UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN CIENCIAS MATEMÁTICAS Y DE LA ESPECIALIZACIÓN EN ESTADÍSTICA APLICADA

UN ESTUDIO DE ALGORITMOS DISTRIBUIDOS DESDE UNA PERSPECTIVA DE LA TOPOLOGÍA ALGEBRAICA

TESIS

QUE PARA OPTAR POR EL GRADO DE DOCTOR EN CIENCIAS

PRESENTA:

M. EN C. FERNANDO ANDRES BENAVIDES AGREDO

TUTOR PRINCIPAL:

Dr. SERGIO RAJSBAUM GORODEZKY (IMUNAM)

MIEMBROS DEL COMITÉ TUTOR

Dr. JAVIER BRACHO CARPIZO (IMUNAM)
Dr. JOSÉ DAVID FLORES (FCIEN)

CIUDAD DE MÉXICO, MAYO DE 2017

Abstract

The celebrated *asynchronous computability theorem* provides a characterization of the class of decision tasks that can be solved in a wait-free manner by asynchronous processes that communicate by writing and taking atomic snapshots of a shared memory. Several variations of the model have been proposed (immediate snapshots and iterated immediate snapshots), all equivalent for wait-free solution of decision tasks, in spite of the fact that the protocol complexes that arise from the different models are structurally distinct. The topological and combinatorial properties of these snapshot protocol complexes have been studied in detail, providing explanations for why the asynchronous computability theorem holds in all the models.

In reality concurrent systems do not provide processes with snapshot operations. Instead, snapshots are implemented (by a wait-free protocol) using operations that write and read individual shared memory locations. Thus, read/write protocols are also computationally equivalent to snapshot protocols. However, the structure of the read/write protocol complex has not been studied. We show that the read/write iterated protocol complex is collapsible (and hence contractible), using discrete Morse theory. Furthermore, we show that a distributed protocol that wait-free implements atomic snapshots in effect is performing the collapses. In addition, using discrete Morse theory, we give a new combinatorial proof of collapsibility of iterated immediate snapshot protocol complex. In fact, we prove that chromatic subdivision is preserved by collapsibility.

Resumen

El Teorema de Computabilidad Asíncronona (TCA) carateriza la clase de las tareas de decisión que son wait-free solubles por procesos asíncronos que se comunican mediante escrituras y snapshots en una memoria compartida. Variaciones de este modelo han sido propuestas (immediate snapshots e iterated immediate snapshots), todos ellos computacionalmente equivalentes a pesar que la estructura de los complejos de protocolo asociados a ellos es diferente. Las propiedades topológicas y combinatorias de los complejos de protocol asociados a modelos snapshots se han estudiado en detalle, suministrando explicaciones del porque el TCA se verifica en todos los modelos.

En realidad los sistemas de cómputo concurrente no suministran procesos con operaciones de snapshot. En su lugar, las operaciones snapshot son implementadas (de una manera wait-free) usando operaciones de lectura y escritura en registros de memoria compartida. De esta menera, los protocolos de lectura y escritura son computacionalmente equivalentes a los protocolos snapshot. Sin embargo, la estructura del complejo de protocolo de lectura y escritura no ha sido estudiada. Nosotros probamos que el complejo de protocolo de lectura y escritura iterado es colapsable (y asi contraible), usando teoría de Morse discreta. Además, mostramos que un protocolo distribuido que implementa de una manera wait-free las operaciones atomicas de snapshot ejecuta los colapsos. Adicionalmente, usando teoría de Morse discreta, damos una nueva prueba combinatoria de la colapsabilidad del complejo de protocolo iterado immediate snapshot. De hecho, probamos que la subdivisión cromática se preserva bajo colapsabilidad.

Agradecimientos

Al acercarse el final de esta etapa de mi vida, la cual me ha traido muchas alegrias y satisfacciones, es indiscutible pensar en cada uno de los que me apoyaron y fortalecieron en este camino. Por lo cual expreso mis sinceros agradecimientos:

A Dios padre por haberme acompañado, guiado y brindarme la oportunidad de disfrutar esta experiencia llena de aprendizajes y felicidad.

A mi tutor el Dr. Sergio Rajsbaum Gorodezky por su invaluable apoyo, confianza y principalmente por su amistad. A mi comite tutoral, el Dr. David Flores Peñalosa y el Dr. Javier Bracho Carpizo. Así como al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo recibido durante cuatro años mediante la beca de doctorado y a los proyecto UNAM-PAPIIT IN109917 y ECOS-NORD-Conacyt M12M01.

A cada uno de mis amigos del Instituto de Matemáticas. Pero en especial a mis amigos colombianos Diego, Yefferson, Lore y Diana.

Gracias a mis padres y hermanos por el incondicional apoyo que me han brindado desde el momento que inicie mis estudios en matemáticas.

Y finalmente los más importantes, mi esposa Maria Rosa e hijo Santiago quienes me han dado fortaleza en cada uno de los momentos difíciles y me han brindado todo su amor y respaldo a lo largo de nuestra vida juntos.

Fernando Andres Benavides

Introduction

A *decision task* is the distributed equivalent of a function, where each process knows only part of the input, and after communicating with the other processes, each process computes part of the output. For instance, in the *k-set agreement* task (Chaudhuri, 1993) processes have to agree on at most k of their input values; when $k = 1$ we get the *consensus* task (Lamport et al., 1982).

