\mu : \chi^k(\mathcal{I}^n) \to \mathcal{O}^n$ that agrees with $\triangle$, then there exists an anonymous wait-free protocol which solves $\langle \mathcal{I}^n, \mathcal{O}^n, \triangle \rangle$.*

Notice that Theorem 3.4.2 only considers a specific class of chromatic subdivisions, namely, iterated standard chromatic subdivisions. However, it is explained in [55] that any chromatic subdivision $\lambda(\mathcal{I}^n)$ and color-preserving anonymous simplicial map $\mu : \lambda(\mathcal{I}^n) \to \mathcal{O}^n$ that agrees with $\triangle$, imply the existence of a protocol for $\langle \mathcal{I}^n, \mathcal{O}^n, \triangle \rangle$. Very roughly, the idea is that for any chromatic subdivision $\lambda(\mathcal{I}^n)$, there exists an integer k and a simplicial map $\delta : \chi^k(\mathcal{I}^n) \to \lambda(\mathcal{I}^n)$ such that the composition of δ and μ is the simplicial map required in Theorem 3.4.2. Borowsky and Gafni give an algorithmic proof of that result in [21]. Therefore, we get the following result.

Theorem 3.4.3 *Let $\langle \mathcal{I}^n, \mathcal{O}^n, \triangle \rangle$ be a task. If there is a ccosdi of $\mathcal{I}^n$ and there is a color-preserving (on ids), anonymous simplicial map from it to $\mathcal{O}^n$ that agrees with $\triangle$, then there exists an anonymous wait-free protocol which solves $\langle \mathcal{I}^n, \mathcal{O}^n, \triangle \rangle$.*

3.5 Necessary and Sufficient Solvability Conditions for WSB

Necessity. Let $\mathcal{P}$ be a WSB protocol. By Theorem 3.4.1, there is a ccodi ξ^n of $\mathcal{I}^n_{wsb}$ and there is a color-preserving anonymous simplicial map $\delta_\mathcal{P} : \xi^n \to \mathcal{O}^n_{wsb}$ that agrees with WSB, where $\mathcal{I}^n_{wsb}$ and $\mathcal{O}^n_{wsb}$ are the input and output complexes for WSB and $\delta_\mathcal{P}$ is the decision map induced by $\mathcal{P}$. Let us add the output binary coloring b to ξ^n defined as $b(v) = \delta_\mathcal{P}(v)$, for each vertex $v \in \xi^n$. Let φ be the function such that ξ^n is a ccodi of $\mathcal{I}^n_{wsb}$ under φ. As explained in Section 3.4, for every $\sigma \in \mathcal{I}^n_{wsb}$, $\varphi(\sigma)$ contains all IS execution in which processes start with inputs σ. Therefore, for each n-simplex $\sigma \in \xi^n$, $\varphi(\sigma)$ cannot contain monochromatic n-simplexes because these simplexes correspond to executions in which all processes decide the same output value, contradicting the specification of WSB. Now, by Lemma 2.2.5, we can assume that $\mathcal{P}$ is comparison-based. Since $\mathcal{P}$ is anonymous, the output coloring b is a function of the rank of input names in a given execution. As we shall see, this induces a symmetry on b on the boundary of $\varphi(\sigma)$, for each n-simplex σ of $\mathcal{I}^n_{wsb}$. The following definitions are used to formalize this intuition.

In what follows, for a simplex σ^n, let σ^n denote the complex $\mathcal{M}(\sigma^n)$. Consider a chromatic divided image $\mathcal{K}^n$ of σ^n under ψ. Let σ and σ' be i-faces of σ^n. A simplicial bijection $\mu : \psi(\sigma) \to \psi(\sigma')$ is *id-preserving* if for every vertexes $u, v \in \psi(\sigma)$, if $id(u) = id(v)$ then $id(\mu(u)) = id(\mu(v))$. If in addition, for every $u \in \psi(\sigma)$, $rk(id(u)) = id(\mu(u))$, where $rk : id(\sigma) \to id(\sigma')$ is the bijection such that if $x < y$ then $rk(x) < rk(y)$, then μ is *id-rank-preserving.* Notice that there can be only one id-rank-preserving bijection. We say that $\mathcal{K}^n$ has *structural-symmetry* if for every two i-faces σ and σ' of σ^n, there is an id-preserving simplicial bijection between $\psi(\sigma)$ and $\psi(\sigma')$. Similarly, $\mathcal{K}^n$ has *structural-rank-symmetry* if for every two i-faces σ and σ' of σ^n, there is an *id-rank-preserving* simplicial bijection between $\psi(\sigma)$ and $\psi(\sigma')$. Clearly, if $\mathcal{K}^n$ has structural-rank-symmetry, it has structural-symmetry.

Assume $\mathcal{K}^n$ has structural-symmetry. For every i-faces σ and σ' of σ^n, fix an id-preserving simplicial bijection $\mu_{\sigma\sigma'} : \psi(\sigma) \to \psi(\sigma')$ such that $\mu^{-1}_{\sigma\sigma'} = \mu_{\sigma'\sigma}$. Let $\mathcal{F}$ be the family of simplicial bijections $\mu_{\sigma\sigma'}$. Then $\mathcal{K}^n$ has *structural-symmetry with respect to* $\mathcal{F}$. For each $\mu_{\sigma\sigma'}$, vertexes $u \in \psi(\sigma)$ and $v \in \psi(\sigma')$ are *isomorphic with respect to* μ if $\mu(u) = v$. Isomorphic simplexes with respect to μ are defined similarly. Observe that isomorphic simplexes between $\psi(\sigma)$ and $\psi(\sigma')$ are well defined since $\mu^{-1}_{\sigma\sigma'} = \mu_{\sigma'\sigma}$.

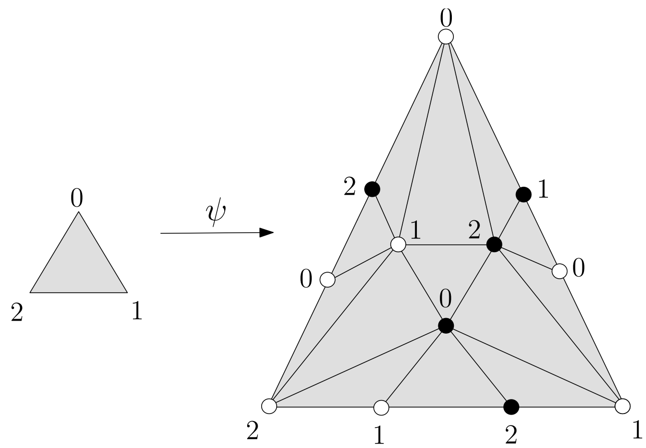

Figure 3.10: A chromatic divided image with a symmetric binary coloring.

Let $\mathcal{K}^n$ be a chromatic divided image of σ^n under ψ, with structural-symmetry with respect to a family $\mathcal{F}$, and with a binary coloring b. The coloring b is *symmetric with respect to* $\mathcal{F}$ if every $\mu_{\sigma\sigma'} \in \mathcal{F}$ is color-preserving, i.e., for every vertex $v \in \psi(\sigma)$, $b(v) = b(\mu(v))$. If there is a family of simplicial bijections such that b is symmetric with respect to it, then b is *symmetric.* Also, b is *rank-symmetric* if it is symmetric with respect to the family of id-rank-preserving simplicial bijections. Therefore a divided image with a (rank-)symmetric binary coloring, has structural-(rank-)symmetry. Figure 3.10 presents a chromatic divided image with a rank-symmetric binary coloring (white and black circles represent binary colors 0 and 1).

Consider an n-simplex $\sigma^n \in \mathcal{I}^n_{wsb}$ with inputs $j_0 < \ldots < j_n$ such that process with *id* i has input j_i. We have that $\mathcal{K}^n = \varphi(\sigma^n)$ is a ccodi of σ^n under $\psi = \varphi|\mathcal{M}(\sigma^n)$. Consider two i-faces σ and σ' of σ^n. Complex $\psi(\sigma)$ contains all IS executions with participating set $id(\sigma)$ and inputs σ. Since $\mathcal{P}$ is comparison-based, hence anonymous, coloring b of $\psi(\sigma)$ is completely on function on σ's input relatively order, captured by the *id* colors. That is, for each i-simplex $\tau \in \psi(\sigma)$, corresponding to an IS execution, there is an i-simplex $\tau' \in \psi(\sigma')$ associated to the execution obtained by replacing the smallest *id* in $id(\sigma)$ with the smallest *id* in $id(\sigma')$, etc. Therefore, there must be an id-rank-preserving simplicial bijection $\mu : \psi(\sigma) \to \psi(\sigma')$. Moreover, for every vertex $v \in \tau$, $b(v) = b(\mu(v))$, i.e., μ is color-preserving. Thus, b is rank-symmetric. This gives a necessary solvability condition for WSB, Theorem 3.5.1, recalling that $\mathcal{K}^n$ does not have monochromatic n-simplexes. Although b is rank-symmetric, Theorem 3.5.1 only considers that b is symmetric because that property of b is enough for proving the impossibility of WSB.

Theorem 3.5.1 *If there exists an anonymous wait-free WSB protocol then there exists a ccodi of an n-simplex with a symmetric binary coloring and no monochromatic n-simplexes.*

Sufficiency. By Theorem 3.4.3, if there is a ccosdi $\mathcal{L}^n$ of $\mathcal{I}^n_{wsb}$ and there is a color-preserving anonymous simplicial map $\delta : \mathcal{L}^n \to \mathcal{O}^n_{wsb}$ that agrees with WSB, then there is a WSB protocol. In what follows it is proved that using a ccosdi of an n-simplex with a rank-symmetric binary coloring and no monochromatic n-simplexes, one can construct $\mathcal{L}^n$ and δ.

First, we verify that $\mathcal{I}^n_{wsb}$ and $\mathcal{O}^n_{wsb}$ are symmetric complexes. Recall that $\mathcal{I}^n_{wsb}$ contains every *id* properly colored n-simplex of $ID^n \times \{1, \ldots, 2n+1\}$ such that any pair of vertexes have distinct input values, and $\mathcal{O}^n_{wsb}$ contains every *id* properly colored n-simplex of $ID^n \times \{0, 1\}$ such that it is not monochromatic, considering the output values. Consider a permutation π of ID^n. Notice that for every simplex $\{(i_0, name_0), \ldots, (i_j, name_j)\} \in \mathcal{I}^n_{wsb}$, $0 \le j \le n$, input complex $\mathcal{I}^n_{wsb}$ contains the simplex $\{(\pi(i_0), name_0), \ldots, (\pi(i_j), name_j)\}$. Thus, the simplicial map $\pi_{\mathcal{I}} : \mathcal{I}^n_{wsb} \to \mathcal{I}^n_{wsb}$ that maps each vertex $(i, name) \in \mathcal{I}^n_{wsb}$ to $(\pi(i), name)$, is a bijection that respects π. Hence, $\mathcal{I}^n_{wsb}$ is symmetric. Observe that $\pi_{\mathcal{I}}$ preserves the input coloring. Similarly, we can see that, for $\mathcal{O}^n_{wsb}$, there is a simplicial bijection $\pi_{\mathcal{O}} : \mathcal{O}^n_{wsb} \to \mathcal{O}^n_{wsb}$ that respects π and preserves the output coloring.

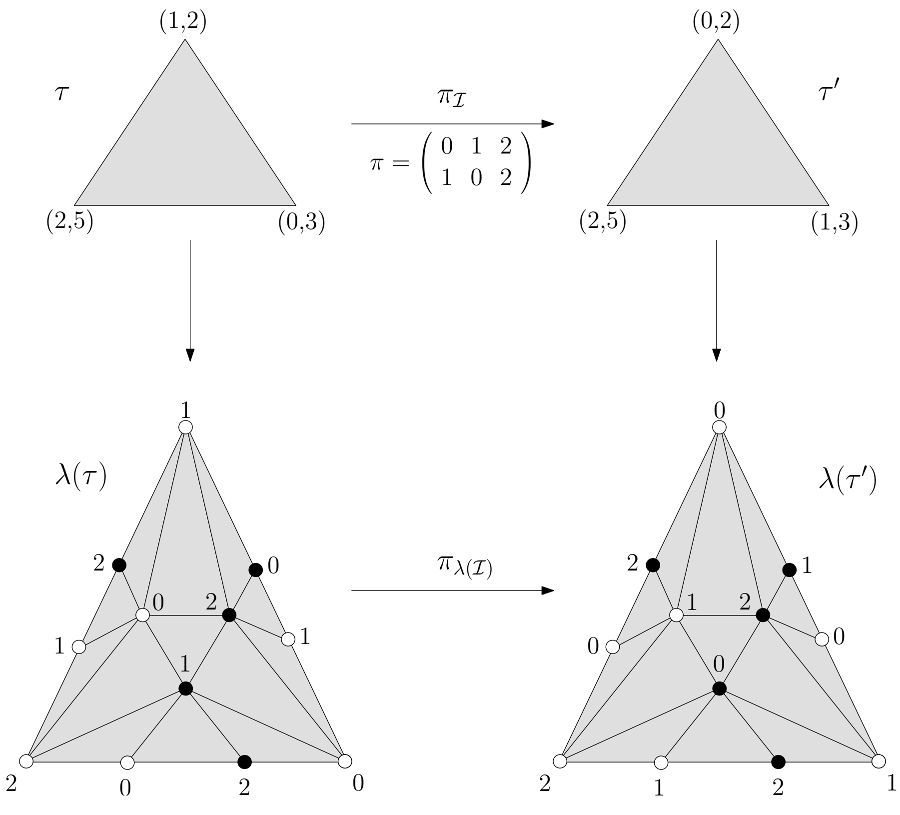

Figure 3.11: Obtaining $\lambda(\mathcal{I}^n_{wsb})$ and δ.

Let $\mathcal{K}^n$ be a ccosdi of σ^n under ψ, with a rank-symmetric binary coloring b and no monochromatic n-simplexes. In what follows, let $\sigma|i_0, \ldots, i_j$, $0 \leq j \leq n$, denote the face of σ^n with colors $i_0, \ldots, i_j$. Let $\lambda(\mathcal{I}^n_{wsb})$ be the chromatic subdivision of $\mathcal{I}^n_{wsb}$ obtained by subdividing each simplex of $\mathcal{I}^n_{wsb}$ with a subcomplex of $\mathcal{K}^n$ in the following way. Each simplex $\tau = \{(i_0, name_0), \ldots, (i_j, name_j)\}$ of $\mathcal{I}^n_{wsb}$, $0 \leq j \leq n$, with $name_0 < \ldots < name_j$, is subdivided with $\psi(\sigma|0, \ldots, j)$ in such a way that each vertex $(i_k, name_k) \in \tau$ is replaced with $\psi(\sigma|k)$. In other words, τ is subdivided with $\psi(\sigma|0, \ldots, j)$, preserving the rank of the *ids* of $\sigma|0, \ldots, j$ and the input names of τ. The resulting subdivision is denoted $\lambda(\tau)$. Every vertex $u \in \psi(\sigma|0, \ldots, j)$ induces a vertex $v \in \lambda(\tau)$ with two colorings: $id(v) = i_{id(u)}$ and $b(v) = b(u)$. Thus, $\lambda(\tau)$ is a chromatic subdivision of τ.

Figure 3.11 shows the subdivisions $\lambda(\tau)$ and $\lambda(\tau')$ for two simplexes of dimension 2. These subdivisions were obtained using the ccosdi $\mathcal{K}^2$ in Figure 3.10, which has a rank-symmetric binary coloring and a monochromatic 2-simplex. For the example it is not important that last property of $\mathcal{K}^2$. (in Chapter 5 it will be proved that any ccosdi of a 2-simplex with a rank-symmetric binary coloring, has at least one monochromatic 2-simplex)

Since each j-simplex $\tau \in \mathcal{I}^n_{wsb}$ is subdivided with $\psi(\sigma|0, \ldots, j)$ preserving the rank of the *ids* of $\sigma_{0,\ldots,j}$ and the input names of τ, then using the simplicial map $\pi_{\mathcal{I}} : \mathcal{I}^n_{wsb} \to \mathcal{I}^n_{wsb}$, defined above, we can get a simplicial bijection $\pi_{\lambda(\mathcal{I})} : \lambda(\mathcal{I}^n_{wsb}) \to \lambda(\mathcal{I}^n_{wsb})$ that respects π and preserves b. Figure 3.11 depicts a 2-dimensional example. Also, because $\mathcal{K}^n$ does not have monochromatic n-simplexes, $\lambda(\mathcal{I}^n_{wsb})$ does not have monochromatic n-simplexes, thus its colorings id and b specify a simplicial map $\delta : \lambda(\mathcal{I}^n_{wsb}) \to \mathcal{O}^n_{wsb}$ defined as $\delta(v) = (id(v), b(v))$, for each $v \in \lambda(\mathcal{I}^n_{wsb})$. Thus, δ agrees with WSB. Moreover, δ preserves b. Therefore, δ is anonymous, since $\pi_{\lambda(\mathcal{I})}$ and $\pi_{\mathcal{O}}$ also preserve b. Finally, it is not hard to see that $\lambda(\mathcal{I}^n_{wsb})$ is a ccosdi of $\mathcal{I}^n_{wsb}$ under φ such that $\varphi(\tau) = \lambda(\tau)$, for every $\tau \in \mathcal{I}^n_{wsb}$. The following result follows:

Theorem 3.5.2 *If there exists a ccosdi of an n-simplex with a rank-symmetric binary coloring and no monochromatic n-simplexes, then there exits an anonymous wait-free WSB protocol.*

Chapter 4

Combinatorial Topology Tools

This chapter presents the combinatorial topology tools that will be used to obtain two topological results, which will imply that WSB is not wait-free solvable for certain values of n, while it is solvable for the other values of n.

The first tool is Index Lemma. Intuitively, this lemma relates the number of properly colored n-simplexes of a colored n-pseudomanifold, with the number of properly colored $(n-1)$-simplexes on its boundary. Using Index Lemma, the chapter shows that the boundary of a chromatic and binary colored n-pseudomanifold, induces the number of monochromatic n-simplexes inside it. The second tool is the cone construction, which is a generalization of the standard chromatic subdivision. This operation is useful for constructing divided images recursively.

4.1 Index Lemma and Counting Monochromatic Simplexes

Index Lemma counts the properly colored n-simplexes inside a (not necessarily properly) colored and coherently oriented pseudomanifold, "the content", by counting the properly colored $(n-1)$-simplexes on the boundary, "the index".

Consider an oriented simplex σ with a proper coloring c. Let $\langle c_0, \ldots, c_{dim(\sigma)} \rangle$ be the sequence of the c colors of σ in ascending order. Simplex σ is *counted by orientation with respect to* c in the following way. It is counted as $+1$ if the sequence $\langle c_0, \ldots, c_{dim(\sigma)} \rangle$ belongs to its orientation, i.e., the sequence of vertexes $\langle v_0, v_1, \ldots, v_n \rangle$ such that $c(v_i) = c_i$, $0 \leq i \leq dim(n)$, belongs to the orientation of σ. Otherwise it is counted as -1. For example, the 2-simplex in Figure 4.1 is counterclockwise oriented and it is counted as -1 because colors 0, 1 and 2, in this order, follow the clockwise direction. In what follows, for $i \in ID^n$, let ID^n_i be the set of colors $ID^n - \{i\}$.

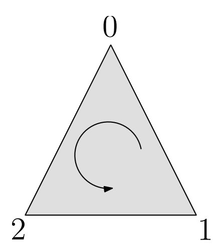

Figure 4.1: Counting by orientation.

Definition 4.1.1 (Index and Content) *Consider a coherently oriented pseudomanifold $\mathcal{K}^n$ with the induced orientation on its boundary. Let c be a coloring, not necessarily proper, of $\mathcal{K}^n$ with ID^n.*

1. *The* content *of $\mathcal{K}^n$, $\mathcal{C}(\mathcal{K}^n)$, is the number of the properly colored n-simplexes of $\mathcal{K}^n$ counted by orientation.*
2. *The* index i *of $\mathcal{K}^n$, $\mathcal{I}_i(\mathcal{K}^n)$, is the number of properly colored $(n-1)$-simplexes of $bd(\mathcal{K}^n)$ with ID_i^n counted by orientation.*

If there is no ambiguity, we just write $\mathcal{C}$ or $\mathcal{I}_i$. Lemma 4.1.2 below is the restatement of Corollary 2 in [36] using our notation. This lemma is a generalization of the 2-dimensional Index Lemma in [50, pp. 46-47]. For completeness, a proof is presented.

Lemma 4.1.2 (Index Lemma) *Let $\mathcal{K}^n$ be a coherently oriented, connected and colored pseudomanifold with ID^n. Then $\mathcal{C} = (-1)^i\mathcal{I}_i$.*

Proof: First, we prove that the lemma holds when $\mathcal{K}^n = \mathcal{M}(\sigma)$, for an n-simplex σ. There are two cases. In the first case, σ is properly colored and hence $\mathcal{C} = \pm 1$. Suppose, w.l.o.g., that σ is positively oriented, i.e., the sequence $\langle 0, 1, \dots n\rangle$ belong to its orientation. Thus, $\mathcal{C} = +1$. Let σ_i be the $(n-1)$-face of σ without the vertex colored i under c. Notice that if i is even, σ_i has the induced positive orientation and hence $\langle 0, 1, \dots \widehat{i} \dots n\rangle$ belongs to its orientation. Therefore, $\mathcal{I}_i = +1$. Similarly, if i is odd, σ_i has the induced negative orientation and thus $\mathcal{I}_i = -1$. Therefore, $\mathcal{C} = (-1)^i\mathcal{I}_i$.

In the second case, σ is not properly colored, and then $\mathcal{C} = 0$. Clearly $\mathcal{I}_i = 0$ if σ has no properly colored $(n-1)$-faces with ID_i^n. Thus, suppose that at least one $(n-1)$-face of σ is properly colored with ID_i^n. Observe that exactly two $(n-1)$-faces of σ are properly colored with ID_i^n. Let $v_0, v_1 \dots v_n$ denote the vertexes of σ such that $c(v_0) = 0, c(v_1) = 1, \dots c(v_i) = i, c(v_{i+1}) = i, c(v_{i+2}) = i+1, \dots c(v_n) = n-1$. Suppose, w.l.o.g., that σ is oriented $\langle v_0, v_1 \dots v_n\rangle$. Let σ_i and σ_{i+1} be the $(n-1)$-faces of σ without vertexes v_i and v_{i+1}, respectively. If i is even then σ_i is oriented $\langle v_0, v_1, \dots \widehat{v_i} \dots v_n\rangle$ and σ_{i+1} is oriented $\langle v_1, v_0, \dots \widehat{v_{i+1}} \dots v_n\rangle$. Notice that σ_i and σ_{i+1} are counted as $+1$ and -1, respectively, by $\mathcal{I}_i$, hence $\mathcal{I}_i = 0$. Something similar happens when i is odd. Thus, $\mathcal{C} = (-1)^i\mathcal{I}_i = 0$.

We now prove that the lemma holds for an arbitrary pseudomanifold $\mathcal{K}^n$. Let $\mathcal{S}_i$ be the number of properly colored $(n-1)$-simplexes of $\mathcal{K}^n$ with ID_i^n, counting by orientation. Thus, each internal properly colored $(n-1)$-simplex is counted twice, as it has two induced orientations (it is shared by two n-simplexes), and each external properly colored $(n-1)$-simplex is counted one time, as it has just one induced orientation. Observe that every internal properly colored $(n-1)$-simplex of $\mathcal{K}^n$ with ID_i^n, adds 0 to $\mathcal{S}_i$, since it has opposite induced orientations. Hence, $\mathcal{I}_i(\mathcal{K}^n) = \mathcal{S}_i(\mathcal{K}^n)$. Consider now an n-simplex $\sigma \in \mathcal{K}^n$. It was proved above that $\mathcal{C}(\mathcal{M}(\sigma)) = (-1)^i\mathcal{I}_i(\mathcal{M}(\sigma))$. And also $\mathcal{I}_i(\mathcal{M}(\sigma)) = \mathcal{S}_i(\mathcal{M}(\sigma))$. Thus, $\mathcal{C}(\mathcal{M}(\sigma)) = (-1)^i\mathcal{S}_i(\mathcal{M}(\sigma))$. From the definitions of $\mathcal{C}$ and $\mathcal{S}_i$, it is not hard to see that $\mathcal{C}(\mathcal{K}^n) = (-1)^i\mathcal{S}_i(\mathcal{K}^n)$ and hence $\mathcal{C}(\mathcal{K}^n) = (-1)^i\mathcal{I}_i(\mathcal{K}^n)$.

■

Figure 4.2 shows a pseudomanifold with its 2-simplexes counterclockwise oriented. Notice that colors 0, 1 and 2, in this order, of the unique properly colored 2-simplex, denoted by the circular arrow, follow the counterclockwise direction, and thus $\mathcal{C} = +1$. An edge in the boundary with colors

0 and 1, is counted $+1$ or -1 according to its induced orientation and the direction followed by 0 and 1, in this order. Hence $\mathcal{I}_2 = +1$. It can be easily verified that $(-1)^2\mathcal{I}_2 = (-1)^1\mathcal{I}_1 = (-1)^0\mathcal{I}_0$. The reader familiar with topology may notice that the coloring c induces a simplicial map from $\mathcal{K}^n$ to a properly colored simplex σ^n. Thus we can think of the index of $\mathcal{K}^n$ as the number of times that $bd(\mathcal{K}^n)$ is "wrapped around" $bd(\sigma^n)$, i.e., a combinatorial version of the notion of degree.

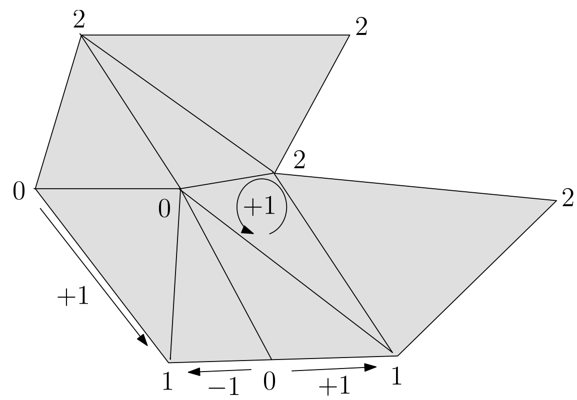

Figure 4.2: The Index Lemma.

Index Lemma says that the boundary induces the content, the number of properly colored simplexes inside the pseudomanifold (counting by orientation). Therefore, two orientable pseudomanifolds with the same boundary have the same content.

Lemma 4.1.3 *Consider two orientable, connected and colored pseudomanifolds $\mathcal{K}^n$ and $\mathcal{L}^n$. If $bd(\mathcal{K}^n) = bd(\mathcal{L}^n)$ then $\mathcal{C}(\mathcal{K}^n) = \mathcal{C}(\mathcal{L}^n)$.*

Subsequent chapters are interested in the number of monochromatic simplexes of a chromatic pseudomanifold with a binary coloring. In what follows it is showed that Index Lemma, which is about colorings that uses $n+1$ colors, can be adapted to count such simplexes.

For a chromatic pseudomanifold with a binary coloring, it is defined the coloring c, Definition 4.1.4, that uses colors ID^n.

Definition 4.1.4 *Let $\mathcal{K}^n$ be a chromatic pseudomanifold with a binary coloring. For every $v \in \mathcal{K}^n$, the coloring c is defined as $c(v) = (id(v) + b(v)) \mod (n+1)$, where id and b are the chromatic and binary coloring of $\mathcal{K}^n$, respectively.*

Figure 4.3 presents a 2-pseudomanifold with the three colorings, id, b and c, associated to each vertex. The binary coloring b is represented by white and black circles, and the id and c colorings, in this order, are the numbers near the vertexes.

Lemma 4.1.5 proves that the number of monochromatic n-simplexes under b and the properly colored n-simplexes under c are related. Thus the boundary induces the number of monochromatic n-simplexes, by Index Lemma.

Lemma 4.1.5 *Let $\mathcal{K}^n$ be chromatic pseudomanifold with a binary coloring b and a coloring c as in Definition 4.1.4. An n-simplex of $\mathcal{K}^n$ is monochromatic under b if and only if it is properly colored under c.*

Proof: Let id the chromatic coloring of $\mathcal{K}^n$. From the definition of c, Definition 4.1.4, it is easy to see that every monochromatic n-simplex under b is properly colored under c. For the other direction, assume, for sake of contradiction, that a properly colored n-simplex τ under c, is not monochromatic under b. Let τ_i be the maximal face of τ such that for every $v \in \tau_i$, $b(v) = i$, i.e., the maximal i-monochromatic face of τ. By definition of c, Definition 4.1.4, for every $v \in \tau_0$, $c(v) = id(v)$, and for every $v \in \tau_1$, $c(v) = (id(v) + 1) \mod (n + 1)$. Since τ is properly colored under c, for every $v \in \tau_1$, τ_1 contains the unique vertex $u \in \tau$ such that $c(v) = id(u)$. Therefore, τ_1 contains all vertexes of τ and then τ is monochromatic.

■

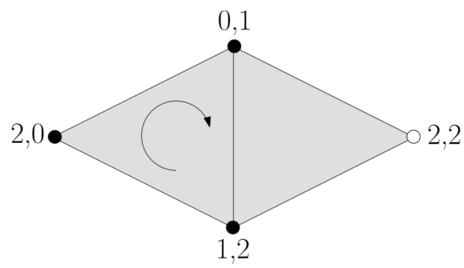

Figure 4.3: A pseudomanifold with the three colorings

Lemma 4.1.6[1] below shows that $\mathcal{C}$ can be easily computed by counting the monochromatic n-simplexes according to their orientation. Recall that an oriented and properly colored n-simplex has orientation $+1$ if the sequence $\langle 0, 1, \ldots, n\rangle$ belongs to its orientation, otherwise it has orientation -1. This lemma says that a 0-monochromatic n-simplex with orientation d is counted as d, but a 1-monochromatic n-simplex with orientation d is counted as d if n is even and as $-d$ if n is odd.

For example, assume the 2-simplexes in Figure 4.3 are counterclockwise oriented. The id colors 1, 0 and 2, in this order, of the unique monochromatic simplex, follow the counterclockwise direction and therefore the simplex has orientation -1. Also observe that the c colors 0, 1 and 2 follow the clockwise direction, as it is indicated by the arrow. Because the simplex has the opposite orientation, it is counted as -1 by $\mathcal{C}$, exactly as Lemma 4.1.6 states.

Lemma 4.1.6 *Let $\mathcal{K}^n$ be a chromatic, coherently oriented and connected pseudomanifold with a binary coloring b and a coloring c as in Definition 4.1.4. Let τ be a b-monochromatic n-simplex of $\mathcal{K}^n$ with orientation d. The simplex τ is counted as $(-1)^{b*n}d$ by $\mathcal{C}$.*

Proof: Let id be the chromatic coloring of $\mathcal{K}^n$. Consider the sequence $S = \langle v_0, v_1, \ldots, v_n\rangle$ of τ such that $id(v_i) = i$, i.e., considering the coloring id, the sequence S is $\langle 0, 1, \ldots, n\rangle$. By definition of orientability, if S belongs to the orientation of τ then τ has orientation $d = +1$, otherwise it has orientation $d = -1$.

Suppose first τ is 0-monochromatic. By definition of the coloring c, Definition 4.1.4, $c(v_i) = id(v_i) = i$. Thus, considering the coloring c, the sequence S is $\langle 0, 1, \ldots, n\rangle$. Therefore, if S belongs to the orientation of τ then $\mathcal{C}$ counts τ as $+1$, otherwise as -1. That is, τ is counted as d.

[1]Actually, Lemmas 4.1.5 and 4.1.6 hold for any coloring c in which $c(v) = id(v)$ if $b(v) = 0$, and $c(v) = f(v)$ if $b(v) = 1$, where $f : ID^n \to ID^n$ is a permutation such that the restriction $f|I$ to any proper subset $I \subset ID^n$ is not a permutation of I.

Suppose now τ is 1-monochromatic. By definition of the coloring c, Definition 4.1.4, $c(v_i) = (i+1) \bmod (n+1)$, i.e., considering the coloring c, the sequence S is $\langle 1, \ldots, n, 0 \rangle$. Observe that applying n permutations over S, we can get the sequence $S' = \langle v_n, v_0, \ldots, v_{n-1} \rangle$. Considering the coloring c, the sequence S' is $\langle 0, 1, \ldots, n \rangle$. Notice that if n is even then S' belongs to the orientation of τ if and only if S belongs to the orientation of τ. Thus $\mathcal{C}$ counts τ as d. Similarly, if n is odd then S' belongs to the orientation of τ if and only if S does not belong to the orientation of τ. Thus $\mathcal{C}$ counts τ as $-d$.

■

4.2 The Cone Construction

This section introduces the cone construction operation, which is a generalization of the standard chromatic subdivision. First, some definitions are presented.

Let σ and τ be two simplexes. The *join* of σ and τ, denoted $\sigma * \tau$, is the simplex $\sigma \cup \tau$. Consider two complexes $\mathcal{K}$ and $\mathcal{L}$. The *join* of $\mathcal{K}$ and $\mathcal{L}$, $\mathcal{K} * \mathcal{L}$, is the complex $\{\sigma * \tau | \sigma \in \mathcal{K} \text{ and } \tau \in \mathcal{L}\}$. Since $\mathcal{K}$ and $\mathcal{L}$ contain the empty simplex, $\mathcal{K}, \mathcal{L} \subseteq \mathcal{K} * \mathcal{L}$. If $\mathcal{K}$ only contains one simplex σ, we just write $\sigma * \mathcal{L}$. The following is a "chromatic" version of the join operator for properly colored simplexes and complexes. Let σ and τ be properly colored simplexes under id. Simplexes σ and τ are *compatible* if $id(\sigma) \cap id(\tau) = \emptyset$. From now on, we only consider the join operator for compatible simplexes. Therefore, if $\mathcal{K}$ and $\mathcal{L}$ are chromatic complexes then $\mathcal{K} * \mathcal{L} = \{\sigma * \tau | \sigma \in \mathcal{K}, \tau \in \mathcal{L} \text{ and } \sigma \text{ and } \tau \text{ are compatible}\}$.

Consider a ccodi $\varphi(bd(\sigma^n))$ of $bd(\sigma^n)$ and a properly colored simplex τ^n with ID^n. Intuitively, the cone over $\varphi(bd(\sigma^n))$ for τ^n, is obtained by putting τ^n at the center of $\varphi(bd(\sigma^n))$ and joining every face of τ^n with the j-simplexes of $\varphi(bd(\sigma^n))$ with compatible carriers of dimension j. Figure 4.4 contains an example of the cone of dimension 2 in which $\varphi(bd(\sigma^n))$ is the boundary of the complex and τ^n is the simplex at the center.

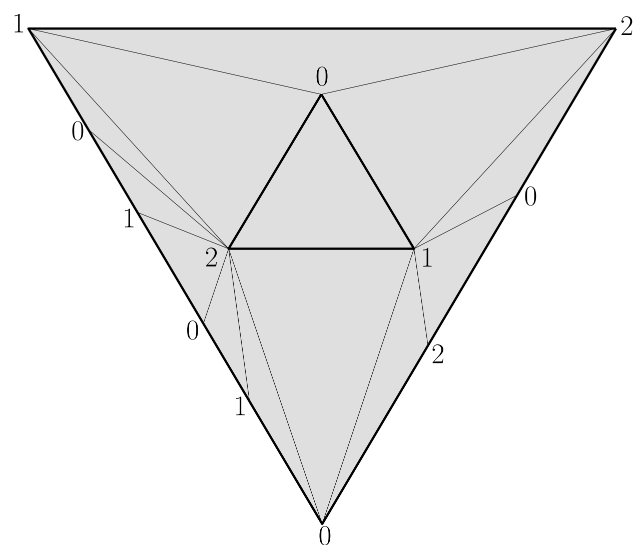

Figure 4.4: The cone construction.

Definition 4.2.1 *Let $\varphi(bd(\sigma^n))$ be a ccodi of $bd(\sigma^n)$ and τ^n be a properly colored simplex with ID^n. The* cone *over $\varphi(bd(\sigma^n))$ for τ^n, denoted $\tau^n \circledast \varphi(bd(\sigma^n))$, is the complex consisting of the union of complexes $\tau * \varphi(\sigma)$, for every compatible simplexes $\tau \in \mathcal{M}(\tau^n)$ and $\sigma \in bd(\sigma^n)$.*

Notice that $\mathcal{M}(\tau^n), \varphi(bd(\sigma^n)) \subset \tau^n \circledast \varphi(bd(\sigma^n))$. The following lemma that directly follows from Definition 4.2.1 and characterizes the n-simplexes of an n-dimensional cone.

Lemma 4.2.2 *For each $\gamma \in \varphi(bd(\sigma^n))$ with $dim(\gamma) = dim(carr(\gamma))$, there is a unique non-empty face of τ^n such that $\tau * \gamma$ is an n-simplex of $\tau^n \circledast \varphi(bd(\sigma^n))$. Simplex $\tau * \gamma$ is the n-simplex* generated *by γ. Moreover, each n-simplexes in $\tau^n \circledast \varphi(bd(\sigma^n))$ is generated in this way.*

Lemma 4.2.3 presents the main property of the cone construction.[2]

Lemma 4.2.3 *Let $\varphi(bd(\sigma^n))$ be a ccodi of $bd(\sigma^n)$ and τ^n be a properly colored simplex with ID^n. The cone $\tau^n \circledast \varphi(bd(\sigma^n))$ is a ccodi of σ^n under the extension ψ of φ in which $\psi(\sigma^n) = \tau^n \circledast \varphi(bd(\sigma^n))$. Moreover, if $\varphi(bd(\sigma^n))$ ccosdi of $bd(\sigma^n)$, then $\tau^n \circledast \varphi(bd(\sigma^n))$ is a ccosdi of σ^n.*

Generally speaking, the proof of Lemma 4.2.3 consists on the following. We first prove that $\mathcal{M}(\tau^n) * bd(\sigma^n)$ is a ccodi of σ^n under ξ that maps $bd(\sigma^n)$ to $bd(\sigma^n)$, and σ^n to $\mathcal{M}(\tau^n) * bd(\sigma^n)$, Lemmas 4.2.4 and 4.2.5. Then we show that $\tau^n \circledast \varphi(bd(\sigma^n))$ is a ccodi of $\mathcal{M}(\tau^n) * bd(\sigma^n)$ under ϕ that maps $bd(\mathcal{M}(\tau^n) * bd(\sigma^n))$ to the boundary of the cone, and $\mathcal{M}(\tau^n) * bd(\sigma^n)$ to all the cone, Lemmas 4.2.6 and 4.2.7. As we shall see, the composition $\phi \circ \xi$ is exactly the extension ψ in Lemma 4.2.3. Figure 4.5 depicts examples of Lemmas 4.2.4 and 4.2.6.

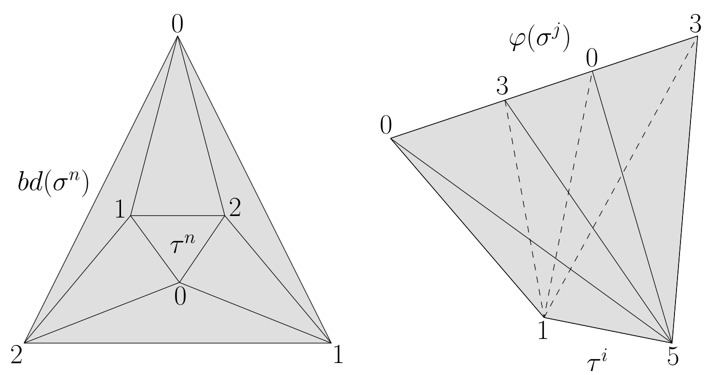

Figure 4.5: Examples of Lemmas 4.2.4 and 4.2.6.

Since any complex contains the empty simplex, for complexes $\mathcal{K}$ and $\mathcal{L}$, any simplex of $\mathcal{K} * \mathcal{L}$ can be represented as the join of two simplexes of $\mathcal{K}$ and $\mathcal{L}$, respectively. Therefore, in what follows, $\tau * \gamma \in \mathcal{K} * \mathcal{L}$ denotes a simplex of $\mathcal{K} * \mathcal{L}$ such that $\tau \in \mathcal{K}$ and $\gamma \in \mathcal{L}$.

Lemma 4.2.4 *The complex $\mathcal{M}(\tau^n) * bd(\sigma^n)$ is a chromatic, connected and orientable n-pseudo-manifold with $bd(\mathcal{M}(\tau^n) * bd(\sigma^n)) = bd(\sigma^n)$. Moreover, $\mathcal{M}(\tau^n) * bd(\sigma^n)$ is a subdivision of an n-simplex.*

[2] Lemma 4.2.3 is still true in a extended definition of the cone in which τ^n is replaced by a ccodi or ccosdi of it.

Proof: It is clear that $\mathcal{M}(\tau^n) * bd(\sigma^n)$ is a chromatic complex, and also it is complete for dimension n. In order to prove that it is an n-pseudomanifold, consider an $(n-1)$-simplex $\tau^i * \sigma^j \in \mathcal{M}(\tau^n) * bd(\sigma^n)$. Observe that σ^j and τ^i can be $\emptyset$, but not both at the same time. Consider $k \in id(\tau^n)$ such that $k \notin id(\tau^i * \sigma^j)$. Suppose first $\tau^i \neq \emptyset$. Notice that $\tau^i * \sigma^j$ only belongs to the n-simplexes $\tau^{i+1} * \sigma^j$ and $\tau^i * \sigma^{j+1}$, where $\tau^{i+1} \subseteq \tau^n$ and $\sigma^{j+1} \subset \sigma^n$ such that $\tau^i \subset \tau^{i+1}$, $\sigma^j \subset \sigma^{j+1}$, $k \in id(\tau^{i+1})$ and $k \in id(\sigma^{j+1})$. Suppose now $\tau^i = \emptyset$. In this case, $\tau^i * \sigma^j = \sigma^j$ only belongs to the n-simplex $v * \sigma^j$, where $v \in \tau^n$ and $id(v) = k$. Moreover, observe that $\sigma^j \in bd(\sigma^n)$. Therefore, $\mathcal{M}(\tau^n) * bd(\sigma^n)$ is an n-pseudomanifold with $bd(\mathcal{M}(\tau^n) * bd(\sigma^n)) = bd(\sigma^n)$.

We now prove that $\mathcal{M}(\tau^n) * bd(\sigma^n)$ is orientable. Consider an n-simplex $\tau^i * \sigma^j$. In the previous paragraph we can see that an n-simplex of $\mathcal{M}(\tau^n) * bd(\sigma^n)$ sharing an $(n-1)$-face with $\tau^i * \sigma^j$, has the form either $\tau^{i+1} * \sigma^{j-1}$ or $\tau^{i-1} * \sigma^{j+1}$. Assume that an n-simplex $\tau * \sigma$ has orientation $+1$ if $dim(\sigma)$ is odd, otherwise it has orientation -1. Observe that any two n-simplexes sharing an $(n-1)$-face, have opposite orientations. Thus, by Lemma 3.1.1, $\mathcal{M}(\tau^n) * bd(\sigma^n)$ is orientable.

For an n-simplex $\tau^i * \sigma^j$, we have seen that if $\sigma^j \neq \emptyset$ then there exists an n-simplex $\tau^{i+1} * \sigma^{j-1}$ that shares an $(n-1)$-face with it. Therefore, there exists an n-path from any n-simplex to τ^n and hence there exists an n-path between any two n-simplexes. That is, $\mathcal{M}(\tau^n) * bd(\sigma^n)$ is connected. Finally, it is easy to see that $\mathcal{M}(\tau^n) * bd(\sigma^n)$ is a subdivision of an n-simplex.

■

Lemma 4.2.5 *The complex $\mathcal{M}(\tau^n) * bd(\sigma^n)$ is a ccosdi of σ^n under ξ, where for each $\sigma \subset \sigma^n$, $\xi(\sigma) = \sigma$, and for σ^n, $\xi(\sigma^n) = \mathcal{M}(\tau^n) * bd(\sigma^n)$.*

Proof: Notice that $\xi(bd(\sigma^n)) = bd(\sigma^n)$ and hence $\xi(bd(\sigma^n))$ is a ccosdi of $bd(\sigma^n)$ under $\xi|_{bd(\sigma^n)}$. Also, by Lemma 4.2.4, $\xi(\sigma^n) = \mathcal{M}(\tau^n) * bd(\sigma^n)$ is a chromatic, connected and orientable n-pseudomanifold with $bd(\xi(\sigma^n)) = bd(\sigma^n) = \xi(bd(\sigma^n))$. Moreover, $\xi(\sigma^n)$ is also a subdivision of an n-simplex. Therefore, $\mathcal{M}(\tau^n) * bd(\sigma^n)$ is a ccosdi of σ^n under ξ.

■

Lemma 4.2.6 *Consider a simplex $\tau * \sigma \in \mathcal{M}(\tau^n) * bd(\sigma^n)$. The complex $\tau * \varphi(\sigma)$ is a chromatic, connected and orientable $dim(\tau * \sigma)$-pseudomanifold with $bd(\tau * \varphi(\sigma)) = \cup_{\tau' * \sigma' \in bd(\tau * \sigma)} \tau' * \varphi(\sigma')$. Moreover, if $\varphi(\sigma)$ is a subdivision of σ, $\tau * \varphi(\sigma)$ is a subdivision of $\tau * \sigma$.*

Proof: It is clear that $\tau * \varphi(\sigma)$ is a chromatic complex, and also it is complete for dimension $dim(\tau * \sigma)$. In order to prove it is a $dim(\tau * \sigma)$-pseudomanifold, consider a $(dim(\tau * \sigma) - 1)$-simplex $\rho * \gamma \in \tau * \varphi(\sigma)$. Notice that either (a) $dim(\rho) = dim(\tau) - 1$ and $dim(\gamma) = dim(\sigma)$, or (b) $dim(\rho) = dim(\tau)$ and $dim(\gamma) = dim(\sigma) - 1$. By Definition 3.1.2 of a divided image, $\varphi(\sigma)$ is a $dim(\sigma)$-pseudomanifold with $bd(\varphi(\sigma)) = \varphi(bd(\sigma))$. For case (a), observe that $\tau * \gamma$ is the unique $dim(\tau * \sigma)$-simplex containing $\rho * \gamma$. For case (b), we have two subcases: If $\gamma \in bd(\varphi(\sigma))$ then $\rho * \gamma$ only belongs to the $dim(\tau * \sigma)$-simplex $\rho * \lambda$, where λ is the unique $dim(\sigma)$-simplex of $\varphi(\sigma)$ containing γ. And If $\gamma \notin bd(\varphi(\sigma))$ then $\rho * \gamma$ only belongs to the $dim(\tau * \sigma)$-simplexes $\rho * \lambda_1$ and $\rho * \lambda_2$, where λ_1 and λ_2 are the unique distinct $dim(\sigma)$-simplexes of $\varphi(\sigma)$ containing γ. Therefore, $\tau * \varphi(\sigma)$ is a $dim(\tau * \sigma)$-pseudomanifold. Moreover, because $bd(\varphi(\sigma)) = \varphi(bd(\sigma))$, we conclude

$$\begin{aligned} bd(\tau * \varphi(\sigma)) &= (\cup_{\tau' \in bd(\tau)} \tau' * \varphi(\sigma)) \cup (\cup_{\sigma' \in bd(\sigma)} \tau * \varphi(\sigma')) \\ &= \cup_{\tau' * \sigma' \in bd(\tau * \sigma)} \tau' * \varphi(\sigma') \end{aligned}$$

We now prove that $\tau * \varphi(\sigma)$ is orientable. Assume $\varphi(\sigma)$ has a coherent orientation. By Lemma 3.1.1, every two $dim(\sigma)$-simplexes of $\varphi(\sigma)$ that share a $(dim(\sigma)-1)$-face, have opposite orientations. Consider a $dim(\tau * \sigma)$-simplex $\rho * \gamma \in \tau * \varphi(\sigma)$. Assume $\rho * \gamma$ has the same orientation as γ in $\varphi(\sigma)$. As it is explained in the last paragraph (case (b) second subcase), a $dim(\tau * \sigma)$-simplex $\varrho * \lambda \in \tau * \varphi(\sigma)$ contains a $(dim(\tau * \sigma) - 1)$-face of $\rho * \gamma$ if and only if γ and λ share a $(dim(\sigma) - 1)$-face. Therefore, $\rho * \gamma$ and $\varrho * \lambda$ have opposite orientations, and hence, by Lemma 3.1.1, $\tau * \varphi(\sigma)$ is orientable.

For proving that $\tau * \varphi(\sigma)$ is $dim(\tau * \sigma)$-connected, just notice that, for any two $dim(\tau * \sigma)$-simplexes $\rho * \gamma$ and $\varrho * \lambda$, there is a $dim(\tau * \sigma)$-path in $\tau * \varphi(\sigma)$ because there is a $dim(\sigma)$-path in $\varphi(\sigma)$ between γ and λ (by assumption $\varphi(\sigma)$ is $dim(\sigma)$-connected). Finally, it is not hard to see that if $\varphi(\sigma)$ is a subdivision of σ then $\tau * \varphi(\sigma)$ is a subdivision of $\tau * \sigma$.

■

Lemma 4.2.7 *The cone $\tau^n \circledast \varphi(bd(\sigma^n))$ is a ccodi of $\mathcal{M}(\tau^n) * bd(\sigma^n)$ under ϕ defined as $\phi(\tau * \sigma) = \tau * \varphi(\sigma)$ for every $\tau * \sigma \in \mathcal{M}(\tau^n) * bd(\sigma^n)$. Also, if $\varphi(bd(\sigma^n))$ ccosdi of $bd(\sigma^n)$ then $\tau^n \circledast \varphi(bd(\sigma^n))$ is a ccosdi of $\mathcal{M}(\tau^n) * bd(\sigma^n)$.*

Proof: It is not hard to see that ϕ holds conditions 1, 2 and 3 in Definition 3.1.2 of a divided image. For condition 4, consider $\tau_1 * \sigma_1, \tau_2 * \sigma_2 \in \mathcal{M}(\tau^n) * bd(\sigma^n)$. By definitions of ϕ and join operator,

$$
\begin{aligned}
\phi(\tau_1 * \sigma_1) \cap \phi(\tau_2 * \sigma_2) &= (\tau_1 * \varphi(\sigma_1)) \cap (\tau_2 * \varphi(\sigma_2)) \\
&= \{\tau_1 * \gamma | \gamma \in \varphi(\sigma_1)\} \cap \{\tau_2 * \gamma | \gamma \in \varphi(\sigma_2)\} \\
&= \{(\tau_1 \cap \tau_2) * \gamma | \gamma \in \varphi(\sigma_1) \cap \varphi(\sigma_2)\} \\
&= (\tau_1 \cap \tau_2) * (\varphi(\sigma_1) \cap \varphi(\sigma_2))
\end{aligned}
$$

and by condition 4 in Definition 3.1.2,

$$
\begin{aligned}
(\tau_1 \cap \tau_2) * (\varphi(\sigma_1) \cap \varphi(\sigma_2)) &= (\tau_1 \cap \tau_2) * \varphi(\sigma_1 \cap \sigma_2) \\
&= \phi((\tau_1 \cap \tau_2) * (\sigma_1 \cap \sigma_2))
\end{aligned}
$$

Also, observe that $(\tau_1 * \sigma_1) \cap (\tau_2 * \sigma_2) = (\tau_1 \cap \tau_2) * (\sigma_1 \cap \sigma_2)$, and thus ϕ holds condition 4. For condition 5, consider a simplex $\tau * \sigma \in \mathcal{M}(\tau^n) * bd(\sigma^n)$. By Lemma 4.2.6, $\phi(\tau * \sigma) = \tau * \varphi(\sigma)$ is a $dim(\tau * \sigma)$-pseudomanifold with $bd(\phi(\tau * \sigma)) = bd(\tau * \varphi(\sigma)) = \cup_{\tau' * \sigma' \in bd(\tau * \sigma)} \tau' * \varphi(\sigma') = \phi(bd(\tau * \sigma))$. Therefore, $\tau^n \circledast \varphi(bd(\sigma^n))$ is a divided image of $\mathcal{M}(\tau^n) * bd(\sigma^n)$ under ϕ. Also, by Lemma 4.2.6, for each $\tau * \sigma \in \mathcal{M}(\tau^n) * bd(\sigma^n)$, $\phi(\tau * \sigma) = \tau * \varphi(\sigma)$ is chromatic, connected and orientable. Moreover, for every $\tau' * \sigma' \in bd(\tau * \sigma)$, $\phi(\tau' * \sigma') = \tau' * \varphi(\sigma')$ is connected and hence $\phi(bd(\tau * \sigma))$ is connected, if $dim(\tau * \sigma) \geq 2$. Therefore $\tau^n \circledast \varphi(bd(\sigma^n))$ is a ccodi of $\mathcal{M}(\tau^n) * bd(\sigma^n)$.

Now, if $\varphi(bd(\sigma^n))$ ccosdi of $bd(\sigma^n)$ then for each $\tau * \sigma \in \mathcal{M}(\tau^n) * bd(\sigma^n)$, $\phi(\tau * \sigma) = \tau * \varphi(\sigma)$ a subdivision of $\tau * \sigma$, because $\varphi(\sigma)$ is a subdivision and by Lemma 4.2.6. Thus, $\tau^n \circledast \varphi(bd(\sigma^n))$ is a ccosdi of $\mathcal{M}(\tau^n) * bd(\sigma^n)$.

■

Proof of Lemma 4.2.3: Consider maps ξ and ϕ as defined in Lemmas 4.2.5 and 4.2.7. We have that $\phi \circ \xi(\sigma^n) = \tau^n \circledast \varphi(bd(\sigma^n))$. And by Lemma 4.2.4, $bd(\mathcal{M}(\tau^n) * bd(\sigma^n)) = bd(\sigma^n)$, and so

$\phi \circ \xi(bd(\sigma^n)) = \varphi(bd(\sigma^n))$. Therefore, $\psi = \phi \circ \xi$. Also, we have that $\xi(\sigma^n) = \mathcal{M}(\tau^n) * bd(\sigma^n)$. And by Lemma 4.2.6, for each simplex $\tau * \sigma \in \mathcal{M}(\tau^n) * bd(\sigma^n)$, $\phi(\tau * \sigma)$ is a chromatic, connected and orientable $dim(\tau * \sigma)$-pseudomanifold. In addition, by Definition 3.1.2 of a divided image, $bd(\phi(\tau * \sigma)) = \phi(bd(\tau * \sigma))$. In other words, every simplex of $\mathcal{M}(\tau^n) * bd(\sigma^n)$ is replaced in $\tau^n \circledast \varphi(bd(\sigma^n))$ by a chromatic, connected and orientable pseudomanifold that respects its boundary. Therefore, $\tau^n \circledast \varphi(bd(\sigma^n))$ is also a ccodi of σ^n under $\phi \circ \xi$. Finally, if $\varphi(bd(\sigma^n))$ ccosdi of $bd(\sigma^n)$, $\mathcal{M}(\tau^n) * bd(\sigma^n)$ is a ccosdi of σ^n and $\tau^n \circledast \varphi(bd(\sigma^n))$ is a ccosdi of $\mathcal{M}(\tau^n) * bd(\sigma^n)$, by Lemmas 4.2.5 and 4.2.7, hence $\tau^n \circledast \varphi(bd(\sigma^n))$ is a ccosdi of σ^n.

■

Chapter 5

WSB Impossibility for Non-Exceptional Values of n

This chapter presents a characterization of the number of monochromatic n-simplexes of a ccodi $\mathcal{K}^n$ of an n-simplex, with a symmetric binary coloring. This characterization, together with a well-known result in number theory, implies that, for certain non-exceptional values of n, $\mathcal{K}^n$ has at least one monochromatic n-simplex, thus WSB is not solvable, as explained in Chapter 3.

Two purely combinatorial proofs of the characterization are presented. The first step in both proofs is that $bd(\mathcal{K}^n)$ determines the number of monochromatic n-simplexes of $\mathcal{K}^n$. In the first proof, the interior of $\mathcal{K}^n$ is replaced by another "convenient" pseudomanifold, to get another ccodi $\mathcal{L}^n$ with $bd(\mathcal{K}^n) = bd(\mathcal{L}^n)$, hence with the same number of monochromatic n-simplexes. A simple counting argument using the symmetry of the binary coloring gives the characterization.

The second proof consists of an inductive process that characterizes the number of monochromatic n-simplexes of $\mathcal{K}^n$ by directly counting simplexes on $bd(\mathcal{K}^n)$. Roughly, the inductive process starts with a binary coloring on $bd(\mathcal{K}^n)$ such that each vertex has binary color 0, and then gradually changes the coloring of the vertexes until $bd(\mathcal{K}^n)$ gets its original coloring. The characterization comes from computing how each one of these changes affects the number of monochromatic n-simplexes, considering that each time the binary coloring of a vertex is changed, the coloring of the corresponding isomorphic vertexes on the boundary also has to change, to preserve the symmetry of the binary coloring.

5.1 A Combinatorial Characterization

Consider a ccodi $\mathcal{K}^n$ of an n-simplex with a symmetric binary coloring. Let id and b be the chromatic and binary colorings of $\mathcal{K}^n$. Assume $\mathcal{K}^n$ has the coloring c defined in Definition 4.1.4, i.e., for each vertex $v \in \mathcal{K}^n$, $c(v) = (id(v) + b(v)) \bmod (n+1)$. Lemma 4.1.5 states that each n-simplex of $\mathcal{K}^n$ is monochromatic under b if and only if it is properly colored under c. Therefore, the content $\mathcal{C}$ of $\mathcal{K}^n$, Definition 4.1.1, counts the monochromatic n-simplexes of $\mathcal{K}^n$. The characterization is presented in terms of $\mathcal{C}$.

Theorem 5.1.1 *Let $\mathcal{K}^n$ be a ccodi of σ^n with a symmetric binary coloring. Then, for some $k_i \in \mathbb{Z}$*

$$\mathcal{C} = 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i$$

5.2 A Cone Based Proof

In Section 4.1 it is observed that Index Lemma implies that the content of two pseudomanifolds with the same boundary, are the same, Lemma 4.1.3. Therefore, to characterize $\mathcal{C}(\mathcal{K}^n)$, it can be used a particular convenient divided image $\mathcal{L}^n$ such that $bd(\mathcal{K}^n) = bd(\mathcal{L}^n)$. Intuitively, the proof consists of putting a 0-monochromatic n-simplex τ at the center of $bd(\mathcal{K}^n)$ and constructing the cone of $bd(\mathcal{K}^n)$ and τ, to obtain $\mathcal{L}^n$. As proved in Section 4.2, every n-simplex of $\mathcal{L}^n$ contains at least one vertex of τ, hence its monochromatic n-simplexes are 0-monochromatic because τ is 0-monochromatic. This will simplify characterizing $\mathcal{C}(\mathcal{L}^n)$. The key of the proof is that monochromatic n-simplexes of $\mathcal{L}^n$ that contain monochromatic and isomorphic simplexes of $bd(\mathcal{K}^n)$, are counted in the same way by $\mathcal{C}(\mathcal{L}^n)$.

5.2.1 An Example

Figure 5.1 presents an example of the proof. The divided image $\mathcal{L}^2$ is obtained by taking the boundary of $\mathcal{K}^2$ and constructing the cone with a 0-monochromatic 2-simplex, τ. Since $\mathcal{K}^2$ and $\mathcal{L}^2$ have the same boundary, they have the same number of monochromatic 2-simplexes, counting as Lemma 4.1.6 states, namely, each b-monochromatic n-simplex with orientation d is counted as $(-1)^{b*n}d$. The reader can easily verify that $\mathcal{C}(\mathcal{K}^2) = \mathcal{C}(\mathcal{L}^2)$ by giving a coherent orientation to $\mathcal{K}^2$ and $\mathcal{L}^2$ and counting each monochromatic 2-simplex with orientation d as d (if $\mathcal{C}(\mathcal{K}^2) = -\mathcal{C}(\mathcal{L}^2)$ then we get $\mathcal{C}(\mathcal{K}^2) = \mathcal{C}(\mathcal{L}^2)$ by giving the opposite orientation to the simplexes of either $\mathcal{K}^2$ or $\mathcal{L}^2$).

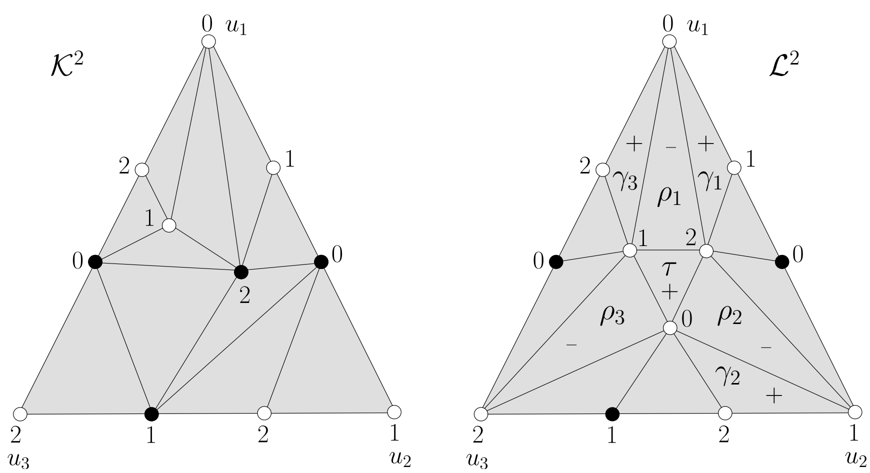

Figure 5.1: An example of the proof.

Now, observe that vertex u_1 together with two vertexes of τ generates a 0-monochromatic 2-simplex ρ_1 in $\mathcal{L}^2$. Similarly, vertexes u_2 and u_3 generate ρ_2 and ρ_3, respectively. So, in total we have three 2-simplexes, ρ_1, ρ_2 and ρ_3, that share a 1-face with τ. Moreover, these three 2-simplexes are 0-monochromatic because the binary coloring is symmetric on the boundary, and also have the same orientation -1, since u_1, u_2 and u_3 are isomorphic. Something similar happens with the isomorphic 0-monochromatic 1-simplexes on the boundary, see for example γ_1, γ_2 and γ_3. Thus, $\mathcal{C}(\mathcal{L}^2) = +1$, because the 2-simplex at the center τ gives $+1$, the 2-simplexes ρ_1, ρ_2 and ρ_3 generated by the vertexes u_1, u_2 and u_3, give $3 \cdot (-1)$, and the 2-simplexes γ_1, γ_2 and γ_3 generated by the 0-monochromatic 1-simplexes on the boundary give $3 \cdot (+1)$.

5.2.2 The Proof

Let φ be a map such that $\mathcal{K}^n$ is a divided image of σ^n under φ. It is not hard to see that $\varphi(bd(\sigma^n))$ is a ccodi of $bd(\sigma^n)$ under $\varphi|bd(\sigma^n)$. Consider a 0-monochromatic and properly colored n-simplex τ^n. By Lemma 4.2.3, the cone $\mathcal{L}^n = \tau^n \circledast \varphi(bd(\sigma^n))$ is a ccodi of σ^n under the extension ψ of $\varphi|bd(\sigma^n)$ such that $\psi(\sigma^n) = \mathcal{L}^n$. Also, notice that $bd(\mathcal{L}^n) = bd(\psi(\sigma^n)) = \psi(bd(\sigma^n)) = \varphi(bd(\sigma^n)) = bd(\mathcal{K}^n)$.

By Lemma 4.2.2, every n-simplex of $\mathcal{L}^n$ has the form $\tau * \gamma$, where τ is a non-empty face of τ^n, $\gamma \in \varphi(bd(\sigma^n))$ and $dim(\gamma) = dim(carr(\gamma))$. In addition, τ is the unique face of τ^n such that $\tau * \gamma$ is an n-simplex of $\mathcal{L}^n$. Recall that simplex $\tau * \gamma$ is the n-simplex generated by γ. Observe that $\tau * \gamma$ has at least one vertex in τ^n. Moreover, $\tau * \gamma$ is 0-monochromatic if and only if γ is 0-monochromatic. Since τ^n is 0-monochromatic and each n-simplex contains at least one vertex of τ^n, $\mathcal{L}^n$ does not contain 1-monochromatic n-simplexes.

Let us add the coloring c, Definition 4.1.4, to $\mathcal{L}^n$ and $\mathcal{K}^n$. By Lemma 4.1.3, $\mathcal{C}(\mathcal{K}^n) = \mathcal{C}(\mathcal{L}^n)$, both with respect to c. By Lemmas 4.1.5 and 4.1.6, the proof of Theorem 5.1.1 consists of computing $\mathcal{C}(\mathcal{L}^n)$ by counting each 0-monochromatic n-simplex with orientation d as d.

Intuitively, the *j-corners* of a divided image are its j-simplexes that have an i-face in the boundary, for every $i \leq j$. These simplexes help in proving Lemma 5.2.2 below, which is key in the proof of Theorem 5.1.1. Figure 5.2 shows a 2-dimensional divided image and its 2-corners marked with a small cross.

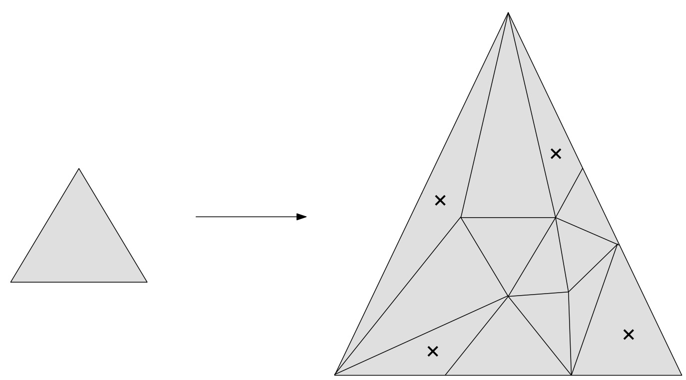

Figure 5.2: The 2-corners of a divided image.

Definition 5.2.1 *Let $\mathcal{K}^n$ be a divided image of σ^n under ψ, and σ^j be a j-face of σ^n. The set of* j-corners *of $\psi(\sigma^j)$ is:*

$$\begin{aligned} j\text{-corners}(\psi(\sigma^j)) = \quad & \{\tau^j \in \psi(\sigma^j) | \forall\, 0 \leq k \leq j, \exists\, \sigma^k, \rho^k, \textit{ such that} \\ & \sigma^k \subseteq \sigma^j, \rho^k \in \psi(\sigma^k) \textit{ and } \rho^0 \subset \rho^1 \subset \ldots \subset \rho^j = \tau^j\} \end{aligned}$$

It follows from Definition 3.1.2 of a divided image and because $\sigma^i \in bd(\sigma^{i+1})$, that j-corners$(\psi(\sigma^j))$ is not empty: for any two faces $\sigma^i \subset \sigma^{i+1}$ of σ^n, for every $\tau^i \in \psi(\sigma^i)$ there exists a simplex $\tau^{i+1} \in \psi(\sigma^{i+1})$ such that $\tau^i \subset \tau^{i+1}$.

Consider again the divided image $\mathcal{L}^n$ of σ^n under ψ. Recall that the binary coloring b of $\mathcal{L}^n$ is symmetric. Let $\mathcal{F}$ be a family of simplicial bijections such that b is symmetric with respect to $\mathcal{F}$. Consider i-faces σ_1 and σ_2 of σ^n, and $f_{\sigma_1\sigma_2} \in \mathcal{F}$. Let γ_1 and γ_2 be isomorphic i-simplexes with respect to $f_{\sigma_1\sigma_2}$, i.e., $f_{\sigma_1\sigma_2}(\gamma_1) = \gamma_2$.

Lemma 5.2.2 *The n-simplexes generated by γ_1 and γ_2, respectively, have the same orientation in a coherent orientation of $\mathcal{L}^n$.*

Proof: Let ρ_1 be an i-corner of $\psi(\sigma_1)$. Since $\psi(\sigma_1)$ is connected, there is an i-path, P_1, from γ_1 to ρ_1 in $\psi(\sigma_1)$. Notice that the i-simplexes of $\psi(\sigma_2)$, which are isomorphic to the i-simplexes of P_1, form an i-path, P_2, from γ_2 to ρ_2, the isomorphic simplex to ρ_1. One can easily prove by induction on i, that f maps i-corners to i-corners. Then ρ_2 is an i-corner of $\psi(\sigma_2)$. We have that the i-simplexes of P_1 generate n-simplexes in $\mathcal{L}^n$. Moreover, consecutive i-simplexes in P_1 (sharing an $(i-1)$-face) generate n-simplexes that share an $(n-1)$-face. In other words, P_1 generates an n-path in $\mathcal{L}^n$. Something similar happens with P_2. By Lemma 3.1.1, we conclude that the n-simplexes generated by γ_1 and γ_2 have the same orientation if and only if the n-simplexes generated by ρ_1 and ρ_2 have the same orientation. Therefore, if the latter is proved then the lemma follows. Moreover, by Lemma 3.1.1, it is enough to prove that, for any i-face of σ^n, for each i-corner λ of its divided image, there is an n-path of length $i+2$ from the n-simplex generated by λ to the n-simplex at the center of the cone, τ^n.

Consider an i-face σ of σ^n. Let ρ be an i-corner of $\psi(\sigma)$. By Definition 5.2.1 of i-corners, for some faces $\sigma^0, \sigma^1, \ldots, \sigma^i = \sigma$, there exist simplexes $\rho^0 \in \psi(\sigma^0), \rho^1 \in \psi(\sigma^1), \ldots, \rho = \rho^i \in \psi(\sigma^i)$ such that $\rho^0 \subset \rho^1 \subset \ldots \subset \rho^i$. We have that simplexes $\rho^0, \rho^1, \ldots, \rho^i$ generate n-simplexes in $\mathcal{L}^n$. Let $\tau_j * \rho^j$ be the n-simplex generated by ρ^j. Notice that $\tau_j * \rho^j$ and $\tau_{j+1} * \rho^{j+1}$ share an $(n-1)$-face. Also $\tau_0 * \rho^0$ and τ^n share an $(n-1)$-face. Therefore, simplexes $\tau^n, \tau_0 * \rho^0, \tau_1 * \rho^1, \ldots, \tau_i * \rho^i$ are a n-path of length $i+2$.

■

Suppose $\mathcal{L}^n$ has a coherent orientation such that τ^n is positively oriented. Then $\mathcal{C}$ is equal to the number of the 0-monochromatic n-simplexes generated by the i-simplexes of $bd(\mathcal{L}^n)$ with carriers of dimension i, plus 1, counting τ^n itself. Consider i-faces σ_1 and σ_2 of σ^n. As explained before, a 0-monochromatic i-simplex of $\psi(\sigma_1)$ or $\psi(\sigma_2)$, generates a 0-monochromatic n-simplex of $\mathcal{L}^n$. Let k_i and ℓ_i be the number of 0-monochromatic n-simplexes counted by $\mathcal{C}$, and generated by the i-simplexes of $\psi(\sigma_1)$ and $\psi(\sigma_2)$, respectively. By Lemma 5.2.2, n-simplexes generated by isomorphic i-simplexes of $\psi(\sigma_1)$ and $\psi(\sigma_2)$, with respect to $f_{\sigma_1\sigma_2}$, have the same orientation, and thus $k_i = \ell_i$ (recall that b is a symmetric binary coloring of $\mathcal{L}^n$). As σ^n has $\binom{n+1}{i+1}$ i-faces, the number of the 0-monochromatic n-simplexes generated by the divided images of all i-faces of σ^n, is $\binom{n+1}{i+1}k_i$. Therefore $\mathcal{C} = 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1}k_i$, for some $k_i \in \mathbb{Z}$.

5.3 An Inductive Based Proof

This section presents an inductive based proof of Theorem 5.1.1. Actually, the section proves a slightly weaker result, contained in Theorem 5.3.3. Roughly, Theorem 5.3.3 assumes that the binary coloring is rank-symmetric, and the divided image of each face of σ^n contains at least one vertex that does not belong to its boundary. These two additional assumptions make the proof easier by reducing the number of cases to consider, and they do not restrict the applicability of the result, because every WSB protocol indeed induces a rank-symmetric binary coloring (see Section 3.5) and the other assumption is satisfied by the model of computation. The formal statement of Theorem 5.3.3 is postponed a bit, first an example of the proof and some extra definitions and lemmas are presented.

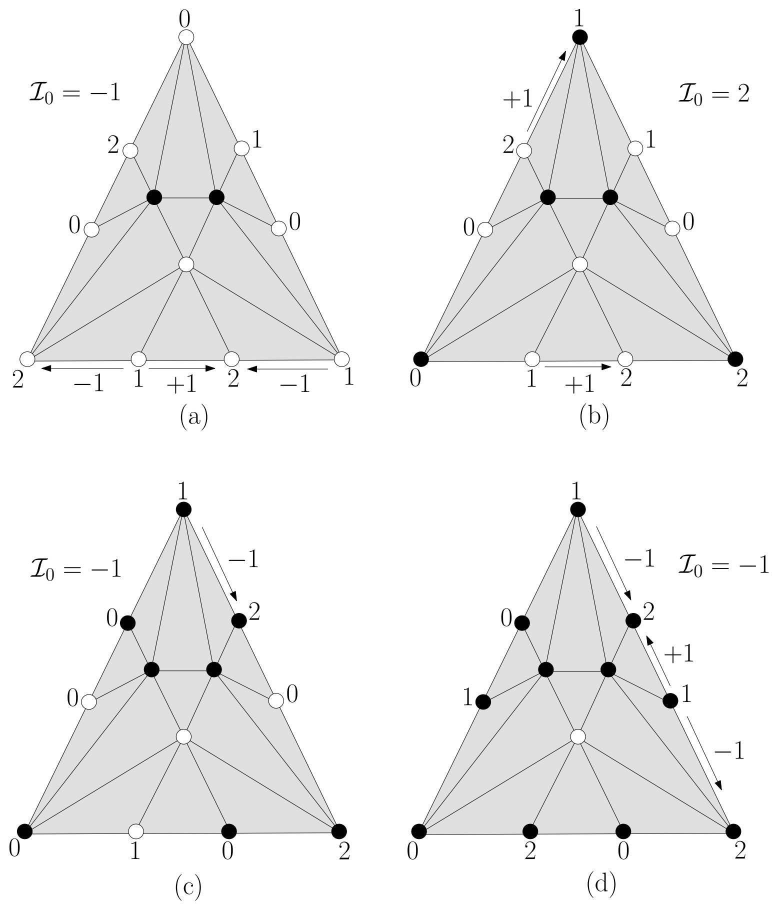

Figure 5.3: An example of the inductive process.

5.3.1 An Example

Recall that the index and content are computed with respect to the coloring c defined as $c(v) = (id(v) + b(v)) \bmod (n+1)$ for each vertex $v \in \mathcal{K}^n$, where id and b are the chromatic and binary colorings of $\mathcal{K}^n$. The strategy is to start with a binary coloring equal to 0 on the boundary, $\forall v \in bd(\mathcal{K}^n), b(v) = 0$, and then *process* groups of isomorphic vertexes (change their binary color to 1) with ℓ-dimensional carriers, until $bd(\mathcal{K}^n)$ gets its original binary coloring. This action is called *ℓ-step* and it may be done more than once in each dimension ℓ. A step guarantees that after executing it, the coloring b of $\mathcal{K}^n$ remains rank-symmetric. Moreover, steps are done by dimension: a vertex with $(\ell+1)$-dimensional carrier is processed if and only if every vertex with ℓ-dimensional carrier has its correct binary color. For example, for dimension 3, first, if necessary, the corners are processed, then the vertexes inside the divided images of the edges, and finally the vertexes inside the divided images of the triangles. The vertexes inside the divided image of the tetrahedron are not modified and actually their coloring does not matter. The main part of the proof is to analyze how all these steps affect the index of $\mathcal{K}^n$. It will be proved that all changes in a step affect the index in the same way. The main difference between this proof and the proof in Section 5.2, is that this one characterizes the way the content varies as the coloring is gradually modified.

Figure 5.3 presents an example of the inductive process. The vertexes have colorings b and c. Assume the 2-simplexes are counterclockwise oriented. For a properly colored 1-simplex on the boundary, the arrow shows the direction followed by c colors 1 and 2, and -1 or $+1$ denotes how this simplex is counted by $\mathcal{I}_0$. The process begins with a binary coloring equal to 0 on the boundary, Figure 5.3 (a). The index at the beginning of the process always is equal to ± 1, according to the orientation. The process has a 0-step, Figure 5.3 (b), that adds a multiple of three to the index because a 2-simplex has three 0-faces. Figure 5.3 (c) shows a 1-step which adds a multiple of three to the index because a 2-simplex has three 1-faces. The process ends with the 1-step in Figure 5.3 (d).

5.3.2 Additional Definitions and Lemmas

Let $\mathcal{K}^n$ be a divided image of σ^n under ψ. A *cross edge* of $\mathcal{K}^n$ is a 1-simplex $\{u, v\} \in bd(\mathcal{K}^n)$ such that there exist distinct i-faces σ, σ' of σ^n, $0 \leq i \leq n-2$, such that $u \in \psi(\sigma)$, $u \notin \psi(\sigma')$, $v \notin \psi(\sigma)$ and $v \in \psi(\sigma')$. This implies that if $\mathcal{K}^n$ has no cross edges then for every $\sigma \subset \sigma^n$, there exists $v \in \psi(\sigma)$ such that $carr(v) = \sigma$. Figure 5.4 depicts two divided images of a 2-simplex and a 3-simplex, respectively, in which the bold edges are examples of cross edges. It is not hard to see that the IS complex of a protocol in which all processes are active in at least one round, has no cross edges.

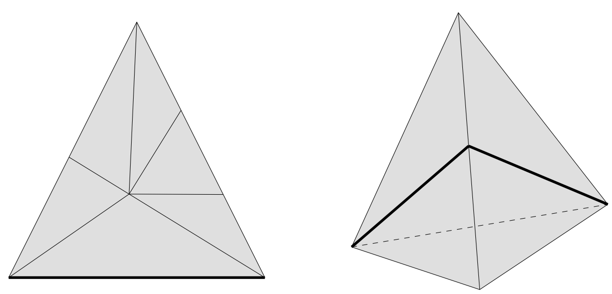

Figure 5.4: Divided images with cross edges.

The following two lemmas concerning the orientability of isomorphic simplexes on the boundary of a divided image, will be useful in the proof of Theorem 5.3.3. For clarity of presentation, these lemmas are proved in Section 5.3.4. For the rest of the chapter, for a properly colored simplex σ^n with ID^n, let σ_i^{n-1} denote the $(n-1)$-face of σ^n without color $i \in ID^n$.

Lemma 5.3.1 *Let $\mathcal{K}^n$ be a ccodi of σ^n under ψ. In any coherent orientation of $\mathcal{K}^n$, $\psi(\sigma_i^{n-1})$ has a coherent induced orientation.*

Lemma 5.3.2 *Let $\mathcal{K}^n$ be a ccodi of σ^n with structural symmetry with respect to to $\mathcal{F}$. In any coherent orientation of $\mathcal{K}^n$, the n-simplexes of $\mathcal{K}^n$ that contain isomorphic $(n-1)$-simplexes, with respect to $\mathcal{F}$, of $bd(\mathcal{K}^n)$, have the same orientation.*

5.3.3 The Inductive Process

This section presents the proof of Theorem 5.3.3 via the inductive process described in Section 5.3.1.

Theorem 5.3.3 *Let $\mathcal{K}^n$ be a ccodi of σ^n with a rank-symmetric binary coloring and no cross edges. Then, for some $k_i \in \mathbb{Z}$*

$$\mathcal{C} = 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i$$

The index and content of $\mathcal{K}^n$ are computed with respect to the coloring c defined in Definition 4.1.4. The following lemma computes the value of the index at the beginning of the process.

Lemma 5.3.4 *If for every $v \in bd(\mathcal{K}^n)$, $b(v) = 0$, then $\mathcal{I}_i = \pm 1$, according to the orientation of $\mathcal{K}^n$.*

Proof: Consider the faces $\sigma^0, \sigma^1, \ldots, \sigma^{n-1}$ of σ^n such that $id(\sigma^i) = ID^i$. Let $\mathcal{K}^i$ denote $\psi(\sigma^i)$. It is clear that $\mathcal{K}^i$ is a ccodi of σ^i under $\psi|\sigma^i$. Assume $\mathcal{K}^i$ has the the induced orientation by $\mathcal{K}^{i+1}$. By Lemma 5.3.1, $\mathcal{K}^i$ has a coherent induced orientation. By the definition of c, Definition 4.1.4, for every $v \in bd(\mathcal{K}^n)$, $c(v) = id(v)$. Notice that $\mathcal{K}^{n-1}$ contains all the properly colored $(n-1)$-simplexes of $bd(\mathcal{K}^n)$ with ID^{n-1}. Actually, every $(n-1)$-simplex of $\mathcal{K}^{n-1}$ is properly colored with ID^{n-1}. Therefore, we can recursively use Index Lemma 4.1.2. That is, $\mathcal{I}_n(\mathcal{K}^n) = \mathcal{C}(\mathcal{K}^{n-1})$ and, by Index Lemma, $\mathcal{I}_n(\mathcal{K}^n) = (-1)^{n-1}\mathcal{I}_{n-1}(\mathcal{K}^{n-1})$. We can do the same with $\mathcal{K}^{n-1}$ and $\mathcal{K}^{n-2}$, i.e., $\mathcal{I}_{n-1}(\mathcal{K}^{n-1}) = (-1)^{n-2}\mathcal{I}_{n-2}(\mathcal{K}^{n-2})$, and so on. Thus, $\mathcal{I}_n(\mathcal{K}^n) = (-1)^{1+2+\ldots+n-1}\mathcal{I}_1(\mathcal{K}^1)$. Observe that $\mathcal{I}_1(\mathcal{K}^1) = \pm 1$, according to the orientation. And by Index Lemma, $\mathcal{I}_i(\mathcal{K}^n) = \pm 1$.

■

In what follows, the binary color of a vertex on the boundary is modified. Consider $v \in bd(\mathcal{K}^n)$ such that $b(v) = 0$. The vertex v is *processed* when its binary color is changed from 0 to 1. Colorings, simplexes and values after processing v, are marked with a dot ($\dot{}$). Thus, $\mathcal{I}_i$ and $\dot{\mathcal{I}}_i$ denote the index of $\mathcal{K}^n$ before and after processing v, and $c(v)$ and $\dot{c}(v)$ are its coloring c before and after processing it.

Let k denote $\binom{n+1}{\ell+1}$. Let $\sigma_1, \sigma_2 \ldots \sigma_k$ be the ℓ-faces of σ^n. For $1 \leq j \leq k$, consider the id-rank-preserving bijection $f_j : \psi(\sigma_1) \to \psi(\sigma_j)$. Let us assume that theres exists $v \in \psi(\sigma_1)$ such that $carr(v) = \sigma_1$ and $b(v) = 0$. For $1 \leq j \leq k$, let v_j denote $f_j(v)$, the isomorphic vertex of v in $\psi(\sigma_j)$. Thus $v = v_1$. Also $b(v_j) = 0$ and $carr(v_j) = \sigma_j$. An *ℓ-step* consists of processing one by one the vertexes $v_1, v_2 \ldots v_k$. The vertexes of $bd(\mathcal{K}^n)$ are processed by dimension, i.e., the process applies an ℓ-step if and only if all necessary $(\ell-1)$-steps have been done. Therefore, when an ℓ-step is the next step in the process, each vertex with carrier of dimension smaller than ℓ, has its correct binary color, and each vertex with carrier of dimension greater than ℓ, has binary color 0. Colorings, simplexes and values after a step are denoted with a circumflex ($\hat{}$). For the rest of the proof, fix the ℓ-step associated with the vertexes in the set $\mathcal{V} = \{v_1 \ldots v_k\}$, and assume that none of the vertexes of $\mathcal{V}$ has been processed. The assumption that the binary coloring b of $\mathcal{K}^n$ is rank-symmetric helps in proving that b remains rank-symmetric after an ℓ-step. Also, it is clear that b is rank-symmetric at the beginning of the process. However, b is not symmetric in the middle of a step.

The core of the proof is computing how the index of $\mathcal{K}^n$ changes when a vertex of $\mathcal{V}$ is processed. For doing that, the definition of content is extended for colored pseudomanifolds with an arbitrary number of colors. For a colored and oriented pseudomanifold $\mathcal{L}^n$ (it can be colored with more that $n+1$ colors) and a set of $n+1$ colors $\mathcal{H}$, $\mathcal{C}(\mathcal{L}^n, \mathcal{H})$ denotes the number of properly colored n-simplexes in $\mathcal{L}^n$ with $\mathcal{H}$, counted by orientation. We say that $\mathcal{C}(\mathcal{L}^n, \mathcal{H})$ is the *content of* $\mathcal{L}^n$ *with* $\mathcal{H}$. Recall that, for a vertex $v \in \mathcal{L}^n$, $st(v, \mathcal{L}^n)$ is the complex consisting of those simplexes of $\mathcal{L}^n$ that contain the vertex v. For $st(v, \mathcal{L}^n)$, we write $\mathcal{C}(v, \mathcal{L}^n, \mathcal{H})$ instead of $\mathcal{C}(st(v, \mathcal{L}^n), \mathcal{H})$. Figure 5.5 presents a colored pseudomanifold $\mathcal{L}^2$ in which $st(u, \mathcal{L}^2)$ and $st(v, \mathcal{L}^2)$ are the regions bounded by bold lines. The reader can check that $\mathcal{C}(u, \mathcal{L}^2, \{3, 4, 5\}) = -1$, $\mathcal{C}(v, \mathcal{L}^2, \{1, 2, 3\}) = 1$ and $\mathcal{C}(u, v, \mathcal{L}^2, \{0, 2, 3\}) = -1$, assuming each 2-simplex is counterclockwise oriented.

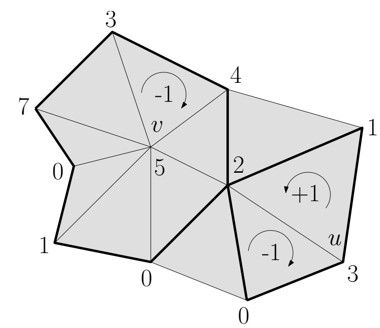

Figure 5.5: The extended definition of content.

Lemma 5.3.5 below describes how the index $\mathcal{I}_i$ changes when a vertex in $bd(\mathcal{K}^n)$ is processed. For the rest of the section, assume $bd(\mathcal{K}^n)$ has the induced orientation by $\mathcal{K}^n$. Observe that $\mathcal{C}(v, bd(\mathcal{K}^n), ID_i^n)$ is the number of $(n-1)$-simplexes in $st(v, bd(\mathcal{K}^n))$ that are properly colored under c with ID_i^n.

Lemma 5.3.5 *Consider a vertex $v \in bd(\mathcal{K}^n)$ such that $b(v) = 0$. If v is processed then $\dot{\mathcal{I}}_i = \mathcal{I}_i + \dot{\mathcal{C}}(v, bd(\mathcal{K}^n), ID_i^n) - \mathcal{C}(v, bd(\mathcal{K}^n), ID_i^n)$.*

Proof: First, observe $c(v) \neq \dot{c}(v)$. Consider an $(n-1)$-simplex $\tau \in st(v, bd(\mathcal{K}^n))$. We have two cases. If $c(\tau) \neq ID_i^n$ then it is possible $\dot{c}(\tau) = ID_i^n$. In the other case, if $c(\tau) = ID_i^n$ then $\dot{c}(\tau) \neq ID_i^n$. Thus, $\dot{\mathcal{I}}_i$ is $\mathcal{I}_i$ plus all those $(n-1)$-simplexes of $bd(\mathcal{K}^n)$ that will be properly colored with ID_i^n after v is processed, minus all those properly colored $(n-1)$-simplexes of $bd(\mathcal{K}^n)$ with ID_i^n before v is processed. Also, notice that $st(v, bd(\mathcal{K}^n))$ contains all those $(n-1)$-simplexes that change their coloring c when v is processed. Therefore, $\dot{\mathcal{I}}_i = \mathcal{I}_i + \dot{\mathcal{C}}(v, bd(\mathcal{K}^n), ID_i^n) - \mathcal{C}(v, bd(\mathcal{K}^n), ID_i^n)$.

■

The following two lemmas and corollary intuitively say that when a vertex of $\mathcal{V}$ is processed, $\dot{\mathcal{I}}_i$ can be computed by counting in a specific "region" of $bd(\mathcal{K}^n)$.

Lemma 5.3.6 *Consider an* $(n-1)$*-simplex* $\tau \in st(v, bd(\mathcal{K}^n))$. *If* $c(\tau) = ID_i^n$ *then* $\tau \in st(v, \psi(\sigma_i^{n-1}))$.

Proof: First, by Definition 3.1.2 of a divided image and because $\sigma_i^{n-1} \in bd(\sigma^n)$, we have that $\psi(\sigma_i^{n-1}) \subset bd(\mathcal{K}^n)$. We have two cases. If $\ell = n-1$, $\mathcal{V}$ is an $n-1$ step, then v has a carrier of dimension $n-1$ and hence $v \notin bd(\psi(\sigma_i^{n-1}))$ (notice if $v \in bd(\psi(\sigma_i^{n-1}))$ then it cannot have a carrier of dimension $n-1$). Thus $st(v, bd(\mathcal{K}^n)) = st(v, \psi(\sigma_i^{n-1}))$.

The second case is $\ell < n-1$. Let σ be the carrier of v. Consider a face σ_j^{n-1} of σ^n such that $\sigma \subset \sigma_j^{n-1}$ and $\sigma_i^{n-1} \neq \sigma_j^{n-1}$. We have $v \in \psi(\sigma_j^{n-1})$ and $i \in id(\sigma_j^{n-1})$. Consider an ℓ-simplex $\rho \in st(v, \psi(\sigma))$. Let γ be a simplex of $st(v, \psi(\sigma_j^{n-1}))$ such that $\rho \subset \gamma$. Let u be the vertex of γ such that $id(u) = i$. Observe that $u \notin \rho$ and hence $u \notin \psi(\sigma)$. Since $\mathcal{K}^n$ does not have cross edges, u has a carrier of dimension greater than ℓ. Thus $b(w) = 0$, hence $c(w) = i$ and $c(\gamma) \neq ID_i^n$. This implies $st(v, \psi(\sigma_i^{n-1}))$ contains every properly colored $(n-1)$-simplexes of $st(v, bd(\mathcal{K}^n))$ with ID_i^n.

■

Corollary 5.3.7 *Let* v *be a vertex of* $\mathcal{V}$ *such that* $v \in \psi(\sigma_i^{n-1})$. *Then* $\mathcal{C}(v, bd(\mathcal{K}^n), ID_i^n) = \mathcal{C}(v, \psi(\sigma_i^{n-1}), ID_i^n)$.

Lemma 5.3.8 *Let* v *be a vertex of* $\mathcal{V}$ *such that* $v \in \psi(\sigma_i^{n-1})$. *If* v *is processed then* $\dot{\mathcal{I}}_i = \mathcal{I}_i - \mathcal{C}(v, \psi(\sigma_i^{n-1}), ID_i^n)$.

Proof: By Lemma 5.3.5, if v is processed, $\dot{\mathcal{I}}_i = \mathcal{I}_i + \dot{\mathcal{C}}(v, bd(\mathcal{K}^n), ID_i^n) - \mathcal{C}(v, bd(\mathcal{K}^n), ID_i^n)$. And by Lemma 5.3.7, $\mathcal{C}(v, bd(\mathcal{K}^n), ID_i^n) = \mathcal{C}(v, \psi(\sigma_i^{n-1}), ID_i^n)$. Consider an $(n-1)$-simplex $\tau \in st(v_i, \psi(\sigma_i^{n-1}))$. Recall that $id(\tau) = ID_i^n$. By the definition of coloring c, Definition 4.1.4, one can conclude $c(\tau) = ID_i^n$ if and only if τ is 0-monochromatic. Also observe $\dot{\tau}$ is not 0-monochromatic and hence $\dot{c}(\tau) \neq ID_i^n$. Thus, $\dot{\mathcal{C}}(v_i, bd(\mathcal{K}^n), ID_i^n) = 0$ and so $\dot{\mathcal{I}}_i = \mathcal{I}_i - \mathcal{C}(v, bd(\mathcal{K}^n), ID_i^n)$. By Corollary 5.3.7, the lemma follows.

■

Lemma 5.3.9 shows that the content of vertexes $u, v \in \mathcal{V}$ in $\psi(\sigma_i^{n-1})$, are essentially the same, assuming none of them have been processed. This property will imply that all the modifications in a step affect the index in the same way.

Lemma 5.3.9 *Consider vertexes* $u, v \in \mathcal{V}$ *that belong to* $\psi(\sigma_i^{n-1})$ *and* $\psi(\sigma_j^{n-1})$*, respectively. Then* $(-1)^i \mathcal{C}(u, \psi(\sigma_i^{n-1}), ID_i^n) = (-1)^j \mathcal{C}(v, \psi(\sigma_j^{n-1}), ID_j^n)$.

Proof: Consider the faces $\sigma^{\ell+1}, \ldots, \sigma^{n-1}, \sigma^n$ such that $id(\sigma^m) = ID^m$, $\ell < m \leq n$. For $m < n$, assume $\psi(\sigma^m)$ has the induced orientation by $\psi(\sigma^{m+1})$. By Lemma 5.3.1, $\psi(\sigma^m)$ has a coherent induced orientation. It is clear that $\psi(\sigma^m)$ is a ccodi of σ^m under $\psi|\sigma^m$ with a rank symmetric binary coloring and no cross edges. By induction on m, we prove that the lemma holds for the vertexes of $\mathcal{V}$ that belong to $\psi(\sigma^m)$.

For the base of the induction, $\ell + 1$, consider faces σ_i^ℓ and σ_j^ℓ of $\sigma^{\ell+1}$, without id color i and j. Let $\mathcal{L}_i^\ell$ and $\mathcal{L}_j^\ell$ denote $\psi(\sigma_i^\ell)$ and $\psi(\sigma_j^\ell)$. Notice that $\mathcal{L}_i^\ell$ and $\mathcal{L}_j^\ell$ only contain one vertex of $\mathcal{V}$, respectively. Consider $u, v \in \mathcal{V}$ such that $u \in \mathcal{L}_i^\ell$ and $v \in \mathcal{L}_j^\ell$. By the definition of coloring c, Definition 4.1.4, one can conclude that, for $\tau \in st(u, \mathcal{L}_i^\ell)$, $c(\tau) = id(\tau)$ if and only if τ is 0-monochromatic. Consider an ℓ-simplex $\tau \in st(u, \mathcal{L}_i^\ell)$. Recall that $id(\tau) = ID_i^{\ell+1}$. Suppose τ is

0-monochromatic. Thus, for each $w \in \tau$, $c(w) = id(w)$. Notice that if τ has induced orientation d then $\mathcal{C}(u, \mathcal{L}_i^\ell, ID_i^{\ell+1})$ counts τ as d. Something similar happens with ℓ-simplexes in $st(v, \mathcal{L}_j^\ell)$. Consider the isomorphic ℓ-simplex ρ of τ in $st(v, \mathcal{L}_j^\ell)$. Notice that ρ is 0-monochromatic. By Lemma 5.3.2, τ and ρ have the same orientation, multiplied by $(-1)^i$ and $(-1)^j$, respectively. Therefore, $(-1)^i\mathcal{C}(u, \mathcal{L}_i^\ell, ID_i^{\ell+1}) = (-1)^j\mathcal{C}(v, \mathcal{L}_j^\ell, ID_j^{\ell+1})$.

Suppose the lemma is true for $m-1$. We prove it is true for m. Consider the faces σ^{m-1} and σ_k^{m-1} of σ^m. Thus, $id(\sigma^{m-1}) = ID^{m-1}$ and $id(\sigma_k^{m-1}) = ID_k^m$. Let $\mathcal{L}^{m-1}$ and $\mathcal{L}_k^{m-1}$ denote $\psi(\sigma^{m-1})$ and $\psi(\sigma_k^{m-1})$. We have that $\mathcal{L}^{m-1}$ and $\mathcal{L}_k^{m-1}$ contain more than one vertex of $\mathcal{V}$. Consider $u, v \in \mathcal{V}$ such that $u \in \mathcal{L}^{m-1}$, $v \in \mathcal{L}_k^{m-1}$ and they are isomorphic. As in the base of the induction, it can be easily proved that $(-1)^m\mathcal{C}(u, \mathcal{L}^{m-1}, ID^{m-1}) = (-1)^k\mathcal{C}(v, \mathcal{L}_k^{m-1}, ID_k^m)$. Consider a vertex $w \in \mathcal{V}$ such that $w \neq u$ and $w \in \mathcal{L}^{m-1}$. Observe that if we prove $\mathcal{C}(u, \mathcal{L}^{m-1}, ID^{m-1}) = \mathcal{C}(w, \mathcal{L}^{m-1}, ID^{m-1})$, the lemma follows.

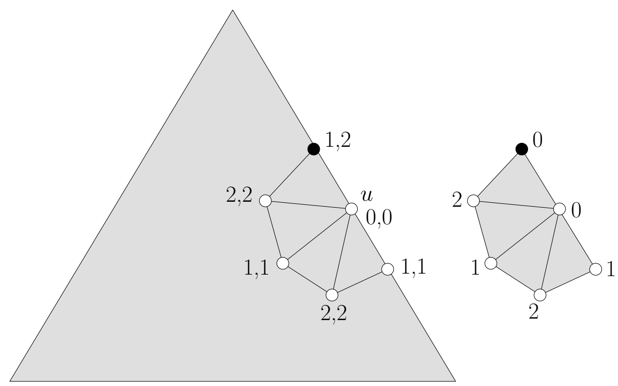

Figure 5.6: Example of coloring c'.

Let σ_i^{m-2} and σ_j^{m-2} be faces of σ^{m-1} such that $u \in \psi(\sigma_i^{m-2})$ and $w \in \psi(\sigma_j^{m-2})$. Assume, w.l.o.g., faces σ_i^{m-2} and σ_j^{m-2} do not have colors i and j. Let $\mathcal{L}_i^{m-2}$ and $\mathcal{L}_j^{m-2}$ denote $\psi(\sigma_i^{m-2})$ and $\psi(\sigma_j^{m-2})$. The idea is to prove that $\mathcal{C}(u, \mathcal{L}^{m-1}, ID^{m-1}) = (-i)^i\mathcal{C}(u, \mathcal{L}_i^{m-2}, ID_i^{m-1})$ and $\mathcal{C}(w, \mathcal{L}^{m-1}, ID^{m-1}) = (-i)^j\mathcal{C}(w, \mathcal{L}_j^{m-2}, ID_j^{m-1})$, by using Index Lemma on complexes $st(u, \mathcal{L}^{m-1})$ and $st(w, \mathcal{L}^{m-1})$. Then, $\mathcal{C}(u, \mathcal{L}^{m-1}, ID^{m-1}) = \mathcal{C}(w, \mathcal{L}^{m-1}, ID^{m-1})$, by induction hypothesis. However, it is possible c colors $st(u, \mathcal{L}^{m-1})$ and $st(w, \mathcal{L}^{m-1})$ with more than m colors, and thus Index Lemma cannot be used on them. So it is defined an extra coloring c' for these two complexes that uses m colors.

Consider $st(u, \mathcal{L}^{m-1})$. The coloring c' is defined as follows. For each vertex $x \in st(u, \mathcal{L}^{m-1})$, if $b(x) = 0$ then $c'(x) = c(x)$, otherwise $c'(x) = c(u)$. Figure 5.6 contains an example of coloring c' where the vertexes inside the triangle have colorings id, b and c, and the stars complex of u, at the right, has the coloring c'. Since $b(u) = 0$ and for every $x \in st(u, \mathcal{L}^{m-1})$, $id(x) \in ID^{m-1}$, we have that c' uses colors ID^{m-1}. Therefore, Index Lemma can be applied on $st(u, \mathcal{L}^{m-1})$. Also, as noticed above, for each $\tau \in st(u, \mathcal{L}^{m-1})$, $c(\tau) = id(\tau)$ if and only if τ is 0-monochromatic, and thus $c'(\tau) = id(\tau)$ if and only if $c(\tau) = id(\tau)$. Thus, $\mathcal{C}(st(u, \mathcal{L}^{m-1}))$ and $\mathcal{C}(st(u, \mathcal{L}_i^{m-2}))$ with respect to c', are equal to $\mathcal{C}(u, \mathcal{L}^{m-1}, ID^{m-1})$ and $\mathcal{C}(u, \mathcal{L}_i^{m-2}, ID_i^{m-1})$ with respect to c, respectively. Now, for an $(m-2)$-simplex $\tau \in bd(st(u, \mathcal{L}^{m-1}))$, if $c'(\tau) = ID_i^{m-1}$ then $\tau \in st(u, \mathcal{L}_i^{m-2})$. In other words, $\mathcal{L}_i^{m-2}$ is the only "region" of $bd(st(u, \mathcal{L}^{m-1}))$ containing properly colored simplexes with ID_i^{m-1}. First observe that $\sigma_i^{m-2} \in bd(\sigma^{m-1})$ and hence $\mathcal{L}_i^{m-2} \subset \mathcal{L}^{m-1}$. Also if $\tau \notin \mathcal{L}_i^{m-2}$ then there must exists $x \in \tau$ such that $id(x) = i$. Since $x \in st(u, \mathcal{L}^{m-1})$, there exists a 1-simplex connecting x and u. Because there are no cross edges, x has a carrier of dimension greater than ℓ and hence $b(x) = 0$. Therefore $c'(x) = i$ and so $c'(\tau) \neq ID_i^{m-1}$.

By Index Lemma 4.1.2, $\mathcal{C}(st(u, \mathcal{L}^{m-1})) = (-i)^i \mathcal{I}_i(st(u, \mathcal{L}^{m-1}))$. Also, we get $\mathcal{I}_i(st(u, \mathcal{L}^{m-1})) = \mathcal{C}(st(u, \mathcal{L}_i^{m-2}))$ because for every $\tau^{m-2} \in bd(st(u, \mathcal{L}^{m-1}))$ such that $c'(\tau^{m-2}) = ID_i^{m-1}$, we have that $\tau^{m-2} \in \mathcal{L}_i^{m-2}$. Therefore, $\mathcal{C}(st(u, \mathcal{L}^{m-1})) = (-i)^i \mathcal{C}(st(u, \mathcal{L}_i^{m-2}))$. Similarly, by adding the appropriate c' coloring to $st(w, \mathcal{L}^{m-1})$, we get $\mathcal{C}(st(w, \mathcal{L}^{m-1})) = (-i)^j \mathcal{C}(st(w, \mathcal{L}_j^{m-2}))$. Finally, by induction hypothesis, $(-i)^i \mathcal{C}(st(u, \mathcal{L}_i^{m-2})) = (-i)^j \mathcal{C}(st(w, \mathcal{L}_j^{m-2}))$ and thus $\mathcal{C}(st(u, \mathcal{L}^{m-1})) = \mathcal{C}(st(w, \mathcal{L}^{m-1}))$.

■

Lemma 5.3.10 shows how the ℓ-step associated to $\mathcal{V}$ affects the index. Recall that a dot ($\dot{}$) denotes a value after a vertex is processed and a circumflex ($\widehat{}$) denotes a value after a step is done.

Lemma 5.3.10 *After the ℓ-step associated to $\mathcal{V}$ is done, we have that $\widehat{\mathcal{I}}_i = \mathcal{I}_i + \binom{n+1}{i+1}k$, for some $k \in \mathbb{Z}$.*

Proof: Consider vertexes $v_i, v_j \in \mathcal{V}$ such that $v_i \in \psi(\sigma_i^{n-1})$ and $v_j \in \psi(\sigma_j^{n-1})$. By Index Lemma 4.1.2, $(-1)^{i-j}\mathcal{I}_i = \mathcal{I}_j$ and $(-1)^{i-j}\dot{\mathcal{I}}_i = \dot{\mathcal{I}}_j$. And by Lemma 5.3.8, $\dot{\mathcal{I}}_j = \mathcal{I}_j - \mathcal{C}(v_j, \psi(\sigma_j^{n-1}), ID_j^n)$, when v_j is processed. Combining these three equations, we get $\dot{\mathcal{I}}_i = \mathcal{I}_i - (-1)^{j-i}\mathcal{C}(v_j, \psi(\sigma_j^{n-1}), ID_j^n)$. Using Lemma 5.3.9, we get $\mathcal{C}(v_i, \psi(\sigma_i^{n-1}), ID_i^n) = (-1)^{j-i}\mathcal{C}(v_j, \psi(\sigma_j^{n-1}), ID_j^n)$, and hence $\dot{\mathcal{I}}_i = \mathcal{I}_i - \mathcal{C}(v_i, \psi(\sigma_i^{n-1}), ID_i^n)$. In other words, when v_j is processed, the index $\mathcal{I}_i$ changes as if v_i is processed. Since $\mathcal{K}^n$ does not have cross edges, there is not a 1-simplex connecting v_i and v_j, and so $v_j \notin st(v_i, bd(\mathcal{K}^n))$. Therefore, the c coloring of the $(n-1)$-simplexes in $st(v_i, \psi(\sigma_i^{n-1}))$ do not change when v_j is processed, hence $\mathcal{C}(v_i, \psi(\sigma_i^{n-1}), ID_i^n) = \dot{\mathcal{C}}(v_i, \psi(\sigma_i^{n-1}), ID_i^n)$. Thus, we have $\widehat{\mathcal{I}}_i = \mathcal{I}_i - \mathcal{C}(v_i, \psi(\sigma_i^{n-1}), ID_i^n)\binom{n+1}{i+1}$ at the end of the step, because $\mathcal{V}$ contains $\binom{n+1}{i+1}$ vertexes.

■

Lemma 5.3.10 directly implies the following lemma.

Lemma 5.3.11 *Let $\mathcal{I}_i$ and $\widehat{\mathcal{I}}_i$ be the indexes of $\mathcal{K}^n$ before and after all the ℓ-steps in the process are done. Then $\widehat{\mathcal{I}}_i = \mathcal{I}_i - \binom{n+1}{i+1}k$, for some $k \in \mathbb{Z}$.*

By Lemma 5.3.4, $\mathcal{I}_i = \pm 1$ at the beginning of the process, according to the orientation. And by Lemma 5.3.11, after all ℓ-steps in the process, $\widehat{\mathcal{I}}_i = \mathcal{I}_i - \binom{n+1}{i+1}k$, Therefore, at the end of the process, $\mathcal{I}_i = 1 + \sum_{i=0}^{n-1}\binom{n+1}{i+1}k_\ell$, for some $k_\ell \in \mathbb{Z}$. By Index Lemma 4.1.2, Theorem 5.3.3 follows.

5.3.4 Concluding the Proof

This section completes the proof of Theorem 5.3.3 by proving Lemmas 5.3.1 and 5.3.2. First, some lemmas concerning the orientability of the simplexes of a divided image are presented. In what follows, for $\{i_1, \ldots, i_k\} \subset ID^n$, let $ID^n_{\{i_1,\ldots,i_k\}}$ denote the set $ID^n - \{i_1, \ldots, i_k\}$.

Let $\mathcal{K}^n$ be a chromatic and coherently oriented divided image of σ^n under ψ. Suppose $\psi(\sigma_i^{n-1})$ contains two distinct simplexes τ^{n-1} and ρ^{n-1} such that they share an face γ^{n-2}. By condition 5 in Definition 3.1.2 of divided image, and since $\sigma_i^{n-1} \in bd(\sigma^n)$, there exist distinct simplexes τ_1^n, $\rho_1^n \in \mathcal{K}^n$ such that $\tau^{n-1} \subset \tau_1^n$ and $\rho^{n-1} \subset \rho_1^n$.

Lemma 5.3.12 *Simplexes τ_1^n and ρ_1^n have opposite orientations.*

Proof: By condition 5 in Definition 3.1.2 of divided image, there exist simplexes $\tau_1^n = \lambda_1^n, \lambda_2^n \ldots \lambda_{t-1}^n$, $\lambda_t^n = \rho_1^n$ of $\mathcal{K}^n$ such that all of them contain the simplex γ^{n-2}, and λ_k^n and λ_{k+1}^n share a face ϱ_k^{n-1}, where $\gamma^{n-2} \subset \varrho_k^{n-1}$ (see Figure 5.7 for an example). In other words, these simplexes form an n-path in which adjacent simplexes share an $(n-1)$-face containing γ^{n-2}. Observe that if this path does not exist then $bd(\psi(\sigma^n)) \neq \psi(bd(\sigma^n))$. Since $\mathcal{K}^n$ is a chromatic, then $id(\rho^{n-1}) = id(\tau^{n-1}) = ID_i^n$ and $id(\gamma^{n-2}) = ID_{\{i,j\}}^n$ for some $j \in ID_i^n$. Also, notice that either $id(\varrho_k^{n-1}) = ID_i^n$ or $id(\varrho_k^{n-1}) = ID_j^n$ (recall $\gamma^{n-2} \subset \varrho_k^{n-1}$). We have that $id(\varrho_1^{n-1}) = ID_j^n$ because $\tau^{n-1} \subset \tau_1^n = \lambda_1^n$ and $id(\tau^{n-1}) = ID_i^n$. And $id(\varrho_2^{n-1}) = ID_i^n$ because $\varrho_1^{n-1} \subset \lambda_2^n$ and $id(\varrho_1^{n-1}) = ID_j^n$, and so on. Therefore, if k is even then $id(\varrho_k^{n-1}) = ID_i^n$, and if it is odd then $id(\varrho_k^{n-1}) = ID_j^n$. We can conclude that t must be even and hence the path contains an even number of n-simplexes. By Lemma 3.1.1, τ_1^n and ρ_1^n have opposite orientations.

■

Consider now faces σ_i^{n-1} and σ_j^{n-1} of σ^n with $i \neq j$. By conditions 4 and 5 in Definition 3.1.2 of divided image, there exist simplexes $\tau^{n-1} \in \psi(\sigma_i^{n-1})$ and $\rho^{n-1} \in \psi(\sigma_j^{n-1})$ such that they share a face γ^{n-2}, where $\gamma^{n-2} \in \psi(\sigma_i^{n-1} \cap \sigma_j^{n-1})$. Moreover, there exist the simplexes τ_2^n, $\rho_2^n \in \mathcal{K}^n$ such that $\tau^{n-1} \subset \tau_2^n$ and $\rho^{n-1} \subset \rho_2^n$. Observe that ρ_2^n and τ_2^n can be the same simplex.

Lemma 5.3.13 *Simplexes τ_2^n and ρ_2^n have the same orientation.*

Proof: As in the proof of Lemma 5.3.12, there exist simplexes $\tau_2^n = \lambda_1^n, \lambda_2^n \ldots \lambda_{t-1}^n, \lambda_t^n = \rho_2^n$ of $\mathcal{K}^n$ such that all of them contain the simplex γ^{n-2}, and λ_k^n and λ_{k+1}^n share a face ϱ_k^{n-1}, where $\gamma^{n-2} \subset \varrho_k^{n-1}$ (see Figure 5.7 for an example). Because $\mathcal{K}^n$ is chromatic, $id(\tau^{n-1}) = ID_i^n$, $id(\rho^{n-1}) = ID_j^n$ and $id(\gamma^{n-2}) = ID_{\{i,j\}}^n$. Following a reasoning that the one used in the proof of Lemma 5.3.12, we can see that if k is even then $id(\varrho_k^{n-1}) = ID_i^n$, and if it is odd then $id(\varrho_k^{n-1}) = ID_j^n$, thus we conclude that the path contains an odd number of n-simplexes. By Lemma 3.1.1, τ_2^n and ρ_2^n have the same orientation.

■

Lemma 5.3.1 (Restated) *Let $\mathcal{K}^n$ be a ccodi of σ^n under ψ. In any coherent orientation of $\mathcal{K}^n$, $\psi(\sigma_i^{n-1})$ has a coherent induced orientation.*

Proof: By definition of orientability, the orientation of an $(n-1)$-simplex of $\psi(\sigma_i^{n-1})$ is the orientation of the unique n-simplex of $\mathcal{K}^n$ that contains it, multiplied by $(-1)^i$. Consider distinct simplexes $\tau^{n-1}, \rho^{n-1} \in \psi(\sigma_i^{n-1})$ such that they share a face γ^{n-2}. Let τ^n and ρ^n be the simplexes of $\mathcal{K}^n$ such that $\tau^{n-1} \subset \tau^n$ and $\rho^{n-1} \subset \rho^n$. By Lemma 5.3.12, τ^n and ρ^n has opposite orientations and hence τ^{n-1} and ρ^{n-1} has opposite induced orientations. By Lemma 3.1.1, $\psi(\sigma_i^{n-1})$ has a coherent orientation.

■

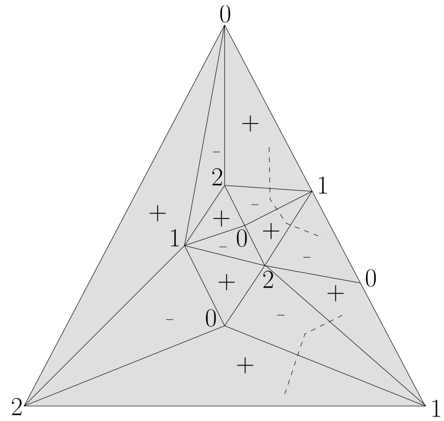

Figure 5.7: Two paths of simplexes.

Lemma 5.3.14 *Let $\mathcal{K}^n$ be a ccodi of σ^n under ψ. In any coherent orientation of $\mathcal{K}^n$, all simplexes of n-corners($\mathcal{K}^n$) have the same orientation.*

Proof: Consider faces $\sigma^1, \sigma^2, \ldots, \sigma^{n-1}$ of σ^n such that $\sigma^1 \subset \sigma^2 \subset \ldots \subset \sigma^{n-1}$. For $1 \le i \le n-1$, let $\mathcal{K}^i$ denote the complex $\psi(\sigma^i)$. Assume $\mathcal{K}^i$ has the induced orientation by $\mathcal{K}^{i+1}$. By Lemma 5.3.1, $\mathcal{K}^i$ is coherently oriented. We proceed by induction on n. The base of the induction is for $\mathcal{K}^1$. Since $\mathcal{K}^1$ is chromatic and connected, $\mathcal{K}^1$ has an odd number of 1-simplexes. Also, the 1-simplexes containing the two vertexes in boundary, are the 1-corners. By Lemma 3.1.1, the simplexes of 1-*corners*($\mathcal{K}^1$) have the same orientation. Suppose the lemma holds for $i-1$. We prove that it holds for i.

By Definition 5.2.1 of n-corners, every simplex of $(i-1)$-*corners*($\mathcal{K}^{i-1}$) is contained in some simplex of i-*corners*($\mathcal{K}^i$). However, it is not necessary true that every simplex of i-*corners*($\mathcal{K}^i$) contains a simplex of $(i-1)$-*corners*($\mathcal{K}^{i-1}$). We first prove that the i-corners of $\mathcal{K}^i$ containing an $(i-1)$-corner of $\mathcal{K}^{i-1}$, have the same orientation. Let τ^{i-1} and ρ^{i-1} be simplexes of $(i-1)$-*corners*($\mathcal{K}^{i-1}$) and τ^i and ρ^i be the simplexes of i-*corners*($\mathcal{K}^i$) such that $\rho^{i-1} \subset \rho^i$ and $\tau^{i-1} \subset \tau^i$. By definition of orientability, τ^i induces its orientation multiplied by $(-1)^k$ to τ^{i-1}, and ρ^i induces its orientation multiplied by $(-1)^k$ to ρ^{i-1}, for some k. Also, by induction hypothesis, the simplexes of $(i-1)$-*corners*($\mathcal{K}^{i-1}$) have the same orientation, and thus τ^i and ρ^i have the same orientation.

Consider now a face λ^{i-1} of σ^i such that $\lambda^{i-1} \neq \sigma^{i-1}$. Let $\mathcal{L}^{i-1}$ denote the complex $\psi(\lambda^{i-1})$ and $\mathcal{L}^{i-2}$ denote the complex $\psi(\sigma^{i-1} \cap \lambda^{i-1})$. We now prove that simplexes of i-*corners*($\mathcal{K}^i$) containing a simplex of $(i-1)$-*corners*($\mathcal{K}^{i-1}$) or $(i-1)$-*corners*($\mathcal{L}^{i-1}$), have the same orientation. Consider a simplex γ^{i-2} of $(i-2)$-*corners*($\mathcal{L}^{i-2}$). Let $\tau^{i-1} \in (i-1)$-*corners*($\mathcal{K}^{i-1}$), $\rho^{i-1} \in (i-1)$-*corners*($\mathcal{L}^{i-1}$) and $\tau^i, \rho^i \in i$-*corners*($\mathcal{K}^i$) be the simplexes such that $\gamma^{i-2} \subset \tau^{i-1} \subset \tau^i$ and $\gamma^{i-2} \subset \rho^{i-1} \subset \rho^i$. By Lemma 5.3.13, τ^i and ρ^i have the same orientation. By the previous case, this one holds. This complete the proof.

■

Lemma 5.3.2 (Restated) *Let $\mathcal{K}^n$ be a ccodi of σ^n with structural symmetry with respect to $\mathcal{F}$. In any coherent orientation of $\mathcal{K}^n$, the n-simplexes of $\mathcal{K}^n$ that contain isomorphic $(n-1)$-simplexes, with respect to $\mathcal{F}$, of bd($\mathcal{K}^n$), have the same orientation.*

Proof: Let $\mathcal{K}_i^{n-1}$ and $\mathcal{K}_j^{n-1}$ denote $\psi(\sigma_i^{n-1})$ and $\psi(\sigma_j^{n-1})$. Consider isomorphic simplexes $\rho^{n-1} \in \mathcal{K}_i^{n-1}$ and $\tau^{n-1} \in \mathcal{K}_j^{n-1}$, i.e., for $f_{\sigma_i^{n-1}\sigma_j^{n-1}} \in \mathcal{F}$, $f_{\sigma_i^{n-1}\sigma_j^{n-1}}(\rho^{n-1}) = \tau^{n-1}$. Let ρ^n and τ^n be the unique simplexes of $\mathcal{K}^n$ such that $\rho^{n-1} \subset \rho^n$ and $\tau^{n-1} \subset \tau^n$. The induced orientation of ρ^{n-1} is the orientation of ρ^n multiplied by $(-1)^i$ and the induced orientation of τ^{n-1} is the orientation of τ^n multiplied by $(-1)^j$. Thus, it is sufficient to prove that ρ^{n-1} and τ^{n-1} have the same induced orientation, multiplied by $(-1)^i$ and $(-1)^j$, respectively.

By Definition 5.2.1, every simplex of $(n-1)$-*corners*$(\mathcal{K}_i^{n-1})$ or $(n-1)$-*corners*$(\mathcal{K}_j^{n-1})$ is face of a simplex in n-*corners*$(\mathcal{K}^n)$. Consider simplexes $\gamma^{n-1} \in (n-1)$-*corners*$(\mathcal{K}_i^{n-1})$ and $\lambda^{n-1} \in (n-1)$-*corners*$(\mathcal{K}_j^{n-1})$. By Lemma 5.3.14, γ^{n-1} and λ^{n-1} have the same induced orientation, multiplied by $(-1)^i$ and $(-1)^j$. By Lemma 3.1.1 and since $\mathcal{K}_i^{n-1}$ is connected, $\mathcal{K}_i^{n-1}$ has only two possible coherent orientations. Therefore, an orientation of an $(n-1)$-simplex of $\mathcal{K}_i^{n-1}$ induces the orientation of the other $(n-1)$-simplexes in a coherent orientation. Something similar happens with $\mathcal{K}_j^{n-1}$. It can be easily proved by induction on n, that any *ids*-preserving simplicial bijection $f : \mathcal{K}_i^{n-1} \to \mathcal{K}_j^{n-1}$, maps $(n-1)$-corners to $(n-1)$-corners. Therefore, an $(n-1)$-simplex of $\mathcal{K}_i^{n-1}$ is isomorphic to an $(n-1)$-simplex of $\mathcal{K}_j^{n-1}$ with the same orientation, multiplied by $(-1)^i$ and $(-1)^j$.

■

5.4 WSB Impossibility

Theorem 3.5.1 states that if there is a wait-free WSB protocol, there is a ccodi $\mathcal{K}^n$ of an n-simplex with a symmetric binary coloring and no monochromatic n-simplexes. That is, if $\mathcal{K}^n$ has the coloring c defined in Definition 4.1.4, $\mathcal{C}$ with respect to c must be zero, since each monochromatic n-simplex is monochromatic under b if and only if it is properly colored under c, by Lemma 4.1.5. Also, by Theorem 5.1.1

$$\mathcal{C} = 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i$$

Therefore, if there is a wait-free WSB protocol, the linear Diophantine equation

$$\binom{n+1}{1} k_0 + \binom{n+1}{2} k_1 + \ldots + \binom{n+1}{n} k_{n-1} = 1 \tag{5.1}$$

has an integer solution. A well-known result in number theory states that for non-zero integers $a_1, \ldots, a_j$ and an integer c, if there exist integers $x_1, \ldots, x_j$ such that $a_1x_1 + \ldots + a_jx_j = c$ then the greatest common divisor of $a_1, \ldots, a_j$, denoted $(a_1, \ldots, a_j)$, divides c (see for example [68, pp. 301]). Therefore, if $(\binom{n+1}{1}, \ldots, \binom{n+1}{n}) \neq 1$, namely, they are not relatively prime, then there are not integers $k_0, \ldots, k_{n-1}$ which satisfy equation (5.1). Since $\binom{n+1}{i+1} = \binom{n+1}{n-i}$, Theorem 5.4.1 focuses $\{\binom{n+1}{i+1} | 0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$.

Theorem 5.4.1 (WSB Impossibility) *If $\{\binom{n+1}{i+1} | 0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$ are not relatively prime then there does not exist an anonymous wait-free protocol that solves WSB.*

For example, it is easy to check that if $n+1$ is prime then $\mathcal{C} \equiv 1 \bmod (n+1)$. Therefore, there exist infinitely many cases in which WSB is not wait-free solvable. Also, it can be easily proved that $\mathcal{C} \equiv 1 \bmod 2$, if $n+1 = 4$.

Corollary 5.4.2 *If $n+1$ is prime or 4 then there does not exist an anonymous wait-free protocol that solves WSB.*

Theorem 5.4.1 implies the following result, which is a special case of Theorem 6.2 in [56] and Theorem 6.3 in [55], recalling that WSB is equivalent to $2n$-renaming (see Section 2.2).

Corollary 5.4.3 (Renaming lower bound) *If $\{\binom{n+1}{i+1} | 0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$ are not relatively prime then there does not exist an anonymous wait-free protocol that solves M-renaming with $M < 2n+1$.*

Chapter 6

A WSB Protocol for Exceptional Values of n

Chapter 5 proved that, for some non-exceptional values of n, WSB is not wait-free solvable. This chapter shows that WSB is indeed wait-free solvable for the other exceptional values of n. More precisely, it presents a construction that produces a ccosdi of an n-simplex with a rank-symmetric binary coloring and without monochromatic n-simplexes, for any exceptional value of n, which implies the existence of a wait-free WSB protocol, by Theorem 3.5.2.

6.1 The Construction

The impossibility proof of WSB in Chapter 5 is based on a characterization of the number of monochromatic n-simplexes, $\mathcal{C}$, of a ccodi of an n-simplex with a symmetric binary coloring; some monochromatic n-simplexes are counted as $+1$ and the other as -1. That characterization states that $\mathcal{C} = 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i$, for some integers $k_0, \ldots, k_{n-1}$. Analyzing this expression one can see that $\mathcal{C} \neq 0$, if n is non-exceptional, which implies the impossibility of WSB.

The construction presented in this chapter consists on the following two steps. The first step, described in Section 6.2, shows that, given integers $k_0, \ldots, k_{n-1}$ with $k_0 \in \{0, -1\}$, it is possible to construct a ccosdi $\mathcal{K}^n$ of an n-simplex with a rank-symmetric binary coloring and $\mathcal{C} = 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i$. The second part of the construction, in Section 6.3, subdivides the interior of $\mathcal{K}^n$ to remove some monochromatic n-simplexes. Intuitively, it repeatedly takes a path of n-simplexes connecting two monochromatic n-simplexes counted as $+1$ and -1, and subdivides it to remove the monochromatic simplexes at its ends. The resulting ccosdi has exactly $|\mathcal{C}|$ monochromatic n-simplexes. Therefore, the construction gives the following result.

Theorem 6.1.1 *Let $k_0, k_1 \ldots k_{n-1}$ be integers such that $k_0 \in \{0, -1\}$. There exists a ccosdi of an n-simplex with a rank-symmetric binary coloring,*

$$\mathcal{C} = 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i$$

and with exactly $|\mathcal{C}|$ monochromatic n-simplexes.

It turns out that if n is exceptional, there exists integers $k_0, \ldots, k_{n-1}$ such that $\mathcal{C} = 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i = 0$, hence WSB is wait-free solvable, by Theorem 3.5.2.

6.2 Constructing Divided Images

Theorem 5.1.1 says that the number of monochromatic n-simplexes, the content $\mathcal{C}$, of a ccodi of an n-simplex with a symmetric binary coloring, is $\mathcal{C} = 1+\sum_{i=0}^{n-1}\binom{n+1}{i+1}k_i$, for some integers $k_0,\ldots,k_{n-1}$. This section presents the opposite direction, namely, it shows a construction that, given integers $k_0,\ldots,k_{n-1}$ with $k_0 \in \{0,-1\}$, produces a ccodi $\mathcal{K}^n$ of an n-simplex with a symmetric binary coloring and $\mathcal{C} = 1+\sum_{i=0}^{n-1}\binom{n+1}{i+1}k_i$. Moreover, $\mathcal{K}^n$ is a ccosdi and its binary coloring is rank-symmetric. This construction implies the following theorem.

Theorem 6.2.1 *Let $k_0, k_1 \ldots k_{n-1}$ be integers such that $k_0 \in \{0,-1\}$. There exists a ccosdi of an n-simplex with a rank-symmetric binary coloring and*

$$\mathcal{C} = 1+\sum_{i=0}^{n-1}\binom{n+1}{i+1}k_i$$

Recall that $\mathcal{C}$ can be easily computed by counting each b-monochromatic n-simple with orientation d, $d \in \{+1,-1\}$, as $(-1)^{b*n}d$, by Lemma 4.1.6.

Section 6.2.1 first presents a small example of the construction with $\mathcal{C} = 0$ and $n = 5$, the first exceptional value of n.

6.2.1 An 5-dimensional example

This section describes how to construct a ccosdi $\mathcal{K}^5$ of σ^5 under ψ, with a rank-symmetric binary coloring and $\mathcal{C} = 0$. Dimension 5 is the smallest dimension in which it is possible to construct such a divided image, because $n = 5$ is the smallest exceptional value, i.e., the integers in the set $\{\binom{n+1}{i+1} : 0 \le i \le \lfloor\frac{n-1}{2}\rfloor\}$ are relatively prime. For the example, let $k_0 = -1, k_1 = -1, k_2 = 1, k_3 = 0, k_4 = 0$, as $1+\binom{6}{1}k_0+\binom{6}{2}k_1+\binom{6}{3}k_2+\binom{6}{4}k_3+\binom{6}{5}k_4 = 0$.

The strategy is essentially the same as the one used in the cone based proof of Theorem 5.1.1, but inverting the process. The idea is to construct $\psi(bd(\sigma^5))$ such that the divided image of any proper i-face of σ^5 has exactly $|k_i|$ 0-monochromatic i-simplexes. Once $\psi(bd(\sigma^5))$ is constructed, $\mathcal{K}^5$ is the cone over $\psi(bd(\sigma^5))$ for a 0-monochromatic 5-simplex. As we shall see, the divided image of any proper i-face generates $|k_i|$ 0-monochromatic 5-simplexes, each one of them counted as $sign(k_i)$.

First, for each face σ^0 of σ^5, $\psi(\sigma^0)$ is a vertex v with $b(v) = 0$ (because $k_0 = -1$). For $i = 1,2,3,4$, assume it has been constructed $\psi(bd(\sigma^j))$, for every $\sigma^j \in bd(\sigma^5)$, $0 \le j < i$. Then, take a face σ^i of σ^5, and construct the cone over $\psi(bd(\sigma^i))$ with some new 1-monochromatic i-simplex. Once this is done, $\psi(\sigma^i)$ is chromatically subdivided until it has exactly $|k_i|$ 0-monochromatic i-simplexes. In the next section we will see that these i-simplexes must have orientation $sign(k_i)$ in a coherent orientation of $\psi(\sigma^i)$ such that at least on of its i-corners has orientation $(-1)^{i+1}$. The goal is that the 0-monochromatic i-simplexes of $\psi(\sigma^i)$ will generate $|k_i|$ 0-monochromatic 5-simplexes in $\mathcal{K}^5$, each one of them counted as $sign(k_i)$. This construction is repeated for each i-face, preserving the rank of id colors, to make sure $\mathcal{K}^5$ will be rank-symmetric.

The case of $i = 1$ is illustrated in Figure 6.1. The goal is to create exactly one 0-monochromatic 1-simplex in the interior of $\psi(\sigma^1)$ (because $|k_1| = 1$) that has orientation $sign(k_1)$ in a coherent orientation of $\psi(\sigma^1)$ in which at least one of its 1-corners is positively oriented (because i is odd). Figure 6.1 (a) contains the cone over $\psi(bd(\sigma^1))$ for a new 1-simplex, and Figures 6.1 (b) and (c) shows how $\psi(\sigma^1)$ is subdivided until it has the desired 0-monochromatic 1-simplex. The reader can verify that in Figures 6.1 (a) and (b) it is not possible to produce such a simplex by changing the binary color of the interior vertexes. Recall that in any coherent orientation of a chromatic n-pseudomanifold, every pair of n-simplexes that share an $(n-1)$-face, have opposite orientations, by Lemma 3.1.1 The construction is the same, preserving id ranking, for every 1-face of σ^5.

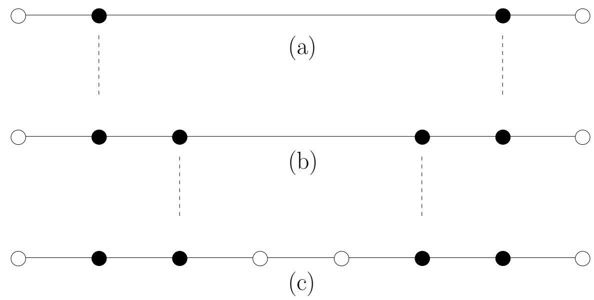

Figure 6.1: The construction for k_1.

For a 2-face σ^2 of σ^5, $\psi(\sigma^2)$ is the cone over $\psi(bd(\sigma^2))$ for a new 1-monochromatic 2-simplex. It is now needed exactly one 0-monochromatic 2-simplex in the interior of $\psi(\sigma^2)$ (because $|k_2| = 1$), and also that simplex must have orientation $sign(k_2)$ in a coherent orientation of $\psi(\sigma^2)$ in which at least one of its 2-corners is negatively oriented (because i is even). In this case it is enough to color with 0 only one vertex of the 2-simplex at the center. Figure 6.2 presents an example of $\psi(\sigma^2)$.

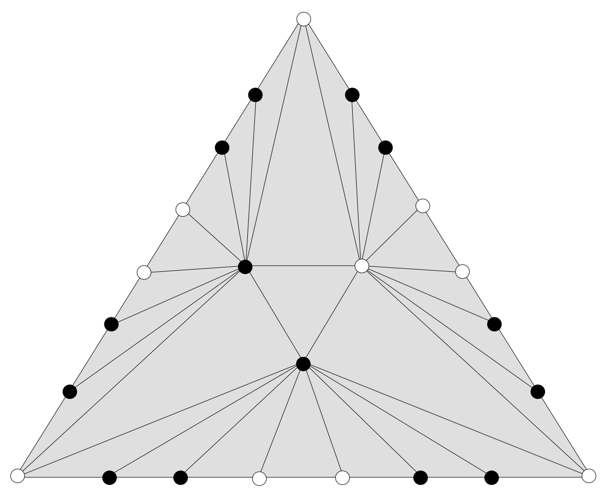

Figure 6.2: The construction for k_2.

Since $k_3 = k_4 = 0$, for $3 \le i \le 5$, $\psi(\sigma^i)$ is cone constructed with a 1-monochromatic i-simplex at the center and without further subdivisions.

Finally, $\mathcal{K}^5$ is the cone of $\psi(bd(\sigma^5))$ for a 0-monochromatic simplex τ^5. Then $\mathcal{K}^5$ is coherently oriented such that τ^5 is positively oriented. Therefore, $\mathcal{C}$ is 1, counting τ^5, plus $6k_0 + 15k_1 + 20k_2 = -6 - 15 + 20$, counting the 0-monochromatic 5-simplexes generated by the divided images of the i-faces of σ^5, $0 \le i \le 2$.

6.2.2 The construction for dimension n

This section presents the general construction that proves Theorem 6.2.1. Figure 6.3 contains the construction **ConstructCcosdi** that takes integers $k_0, \dots, k_{n-1}$ with $k_0 \in \{0, -1\}$, and produces a ccosdi of an n-simplex with a rank-symmetric binary coloring and $\mathcal{C} = 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i$.

ConstructCcosdi($k_0, k_1, \dots, k_{n-1}$)

(01) **init:** for every 0-face σ^0 of σ^n, $\psi(\sigma^0)$ is a vertex v with $id(v) = id(\sigma^0)$ and $b(v) = 1$

(02) **if** $k_0 = -1$ **then** $b(v) \leftarrow 0$

(03) **for** $i \leftarrow 1$ **to** $n-1$ **do**

(04)   **let** σ^i a face of σ^n

(05)   **do** the cone over $\psi(bd(\sigma^i))$ for a new 1-monochromatic τ^i with $id(\tau^i) = id(\sigma^i)$

(06)   **give** a coherent orientation to $\psi(\sigma^i)$ such that at least one of its i-corners has orientation $(-1)^{i+1}$

(07)   **for** $j \leftarrow 1$ **to** $|k_i|$ **do**

(08)   **do** basic chromatic subdivisions over $\psi(\sigma^i)$ using new 1-monochromatic i-simplexes and without modifying its 0-monochromatic i-simplexes, until it has a $sign(k_i)$ oriented non 0-monochromatic i-simplex γ such that $\forall \rho^i \in st(\gamma) \setminus \gamma$, $\exists v \in \rho^i$ such that $v \notin \gamma$ and $b(v) = 1$

(09)   **do** $b(\gamma^i) \leftarrow 0$

(10)   **copy** $\psi(\sigma^i)$ to the divided image of every i-face, preserving the rank of id colors

(11) **do** the cone over $bd(\psi(\sigma^n))$ for a new 0-monochromatic τ^n

(12) **give** a coherent orientation to $\psi(\sigma^n)$ such that τ^n is positively oriented

Figure 6.3: The general construction.

The following two definitions are used in **ConstructCcosdi**. Consider two chromatic n-simplexes σ^n and τ^n. For $bd(\sigma^n)$ and τ^n, the *basic chromatic subdivision* over $bd(\sigma^n)$ for τ^n, denoted $\tau^n \circledast bd(\sigma^n)$, is the cone over $\varphi(bd(\sigma^n))$ for τ^n, where $\varphi(\sigma) = \sigma$ for each $\sigma \in bd(\sigma^n)$. Figure 6.4 depicts a basic chromatic subdivision of dimension 2. Consider a complex $\mathcal{L}$ and one of its vertexes v. If $\mathcal{L}$ is understood, the star complex $st(v, \mathcal{L})$ is denoted $st(v)$. For each simplex $\rho = \{v_0 \dots v_n\}$ of $\mathcal{L}$, let $st(\rho)$ denote $st(v_0) \cup \dots \cup st(v_n)$.

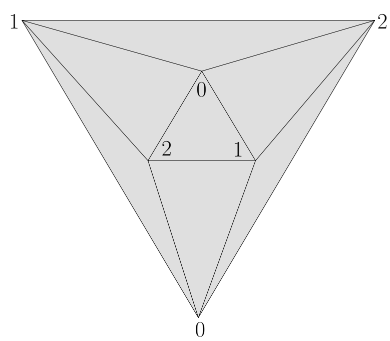

Figure 6.4: The basic chromatic subdivision of dimension 2.

ConstructCcosdi only uses the cone construction and the basic chromatic subdivision to construct $\mathcal{K}^n$, thus $\mathcal{K}^n$ is a ccosdi σ^n under ψ, at the end of line 12, by Lemma 4.2.3. Also, $\mathcal{K}^n$ is coherently oriented. By Lemma 4.2.2, $\mathcal{K}^n$ does not have 1-monochromatic n-simplexes because the simplex at the center τ^n is 0-monochromatic. In addition, line 10 guarantees that $\mathcal{K}^n$ has a rank-symmetric binary coloring. Thus, if we prove that, for any proper i-face σ of σ^n, the 0-monochromatic i-simplexes of $\psi(\sigma)$ generate $|k_i|$ 0-monochromatic n-simplexes oriented $sign(k_i)$, then $\mathcal{C} = 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i$, as each d oriented 0-monochromatic n-simplex is counted as d by $\mathcal{C}$, by Lemma 4.1.6. Recall that, by Lemma 4.2.2, for each i-simplex $\gamma \in \psi(\sigma^i)$, the n-simplex generated by γ is the simplex $\tau * \gamma$, where τ is the unique face of τ^n such that $dim(\tau * \gamma) = n$.

Line 1 guarantees that $\psi(\sigma^0)$ only contains one vertex, v. Also, v generates an n-simplexes, $\tau^{n-1} * v$, in $\mathcal{K}^n$ such that $\tau^{n-1} \subset \tau^n$, hence τ^n and $\tau^{n-1} * v$ share an $(n-1)$-face. By Lemma 3.1.1 and since τ^n is positively oriented, $\tau^{n-1} * v$ is negatively oriented. Moreover, $\tau^{n-1} * v$ is 0-monochromatic if and only if $b(v) = 0$, and $b(v) = 0$ if and only if $k_0 = -1$.

Consider the i-th iteration of the for loop in line 3. Line 6 gives a coherent orientation to $\psi(\sigma^i)$ in which at least one of its i-corners has orientation $(-1)^{i+1}$ (in fact, in any coherent orientation of $\psi(\sigma^i)$, all its i-corners have the same orientation, by Lemma 5.3.14). Line 8 subdivides $\psi(\sigma^i)$ until it has a non 0-monochromatic i-simplex with orientation $sign(k_i)$, which is "surrounded" by i-simplexes that will not be 0-monochromatic after line 9. Therefore, after the for loop in line 7, $\psi(\sigma^i)$ has exactly $|k_i|$ 0-monochromatic i-simplexes, all of them $sign(k_i)$ oriented. The proof that line 8 always finishes is postponed a little bit. Now, notice that the 0-monochromatic i-simplexes of $\psi(\sigma^i)$ generate $|k_i|$ 0-monochromatic n-simplexes in $\mathcal{K}^n$. In what follows it is proved that each one of these n-simplexes has orientation $sign(k_i)$. The argument is similar to the one used in the proof of Lemma 5.2.2.

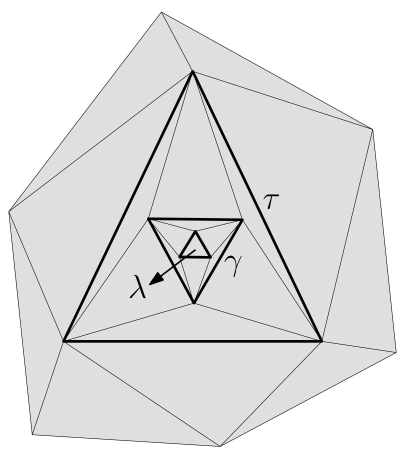

Figure 6.5: Obtaining an isolated simplex.

Consider an i-corner ρ^i of $\psi(\sigma^i)$ with orientation $(-1)^{i+1}$. Let γ^n be the n-simplexes generated by ρ^i. By Definition 5.2.1 of i-corners, for some faces $\sigma^0, \sigma^1, \ldots, \sigma^i$, there exist simplexes $\lambda^0 \in \psi(\sigma^0), \lambda^1 \in \psi(\sigma^1), \ldots, \lambda^i = \rho^i \in \psi(\sigma^i)$ such that $\lambda^0 \subset \lambda^1 \subset \ldots \subset \lambda^i$. Let $\tau_j * \lambda^j$ be the n-simplex generated by λ^j. Hence $\tau_i * \lambda^i = \gamma^n$. Notice that $\tau_j * \lambda^j$ and $\tau_{j+1} * \lambda^{j+1}$, $0 \le j \le i-1$, share an $(n-1)$-face and $\tau_0 * \lambda^0$ and τ^n share τ_0, which is $(n-1)$-dimensional. Therefore, simplexes $\tau^n, \tau_0 * \lambda^0, \tau_1 * \lambda^1, \ldots, \tau_i * \lambda^i = \gamma^n$ are an n-path of length $i+2$. By Lemma 3.1.1 and since $\mathcal{K}^n$ is coherently oriented such that τ^n is positively oriented, γ^n has orientation $(-1)^{i+1}$. In other words, ρ^i and γ^n have the same orientation in $\psi(\sigma^i)$ and $\mathcal{K}^n$, respectively.

Consider now a 0-monochromatic i-simplex $\varrho^i \in \psi(\sigma^i)$. Let λ^n be the n-simplexes generated by ϱ^i. Since $\psi(\sigma^i)$ is connected, there is an i-path from the i-corner ρ^i to ϱ^i. Let $\mathcal{P}$ such a path. The

i-simplexes of $\mathcal{P}$ generate n-simplexes in $\mathcal{K}^n$. Moreover, consecutive i-simplexes in $\mathcal{P}$ (sharing an $(i-1)$-face) generate n-simplexes that share an $(n-1)$-face, i.e., $\mathcal{P}$ generates an n-path from γ^n to λ^n. Thus, ϱ^i and λ^n have the same orientation in $\psi(\sigma^i)$ and $\mathcal{K}^n$, respectively, by Lemma 3.1.1 and because ρ^i and γ^n have the same orientation in $\psi(\sigma^i)$ and $\mathcal{K}^n$.

We now prove that line 8 always finishes. Consider a non 0-monochromatic i-simplex τ of $\psi(\sigma^i)$, and let $\mathcal{L}$ be the complex $st(\tau)$ without τ. Observe that if we do a basic chromatic subdivision over τ for γ and then a basic chromatic subdivision over γ for λ, we get $\lambda \notin \mathcal{L}$ (see Figure 6.5 for an example of dimension 2). Therefore, if we color λ with 0 then exactly one 0-monochromatic i-simplex in $\psi(\sigma^i)$ is created. Theorem 6.2.1 follows.

6.3 Eliminating monochromatic simplexes

Consider an orientable, connected and chromatic pseudomanifold $\mathcal{K}^n$ with a binary coloring, and let $\mathcal{C}$ be its content. This section presents the algorithm **Eliminate**, which takes $\mathcal{K}^n$ as input and produces a subdivision of $\mathcal{K}^n$ with the same boundary and with exactly $|\mathcal{C}|$ monochromatic n-simplexes. Algorithm **Eliminate** implies the following theorem, which together with 6.2.1 prove Theorem 6.1.1.

Theorem 6.3.1 *Let $\mathcal{K}^n$ be a chromatic, connected and orientable pseudomanifold with a binary coloring and content $\mathcal{C}$. There is a chromatic and orientable pseudomanifold $\chi(\mathcal{K}^n)$ with a binary coloring such that it has exactly $|\mathcal{C}|$ monochromatic n-simplexes, $bd(\mathcal{K}^n) = bd(\chi(\mathcal{K}^n))$ and $\chi(\mathcal{K}^n)$ is a subdivision of $\mathcal{K}^n$*

An overview of algorithm **Eliminate** and its main part, function **EliminatePath**, is presented in Section 6.3.1. Some definitions and lemmas that are used for proving the correctness of **EliminatePath**, appear in Sections 6.3.2 and 6.3.3. The correctness proof of **EliminatePath** is contained in Section 6.3.4.

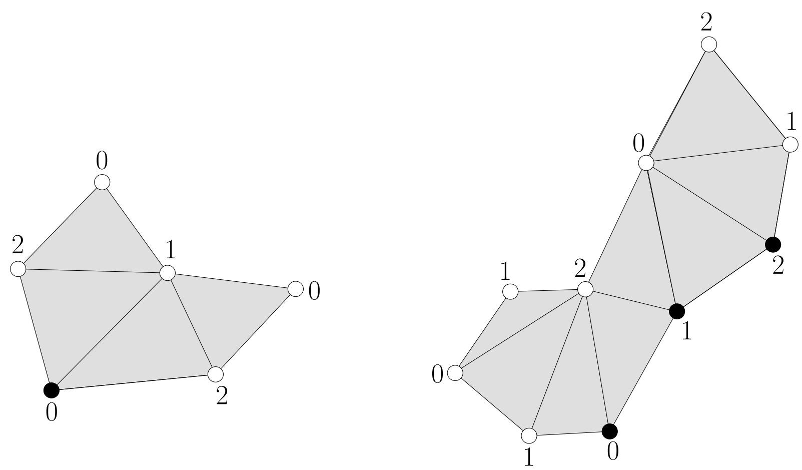

Figure 6.6: Paths in standard and non-standard form.

6.3.1 An Overview of Algorithm Eliminate

Recall that an n-path is a sequence of distinct n-simplexes, and all its faces, such that every two consecutive n-simplexes share an $(n-1)$-face. An n-path $\mathcal{Q}$ with simplexes $\sigma_0, \sigma_1, \ldots$ is denoted $\mathcal{Q} : \sigma_0 - \sigma_1 - \cdots$. Thus, σ_i and σ_{i+1} share an $(n-1)$-face, denoted $\sigma_{i,i+1}$. The segment of the path from σ_i to σ_j, $i \leq j$, is denoted $\mathcal{Q}_{i,j}$. For a simplex σ, let $x(\sigma)$ be the maximal x-monochromatic face of σ, considering its binary coloring, and let $\#x(\sigma)$ be $|x(\sigma)|$.

Consider an n-path $\mathcal{P} : \sigma_0 - \sigma_1 - \cdots - \sigma_q$, $q \geq 1$, with a binary coloring at its vertexes such that the n-simplexes at its ends, σ_0 and σ_q, are 0-monochromatic and it has no other monochromatic n-simplex. We say that $\mathcal{P}$ is in *standard form* if and only if σ_0 and σ_q have opposite orientations in a coherent orientation of $\mathcal{P}$, i.e., $|\mathcal{P}|$ is even, by Lemma 3.1.1. Therefore, σ_0 and σ_q are counted in an opposite way by $\mathcal{C}$. Figure 6.6 depicts two paths of dimension 2, one of them in standard form and the other in non-standard form.

Let $\mathcal{P}$ be a path in standard form. A *good subdivision of* $\mathcal{P}$ is a chromatic subdivision $\chi(\mathcal{P})$ of $\mathcal{P}$ that contains two disjoint paths $\mathcal{P}_1$ and $\mathcal{P}_2$ in standard form (in the sense that they do not share n-simplexes), has no other monochromatic n-simplex and $bd(\mathcal{P}) = bd(\chi(\mathcal{P}))$. A *complete subdivision of* $\mathcal{P}$ is a chromatic subdivision $\chi(\mathcal{P})$ of $\mathcal{P}$ with no monochromatic n-simplex and $bd(\mathcal{P}) = bd(\chi(\mathcal{P}))$.

Eliminate(K^n)
while K^n has more than $|\mathcal{C}|$ monochromatic n-simplexes **then**
    $\mathcal{P} \leftarrow$**FindPathStandardForm**(K^n)
    EliminatePath($\mathcal{P}$)

Figure 6.7: Algorithm **Eliminate**.

The algorithm **Eliminate**, Figure 6.7, repeatedly looks for pairs of 0-monochromatic n-simplexes that are counted as $+1$ and -1 by $\mathcal{C}$. Then it considers a path $\mathcal{P}$ in standard form connecting such a pair of n-simplexes. The main work of the algorithm is done by function **EliminatePath**, Figure 6.8, which produces a complete subdivision of $\mathcal{P}$. This function works as fallows. If the input is the empty path, there is nothing to do. If the length of the input is two, that is, it consists only of two monochromatic n-simplexes, it just subdivides their shared $(n-1)$-face. The function **SubdivideComp** does this subdivision. Figure 6.9 presents an example of dimension 2.

EliminatePath($\mathcal{P}$)
if $\mathcal{P} \neq \emptyset$ **then**
    if $|\mathcal{P}| = 2$ **then**
        SubdivideComp($\mathcal{P}$)
    else
        $m \leftarrow$ **FindSubdividingPoint**($\mathcal{P}$)
        $P1, P2 \leftarrow$ **SubdivideGood**($\mathcal{P}, m$)
        EliminatePath($P1$)
        EliminatePath($P2$)

Figure 6.8: Algorithm **EliminatePath**.

If the input is longer than two, **EliminatePath** invokes function **FindSubdividingPoint**, Figure 6.10, to get a *subdividing point* m where the algorithm will subdivide a shared $(n-1)$-face of $\mathcal{P}$, either $\sigma_{m-1,m}$ or $\sigma_{m,m+1}$. Then **EliminatePath** subdivides that face using **SubdivideGood**, which creates a good subdivision of $\mathcal{P}$ such that $\mathcal{P}_1$ and $\mathcal{P}_2$ are not longer than $\mathcal{P}$. Intuitively, this subdivision creates two monochromatic n-simplexes with opposite orientation inside $\mathcal{P}$ (adding zero to the content), and then obtains two new paths in standard form. Now, if $|\mathcal{P}_i| < |\mathcal{P}|$, for all $i \in \{1, 2\}$, we say that the subdividing point was *progressive*. And if $|\mathcal{P}_i| = |\mathcal{P}|$, for some $i \in \{1, 2\}$, we will prove that $\mathcal{P}_i$ always has a progressive subdividing point. This guarantees that **EliminatePath** makes progress in each invocation. Finally, **EliminatePath** recursively calls itself on the two paths $\mathcal{P}_1$ and $\mathcal{P}_2$.

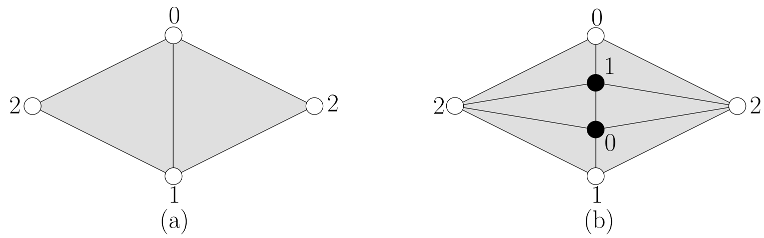

Figure 6.9: A path of length two.

Figure 6.11 (a) presents a 2-dimensional path of length four that has a subdividing point on $m = 1$. **SubdivideGood** subdivides the shared face $\sigma_{1,2}$, Figure 6.11 (b), to create two 0-monochromatic n-simplexes and then obtain a good subdivision with two paths of length two. The resulting paths will be subdivided as in the case presented in Figure 6.9 on the next recursive call of **EliminatePath** on them, Figure 6.11 (c).

FindSubdividingPoint$(\mathcal{P} : \sigma_0 - \sigma_1 - \cdots - \sigma_{2q+1})$

$m \leftarrow 1$
while *true* **do**
  if $\#0(\sigma_{m+1,m+2}) \geq n + 1 - m$ **then**
    return m
  $m \leftarrow m + 1$

Figure 6.10: Algorithm **FindSubdividingPoint**.

The discussion has been focused on 0-monochromatic n-simplexes because a 1-monochromatic n-simplex can be easily transformed into a 0-monochromatic one using the basic chromatic subdivision, Figure 6.12.

Finally, to prove the correctness of **Eliminate**, it is enough to prove the correctness of **EliminatePath**, i.e., it produces a complete subdivision of paths in standard form.

6.3.2 Subdividing Points

This section presents the definition of subdividing point and its properties. Roughly speaking, the subdividing point of an n-path in standard form is a "place" where it is possible to subdivide a shared $(n-1)$-face, and produce a good subdivision with resulting paths of lengths at most the length of the original path. The proofs of the lemmas presented here are simple and appear in Appendix A.

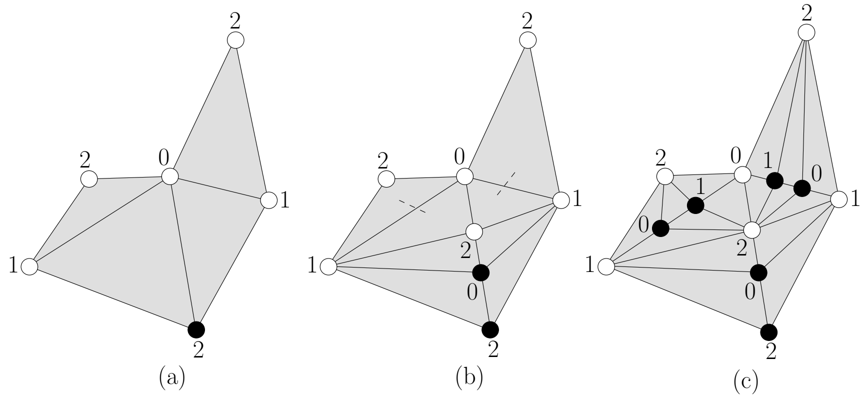

Figure 6.11: A path of length four.

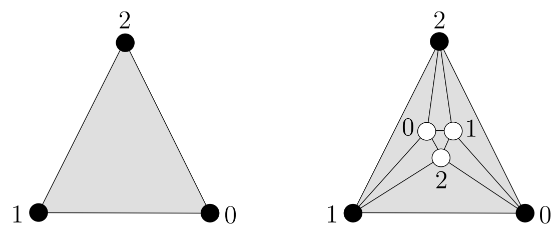

Figure 6.12: Obtaining a 0-monochromatic simplex.

Consider a path $\mathcal{P} : \sigma_0 - \sigma_1 - \cdots - \sigma_{2q+1}$ in standard form. For the rest of the section fix $\mathcal{P}$ and its subdividing point m defined as follows.

Definition 6.3.2 *The* subdividing point *of $\mathcal{P}$ is the smallest value m for which $\#0(\sigma_{m+1,m+2}) \geq n + 1 - m$.*

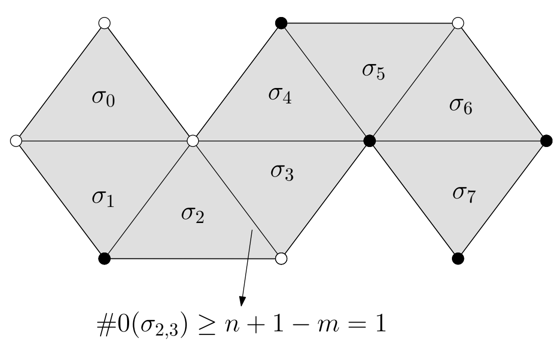

Figure 6.13: A path of dimension $n = 2$ with subdividing point $m = 1$.

Figure 6.13 shows a path in standard form of dimension 2 with subdividing point 1. The following lemmas give bounds for the subdividing point m and relate the length of $\mathcal{P}$ with m.

Lemma 6.3.3 *Let m be the subdividing point of $\mathcal{P}$. Then $1 \leq m \leq min(q, n+1)$.*

Lemma 6.3.4 *Let m be the subdividing point of $\mathcal{P}$. Then $|\mathcal{P}| \geq 2(m+1)$.*

Lemma 6.3.5 below shows the "configuration" of 0's of the n-simplexes and shared $(n-1)$-faces "around" the subdividing point m. It represents such configurations as a segment of $\mathcal{P}$ and the possible number of 0's that an n-simplex or shared $(n-1)$-face can have. The following example considers the number of 0's of the segment from σ_m and σ_{m+1}, including the faces $\sigma_{m-1,m}$ and $\sigma_{m+1,m+2}$.

$$\begin{array}{ccccc}
x+1 & x+1 & & & x \\
x & x & x & x & x-1 \\
\text{———————-} & \sigma_m & \text{———————} & \sigma_{m+1} & \text{———————-}
\end{array}$$

The example shows that $\#0(\sigma_{m-1,m}) \in \{x, x+1\}$, $\#0(\sigma_m) \in \{x, x+1\}$, $\#0(\sigma_{m,m+1}) = x$, $\#0(\sigma_{m+1}) = x$ and $\#0(\sigma_{m+1,m+2}) = x \in \{x, x-1\}$. The possible configurations are all those valid assignments of values. The reader can verify that the invalid assignments are those assignments in which $\#0(\sigma_m) = x$ and $\#0(\sigma_{m-1,m}) = x+1$, since $\sigma_{m-1,m}$ is an $(n-1)$-face of σ_m and hence it can have at most the same number of vertexes with binary color 0 as σ_m. In general, the rule is that if σ' is an $(n-1)$-face of σ then $\#0(\sigma') \in \{\#0(\sigma)-1, \#0(\sigma)\}$.

Lemma 6.3.5 *For the subdividing point m of $\mathcal{P}$:*

$$\begin{array}{ccccc}
n+2-m & n+2-m & n+1-m & n+2-m & n+2-m \\
n+1-m & n+1-m & n-m & n+1-m & n+1-m \\
\text{———————-} & \sigma_m & \text{———————-} & \sigma_{m+1} & \text{———————-}
\end{array}$$

According to the possible configurations described in Lemma 6.3.5, the following cases in Lemma 6.3.6 are identified. Later on, in Section 6.3.4, it is proved that algorithm **EliminatePath** appropriately handles each one of these cases.

Lemma 6.3.6 *For the subdividing point m of $\mathcal{P}$, one of the following cases holds:*

$$\begin{array}{cccccc}
 & n+1-m & n+1-m & n-m & n+1-m & n+1-m \\
A) & \text{———————-} & \sigma_m & \text{———————} & \sigma_{m+1} & \text{———————-}
\end{array}$$

$$\begin{array}{cccccc}
 & n+1-m & n+1-m & n+1-m & n+1-m & n+1-m \\
B) & \text{———————-} & \sigma_m & \text{———————} & \sigma_{m+1} & \text{———————-}
\end{array}$$

$$\begin{array}{cccccc}
 & n+2-m & & & & \\
 & n+1-m & n+2-m & n+1-m & n+1-m & n+1-m \\
C) & \text{———————-} & \sigma_m & \text{———————} & \sigma_{m+1} & \text{———————-}
\end{array}$$

	$n+1-m$	$n+1-m$	$n+1-m$	$n+2-m$	$n+2-m$ $n+1-m$
D)	——————————-	σ_m	——————————	σ_{m+1}	——————————-

	$n+2-m$ $n+1-m$	$n+2-m$	$n+1-m$	$n+2-m$	$n+2-m$ $n+1-m$
E)	——————————-	σ_m	——————————	σ_{m+1}	——————————-

Definition 6.3.7 *The subdividing point m of $\mathcal{P}$ is* progressive *if it holds case A, C or D or case E with $m < n+1$, otherwise it is* non-progressive.

By Lemma 6.3.3, $1 \leq m \leq min(q, n+1)$, and by Lemma 6.3.4, $|\mathcal{P}| \geq 2(m+1)$. And since $n+1-m = 0$ only if $m = n+1$, then for cases A, B, C and D of Lemma 6.3.6, it must be $m \leq min(q, n)$ because σ_m and σ_{m+1} are not monochromatic. Similarly, since $n+2-m = n+1$ only if $m = 1$, then for cases C, D and E it must be $2 \leq m$. Also, for case B we have that $2 \leq m$, because if $m = 1$ then $\#0(\sigma_m) = n+1-m = n$, and hence $\sigma_{0,1}$ is the unique 0-monochromatic $(n-1)$-face of $\sigma_m = \sigma_1$, and it cannot be $\sigma_{0,1} = \sigma_{1,2}$ because $\mathcal{P}$ is a pseudomanifold.

Lemma 6.3.8 *If the subdividing point m of $\mathcal{P}$ holds case E then $2 \leq m \leq min(q, n+1)$, if m holds case A then $1 \leq m \leq min(q, n)$, otherwise $2 \leq m \leq min(q, n)$.*

6.3.3 Crossing and Non-Crossing Paths

As explained in Section 6.3.1, algorithm **EliminatePath** produces a good subdivision of a path in standard form $\mathcal{P}$ by subdividing one of its shared $(n-1)$-faces. This section introduces the classes of paths that appear in the subdivisions done by **SubdivideGood**. First, some definitions are presented.

For the rest of this chapter, let *id* and *b* denote the chromatic and binary coloring of simplexes, paths and subdivisions. For properly colored n-simplexes σ and σ', let $\sigma \xrightarrow{id:k} \sigma'$ denote the n-path of length 2 in which σ and σ' share their $(n-1)$-face without *id* k. This is a *step* that *changes* the vertex with *id* color k. Let $\sigma \xrightarrow{id:A} \sigma'$ denote an n-path of length $|A|+1$ from σ to σ', which contains exactly one step for every element of A. If there is no ambiguity we just write $\sigma \xrightarrow{k} \sigma'$ or $\sigma \xrightarrow{A} \sigma'$.

Consider a path $\sigma_1 \xrightarrow{k} \sigma_2$ and an $(n-1)$-simplex τ that is properly colored with ID_k^n. Let $\sigma_{1,2}$ be the shared $(n-1)$-face between σ_1 and σ_2 and v_i, $i \in \{1,2\}$, be the unique vertex in $\sigma_i \setminus \sigma_{1,2}$. The *double chromatic cone of* $\sigma_1 \xrightarrow{k} \sigma_2$ *and* τ is the complex $v_1 * (\tau \circledast bd(\sigma_{1,2})) \cup v_2 * (\tau \circledast bd(\sigma_{1,2}))$ (recall that $\tau \circledast bd(\sigma_{1,2})$ is the basic chromatic subdivision over $bd(\sigma_{1,2})$ for τ). Since the join operator has been restricted to its chromatic version, $v_1 * (\tau \circledast bd(\sigma_{1,2})) \cup v_2 * (\tau \circledast bd(\sigma_{1,2})) = v_1 * v_2 * (\tau \circledast bd(\sigma_{1,2}))$. By Lemma 4.2.3, $\tau \circledast bd(\sigma_{1,2})$ is a chromatic subdivision of $\sigma_{1,2}$ with $bd(\sigma_{1,2}) = bd(\tau \circledast bd(\sigma_{1,2}))$. Using a similar argument to the one used in the proof of Lemma 4.2.6, it can be proved that $v_1 * v_2 * (\tau \circledast bd(\sigma_{1,2}))$ is a chromatic subdivision of $\sigma_1 \xrightarrow{k} \sigma_2$ with $bd(\sigma_1 \xrightarrow{k} \sigma_2) = bd(v_1 * v_2 * (\tau \circledast bd(\sigma_{1,2})))$.

Consider a simplex $\sigma' \in bd(\sigma_{1,2})$ (possibly empty). Let $\tau' * \sigma'$ be the $(n-1)$-simplex of $\tau \circledast bd(\sigma_{1,2})$ generated by σ'. Notice that $\tau' * \sigma'$ is the unique $(n-1)$-simplex of $\tau \circledast bd(\sigma_{n-1})$ that contains $\tau' \neq \emptyset$ as maximal face of τ. Thus, $\tau' * \sigma'$ can be denoted without ambiguity by $[\tau']$. Also, observe that $[\tau']$ belongs to exactly two n-simplexes of $v_1 * v_2 * (\tau \circledast bd(\sigma_{1,2}))$: $v_1 * [\tau']$ and $v_2 * [\tau']$.

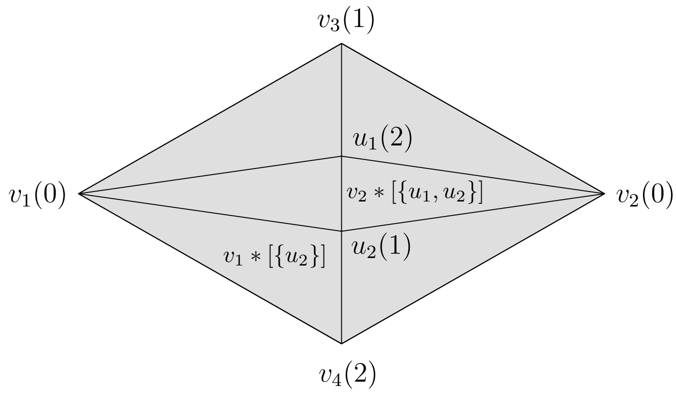

Figure 6.14: A 2-dimensional double chromatic cone.

Figure 6.14 shows an example of a 2-dimensional double chromatic cone. The *id* of every vertex is listed in parentheses, and $\sigma_1 = \{v_1, v_3, v_4\}$, $\sigma_2 = \{v_2, v_3, v_4\}$ and $\tau = \{u_1, u_2\}$. Observe that $\{u_2\}$ is the maximal face of τ that is contained in the simplex $\gamma = \{v_4, u_2\}$ of the basic chromatic subdivision $\tau \circledast \sigma_{1,2}$. Hence γ is denoted $[\{u_2\}]$. Therefore, the simplex $\{v_1, v_4, u_2\}$ of $v_1 * v_2 * (\tau \circledast bd(\sigma_{1,2}))$ is denoted $v_1 * [\{u_2\}]$. Something similar is done with $\{v_1, u_1, u_2\}$. Notice that once $v_1 * v_2 * (\tau \circledast bd(\sigma_{1,2}))$ is constructed, $\sigma_{1,2}$ is no longer a simplex, but sometimes we refer to $\sigma_{1,2}$ as a set of vertexes, for ease of notation.

For the rest of the chapter, for a properly colored simplex σ and $S = \{i_1, i_2 \ldots i_j\} \subseteq id(\sigma)$, let σ_{+S} denote the face of σ with *id*'s S and σ_{-S} denote the face of σ without *id*'s S. If $S = \{i\}$ then we just write σ_{+i} and σ_{-i}. For example, $(\sigma_{1,2})_{-id(\tau')}$ denotes the face of $\sigma_{1,2}$ without *id* colors $id(\tau')$, and $(\sigma_{1,2})_{+id(\tau')\setminus\{k\}}$ is the face of $\sigma_{1,2}$ with *id* colors $id(\tau')$ but k. Similarly, $0(\tau)_{-S}$ represents the simplex containing all the 0 binary colored vertexes of τ minus the vertexes with *id* colors S.

Lemma 6.3.9 *Let $\sigma_1 \xrightarrow{k} \sigma_2$ be an n-path, τ be a properly colored $(n-1)$-simplex with ID^n_k, and for $i \in \{1,2\}$, let v_i be the unique vertex of $\sigma_i \setminus \sigma_{1,2}$. Also, assume that σ_1, σ_2 and τ have a binary coloring. Consider a non-empty face τ' of τ. The simplex $[\tau']$ of $\tau \circledast \sigma_{1,2}$ is j-monochromatic if and only if τ' is j-monochromatic and $(\sigma_{1,2})_{-id(\tau')}$ is j-monochromatic. Moreover, for $i \in \{1,2\}$, the n-simplex $v_i * [\tau'] \in v_1 * v_2 * (\tau \circledast \sigma_{1,2})$ is j-monochromatic if and only if $[\tau']$ is j-monochromatic and $b(v_i) = j$.*

Proof: Let σ' be the simplex of $bd(\sigma_{1,2})$ such that $[\tau'] = \tau' * \sigma'$. Notice that $(\sigma_{1,2})_{-id(\tau')} = \sigma'$. Therefore if $[\tau']$ is j-monochromatic then τ' and σ' are j-monochromatic and vice versa. Now, if $[\tau']$ is j-monochromatic and $b(v_i) = j$ then $v_i * [\tau'] = v_i * \tau' * \sigma'$ is j-monochromatic and vice versa.

■

Lemma 6.3.10 *Let $\sigma_0 \xrightarrow{k_0} \sigma_1 \xrightarrow{k_1} \sigma_2$ be an n-path, τ be a properly colored $(n-1)$-simplex with $ID^n_{k_1}$, and for $i \in \{1,2\}$, let v_i be the unique vertex of $\sigma_i \setminus \sigma_{1,2}$. Consider a face τ' of τ such that $k_0 \in id(\tau')$. Then $v_1 * [\tau'] \xrightarrow{id(\tau')\setminus\{k\}} \rho \xrightarrow{k} \sigma$ is a path in $v_1 * v_2 * (\tau \circledast \sigma_{1,2})$ whose last n-simplex is $\sigma = \sigma_0$ if and only if $k = k_0$, i.e., $\rho = v_1 * [\tau'_{+k_0}]$.*

Proof: First notice that $v_1 * v_2 * (\tau \circledast \sigma_{1,2})$ does not change the boundary of $\sigma_1 - \sigma_2$. Therefore, if $\sigma = \sigma_0$ then σ_0 must share $\sigma_{0,1}$ with ρ. Notice that $k_0 \notin id(\sigma_{0,1})$. Now, let u_{k_0} be the vertex of τ with id color k_0. By assumption, $u_{k_0} \in \tau'$. Observe that each step of $v_1 * [\tau'] \xrightarrow{id(\tau')\setminus\{k_0\}} \rho$, changes a vertex of $\tau' \setminus \{u_{k_0}\}$ for the vertex of $\sigma_{0,1}$ with the same id color. Therefore $\sigma_{0,1} \subset \rho = v_1 * [u_{k_0}]$. Then the last step $\rho \xrightarrow{k_0} \sigma$ changes u_{k_0} and hence $\sigma = \sigma_0$. For the other direction we have that if $\sigma = \sigma_0$ then $\sigma_{0,1} \subset \rho = v_1 * [\tau'']$ for some face τ'' of τ. Observe that τ'' only can be u_{k_0}.

■

They are now presented the classes of paths used in **EliminatePath**. Let $\mathcal{P} : \sigma_0 - \cdots - \sigma_{j-2} \xrightarrow{k_{j-2}} \sigma_{j-1} \xrightarrow{k_{j-1}} \sigma_j \xrightarrow{k_j = k_{j+1}} \sigma_{j+1} \xrightarrow{k_{j+2}} \sigma_{j+2} \xrightarrow{k_{j+3}} \sigma_{j+3} - \cdots - \sigma_{2q+1}$ be an n-path in standard form, and τ be a properly colored $(n-1)$-simplex with $ID^n_{k_j}$. Also assume that τ has a binary coloring. Consider the unique vertexes v_j and v_{j+1} in $\sigma_j \setminus \sigma_{j,j+1}$ and $\sigma_{j+1} \setminus \sigma_{j,j+1}$, and the subdivision $v_j * v_{j+1} * (\tau \circledast \sigma_{j,j+1})$. Let $\chi(\sigma_j)$ and $\chi(\sigma_{j+1})$ denote the subdivision of the simplexes σ_j and σ_{j+1}, respectively, produced by $v_j * v_{j+1} * (\tau \circledast \sigma_{j,j+1})$.

A path in the *non-crossing* class starts either in an n-simplex of $\chi(\sigma_j)$ and ends at σ_0, or starts in an n-simplex of $\chi(\sigma_{j+1})$ and ends at σ_{2q+1}. Intuitively, these paths are non-crossing because they "do not cross" $\tau \circledast \sigma_{m,m+1}$ of $v_j * v_{j+1} * (\tau \circledast \sigma_{j,j+1})$, hence only contain n-simplexes of either $\chi(\sigma_j)$ or $\chi(\sigma_{j+1})$. Figure 6.15 (a) presents an example of two paths in this class. A path in the *crossing* class starts either in an n-simplex of $\chi(\sigma_j)$ followed by an n-simplex of $\chi(\sigma_{j+1})$, and ends at σ_{2q+1}, or starts in an n-simplex of $\chi(\sigma_{j+1})$ followed by an n-simplex of $\chi(\sigma_j)$, and ends at σ_0. The path $\mathcal{P}_2$ in Figure 6.15 (b) is an example of a path in this class. A path in any of these two classes is *left* if it ends at σ_0, or is *right* if it ends at σ_{2q+1}. Formally, these classes are defined as follows.

Definition 6.3.11 *The* non-crossing *path for a face τ' of τ such that $k_{x+\xi} \in id(\tau')$ is*

$$\mathcal{P}_{nc} :\; v_x * [\tau'] \xrightarrow{id(\tau')} \sigma_{x+\xi} \xrightarrow{k_{x+2\xi}} \sigma_{x+2\xi} \xrightarrow{k_{x+3\xi}} \sigma_{x+3\xi} - \cdots$$

where either (i) $x = j$ and $\xi = -1$, or (ii) $x = j+1$ and $\xi = +1$.

The crossing *path for a face τ' of τ such that $k_{x+2\xi} \in id(\tau')$ is*

$$\mathcal{P}_c :\; v_x * [\tau'] \xrightarrow{k_{x+\xi}} v_{x+\xi} * [\tau'] \xrightarrow{id(\tau')} \sigma_{x+2\xi} \xrightarrow{k_{x+3\xi}} \sigma_{x+3\xi} - \cdots$$

where either (i) $x = j$ and $\xi = +1$, or (ii) $x = j+1$ and $\xi = -1$.

Notice that if $\mathcal{P}_{nc}$ satisfies (i) then it is left, otherwise $\mathcal{P}_{nc}$ is right. Similarly, if $\mathcal{P}_c$ satisfies (i) then it is right, otherwise $\mathcal{P}_c$ is left. Lemma 6.3.10 implies that path $\mathcal{P}_{nc} : v_x * [\tau'] \xrightarrow{id(\tau')\setminus\{k_{x+\xi}\}} \lambda_{nc} \xrightarrow{k_{x+\xi}} \sigma_{x+\xi} \xrightarrow{k_{x+2\xi}} \sigma_{x+2\xi} \cdots$ and $\mathcal{P}_c : v_x * [\tau'] \xrightarrow{k_{x+\xi}} v_{x+\xi} * [\tau'] \xrightarrow{id(\tau)\setminus\{k_{x+2\xi}\}} \lambda_c \xrightarrow{k_{x+2\xi}} \sigma_{x+2\xi} \cdots$, where λ_c and λ_{nc} are n-simplexes of $v_j * v_{j+1} * (\tau \circledast \sigma_{j,j+1})$ with an $(n-1)$-face belonging to

$bd(v_j * v_{j+1} * (\tau \circledast \sigma_{j,j+1})) = bd(\sigma_j - \sigma_{j+1})$. This is the reason why Definition 6.3.11 assumes that $k_{x+\xi} \in id(\tau')$ for $\mathcal{P}_{nc}$ and $k_{x+2\xi} \in id(\tau')$ for $\mathcal{P}_c$.

Lemma 6.3.12 below will be the basis for the correctness proof of **EliminatePath** in Section 6.3.4. This lemma uses the following definition. Consider a path $\mathcal{P} : \sigma_0 - \sigma_1 - \cdots - \sigma_{2q+1}$ in standard form. If in addition σ_1 is 0-monochromatic, or σ_{q-1} but no both, and $\mathcal{P}$ has no other monochromatic n-simplex, we say that $\mathcal{P}$ is in *quasistandard* form. Notice that in this case $\mathcal{C}(\mathcal{P}) = \pm 1$. The non-standard formed path in Figure 6.6, is in quasistandard form. The proof of Lemma 6.3.12 is postponed a bit.

Lemma 6.3.12 *Consider paths $\mathcal{P}_c$ and $\mathcal{P}_{nc}$ as defined above.*

1. *For $\mathcal{P}_{nc}$, assume $v_x * [\tau']$ is 0-monochromatic and $|\tau'| = j + 2r$ for some $r \in \mathbb{Z}$.*

   (a) If $(\sigma_{j,j+1})_{+id(\tau') \setminus \{k_{x+\xi}\}}$ is 1-monochromatic then $\mathcal{P}_{nc}$ is in standard form.

   (b) If $(\sigma_{j,j+1})_{+id(\tau') \setminus \{k_{x+\xi}\}}$ is not 1-monochromatic but $(\sigma_{j,j+1})_{+id(\tau') \setminus \{k, k_{x+\xi}\}}$ is 1-monochromatic for some $k \in id(\tau')$ then $\mathcal{P}_{nc}$ is in standard or quasistandard form. Moreover, if $\mathcal{P}_{nc}$ is in quasistandard form then $v_j * [\tau'_{-k}]$ is the 0-monochromatic n-simplex of $\mathcal{P}_{nc}$ that is not at the ends of $\mathcal{P}_{nc}$.

   In both cases, if $\mathcal{P}_{nc}$ is left then $|\mathcal{P}_{nc}| = 2(j + r)$ and if $\mathcal{P}_{nc}$ is right then $|\mathcal{P}_{nc}| = |\mathcal{P}| + 2(r - 1)$.

2. *For $\mathcal{P}_c$, assume $v_x * [\tau']$ is 0-monochromatic, $b(v_{x+\xi}) = 1$ and $|\tau'| = j + 2r + 1$ for some $r \in \mathbb{Z}$. Then $\mathcal{P}_c$ is in standard form. And if $\mathcal{P}_c$ is right then $|\mathcal{P}_c| = |\mathcal{P}| + 2r$ and if $\mathcal{P}_c$ is left then $|\mathcal{P}_c| = 2(j + r + 1)$.*

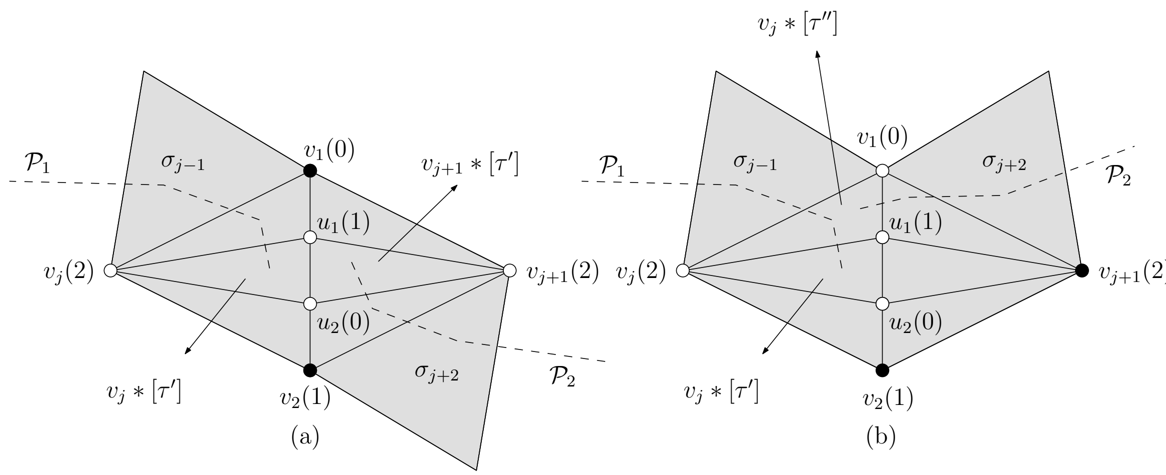

Figure 6.15: Two examples of paths in Lemma 6.3.12.

Figure 6.15 presents an example of the paths mentioned in Lemma 6.3.12. The *id* of every vertex is listed in parentheses. Let us assume $j = 4$. Figure 6.15 (a) contains a left non-crossing path $\mathcal{P}_1$ and a right non-crossing path $\mathcal{P}_2$, both for $\tau' = \{u_1, u_2\}$. For $\mathcal{P}_1$ observe that $k_{x+\xi} = k_{j-1} = 1$ and hence $id(\tau') \setminus \{k_{j-1}\} = \{0\}$. Also $(\sigma_{j,j+1})_{+id(\tau') \setminus \{k_{j-1}\}} = \{v_1\}$ and it is 1-monochromatic. And for $r = -1$, $|\tau'| = j + 2r = 4 - 2 = 2$. Thus, $\mathcal{P}_1$ holds case

1.a of Lemma 6.3.12, hence it is in standard form and $|\mathcal{P}_1| = 2(j+r) = 6$. The idea behind the requirement that $(\sigma_{j,j+1})_{+id(\tau')\setminus\{k_{j-1}\}}$ is 1-monochromatic, is the following. As explained above, $\mathcal{P}_1 : v_j * [\tau'] \xrightarrow{id(\tau')\setminus\{k_{j-1}\}} \lambda_{nc} \xrightarrow{k_{j-1}} \sigma_{j-1} \xrightarrow{k_{j-2}} \sigma_{j-2} \xrightarrow{k_{j-3}} \sigma_{j-3} - \cdots - \sigma_0$, by Lemma 6.3.10. Each step of the subpath $v_j * [\tau'] \xrightarrow{id(\tau')\setminus\{k_{j-1}\}} \lambda_{nc}$ changes a vertex of τ' for a vertex of $(\sigma_{j,j+1})_{+id(\tau')\setminus\{k_{j-1}\}}$. Since $(\sigma_{j,j+1})_{+id(\tau')\setminus\{k_{j-1}\}}$ is 1-monochromatic, each n-simplex of that subpath cannot be 0-monochromatic. Moreover, all these n-simplexes contain at least one vertex of τ', which is 0-monochromatic, and thus they are not 1-monochromatic. Therefore, $v_j * [\tau']$ is the unique monochromatic n-simplex of $v_j * [\tau'] \xrightarrow{id(\tau')\setminus\{k_{j-1}\}} \lambda_{nc}$.

Figure 6.15 (b) presents a left non-crossing path $\mathcal{P}_1$ for $\tau' = \{u_1, u_2\}$ and $k_{x+\xi} = k_{j-1} = 1$. We have $|\tau'| = j+2r$ for $r = -1$. Observe that $(\sigma_{j,j+1})_{+id(\tau')\setminus\{k_{j-1}\}} = \{v_1\}$ is not 1-monochromatic, but $(\sigma_{j,j+1})_{+id(\tau')\setminus\{k,k_{j-1}\}} = \emptyset$ is 1-monochromatic, for $k = 0$. Therefore, $\mathcal{P}_1$ holds case 1.b of Lemma 6.3.12 and hence it is in standard or quasistandard form with $|\mathcal{P}_1| = 2(j+r) = 6$. In this case $\mathcal{P}_1$ is in quasistandard form because its first step changes vertex u_2 for vertex v_1, which is binary colored 0. Figure 6.15 (b) contains a right crossing path $\mathcal{P}_2$ for $\tau'' = \{u_1\}$ and $k_{x+2\xi} = k_{j+2} = 1$. Observe that $|\tau''| = j+2r+1$ for $r = -2$. Path $\mathcal{P}_2$ holds case 2 of Lemma 6.3.12 and thus it is in standard form with $|\mathcal{P}_2| = |\mathcal{P}| + 2r = |\mathcal{P}| - 4$.

Lemma 6.3.12 directly follows from Lemmas 6.3.13 and 6.3.14, which consider the classes of paths $\mathcal{P}_{nc}$ and $\mathcal{P}_c$ introduced in Definition 6.3.11

Lemma 6.3.13 *For $\mathcal{P}_{nc}$, assume $|\tau'| = j+2r$ for some $r \in \mathbb{Z}$. If $\mathcal{P}_{nc}$ is left then $|\mathcal{P}_{nc}| = 2(j+r)$, and if $\mathcal{P}_{nc}$ is right then $|\mathcal{P}_{nc}| = |\mathcal{P}| + 2(r-1)$. For $\mathcal{P}_c$, assume $|\tau'| = j+2r+1$ for some $r \in \mathbb{Z}$. If $\mathcal{P}_c$ is left then $|\mathcal{P}_c| = 2(j+r+1)$, and if $\mathcal{P}_c$ is right then $|\mathcal{P}_c| = |\mathcal{P}| + 2r$.*

Proof: Consider the path $\mathcal{P}_{nc}$. The length of the subpath $v_x * [\tau'] \xrightarrow{id(\tau')} \sigma_{x+\xi}$ is $j+2r+1$ because $|\tau'| = j+2r$. If $\mathcal{P}_{nc}$ is left ($x = j$ and $\xi = -1$) then the length of $\sigma_{x+2\xi} = \sigma_{j-2} - \sigma_{j-3} - \cdots - \sigma_0$ is $j-1$ and thus $|\mathcal{P}_{nc}| = 2(j+r)$. And if $\mathcal{P}_{nc}$ is right ($x = j+1$ and $\xi = 1$) then the length of $\sigma_{x+2\xi} = \sigma_{j+3} - \sigma_{j+4} - \cdots - \sigma_{2q+1}$ is $|\mathcal{P}| - (j+3)$ and hence $|\mathcal{P}_{nc}| = |\mathcal{P}| + 2(r-1)$.

For path $\mathcal{P}_c$, the length of the subpath $v_{x+\xi} * [\tau'] \xrightarrow{id(\tau')} \sigma_{x+2\xi}$ is $j+2r+2$ because $|\tau'| = j+2r+1$. Therefore, the length of $v_x * [\tau'] \xrightarrow{k_{x+\xi}} v_{x+\xi} * [\tau'] \xrightarrow{id(\tau')} \sigma_{x+2\xi}$ is $j+2r+3$. If $\mathcal{P}_c$ is left ($x = j+1$ and $\xi = -1$) then the length of $\sigma_{x+3\xi} = \sigma_{j-2} - \sigma_{j-3} - \cdots - \sigma_0$ is $j-1$ and hence $|\mathcal{P}_c| = 2(j+r+1)$. And if $\mathcal{P}_c$ is right ($x = j$ and $\xi = 1$) then the length of $\sigma_{x+3\xi} = \sigma_{j+3} - \sigma_{j+4} - \cdots - \sigma_{2q+1}$ is $|\mathcal{P}| - (j+3)$ and therefore $|\mathcal{P}_c| = |\mathcal{P}| + 2r$.

■

A path $\mathcal{Q}$ is in *almost standard* or *almost quasistandard* form if it satisfies the conditions of a path in standard or quasistandard form, but $|\mathcal{Q}|$ can be inappropriate. For example, $\mathcal{Q}$ is in almost standard for if the simplexes at its ends are both 0-monochromatic, it does not have any other monochromatic n-simplex and $|\mathcal{Q}|$ is odd. Lemma 6.3.14 below presents necessary conditions for $\mathcal{P}_c$ and $\mathcal{P}_{nc}$ to be in almost standard or almost quasistandard form.

Lemma 6.3.14

1. *For $\mathcal{P}_{nc}$, assume $v_x * [\tau']$ is 0-monochromatic.*

   (a) *If $(\sigma_{j,j+1})_{+id(\tau')\setminus\{k_{x+\xi}\}}$ is 1-monochromatic then $\mathcal{P}_{nc}$ is in almost standard form.*

(b) If $(\sigma_{j,j+1})_{+id(\tau')\backslash\{k_{x+\xi}\}}$ is not 1-monochromatic but $(\sigma_{j,j+1})_{+id(\tau')\backslash\{k,k_{x+\xi}\}}$ is 1-monochromatic for some $k \in id(\tau')$ then $\mathcal{P}_{nc}$ is in almost standard or almost quasistandard form. Moreover, if $\mathcal{P}_{nc}$ is in almost quasistandard form then $v_j * [\tau'_{-k}]$ is the 0-monochromatic n-simplex of $\mathcal{P}_{nc}$ that is not at the ends of $\mathcal{P}_{nc}$.

2. *For $\mathcal{P}_c$, assume $v_x * [\tau']$ is 0-monochromatic and $b(v_{x+\xi}) = 1$. Then $\mathcal{P}_c$ is in almost standard form.*

Proof: Consider case 1. By Lemma 6.3.10, $\mathcal{P}_{nc} : v_x * [\tau'] \xrightarrow{id(\tau')\backslash\{k_{x+\xi}\}} \lambda_{nc} \xrightarrow{k_{x+\xi}} \sigma_{x+\xi} \xrightarrow{k_{x+2\xi}} \sigma_{x+2\xi} - \cdots$. Notice that every n-simplex of the subpath $\mathcal{P}' : v_x * [\tau'] \xrightarrow{id(\tau')\backslash\{k_{x+\xi}\}} \lambda_{nc}$ contains the vertex of τ' with id $k_{x+\xi}$. This vertex is binary colored 0 because $v_x * [\tau']$ is 0-monochromatic. Thus $\mathcal{P}'$ does not contain 1-monochromatic n-simplexes. Also observe that every step in $\mathcal{P}'$ changes a vertex of τ' for a vertex of $(\sigma_{j,j+1})_{+id(\tau')\backslash\{k_{x+\xi}\}}$. In subcase (a), each vertex of $(\sigma_{j,j+1})_{+id(\tau')\backslash\{k_{x+\xi}\}}$ is binary colored 1, hence $\mathcal{P}'$ does not contain 0-monochromatic n-simplexes but $v_x * [\tau']$. Therefore, $\mathcal{P}_{nc}$ is in almost standard form because the subpath $\sigma_{x+\xi} \xrightarrow{k_{x+2\xi}} \sigma_{x+2\xi} - \cdots$ does not contain monochromatic n-simplexes but the last one, which is 0-monochromatic. In subcase (b), $(\sigma_{j,j+1})_{+id(\tau')\backslash\{k_{x+\xi}\}}$ is not 1-monochromatic and for some $k \in id(\tau')$, $(\sigma_{j,j+1})_{+id(\tau')\backslash\{k,k_{x+\xi}\}}$ is 1-monochromatic. This implies that there is a unique $v \in (\sigma_{j,j+1})_{+id(\tau')\backslash\{k_{x+\xi}\}}$ with binary color 0. Observe that if the first step in $\mathcal{P}'$ changes color $id(v)$ then v belong to the second n-simplex $v_x * [\tau'_{-k}]$ of $\mathcal{P}'$. Moreover, this simplex is 0-monochromatic and the other n-simplexes of $\mathcal{P}'$ are not 0-monochromatic but $v_x * [\tau']$. Also if the first step in $\mathcal{P}'$ does not change $id(v)$ then $\mathcal{P}'$ does not contain 0-monochromatic n-simplexes but $v_x * [\tau']$. Therefore, $\mathcal{P}_{nc}$ is in almost standard or almost quasistandard form.

Consider case 2. Lemma 6.3.10 implies $\mathcal{P}_c : v_x * [\tau'] \xrightarrow{k_{x+\xi}} v_{x+\xi} * [\tau'] \xrightarrow{id(\tau)\backslash\{k_{x+2\xi}\}} \lambda_c \xrightarrow{k_{x+2\xi}} \sigma_{x+2\xi} \xrightarrow{k_{x+3\xi}} \sigma_{x+3\xi} - \cdots$. Every n-simplex of the subpath $\mathcal{P}' : v_{x+\xi} * [\tau'] \xrightarrow{id(\tau)\backslash\{k_{x+2\xi}\}} \lambda_c$ contains the vertex of τ' with id $k_{x+2\xi}$, which is binary colored 0, because $v_x * [\tau']$ is 0-monochromatic. Thus $\mathcal{P}'$ does not contain 1-monochromatic n-simplexes. Also each n-simplex of $\mathcal{P}'$ contains vertex $v_{x+\xi}$ because $k_{x+\xi} \notin id(\tau)$. Since $b(v_{x+\xi}) = 1$, $\mathcal{P}'$ does not contain 0-monochromatic n-simplexes. Therefore $v_x * [\tau']$ is the unique monochromatic n-simplex in the subpath $v_x * [\tau'] \xrightarrow{k_{x+\xi}} v_{x+\xi} * [\tau'] \xrightarrow{id(\tau)\backslash\{k_{x+2\xi}\}} \lambda_c$. Since $\sigma_{x+2\xi} \xrightarrow{k_{x+3\xi}} \sigma_{x+3\xi} - \cdots$ does not contain monochromatic n-simplexes but the last one, which is 0-monochromatic, $\mathcal{P}_c$ is in almost standard form.

■

6.3.4 Correctness of EliminatePath

This section shows that algorithm **EliminatePath**, Figure 6.8, produces a complete subdivision of a path in standard form $\mathcal{P}$. **EliminatePath** works as follows. If $\mathcal{P} = \emptyset$ then there is nothing to do. If $|\mathcal{P}| = 2$, then function **SubdivideComp**, Figure 6.16, produces a complete subdivision of $\mathcal{P}$. If $|\mathcal{P}| > 2$, **EliminatePath** finds the subdividing point m of $\mathcal{P}$ and function **SubdivideGood**, Figures 6.17 and 6.18, uses m to produce a good subdivision of $\mathcal{P}$ with paths $\mathcal{P}_i$ such that $|\mathcal{P}_i| \leq |\mathcal{P}|$, $i \in \{1, 2\}$. If the subdividing point m of $\mathcal{P}$ is progressive (see Definition 6.3.7) then for $i \in \{1, 2\}$, $|\mathcal{P}_i| < |\mathcal{P}|$, otherwise it is possible that for some $i \in \{1, 2\}$, $|\mathcal{P}_i| = |\mathcal{P}|$. However, it will be proved that if $|\mathcal{P}_i| = |\mathcal{P}|$ then $\mathcal{P}_i$ has a progressive subdividing point, hence the next recursive call of **EliminatePath** over $\mathcal{P}_i$ will produce a good subdivision with paths of length smaller than $|\mathcal{P}_i| = |\mathcal{P}|$.

Lemma 6.3.15 implies the correctness of **SubdivideComp** and Lemmas 6.3.16 and 6.3.17 give the correctness of **SubdivideGood**. These three lemmas prove the correctness of **EliminatePath**, contained in Lemma 6.3.18. The proofs of Lemmas 6.3.15, 6.3.16 and 6.3.17 are postponed a little bit.

Lemma 6.3.15 *If **SubdivideComp** is invoked with a path $\mathcal{P}$ in standard form then it produces a complete subdivision of $\mathcal{P}$.*

Lemma 6.3.16 *If **SubdivideGood** is invoked with a path $\mathcal{P}$ in standard form and the subdividing point m of $\mathcal{P}$ is progressive then it produces a good subdivision of $\mathcal{P}$ with $|\mathcal{P}_i| < |\mathcal{P}|$, $i \in \{1, 2\}$.*

Lemma 6.3.17 *If **SubdivideGood** is invoked with a path $\mathcal{P}$ in standard form and the subdividing point m of $\mathcal{P}$ is non-progressive then it produces a good subdivision of $\mathcal{P}$ with $|\mathcal{P}_i| \leq |\mathcal{P}|$, $i \in \{1, 2\}$. And if for some $i \in \{1, 2\}$, $|\mathcal{P}_i| = |\mathcal{P}|$ then the subdividing point of $\mathcal{P}_i$ is progressive.*

Lemma 6.3.18 *If **EliminatePath** is invoked with a path $\mathcal{P}$ in standard form then it produces a complete subdivision of $\mathcal{P}$.*

Proof: The proof is by induction on the length of $\mathcal{P}$. If $\mathcal{P} = \emptyset$, clearly the lemma holds. If $|\mathcal{P}| = 2$ then, by Lemma 6.3.15, **EliminatePath** produces a complete subdivision of $\mathcal{P}$. Suppose that the lemma holds for $|\mathcal{P}| \leq 2\ell$. It is proved that it holds for $|\mathcal{P}| = 2(\ell + 1)$.

Line 6 calls **SubdivideGood** with $\mathcal{P}$ and its subdividing point m. If m is progressive, **SubdivideGood** produces a good chromatic of $\mathcal{P}$ with $|\mathcal{P}_i| < |\mathcal{P}|$, $i \in \{1, 2\}$, by Lemma 6.3.16. By induction hypothesis, lines 7 and 8 produce a complete subdivision of $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively. Now, if m is non-progressive, **SubdivideGood** produces a good subdivision of $\mathcal{P}$ with $|\mathcal{P}_i| \leq |\mathcal{P}|$, $i \in \{1, 2\}$, by Lemma 6.3.17. If $\mathcal{P}_i < \mathcal{P}$, line 7 or 8 produces a complete subdivision of $\mathcal{P}_i$, by induction hypothesis. And if $|\mathcal{P}_i| = |\mathcal{P}|$ then the subdividing point of $\mathcal{P}_i$ is progressive, by Lemma 6.3.17. As explained before, **SubdivideGood** produces a complete subdivision of any path of length $2(\ell + 1)$ with subdividing point that is progressive. Hence, line 7 or 8 produce a complete subdivision of $\mathcal{P}_i$.

■

SubdivideComp($\mathcal{P} : \sigma_0 - \sigma_1$)
let $v_0 \leftarrow$ unique vertex of $\sigma_0 \setminus \sigma_{0,1}$
let $v_1 \leftarrow$ unique vertex of $\sigma_1 \setminus \sigma_{0,1}$
let $\tau \leftarrow id$ properly colored $(n-1)$-simplex with $id(\sigma_{0,1})$
$b(\tau) \leftarrow 1$
do $v_0 * v_1 * (\tau \circledast bd(\sigma_{0,1}))$ % double chromatic cone %

Figure 6.16: Function **SubdivideComp**.

Correctness of SubdivideComp

Lemma 6.3.15 (Restated) *If **SubdivideComp** is invoked with a path $\mathcal{P}$ in standard form then it produces a complete subdivision of $\mathcal{P}$.*

Proof: Each n-simplex of $v_0 * v_1 * (\tau \circledast bd(\sigma_{0,1}))$ contains contains either v_0 or v_1. Thus there are no 1-monochromatic n-simplexes in $v_0 * v_1 * (\tau \circledast bd(\sigma_{0,1}))$. Also there are neither 0-monochromatic n-simplexes because each n-simplex contains at least one vertex of τ, which is 1-monochromatic. The boundary is the same because $v_0 * v_1 * (\tau \circledast bd(\sigma_{0,1}))$ only subdivides the shared $(n-1)$-face $\sigma_{0,1}$.

■

SubdivideGood($\mathcal{P}, m$)

$x, \xi \leftarrow$ **ConfigVars($\mathcal{P}, m$)**
let $v_x \leftarrow$ unique vertex of $\sigma_x \setminus \sigma_{x,x+\xi}$
let $v_{x+\xi} \leftarrow$ unique vertex of $\sigma_{x+\xi} \setminus \sigma_{x,x+\xi}$
let $\tau \leftarrow id$ proper colored $(n-1)$-simplex with $id(\sigma_{x,x+\xi})$
$b(\tau_{+id(0(\sigma x,x+\xi))}) \leftarrow 1$
$b(\tau_{+id(1(\sigma x,x+\xi))}) \leftarrow 0$
do $v_x * v_{x+\xi} * (\tau \circledast bd(\sigma_{x,x+\xi}))$ % double chromatic cone %
if case A or B **then**
  $\tau_1 \leftarrow 0(\tau)$
  $\mathcal{P}_1 \leftarrow v_x * [\tau_1] \overset{id(\tau 1)}{\rule{3em}{0.4pt}} \sigma_{x-\xi} \overset{kx-2\xi}{\rule{3em}{0.4pt}} \sigma_{x-2\xi} - \cdots$
  if case A **then**
    $\tau_2 \leftarrow \tau_1$
    $\mathcal{P}_2 \leftarrow v_{x+\xi} * [\tau_2] \overset{id(\tau 2)}{\rule{3em}{0.4pt}} \sigma_{x+2\xi} \overset{kx+3\xi}{\rule{3em}{0.4pt}} \sigma_{x+3\xi} - \cdots$
  else % case B %
    $u_2 \leftarrow$ any vertex of $1(\tau)$
    $b(u_2) \leftarrow 0$
    $\tau_2 \leftarrow \tau_1 \cup \{u_2\}$
    $\mathcal{P}_2 \leftarrow v_x * [\tau_2] \overset{kx+\xi}{\rule{3em}{0.4pt}} v_{x+\xi} * [\tau_2] \overset{id(\tau 2)}{\rule{3em}{0.4pt}} \sigma_{x+2\xi} \overset{kx+3\xi}{\rule{3em}{0.4pt}} \sigma_{x+3\xi} - \cdots$
else if case C, D or case E with $m < n+1$ **then**
  for each $y \in \{x-\xi, x+2\xi\}$ **do if** $k_y \notin id(0(\tau))$ **then**
    $u_y \leftarrow$ vertex of $1(\tau)$ with $id\ k_y$
  else
    $u_y \leftarrow$ any vertex of $1(\tau)$
  $b(u_{x-\xi}) \leftarrow 0$
  $\tau_1 \leftarrow 0(\tau)$
  $\mathcal{P}_1 \leftarrow v_x * [\tau_1] \overset{id(\tau 1)}{\rule{3em}{0.4pt}} \sigma_{x-\xi} \overset{kx-2\xi}{\rule{3em}{0.4pt}} \sigma_{x-2\xi} - \cdots$
  if case C or D **then**
    $\tau_2 \leftarrow \tau_1 \setminus \{u_{x-\xi}\}$
    $\mathcal{P}_2 \leftarrow v_x * [\tau_2] \overset{kx+\xi}{\rule{3em}{0.4pt}} v_{x+\xi} * [\tau_2] \overset{id(\tau 2)}{\rule{3em}{0.4pt}} \sigma_{x+2\xi} \overset{kx+3\xi}{\rule{3em}{0.4pt}} \sigma_{x+3\xi} - \cdots$
    if $\mathcal{P}_1$ and $\mathcal{P}_2$ share n-simplexes **then**
      $\mathcal{P}_1, \mathcal{P}_2 \leftarrow$ **Disconnect($\mathcal{P}_1, \mathcal{P}_2$)**

Figure 6.17: Function **SubdivideGood** (part 1).

(33) **else** % case E with $m < n+1$ %
(34) $b(u_{x+2\xi}) \leftarrow 0$
(33) $\tau_0 \leftarrow \tau_1 \setminus \{u_{x-\xi}\}$
(36) $\tau_2 \leftarrow (\tau_1 \setminus \{u_{x-\xi}\}) \cup \{u_{x+2\xi}\}$
(33) $\mathcal{P}_0 \leftarrow v_x * [\tau_0] \xrightarrow{kx+\xi} v_{x+\xi} * [\tau_0]$
(38) $\mathcal{P}_2 \leftarrow v_{x+\xi} * [\tau_2] \xrightarrow{id(\tau 2)} \sigma_{x+2\xi} \xrightarrow{kx+3\xi} \sigma_{x+3\xi} - \cdots$
(39) **if** $\mathcal{P}_0 \cap \mathcal{P}_1 \neq \emptyset$ **then**
(40) $\mathcal{P}_0, \mathcal{P}_1 \leftarrow$ **Disconnect**$(\mathcal{P}_0, \mathcal{P}_1)$
(41) **if** $\mathcal{P}_0 \cap \mathcal{P}_2 \neq \emptyset$ **then**
(42) $\mathcal{P}_0, \mathcal{P}_2 \leftarrow$ **Disconnect**$(\mathcal{P}_0, \mathcal{P}_2)$
(43) **SubdivideComp**$(\mathcal{P}_0)$
(44) **if** $u_{x-\xi} \neq u_{x+2\xi}$ **then**
(45) $\tau_3 \leftarrow \tau_1 \cup \{u_{x+2\xi}\}$
(46) $\mathcal{Q}_1 : v_x * [\tau_2] \xrightarrow{id(ux+2\xi)} v_x * [\tau_3]$
(47) $\mathcal{Q}_2 : v_{x+\xi} * [\tau_1] \xrightarrow{id(ux-\xi)} v_{x+\xi} * [\tau_3]$
(48) **SubdivideComp**$(\mathcal{Q}_1)$
(49) **SubdivideComp**$(\mathcal{Q}_2)$
(50) **else** % case E with $m = n+1$ %
(51) **if** $k_{x-\xi} \neq k_{x+2\xi}$ **then**
(52) $\lambda_1 \leftarrow \tau_{-kx-\xi}$
(52) $\lambda_2 \leftarrow \tau_{-kx+2\xi}$
(54) $\mathcal{P}_1 \leftarrow v_{x+\xi} * [\tau] \xrightarrow{id(\lambda 1)} v_{x+\xi} * [\tau_{+kx-\xi}] \xrightarrow{kx} v_x * [\tau_{+kx-\xi}] \xrightarrow{kx-\xi} \sigma_{x-\xi} \xrightarrow{kx-2\xi} \sigma_{x-2\xi} - \cdots$
(55) $\mathcal{P}_2 \leftarrow v_x * [\tau] \xrightarrow{id(\lambda 2)} v_x * [\tau_{+kx+2\xi}] \xrightarrow{kx+\xi} v_{x+\xi} * [\tau_{+kx+2\xi}] \xrightarrow{kx+2\xi} \sigma_{x+2\xi} \xrightarrow{kx+3\xi} \sigma_{x+3\xi} - \cdots$
(56) **else**
(57) $\mathcal{P}_1 \leftarrow \cdots - \sigma_{x-\xi} \xrightarrow{kx-\xi} v_x * [\tau_{+kx-\xi}] \xrightarrow{kx+\xi} v_{x+\xi} * [\tau_{+kx-\xi}] \xrightarrow{kx+2\xi} \sigma_{x+2\xi} \xrightarrow{kx+3\xi} \sigma_{x+3\xi} - \cdots$
(58) $\mathcal{P}_2 \leftarrow \emptyset$
(59) **return** $\mathcal{P}_1, \mathcal{P}_2$

Figure 6.18: Function **SubdivideGood** (part 2).

Correctness of SubdivideGood

Considering m, **SubdivideGood** denotes $\mathcal{P}$ as $\sigma_0 - \cdots - \sigma_{m-2} \xrightarrow{k_{m-2}} \sigma_{m-1} \xrightarrow{k_{m-1}} \sigma_m \xrightarrow{k_m = k_{m+1}} \sigma_{m+1} \xrightarrow{k_{m+2}} \sigma_{m+2} \xrightarrow{k_{m+3}} \sigma_{m+3} - \cdots - \sigma_{2q+1}$. Also, it uses function **ConfigVars**, line 1, specified as follows, according to the cases defined in Lemma 6.3.6.

Definition 6.3.19 *Function **ConfigVars** outputs $m-1$ for x only if m holds case B, $m+1$ only if m holds case D, otherwise it outputs m. And for ξ, it outputs -1 only if m holds case D, otherwise it outputs $+1$.*

First, two examples of how **SubdivideGood** works are presented.

Two Examples. Figure 6.19 shows the subdivision produced by **SubdivideGood** for a 2-dimensional path $\mathcal{P}$ with subdividing point $m = 1$ holding case A. By specification of **ConfigVars**,

$x = m = 1$ and $\xi = 1$. Therefore $\sigma_x = \sigma_1$ and $\sigma_{x+\xi} = \sigma_2$, see Figure 6.19 (a). Lines 6 and 7 give a binary coloring to τ such that for each pair $u \in \tau$ and $v \in \sigma_{x,x+\xi} = \sigma_{1,2}$ with $id(v) = id(u)$, u has binary coloring 0 if and only if v has binary color 1. Figure 6.19 (b) shows $\mathcal{P}$ after the double chromatic cone in line 8. Observe that path $\mathcal{P}_1$ in line 11 is a left non-crossing path starting on $v_x * [\tau_1]$, and $\mathcal{P}_2$ in line 14 is a right non-crossing path starting on $v_{x+\xi} * [\tau_2]$. The subdividing point m is progressive because it holds case A, hence we have that $|\mathcal{P}_1|, |\mathcal{P}_2| < |\mathcal{P}|$.

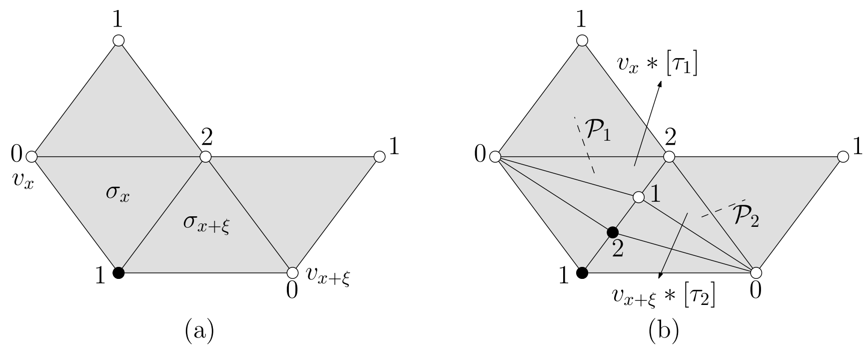

Figure 6.19: **SubdivideGood** working on a 2-dimensional path with subdividing point holding case A.

Figure 6.20 contains an example of **SubdivideGood** working on a 2-dimensional path $\mathcal{P}$ with subdividing point $m = 2$ holding case D. **ConfigVars** outputs $x = m + 1 = 3$ and $\xi = -1$. Therefore $\sigma_x = \sigma_3$ and $\sigma_{x+\xi} = \sigma_2$, see Figure 6.20 (a). Also observe that $k_{x-\xi} = 2$ (recall that $k_{x-\xi}$ is the *id* color changed in the step $\sigma_x - \sigma_{x-\xi}$). As in the previous example, lines 6 and 7 gives a binary coloring to τ such that for each pair $u \in \tau$ and $v \in \sigma_{x,x+\xi} = \sigma_{2,3}$ with $id(v) = id(u)$, u has binary coloring 0 if and only if v has binary color 1. Therefore, $0(\tau)$ in line 21 contains the vertex u_0 of τ with *id* 2, and $1(\tau)$ contains the vertex u_1 of τ with *id* 1. Now, since $k_{x-\xi} = 2$, line 21 is false for $y = x - \xi$, thus line 22 picks any vertex of $1(\tau)$. In this case it must be $u_{x-\xi} = u_1$. After line 25, $0(\tau)$ contains u_1 in addition to u_0. Thus $\tau_1 = \{u_0, u_1\}$ and $\tau_2 = \{u_0\}$, lines 26 and 29. Path $\mathcal{P}_1$ in line 27 is a right non-crossing path starting on $v_x * [\tau_1]$, and path $\mathcal{P}_2$ in line 30 is a left crossing path starting on $v_x * [\tau_2]$. Since m is progressive, $|\mathcal{P}_1|, |\mathcal{P}_2| < |\mathcal{P}|$. However, $\mathcal{P}_1$ and $\mathcal{P}_2$ are not disjoint in the sense that they share an n-simplex, the first n-simplex of $\mathcal{P}_2$. This problem is handled by Function **Disconnect**, Figure 6.23. It applies a second subdivision in order to produce two disjoint paths. Figure 6.21 shows how **Disconnect** solves this problem on dimension 2. This situation is formalized as follows.

Consider distinct paths $\mathcal{P} : \sigma_0 - \sigma_1 - \sigma_2 - \cdots$ and $\mathcal{P}' : \sigma'_0 - \sigma'_1 - \sigma'_2 - \cdots$ in standard and quasistandard form, respectively, with $\sigma_0 = \sigma'_1$. A *good subdivision of* $\mathcal{P} \cup \mathcal{P}'$ is a chromatic subdivision $\chi(\mathcal{P} \cup \mathcal{P}')$ such that it contains two disjoint paths $\mathcal{Q}$ and $\mathcal{Q}'$ in standard form, it has no other monochromatic n-simplex and $bd(\mathcal{P} \cup \mathcal{P}') = bd(\chi(\mathcal{P} \cup \mathcal{P}'))$. For clarity of presentation, the proof of Lemma 6.3.20 is presented later.

Lemma 6.3.20 *If **Disconnect** is invoked with distinct paths $\mathcal{P} : \sigma_0 - \sigma_1 \cdots$ and $\mathcal{P}' : \sigma'_0 - \sigma'_1 \cdots$ in quasistandard and standard form, respectively, with $\sigma_1 = \sigma'_0$, then it produces a good subdivision of $\mathcal{P} \cup \mathcal{P}'$ with paths $\mathcal{Q}$ and $\mathcal{Q}'$ such that $|\mathcal{Q}| = |\mathcal{P}|$ and $|\mathcal{Q}'| = |\mathcal{P}'|$.*

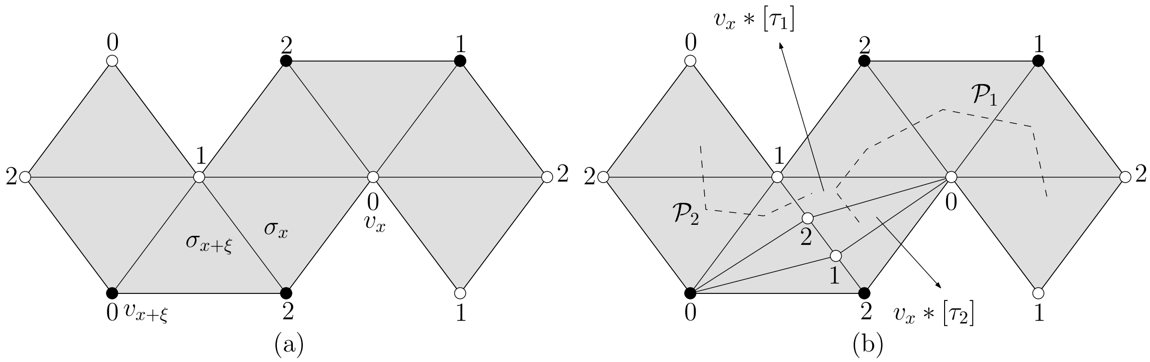

Figure 6.20: **SubdivideGood** working on a 2-dimensional path with subdividing point holding case D.

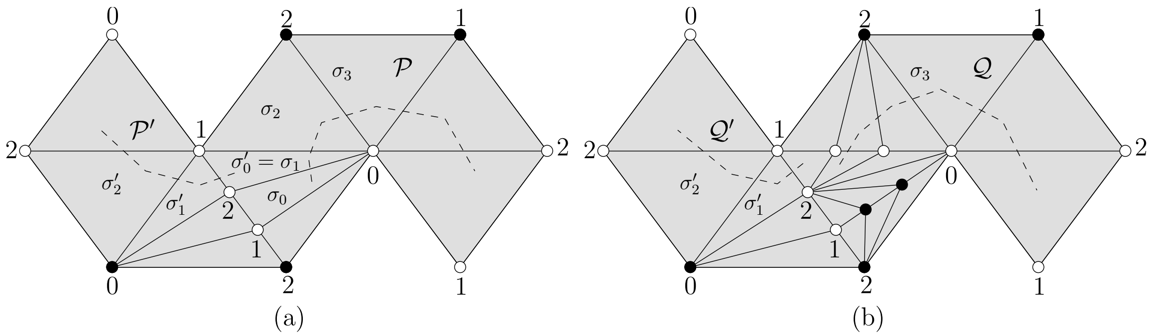

Figure 6.21: An example of function **Disconnect**.

Proof of Lemmas 6.3.16 and 6.3.17. Essentially, the proof of Lemma 6.3.16 consists of using Lemma 6.3.12 to prove that $\mathcal{P}_1$ and $\mathcal{P}_2$ are disjoint and in standard form, and also they are of length smaller than $|\mathcal{P}|$ or $2(m+1)$, as $|\mathcal{P}| \geq 2(m+1)$ by Lemma 6.3.4. In addition, it proves that, at the end of the execution, the subdivisions done by **SubdivideGood** only produce two 0-monochromatic n-simplexes, the n-simplexes at one of the ends of $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. By Definition 6.3.7, m holds case A, C or D, or case E with $m < n+1$. The proof proceeds case by case. Although the proof of Lemma 6.3.16 is simple, it is long and technical; for clarity of presentation, it is contained in Appendix B. Below it is presented an sketch of the proof for cases A and D.

Lemma 6.3.16 (Restated) *If* ***SubdivideGood*** *is invoked with a path $\mathcal{P}$ in standard form and the subdividing point m of $\mathcal{P}$ is progressive then it produces a good subdivision of $\mathcal{P}$ with $|\mathcal{P}_i| < |\mathcal{P}|$, $i \in \{1, 2\}$.*

Sketch of the proof: Let $\mathcal{DCC}^n$ denote $v_x * v_{x+\xi} * (\tau \circledast bd(\sigma_{x,x+\xi}))$ in line 8.

Case A. By the specification of function **ConfigVars**, Definition 6.3.19, $x = m$ and $\xi = +1$. And since m holds case A, $\#0(\sigma_{x=m}) = \#0(\sigma_{x+\xi=m+1}) = n+1-m$ and $\#0(\sigma_{x=m,x+\xi=m+1}) = n-m$, and hence $b(v_m) = b(v_{m+1}) = 0$ and $\#1(\sigma_{m,m+1}) = m$. After line 7, a vertex of τ with *id*

i is binary colored 0 (1) if and only if the vertex of $\sigma_{m,m+1}$ with id i is binary colored 1 (0). Since $\tau_1 = 0(\tau)$, line 10, $id(\tau_1) = id(0(\tau)) = id(1(\sigma_{m,m+1}))$, and thus $|\tau_1| = m$. Moreover, $(\sigma_{m,m+1})_{-id(\tau_1)}$ is 0-monochromatic. Therefore, $v_m * [\tau_1]$ of $\mathcal{DCC}^n$ is 0-monochromatic, by Lemma 6.3.9. Since m holds case A, $\#0(\sigma_{m-1,m}) = n+1-m$. Thus, $k_{m-1} \in id(1(\sigma_{m,m+1})) = id(\tau_1)$, because $\#0(\sigma_m) = n+1-m$. Moreover, since $id(\tau_1) = id(1(\sigma_{m,m+1}))$, then $(\sigma_{m,m+1})_{+id(\tau_1)}$ is 1-monochromatic and hence $(\sigma_{m,m+1})_{+id(\tau_1)\setminus k_{m-1}}$ is 1-monochromatic. Also, $\mathcal{P}_1$ in line 11 is a left non-crossing path. Therefore, by case 1.a of Lemma 6.3.12, $\mathcal{P}_1$ is in standard form and $|\mathcal{P}_1| = 2m$.

For $\mathcal{P}_2$, line 14, $|\tau_2| = m$ because $\tau_2 = \tau_1$ after line 13. As for $\mathcal{P}_1$, one can see that $v_{m+1} * [\tau_2]$ is 0-monochromatic, $k_{m+2} \in id(1(\sigma_{m,m+1}))$ and $(\sigma_{m,m+1})_{+id(\tau_2)\setminus k_{m+2}}$ is 1-monochromatic. $\mathcal{P}_2$ is a right non-crossing path. By case 1.a Lemma 6.3.12, $\mathcal{P}_2$ is in standard form with $|\mathcal{P}_2| = |\mathcal{P}| - 2$. Observe that $\mathcal{P}_1$ and $\mathcal{P}_2$ do not share n-simplexes. Also, it is not hard to check that $v_m * [\tau_1]$ and $v_{m+1} * [\tau_2]$ are the unique monochromatic n-simplexes of $\mathcal{DCC}^n$. Finally, since $\sigma_{m,m+1} \notin bd(\mathcal{P})$, $\mathcal{DCC}^n$ does not affect $bd(\mathcal{P})$.

Case D. By Definition 6.3.19, $x = m+1$ and $\xi = -1$. Also, since m holds case D, $\#0(\sigma_{m+1}) = n+2-m$ and $\#0(\sigma_m) = \#0(\sigma_{m,m+1}) = n+1-m$, and hence $b(v_{m+1}) = 0$, $b(v_m) = 1$ and $\#1(\sigma_{m,m+1}) = m-1$. For $i \in \{0,1\}$, let $\lambda_i = i(\tau)$ in line 8. Observe that $id(\lambda_0) = id(1(\sigma_{m,m+1}))$ and $id(\lambda_1) = id(0(\sigma_{m,m+1}))$. Therefore, $|\lambda_0| = m-1$. Also, $\lambda_1 \neq \emptyset$ because $2 \leq m \leq n$ for case D, by Lemma 6.3.8. Since m holds case D, $\#0(\sigma_{m+1,m+2}) \in \{n+2-m, n+1-m\}$. It is not hard to check that, since $id(\lambda_0) = id(1(\sigma_{m,m+1}))$ and $\#0(\sigma_{m+1}) = n+2-m$, we have that if $\#0(\sigma_{m+1,m+2}) = n+2-m$ then $k_{m+2} \in id(\lambda_0)$, otherwise $k_{m+2} \notin id(\lambda_0)$. After line 26, $\tau_1 = \lambda_0 \cup \{u_{m+2}\}$ and $k_{m+2} \in id(\tau_1)$ (for case D we are not interested in u_{m-1}). Also $|\tau_1| = m$ and τ_1 is 0-monochromatic. Thus, $v_{m+1} * [\tau_1]$ of $\mathcal{DCC}^n$ is 0-monochromatic, by Lemma 6.3.9 and because $(\sigma_{m,m+1})_{-id(\tau_1)}$ is 0-monochromatic. $\mathcal{P}_1$ in line 27 is a right non-crossing path starting on $v_{m+1} * [\tau_1]$. It can be easily proved that if line 21 is true for $k_y = k_{m+2}$, i.e., $k_{m+2} \notin id(\lambda_0)$, then $(\sigma_{m,m+1})_{+id(\tau_1)\setminus k_{m+2}}$ is 1-monochromatic, and hence it can be applied case 1.a of Lemma 6.3.12 on $\mathcal{P}_1$. Thus, $\mathcal{P}_1$ is in standard form and $|\mathcal{P}_1| = |\mathcal{P}|-2$. Also, it can be proved that if line 21 is false for $k_y = k_{m+2}$ then $(\sigma_{m,m+1})_{+id(\tau_1)\setminus k_{m+2}}$ is not 1-monochromatic but $(\sigma_{m,m+1})_{+id(\tau_1)\setminus\{k_{m+2}, id(u_{m+2})\}}$ is 1-monochromatic. By case 1.b of Lemma 6.3.12, $\mathcal{P}_1$ is in standard or quasistandard form and $|\mathcal{P}_1| = |\mathcal{P}|-2$. Also, if $\mathcal{P}_1$ is in quasistandard form then $v_{m+1}*[\tau_{1-id(u_{m+2})}]$ is the monochromatic n-simplex of $\mathcal{P}_1$ that is not at its ends. Notice that $\tau_{1-id(u_{m+2})} = \lambda_0$. As we shall see, $v_{m+1} * [\tau_{1-id(u_{m+2})}]$ is one of the monochromatic n-simplexes at the ends of $\mathcal{P}_2$, line 30.

For $\mathcal{P}_2$ we have that $\tau_2 = \lambda_0$, line 29, hence $|\tau_2| = m-1$, because $id(\lambda_0) = id(1(\sigma_{m,m+1}))$. Moreover, τ_2 and $(\sigma_{m,m+1})_{-id(\tau_2)}$ are 0-monochromatic. By Lemma 6.3.9, $v_{m+1} * [\tau_2]$ of $\mathcal{DCC}^n$ is 0-monochromatic. Also, since m holds case D, $\#0(\sigma_{m-1,m}) = n+1-m$, and thus $k_{m-1} \in id(\tau_2)$, because $\#0(\sigma_m) = n+1-m$. By case 2 of Lemma 6.3.12, $\mathcal{P}_2$ is in standard form and $|\mathcal{P}_2| = 2m$. Now, notice that $v_{m+1} * [\tau_2] = v_{m+1} * [\tau_{1-id(u_{m+2})}]$. Therefore, if $\mathcal{P}_1$ is in quasistandard form then $\mathcal{P}_1$ and $\mathcal{P}_2$ share $v_{m+1}*[\tau_2]$. In this case, by Lemma 6.3.20, function **Disconnect**, line 32, produces a good subdivision of $\mathcal{P}_1 \cup \mathcal{P}_2$ with boundary $bd(\mathcal{P}_1 \cup \mathcal{P}_2)$ and paths of the same length as $\mathcal{P}_1$ and $\mathcal{P}_2$. Finally, function **Disconnect** and $\mathcal{DCC}^n$ do not affect $bd(\mathcal{P})$.

■

The strategy of the proof of Lemma 6.3.17 is essentially the same as the one used for proving Lemma 6.3.16. However, it also proves that if for some $i \in \{1,2\}$, $|\mathcal{P}_i| = |\mathcal{P}|$ then the subdividing point of $\mathcal{P}_i$ is progressive. By Definition 6.3.7, m holds case B or case E with $m = n+1$. Below it is presented an sketch of the proof for case B. The whole proof appears in Appendix B.

Lemma 6.3.17 (Restated) *If* ***SubdivideGood*** *is invoked with a path $\mathcal{P}$ in standard form and the subdividing point m of $\mathcal{P}$ is non-progressive then it produces a good subdivision of $\mathcal{P}$ with $|\mathcal{P}_i| \leq |\mathcal{P}|$, $i \in \{1, 2\}$. And if for some $i \in \{1, 2\}$, $|\mathcal{P}_i| = |\mathcal{P}|$ then the subdividing point of $\mathcal{P}_i$ is progressive.*

Sketch of the proof: Let $\mathcal{DCC}^n$ denote $v_x * v_{x+\xi} * (\tau \circledast bd(\sigma_{x,x+\xi}))$ in line 8.

Case B. Since m holds case B, we have that $\#0(\sigma_{m-1,m}) = \#0(\sigma_m) = \#0(\sigma_{m,m+1}) = \#0(\sigma_{m+1}) = \#0(\sigma_{m+1,m+2}) = n + 1 - m$, and hence $b(v_m) = b(v_{m+1}) = 1$, where v_m and v_{m+1} are the unique vertexes of $\sigma_m \setminus \sigma_{m,m+1}$ and $\sigma_{m+1} \setminus \sigma_{m,m+1}$, respectively. Therefore, it is not possible to produce 0-monochromatic n-simplexes by subdividing on $\sigma_{m,m+1}$. Figure 6.22 (a) depicts an example of a 2-dimensional path with subdividing point $m = 2$ holding case B. **SubdivideGood** handles this problem by subdividing on $\sigma_{m,m-1}$. By the specification of **ConfigVars**, Definition 6.3.19, $x = m - 1$ and $\xi = +1$. Therefore, $\#0(\sigma_{x=m-1,x+\xi=m}) = n + 1 - m$, $\#0(\sigma_{x+\xi=m}) = n + 1 - m$ and $\#0(\sigma_{x+\xi=m,x+2\xi=m+1}) = n + 1 - m$. It is no hard to prove that $\#0(\sigma_{m-1}) = n + 2 - m$ and $\#0(\sigma_{m-2,m-1}) = n+2-m$. Thus, $\#0(\sigma_{x-\xi=m-2,x=m-1}) = n+2-m$ and $\#0(\sigma_{x=m-1}) = n+2-m$. Notice that $b(v_{m-1}) = 0$ and $b(v_m) = 1$.

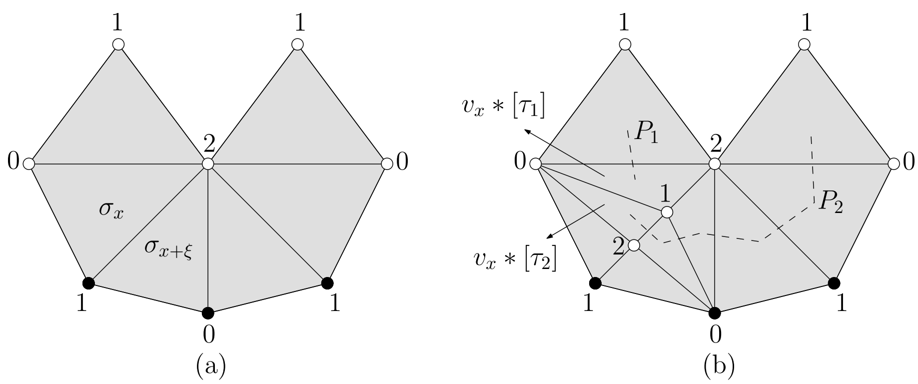

Figure 6.22: **SubdivideGood** working on a 2-dimensional path with subdividing point holding case B.

For $i \in \{0, 1\}$, let $\lambda_i = i(\tau)$ in line 8. Observe that $id(\lambda_0) = id(1(\sigma_{m,m+1}))$ and $id(\lambda_1) = id(0(\sigma_{m,m+1}))$. Therefore, $|\lambda_0| = m - 1$. Also, $\lambda_1 \neq \emptyset$ because $2 \leq m \leq n$, by Lemma 6.3.8. In line 10 we have that $\tau_1 = \lambda_0$. Using a similar argument to the one used in case A in the sketch of the proof of Lemma 6.3.16, it can be proved that $(\sigma_{x,x+\xi})_{-id(\tau_1)}$ is 0-monochromatic, $k_{m-2} \in id(\tau_1)$ and $(\sigma_{m-1,m})_{+id(\tau_1)\setminus k_{m-2}}$ is 1-monochromatic. By Lemma 6.3.9, $v_{m-1} * [\tau_1]$ of $\mathcal{DCC}^n$ is 0-monochromatic. $\mathcal{P}_1$ in line 11 is a left non-crossing path. By case 1.a of Lemma 6.3.12, $\mathcal{P}_1$ is in standard form and $|\mathcal{P}_1| = 2(m - 1)$.

For $\mathcal{P}_2$ in line 19, $|\tau_2| = m$ because $\tau_2 = \tau_1 \cup \{u_2\} = \lambda_0 \cup \{u_2\}$, line 18. Observe that $(\sigma_{m-1,m})_{-id(\tau_2)}$ is 0-monochromatic because $\tau_2 = \lambda_0 \cup \{u_2\}$ and $id(\lambda_0) = id(1(\sigma_{m,m+1}))$. Thus, by Lemma 6.3.9, $v_{m-1} * [\tau_1]$ is 0-monochromatic. Also, since $\#0(\sigma_{m,m+1}) = n + 1 - m$ and $\#0(\sigma_m) = n + 1 - m$, $k_{m+1} \in id(1(\sigma_{m-1,m}))$ and then $k_{m+1} \in id(\tau_2)$. $\mathcal{P}_2$ is a right crossing path. Thus, $\mathcal{P}_2$ is in standard form with $|\mathcal{P}_2| = |\mathcal{P}|$, by case 2 of Lemma 6.3.12. Observe that $\mathcal{P}_1$ and $\mathcal{P}_2$ do not share n-simplexes. Also, it is not hard to check that $v_{m-1} * [\tau_1]$ and $v_{m-1} * [\tau_2]$ are the unique monochromatic n-simplexes of $\mathcal{DCC}^n$, and $\mathcal{DCC}^n$ does not affect $bd(\mathcal{P})$.

For proving that the subdividing point of $\mathcal{P}_2$ is progressive, let $\rho_0 - \rho_1 - \cdots - \rho_m - \rho_{m+1} - \cdots$ denote $\mathcal{P}_2$, where $\rho_0 = v_{m-1} * [\tau_2]$, $\rho_1 = v_m * [\tau_2]$ and for $j \geq m+1$, ρ_j is σ_j of $\mathcal{P}$. Roughly speaking, the proof consists of showing that there is no $m' < m$ such that $\#0(\sigma_{m'+1,m'+2}) \geq n+1-m'$, hence m is the subdividing point of $\mathcal{P}_2$, because m is the subdividing point of $\mathcal{P}$, by hypothesis. Observe that the step from ρ_m to ρ_{m+1} changes the id color k_{m+1} (see Lemma 6.3.10). Also, as explained above, $k_{m+1} \in id(\tau_2)$, and hence the vertex of ρ_m with id k_{m+1} has binary color 0. In addition, at the beginning of the proof it was noticed that the vertex of ρ_{m+1} with id k_{m+1} has binary color 1. Therefore, $\#0(\rho_m) > \#0(\rho_{m+1})$. Thus, m can only hold case either A or C, and then it is progressive (see Figure 6.22 (b)).

■

For completing the correctness proof of **SubdivideGood**, we now prove Lemma 6.3.20 concerning function **Disconnect**, Figure 6.23. For **Disconnect**, let k_i and k'_i be the id's changed in the i-th step of $\mathcal{P}$ and $\mathcal{P}'$, respectively, i.e., $\mathcal{P} : \sigma_0 \xrightarrow{k_0} \sigma_1 \xrightarrow{k_1} \sigma_2 \xrightarrow{k_2} \sigma_3 \cdots$ and $\mathcal{P}' : \sigma'_0 \xrightarrow{k'_0} \sigma'_1 \xrightarrow{k'_1} \sigma'_2 \xrightarrow{k'_2} \sigma'_3 \cdots$.

Disconnect$(\mathcal{P} : \sigma_0 - \sigma_1 \cdots, \mathcal{P}' : \sigma'_0 - \sigma'_1 \ldots)$
let $v_1 \leftarrow$ unique vertex of $\sigma_1 \setminus \sigma_{1,2}$
let $v_2 \leftarrow$ unique vertex of $\sigma_2 \setminus \sigma_{1,2}$
let $\tau \leftarrow id$ properly colored $(n-1)$-simplex with $id(\sigma_{1,2})$
$u_0 \leftarrow$ vertex of τ with id k_0
$u_2 \leftarrow$ vertex of τ with id k_2
$u'_0 \leftarrow$ vertex of τ with id k'_0
$b(\tau) \leftarrow 1$
$b(u_0) \leftarrow 0$
$b(u'_0) \leftarrow 0$
do $v_1 * v_2 * (\tau \circledast bd(\sigma_{1,2}))$ % double chromatic cone %
$\mathcal{Q}' \leftarrow v_1 * [\{u'_0\}] \xrightarrow{k'_0} \sigma'_1 \xrightarrow{k'_1} \sigma'_2 - \cdots$
$\mathcal{R}_1 \leftarrow \sigma_0 \xrightarrow{k_0} v_1 * [\{u_0\}]$
SubdivideComp$(\mathcal{R}_1)$
if $k_0 = k_2$ **or** $k'_0 = k_2$ **then**
    $\mathcal{Q} \leftarrow v_1 * [\{u_0, u'_0\}] \xrightarrow{k_1} v_2 * [\{u_0, u'_0\}] \xrightarrow{\{k_0,k'_0\}-\{k_2\}} \rho \xrightarrow{k_2} \sigma_3 - \cdots$
else
    $b(u_2) \leftarrow 0$
    $\mathcal{Q} \leftarrow v_1 * [\{u_0, u_2\}] \xrightarrow{k_1} v_2 * [\{u_0, u_2\}] \xrightarrow{k_0} v_2 * [\{u_2\}] \xrightarrow{k_2} \sigma_3 - \cdots$
    $\mathcal{R}_2 \leftarrow v_1 * [\{u_0, u'_0, u_2\}] \xrightarrow{k_2} v_1 * [\{u_0, u'_0\}]$
    $\mathcal{R}_3 \leftarrow v_1 * v_1 * [\{u_2, u'_0\}] \xrightarrow{k'_0} [\{u_2\}]$
    SubdivideComp$(\mathcal{R}_2)$
    SubdivideComp$(\mathcal{R}_3)$
return $\mathcal{Q}, \mathcal{Q}'$

Figure 6.23: Function **Disconnect**.

Lemma 6.3.20 (Restated) *If **Disconnect** is invoked with distinct paths $\mathcal{P} : \sigma_0 - \sigma_1 \cdots$ and $\mathcal{P}' : \sigma'_0 - \sigma'_1 \cdots$ in quasistandard and standard form, respectively, with $\sigma_1 = \sigma'_0$, then it produces a good subdivision of $\mathcal{P} \cup \mathcal{P}'$ with paths $\mathcal{Q}$ and $\mathcal{Q}'$ such that $|\mathcal{Q}| = |\mathcal{P}|$ and $|\mathcal{Q}'| = |\mathcal{P}'|$.*

Proof: First, since $\mathcal{P} \cup \mathcal{P}'$ is a pseudomanifold, $k'_0 \neq k_0$ and $k'_0 \neq k_1$, but it is possible $k_2 = k_0$ or $k_2 = k'_0$. Observe that $b(v_1) = 0$, $b(v_2) = 1$ and $\sigma_{1,2}$ is 0-monochromatic. Let $\mathcal{DCC}^n$ be the double chromatic cone in line 10. Consider the execution at line 11. Since $\sigma_{1,2}$ is 0-monochromatic, for each face τ' of τ, $(\sigma_{1,2})_{-id(\tau')}$ is 0-monochromatic. Thus n-simplex $v_2 * [\tau']$ of $\mathcal{DCC}^n$ is not 1-monochromatic, by Lemma 6.3.9. Also, $\{u_0\}$, $\{u'_0\}$ and $\{u_0, u'_0\}$ are the unique 0-monochromatic faces of τ. By Lemma 6.3.9, the n-simplexes $v_1 * [\{u_0\}]$, $v_1 * [\{u'_0\}]$ and $v_1 * [\{u_0, u'_0\}]$ of $\mathcal{DCC}^n$ are 0-monochromatic. Moreover, for any face τ' of τ distinct to $\{u_0\}$, $\{u'_0\}$ and $\{u_0, u'_0\}$, $v_1 * [\tau']$ is not 0-monochromatic because τ' in not 0-monochromatic. Notice that $v_1 * [\{u'_0\}] = \sigma'_{0,1} \cup \{u'_0\}$ and hence $v_1 * [\{u'_0\}]$ and σ'_1 share $\sigma'_{0,1}$. Therefore $\mathcal{Q}'$ in line 11 is a path in standard form with $|\mathcal{Q}'| = |\mathcal{P}'|$. Similarly, $v_1 * [\{u_0\}] = \sigma_{0,1} \cup \{u_0\}$ and hence $v_1 * [\{u_0\}]$ and σ_0 share $\sigma_{0,1}$. Then $\mathcal{R}_1$ in line 12 is a path in standard form. By Lemma 6.3.15, line 13 produces a complete subdivision of $\mathcal{R}_1$.

For $\mathcal{Q}$ there are two cases. If $k_2 = k_0$ or $k_2 = k'_0$ then for $\mathcal{Q}$ in line 15 $\rho = v_2 * [\{u_0\}]$, supposing w.l.o.g. that $k_2 = k_0$. Observe that $v_2 * [\{u_0\}] = \sigma_{2,3} \cup \{u_0\}$ and hence $v_2 * [\{u_0\}]$ and σ_3 share $\sigma_{2,3}$. Thus, $\mathcal{Q}$ is in standard form and $|\mathcal{Q}| = |\mathcal{P}|$. In the other case, $v_1 * [\{u_2\}]$, $v_1 * [\{u_0, u_2\}]$, $v_1 * [\{u_2, u'_0\}]$ and $v_1 * [\{u_0, u_2, u'_0\}]$ are 0-monochromatic, after line 17. Notice that $v_1 * [\{u_2\}] = \sigma_{2,3} \cup \{u_2\}$ and hence $v_2 * [\{u_2\}]$ and σ_3 share $\sigma_{2,3}$. Therefore, $\mathcal{Q}$ in line 18 is in standard form and $|\mathcal{Q}| = |\mathcal{P}|$. It is not hard to see that paths $\mathcal{R}_2$ and $\mathcal{R}_3$ in lines 19 and 20 are in standard form, and lines 21 and 22 produce complete subdivisions of them, by Lemma 6.3.15.

■

6.4 A WSB Protocol

Theorem 3.5.2 says that if there exists a ccosdi of an n-simplex with a rank-symmetric binary coloring and no monochromatic n-simplexes, i.e., it content $\mathcal{C}$ is zero, then there exists a wait-free protocol that solves WSB. Also, by Theorem 6.1.1, for every integers $k_0, k_1 \dots k_{n-1}$ with $k_0 \in \{0, -1\}$, there exists a ccosdi of an n-simplex with a rank-symmetric binary coloring,

$$\mathcal{C} = 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i$$

and with exactly $|\mathcal{C}|$ monochromatic n-simplexes. Thus, if $\mathcal{C} = 0$ then

$$\binom{n+1}{1} k_0 + \binom{n+1}{2} k_1 + \ldots + \binom{n+1}{n} k_{n-1} = 1 \tag{6.1}$$

has an integer solution. It is well-known that for non-zero integers $a_1, \dots, a_j$ and an integer c, if $(a_1, \dots, a_j)$ divides c then there exist integers $x_1, \dots, x_j$ such that $a_1 x_1 + \ldots + a_j x_j = c$ (see for example [68, pp. 301]). Therefore, if $(\binom{n+1}{1}, \dots, \binom{n+1}{n}) = 1$, namely, they are relatively prime, equation (6.1) has a solution on integers and hence $\mathcal{C} = 0$. Notice that the restriction $k_0 \in \{0, -1\}$ is not a problem since $\binom{n+1}{i+1} = \binom{n+1}{n-i}$.

Theorem 6.4.1 (WSB Protocol) *If* $\{\binom{n+1}{i+1} | 0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$ *are relatively prime then there exists an anonymous wait-free protocol that solves WSB.*

For example, consider a prime $p \geq 3$. If $n+1 = 2p$ then $\{\binom{n+1}{i+1} : 0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$ are relatively prime. Just notice that $\binom{2p}{1} = 2p$ and 2 is not factor of $\binom{2p}{2} = p(2p-1)$. Also, it is well-known that $\binom{2p}{p} = \frac{2^p(1*3*5...(2p-1))}{p!} = \frac{2^p(1*3*5...(p-2)*(p+2)...(2p-1))}{(p-1)!}$, hence p is not factor of $\binom{2p}{p}$. Thus, there are an infinite number of cases for which WSB is wait-free solvable.

Corollary 6.4.2 *If* $n+1 = 2p$ *where* $p \geq 3$ *is prime then there exists an anonymous wait-free protocol that solves WSB.*

Since WSB is equivalent to $2n$-renaming (see Section 2.2), Theorem 6.4.1 implies the following renaming upper bound.

Corollary 6.4.3 (Renaming Upper Bound) *If* $\{\binom{n+1}{i+1} | 0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$ *are relatively prime then there exists an anonymous wait-free protocol that solves* $2n$*-renaming.*

Chapter 7

An Algebraic Perspective

This chapter presents the meaning of Theorems 5.1.1 and 6.1.1 in the abstract language of algebraic topology. More precisely, it shows the relation of these theorems with equivariant maps between chain complexes. Generally speaking, a chain complex is an algebraic structure associated with a complex, and an equivariant map is a map between chain complexes that holds certain symmetry properties. The chapter also explains where is the flaw in previous renaming lower bound proofs.

7.1 Algebraic Topology Preliminaries

This section presents a review of some basic notions of algebraic topology that can be found in any textbook such as [64, 33].

Chains. Consider a complex $\mathcal{K}$. A *q-chain* of $\mathcal{K}$ is a formal sum of oriented q-simplexes: $\sum_{j=0} \lambda_j \sigma_j^q$, where λ_j is an integer. For a q-chain, usually simplexes with zero coefficients are omitted, unless they are all zero, and hence the q-chain is denoted 0. We write $1 \cdot \sigma^q$ as σ^q and $-1 \cdot \sigma^q$ as $-\sigma^q$. For $q > 1$, $-\sigma^q$ is identified with σ^q having the opposite orientation. The q-chains of $\mathcal{K}$ form a free Abelian group under the component-wise addition, called the *q-th chain group* of $\mathcal{K}$, denoted $\mathcal{C}_q(\mathcal{K})$. For dimension -1, we adjoin the infinite cyclic group $\mathbb{Z}$, $\mathcal{C}_{-1}(\mathcal{K}) = \mathbb{Z}$.

Boundary Operator. A *boundary* operator $\partial_q : \mathcal{C}_q(\mathcal{K}) \to \mathcal{C}_{q-1}(\mathcal{K})$ is a homomorphisms that satisfies

$$\partial_{q-1}\partial_q\alpha = 0$$

and an *augmentation* $\partial_0 : \mathcal{C}_0(\mathcal{K}) \to \mathcal{C}_{-1}(\mathcal{K})$ which is an epimorphism. For an oriented simplex $\sigma = \langle v_0, v_1, \dots, v_q\rangle$, let σ_j be the $(q-1)$-face of σ without vertex v_j, i.e., $\sigma_j = \langle v_0, \dots, \widehat{v_j}, \dots, v_q\rangle$, where circumflex ($\widehat{\ }$) denotes omission. For $q > 0$, the usual boundary operator $\partial_q : \mathcal{C}_q(\mathcal{K}) \to \mathcal{C}_{q-1}(\mathcal{K})$ is defined on simplexes:

$$\partial_q\sigma = \sum_{j=0}^{q}(-1)^j\sigma_j$$

Boundary ∂_q extends additively to chains: $\partial_q(\alpha+\beta) = \partial_q\alpha + \partial_q\beta$. For $q = 0$, $\partial_0(v) = 1$ and extend linearly. We sometimes omit subscripts from boundary operators.

A q-chain α is a *boundary* if $\alpha = \partial\beta$ for some $(q+1)$-chain β, and it is a *cycle* if $\partial\alpha = 0$. Since $\partial\partial\alpha = 0$, every boundary is a cycle.

Homology Groups. Considering the subgroups $ker(\partial_q)$ and $im(\partial_{q+1})$ of $\mathcal{C}_q(\mathcal{K})$, the *q-th homology group* of $\mathcal{K}$, denoted $\mathcal{H}_q(\mathcal{K})$, is the quotient group

$$\mathcal{H}_q(\mathcal{K}) = ker(\partial_q)/im(\partial_{q+1})$$

If $\mathcal{H}_q(\mathcal{K}) = 0$ for $q \leq \ell$, we say that $\mathcal{K}$ is *ℓ-acyclic*, and if $\mathcal{H}_q(\mathcal{K}) = 0$ for every q, we say that $\mathcal{K}$ is *acyclic*. Two q-cycles α, α' of $\mathcal{C}_q(\mathcal{K})$ are *homologous*, denoted $\alpha \sim \alpha'$, if they belong to the same equivalence class in $\mathcal{H}_q(\mathcal{K})$. Equivalently, $\alpha \sim \alpha'$ if and only if $\alpha - \alpha'$ is a q-boundary.

Chain Maps and Chain Homotopies. The *chain complex* $\mathcal{C}(\mathcal{K})$ of $\mathcal{K}$, is the sequence of groups and homomorphisms $\{\mathcal{C}_q(\mathcal{K}), \partial_q\}$. Let $\{\mathcal{C}_q(\mathcal{K}), \partial_q\}$ and $\{\mathcal{C}_q(\mathcal{L}), \partial'_q\}$ be chain complexes for complexes $\mathcal{K}$ and $\mathcal{L}$. An *augmentation-preserving chain map*, or just *chain map*, ϕ is a family of homomorphisms $\phi_q : \mathcal{C}_q(\mathcal{K}) \to \mathcal{C}_q(\mathcal{L})$, that satisfies $\partial'_q \circ \phi_q = \phi_{q-1} \circ \partial_q$. Therefore, ϕ_q preserves cycles and boundaries. That is, if α is a q-cycle (q-boundary) of $\mathcal{C}_q(\mathcal{K})$, $\phi_q(\alpha)$ is a q-cycle (q-boundary) of $\mathcal{C}_q(\mathcal{L})$. Any simplicial map $\mu : \mathcal{K} \to \mathcal{L}$ induces a chain map $\mu_{\#} : \mathcal{C}(\mathcal{K}) \to \mathcal{C}(\mathcal{L})$. For brevity, μ denotes both the simplicial map and $\mu_{\#}$. Similarly, any subdivision induces a chain map.

Let $\mathcal{K}$ and $\mathcal{L}$ be properly colored complexes. A chain map $\phi : \mathcal{C}(\mathcal{K}) \to \mathcal{C}(\mathcal{L})$ is *color-preserving* if each $\tau \in a(\sigma)$ is properly colored with the colors of σ.

Let ψ and ϕ be chain maps from $\mathcal{C}(\mathcal{K})$ to $\mathcal{C}(\mathcal{L})$. A *chain homotopy* from ψ to ϕ is a family of homomorphisms $\mathcal{D}_q : \mathcal{C}_q(\mathcal{K}) \to \mathcal{C}_{q+1}(\mathcal{L})$, such that $\partial'_{q+1}\mathcal{D}_q + \mathcal{D}_{q-1}\partial_q = \psi_q - \phi_q$.

Equivariant Chain Maps. Let $\mathcal{G}$ be a finite group and $\mathcal{C}(\mathcal{K})$ be a chain complex. An *action* of $\mathcal{G}$ on $\mathcal{C}(\mathcal{K})$ is a set $\Phi = \{\phi_g | g \in \mathcal{G}\}$ of chain maps $\phi_g : \mathcal{C}(\mathcal{K}) \to \mathcal{C}(\mathcal{K})$ such that:

1. For the unit element $e \in \mathcal{G}$, ϕ_e is the identity
2. For all $g, h \in \mathcal{G}$, $\phi_g \circ \phi_h = \phi_{gh}$.

For clarity, we write $g(\sigma)$ instead of $\psi_g(\sigma)$. The pair $(\mathcal{C}(\mathcal{K}), \Phi)$ is a *$\mathcal{G}$-chain complex*. When Φ is understood, we just say that $\mathcal{C}(\mathcal{K})$ is a $\mathcal{G}$-chain complex.

Consider two $\mathcal{G}$-chain complexes $(\mathcal{C}(\mathcal{K}), \Phi)$ and $(\mathcal{C}(\mathcal{L}), \Psi)$. Suppose we have a family of homomorphisms

$$\mu_q : \mathcal{C}_q(\mathcal{K}) \to \mathcal{C}_p(\mathcal{L})$$

possibly $q \neq p$. We say that $\mu = \{\mu_q\}$ is *G-equivariant*, or just *equivariant* when $\mathcal{G}$ is understood, if

$$\mu \circ \phi_g = \psi_g \circ \mu$$

for every $g \in G$. This definition can be extended for the case in which we have a family of homomorphisms for each dimension q, i.e, μ is a family of families of homomorphisms. More formally, for each q suppose we have a family of homomorphisms

$$\mu_q^1, \ldots, \mu_q^{i_q} : \mathcal{C}_q(\mathcal{K}) \to \mathcal{C}_p(\mathcal{L})$$

We say that $\mu = \{\mu_q^{i_q}\}$ is *G-equivariant* if for every $g \in G$ and for every $\mu^i \in \mu$,

$$\mu^j \circ \phi_g = \psi_g \circ \mu^i$$

for some $\mu^j \in \mu$.

7.2 An Equivariance Theorem

This section presents a purely algebraic topology theorem, Theorem 7.2.1, that is strongly related with Theorems 5.1.1 and 6.1.1. This theorem is about a chain map a from the chain complex of a properly colored n-simplex, to the chain complex of the WSB output complex WSB. Theorem 7.2.1 states that if a is equivariant with respect to the symmetric group, it exists if and only if n is exceptional.

Let S_n be the symmetric group consisting all of permutations of ID^n under the composition operation. For $0 \leq i \leq q \leq n$, let π_i^q denote the permutation

$$\pi_i^q = \begin{pmatrix} 0 & \dots & i-1 & i & \dots & q-1 & q & q+1 & \dots & n \\ 0 & \dots & i-1 & i+1 & \dots & q & i & q+1 & \dots & n \end{pmatrix}$$

Consider a simplex σ^n with a coloring id that is proper with ID^n. For brevity, let σ^n denote $\mathcal{M}(\sigma^n)$, the complex consisting all faces of simplex σ^n. Let $\langle i_0 i_1 \dots i_j \rangle$ denote the oriented face of σ^n with colors $i_0, i_1, \dots, i_j$ and with the orientation that contains the sequence $\langle i_0 i_1 \dots i_j \rangle$. Clearly, $\mathcal{C}(\sigma^n)$ is a S_n-chain complex: for each $\pi \in S_n$, $\pi(\langle i_0 i_1 \dots i_j \rangle) = \langle \pi(i_0)\pi(i_1) \dots \pi(i_j) \rangle$.

Consider the output complex $\mathcal{O}^n$ for WSB (see Section 3.2). Each vertex of $\mathcal{O}^n$ has the form (i, b), where $i \in ID^n$ and $b \in \{0, 1\}$. $\mathcal{C}(\mathcal{O}^n)$ is a S_n-chain complex: for each $\pi \in S_n$, $\pi(\langle (i_0, b_0)(i_1, b_1) \dots (i_j, b_j) \rangle) = \langle (\pi(i_0), b_0)(\pi(i_1), b_1) \dots (\pi(i_j), b_j) \rangle$.

Theorem 7.2.1 *There is a color-preserving S_n-equivariant chain map $a : \mathcal{C}(\sigma^n) \to \mathcal{C}(\mathcal{O}^n)$ if and only if $\{\binom{n+1}{i+1} | 0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$ are relatively prime.*

The proof of Theorem 7.2.1 appears in Sections 7.3 and 7.4. First it is presented a result derived from it.

Let $\mathbb{Z}_n$ be the finite cyclic group consisting of ID^n under the addition modulo $n+1$ operation. For each $x \in ID^n$, let $\rho_x : ID^n \to ID^n$ be the permutation defined as $\rho_x(y) = (y+x) \mod (n+1)$. $\mathcal{C}(\sigma^n)$ is a $\mathbb{Z}_n$-chain complex: for each $x \in \mathbb{Z}_n$, $x(\langle i_0 i_1 \dots i_q \rangle) = \langle \rho_x(i_0)\rho_x(i_1) \dots \rho_x(i_q) \rangle$.

Corollary 7.2.2 *If $\{\binom{n+1}{i+1} | 0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$ are relatively prime then there is a $\mathbb{Z}_n$-equivariant chain map $b : \mathcal{C}(\sigma^n) \to \mathcal{C}(\sigma^n)$ such that $b(\partial \sigma^n) = 0$.*

Proof: First, $\mathcal{C}(\mathcal{O}^n)$ is a $\mathbb{Z}_n$-chain complex: for each $x \in \mathbb{Z}_n$, $x(\langle (i_0, b_0)(i_1, b_1) \dots (i_q, b_q) \rangle) = \langle (\rho_x(i_0), b_0)(\rho_x(i_1), b_1) \dots (\rho_x(i_q), b_q) \rangle$. By Theorem 7.2.1, there is a color-preserving S_n-equivariant chain map $a : \mathcal{C}(\sigma^n) \to \mathcal{C}(\mathcal{O}^n)$. It is well known that $\mathbb{Z}_n$ is a subgroup of S_n. Thus, a is $\mathbb{Z}_n$-equivariant. Consider the simplicial map $\mu : \mathcal{O}^n \to \sigma^n$ defined as $\mu((i, b)) = (i + b) \mod (n+1)$ for each vertex $(i, b) \in \mathcal{O}^n$, and consider the chain map induced by μ, denoted also μ. It is not hard to verify that, for every $x \in \mathbb{Z}_n$ and $\tau = \langle (i_0, b_0)(i_1, b_1) \dots (i_q, b_q) \rangle \in \mathcal{C}(\mathcal{O}^n)$, $\mu \circ x(\tau) = x \circ \mu(\tau) = \langle \rho_x(i_0 + b_0)\rho_x(i_1 + b_1) \dots \rho_x(i_q + b_q) \rangle$, hence μ is $\mathbb{Z}_n$-equivariant. Therefore, $\mu \circ a : \mathcal{C}(\sigma^n) \to \mathcal{C}(\sigma^n)$ is $\mathbb{Z}_n$-equivariant. One can prove that μ maps an n-simplex of $\mathcal{C}(\mathcal{O}^n)$ to σ^n if and only if it is monochromatic (the argument is the same that the one used in Lemma 4.1.5). Since $\mathcal{O}^n$ has no monochromatic n-simplexes, $\mu \circ a(\sigma^n) = 0$, hence $\mu \circ a(\partial \sigma^n) = 0$ because $\mu \circ a(\partial \sigma^n) = \partial \mu \circ a(\sigma^n)$.

■

7.3 Necessity

This section presents a proof of the "only if" direction of Theorem 7.2.1, i.e., if n is non-exceptional then there is no color-preserving equivariant chain map $a : \mathcal{C}(\sigma^n) \to \mathcal{C}(\mathcal{O}^n)$. The proof consists of proving that a must map the boundary $\partial\sigma^n$ to a cycle of $\mathcal{C}(\mathcal{O}^n)$ that is not a boundary, which is not possible since chain maps preserve cycles and boundaries. More formally, it proves that $a(\partial\sigma^n) \sim k\alpha$, where α is cycle of $\mathcal{C}(\mathcal{O}^n)$ that is not a boundary and k is a non-zero integer. The proof uses ideas of [55, 56].

In what follows, for distinct $i_0, i_1, \ldots, i_q \in ID^n$, $q \leq n-1$, let $\mathcal{S}^q_{i_0 i_1 \ldots i_q}$ denote the subcomplex of $\mathcal{O}^n$ that contains all q-simplexes, and all its faces, that are properly colored with $i_0, i_1, \ldots, i_q$. It is not hard to see that $\mathcal{S}^q_{i_0 i_1 \ldots i_q}$ is a sphere of dimension q. A *sphere of dimension* q is a complex that has a geometric realization that corresponds to a geometric realization of the boundary of a $q+1$-simplex. Figure 7.1 shows the spheres $\mathcal{S}^0_2$ and $\mathcal{S}^1_{01}$ of the 2-dimensional WSB output complex.

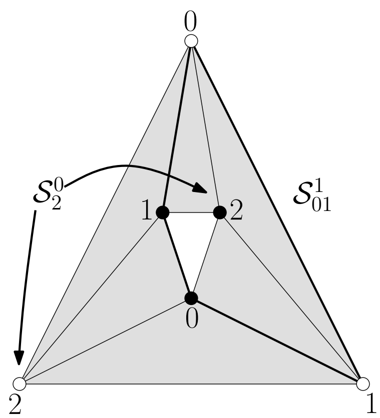

Figure 7.1: The spheres $\mathcal{S}^0_2$ and $\mathcal{S}^1_{01}$ of $\mathcal{O}^2$.

The following lemmas will be useful for proving Lemma 7.3.4 and Theorem 7.3.5 below. Lemma 7.3.1 is a standard result (see textbook [64]). For the rest of the section, let $\partial 0^n$ be the $(n-1)$-cycle of $\mathcal{C}(\mathcal{O}^n)$ defined as

$$\sum_{i=0}^{n} (-1)^i \langle (0,0) \ldots \widehat{(i,0)} \ldots (n,0) \rangle$$

Observe that $\partial 0^n$ is not a boundary.

Lemma 7.3.1 *Let $\mathcal{S}$ be a sphere of dimension n. Then every ℓ-cycle is a boundary, $\ell \leq n-1$.*

Lemma 7.3.2 ([55]) *Let S_i be the cycle obtained by orienting each $(n-1)$-simplex of $\mathcal{S}^{n-1}_{0 \ldots \hat{i} \ldots n}$. Then every $(n-1)$-cycle of $\mathcal{C}(\mathcal{O}^n)$ is homologous to $k\ S_i$ for some integer k.*

Lemma 7.3.3 ([55]) *Let S_i be the cycle obtained by orienting the $(n-1)$-simplexes of $\mathcal{S}^{n-1}_{0 \ldots \hat{i} \ldots n}$ such that its 0-monochromatic $(n-1)$-simplex is oriented in increasing id order. Then $S_i \sim (-1)^i \partial 0^n$.*

Consider the chain map $z : \mathcal{C}(bd(\sigma^n)) \to \mathcal{C}(\mathcal{O}^n)$ that maps each simplex $\langle i_0 \ldots i_j \rangle$ of $\mathcal{C}(bd(\sigma^n))$ to $\langle (i_0, 0) \ldots (i_j, 0) \rangle$. Clearly z is color-preserving and S_n-equivariant. The following lemma is the basis for proving Theorem 7.3.5, which will give the the impossibility of the existence of a. For the rest of the chapter, let $sk^q(\sigma^n)$ denote the complex consisting of all faces of σ^n of dimension at most q.

Lemma 7.3.4 *For each subset s of ID^n there are families of homomorphisms*

$$d_q^s : \mathcal{C}^q(\sigma^n) \to \mathcal{C}^{q+1}(\mathcal{O}^n)$$

$$f_p^s : \mathcal{C}^p(\sigma^n) \to \mathcal{C}^p(\mathcal{O}^n)$$

$-1 \leq q \leq n-2$ *and* $0 \leq p \leq n-1$*, such that* $d = \{d_q^s\}$ *and* $f = \{f_p^s\}$ *are* S_n*-equivariant: For every* $\pi \in S_n$

$$\pi \circ d^s = d^{\pi(s)} \circ \pi$$

$$\pi \circ f^s = f^{\pi(s)} \circ \pi$$

Moreover, for any proper q-face σ of σ^n, the chain

$$a(\sigma) - z(\sigma) - d^{id(\sigma)}(\partial\sigma) - \sum_{\sigma' \in sk^{q-2}(\sigma)} f^{id(\sigma')}(\sigma)$$

is a q-cycle.

Proof: We proceed by induction on the dimension of the faces of σ^n. Unless stated otherwise, $d^s = 0$ and $f^s = 0$. For the rest of the proof let $\sigma_{i_0 i_1 \ldots i_j}$ denote the oriented face $\langle i_0 i_1 \ldots i_j \rangle$ of σ^n.

For dimension 0 it is easy to see that, for each 0-face σ of σ^n, $a(\sigma) - z(\sigma)$ is a 0-cycle. For dimension 1, consider the face σ_0 and the set $\{0, 1\}$. We have that $a(\sigma_0) - z(\sigma_0)$ is a 0-cycle. Since a is color-preserving and by the definition of z, $a(\sigma_0), z(\sigma_0) \in \mathcal{C}(\mathcal{S}_0^0)$. By Lemma 7.3.1 and since $\mathcal{S}_0^0 \subset \mathcal{S}_{01}^1$, there is a 1-chain $d^{01}(\sigma_0) \in \mathcal{C}(\mathcal{S}_{01}^1)$ such that $\partial d^{01}(\sigma_0) = a(\sigma_0) - z(\sigma_0)$. Now, using $d^{01}(\sigma_0)$, we "symmetrically" define the value of d for each pair of 0-face σ and set s of size 2 such that $id(\sigma) \subset s$, namely, $d^{\pi(01)}(\pi(\sigma_0)) = d^s(\sigma) = \pi(d^{01}(\sigma_0))$, where π is a permutation of S_n such that $\sigma = \pi(\sigma_0)$ and $s = \pi(01)$. In this way

$$\begin{aligned} \partial d^s(\sigma) &= \partial \pi(d^{01}(\sigma_0)) = \pi(\partial d^{01}(\sigma_0)) \\ &= \pi(a(\sigma_0) - z(\sigma_0)) = a(\pi(\sigma_0)) - z(\pi(\sigma_0)) = a(\sigma) - z(\sigma) \end{aligned}$$

Observe that $d^s(\sigma) \in \mathcal{C}(\mathcal{S}_s^1)$, and for any $\pi' \in S_n$, $\pi'(d^s(\sigma)) = \pi'(d^{\pi(01)}(\pi(\sigma_0))) = \pi' \circ \pi(d^{01}(\sigma_0)) = d^{\pi' \circ \pi(01)}(\pi' \circ \pi(\sigma_0))$.

For example, for the 0-face σ_1 and the set $\{0, 1\}$, $d^{\pi_0^1(01)}(\pi_0^1(\sigma_0)) = d^{01}(\sigma_1) = \pi_0^1(d^{01}(\sigma_0))$. Observe that the election of $d^{01}(\sigma_0)$ allows to achieve an equivariant d. Thus, we have that $\partial d^{01}(\sigma_0) = a(\sigma_0) - z(\sigma_0)$ and $\partial d^{01}(\sigma_1) = a(\sigma_1) - z(\sigma_1)$, hence

$$\begin{aligned} \partial d^{01}(\sigma_1) - \partial d^{01}(\sigma_0) &= a(\sigma_1) - z(\sigma_1) - (a(\sigma_0) - z(\sigma_0)) \\ \partial d^{01}(\partial \sigma_{01}) &= a(\partial \sigma_{01}) - z(\partial \sigma_{01}) \\ 0 &= \partial(a(\sigma_{01}) - z(\sigma_{01}) - d^{01}(\partial \sigma_{01})) \end{aligned}$$

Thus, $a(\sigma_{01}) - z(\sigma_{01}) - d^{01}(\partial \sigma_{01})$ is a 1-cycle. This complete the basis of the induction, however we present the case for dimension 2 to illustrate the idea.

Consider the face σ_{01} and the set $\{0,1,2\}$. We have proved that $a(\sigma_{01}) - z(\sigma_{01}) - d^{01}(\partial\sigma_{01})$ is a 1-cycle. Moreover, since a and z are color-preserving, and by the previous step, we have that $a(\sigma_{01}), z(\sigma_{01}), d^{01}(\partial\sigma_{01}) \in \mathcal{C}(\mathcal{S}^1_{01})$. By Lemma 7.3.1 and since $\mathcal{S}^1_{01} \subset \mathcal{S}^2_{012}$, there exists a 2-chain $d^{012}(\sigma_{01}) \in \mathcal{C}(\mathcal{S}^2_{012})$ such that $\partial d^{012}(\sigma_{01}) = a(\sigma_{01}) - z(\sigma_{01}) - d^{01}(\partial\sigma_{01})$. Using $d^{012}(\sigma_{01})$, we "symmetrically" define the value of d for all pair of 1-face σ and set s of size 3 such that $id(\sigma) \subset s$. For example, $d^{\pi^2_0(012)}(\pi^2_0(\sigma_{01})) = d^{012}(\sigma_{12}) = \pi^2_0(d^{012}(\sigma_{01}))$ and $d^{\pi^2_1(012)}(\pi^2_1(\sigma_{01})) = d^{012}(\sigma_{02}) = \pi^2_1(d^{012}(\sigma_{01}))$. So we have

$$\begin{aligned}
\partial d^{012}(\sigma_{01}) &= a(\sigma_{01}) - z(\sigma_{01}) - d^{01}(\partial\sigma_{01}) \\
\partial d^{012}(\sigma_{12}) &= a(\sigma_{12}) - z(\sigma_{12}) - d^{12}(\partial\sigma_{12}) \\
\partial d^{012}(\sigma_{02}) &= a(\sigma_{02}) - z(\sigma_{02}) - d^{02}(\partial\sigma_{02})
\end{aligned}$$

Taking the alternating sign sum over $\sigma_{01}, \sigma_{12}, \sigma_{02}$,

$$\begin{aligned}
\partial d^{012}(\sigma_{01}) - d^{012}(\sigma_{02}) + d^{012}(\sigma_{12}) &= +(a(\sigma_{01}) - z(\sigma_{01}) - d^{01}(\partial\sigma_{01})) \\
&\quad -(a(\sigma_{02}) - z(\sigma_{02}) - d^{02}(\partial\sigma_{02})) \\
&\quad +(a(\sigma_{12}) - z(\sigma_{12}) - d^{12}(\partial\sigma_{12})) \\
\partial d^{012}(\partial\sigma_{012}) &= a(\partial\sigma_{012}) - z(\partial\sigma_{012}) - \gamma
\end{aligned}$$

where $\gamma = d^{12}(\partial\sigma_{12}) - d^{02}(\partial\sigma_{02}) + d^{01}(\partial\sigma_{01})$. Thus

$$\partial(a(\sigma_{012}) - z(\sigma_{012}) - d^{012}(\partial\sigma_{012})) - \gamma = 0 \tag{7.1}$$

Now, we have that

$$\begin{aligned}
\gamma &= d^{12}(\partial\sigma_{12}) - d^{02}(\partial\sigma_{02}) + d^{01}(\partial\sigma_{01})) \\
&= (d^{12}(\sigma_2) - d^{12}(\sigma_1)) - (d^{02}(\sigma_2) - d^{02}(\sigma_0)) + (d^{01}(\sigma_1) - d^{01}(\sigma_0))
\end{aligned}$$

Considering the result of the boundary operator over the terms where σ_0 appears, we get

$$\begin{aligned}
\partial(d^{02}(\sigma_0) - d^{01}(\sigma_0)) &= \partial d^{02}(\sigma_0) - \partial d^{01}(\sigma_0) \\
&= a(\sigma_0) - z(\sigma_0) - (a(\sigma_0) - z(\sigma_0)) \\
&= 0
\end{aligned}$$

Thus, $d^{02}(\sigma_0) - d^{01}(\sigma_0)$ is a 1-cycle. The same happens with the terms where σ_1 and σ_2 appear, respectively. Now, by Lemma 7.3.1 and since $d^{01}(\sigma_0) \in \mathcal{C}(\mathcal{S}^1_{01})$ and $d^{02}(\sigma_0) \in \mathcal{C}(\mathcal{S}^1_{02})$, there is a 2-chain $f^0(\sigma_{012}) \in \mathcal{C}(\mathcal{S}^2_{012})$ such that $\partial f^0(\sigma_{012}) = d^{02}(\sigma_0) - d^{01}(\sigma_0)$. The value $f^0(\sigma_{012})$ induces the value of f for all pair of 2-face σ and set s of size 1 such that $s \subset id(\sigma)$. For example, $f^{\pi^2_0(0)}(\pi^2_0(\sigma_{012})) = f^1(\sigma_{120}) = f^1(\sigma_{012}) = \pi^2_0(f^0(\sigma_{012}))$. Observe that $f^1(\sigma_{012}), f^2(\sigma_{012}) \in \mathcal{C}(\mathcal{S}^2_{012})$. Therefore,

$$\gamma = \partial f^0(\sigma_{012}) + \partial f^1(\sigma_{012}) + \partial f^2(\sigma_{012}) = \partial \sum_{\sigma \in sk^0(\sigma_{012})} f^{id(\sigma)}(\sigma_{012}) \tag{7.2}$$

Combining equations (7.1) and (7.2) we get

$$0 = \partial\left(a(\sigma_{012}) - z(\sigma_{012}) - d^{012}(\partial\sigma_{012}) - \sum_{\sigma \in sk^0(\sigma_{012})} f^{id(\sigma)}(\sigma_{012})\right)$$

hence the lemma holds for $n = 2$. Roughly speaking, $f^i(\sigma_{012})$, $i \in \{0,1,2\}$, is what the 0-dimensional face σ_i of σ_{012} adds in obtaining the 2-cycle for σ_{012}.

Assume the lemma holds for faces of dimension at most $q-1$, $0 \leq q \leq n-1$. We prove the lemma holds for faces of dimension q. Also, for each $(q-1)$-dimensional face $\sigma = \sigma_{c_0 \ldots c_{q-1}}$, assume the following.

1. For every $(q-2)$-dimensional face σ' of σ, $d^{id(\sigma)}(\sigma') \in \mathcal{C}(\mathcal{S}^{q-1}_{id(\sigma)})$, and for each ℓ-dimensional face σ' of σ, $\ell \leq q-3$, $f^{id(\sigma')}(\sigma) \in \mathcal{C}(\mathcal{S}^{q-1}_{id(\sigma)})$.
2. For every $(q-2)$-dimensional face σ' of σ,
$$\partial d^{id(\sigma)}(\sigma') = a(\sigma') - z(\sigma') - d^{id(\sigma')}(\partial\sigma') - \sum_{\sigma'' \in sk^{q-4}(\sigma')} f^{id(\sigma'')}(\sigma')$$
3. For every $(q-3)$-dimensional face $\sigma' = \sigma_{c_0 \ldots \widehat{c_i} \ldots \widehat{c_j} \ldots c_{q-1}}$ of σ,
$$\partial f^{id(\sigma')}(\sigma) = (-1)^{i+j} d^{id(\sigma_j)}(\sigma') + (-1)^{i+j-1} d^{id(\sigma_i)}(\sigma')$$
where $\sigma_i = \sigma_{c_0 \ldots \widehat{c_i} \ldots c_{q-1}}$ and $\sigma_j = \sigma_{c_0 \ldots \widehat{c_j} \ldots c_{q-1}}$.
4. For every k-dimensional face σ' of σ, $k \leq q-4$,
$$\partial f^{id(\sigma')}(\sigma) = \sum_{c_i \in id(\sigma), c_i \notin id(\sigma')} (-i)^i f^{id(\sigma')}(\sigma_i)$$
where $\sigma_i = \sigma_{c_0 \ldots \widehat{c_i} \ldots c_{q-1}}$.

Consider the q-simplex $\sigma = \sigma_{0 \ldots q}$. Let σ_i be the $(q-1)$-dimensional face $\sigma_{0 \ldots \hat{i} \ldots q}$ of σ. By induction hypothesis,
$$a(\sigma_i) - z(\sigma_i) - d^{id(\sigma_i)}(\partial\sigma_i) - \sum_{\sigma' \in sk^{q-3}(\sigma_i)} f^{id(\sigma')}(\sigma_i)$$
is a $(q-1)$-cycle. Consider the $(q-1)$-dimensional face σ_q. By induction hypothesis, for each $(q-2)$-dimensional face σ' of σ_q, $d^{0 \ldots q-1}(\sigma') \in \mathcal{C}(\mathcal{S}^{q-1}_{0 \ldots q-1})$, and for each ℓ-dimensional face σ' of σ_q, $\ell \leq q-3$, $f^{id(\sigma')}(\sigma_q) \in \mathcal{C}(\mathcal{S}^{q-1}_{0 \ldots q-1})$. Also, $a(\sigma_q), z(\sigma_q) \in \mathcal{C}(\mathcal{S}^{q-1}_{0 \ldots q-1})$, because a and z are color-preserving. By Lemma 7.3.1 and since $\mathcal{S}^{q-1}_{0 \ldots q-1} \subset \mathcal{S}^{q}_{0 \ldots q}$, there is a q-chain $d^{0 \ldots q}(\sigma_q) \in \mathcal{C}(\mathcal{S}^{q}_{0 \ldots q})$ such that
$$\partial d^{0 \ldots q}(\sigma_q) = a(\sigma_q) - z(\sigma_q) - d^{id(\sigma_q)}(\partial\sigma_q) - \sum_{\sigma' \in sk^{q-3}(\sigma_q)} f^{id(\sigma')}(\sigma_q)$$
Using $d^{0 \ldots q}(\sigma_q)$, we "symmetrically" define the value of $d^s(\sigma') = \pi(d^{0 \ldots q}(\sigma_q))$, where $dim(\sigma') = q-1$, $|s| = q+1$, $id(\sigma') \subset s$, and π is a permutation of S_n such that $\pi(\sigma_q) = \sigma'$ and $\pi(\{0, \ldots, q\}) = s$. Notice that for every $\pi' \in S_n$, $\pi'(d^s(\sigma')) = \pi' \circ \pi(d^{0 \ldots q}(\sigma_q)) = d^{\pi' \circ \pi(0 \ldots q)}(\pi' \circ \pi(\sigma_q))$. Therefore, for each face σ_i of σ
$$\partial d^{0 \ldots q}(\sigma_i) = a(\sigma_i) - z(\sigma_i) - d^{id(\sigma_i)}(\partial\sigma_i) - \sum_{\sigma' \in sk^{q-3}(\sigma_i)} f^{id(\sigma')}(\sigma_i)$$

and $d^{0\ldots q}(\sigma_i) \in \mathcal{C}(\mathcal{S}^q_{0\ldots q})$.

Taking the alternating sign sum over all $(q-1)$-faces of σ, we get

$$\begin{aligned}
\sum_{i=0}^{q}(-1)^i \partial d^{0\ldots q}(\sigma_i) &= \sum_{i=0}^{q}(-1)^i (a(\sigma_i) - z(\sigma_i) - d^{id(\sigma_i)}(\partial\sigma_i) - \sum_{\sigma' \in sk^{q-3}(\sigma_i)} f^{id(\sigma')}(\sigma_i)) \\
\partial d^{0\ldots q}(\partial\sigma) &= a(\partial\sigma) - z(\partial\sigma) - \gamma - \lambda \\
0 &= \partial(a(\sigma) - z(\sigma) - d^{0\ldots q}(\partial\sigma)) - \gamma - \lambda
\end{aligned}$$

where

$$\begin{aligned}
\gamma &= \sum_{i=0}^{q}(-1)^i d^{id(\sigma_i)}(\partial\sigma_i) \\
\lambda &= \sum_{i=0}^{q}(-1)^i \sum_{\sigma' \in sk^{q-3}(\sigma_i)} f^{id(\sigma')}(\sigma_i)
\end{aligned}$$

We now extend d and f such that

$$\partial(a(\sigma) - z(\sigma) - d^{id(\sigma)}(\partial\sigma)) - \gamma - \lambda \tag{7.3}$$

is a q-cycle. Intuitively, we will see that γ and λ are made of $(q-1)$-cycles, hence there are q-chains γ' and λ' such that $\partial\gamma' = \gamma$ and $\partial\lambda' = \lambda$. Combining $\partial\gamma'$ and $\partial\lambda'$ with Equation (7.3), we get $a(\sigma) - z(\sigma) - d^{id(\sigma)}(\partial\sigma) - \gamma' - \lambda'$ is a q-cycle, since we know that $\partial(a(\sigma) - z(\sigma) - d^{id(\sigma)}(\partial\sigma)) - \gamma - \lambda = 0$. As we shall see, γ' and λ' are the q-chains the lemma requires.

First, let us consider γ. Let σ_{ij} denote the $(q-2)$-dimensional face $\sigma_{0\ldots\widehat{i}\ldots\widehat{j}\ldots q}$ of σ. Observe that

$$\begin{aligned}
\partial\gamma = \partial\sum_{i=0}^{q}(-1)^i d^{id(\sigma_i)}(\partial\sigma_i) &= \partial\sum_{i=0}^{q}(-1)^i \left(\sum_{j=0}^{i-1}(-1)^j d^{id(\sigma_i)}(\sigma_{ji}) + \sum_{j=i+1}^{q}(-1)^{j-1} d^{id(\sigma_i)}(\sigma_{ij})\right) \\
&= \sum_{i=0}^{q}\sum_{j=i+1}^{q}(-1)^{i+j}\partial\left(d^{id(\sigma_j)}(\sigma_{ij}) - d^{id(\sigma_i)}(\sigma_{ij})\right)
\end{aligned}$$

By induction hypothesis, $\partial d^{id(\sigma_j)}(\sigma_{ij}) = \partial d^{id(\sigma_i)}(\sigma_{ij})$, thus the $(q-1)$-chain $d^{id(\sigma_j)}(\sigma_{ij}) - d^{id(\sigma_i)}(\sigma_{ij})$ is a cycle. In addition, $d^{id(\sigma_i)}(\sigma_{ij}) \in \mathcal{C}(\mathcal{S}^{q-1}_{id(\sigma_i)})$ and $d^{id(\sigma_j)}(\sigma_{ij}) \in \mathcal{C}(\mathcal{S}^{q-1}_{id(\sigma_j)})$, by induction hypothesis. By Lemma 7.3.1 and since $\mathcal{S}^{q-1}_{id(\sigma_i)}, \mathcal{S}^{q-1}_{id(\sigma_j)} \subset \mathcal{S}^q_{0\ldots q}$, there exists a q-chain $f^{id(\sigma_{ij})}(\sigma) \in \mathcal{C}(\mathcal{S}^q_{0\ldots q})$ such that

$$\partial f^{id(\sigma_{ij})}(\sigma) = (-1)^{i+j}\left(d^{id(\sigma_j)}(\sigma_{ij}) - d^{id(\sigma_i)}(\sigma_{ij})\right)$$

We use $f^{id(\sigma_{ij})}(\sigma)$ to "symmetrically" define the value of $f^s(\sigma')$ for $dim(\sigma') = q$, $|s| = q-1$ and $s \subset id(\sigma')$. So we have

$$\gamma = \partial \sum_{\sigma' \in sk^{q-2}(\sigma), dim(\sigma') = q-2} f^{id(\sigma')}(\sigma) \tag{7.4}$$

Consider now λ. It is not hard to see that

$$\lambda = \sum_{i=0}^{q}(-1)^i \sum_{\sigma' \in sk^{q-3}(\sigma_i)} f^{id(\sigma')}(\sigma_i) = \sum_{\sigma' \in sk^{q-3}(\sigma)} \sum_{i \in [q]-id(\sigma')} (-1)^i f^{id(\sigma')}(\sigma_i)$$

We prove that $\sum_{i\in[q]-id(\sigma')}(-1)^i f^{id(\sigma')}(\sigma_i)$ is a $(q-1)$-cycle. Observe that σ' is a face of σ_i. Fix some $\sigma' \in sk^{q-3}(\sigma)$. We consider two cases, $dim(\sigma') = q-3$ and $dim(\sigma') \le q-4$.

Case $dim(\sigma') = q-3$**.** Assume, without loss of generality, $[q] - id(\sigma') = \{a, b, c\}$ with $a < b < c$. We have that

$$\partial \sum_{i\in[q]-id(\sigma')} (-1)^i f^{id(\sigma')}(\sigma_i) = (-1)^c \partial f^{id(\sigma')}(\sigma_c) + (-1)^b \partial f^{id(\sigma')}(\sigma_b) + (-1)^a \partial f^{id(\sigma')}(\sigma_a)$$

Let σ_{ijk} denote the face $\sigma_{0\ldots\widehat{i}\ldots\widehat{j}\ldots\widehat{k}\ldots q}$ of σ. By induction hypothesis,

$$\begin{array}{rcl}
\partial f^{id(\sigma')}(\sigma_c) & = & (-1)^{a+b} f^{id(\sigma_{bc})}(\sigma_{abc}) + (-1)^{a+b-1} f^{id(\sigma_{ac})}(\sigma_{abc}) \\
\partial f^{id(\sigma')}(\sigma_b) & = & (-1)^{a+c-1} f^{id(\sigma_{bc})}(\sigma_{abc}) + (-1)^{a+c-2} f^{id(\sigma_{ab})}(\sigma_{abc}) \\
\partial f^{id(\sigma')}(\sigma_a) & = & (-1)^{b+c-2} f^{id(\sigma_{ac})}(\sigma_{abc}) + (-1)^{b+c-3} f^{id(\sigma_{ab})}(\sigma_{abc})
\end{array}$$

and thus

$$\begin{array}{rcl}
\partial \sum_{i\in[q]-id(\sigma')} (-1)^i f^{id(\sigma')}(\sigma_i) & = & (-1)^{a+b+c} f^{id(\sigma_{bc})}(\sigma_{abc}) + (-1)^{a+b+c-1} f^{id(\sigma_{ac})}(\sigma_{abc}) \\
& & +(-1)^{a+b+c-1} f^{id(\sigma_{bc})}(\sigma_{abc}) + (-1)^{a+b+c-2} f^{id(\sigma_{ab})}(\sigma_{abc}) \\
& & +(-1)^{a+b+c-2} f^{id(\sigma_{ac})}(\sigma_{abc}) + (-1)^{a+b+c-3} f^{id(\sigma_{ab})}(\sigma_{abc}) \\
& = & 0
\end{array}$$

Therefore, $\sum_{i\in[q]-id(\sigma')}(-1)^i f^{id(\sigma')}(\sigma_i)$ is a $(q-1)$-cycle. By induction hypothesis, $f^{id(\sigma')}(\sigma_i) \in \mathcal{C}(\mathcal{S}^{q-1}_{id(\sigma_i)})$. By Lemma 7.3.1 and since $\mathcal{S}^{q-1}_{id(\sigma_i)} \subset \mathcal{S}^q_{0\ldots q}$, there exists a q-chain $f^{id(\sigma')}(\sigma) \in \mathcal{C}(\mathcal{S}^q_{0\ldots q})$ such that $\partial f^{id(\sigma')}(\sigma) = \sum_{i\in[q]-id(\sigma')}(-1)^i f^{id(\sigma')}(\sigma_i)$. We use $f^{id(\sigma')}(\sigma)$ to "symmetrically" define the value of $f^s(\sigma'')$ for $dim(\sigma'') = q$, $|s| = q-2$ and $s \subset id(\sigma')$. Therefore,

$$\sum_{\sigma'\in sk^{q-3}(\sigma), dim(\sigma')=q-3} \; \sum_{i\in[q]-id(\sigma')} (-1)^i f^{id(\sigma')}(\sigma_i) = \partial \sum_{\sigma'\in sk^{q-3}(\sigma), dim(\sigma')=q-3} f^{id(\sigma')}(\sigma) \tag{7.5}$$

Case $dim(\sigma') \le q-4$**.** By induction hypothesis, for every $i \in [q] - id(\sigma')$,

$$\partial f^{id(\sigma')}(\sigma_i) = \sum_{j=0, j\notin id(\sigma')}^{i-1} (-1)^j f^{id(\sigma')}(\sigma_{ji}) + \sum_{j=i+1, j\notin id(\sigma')}^{q} (-1)^{j-1} f^{id(\sigma')}(\sigma_{ij})$$

Thus

$$\begin{array}{rcl}
\partial \sum_{i\in[q]-id(\sigma')} (-1)^i f^{id(\sigma')}(\sigma_i) & = & \sum_{i\in[q]-id(\sigma')} \sum_{j=0, j\notin id(\sigma')}^{i-1} (-1)^{i+j} f^{id(\sigma')}(\sigma_{ji}) \\
& & + \sum_{i\in[q]-id(\sigma')} \sum_{j=i+1, j\notin id(\sigma')}^{q} (-1)^{i+j-1} f^{id(\sigma')}(\sigma_{ij}) \\
& = & 0
\end{array}$$

Therefore, $\sum_{i\in[q]-id(\sigma')}(-1)^i f^{id(\sigma')}(\sigma_i)$ is a $(q-1)$-cycle. By induction hypothesis, $f^{id(\sigma')}(\sigma_i) \in \mathcal{C}(\mathcal{S}^{q-1}_{id(\sigma_i)})$. By Lemma 7.3.1 and since $\mathcal{S}^{q-1}_{id(\sigma_i)} \subset \mathcal{S}^{q}_{0\dots q}$, there exists a q-chain $f^{id(\sigma')}(\sigma) \in \mathcal{C}(\mathcal{S}^{q}_{0\dots q})$ such that $\partial f^{id(\sigma')}(\sigma) = \sum_{i\in[q]-id(\sigma')}(-1)^i f^{id(\sigma')}(\sigma_i)$. We use $f^{id(\sigma')}(\sigma)$ to "symmetrically" define the value of $f^s(\sigma'')$ for $dim(\sigma'') = q$, $|s| \le q-3$ and $s \subset id(\sigma'')$. Thus, we get

$$\sum_{\sigma'\in sk^{q-4}(\sigma)} \sum_{i\in[q]-id(\sigma')} (-1)^i f^{id(\sigma')}(\sigma_i) = \partial \sum_{\sigma'\in sk^{q-4}(\sigma)} f^{id(\sigma')}(\sigma) \tag{7.6}$$

Combining Equations (7.5) and (7.6)

$$\lambda = \partial \sum_{\sigma'\in sk^{q-3}(\sigma)} f^{id(\sigma')}(\sigma) \tag{7.7}$$

Finally, from Equations (7.3), (7.4) and (7.7), we conclude

$$a(\sigma) - z(\sigma) - d^{id(\sigma)}(\partial\sigma) - \sum_{\sigma'\in sk^{q-2}(\sigma)} f^{id(\sigma')}(\sigma)$$

is a q-cycle, hence the lemma holds for faces of dimension q.

■

Theorem 7.3.5 is the algebraic counterpart of Theorem 5.1.1, which characterizes the number of monochromatic n-simplexes of a ccodi of an n-simplex with a symmetric binary coloring.

Theorem 7.3.5 *Let $a : \mathcal{C}(\sigma^n) \to \mathcal{C}(\mathcal{O}^n)$ be a color-preserving equivariant chain map. For some integers $k_0, \dots, k_{n-1}$,*

$$a(\partial\sigma^n) \sim \left(1 + \sum_{q=0}^{n-1} k_q \binom{n+1}{q+1}\right) \partial 0^n.$$

Proof: Consider the chain map $z : \mathcal{C}(bd(\sigma^n)) \to \mathcal{C}(\mathcal{O}^n)$ that maps each simplex $\langle c_0 \dots c_i\rangle$ to $\langle (c_0, 0) \dots (c_i, 0)\rangle$. Observe that $z(\sigma^n) = \partial 0^n$. Let σ_i denote the oriented face $\langle 0 \dots \widehat{i} \dots n\rangle$ of σ^n. Let S_i be the cycle obtained by orienting the $(n-1)$-simplexes of $\mathcal{S}^{n-1}_{0\dots\widehat{i}\dots n}$ such that its 0-monochromatic $(n-1)$-simplex is oriented in increasing id order. By Lemma 7.3.4,

$$\alpha_i = a(\sigma_i) - z(\sigma_i) - d^{id(\sigma_i)}(\partial\sigma_i) - \sum_{\sigma\in sk^{n-3}(\sigma_i)} f^{id(\sigma)}(\sigma_i)$$

is an $(n-1)$-cycle. Consider the cycle α_n. By Lemma 7.3.2,

$$\alpha_n \sim k_{n-1} S_n$$

for some integer k_{n-1}. It is not hard to see that $\pi^n_i(\sigma_n) = \sigma_i$ and $\pi^n_i(S_n) = S_i$. Thus, $\pi^n_i(\alpha_n) = \alpha_i$, because a, z, d and f are equivariant. Moreover,

$$\pi^n_i(\alpha_n) = \alpha_i \sim k_{n-1}\pi^n_i(S_n) = k_{n-1} S_i$$

and by Lemma 7.3.3

$$\alpha_i \sim (-1)^i k_{n-1} \partial 0^n$$

Considering the alternating sign sum over all $(n-1)$-faces of σ^n, we get

$$\sum_{i=0}^{n}(-1)^i\left(a(\sigma_i)-z(\sigma_i)-d^{id(\sigma_i)}(\partial\sigma_i)-\sum_{\sigma\in sk^{n-3}(\sigma_i)}f^{id(\sigma)}(\sigma_i)\right)\sim\sum_{i=0}^{n}(-1)^i(-1)^ik_{n-1}\partial 0^n$$

hence

$$a(\partial\sigma^n)-z(\partial\sigma^n)-\sum_{i=0}^{n}(-1)^id^{id(\sigma_i)}(\partial\sigma_i)-\sum_{i=0}^{n}(-1)^i\sum_{\sigma\in sk^{n-3}(\sigma_i)}f^{id(\sigma)}(\sigma_i)\sim k_{n-1}(n+1)\partial 0^n$$

And since $z(\partial\sigma^n)=\partial 0^n$

$$a(\partial\sigma^n)\sim\left(1+k_{n-1}(n+1)\right)\partial 0^n+\sum_{i=0}^{n}(-1)^id^{id(\sigma_i)}(\partial\sigma_i)+\sum_{i=0}^{n}(-1)^i\sum_{\sigma\in sk^{n-3}(\sigma_i)}f^{id(\sigma)}(\sigma_i)$$

Notice that if we prove

$$\sum_{i=0}^{n}(-1)^id^{id(\sigma_i)}(\partial\sigma_i)\sim k_{n-2}\binom{n+1}{n-1}\partial 0^n \tag{7.8}$$

and

$$\sum_{i=0}^{n}(-1)^i\sum_{\sigma\in sk^{n-3}(\sigma_i)}f^{id(\sigma)}(\sigma_i)\sim\sum_{q=0}^{n-3}k_q\binom{n+1}{q+1}\partial 0^n \tag{7.9}$$

then

$$a(\partial\sigma^n)\sim\left(1+\sum_{q=0}^{n-1}k_q\binom{n+1}{q+1}\right)\partial 0^n$$

Proof of equation (7.8). For $i,j\in ID^n$ such that $i<j$, let α_{ij} be $(-1)^{i+j}(d^{id(\sigma_j)}(\sigma_{ij})-d^{id(\sigma_i)}(\sigma_{ij}))$, where σ_{ij} is the $(n-2)$-face $\langle 0\ldots\widehat{i}\ldots\widehat{j}\ldots n\rangle$ of σ^n. The proof of Lemma 7.3.4 shows that

$$\sum_{i=0}^{n}(-1)^id^{id(\sigma_i)}(\partial\sigma_i)=\sum_{i=0}^{n}\sum_{j+1}^{n}\alpha_{ij}$$

and α_{ij} is an $(n-1)$-cycle.

Consider $i,j\in ID^n$ such that $i<j<n$. We have that

$$\begin{aligned}\alpha_{ij}&=(-1)^{i+j}(d^{id(\sigma_j)}(\sigma_{ij})-d^{id(\sigma_i)}(\sigma_{ij}))\\ \alpha_{ij+1}&=(-1)^{i+j+1}(d^{id(\sigma_{j+1})}(\sigma_{ij+1})-d^{id(\sigma_i)}(\sigma_{ij+1}))\end{aligned}$$

It is easy to see that $\pi_j^{j+1}(\sigma_{ij})=\sigma_{ij+1}$, $\pi_j^{j+1}(\sigma_j)=\sigma_{j+1}$ and $\pi_j^{j+1}(\sigma_i)=-\sigma_i$. Thus, $\pi_j^{j+1}(\alpha_{ij})=\alpha_{ij+1}$, because d is equivariant. By Lemma 7.3.2, for some integer k_{ij},

$$\alpha_{ij}\sim(-1)^ik_{ij}S_i \tag{7.10}$$

It can be easily proved that $\pi_j^{j+1}(S_i)=-S_i$. Applying π_j^{j+1} on both sides of Equation (7.10) and then multiplying by -1, we get

$$\alpha_{ij+1}\sim(-1)^ik_{ij+1}S_i \tag{7.11}$$

By Lemma 7.3.3 and Equations (7.10) and (7.11), $\alpha_{ij} \sim k_{ij}\partial 0^n$ and $\alpha_{ij+1} \sim k_{ij}\partial 0^n$. A similar analysis gives that, for every $i, j \in ID^n$ such that $i < j-1$, $\alpha_{ij} \sim k_{ij}\partial 0^n$ and $\alpha_{i+1j} \sim k_{ij}\partial 0^n$.

We can repeatedly use these two arguments to prove that $\alpha_{ij} \sim k_{ij}\partial 0^n$ and $\alpha_{i'j'} \sim k_{ij}\partial 0^n$, for every $i, i', j, j' \in ID^n$, $i < j$ and $i' < j'$. Therefore,

$$\sum_{i=0}^{n}\sum_{j+1}^{n}\alpha_{ij} \sim \binom{n+1}{n-1}k_{n-2}\partial 0^n$$

for some integer k_{n-2}.

Proof of equation (7.9). The argument is very similar to the one used for Equation (7.8). The proof of Lemma 7.3.4 shows that

$$\sum_{i=0}^{n}(-1)^i \sum_{\sigma \in sk^{n-3}(\sigma_i)} f^{id(\sigma)}(\sigma_i) = \sum_{\sigma \in sk^{n-3}(\sigma^n)} \sum_{i \in ID^n - id(\sigma)} (-1)^i f^{id(\sigma)}(\sigma_i)$$

Also it shows that $\sum_{i \in ID^n - id(\sigma)}(-1)^i f^{id(\sigma)}(\sigma_i)$ is an $(n-1)$-cycle. For each $\sigma \in sk^{n-3}(\sigma^n)$, let α_σ be the cycle $\sum_{i \in ID^n - id(\sigma)}(-1)^i f^{id(\sigma)}(\sigma_i)$.

Consider $\sigma, \sigma' \in sk^{n-3}(\sigma^n)$ of same dimension such that for some $P \subset ID^n$ and $j \in ID^n$, $id(\sigma) = P \cup \{j\}$, $id(\sigma') = P \cup \{j+1\}$ and $j, j+1 \notin P$. Note

$$\begin{aligned}
\alpha_\sigma &= \sum_{i \in ID^n - id(\sigma)} (-1)^i f^{id(\sigma)}(\sigma_i) = (-1)^{j+1} f^{id(\sigma)}(\sigma_{j+1}) + \sum_{i \in ID^n - P} (-1)^i f^{id(\sigma)}(\sigma_i) \\
\alpha_{\sigma'} &= \sum_{i \in ID^n - id(\sigma')} (-1)^i f^{id(\sigma')}(\sigma_i) = (-1)^{j} f^{id(\sigma')}(\sigma_{j}) + \sum_{i \in ID^n - P} (-1)^i f^{id(\sigma')}(\sigma_i)
\end{aligned}$$

It is easy to see that $\pi_j^{j+1}(\sigma) = \sigma'$, $\pi_j^{j+1}(\sigma_{j+1}) = \sigma_j$ and $\pi_j^{j+1}(\sigma_i) = -\sigma_i$ for each $i \in ID^n - P$. Then, $\pi_j^{j+1}(\alpha_\sigma) = -\alpha_{\sigma'}$, since f is equivariant.

Fix an $i \in ID^n - id(\sigma)$. By Lemma 7.3.2, for some integer k_σ

$$\alpha_\sigma \sim (-1)^i k_\sigma S_i \tag{7.12}$$

It can be easily proved that $\pi_j^{j+1}(S_i) = -S_i$. Applying π_j^{j+1} on both sides of Equation (7.13) and then multiplying by -1, we get

$$\alpha_{\sigma'} \sim (-1)^i k_\sigma S_i \tag{7.13}$$

By Lemma 7.3.3 and Equations (7.12) and (7.13), $\alpha_\sigma \sim k_\sigma\partial 0^n$ and $\alpha_{\sigma'} \sim k_\sigma\partial 0^n$. We can repeatedly use this argument to prove that $\alpha_\sigma \sim k_\sigma\partial 0^n$ and $\alpha_{\sigma'} \sim k_\sigma\partial 0^n$, for every $\sigma, \sigma' \in sk^{n-3}(\sigma^n)$ of same dimension. Therefore,

$$\sum_{\sigma \in sk^{n-3}(\sigma^n)} \sum_{i \in ID^n - id(\sigma)} (-1)^i f^{id(\sigma)}(\sigma_i) \sim \sum_{q=0}^{n-3} k_q \binom{n+1}{q+1}\partial 0^n$$

■

As explained in Section 5.4, if $\{\binom{n+1}{i+1} | 0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$ are not relatively prime, then there are not integers $k_0, \ldots, k_{n-1}$ such that $1 + \sum_{q=0}^{n-1} k_q \binom{n+1}{q+1} = 0$, hence, by Theorem 7.3.5, the boundary $a(\partial \sigma^n)$ is homologous to $k\, \partial 0^n$, where k is a non-zero integer. Since the cycle $\partial 0^n$ is not a boundary, a cannot exist. The following lemma follows.

Lemma 7.3.6 *If $\{\binom{n+1}{i+1} | 0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$ are not relatively prime, then there is no color-preserving equivariant chain map $a : \mathcal{C}(\sigma^n) \to \mathcal{C}(\mathcal{O}^n)$.*

7.4 Sufficiency

This section proves the "if" direction of Theorem 7.2.1, namely, if n is exceptional then there is a color-preserving equivariant chain map $a : \mathcal{C}(\sigma^n) \to \mathcal{C}(\mathcal{O}^n)$. It proves that, for exceptional n, the ccosdi's without monochromatic n-simplexes produced by the construction presented in Chapter 6, induce chain maps like a.

Lemma 7.4.1 *If $\{\binom{n+1}{i+1} | 0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$ are relatively prime, then there is a color-preserving equivariant chain map $a : \mathcal{C}(\sigma^n) \to \mathcal{C}(\mathcal{O}^n)$.*

Proof: As explained in Section 6.4, Theorem 6.1.1 implies that if n is exceptional, then there is a ccosdi $\mathcal{K}^n$ of an n-simplex with a rank-symmetric binary coloring and without monochromatic n-simplexes. Subdivisions induce chain maps, hence $\mathcal{K}^n$ induce a chain map $\mu_1 : \mathcal{C}(\sigma^n) \to \mathcal{C}(\chi(\sigma^n))$. Let id and b the chromatic and binary coloring of $\mathcal{K}^n$. Since $\mathcal{K}^n$ does not have monochromatic n-simplexes, id and b induces a simplicial map $\mathcal{K}^n \to \mathcal{O}^n$, which induces a chain map $\mu_2 : \mathcal{C}(\mathcal{K}^n) \to \mathcal{C}(\mathcal{O}^n)$, because simplicial maps induce chain maps. Let a be the composition of μ_1 and μ_2. Clearly, μ_1 and μ_2 are color-preserving, thus a is color-preserving. In what follows, we prove a is equivariant.

Let $\dot{\mathcal{O}}^n$ be the complex containing $\mathcal{O}^n$ and the monochromatic n-simplexes $\{(0,1), \ldots, (n,0)\}$ and $\{(0,1), \ldots, (n,1)\}$. Observe that a is also a chain map $\mathcal{C}(\sigma^n) \to \mathcal{C}(\dot{\mathcal{O}}^n)$. For technical reasons we think of a in this way. By induction on q, we prove the following proposition.

Proposition 7.4.2 *The restriction $a|\mathcal{C}(sk^q(\sigma^n))$, $0 \leq q \leq n$, is equivariant.*

By symmetry of the binary coloring of $\mathcal{K}^n$, Proposition 7.4.2 clearly holds for $q = 0$. Suppose Proposition 7.4.2 holds for dimension $q - 1$. We prove it holds for dimension q.

Consider two q-simplexes $\sigma, \sigma' \in \mathcal{C}(sk^q(\sigma^n))$ that are oriented in increasing id order.Because the binary coloring of $\mathcal{K}^n$ is rank symmetric, we have that $a \circ \pi'(\sigma) = a(\sigma') = \pi' \circ a(\sigma)$, where π' is any permutation of S_n such that $\sigma' = \pi'(\sigma)$, i.e., it maps the colors of $id(\sigma)$ to the colors of $id(\sigma')$ preserving the order. Notice that if we prove that $\pi \circ a(\sigma) = a \circ \pi(\sigma)$ for every $\pi \in S_n$, then $\pi \circ a(\sigma') = \pi \circ \pi' \circ a(\sigma) = a \circ \pi \circ \pi'(\sigma) = a \circ \pi(\sigma')$.

Suppose, without loss of generality, that $\sigma = \langle 0 \ldots q \rangle$. Let L_q be $\{\tau | \tau \in sk^q(\dot{\mathcal{O}}^n)$ and $id(\tau) = ID^q\}$. For $\tau \in L_q$, let $\#1(\tau)$ be the number of its vertexes with binary color 1, and let $inv(\tau, i)$, $0 \leq i \leq q$, denote the simplex of L_q with the same vertexes as τ but with the vertex with id i having the opposite binary coloring to the binary coloring of the vertex with id i of τ. For $0 \leq k \leq q+1$, let $L_{q,k}$ denote the set $\{\tau | \tau \in L_q$ and $\#1(\tau) = k\}$. Thus $|L_{q,k}| = \binom{q+1}{k}$. Since a is color-preserving, we can write

$$a(\sigma) = \sum_{\tau \in L_q} k_\tau \tau$$

where $k_\tau \in \mathbb{Z}$. Obviously if $q = n$ then $k_{\{(0,0),\ldots,(n,0)\}} = k_{\{(0,1),\ldots,(n,1)\}} = 0$, since $\mathcal{O}^n$ does not have monochromatic n-simplexes. We prove the following proposition.

Proposition 7.4.3 *For every $\tau, \tau' \in L_{q,k}$, $k_\tau = k_{\tau'}$, $0 \leq k \leq q+1$.*

For example, for $\sigma = \langle 012 \rangle$ and $k = 2$, Proposition 7.4.3 says that if $\langle (0,0)(1,1)(2,1) \rangle$ appears in $a(\sigma)$ with coefficient ℓ, then $\langle (0,1)(1,1)(2,0) \rangle$ and $\langle (0,1)(1,0)(2,1) \rangle$ appear in $a(\sigma)$ with coefficient ℓ too. It is not hard to see that this proves $a \circ \pi(\sigma) = \pi \circ a(\sigma)$ for every $\pi \in \mathcal{S}_n$, hence Proposition 7.4.2 holds for q.

We proceed by induction on k. For $k = 0$ we have that $|L_{q,k}| = 1$, thus Proposition 7.4.3 trivially holds. Suppose Proposition 7.4.3 holds for $k-1$. We prove it holds for k.

Notice that

$$\partial a(\sigma) = \sum_{\tau \in L_q} k_\tau \partial \tau = \sum_{\tau \in L_q} k_\tau \sum_{i=0}^{q} (-1)^i \tau_i = a(\partial \sigma) = \sum_{i=0}^{q} (-1)^i a(\sigma_i)$$

where $\tau_i = \langle (0, b_0) \ldots \widehat{(i, b_i)} \ldots (q, b_q) \rangle$ for $\tau = \langle (0, b_0) \ldots (i, b_i) \ldots (q, b_q) \rangle$. Consider a simplex $\tau \in L_q$ and $i \in \{0, \ldots, q\}$. Observe that the $(q-1)$-simplex τ_i appears in $\partial a(\sigma)$ with coefficient $(-1)^i (k_\tau + k_{inv(\tau,i)})$, since τ_i is face of τ and $inv(\tau, i)$. Moreover, τ_i appears in $a(\sigma_i)$ with coefficient $k_\tau + k_{inv(\tau,i)}$, because $\partial a(\sigma) = a(\partial \sigma)$ and a is color-preserving. Also notice that either $\#1(\tau) = \#1(\tau_i)$ and $\#1(inv(\tau,i)) = \#1(\tau_i) + 1$, or $\#1(\tau) = \#1(\tau_i) + 1$ and $\#1(inv(\tau,i)) = \#1(\tau_i)$.

Consider the set $N = \{\tau | \tau \in L_{q,k} \text{ and } \#1(\tau_q) = k-1\}$. Note $|N| = \binom{q}{k-1}$. For each $\tau \in N$, observe that $\#1(inv(\tau,q)) = k-1$, hence $inv(\tau,q) \in L_{q,k-1}$. Consider a simplex $\tau \in N$. As noticed above, τ_q appears in $a(\sigma_q)$ with coefficient $k_\tau + k_{inv(\tau,q)}$. Consider $i \in \{0, \ldots, q\}$. Let ρ_i and ρ be the simplexes $\pi_i^q(\tau_q)$ and $\pi_i^q(\tau)$. Observe that ρ_i is a face of ρ, $\#1(\rho_i) = k-1$ and $\#1(\rho) = k$. As for τ_q, we have that ρ_i appears in $a(\sigma_i)$ with coefficient $k_\rho + k_{inv(\rho,i)}$, where $\sigma_i = \pi_i^q(\sigma_q)$. By induction hypothesis, $a|_{\mathcal{C}(sk^{q-1}(\sigma^n))}$ is equivariant, hence $a \circ \pi_i^q(\sigma_q) = a(\sigma_i) = \pi_i^q \circ a(\sigma_q)$. Therefore, $k_\tau + k_{inv(\tau,q)} = k_\rho + k_{inv(\rho,i)}$. Moreover, $k_{inv(\tau,q)} = k_{inv(\rho,i)}$ because $\#1(inv(\tau,q)) = \#1(inv(\rho,i)) = k-1$ and, by the induction hypothesis, Proposition 7.4.3 holds for $k-1$. Thus, we get $k_\tau = k_\rho$.

For each $\tau \in N$, let M_τ be $\{\pi_i^q(\tau) | 0 \leq i \leq q\}$. The previous paragraph proved that for every $\rho, \rho' \in M_\tau$, $k_\rho = k_{\rho'}$. It is not hard to see that $|M_\tau| = (q+1) - (k-1)$ for every $\tau \in N$, and $L_{q,k} = \cup_{\tau \in N} M_\tau$. Moreover, we have that the sets M_τ's are not a partition of $L_{q,k}$ because

$$\frac{\binom{q+1}{k}}{((q+1)-(k-1))\binom{q}{k-1}} = \frac{q+1}{k((q+1)-(k-1))} < 1$$

Thus, these sets intersect each other, hence $\tau, \tau' \in L_{q,k}$, $k_\tau = k_{\tau'}$. This completes the proof.

■

7.5 Previous Renaming Lower Bound Proofs

Two algebraic renaming lower bound proof stating that M-renaming is not wait-free solvable if $M < 2n+1$, are presented in [55, 56]. This section explains where the flaw is in these proofs.

In [55] it is proved that a wait-free WSB protocol (called reduced renaming in that paper) implies the existence of a color-preserving S_n-equivariant chain map $a : \mathcal{C}(\sigma^n) \to \mathcal{C}(\mathcal{O}^n)$. Then it is proved that a cannot exist for any value of n. Therefore, WSB is not wait-free solvable, hence neither $2n$-renaming.

The proof of impossibility of a in [55], is based on Lemma 6.12 in that paper, which says that that there are homomorphisms $d_q : \mathcal{C}_q(\sigma^n) \to \mathcal{C}_{q+1}(\mathcal{O}^n)$, $-1 \leq q \leq n-2$, such that $d = \{d_q\}$ is S_n-equivariant and for any proper face σ of σ^n, the chain $a(\sigma) - z(\sigma) - d(\partial\sigma)$ is a $dim(\sigma)$-cycle. Essentially, d is a equivariant chain homotopy from the restriction $a|\mathcal{C}(bd(\sigma^n))$ to the restriction $z|\mathcal{C}(bd(\sigma^n))$. Then, using d, it is proved that $a(\partial\sigma^n) \sim (1+(n+1)k)\partial 0^n$, for some integer k. Since there is no integer k such that $(1+(n+1)k)$ is zero, $a(\partial\sigma^n)$ is not a boundary, for any value of n.

The problem with Lemma 6.12 in [55] is that it is not true that always there is such equivariant d. Consider a permutation $\pi \in S_n$. Chain map π partitions the simplexes of $\mathcal{C}(\sigma^n)$ and $\mathcal{C}(\mathcal{O}^n)$ into orbits: the *orbit* of a simplex σ of $\mathcal{C}(\sigma^n)$ or $\mathcal{C}(\mathcal{O}^n)$ is the set containing the simplexes $\pi^j(\sigma)$ for $j \geq 0$, where π^j denotes the j-fold composition of π. Consider a proper face σ of σ^n. We have that $d(\sigma)$ has the form $\sum \lambda_i \tau_i$. The problem comes when the orbits of σ and some τ_i are of distinct size. Consider the value of j such that $\pi^j(\sigma) = \sigma$. In this case we must have $\pi^j(\tau_i) = \tau_i$, since d is equivariant. However, it is not true that the orbits of σ and τ_i are of same size of for any $\pi \in S_n$, as σ and τ_i are simplexes of distinct dimension. This precludes to obtain an equivariant d.

The renaming lower bound proof of [55] is based on the one in [56], hence the the proof in [56] essentially has the same flaw. Generally speaking, first it is proved that a wait-free $2n$-renaming algorithm implies the existence of a $\mathbb{Z}_n$-equivariant chain map $b : \mathcal{C}(\sigma^n) \to \mathcal{C}(\sigma^n)$ such that $b(\partial\sigma^n) = 0$. Then it is claimed that there is a $\mathbb{Z}_n$-equivariant chain homotopy $\mathcal{D}$ from b to the identity chain map $i : \mathcal{C}(\sigma^n) \to \mathcal{C}(\sigma^n)$. Using $\mathcal{D}$, it is proved that $b(\partial\sigma^n) = (1+(n+1)k)\partial\sigma^n$, Lemma 6.1 in [56], hence $b(\partial\sigma^n)$ cannot be zero. As in [55], the problem is that it is not true that always there is such equivariant chain homotopy $\mathcal{D}$.

Chapter 8

A Panoramic View of Renaming

Since Attiya et al. [7] proposed renaming, variants of it have been introduced, concerning the size of the output name or time complexity. For example, a protocol for renaming is *size-adaptive* if the size of the output name space depends on the number of processes that actually participate in a given execution, and not on the total number of processes of the system. *Time-adaptive* protocols for renaming are defined similarly. These two properties of a renaming protocol are important because a desirable characteristic of a real distributed system is that the size of the output name space, or time complexity, gradually grows as the number of participating processes grows.

Another variant of renaming interesting in its own, is the *long-lived* renaming [60]. In this version of renaming every process of the system can repeatedly acquire and release output names in an execution. Long-lived renaming can be useful in a system in which processes are obtaining and releasing identical resources; the output names are the resources and the long-lived renaming protocol controls access to them.

Moreover, renaming has been generalized for groups of processes. In the *group renaming* task [40], the processes are partitioned into groups, and each process knows the input name of its group. It is required that each processor decides an output name for its group such that every two processes belonging to distinct groups decide distinct output names.

There are upper and lower bounds for all these renaming tasks in different timing models: synchronous, semi-synchronous and asynchronous. And there are results about the relations between renaming and other tasks such as set agreement [31]. This chapter presents a panoramic view of most of these results in an abridged form. It also describes some protocols, lower bounds and equivalences

8.1 Upper Bounds

Table 8.1 presents some of the existing protocols for renaming and its variants. The protocols are presented in three groups, renaming, long-lived renaming and group renaming. The second and fourth columns indicate if the corresponding protocol is size-adaptive or time-adaptive, respectively. The third and fifth columns show the output name space and time complexity. Variables p and N denote the number of processes that actually participate in a given execution and the size of the input name space, respectively, and c denotes a constant. The last column contains the year the protocols were proposed.

	Size-adap	Output space	Time-adap	Time complex	Year
Renaming					
Attiya et al. [8]	Yes	$2p-1$	No	$O(c^n)$	1990
Borowsky and Gafni [20]	Yes	$2p-1$	No	$O(n^3)$	1993
Moir and Anderson [60]	Yes	$p(p+1)/2$	Yes	$\Theta(p)$	1995
Moir and Anderson [60]	Yes	$2p-1$	Yes	$\Theta(p^4)$	1995
Afek and Merrit [2]	Yes	$2p-1$	Yes	$O(p^2)$	1999
Attiya and Fouren [12]	Yes	$2p-1$	No	$O(N)$	2001
Attiya and Fouren [12]	Yes	$6p-1$	Yes	$O(p \log p)$	2001
Attiya et al. [15]	Yes	$2p-1$	Yes	$O(p^3)$	2002
Afek et al. [4]	Yes	$4p^2$	Yes	$O(p^2)$	2002
Castañeda and Rajsbaum [27]	No	$2n-2$	—	—	2008
Gafni and Rajsbaum [48]	Yes	$2p-1$	No	$O(n^2)$	2009
Long-Lived Renaming					
Burns and Peterson [26]	Yes	$2p-1$	No	$O(c^n)$	1989
Buhrman et al. [25]	Yes	3^p	Yes	$\Theta(p)$	1995
Buhrman et al. [25]	Yes	$72p^2$	Yes	$\Theta(p \log p)$	1995
Buhrman et al. [25]	Yes	$p(p+1)/2$	Yes	$\Theta(p^3)$	1995
Moir and Anderson [60]	Yes	$p(p+1)/2$	No	$\Theta(pN)$	1995
Moir [61]	Yes	$p(p+1)/2$	Yes	$\Theta(p^2)$	1998
Moir [61]	Yes	$2p-1$	Yes	$\Theta(p^4)$	1998
Afek et al. [3]	Yes	$O(p^2)$	Yes	$O(p^2)$	1999
Afek et al. [3]	Yes	$O(p^2)$	Yes	$O(p^2 \log p)$	1999
Attiya and Fouren [11]	Yes	$2p-1$	Yes	$O(p^4)$	2000
Inoue et al. [58]	Yes	$O(p^2)$	Yes	$O(p^2)$	2001
Group Renaming					
Gafni [40]	Yes	$p(p+1)/2$	No	$O(n \log n)$	2004

Table 8.1: Upper bounds for renaming and its variants.

8.1.1 First Protocol

Attiya et al. in 1990 [8] (the journal version of [7]) presented the first wait-free protocol that solves $(2n+1)$-renaming. In fact, this protocol, $ABDPR$, was first presented in the message passing model and uses output name space $1, 2, \ldots, (n+1)+t$, where $t < (n+1)/2$ denotes the number of processes that may crash. In [16] it is proved that $ABDPR$ can be extended to the asynchronous wait-free read/write shared memory model, hence the resulting output name space is $1, 2, \ldots, 2n+1$ since $t = n$ for the wait-free case. Moreover, $ABDPR$ has a nice property, namely, it is size-adaptive. Recall that a protocol for renaming is size-adaptive if the output name space depends on the number of processes that actually participate in a given execution, and not on the total number of processes of the system. Specifically, $ABDPR$ uses $2p-1$ output names if $p \leq n+1$ processes participate. This is a desirable property because it is reasonable to ask for protocols that use few output names when few processes participate. Unfortunately, $ABDPR$ has exponential time complexity [37].

Code for process p_i **with input name** $inName_i$

> $prop_i \leftarrow 1$
> **while** $true$ **do**
>   **write**$_i$($< inName_i, prop_i >$)
> $(< x_1, s_1 >, < x_2, s_2 >, \ldots, < x_n, s_n >) \leftarrow$ **snapshot**$_i$()
>   **if** $prop_i = s_j$ for some $j \neq i$ **then** %each register initially contains $<\perp, \perp>$%
> $r_i \leftarrow$ rank of $inName_i$ in $\{x_j \neq \perp \mid 1 \leq i \leq n\}$
> $prop_i \leftarrow r_i$-th integer not in $\{s_j \neq \perp \mid 1 \leq i \neq j \leq n\}$
>   **else**
>   **decide** $prop_i$

Figure 8.1: Attiya et al. wait-free protocol for $(2p-1)$-renaming [9, pp 391-394].

Figure 8.1 contains the protocol for $(2p-1)$-renaming in [9, pp 391-394], which is an adaptation of the protocol in [8] to the asynchronous wait-free read/write shared memory model. The idea is simple, each process proposes an output name for itself, and if it sees no conflict then it picks its proposal, otherwise it proposes a distinct name among the not suggested names, and repeats until it does not see a conflict. Specifically, each process, p_i, writes in its own register a pair consisting of its initial input name, $inName_i$, and an output name it proposes for itself, $prop_i$, and then it takes an atomic snapshot of the whole memory. If no process proposed $prop_i$ then p_i decides its proposal. Otherwise p_i computes the rank, r, of $inName_i$ among the input names that appear in its snapshot, and proposes the r-th name that is not in its snapshot. Atomic snapshots can be wait-free implemented in $O(n \log n)$ steps [10].

Since the snapshot operation is atomic, the snapshots of any two processes occur one before the other, hence if there is a conflict between the proposals of the two processes, at least one of the processes is aware of it. Therefore, distinct processes pick distinct output names. Moreover, in an execution in which only $p \leq n+1$ processes participate, the rank of a processes is at most p, and any snapshot contains at most $p-1$ distinct proposal by other processes. Thus, the output name space is $\{1, \ldots, 2p-1\}$, i.e., the protocol is size-adaptive.

It is not hard to see that there cannot be an execution such that some non-faulty processes never decide, i.e., the protocol is wait-free. Suppose, for the sake of contradiction, that there is an execution such that some non-faulty processes never decide. Thus, there is a moment in which all non-faulty processes, and some of the faulty processes, have written in the memory. After this moment each process that never decides, gets a distinct rank, and thus it eventually will propose a name that does not conflict with any other proposal. A contradiction.

In [37] it is proved that $ABDPR$ has exponential time complexity $O(c^n)$. Very roughly, the idea is the following. Suppose there is an execution α such that each process p_i has input name i, and for some j, p_j and p_{j+1} have written in the memory the same proposition u. Therefore, when p_j and p_{j+1} get a snapshot (not necessarily concurrently), they will see a conflict. Consider the following extension of α: (1) p_{j+1} takes a snapshot, hence it will propose a value v_{j+1}, which in not in the memory, (2) a process p_i, with $i < j$, writes in the memory, and (3) p_j takes a snapshot, hence it will propose a value v_j. It is not hard to see that $v_j = v_{j+1}$, and thus p_j and p_{j+1} will see a conflict again. In [37] it is used this execution as a building block to produce an execution in which a pair of processes see a conflict an exponential number of times.

8.1.2 Polynomial Time Protocols

This section presents two size-adaptive protocols for $(2p-1)$-renaming with polynomial time complexity. Both protocols are recursive. The first protocol is simple and elegant, and is presented as an introduction to the second one, which was the first protocol for renaming with polynomial time complexity.

A Simple protocol

Gafni and Rajsbaum recently proposed [48] a simple, recursive and size-adaptive protocol for $(2p-1)$-renaming with time complexity $O(n^2)$. This protocol, GR, uses ideas from the protocol in [20], presented in the next section. In paper [48], Gafni and Rajsbaum promote the use of recursion in distributed computing by presenting simple protocols for various tasks that illustrate the advantages of thinking recursively.

Figure 8.2 presents the GR protocol. Each process p_i starts with input name $inName_i$. The specification of $\textbf{rename}_n$ is that if at most n processes invoke $\textbf{rename}_n(first, direction)$, where $first$ is an integer and $direction \in \{+1, -1\}$, each process decides an output name in the range $first_i, \ldots, first_i + (2n+2)$, when $direction_i = +1$, and decides and output name in the range $first_i - (2n-2), \ldots, first_i$, when $direction_i = -1$.

The first invocation of each process is with $first_i = 1$ and $direction_i = 1$, hence it receives an output name in the range $1, \ldots, 2n+1$. In GR each process p_i first writes its input name in a shared array M (each recursive invocation uses a distinct and clean shared array) and gets a view, S_i, of M by calling the **scan** function, which reads one by one the entries of M. Then, p_i updates $last_i$ containing the last name of the range of names it can choose. An idea used in the protocol is that a range of names, say $1, \ldots, m$, can be filled "going-up" or "going-down". The latter means that name m is considered as the first name, while the former considers 1 as the first name. In GR when $direction_i = 1$ the range of p_i is $first_i, \ldots, last_i = first_i + (2n+2)$, thus the range is filled going-up, and when $direction_i = -1$ the range is $last_i = first_i - (2n-2), \ldots, first_i$, hence the range is filled going-down.

Now, if $|S_i| < n$ then p_i invokes an instance of the protocol for $n-1$ processes and solves the problem with the processes that get a view of size less than n. And if $|S_i| = n$ then p_i verifies if its input name is the largest input name in S_i. If so, p_i decides the last name in the range, otherwise it calls a distinct instance of the protocol for $n-1$ processes and solves the problem with the processes that get a view of size n. Observe that $last_i$ is reserved for the process with largest input name, however, it is possible that no process picks $last_i$ (the process with largest input name does not necessarily get a view of size n). Although it is possible no process decides in an invocation of the protocol, the key is that the processes are partitioned into two non-empty sets, and then solve renaming recursively in ranges of names that do not overlap.

Code for process p_i with input name $inName_i$

Initially:
$first_i \leftarrow 1$
$direction_i \leftarrow 1$

rename$(n, first_i, direction_i)$
$M[i] \leftarrow inName_i$
$S_i \leftarrow$ **scan**(M)
$last_i \leftarrow first_i + direction_i * (2n-2)$
if $|S_i| = n$ **then**
  if $inName_i = \max(S_i)$ **then decide** $last_i$
  rename$(n-1, last_i - direction_i, -direction_i)$ % view of size n %
else
  rename$(n-1, first_i, direction_i)$ % distinct invocation for view of size less than n%

Figure 8.2: GR protocol for $(2p-1)$-renaming [48].

If $n = 1$ then a processes invoking **rename**$(n, first, direction)$ decides $first = last$. For induction hypothesis, assume that if $k \leq n-1$ processes call **rename**$(n-1, first, direction)$, they get names in the range $first, \ldots, first + (2k-2)$ if $direction = 1$ (going-up), and get names in the range $first - (2k-2), \ldots, first$ if $direction = -1$ (going-down). Now, assume $\ell \leq n$ processes call **rename**$(n, first, direction)$ going-up, i.e., $direction = 1$ (the going-down case is very similar). If $\ell < n$ then no process gets a view of size n, hence all process go to the same invocation **rename**$(n-1, first, direction)$. By induction hypothesis, the process get names in the range $first, \ldots, first + (2\ell - 2)$. Note this shows GR is size-adaptive.

Consider now the case $\ell = n$. Let $X_{partial}$ and X_{full} be set of processes that call **rename**$(n-1, first, direction)$ and **rename**$(n-1, last-1, -direction)$, respectively. We need to prove that processes in $X_{partial}$ and X_{full} get names in ranges that do not overlap. Let $\ell_{partial}$ and ℓ_{full} be $|X_{partial}|$ and $|X_{full}|$. Notice $\ell_{full} + \ell_{partial} = n$. By induction hypothesis, processes in $X_{partial}$ get names in $R_{partial} = [first, \ldots, first + (2\ell_{partial} - 2)]$ and processes in X_{full} get names in $R_{full} = [(last-1)-(2\ell_{full}-2), \ldots, (last-1)]$. Since $last = first + (2n-2)$, $R_{full} = [first+(2n-3)-(2\ell_{full}-2), \ldots, first+(2n-3)]$. It must be that $first+(2\ell_{partial}-2) < first+(2n-3)-(2\ell_{full}-2)$, because if $first+(2\ell_{partial}-2) \geq first+(2n-3)-(2\ell_{full}-2)$ then it can be proved $2(\ell_{partial}+\ell_{full}) \geq 2n+1$, which is not possible since $\ell_{full} + \ell_{partial} = n$. Therefore, $R_{partial}$ and R_{full} are disjoint.

Finally, GR has time complexity $O(n^2)$: a process executes at most n recursive calls, and in each recursive call it executes a **scan** operation, which has time complexity $\Theta(n)$.

First Polynomial Time Protocol

Borowsky and Gafni presented the first solution for renaming with polynomial time complexity in [20]. This protocol, BG, is size-adaptive and solves $(2p-1)$-renaming in $O(n^3)$ steps. They presented BG in the immediate snapshot (IS) model [19, 20, 70], which includes only a subset of all possible executions.[1] Roughly speaking, in this model each process has just one atomic operation that writes on its register and takes a snapshot of the whole memory. See Section 2.1 for more details.

Figure 8.3 presents the BG protocol. It proceeds in independent asynchronous rounds that use one-shot IS objects (objects that allow to execute just one immediate snapshot per process). Processes access these objects using the function **writeSnapshot**. In every round, each process updates the first name of the range of names it can choose, using function **newRange**, according with the snapshot S it gets from **writeSnapshot**. The size of this range is $2|S|-1$. The idea is that the processes with snapshot S solve the problem using this range of names. The first name of that range, $firstName$, is reserved for the process with the highest input name among the input names in the snapshot. If such process gets the same snapshot (it does not necessarily get it) then it chooses $firstName$. The other processes execute an independent round for solving the problem recursively in a range of names without $firstName$. Recall that for a range of names $1, \ldots, m$, going-up means that name 1 is considered as the first name, while going-down considers m as the first name. In BG, processes fill their respective ranges going-down in the first round, going-up in the second round, going-down in the third one and so on.

Code for process p_i **with input name** $inName_i$

> **Initially:**
> > $firstName_i \leftarrow 0$
> > $direction_i \leftarrow true$
> > $round_i \leftarrow 0$
>
> **rename**$_i$**(**$round_i, firstName_i, direction_i$**)**
> > $snapshot_i \leftarrow$ **writeSnapshot(**$inName_i, round_i$**)**
> > $firstName_i \leftarrow$ **newRange(**$firstName_i, direction_i, |snapshot_i|$**)**
> > **if** $inName_i = \max(inName_j \in snapshot_i)$ **then**
> > > **decide** $firstName_i$
> >
> > **else**
> > > $round_i \leftarrow \text{append}(round_i, |snapshot_i|)$
> > > **rename**$_i$**(**$round_i, firstName_i, \neg direction_i$**)**
>
> **newRange(**$name, direction, snapSize$**)**
> > **if** $direction$ **then**
> > > **return** $name + (2snapSize - 1)$
> >
> > **else**
> > > **return** $name - (2snapSize - 1)$

Figure 8.3: BG protocol for $(2p-1)$-renaming [20].

As an example, consider the execution in Figure 8.4 with three processes with input names A, B and C. In the first round C executes an IS operation alone and then A and B concurrently. Thus,

[1] The IS model is presented in [20] via the *participating set* task introduced in the same paper.

the snapshot of C only contains C and the snapshots of A and B contain all processes. Hence C decides 1 and A and B reserve 5 for C (they fill going-down their range). Since A and B got the same snapshot, they execute together an independent round. In that round A and B execute an IS operation concurrently, hence they get the same snapshot again. Process B decides 2 (A and B now fill going-up). Finally, A decides 3 in the last round.

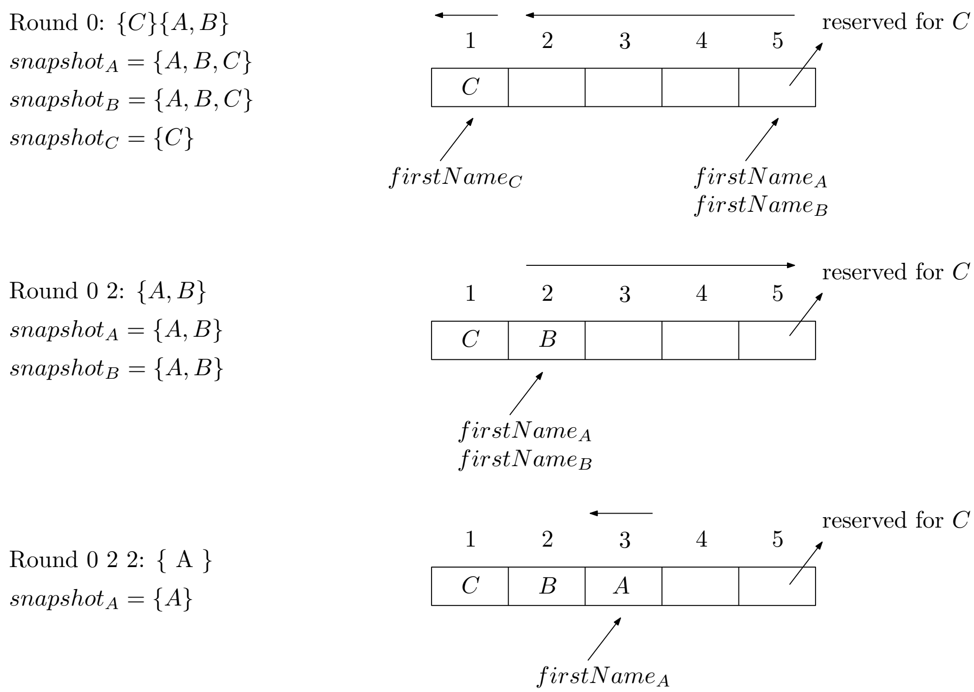

Figure 8.4: Execution with processes with names A, B and C.

In Section 2.1 it is explained that processes that call **writeSnapshot** are partitioned in a series of non-empty sets, $A_1, A_2, \ldots, A_m$. All processes in each set A_i get the same snapshot; they write concurrently first and then read the whole memory. We can think of the snapshots obtained by processes as sets containing the id's that appear in them. Thus, **writeSnapshot** outputs a series of sets ordered by containment, i.e., $T_1 \subset T_2 \subset \ldots \subset T_m$, where T_i is the set of id's corresponding to the snapshot obtained by processes in A_i. Note that $A_i = T_i - T_{i-1}$ (for $i = 1$, $T_0 = \emptyset$). In other words, the processes that obtain snapshot T_i are the processes in $T_i - T_{i-1}$. This is the property of IS executions that allow processes decide distinct output names. When processes are going-down (the going-up case is similar), the first name of the range of processes with snapshot T_i is $2|T_i| - 1$, and the last one is $(2|T_i| - 1) - (2|T_i - T_{i-1}| - 1) = 2|T_{i-1}|$, because the processes that get T_i are the processes in $T_i - T_{i-1}$. Moreover, the first name of the processes with snapshot T_{i-1} is $2|T_{i-1}| - 1$. Therefore, the range of name of processes with snapshots T_i and T_{i-1}, respectively, do not overlap, hence processes decide distinct output names. Also, observe that the range of names of a process depends on the size of the snapshot it gets, hence BG is size-adaptive.

Notice that at least the process with the highest input name decides in each round, and thus BG requires at most n independent rounds. Since function **writeSnapshot** can be implemented in $O(n^2)$ steps [20], the time complexity of BG is $O(n^3)$. However, Gafni and Rajsbaum observed

[48] that the time complexity of BG is actually $O(n^2)$. They give a recursive implementation of **writeSnapshot** with time complexity $\Theta(n(n-s+1))$, where s is the size of the snapshot received by **writeSnapshot**. Also they modify BG such that each process executes a **writeSnapshot** function for x processes, where x is the size of the snapshot it gets in the previous round, minus one (in the first round **writeSnapshot** is for $n+1$ processes). Therefore, the steps a process executes in a given round is $\Theta((s_p-1)((s_p-1)-s_c+1))$, where s_p and s_c are the size of the snapshots that the process gets in the previous and current rounds, respectively. The worst execution is the one in which a process executes n rounds, hence in the i-th round it gets a snapshot of size $n-i+1$. (this is possible if all processes run lock-step, and thus they obtain the same snapshot in each round) Therefore, the process executes $n-i+1$ steps in the i-th round, giving a time complexity $O(n^2)$.

8.1.3 Time-Adaptive Renaming

The time complexity of $ABDPR$, GR and BG in Sections 8.1.1 and 8.1.2, depend on the total number of processes of the system. However, in real distributed systems it is desirable that the time complexity of a protocol only depends on the number p, $p \leq n+1$, of processes that actually participate, i.e., its time complexity adjust to the number of active processes. More precisely, the time complexity is constant if just one process participates and it gradually grows as the number of participating processes grows. Such a protocol is called *time-adaptive*.[2]

Moir and Anderson [60] presented a size- and time-adaptive protocol, MA, for $(p(p+1)/2)$-renaming with time complexity $O(p)$, using wait-free building blocks. Intuitively, each one of these blocks is capable of "splitting" a group of processes into smaller subsets. Formally, a *splitter*[3] is a shared object that outputs *stop*, *down* or *right* such that if m processes execute it, at most one process obtains *stop*, at most $m-1$ processes obtain *down* and at most $m-1$ obtain *right*. Also, when a single process executes the splitter, it obtains *stop*.

$(p(p+1)/2)$-**renaming**$_i(inName_i)$
  $move_i \leftarrow \perp$ % local %
  $i, j \leftarrow 0$ % local %
  while $move \neq stop$
    $move \leftarrow$ **splitter**$_{i,j}(inName_i)$
    if $move = down$ **then** $i++$
    if $move = right$ **then** $j++$
  decide number of splitter Splitter$_{i,j}$

splitter$_{i,j}(name)$
  $X_{i,j} \leftarrow name$ % $X_{i,j}$ and $Y_{i,j}$ are multi read/write %
  if $Y_{i,j}$ **then return** $right$ % shared variables. $Y_{i,j}$ is iniatially *false* %
  else $Y_{i,j} \leftarrow true$
  if $X_{i,j} = name$ **then return** $stop$
  else return $down$

Figure 8.5: Splitter based protocol for $(p(p+1)/2)$-renaming in [60].

[2] Some authors use the term *fast*.
[3] The term "splitter" was introduced in [12].

Figure 8.5 contains the protocol MA and a wait-free implementation of a splitter building blocks. Each splitter is implemented using two multi-reader/multi-writer shared variables X and Y. Initially, Y contains $false$. When a process q executes a splitter, first it writes its input name in X and then checks the value of Y. If Y contains $true$ then q gets $right$, otherwise q sets Y to $true$. Now q checks if X still contains its name. If so, then q gets *stop*, otherwise it gets *down*. Observe that if some processes get $right$ then at least one process has to set Y to $true$. Thus, not all processes can get $right$. Similarly, not all processes can get *down* because the last process writing its name in X, will see X still contains its name. Finally, a processes, q, obtaining *stop* has to read its name from X, hence no process executed the splitter from the moment when q wrote its name in X to the moment when q reads from it, i.e., q is the first process accessing the splitter. Therefore, q is the only process that gets *stop*.

Protocol MA uses a grid of $(n+1) \times (n+1)$ splitters numbered from 1 to $(n+1)^2$, as in Figure 8.6. Each process starts to execute the splitter 1 at the upper left corner. If a process receives *stop* from splitter i then it picks name i, otherwise it moves right or down through the grid, according to the output received from the splitter. The protocol appears in Figure 8.5. In an execution with p processes, each process executes at most p splitters, thus it picks a name in the p-th diagonal, in the worst case. Therefore, MA solves $p(p+1)/2$-renaming in time complexity $O(p)$.

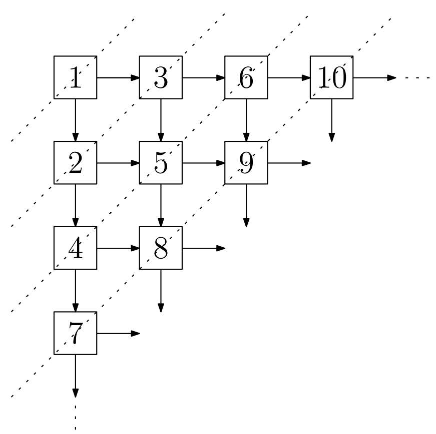

Figure 8.6: Grid of splitters for the $O(p^2)$-renaming protocol with time complexity $O(p)$ in [60].

Attiya and Fouren present in [12] a size-adaptive protocol, $AF_{(2p-1)O(N)}$, for $(2p-1)$-renaming with time complexity $O(N)$, where N is the size of the input name space, and a time-adaptive protocol, AF_{as}, for one-shot atomic snapshot with time complexity $O(p \log p)$. Attiya and Fouren combined [12] protocols MA, $AF_{(2p-1)O(N)}$ and AF_{as} to produce a size- and time-adaptive protocol for $(6p-1)$-renaming, AF_{6p-1}, with time complexity $O(p \log p)$. Very roughly, they first use MA to reduce the input name space, namely $N = O(p^2)$, and then, using various instances of $AF_{(2p-1)O(N)}$, they achieve a size-adaptive $(2p-1)$-renaming protocol, $AF_{(2p-1)O(n \log n)}$, with time complexity $O(n \log n)$. Finally, in AF_{6p-1}, the processes that obtain the same snapshot from AF_{as}, execute an independent instance of $AF_{(2p-1)O(n \log n)}$ designed for that specific snapshot.

In [2] Afek and Merrit present a size- and time-adaptive protocol for $(2p-1)$-renaming with time complexity $O(p^2)$. To the best of our knowledge, this is the best protocol for size-adaptive renaming, considering time complexity and output space. First, Afek and Merrit achieve a size-and time-adaptive $(2p-1)$-renaming protocol, AM, with time complexity $O(f(p)^2)$, where $f(p)$ is an upper bound for the size of the input name space when p processes participate. Then they use AF_{6p-1} in [12] to reduce the name space to $f(p) = 6p-1$ in time $O(p \log p)$, and then AM achieves the output name space $2p-1$ in time $O(p^2)$.

8.1.4 Long-Lived Renaming

Moir and Anderson introduced [60] the *long-lived* version of renaming in which each process can repeatedly acquire and release output names. Similarly to one-shot renaming, a protocol for long-lived renaming can be time- or size-adaptive. In this case, size-adaptive means that the output space is on function of the number of processes p, $p \leq n+1$, that are "simultaneously" obtaining output names or have output names in their possession. And time-adaptive means that the step complexity of acquiring or releasing a name is on function of p. Value p is usually called the *point contention*.

This problem is interesting on its own. Long-lived renaming can be useful in a system in which processes are obtaining and releasing identical resources; the output names are the resources and the protocol controls access to them. Moreover, a protocol for long-lived renaming can be useful in improving the performance of a shared object [1]. Intuitively, the idea is that a protocol for long-lived renaming can be used to produce a protocol that bounds the maximum number of processes that can concurrently access a shared object, hence the performance of the object is improved.

The first protocol for long-lived renaming was developed by Burns and Peterson [26], which is size-adaptive, uses output space $2p-1$, and has exponential time complexity [37].

Moir and Anderson showed [60] that their splitter-based protocol for renaming (see Section 8.1.3) can be modified to produce a long-lived renaming protocol. Each splitter replaces its single boolean variable Y with a boolean array Y' of length N, one entry per input name. The idea is to use Y' to make the splitter "resettable". A splitter now proceeds in the following way. Each process, q, executing the splitter first writes its name in X and then inspects all Y'. If there is at least one entry containing *true* then q receives *right*. Otherwise q writes *true* in $Y'[q's\_in\_name]$ and verifies if X still contains its input name. if so then q gets *stop*, otherwise q writes $false$ in $Y'[q's\_in\_name]$ and receives *down*. When q releases a splitter, it simply does $Y'[q's\_in\_name] = false$. The resulting protocol is size adaptive and uses output name space $p(p+1)/2$, and has time complexity $O(Np)$. Moir [61] improve the time complexity of this protocol to $O(p^2)$ by reducing the size of Y' in each splitter.

The best solution we know for long-lived renaming is presented by Attiya and Fouren in [11]. This protocol is size- and time- adaptive, it solves $(2p-1)$-long-lived renaming on $O(p^4)$ steps. See [3, 4, 25, 58] for various protocols for long-lived renaming. Also papers [1] contain solutions for long-lived renaming using red-modify-write operations.

8.1.5 Group Renaming

Gafni proposed [40] the *group renaming* task, a generalization of renaming in which the processes are partitioned into groups and each group has a unique input name. Each process must decide an output name for its group such that two processes belonging to distinct groups choose distinct output names. For group renaming, size-adaptive means that the number of output names is on function of the number of groups that participate.

In [40] Gafni presents a size-adaptive protocol for $p(p+1)/2$-group renaming, G_{group}, where p denotes the number of groups that actually participate. Each group has a multi-reader/multi-writer register initialized to $\perp$. Every process, q, of group i, writes i on the register dedicated to its group, and then takes a snapshot S of the memory. Process q picks output name $|S|(|S|-1)/2+r$, where r is the rank of i in S. Observe that processes with snapshots of same size have the same snapshot, hence processes with snapshots of same size and in distinct groups, cannot decide the same output name. Also, processes with snapshots of different size use ranges of output names that do not overlap. Moreover, if p groups participate, the output name space is $1, 2, \ldots, p(p-1)/2 + p = p(p+1)/2$. Finally, since snapshots can be implemented on $O(n \log n)$ steps [10], the time complexity of G_{group} is $O(n \log n)$.

Afek et al. [6] present a protocol for a strong version of group renaming, in which processes of the same group decide the same output name. The protocol in [6] solves $2(p-1)$-group renaming and uses g-consensus objects and read/write registers, where p is the number of groups that participate, g is the maximum number of processes in a group and a g-consensus object solves consensus on g processes. Generally speaking, this algorithm is a modification of $ABDPR$ in [6] for groups of processes. Using g-consensus objects, processes of a group agree on a snapshot of some process in the group, and use it to propose an output name to their group. Therefore, processes of the same group decide the same output name. In [6] it is also proved that $2(p-1)$-group renaming is not solvable with $(g-1)$-consensus objects and read/write registers.

8.2 Relations of Renaming with Other Tasks

An interesting topic in distributed computing is the relative power of different tasks. That is, given two tasks, the question is if one of them can be used to implement the other or if they are incomparable. Recall that a task $\mathcal{A}$ implements a task $\mathcal{B}$ if there is a wait-free protocol that solves $\mathcal{B}$ from objects that solve $\mathcal{A}$ and read/write registers. And $\mathcal{A}$ and $\mathcal{B}$ are equivalent if $\mathcal{A}$ implements $\mathcal{B}$ and $\mathcal{B}$ implements $\mathcal{A}$.

One way to measure the power of tasks is the *consensus number* [52] introduced by Herlihy. The consensus number of an object is a number that denotes the maximum number of processes for which the object can solve consensus in a wait-free manner. For example, the consensus number of read/write registers is 1, because it is not possible to achieve consensus on two processes, or more, by using only read/write memory [39]. On the other hand, *test&set*, *queue* and *stack* objects have consensus number 2 [52], since they can solve consensus on two process but not on three. In general, an object with consensus number n can be used to solve consensus on n processes but not on $n+1$ processes. Moreover, an object with consensus number n is universal in a system with n processes in the sense that it can be used to construct a protocol that solves any task.

However, tasks as renaming, WSB and set agreement (explained below) represent a problem in this consensus hierarchy. It has been proved that they are not wait-free solvable (for certain parameters) [14, 19, 27, 55, 56, 70], but they are too weak to solve consensus on two processes. Thus, it is important to study the relative power of the tasks in this class. These tasks are called *subconsensus* tasks.

8.2.1 Renaming, Set Agreement, WSB and SSB

The *k-set agreement* task, proposed by Chaudhuri in [31], is a generalization of the classic consensus task. In this task each process starts with a private input value and must choose a private output value. It is required that the value chosen by a process is the input of some process, and at most k distinct values may be chosen. Clearly, for $k = 1$, k-set agreement task is consensus. For brevity, we use *set agreement* instead of n-set agreement. There are several proofs [14, 19, 55, 56, 70] showing that set agreement is not wait-free solvable in the asynchronous wait-free read/write shared memory model, thus neither is k-set agreement for $k < n$.

The *k-test&set* task [19] (called $(n + 1, k)$-set-test-and-set in [19]) is a generalization of the well-known test&set. The output values are 0 (winner) or 1 (loser). In every execution, at least one and at most k processes get winner. The n-test&set task is called *strong symmetry breaking* (SSB). Observe that SSB is a stronger version of WSB, which requires that that in every execution at least one process decides 0, in addition to the requirement for WSB that not all decide either 0 or 1.

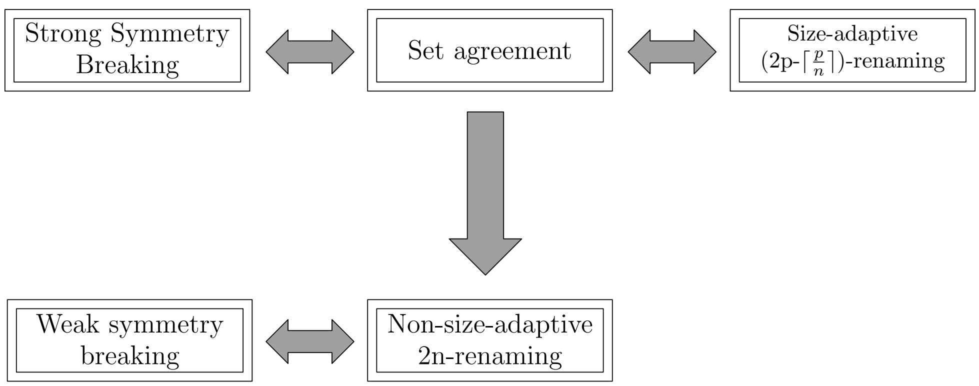

Figure 8.7: Hierarchy of subconsensus tasks.

Figure 8.7 presents some of the known relations between set agreement, renaming, WSB and SSB. An arrow from a task $\mathcal{A}$ to a task $\mathcal{B}$ means that $\mathcal{A}$ implements $\mathcal{B}$. Thus, size-adaptive $(2p - \lceil\frac{p}{n}\rceil)$-renaming ($p$ is the number of participating processes) and set agreement are equivalent, while set agreement and non-size-adaptive $2n$-renaming are not. Observe that, for size-adaptive $(2p-\lceil\frac{p}{n}\rceil)$-renaming, if not all processes participate ($p < n+1$), the output name space is $1, \ldots, 2p-1$, and if all processes participate, the output name space is $1, \ldots, 2p-2 = 2n$. Since set agreement is not wait-free solvable, size-adaptive $(2p-\lceil\frac{p}{n}\rceil)$-renaming is not wait-free solvable. The equivalence between WSB and non-size-adaptive $2n$-renaming is explained in detail in Section 2.2.

8.2.2 Set Agreement Implements $2n$-Renaming

Gafni et al. prove in [42] that set agreement implements $2n$-renaming by showing that set agreement implements WSB. The implementation appears in Figure 8.8. It uses two shared arrays M_1 and M_2 of integers, two set agreement objects SA_1 and SA_2 and a $(2n + 1)$-renaming object R. It is not important if R is size-adaptive or not, thus R can be any of the size-adaptive protocols for $(2p-1)$-renaming in Section 8.1. The processes first call R to choose an *id* in the range $1, \ldots, 2n+1$.

Since R is anonymous, this step guarantees that the protocol is anonymous. The processes with id in the range $1, \ldots, n+1$ call SA_1 using their id as input. Then they write into M_1 what they receive from SA_1. A process decides 0 if it reads its id from M_1, otherwise it decides 1. The processes with id in the range $n+2, \ldots, 2n+1$ do the same with SA_2 and M_2, except that they decide inverted values.

Consider an execution in which all processes decide. Let s be the number of processes with id in the range $1, \ldots, n+1$. Notice that $s \geq 1$, and obviously, $s \leq n+1$. Among those s processes, the one whose id is first to be written in M_1, decides 0. If $s = n+1$ then all processes call SA_1 and, by the specification of set agreement, it is not possible that all decide 0. If $s < n+1$ then $n+1-s$ processes call SA_2 and the process whose id is first to be written in M_2 decides 1. Thus, the protocol in Figure 8.8 solves WSB.

```
int[n+1] M₁   %(n+1)-element array, initally 0%
int[n+1] M₂   %(n+1)-element array, initally 0%
Renaming R = new Renaming()   %(2n+1)-renaming object%
Set agreement SA₁ = new Set Agreement()   %set agreement object%
Set agreement SA₂ = new Set Agreement()   %set agreement object%

WSB_i(intName_i)
    name_i ← R.choose(inName_i)   %anonymous id%
    if name_i ≤ n+1 then
        M₁[id_i] = SA₁.decide(id_i)
        for each j ∈ M₁ do
            if j = id_i then decide 0
        decide 1
    else
        M₂[id_i] = SA₂.decide(id_i)
        for each j ∈ M₂ do
            if j = id_i then decide 1
        decide 0
```

Figure 8.8: From set agreement to WSB [42].

In [42] it is also proved that $2n$-renaming is strictly weaker than set agreement, namely, there is no implementation of set agreement from $2n$-renaming. However, they only prove that such implementation does not exist for n even. It remains an open question what is the relationship between these two tasks for n odd.

8.2.3 Equivalence between Set Agreement, $(2p - \lceil \frac{p}{n} \rceil)$-Renaming and SSB

Borowsky and Gafni prove in [19] that k-set agreement is equivalent to k-test&set for any value of k, hence set agreement is equivalent to SSB. Also, Gafni shows [41] that SSB and size-adaptive $(2p - \lceil \frac{p}{n} \rceil)$-renaming are equivalent. Mostéfaoui et al. [62] and Gafni et al. [43] generalize that result proving that k-test&set and $(2p - \lceil \frac{p}{k} \rceil)$-renaming are equivalent for any value of k. Gafni et al. [46] present a t-resilient protocol for k-set agreement that uses size-adaptive $(p+k-1)$-renaming objects, with $t = k$. They also show that there is no such protocol if $k < t$ and $k < (n+1)/2$. Also, Gafni shows [47] that k-set agreement implements size-adaptive $(p+k-1)$-renaming.

The implementation of SSB from $(2p - \lceil \frac{p}{n} \rceil)$-renaming is simple. Processes call a $(2p - \lceil \frac{p}{n} \rceil)$-renaming object with their ids as inputs. If a process gets a name less than $n+1$, it decides 0, otherwise decides 1. Observe that if $p < n+1$ processes participate then they get names in the range $1, \ldots, 2p-1 = p + (p-1)$. Thus at least one process gets a name in the range $1, \ldots, p$ and decides 0. And if all processes participate, $p = n+1$, then they get names in the range $1, \ldots, 2n$, hence at least one process decides 0 and at least one process decides 1.

Since set agreement and SSB are equivalent and $(2p - \lceil \frac{p}{n} \rceil)$-renaming implements SSB, then $(2p - \lceil \frac{p}{n} \rceil)$-renaming is not wait-free solvable (recall that set agreement is not wait-free solvable). Moreover, observe that in the proof of that $(2p - \lceil \frac{p}{n} \rceil)$-renaming implements SSB, it is irrelevant if a protocol for $(2p - \lceil \frac{p}{n} \rceil)$-renaming is anonymous or not. Therefore, $(2p - \lceil \frac{p}{n} \rceil)$-renaming is impossible even if we drop the anonymity requirement for renaming. In other words, what makes $(2p - \lceil \frac{p}{n} \rceil)$-renaming impossible is the requirement that the output name space gradually grows as the number of participating processes grows.

Figure 8.9 contains a $(2p - \lceil \frac{p}{n} \rceil)$-renaming implementation from an n-immediate snapshot object, ISn, and an array, R, with $n+1$ size-adaptive $(2p-1)$-renaming objects. This protocol is the protocol in [62] for $k = n$. Recall that in the IS model each process has just one operation that atomically writes on its register and then takes a snapshot of the whole memory (see Section 8.1.2). An n-IS object guarantees at most n processes get the same snapshot.[4] Such object can be implemented from SSB [62], thus SSB implements $(2p - \lceil \frac{p}{n} \rceil)$-renaming.

Renaming[n] R   %(n+1)-element array with $(2n+1)$-renaming objects%
ImmediateSnapshot ISn   %n-immediate snapshot set object%

$(2p - \lceil \frac{p}{n} \rceil)$**-renaming**$_i(intName_i)$
  $S_i \leftarrow$ ISn.snapshot(id_i) %at most n processes get the same snapshot%
  $base_i \leftarrow 2|S_i| - \lceil \frac{|S_i|}{n} \rceil$
  $\mathit{offset}_i \leftarrow$ R$[|S_i|]$.rename$(inName_i)$
  decide $base_i - \mathit{offset}_i + 1$

Figure 8.9: From n-immediate snapshot to size-adaptive $(2p - \lceil \frac{p}{n} \rceil)$-renaming [62].

The protocol in Figure 8.9 works as follows. Each process, q, first gets a snapshot, S, from object ISn. Then, q computes a base name according with the size of S, and calls the $(2p-1)$-renaming object associated with snapshots of size $|S|$. Therefore, processes with snapshots of same size call the same renaming object (actually snapshots of same size are equal). Finally, q decides a name according with its base name and the name it gets from the renaming object.

Clearly, processes that receive the same snapshot from object ISn, do not decide the same output name. For processes with different snapshots, the structure of IS executions allow them do not decide the same name. The argument is essentially the same as for BG protocol in Section 8.1.2. Object ISn outputs a series of snapshots ordered by containment, $T_1 \subset T_2 \subset \ldots \subset T_m$, and the processes that obtain snapshot T_i are the processes in $T_i - T_{i-1}$ (for $i = 1$, $T_0 = \emptyset$). Consider a snapshot T_i. First suppose that $|T_i| < n+1$. Since the objects in R solve size-adaptive $(2p-1)$-renaming, the processes with snapshot T_i pick names in the range $(2|T_i|-1)-(2|T_i - T_{i-1}|-1)+1 = 2|T_{i-1}|+1, \ldots, 2|T_i|-1$. Also, observe that the processes with snapshot T_{i-1} do not decide a name greater than $2|T_{i-1}|-1$. Now suppose that $|T_i| = n+1$. ISn guarantees that at most n processes

[4]In [62], this object is called *n-participating set* since the IS model is presented in [20] via the participating set task.

get T_i as snapshot. Therefore, T_{i-1} must contain at least one element. The processes with snapshot T_i decide names in the range $(2|T_i| - 2) - (2|T_i - T_{i-1}| - 1) + 1 = 2|T_{i-1}|, \dots, 2|T_i| - 2$, and the processes with snapshot T_{i-1} do not pick a name greater than $2|T_{i-1}| - 1$.

8.2.4 Renaming and Failure Detectors

Chandra and Toueg introduced the concept of *failure detector* [30] in order to evade the impossibility of consensus in an asynchronous environment. They propose to augment the system with a failure detector mechanism that gives some information about faulty processes. This mechanism may give distinct information to distinct processes. Moreover, it can be unreliable, but it eventually guarantees some properties about the information it gives to the processes. Failure detectors allow to design protocol in a more modular way: a protocol does not have to take care about the model, it just has to deal with the specification of the failure detector.

The relation between renaming and failure detectors has been studied. Mostéfaoui et al. [62] show that a failure detector of the class Ω^k can implement $(2p - \lceil \frac{p}{k} \rceil)$-renaming. The class Ω^k [65], $1 \leq k \leq n+1$, of failure detectors was introduced by Neiger. Each time a failure detector in this class is invoked, it outputs a set of *id*'s of size at most k. The failure detector guarantees that there is a moment after it always outputs the same set, which contains at least one correct process. Also, in [63] it is presented an implementation of $min(2p-1, p+k-1)$-renaming from a failure detector in the class Ω^k_*. The class Ω^k_* is introduced by Raynal and Travers in [69], and is a variation of the class Ω^k. The implementation in [63] is a modification of protocol $ABDPR$ in Section 8.1.1.

Afek and Nir [5] investigate failure detectors in loosely named systems . In a *tightly named system* the *id*s are drawn from a space of size $n+1$ and in a *loosely named system* the *id*s are drawn from a space of size greater than $n+1$. That paper introduces a failure detector class capable of solving WSB but that cannot solve set agreement.

8.3 Lower Bounds

This section presents a brief overview of some of the known lower bounds for renaming. For size-adaptive and non-size-adaptive renaming there are lower bounds concerning the output name space. Also there are lower bounds for the smallest number of rounds needed for achieve strong renaming in synchronous and semy-synchronous message passing models. In the *strong* version of renaming the size of the output name space is equal to the number of processes of the system. To our knowledge, there are no lower bounds for group renaming.

8.3.1 Size-Adaptive Renaming

Section 8.2.3 explains that size-adaptive $(2p - \lceil \frac{p}{n} \rceil)$-renaming is equivalent to set agreement. Also, it has been proved [14, 19, 55, 56, 70] that set agreement is not wait-free solvable, hence, size-adaptive $(2p - \lceil \frac{p}{n} \rceil)$-renaming is not wait-free solvable. Therefore, the size-adaptive protocols for $(2p-1)$-renaming presented in Section 8.1, are optimal. In other words, this lower bound says that if we ask $(2p-1)$-renaming to save one name only when all processes participate, the task becomes unsolvable. As explained in Section 8.2.3, $(2p - \lceil \frac{p}{n} \rceil)$-renaming is impossible even if we drop the anonymity requirement for renaming. This lower bound also holds for size-adaptive long-lived renaming.

8.3.2 Non-Size-Adaptive Renaming

As explained in Chapter 1, Attiya et al. [8] proved that there is no wait-free protocol for M-renaming for $M \leq n+2$. Then, Herlihy and Shavit presented [55] a lower bound stating that no wait-free M-renaming protocol exists when $M < 2n+1$. There are various proofs of the same result [14, 56, 53], all closely related and based on algebraic topology. However, Chapters 5 and 6 prove that, for some values of n, this lower bound is incorrect, namely, there is a wait-free $2n$-renaming protocol, while for the other values of n, the lower bound holds.

8.3.3 Strong Renaming

Attiya et al. [8] showed that there is no renaming protocol if the output name space is $1, \ldots, n+2$, implying that strong renaming is not wait-free solvable. Recall that the output name space is $1, \ldots, n+1$ for strong renaming. However, there are protocols for strong renaming that use powerful primitives, read-modify-write objects, and in synchronous and semi-synchronous models. Moreover, there are lower bounds for the smallest number of rounds needed for solving strong renaming.

Raynal presents in [67] a wait-free protocol for strong renaming in the asynchronous read/write shared memory that is enriched with compare&swap objects. Chaudhuri et al. show [32] a size- and time-adaptive protocol for strong p-renaming in $O(\log p)$ rounds in the synchronous message passing model. This protocol is comparison-based. In this case time-adaptive means that the number of rounds adapts to the number p, $p \leq n+1$, of participating processes. In [32] it is also proved that any comparison-based protocol for strong renaming needs $\Omega(\log p)$ rounds. Okun et al. [66] go beyond and consider the synchronous message passing model with Byzantine failures. They achieve two comparison-based protocols for strong renaming in $O(\log n)$ and $O(n \log^2 \lceil N_0/n \rceil)$ rounds, respectively, where N_0 is the largest input name among all correct processes. The first protocol considers that an arbitrary number of processes can fail, while the second one assumes that less than $(n+1)/3$ processes can fail.

Djerassi-Shintel presents in [34] a protocol for strong renaming in a semi-synchronous message passing model, in which there is inexact information about time. The amount of real time between two consecutive steps of a non-faulty process is at least c_1 and at most c_2, and a message sent by a non-faulty process is delivered within time at most d. A faulty processes do not necessarily obey the timing requirements. In this model, Djerassi-Shintel achieves a comparison-based protocol in $O(\frac{\log n}{\log \log n}(c_2 \log n + \frac{c_2}{c_1} d))$ rounds. This protocol is optimal, since Attiya and Djerassi-Shintel show [13] that any comparison-based protocol for strong renaming needs $\Omega(\log p \frac{c_2}{c_1} d)$ rounds in presence of $p-1$ timing faults, where $p \leq n+1$ is the number of processes that participate.

The strong version of long-lived renaming has also been studied. Moir and Anderson [60], Brodsky et al. [24] and Herlihy et al. [57] present various size- and time-adaptive protocols for strong long-lived p-renaming in the asynchronous wait-free read/write shared memory model that is enriched with read-modify-write operations.

Chapter 9

Conclusions and Further Research

This chapter presents the conclusions obtained from all previous chapters, and explains that the results about the wait-free solvability of WSB in Chapters 5 and 6, have implications for the t-resilient case. Also, it presents some directions for future research.

9.1 Conclusions

This thesis is a study of WSB and its relation with renaming, in the asynchronous wait-free read/write shared memory model with $n + 1$ processes. First, it presents a detailed proof of the equivalence between WSB and $2n$-renaming, and then, using the topology approach to distributed computing, it achieves a necessary and sufficient condition for wait-free WSB solvability. Essentially, this condition states that WSB is wait-free solvable if and only if there is a chromatic subdivision of an n-dimensional simplex with a binary coloring which is symmetric on the boundary, and without monochromatic n-dimensional simplexes. Then, the thesis proves the following two combinatorial topology results:

1. Any chromatic subdivision of an n-dimensional simplex with a symmetric binary coloring has

$$1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i$$

   monochromatic n-dimensional simplexes, for some integers $k_0, \ldots, k_n$; some monochromatic simplexes are counted as $+1$ and the other as -1.

2. For any integers $k_0, k_1 \ldots k_{n-1}$ with $k_0 \in \{0, -1\}$, there exists a chromatic subdivision of an n-dimensional simplex with a symmetric binary coloring and with exactly

$$1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i$$

   monochromatic n-dimensional simplexes.

Therefore, the solvability of WSB completely depends on whether the equation

$$\binom{n+1}{1} k_0 + \binom{n+1}{2} k_2 + \ldots + \binom{n+1}{n} k_{n-1} = 1$$

has an integer solution. A well known result in number theory implies that this happens if and only if n is such that the integers in the set $\{\binom{n+1}{i+1} | 0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor\}$ are relatively prime. Such a value of n is called *exceptional*.

Thus, if n is non-exceptional, any chromatic subdivision of an n-dimensional simplex with a symmetric binary coloring, contains at least one monochromatic n-dimensional simplex. Therefore, for these values of n, there is no wait-free protocol that solves WSB. Since WSB is equivalent to $2n$-renaming, this result implies that M-renaming with $M < 2n+1$, is not wait-free solvable. This gives the first, fully combinatorial renaming lower bound proof, for non-exceptional values of n.

Now, if n is exceptional, there exists a chromatic subdivision of an n-dimensional simplex with a symmetric binary coloring and without monochromatic n-dimensional simplexes. This implies that there exists a wait-free protocol that solves WSB, hence there is a wait-free protocol that solves $2n$-renaming. Therefore, all previous renaming lower bound proofs stating that M-renaming is not wait-free solvable if $M < 2n+1$, are flawed.

More precisely, the main result states that there exists a wait-free WSB protocol if and only if n is exceptional. For example, such a protocol exists for $n = 5, 9, 11, 13, 14$, and does not exist for the other values smaller than 14. Moreover, there are infinitely many cases for which WSB is wait-free solvable.

Also, the thesis studied the relation of the combinatorial topology results 1 and 2 with classic topic of algebraic topology, more precisely, with equivariant chain maps. In addition, it explained where the mistake is in previous renaming lower bound proofs and showed how it can be fixed.

Finally, it presented a survey of most of the known results concerning renaming and WSB.

9.2 The t-Resilient Case

Although this thesis focused on the wait-free case, the results it contains have implications for the t-resilient case.

The BG simulation [19, 23] shows that a set of $t+1$ processes can wait-free simulate a system with n processes that tolerates at most t faults. Using this simulation, $t+1$ processes can use a t-resilient protocol $\mathcal{P}$ that solves $\triangle$ on n processes, to wait-free solve $\triangle$. Intuitively, the $t+1$ processes simulate the n codes of $\mathcal{P}$, and as soon as a process "sees" that one of the n simulated codes have decided, it adopts that decision as its output value. Therefore, if $\triangle$ is not wait-free solvable on $t+1$ processes, then it is not t-resilient solvable on n processes. Moreover, the other direction is true. That is, given a wait-free protocol $\mathcal{P}$ that solves $\triangle$ on $t+1$ processes, n processes can t-resilient solve $\triangle$: $t+1$ processes of n execute $\mathcal{P}$, and the rest wait for a decision of the processes executing $\mathcal{P}$; since $\mathcal{P}$ is wait-free, at least one processes decides.

However, these ideas does not work for a task in which a process cannot adopt the output (or input) of another process as its decision, such as renaming or WSB. The extended version of the BG simulation (EBG) [44] works for any task. Roughly speaking, for a task $\triangle$ on n processes, EBG creates a task $\triangle'$ on $t+1$ processes such that $\triangle$ is t-resilient solvable on n processes if and only if $\triangle'$ is wait-free solvable on $t+1$ processes. In [44] it is claimed that if $\triangle$ is $(n-t-1)$-renaming, then $\triangle'$ is WSB, considering the $2t$-resilient case. More precisely, $(n+t-1)$-renaming is $2t$-resilient solvable on n processes if and only if WSB is wait-free solvable on $t+1$ processes[1]. Therefore, the

[1]Paper [44] uses definitions of WSB and renaming that are different to the definitions used in this thesis. That paper claims without a proof that the definitions are equivalent.

results of [44] imply that $(n+t-1)$-renaming is $2t$-resilient solvable on n processes if and only if t is exceptional.

9.3 Further Research

There are three main open questions:

1. For exceptional n, exhibit an explicit code for a protocol that solves WSB. In some sense, such code does exists. In [21] it is presented a wait-free protocol that solves *simplex agreement* on a chromatic subdivision of an input complex. If the subdivision agrees with $\triangle$, then the protocol solves $\triangle$. Generally speaking, in any execution with participating set I, the processes agree on a simplex of the subdivision that is properly colored with I, and then each process decides the output value of the vertex with its *id*. Section 3.5 explained that the subdivisions constructed for exceptional values of n, give a chromatic subdivision $\chi(\mathcal{I})$ of the WSB input complex that agrees with WSB. Thus, the protocol in [21] and $\chi(\mathcal{I})$ provide a protocol that solves WSB. However, it would be good to have an explicit code that does not depend on any subdivision at all.

2. Renaming lower bound for exceptional n. Is the $2n$-renaming protocol for exceptional n optimal with respect to the size of output space? One direction for achieving a partial answer to this question is the following. Suppose n is exceptional. Notice that a $2(n-1)$-renaming protocol $\mathcal{P}$ for $n+1$ processes gives $2(n-1)$-renaming protocol $\mathcal{P}'$ for n processes. If we prove that $n-1$ is non-exceptional, this thesis shows that $\mathcal{P}'$ cannot exist. Therefore, M-renaming is not wait-free solvable when $M \leq 2(n-1)$ provided that if n is exceptional then $(n-1)$ is non-exceptional.

3. Extend the results for the t-resilient case. As explained above, if t is exceptional then $(n+t-1)$-renaming is $2t$-resilient solvable on n processes. What is the minimum t' such that $(n+t-1)$-renaming is t'-resilient solvable?

Appendix A

Subdividing Points

This section contains the proofs of the lemmas presented in Section 6.3.2. These lemmas consider a path $\mathcal{P} : \sigma_0 - \sigma_1 - \cdots - \sigma_{2q+1}$ in standard form and its subdividing point m.

Lemma 6.3.3 (Restated) *Let m be the subdividing point of $\mathcal{P}$. Then $1 \leq m \leq min(q, n+1)$.*

Proof: First, for $m = 0$ it is not possible $\#0(\sigma_{m+1,m+2}) \geq n+1-m = n+1$ because $dim(\sigma_{m+1,m+2}) = n$. If $n+1 \leq q$ then m is at most $n+1$ because if $m = n+1$ then $\#0(\sigma_{m+1,m+2}) \geq n+1-m = 0$. Now, if $q < n+1$ then σ_{q+2} has at most $q-1$ vertexes with binary coloring 1 because there are $q-1$ simplexes in $\mathcal{P}_{q+3,2q+1}$ and $\#0(\sigma_{2q+1}) = n+1$. Thus, $\#0(\sigma_{q+2}) \geq n+2-q$ and hence $\#0(\sigma_{q+1,q+2}) \geq n+2-q$ (any $(n-1)$-face of σ_{q+2} has all the vertexes of σ_{q+2} with binary color 0 or all of them but one). Therefore, m is at most q.

$\blacksquare$

Lemma 6.3.4 (Restated) *Let m be the subdividing point of $\mathcal{P}$. Then $|\mathcal{P}| \geq 2(m+1)$.*

Proof: By the definition of subdividing point, Definition 6.3.2, $\#0(\sigma_{m+1,m+2}) \geq n+1-m$. We are interested in the case which σ_{m+2} has as many as possible vertexes with binary color 0. (if σ_{m+2} has fewer vertexes with binary color 0, the segment $\mathcal{P}_{m+2,2q+1}$ is longer because σ_{2q+1} is 0-monochromatic) Notice that if $\#0(\sigma_{m+1,m+2}) = n+3-m$ then $\#0(\sigma_{m+1}) \geq n+3-m$ and hence $\#0(\sigma_{m,m+1}) \geq n+2-m = n+1-(m-1)$, contradicting that m is the subdividing point of $\mathcal{P}$, Definition 6.3.2. Therefore, in the best case $\#0(\sigma_{m+1,m+2}) = n+2-m$, and thus $\#0(\sigma_{m+2}) = n+3-m$, i.e., σ_{m+2} has $m-2$ vertexes with binary color 1. The latter implies that $|\mathcal{P}_{m+3,2q+1}| = m-2$. Also, we have that $|\mathcal{P}_{0,m+2}| = m+3$. Thus $|\mathcal{P}| = 2m+1$, however, $|\mathcal{P}|$ must be even. Therefore, we have that $|\mathcal{P}| = 2(m+1)$.

$\blacksquare$

Lemma 6.3.5 (Restated) *For the subdividing point m of $\mathcal{P}$:*

$$
\begin{array}{ccccc}
n+2-m & n+2-m & n+1-m & n+2-m & n+2-m \\
n+1-m & n+1-m & n-m & n+1-m & n+1-m \\
\rule{3cm}{0.4pt} & \sigma_m & \rule{3cm}{0.4pt} & \sigma_{m+1} & \rule{3cm}{0.4pt}
\end{array}
$$

Proof: By the definition of subdividing point, Definition 6.3.2, m is the smallest value such that $\#0(\sigma_{m+1,m+2}) \geq n+1-m$, thus we have $\#0(\sigma_{m,m+1}) < n+1-(m-1) = n+2-m$. Observe that if $\#0(\sigma_{m+1,m+2}) \geq n+3-m$ then $\#0(\sigma_{m+1}) \geq n+3-m$, and hence $\#0(\sigma_{m,m+1}) \geq n+2-m$. Therefore, $\#0(\sigma_{m+1,m+2}) \in \{n+1-m, n+2-m\}$ and $\#0(\sigma_{m+1}) \in \{n+1-m, n+2-m\}$. Moreover, since $\#0(\sigma_{m+1}) \in \{n+1-m, n+2-m\}$ and $\#0(\sigma_{m,m+1}) < n+2-m$, then $\#0(\sigma_{m,m+1}) \in \{n-m, n+1-m\}$. Now, for an n-simplex σ_x of $\mathcal{P}$, we have $\#0(\sigma_x) \geq n+1-x$ because σ_0 is 0-monochromatic. Therefore, $\#0(\sigma_{m-1}) \geq n+2-m$ and $\#0(\sigma_m) \geq n+1-m$, and hence $\#0(\sigma_{m-1,m}) \geq n+1-m$. Since $\#0(\sigma_{m,m+1}) \in \{n-m, n+1-m\}$ and $\#0(\sigma_{m-1,m}) \geq n+1-m$, then $\#0(\sigma_m) \in \{n+1-m, n+2-m\}$, thus $\#0(\sigma_{m-1,m}) \in \{n+1-m, n+2-m\}$.

$\blacksquare$

Appendix B

Correctness Proof of SubdivideGood

First we present two simple lemmas about Function **SubdivideGood** that are useful for proving Lemmas 6.3.16 and 6.3.17.

Lemma B.1 *Consider any execution of **SubdivideGood** with a path in standard form $\mathcal{P}$ and the subdividing point m of $\mathcal{P}$. Consider some $i \in \{0,1\}$ and let $\bar{i}$ be $(i+1) \mod 2$. If $\#i(\sigma_x) = \#i(\sigma_{x,x-\xi})$ then $k_{x-\xi} \in id(\bar{i}(\sigma_{x,x+\xi}))$, otherwise $k_{x-\xi} \in id(i(\sigma_{x,x+\xi}))$. Similarly, if $\#i(\sigma_{x+\xi}) = \#i(\sigma_{x+\xi,x+2\xi})$ then $k_{x+2\xi} \in id(\bar{i}(\sigma_{x,x+\xi}))$, otherwise $k_{x+2\xi} \in id(i(\sigma_{x,x+\xi}))$.*

Proof: First, by the specification of **ConfigVars**, Definition 6.3.19, and since $\mathcal{P}$ and m are valid inputs to **SubdivideGood**, $\sigma_{x-\xi,x}$, σ_x, $\sigma_{x,x+\xi}$, $\sigma_{x+\xi}$ and $\sigma_{x+\xi,x+2\xi}$ are simplexes of $\mathcal{P}$. Consider the unique vertex v of $\sigma_x \setminus \sigma_{x,x-\xi}$. Notice that $k_{x-\xi} = id(v)$ (recall that $k_{x-\xi}$ is the id color changed in the step $\sigma_{x-\xi} - \sigma_x$). if $\#i(\sigma_x) = \#i(\sigma_{x,x-\xi})$ then $b(v) = \bar{i}$ and thus $id(v) \in id(\bar{i}(\sigma_{x,x+\xi}))$. Otherwise $b(v) = i$ and hence $id(v) \in id(i(\sigma_{x,x+\xi}))$. The proof for $\sigma_{x+\xi}, \sigma_{x+\xi,x+2\xi}$ and $k_{x+2\xi}$ is identical.

■

Lemma B.2 *Consider any execution of **SubdivideGood** with a path in standard form $\mathcal{P}$ and the subdividing point m of $\mathcal{P}$. Consider some $i \in \{0,1\}$ and let $\bar{i}$ be $(i+1) \mod 2$. At line 9 we have the following:*

1. *$id(i(\tau)) = id(\bar{i}(\sigma_{x,x+\xi}))$*
2. *$(\sigma_{x,x+\xi})_{-id(i(\tau))}$ is i-monochromatic*
3. *for any face τ' of $\bar{i}(\tau)$, $(\sigma_{x,x+\xi})_{-id(i(\tau)\cup\tau')}$ is i-monochromatic*
4. *for any proper face τ' of $i(\tau)$, $(\sigma_{x,x+\xi})_{-id(\tau')}$ is not i-monochromatic*

Proof: By the specification of **ConfigVars**, Definition 6.3.19, and since $\mathcal{P}$ and m are valid inputs to **SubdivideGood**, σ_x, $\sigma_{x,x+\xi}$ and $\sigma_{x+\xi}$ are simplexes of $\mathcal{P}$. Now, observe that after line 7 we have $id(0(\tau)) = id(1(\sigma_{x,x+\xi}))$ and $id(1(\tau)) = id(0(\sigma_{x,x+\xi}))$, i.e., $id(i(\tau)) = id(\bar{i}(\sigma_{x,x+\xi}))$. Therefore $(\sigma_{x,x+\xi})_{-id(i(\tau))} = i(\sigma_{x,x+\xi})$ and so $(\sigma_{x,x+\xi})_{-id(i(\tau))}$ is i-monochromatic. Consider now a face τ' of $\bar{i}(\tau)$. Notice that $(\sigma_{x,x+\xi})_{-id(i(\tau)\cup\tau')}$ is a face of $(\sigma_{x,x+\xi})_{-id(i(\tau))}$ and hence $(\sigma_{x,x+\xi})_{-id(i(\tau)\cup\tau')}$ is

i-monochromatic. Finally, let τ' be a proper face of $i(\tau)$. Consider an element $k \in id(i(\tau) \setminus \tau')$. Let v be the vertex of $\sigma_{x,x+\xi}$ such that $id(v) = k$. Notice that $v \in (\sigma_{x,x+\xi})_{-id(\tau')}$. Also we have that $b(v) = \bar{i}$ because $id(i(\tau)) = id(\bar{i}(\sigma_{x,x+\xi}))$. Thus $(\sigma_{x,x+\xi})_{-id(\tau')}$ is not i-monochromatic.

■

Lemma 6.3.16 (Restated) *If* ***SubdivideGood*** *is invoked with a path $\mathcal{P}$ in standard form and the subdividing point m of $\mathcal{P}$ is progressive then it produces a good subdivision of $\mathcal{P}$ with $|\mathcal{P}_i| < |\mathcal{P}|$, $i \in \{1, 2\}$.*

Proof: Let $\mathcal{DCC}^n$ denote $v_x * v_{x+\xi} * (\tau \circledast bd(\sigma_{x,x+\xi}))$ in line 8. Observe that neither of $\mathcal{DCC}^n$, **SubdivideComp** and **Disconnect**, affects $bd(\mathcal{P})$. Also, by Lemma 6.3.4, $|\mathcal{P}| \geq 2(m + 1)$.

Case A. By the specification of **ConfigVars**, Definition 6.3.19, $x = m$ and $\xi = +1$. Also by Lemma 6.3.6, $\#0(\sigma_x) = \#0(\sigma_{x+\xi}) = n + 1 - m$ and $\#0(\sigma_{x,x+\xi}) = n - m$, and hence $b(v_x) = b(v_{x+\xi}) = 0$ and $\#1(\sigma_{x,x+\xi}) = m$. In line 10 we have that $\tau_1 = 0(\tau)$. By case 1 of Lemma B.2, $|\tau_1| = m$ because $id(\tau_1) = id(0(\tau)) = id(1(\sigma_{x,x+\xi}))$. By case 2 of Lemma B.2, $(\sigma_{x,x+\xi})_{-id(\tau_1)}$ is 0-monochromatic and thus $v_x * [\tau_1]$ of $\mathcal{DCC}^n$ is 0-monochromatic, by Lemma 6.3.9. By Lemma 6.3.6, $\#0(\sigma_{x-1,x}) = n + 1 - m$ and thus $k_{x-\xi} \in id(1(\sigma_{x,x+\xi})) = id(\tau_1)$, by Lemma B.1. Moreover, since $id(\tau_1) = id(1(\sigma_{x,x+\xi}))$, $(\sigma_{x,x+\xi})_{+id(\tau_1)}$ is 1-monochromatic and hence $(\sigma_{x,x+\xi})_{+id(\tau_1)\setminus k_{x-\xi}}$ is 1-monochromatic. Also notice that path $\mathcal{P}_1$ in line 11 is a left non-crossing path. By case 1.a of Lemma 6.3.12, $\mathcal{P}_1$ is in standard form and $|\mathcal{P}_1| = 2m$.

For $\mathcal{P}_2$, line 14, we have that $|\tau_2| = m$ because $\tau_2 = \tau_1$ after line 13. And by Lemma 6.3.9, $v_{x+\xi} * [\tau_2]$ is 0-monochromatic. By Lemma 6.3.6, $\#0(\sigma_{x+\xi,x+2\xi}) = n + 1 - m$ and thus $k_{x+2\xi} \in id(1(\sigma_{x,x+\xi})) = id(\tau_2)$, by Lemma B.1. Moreover, $(\sigma_{x,x+\xi})_{+id(\tau_2)}$ is 1-monochromatic and hence $(\sigma_{x,x+\xi})_{+id(\tau_2)\setminus k_{x+2\xi}}$ is 1-monochromatic. Notice that $\mathcal{P}_2$ is a right non-crossing path and it also is in standard form with $|\mathcal{P}_2| = |\mathcal{P}| - 2$ by case 1.a of Lemma 6.3.12. Observe that $\mathcal{P}_1$ and $\mathcal{P}_2$ do not share n-simplexes.

We now prove that $v_x * [\tau_1]$ and $v_{x+\xi} * [\tau_2]$ are the unique monochromatic n-simplexes of $\mathcal{DCC}^n$. First, by Lemma 6.3.9, $\mathcal{DCC}^n$ does not have 1-monochromatic n-simplexes because $b(v_x) = b(v_{x+\xi}) = 0$. Second, for every face τ' of τ such that $0(\tau) \subset \tau'$, there is a vertex of $v \in \tau'$ with $b(v) = 1$. Thus $[\tau']$ of $\mathcal{DCC}^n$ is not 0-monochromatic and hence any n-simplex of $\mathcal{DCC}^n$ containing $[\tau']$ is not 0-monochromatic, by Lemma 6.3.9. And third, by case 4 of Lemma B.2, for each proper face τ' of $0(\tau)$, $(\sigma_{x,x+\xi})$ is not 0-monochromatic and thus any n-simplex of $\mathcal{DCC}^n$ containing $[\tau']$ is not 0-monochromatic, by Lemma 6.3.9. This complete the proof for case A.

Case C. First, Figure B.1 depicts an example of how **SubdivideGood** works on a 2-dimensional path with subdividing point holding case C. Now, by the specification of **ConfigVars**, Definition 6.3.19, $x = m$ and $\xi = +1$. Also by Lemma 6.3.6, $\#0(\sigma_x) = n+2-m$ and $\#0(\sigma_{x+\xi}) = \#0(\sigma_{x,x+\xi}) = n + 1 - m$, and hence $b(v_x) = 0$, $b(v_{x+\xi}) = 1$ and $\#1(\sigma_{x,x+\xi}) = m - 1$. For both $i \in \{0, 1\}$, let $\lambda_i = i(\tau)$ in line 8. By case 1 of Lemma B.2, $id(\lambda_i) = id(\bar{i}(\sigma_{x,x+\xi}))$, where $\bar{i} = (i + 1) \bmod 2$. Therefore, $|\lambda_0| = m - 1$. Also, by Lemma 6.3.8, we have that $2 \leq m \leq n$ and thus $\lambda_1 \neq \emptyset$. Now, by Lemma 6.3.6, $\#0(\sigma_{x-\xi,x}) \in \{n+2-m, n+1-m\}$. By Lemma B.2 and since $id(\lambda_0) = id(1(\sigma_{x,x+\xi}))$, if $\#0(\sigma_{x-\xi,x}) = n + 2 - m$ then $k_{x-\xi} \in id(\lambda_0)$, otherwise $k_{x-\xi} \notin id(\lambda_0)$. After line 26 we have that $\tau_1 = \lambda_0 \cup \{u_{x-\xi}\}$ and $k_{x-\xi} \in id(\tau_1)$ (for case C we are not interested in $u_{x+2\xi}$). Also $|\tau_1| = m$ and

τ_1 is 0-monochromatic. By case 3 of Lemma B.2, $(\sigma_{x,x+\xi})_{-id(\tau_1)}$ is 0-monochromatic. By Lemma 6.3.9, $v_x * [\tau_1]$ of $\mathcal{DCC}^n$ is 0-monochromatic. Notice that path $\mathcal{P}_1$ at line 27 is a left non-crossing path starting from $v_x * [\tau_1]$. We now check which case of Lemma 6.3.12 can be applied on $\mathcal{P}_1$. Let v be the vertex of $\sigma_{x,x+\xi}$ with id color $id(u_{x-\xi})$. Since $id(\lambda_1) = id(0(\sigma_{x,x+\xi}))$, we have that $b(v) = 0$. For $k_{x-\xi}$, if line 21 was true then $id(v) = k_{x-\xi}$. Therefore $v \notin (\sigma_{x,x+\xi})_{+id(\tau_1)\setminus k_{x-\xi}}$ and so $(\sigma_{x,x+\xi})_{+id(\tau_1)\setminus k_{x-\xi}}$ is 1-monochromatic. Thus, by case 1.a of Lemma 6.3.12, $\mathcal{P}_1$ is in standard form and $|\mathcal{P}_1| = 2m$. Now if line 21 was false then $id(v) \neq k_{x-\xi}$. Therefore $v \in (\sigma_{x,x+\xi})_{+id(\tau_1)\setminus k_{x-\xi}}$ and so $(\sigma_{x,x+\xi})_{+id(\tau_1)\setminus k_{x-\xi}}$ is not 1-monochromatic. However, observe that $(\sigma_{x,x+\xi})_{+id(\tau_1)\setminus\{k_{x-\xi},id(u_{x-\xi})\}}$ is 1-monochromatic. By case 1.b of Lemma 6.3.12, $\mathcal{P}_1$ is in standard or quasistandard form and $|\mathcal{P}_1| = 2m$. Also, if $\mathcal{P}_1$ is in quasistandard form then $v_x*[\tau_{1-id(u_{x-\xi})}]$ is the monochromatic n-simplex of $\mathcal{P}_1$ that is not at its ends. Notice that $\tau_{1-id(u_{x-\xi})} = \lambda_0$. We shall prove that $v_x * [\tau_{1-id(u_{x-\xi})}]$ is one the monochromatic n-simplexes at the ends of $\mathcal{P}_2$, line 30.

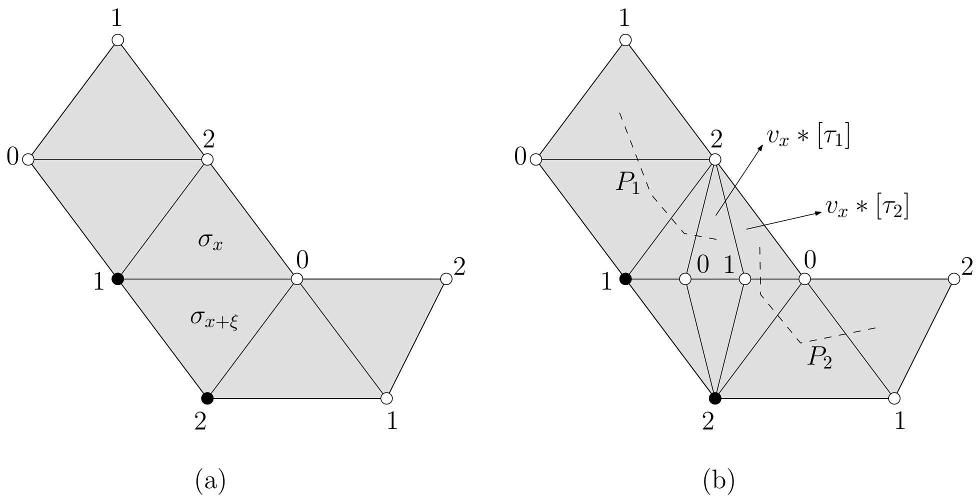

Figure B.1: **SubdivideGood** working on a 2-dimensional path with subdividing point holding case C.

For path $\mathcal{P}_2$ we have the following. Observe that $\tau_2 = \lambda_0$, line 29, and also $|\tau_2| = m - 1$ because $id(\lambda_0) = id(1(\sigma_{x,x+\xi}))$. Moreover, τ_2 and $(\sigma_{x,x+\xi})_{-id(\tau_2)}$ are 0-monochromatic. By Lemma 6.3.9, $v_x * [\tau_2]$ of $\mathcal{DCC}^n$ is 0-monochromatic. Also by Lemma 6.3.6, $\#0(\sigma_{x+\xi,x+2\xi}) = n + 1 - m$ and hence $k_{x+2\xi} \in id(\tau_2)$, by Lemma B.1. By case 2 of Lemma 6.3.12, $\mathcal{P}_2$ is in standard form and $|\mathcal{P}_2| = |\mathcal{P}| - 2$. Now, notice that $v_x*[\tau_2] = v_x*[\tau_{1-id(u_{x-\xi})}]$. Therefore if $\mathcal{P}_1$ is in quasistandard form, $\mathcal{P}_1$ and $\mathcal{P}_2$ share $v_x * [\tau_2]$. However, by Lemma 6.3.20, function **Disconnect** in line 32 produces a good subdivision of $\mathcal{P}_1 \cup \mathcal{P}_2$ with boundary $bd(\mathcal{P}_1 \cup \mathcal{P}_2)$ and paths of the same length as inputs $\mathcal{P}_1$ and $\mathcal{P}_2$.

We now prove that the subdivisions **SubdivideGood** did on $\mathcal{P}$ (lines 8 and 32), only produced the monochromatic n-simplexes for $\mathcal{P}_1$ and $\mathcal{P}_2$. For all $i \in \{0, 1\}$, let $\lambda_i = i(\tau)$ at line 8. Consider the execution in line 31. We have that $\tau_1 = \lambda_0 \cup \{u_{k-\xi}\}$ and $\tau_2 = \lambda_0$. Let τ' be a face of τ such that $\lambda_0 \subset \tau'$ and $\tau' \neq \tau_1$. Notice that there exists $u, v \in \tau'$ such that $b(v) = 0$ and $b(u) = 1$. Therefore $[\tau']$ is not monochromatic and hence neither $v_x * [\tau']$ nor $v_{x+\xi} * [\tau']$ of $\mathcal{DCC}^n$ are monochromatic, by Lemma 6.3.9. By case 4 of Lemma B.2, for each non-empty proper face τ' of λ_0, $(\sigma_{x,x+\xi})$ is not 0-monochromatic. Moreover, observe that τ' is not 1-monochromatic. By Lemma 6.3.9, neither

$v_x * [\tau']$ nor $v_{x+\xi} * [\tau']$ of $\mathcal{DCC}^n$ are monochromatic. That is, $v_x * [\tau_1]$ and $v_x * [\tau_2]$ are the unique monochromatic n-simplexes of $\mathcal{DCC}^n$ at line 31. If line 32 is executed, Lemma 6.3.20 guarantees that **Disconnect** only produces the two monochromatic n-simplexes for resulting paths $\mathcal{P}_1$ and $\mathcal{P}_2$. This complete the proof for case C.

Case D. This case is symmetric to case C and hence its proof is almost identical. The difference is that in the end of the analysis, $|\mathcal{P}_1| = |\mathcal{P}| - 2$ and $|\mathcal{P}_2| = 2m$.

Case E with $m < n+1$. First, Figure B.2 shows an example of how **SubdivideGood** works on 2-dimensional path with subdividing point holding case E with $m < n+1$. By the specification of **ConfigVars**, Definition 6.3.19, $x = m$ and $\xi = +1$. Also by Lemma 6.3.6, $\#0(\sigma_x) = \#0(\sigma_{x+\xi}) = n+2-m$ and $\#0(\sigma_{x,x+\xi}) = n+1-m$, and hence $b(v_x) = b(v_{x+\xi}) = 0$ and $\#1(\sigma_{x,x+\xi}) = m-1$. For all $i \in \{0,1\}$, let $\lambda_i = i(\tau)$ at line 8. By case 1 of Lemma B.2, $id(\lambda_i) = id(\bar{i}(\sigma_{x,x+\xi}))$, where $\bar{i} = (i+1) \bmod 2$. Therefore, $|\lambda_0| = m-1$. Since $m \leq n$, $\lambda_1 \neq \emptyset$. Now, by Lemma 6.3.6, $\#0(\sigma_{x-\xi,x}) \in \{n+2-m, n+1-m\}$. Doing the same analysis than the one used in case C, we can prove that $\mathcal{P}_1$ in line 27 is in standard or quasistandard form and $|\mathcal{P}_1| = 2m$. Also, if $\mathcal{P}_1$ is in quasistandard form then $v_x * [\tau_{1-id(u_{x-\xi})}]$ is the monochromatic n-simplex of $\mathcal{P}_1$ that is not at its ends. And $\tau_1 = \lambda_0 \cup \{u_{x-\xi}\}$.

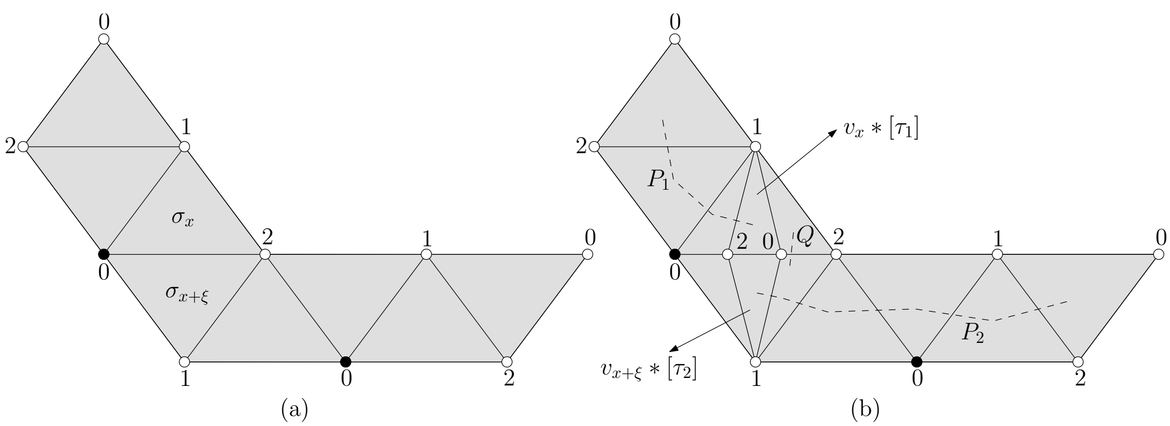

Figure B.2: **EliminatePath** working on a 2-dimensional path with subdividing point m holding case E and $m < n+1$.

We now see what happens with path $\mathcal{P}_2$ in line 38. The analysis is very similar to the one used for $\mathcal{P}_1$. By Lemma 6.3.6, $\#0(\sigma_{x+\xi,x+2\xi}) \in \{n+2-m, n+1-m\}$. By Lemma B.2 and since $id(\lambda_0) = id(1(\sigma_{x,x+\xi}))$, if $\#0(\sigma_{x-\xi,x}) = n+2-m$ then $k_{x+2\xi} \in id(\lambda_0)$, otherwise $k_{x+2\xi} \notin id(\lambda_0)$. After line 36 we have that $\tau_2 = \lambda_0 \cup \{u_{x+2\xi}\}$ and $k_{x+2\xi} \in id(\tau_2)$. Also $|\tau_2| = m$ and τ_2 is 0-monochromatic. By case 3 of Lemma B.2, $(\sigma_{x,x+\xi})_{-id(\tau_2)}$ is 0-monochromatic. By Lemma 6.3.9, $v_{x+\xi} * [\tau_2]$ of $\mathcal{DCC}^n$ is 0-monochromatic. Notice that path $\mathcal{P}_2$ is a right non-crossing path starting from $v_{x+\xi} * [\tau_2]$. Let v be the vertex of $\sigma_{x,x+\xi}$ with id color $id(u_{x+2\xi})$. Since $id(\lambda_1) = id(0(\sigma_{x,x+\xi}))$, we have that $b(v) = 0$. For $k_{x+2\xi}$, if line 21 was true then $id(v) = k_{x+2\xi}$. Therefore $v \notin (\sigma_{x,x+\xi})_{+id(\tau_2)\setminus k_{x+2\xi}}$ and so $(\sigma_{x,x+\xi})_{+id(\tau_2)\setminus k_{x+2\xi}}$ is 1-monochromatic. Thus, by case 1.a of Lemma 6.3.12, $\mathcal{P}_2$ is in standard form and $|\mathcal{P}_2| = |\mathcal{P}| - 2$. Now if line 21 was false then $id(v) \neq k_{x+2\xi}$. Therefore $v \in (\sigma_{x,x+\xi})_{+id(\tau_2)\setminus k_{x+2\xi}}$ and so $(\sigma_{x,x+\xi})_{+id(\tau_2)\setminus k_{x+2\xi}}$ is not 1-monochromatic. However, $(\sigma_{x,x+\xi})_{+id(\tau_2)\setminus \{k_{x+2\xi}, id(u_{x+2\xi})\}}$ is

1-monochromatic. By case 1.b of Lemma 6.3.12, $\mathcal{P}_2$ is in standard or quasistandard form and $|\mathcal{P}_2| = |\mathcal{P}| - 2$. Also, if $\mathcal{P}_2$ is in quasistandard form then $v_{x+\xi} * [\tau_{2-id(u_{x+2\xi})}]$ is the monochromatic n-simplex of $\mathcal{P}_2$ that is not at its ends.

We now prove that the subdivisions **SubdivideGood** did on $\mathcal{P}$ (lines 8, 40, 42, 43, 48 and 49), only produced the monochromatic n-simplexes $v_x * [\tau_1]$ and $v_{x+\xi} * [\tau_2]$ for $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively. Consider the state of the execution at line 38. Let $\gamma_1 = \{u_{x-\xi}\}$, $\gamma_2 = \{u_{x+2\xi}\}$ and $\gamma_3 = \{u_{x-\xi}, u_{x+2\xi}\}$. Notice that $\tau_0 = \lambda_0$, $\tau_1 = \lambda_0 \cup \gamma_1$ and $\tau_2 = \lambda_0 \cup \gamma_2$. Therefore $v_x * [\tau_1] = v_x * [\lambda_0 \cup \gamma_1]$ and $v_{x+\xi} * [\tau_2] = v_{x+\xi} * [\lambda_0 \cup \gamma_2]$. We proved above that $v_x * [\lambda_0 \cup \gamma_1]$ is 0-monochromatic and thus $v_{x+\xi} * [\lambda_0 \cup \gamma_1]$ is 0-monochromatic. Similarly, $v_x * [\lambda_0 \cup \gamma_2]$ is 0-monochromatic because we proved that $v_{x+\xi} * [\lambda_0 \cup \gamma_2]$ is 0-monochromatic. Now, we have that λ_0 is 0-monochromatic and so $(\sigma_{x,x+\xi})_{-id(\lambda_0)}$ is 0-monochromatic. Also $(\sigma_{x,x+\xi})_{-id(\lambda_0)}$ and $(\sigma_{x,x+\xi})_{-id(\lambda_0 \cup \gamma_3)}$ are 0-monochromatic because $id(\lambda_0) = id(1(\sigma_{x,x+\xi}))$. Since $b(v_x) = b(v_{x+\xi}) = 0$ and by Lemma 6.3.9, the n-simplexes $v_x * [\lambda_0]$, $v_{x+\xi} * [\lambda_0]$, $v_x * [\lambda_0 \cup \gamma_3]$ and $v_{x+\xi} * [\lambda_0 \cup \gamma_3]$ are 0-monochromatic. We now prove that these 8 n-simplexes are all the monochromatic n-simplexes of $\mathcal{DCC}^n$. By Lemma 6.3.9, $\mathcal{DCC}^n$ does not have 1-monochromatic n-simplexes because $b(v_x) = b(v_{x+\xi}) = 0$. Also for every face τ' of τ such that $\lambda_0 \subset \tau'$ and $\gamma_j \cap \tau' = \emptyset$ for all $j \in \{1, 2, 3\}$, there is a vertex of $v \in \tau'$ with $b(v) = 1$. Thus $[\tau']$ of $\mathcal{DCC}^n$ is not 0-monochromatic and hence any n-simplex of $\mathcal{DCC}^n$ containing $[\tau']$ is not 0-monochromatic, by Lemma 6.3.9. And by case 4 of Lemma B.2, for each proper face τ' of λ_0, $(\sigma_{x,x+\xi})$ is not 0-monochromatic and thus any n-simplex of $\mathcal{DCC}^n$ containing $[\tau']$ is not 0-monochromatic, by Lemma 6.3.9.

Now, consider the path $\mathcal{P}_0 : v_x * [\lambda_0] - v_{x+\xi} * [\lambda_0]$ in line 37. If $\mathcal{P}_1$ is in quasistandard form then $v_x * [\tau_{1-id(u_{x-\xi})}]$ is the monochromatic n-simplex of $\mathcal{P}_1$ that is not at its ends. Observe that $\tau_{1-id(u_{x-\xi})} = \lambda_0$ and hence $\mathcal{P}_1$ and $\mathcal{P}_0$ share $v_x * [\lambda_0]$. Similarly, if $\mathcal{P}_2$ is in quasistandard form then $v_{x+\xi} * [\tau_{2-id(u_{x+2\xi})}]$ is the monochromatic n-simplex of $\mathcal{P}_2$ that is not at its ends. We have that $\tau_{2-id(u_{x+2\xi})} = \lambda_0$ and so $\mathcal{P}_2$ and $\mathcal{P}_0$ share $v_{x+\xi} * [\lambda_0]$. Anyway, $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_0$ do not share n-simplexes in line 43, by Lemma 6.3.20. And **SubdivideComp** in line 43 produces a complete subdivision of $\mathcal{P}_0$, by Lemma 6.3.15. Consider now the line 44. if $u_{x-\xi} = u_{x+2\xi}$ then $\gamma_1 = \gamma_2 = \gamma_3$ and thus there is nothing more to do. Consider now the case $u_{x-\xi} \neq u_{x+2\xi}$. Observe that $\tau_3 = \lambda_0 \cup \gamma_3$, line 46. Moreover, we have that $\mathcal{Q}_1 : v_x * [\lambda_0 \cup \gamma_2] - v_x * [\lambda_0 \cup \gamma_3]$ and $\mathcal{Q}_2 : v_{x+\xi} * [\lambda_0 \cup \gamma_1] - v_{x+\xi} * [\lambda_0 \cup \gamma_3]$, lines 47 and 48. By Lemma 6.3.15, lines 48 and 49 produce a complete chromatic subdivision of $\mathcal{Q}_1$ and $\mathcal{Q}_2$. This complete the proof for case E.

■

Lemma 6.3.17 (Restated) *If **SubdivideGood** is invoked with a path $\mathcal{P}$ in standard form and the subdividing point m of $\mathcal{P}$ is non-progressive then it produces a good subdivision of $\mathcal{P}$ with $|\mathcal{P}_i| \leq |\mathcal{P}|$, $i \in \{1, 2\}$. And if for some $i \in \{1, 2\}$, $|\mathcal{P}_i| = |\mathcal{P}|$ then the subdividing point of $\mathcal{P}_i$ is progressive.*

Proof: Let $\mathcal{DCC}^n$ denote $v_x * v_{x+\xi} * (\tau \circledast bd(\sigma_{x,x+\xi}))$ in line 8. Observe that neither of $\mathcal{DCC}^n$, **SubdivideComp** and **Disconnect**, affects $bd(\mathcal{P})$. Also, by Lemma 6.3.4, $|\mathcal{P}| \geq 2(m + 1)$.

Case B. By Lemma 6.3.6, $\#0(\sigma_m) = \#0(\sigma_{m+1}) = \#0(\sigma_{x,x+\xi}) = n + 1 - m$ and hence $b(v_m) = b(v_{m+1}) = 1$, where v_m and v_{m+1} are the unique vertexes of $\sigma_m \setminus \sigma_{m,m+1}$ and $\sigma_{m+1} \setminus \sigma_{m,m+1}$ respectively. Therefore it is not possible to produce 1-monochromatic n-simplexes by subdividing on $\sigma_{m,m+1}$. **SubdivideGood** handles this problem by subdividing on $\sigma_{m,m-1}$. By Lemma 6.3.8,

$m \geq 2$ and hence σ_{m-1} and σ_{m-2} are simplexes of $\mathcal{P}$. Notice that for each n-simplex σ_y of $\mathcal{P}$, $\#0(\sigma_y) \geq n+1-y$. Thus, $\#0(\sigma_{m-2}) \geq n+3-m$ and $\#0(\sigma_{m-1}) \geq n+2-m$. Therefore, $\#0(\sigma_{m-2}) = n+3-m$ and $\#0(\sigma_{m-1}) = n+2-m$, because $\#0(\sigma_{m-1,m}) = n+1-m$ by Lemma 6.3.6. Moreover, $\#0(\sigma_{m-2,m-1}) = n+2-m$.

By the specification of **ConfigVars**, Definition 6.3.19, $x = m-1$ and $\xi = +1$. Therefore $\#0(\sigma_x) = n+2-m$ and $\#0(\sigma_{x+\xi}) = \#0(\sigma_{x,x+\xi}) = n+1-m$, and hence $b(v_x) = 0$, $b(v_{x+\xi}) = 1$ and $\#1(\sigma_{x,x+\xi}) = m-1$. For $i \in \{0,1\}$, let $\lambda_i = i(\tau)$ in line 8. By case 1 of Lemma B.2, $id(\lambda_i) = id(\bar{i}(\sigma_{x,x+\xi}))$, where $\bar{i} = (i+1) \bmod 2$. Therefore, $|\lambda_0| = m-1$. Since $m \leq n$, then $\lambda_1 \neq \emptyset$. In line 10 we have that $\tau_1 = \lambda_0$. By case 1 of Lemma B.2 and since $id(\tau_1) = id(\lambda_0) = id(1(\sigma_{x,x+\xi}))$, $|\tau_1| = m-1$. By case 2 of Lemma B.2, $(\sigma_{x,x+\xi})_{-id(\tau_1)}$ is 0-monochromatic, and thus $v_x * [\tau_1]$ of $\mathcal{DCC}^n$ is 0-monochromatic, by Lemma 6.3.9. Also, since $\#0(\sigma_{x-1,x}) = n+2-m$ and by Lemma B.1, we have that $k_{x-\xi} \in id(1(\sigma_{x,x+\xi})) = id(\tau_1)$. Moreover, since $id(\tau_1) = id(1(\sigma_{x,x+\xi}))$, $(\sigma_{x,x+\xi})_{+id(\tau_1)}$ is 1-monochromatic and hence $(\sigma_{x,x+\xi})_{+id(\tau_1)\setminus k_{x-\xi}}$ is 1-monochromatic. $\mathcal{P}_1$ in line 11 is a left non-crossing path. By case 1.a of Lemma 6.3.12, $\mathcal{P}_1$ is in standard form and $|\mathcal{P}_1| = 2(m-1)$.

For $\mathcal{P}_2$, line 19, we have that $|\tau_2| = m$ because $\tau_2 = \tau_1 \cup \{u_2\} = \lambda_0 \cup \{u_2\}$, line 18. Notice that τ_2 is 0-monochromatic. And by Lemma 6.3.9, $v_{x+\xi} * [\tau_2]$ is 0-monochromatic. By case 3 of Lemma B.2, $(\sigma_{x,x+\xi})_{-id(\tau_2)}$ is 0-monochromatic. By Lemma 6.3.9, $v_x * [\tau_1]$ of $\mathcal{DCC}^n$ is 0-monochromatic. Also, since $\#0(\sigma_{x+1,x+2}) = n+1-m$ and by Lemma B.1, $k_{x+2\xi} \in id(1(\sigma_{x,x+\xi}))$ and so $k_{x+2\xi} \in id(\tau_2)$. $\mathcal{P}_2$ is a right crossing path, and hence it is in standard form with $|\mathcal{P}_2| = |\mathcal{P}|$, by case 2 of Lemma 6.3.12. Observe that $\mathcal{P}_1$ and $\mathcal{P}_2$ do not share n-simplexes.

We now prove that $v_x * [\tau_1]$ and $v_x * [\tau_2]$ are the unique monochromatic n-simplexes of $\mathcal{DCC}^n$. We have that $\tau_1 = \lambda_0$ and $\tau_2 = \lambda_0 \cup \{u_2\}$. Let τ' be a face of τ such that $\lambda_0 \subset \tau'$ and $\tau' \neq \tau_2$. Notice that there exists $u, v \in \tau'$ such that $b(v) = 0$ and $b(u) = 1$. Therefore $[\tau']$ is not monochromatic and hence neither $v_x * [\tau']$ nor $v_{x+\xi} * [\tau']$ of $\mathcal{DCC}^n$ are monochromatic, by Lemma 6.3.9. By case 4 of Lemma B.2, for each non-empty proper face τ' of λ_0, $(\sigma_{x,x+\xi})$ is not 0-monochromatic. Moreover, observe that τ' is not 1-monochromatic. By Lemma 6.3.9, neither $v_x * [\tau']$ nor $v_{x+\xi} * [\tau']$ of $\mathcal{DCC}^n$ are monochromatic. Therefore $v_x * [\tau_1]$ and $v_x * [\tau_2]$ are the unique monochromatic n-simplexes of $\mathcal{DCC}^n$.

For proving that the subdividing point of $\mathcal{P}_2$ is progressive, let γ be the very previous n-simplex to $\sigma_{x+2\xi}$. Therefore $\mathcal{P}_2 : v_x * [\tau_2] - v_{x+\xi} * [\tau_2] - \cdots - \gamma - \sigma_{x+2\xi} - \sigma_{x+3\xi} - \cdots$. Since $|\mathcal{P}_2| = |\mathcal{P}|$, the length of the subpath $\mathcal{Q} : v_x * [\tau_2] - v_{x+\xi} * [\tau_2] - \cdots - \gamma$ is $m+1$, and hence there are m steps in $\mathcal{Q}$. The step from γ to $\sigma_{x+2\xi}$ changes the id color $k_{x+2\xi}$ and thus $id(u_2)$ is changed in some step of $v_{x+\xi} * [\tau_2] - \cdots - \gamma$. Moreover, notice that $id(u_2) \in id(\lambda_1) = id(0(\sigma_{x,x+\xi}))$, $b(u_2) = 0$, $\tau_2 = \lambda_0 \cup \{u_2\}$ and $id(\lambda_0) = id(1(\sigma_{x,x+\xi}))$. Therefore, the step of $\mathcal{Q}$ that changes $id(u_2)$, changes u_2 for a vertex of $\sigma_{x,x+\xi}$ with binary color 0, and all the other steps change vertexes with binary color 0 for vertexes of $\sigma_{x,x+\xi}$ with binary color 1. Since there are m steps in $\mathcal{Q}$ and $v_x * [\tau_2]$ is 0-monochromatic, $\#0(\gamma) = n+1-(m-1) = n+2-m$. Let us rewrite $\mathcal{P}_2$ as $\rho_0 - \rho_1 - \cdots - \rho_m - \rho_{m+1} - \rho_{m+2} - \cdots$. Thus $\rho_m = \gamma$ and $\rho_{m+1} = \sigma_{x+2\xi}$. It was noticed above that $\#0(\rho_{m+1}) = \#0(\rho_{m+1,m+2}) = n+1-m$. By Lemma 6.3.6 and Definition 6.3.7, if it is proved that m is the subdividing point of $\mathcal{P}_2$, then it is progressive, it can only hold case either A or C.

By contradiction, suppose that k is the subdividing point of $\mathcal{P}_2$ and $k = m-c$ for some $c > 0$. By definition of subdividing point, Definition 6.3.2, $\#0(\rho_{k+1,k+2}) \geq n+1-k$ and thus $\#0(\rho_{k+2}) \geq n+1-k$. Notice that the length of the subpath $\rho_{k+3} - \cdots - \rho_m$ is $m-(k+3)+1 = m-k-2 = c-2$. Therefore, $\#0(\rho_m) \geq n+1-k-(c-2) = n+3-(k+c)$. However, it was proved in the previous paragraph that $\#0(\rho_m) = n+2-m = n+2-(k+c)$. A contradiction. Thus m is the subdividing point of $\mathcal{P}_2$. This complete the proof for case B.

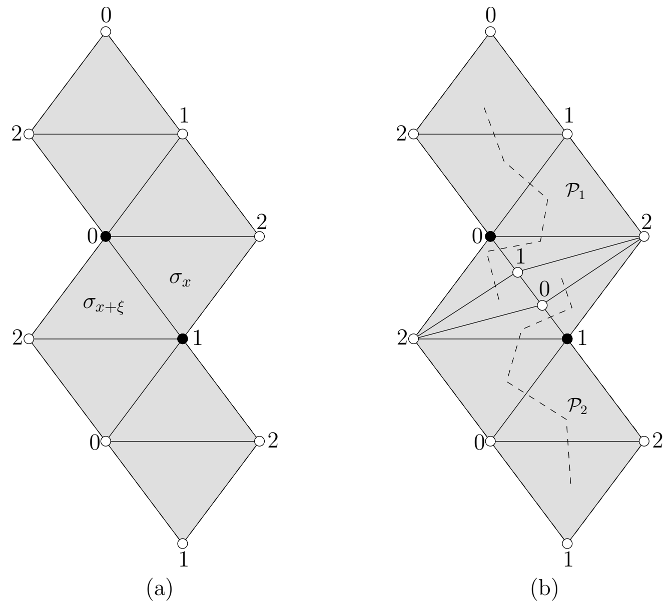

Figure B.3: **SubdivideGood** working on a 2-dimensional path with subdividing point m holding case E, $m = n+1$ and $k_{x-\xi} \neq k_{x+2\xi}$.

Case E with $m = n+1$. By the specification of **ConfigVars**, Definition 6.3.19, $x = m$ and $\xi = +1$. By Lemma 6.3.6, $\#0(\sigma_x) = \#0(\sigma_{x+\xi}) = n+2-m = 1$ and $\#0(\sigma_{x,x+\xi}) = n+1-m = 0$, and hence $b(v_x) = b(v_{x+\xi}) = 0$ and $\#1(\sigma_{x,x+\xi}) = n$. Notice that $\tau = 0(\tau)$ in line 54. Also $(\sigma_{x,x+\xi})_{-id(\tau)}$ is 0-monochromatic by vacuity. By Lemma 6.3.9, $v_x * [\tau]$ and $v_{x+\xi} * [\tau]$ are 0-monochromatic.

Consider first the case $k_{x-\xi} \neq k_{x+2\xi}$. See Figure B.3 for an example of dimension 2. Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be the left and right crossing paths starting on $v_{x+\xi} * [\tau]$ and $v_x * [\tau]$ respectively. That is, $\mathcal{Q}_1 : v_{x+\xi} * [\tau] \xrightarrow{k_x} v_x * [\tau] \xrightarrow{id(\tau_{-k_{x-\xi}})} v_x * [\tau_{+k_{x-\xi}}] \xrightarrow{k_{x-\xi}} \sigma_{x-\xi} - \cdots$ and $\mathcal{Q}_2 : v_x * [\tau] \xrightarrow{k_{x+\xi}} v_{x+\xi} * [\tau] \xrightarrow{id(\tau_{-k_{x+2\xi}})} v_{x+\xi} * [\tau_{+k_{x+2\xi}}] \xrightarrow{k_{x+2\xi}} \sigma_{x+2\xi} - \cdots$. Observe that $\mathcal{Q}_1$ is not in standard form because $v_x * [\tau]$ is 0-monochromatic. If $v_x * [\tau]$ would be non-monochromatic then $\mathcal{Q}_1$ would be in standard form and $|\mathcal{Q}_1| = 2m = 2(n+1)$, by case 2 of Lemma 6.3.12. Moreover, it does not matter $v_x * [\tau]$ is 0-monochromatic, we have that $|\mathcal{Q}_1| = 2(n+1)$. Similarly, $\mathcal{Q}_2$ is not in standard form because $v_{x+\xi} * [\tau]$ is monochromatic but if $v_{x+\xi} * [\tau]$ would be non-monochromatic then $\mathcal{Q}_2$ would be in standard form and $|\mathcal{Q}_2| = |\mathcal{P} - 2$, by case 2 of Lemma 6.3.12. What **SubdivideGood** does is the following. For path $\mathcal{P}_1$ in line 54, the subpath from $v_{x+\xi} * [\tau]$ to $v_{x+\xi} * [\tau_{+k_{x-\xi}}]$ looks like a non-crossing path, then $\mathcal{P}_1$ "crosses" the subdivision from $v_{x+\xi} * [\tau_{+k_{x-\xi}}]$ to $v_x * [\tau_{+k_{x-\xi}}]$ and finally it goes from $v_x * [\tau_{+k_{x-\xi}}]$ to σ_0, as a left crossing path. It is clear that $\mathcal{P}_1$ is in standard form and also $|\mathcal{P}_1| = |\mathcal{Q}_1| = 2(n+1)$. Something similar happens with $\mathcal{P}_2$. Therefore $\mathcal{P}_2$ is in standard form and $|\mathcal{P}_2| = |\mathcal{P}| - 2$. Finally, notice that $\mathcal{P}_1$ and $\mathcal{P}_2$ do not share n-simplexes because $k_{x-\xi} \neq k_{x+2\xi}$.

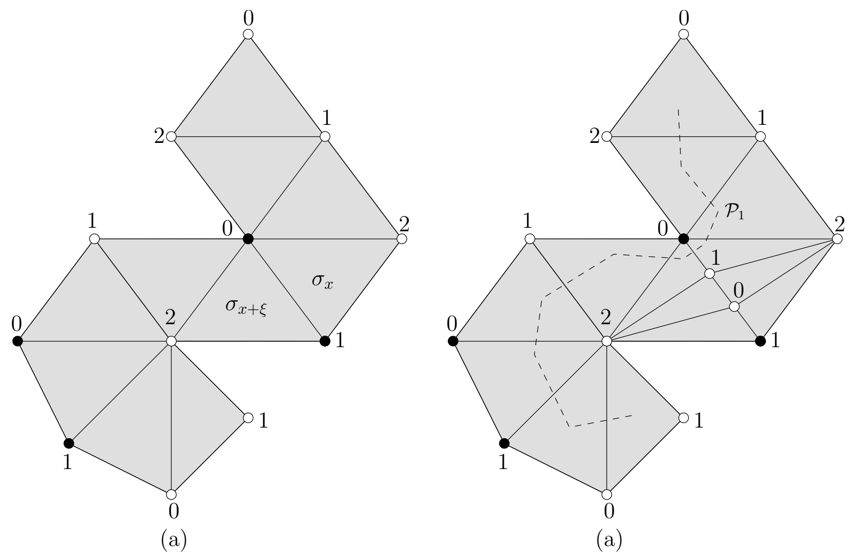

Figure B.4: **SubdivideGood** working on 2-dimensional path with subdividing point m holding case E, $m = n+1$ and $k_{x-\xi} = k_{x+2\xi}$.

Consider now the case $k_{x-\xi} = k_{x+2\xi}$. See Figure B.4 for an example of dimension 2. **SubdivideGood** cannot do the same trick as in the previous case. Since $k_{x-\xi} = k_{x+2\xi}$, it is not possible to get paths that do not share n-simplexes. Therefore **SubdivideGood** outputs a path in standard form $\mathcal{P}_1$ with $|\mathcal{P}_1| = |\mathcal{P}|$ and progressive subdividing point, and an empty path $\mathcal{P}_2$, lines 57 and 58. Consider the n-simplexes $v_x * [\tau_{+k_{x-\xi}}]$ and $v_{x+\xi} * [\tau_{+k_{x-\xi}}]$. By Lemma 6.3.9 and because each face of $(\sigma_{x,x+\xi})$ is 1-monochromatic, $v_x * [\tau_{+k_{x-\xi}}]$ and $v_{x+\xi} * [\tau_{+k_{x-\xi}}]$ are not 0-monochromatic. Therefore $\mathcal{P}_1$ is in standard form and $|\mathcal{P}_1| = |\mathcal{P}|$. Notice that $\tau_{+k_{x-\xi}}$ contains just one vertex of τ, which is binary colored 0. Also, v_x and $v_{x+\xi}$ are binary colored 0. Therefore, $\#0(v_x * [\tau_{+k_{x-\xi}}]) = \#0(v_{x+\xi} * [\tau_{+k_{x-\xi}}]) = 2$ and $\#0([\tau_{+k_{x-\xi}}]) = 1$. Observe that $v_x * [\tau_{+k_{x-\xi}}]$ and $v_{x+\xi} * [\tau_{+k_{x-\xi}}]$ share $[\tau_{+k_{x-\xi}}]$. Let us rewrite $\mathcal{P}_1$ as $\rho_0 - \cdots - \rho_n - \rho_{n+1} - \rho_{n+2} - \rho_{n+3} - \cdots$. That is $\rho_n = \sigma_{x-\xi} = \sigma_n$ and $\rho_{n+1} = v_x * [\tau_{+k_{x-\xi}}]$. Therefore, $\#0(\rho_{n+1}) = 2$ and $\#0(\rho_{n+1,n+2}) = 1$. Let k be the subdividing point of $\mathcal{P}_1$. It is proved that k is progressive and $k = n$. Observe that it cannot be $k < n$ because the subdividing point of $\mathcal{P}$ was $n+1$, and for $0 \leq y \leq n$, $\sigma_y = \rho_y$ and $\sigma_{y,y+1} = \rho_{y,y+1}$. And also for $k = n$, $\#0(\rho_{n+1,n+2}) \geq n+1-k = 1$. Thus, n is the subdividing point of $\mathcal{P}_1$. Moreover, $\#0(\rho_{k+1}) = \#0(\rho_{n+1}) = n+2-k = 2$ and thus k holds case either D or E with $k < n+1$, by Lemma 6.3.6. By Definition 6.3.7, k is progressive. This complete the proof of case E.

■

Bibliography

[1] J. Anderson & M. Moir, *Using Local-Spin k-Exclusion Algorithms to Improve Wait-Free Object Implementations*, Distributed Computing 11(1), pp. 1–20, 1997.

[2] Y. Afek & M. Merritt, *Fast, Wait-Free* $(2k-1)$*-renaming*, in Proc. of the 18th ACM Symposium on Principles of Distributed Computing, pp. 105–112, 1999.

[3] Y. Afek, H. Attiya, A. Fouren, G. Stupp & D. Touitou, *Long-Lived Renaming Made Adaptive*, in Proc. of the 18th ACM Symposium on Principles of Distributed Computing, pp. 91–103, 1999.

[4] Y. Afek, G. Stupp & D. Touitou, *Long Lived Adaptive Splitter and Applications*, Distributed Computing 15(2), pp. 67–86, 2002.

[5] Y. Afek & I. Nir, *Failure Detectors in Loosely Named Systems*, in Proc. of the 27th ACM Symposium on Principles of Distributed Computing, pp. 67–74, 2008.

[6] Y. Afek, I. Gamzu, I. Levy, M. Merritt & G. Taubenfeld, *Group Renaming*, in Proc. of the 12th International Conference on Principles of Distributed Systems, pp. 58–72, 2008.

[7] H. Attiya, A. Bar-Noy, D. Dolev, D. Koller, D. Peleg & R. Reischuck, *Achievable Cases in an Asynchronous Environment*, in Proc. of the 28th IEEE Symposium on Foundations of Computer Science, pp. 337–346, 1987.

[8] H. Attiya, A. Bar-Noy, D. Dolev, D. Peleg & R. Reischuck, *Renaming in Asynchronous Environment*, Journal of the ACM 37(3), pp. 524–548, 1990.

[9] H. Attiya & J. Welch, *Distributed Computing: Fundamentals, Simulations and Advanced Topics*, McGraw-Hill 1998.

[10] H. Attiya & O. Rachman, *Atomic Snapshots in* $o(n \log n)$ *Operations*, SIAM Journal on Computing 27 (2), pp. 319–340, 1998.

[11] H. Attiya & A. Fouren, *Polynomial and Adaptive Long-Lived (2k-1)-renaming*, in Proc. of the 14th International Symposium on Distributed Computing, pp. 149–163, 2000.

[12] H. Attiya & A. Fouren, *Adaptive and Efficient Algorithms for Lattice Agreement and Renaming*, SIAM Journal on Computing 31(2), pp. 642–664, 2001.

[13] H. Attiya & T. Djerassi-Shintel, *Time Bounds for Decision Problems in the Presence of Timing Uncertainty and Failures*, Journal on Parallel Distributed Computing 61(8), pp. 1096–1109, 2001.

[14] H. Attiya & S. Rajsbaum, *The Combinatorial Structure of Wait-Free Solvable Tasks*, SIAM Journal on Computing 31(4), pp. 1286–1313, 2002.

[15] H. Attiya, A. Fouren & E. Gafni, *An Adaptive Collect Algorithm with Applications*, Distributed Computing 15(2), pp. 87–96, 2002.

[16] A. Bar-Noy & D. Dolev, *A Partial Equivalence Between Shared-Memory and Message-Passing in an Asynchronous Fail-Stop Distributed Environment*, Mathematical Systems Theory 26(1), pp. 21–39, 1993.

[17] O. Biran, S. Moran & S. Zaks, *A Combinatorial Characterization of the Distributed Tasks Which Are Solvable in the Presence of One Faulty Processor*, in Proc. of the 7th. ACM Symposium on Principles of Distributed Computing, pp. 263–275, 1988.

[18] O. Biran, S. Moran & S. Zaks, *A Combinatorial Characterization of the Distributed 1-Solvable Tasks,* J. Algorithms 11(3), pp. 420–440, 1990.

[19] E. Borowsky & E. Gafni, *Generalized FLP Impossibility Result for t-Resilient Asynchronous Computations*, in Proc. of the 25th. ACM Symposium on Theory of Computing, pp. 91–100, 1993.

[20] E. Borowsky & E. Gafni, *Immediate Atomic Snapshots and Fast Renaming*, in Proc. of the 12th. ACM Symposium on Principles of Distributed Computing, pp. 41–51, 1993.

[21] E. Borowsky & E. Gafni, *A Simple Algorithmically Reasoned Characterization of Wait-Free Computations*, in Proc. of the 16th. ACM Symposium on Principles of Distributed Computing, pp. 189–198, 1997.

[22] E. Borowsky, *Capturing the Power of Resiliency and Set Consensus in Distributed Systems*, PhD Thesis, University of California, Los Angeles, 1995.

[23] E. Borowsky, E. Gafni, N. Lynch & S. Rajsbaum, *The BG Distributed Simulation Algorithm*, Distributed Computing 14(3), pp. 127–146, 2001.

[24] A. Brodsky, F. Ellen & P. Woelfel, *Fully-Adaptive Algorithms for Long-Lived Renaming*, in Proc. of the 20th International Symposium on Distributed Computing, pp. 413–427, 2006.

[25] H. Buhrman, J. A. Garay, J. Hoepman & M. Moir, *Long-Lived Renaming Made Fast*, in Proc. of the 14th. ACM Symposium on Principles of Distributed Computing, pp. 194–203, 1995.

[26] J. Burns & G. Peterson, *The Ambiguity of Choosing*, in Proc. of the 8th ACM Symposium on Principles of Distributed Computing, pp. 145–158, 1989.

[27] A. Castañeda & S. Rajsbaum, *New Combinatorial Topology Upper and Lower Bounds for Renaming*, in Proc. of the 27th. ACM Symposium on Principles of Distributed Computing, pp. 295–304, 2008.

[28] A. Castañeda & S. Rajsbaum, *New Combinatorial Topology Upper and Lower Bounds for Renaming: The Lower Bound*. Distributed Computing 22(5), pp. 287–301, 2010.

[29] A. Castañeda & S. Rajsbaum. *New Combinatorial Topology Upper and Lower Bounds for Renaming: The Upper Bound*. Publicación Preliminar No. 872 del Instituto de Matemáticas, Universidad Nacional Autónoma de México, Dec. 2009. `http://texedores.matem.unam.mx/publicaciones/index.php?option=com_remository&Itemid=57&func=fileinfo&id=163`. Submitted for publication. Preliminary version is in [27].

[30] T. Chandra & S. Toueg. *Unreliable Failure Detectors for Reliable Distributed Systems*. Journal of the ACM 43(2): 225–267 (1996).

[31] S. Chaudhuri, *More Choices Allow More Faults: Set Consensus Problems in Totally Asynchronous Systems*. Information and Computation 105(1): 132–158 (1993).

[32] S. Chaudhuri, M. Herlihy & M. R. Tuttle, *Wait-Free Implementations in Message-Passing Systems*, Theoretical Computer Science 220(1), 211–245, 1999.

[33] T. tom Dieck. *Transformation Groups*. Gruiter Studies in Mathematics (1987).

[34] T. Djerassi-Shintel, *Lower Bounds for Decision Problems in Semi-synchronous Systems*, M.Sc.thesis, Department of Computer Science, The Technion, Oct. 1993. [In Hebrew]

[35] D. Dolev, C. Dwork & L. Stockmeyer, *On the Minimal Synchronism Needed for Distributed Consensus*, Journal of the ACM 34(1), pp. 77–97, 1987.

[36] K. Fan. *Simplicial Maps from an Orientable n-Pseudomanifold into S^m with the Octahedral Triangulation*. Journal of Combinatorial Theory 2: 588–602 (1967).

[37] A Fouren, *Exponential examples of Two Renaming Algorithms*, available at `http://cs.technion.ac.il/~hagit/publications/expo.ps.gz`, 1999.

[38] G. N. Fredrickson & N. A. Lynch, *Electing a leader in a synchronous ring*, Journal of the ACM 34 (1) (1987), 89–115.

[39] M. Fischer, N. Lynch & M. Paterson, *Impossibility of Distributed Consensus with One Faulty Process*, Journal of the ACM 32(2), pp. 374–382, 1985.

[40] E. Gafni, *Group-Solvability*, in Proc. of the 18th International Symposium on Distributed Computing, pp.30–40, 2004.

[41] E. Gafni, *DISC/GODEL presentation: R/W Reductions* (DISC04), 2004. `http://www.cs.ucla.edu/eli/eli/godel.ppt`

[42] E. Gafni, S. Rajsbaum & M. Herlihy, *Subconsensus Tasks: Renaming is Weaker Than Set Agreement*, in Proc. of the 20th International Symposium on Distributed Computing, pp.329–338, 2006.

[43] E. Gafni, M. Raynal & C. Travers, *Test & Set, Adaptive Renaming and Set Agreement: a Guided Visit to Asynchronous Computability*, in Proc. of the 26th IEEE Symposium on Reliable Distributed Systems, pp. 93–102, 2007.

[44] E. Gafni. *The Extended BG-Simulation and the Characterization of t-Resiliency*. Proceedings of the 41st. Annual ACM Symposium on Theory of Computing, 85–92 (2009).

[45] E. Gafni. *The* $0-1$ *exclusion families of tasks.* Proceedings of the 12th International Conference on Principles of Distributed Systems, 246–258 (2008).

[46] E. Gafni, A. Mostéfaoui, M. Raynal & C. Travers, *From Adaptive Renaming to Set Agreement*, Theoretical Computer Science 410(14): 1328–1335 (2009).

[47] E. Gafni, *Renaming with k-Set-Consensus: An Optimal Algorithm into* $n + k - 1$ *Slots.* Proceedings of the 10th International Conference on Principles of Distributed Systems, 36–44 (2006).

[48] E. Gafni & S. Rajsbaum, *Recursion in Distributed Computing*, Proceedings of the 12th International Symposium on Stabilization, Safety, and Security of Distributed Systems, 362-376 (2010).

[49] R. L. Graham, B. L. Rothschild & J. H. Spencer. *Ramsey Theory.* Wiley 1980.

[50] M. Henle. *A Combinatorial Introduction to Topology.* Dover 1994.

[51] M. Herlihy & J. Wing, *Linearizability: a correctness condition for concurrent objects.* ACM Transactions on Programming Languages and Systems 12(3), pp 463–492, 1990.

[52] M. Herlihy, *Wait-free Synchronization*, ACM Transactions On Programming Languages and Systems 13(1), pp. 123–149, 1991.

[53] M. Herlihy & N. Shavit, *The asynchronous computability theorem for t-resilient tasks*, in Proc. of the 25th ACM Symposium on the Theory of Computing, pp. 111–120, 1993.

[54] M. Herlihy, *A Tutorial on Algebraic Topology and Distributed Computing*, notes for a tutorial presented in UCLA, August 1995.

[55] M. Herlihy & N. Shavit, *The Topological Structure of Asynchronous Computability*, Journal of the ACM 46(6), pp. 858–923, 1999.

[56] M. Herlihy & S. Rajsbaum, *Algebraic Spans*, Mathematical Structures in Computer Science 10(4), pp. 549–573, 2000.

[57] M. Herlihy, V. Luchangco & M. Moir, *Space and Time Adaptive Non-blocking Algorithms*, Electronic Notes in Theoretical Computer Science 78, 2003.

[58] M. Inoue, S. Umetani, T. Masuzawa & H. Fujiwara. *Adaptive Long-Lived* $O(k^2)$*-Renaming with* $O(k^2)$ *Steps.* in Proc. of the 15th International Symposium on Distributed Computing, pp. 123–135, 2001.

[59] N. Lynch. *Distributed Algorithms.* Morgan Kaufman Pub. 1996.

[60] M. Moir & J. Anderson, *Wait-Free Algorithms for Fast, Long-Lived Renaming*, Science of Computer Programming 25(1), pp. 1–39, 1995.

[61] M. Moir, *Fast, Long-Lived Renaming Improved and Simplified*, Science of Computer Programming 30(3), pp. 287–308, 1998.

[62] A. Mostéfaoui, M. Raynal & C. Travers, *Exploring Gafni's Reduction Land: From Ω^k to Wait-Free Adaptive $(2p - \lceil p/k \rceil)$-Renaming Via k-Set Agreement*, in Proc. of the 20th International Symposium on Distributed Computing, pp. 1–15, 2006.

[63] A. Mostéfaoui, M. Raynal & C. Travers. *From Renaming to Set Agreement.* in Proc. of the 14th International Colloquium on Structural Information and Communication Complexity, pp. 66–80, 2007.

[64] J. R. Munkres. *Elements of Algebraic Topology.* Addison-Wesley 1993.

[65] G. Neiger. *Failure Detectors and the Wait-Free Hierarchy* in Proc. of the 14th ACM Symposium on Principles of Distributed Computing, pp. 100–109, 1995.

[66] M. Okun, A. Barak & E. Gafni, *Renaming in Synchronous Message Passing Systems with Byzantine Failures*, Distributed Computing 20(6), pp. 203–413, 2008.

[67] M. Raynal, *Locks Considered Harmful: A Look at Non-traditional Synchronization*, in Proc. of the 6th International Workshop on Software Technologies for Embedded and Ubiquitous Systems, pp. 369–380, 2008.

[68] D. Redmond. *Number Theory. An Introduction.* Mercel Dekker 1996.

[69] M. Raynal & C. Travers, *In Search of the Holy Grail: Looking for the Weakest Failure Detector for Wait-Free Set Agreement*, in Proc. of the 10th International Conference on Principles of Distributed Systems, pp. 3–19, 2006.

[70] M. Saks & F. Zaharoglou, *Wait-Free k-Set Agreement Is Impossible: The Topology of Public Knowledge*, SIAM Journal on Computing 29(5), pp. 1449–1483, 2000.