A central concern in distributed computability is studying which tasks are solvable in a given distributed computing model, as determined by the type of communication mechanism available and the reliability of the processes. Early on it was shown that consensus is not solvable even if only one process can fail by crashing, when the processes are asynchronous and they communicate by message passing (Fischer et al., 1985), or even by writing and reading a shared memory (Loui and Abu-Amara, 1987). A graph theoretic characterization of the tasks solvable in the presence of at most one process failure appeared soon after (Biran et al., 1990).

The *asynchronous computability theorem* (Herlihy and Shavit, 1999) exposed that moving from tolerating one process failure, to any number of process failures, yields a characterization of the class of decision tasks that can be solved in a wait-free manner by asynchronous processes based on simplicial complexes, which are higher dimensional versions of graphs. In particular, n-set agreement is not wait-free solvable, with $n+1$ processes (Borowsky and Gafni, 1993a; Herlihy and Shavit, 1999; Saks and Zaharoglou, 2000).

Computability theory through combinatorial topology has evolved to encompass non-independent process failures, arbitrary malicious failures, synchronous and partially synchronous processes, and various communication mechanisms (Herlihy et al., 2013a). Still, the original wait-free model of the asynchronous computability theorem, where crash-prone processes that communicate wait-free by writing and reading a shared memory is fundamental. Topology techniques are derived in this model, and then extended to other models, e.g. Herlihy and Rajsbaum (2013). Also, the question of solvability in other models (e.g. t crash failures, for $1 \leq t \leq n$), can in many cases be reduced to the question of wait-free solvability ($t = n$), as shown by Borowsky et al. (2001) and Herlihy and Rajsbaum (2012).

More specifically, in the *AS model* of Herlihy et al. (2013a) each process can write its own location of the shared-memory, and it is able to read the entire shared memory in one atomic step, called a *snapshot*. The characterization is based on the *protocol complex,* which is a geometric representation of the various possible executions of a protocol. Simpler variations of this model have been considered. In the *immediate snapshot* (IS) version (Attiya and Rajsbaum, 2002; Borowsky and Gafni, 1993a; Saks and Zaharoglou, 2000), processes can execute a combined write-snapshot operation. The *iterated immediate snapshot* (IIS) model (Borowsky and Gafni, 1997) is even simpler to analyze, and can be extended (IRIS) to analyze partially synchronous models (Rajsbaum et al., 2008). Processes communicate by accessing a sequence of shared arrays, through immediate snapshot operations, one such

operation in each array. The success of the entire approach hinges on the fact that the topology of the protocol complex of a model determines critical information about the solvability of the task and, if solvable, about the complexity of solution (Hoest and Shavit, 1997).

All these snapshot models, AS, IS, IIS and IRIS can solve exactly the same set of tasks. However, the protocol complexes that arise from the different models are structurally distinct. The combinatorial topology properties of these complexes have been studied in detail, providing insights for why some tasks are solvable and others are not in a given model.

Results In reality concurrent systems do not provide processes with snapshot operations. Instead, snapshots are implemented (by a wait-free protocol) using operations that write and read individual shared memory locations (Afek et al., 1993). Thus, read/write protocols are also computationally equivalent to snapshot protocols. However, the structure of the read/write protocol complex has not been studied.

We show that discrete Morse theory (Forman, 1995) can be used in a natural way to study distributed computing protocol complexes, and derive the following results.

1. The one-round read/write protocol complex is collapsible to the IS protocol, i.e. to a chromatic subdivision of the input complex. The collapses can be performed simultaneously in entire orbits of the natural symmetric group action.

2. Furthermore, the distributed protocol that wait-free implements immediate snapshots of Borowsky and Gafni (1993b) in effect is performing the collapses.

3. Also the multi-round iterated read/write complex is collapsible.

4. A new combinatorial proof of collapsibility of iterated immediate snapshot protocol complex. In fact, we prove that chromatic subdivision is preserved by collapsibility.

Related work The asynchronous computability theorem (Herlihy and Shavit, 1999) characterizes the tasks that are solvable wait-free by $n+1$ processes, communicating by writing and taking snapshots of a shared memory. At the core of the necessity part of the theorem, is shown that the protocol complex in n-connected (Lemma 4.12), something that is implied by our collapsibility result, for the more complicated read/write protocol complex. The necessity part of the asynchronous computability theorem is what is used to prove impossibility results.

The one-round immediate snapshot protocol complex is the simplest, with an elegant combinatorial representation; it is a chromatic subdivision of the input complex (Herlihy et al., 2013a; Kozlov, 2012), and so is the (multi-round) IIS protocol (Borowsky and Gafni, 1997). The multi-round (single shared memory array) IS protocol complex is harder to analyze, combinatorially it can be shown to be a pseudomanifold (Attiya and Rajsbaum, 2002). IS and IIS protocols are homeomorphic to the input complex. An AS protocol complex is not generally homeomorphic to the underlying input complex, but it is homotopy equivalent to it (Havlicek, 2004). The span of Herlihy and Shavit (1999) provides an homotopy equivalence of the (multi-round) AS protocol complex to the input complex (Havlicek, 2004), clarifying the basis of the obstruction method (Havlicek, 2000) for detecting impossibility of solution of tasks.

Later on stronger results were proved, about the collapsibility of the protocol complex. The one-round IS protocol complex is collapsible (Kozlov, 2014) and homeomorphic to

closed balls. The structure of the AS is more complicated, it was known to be contractible (Havlicek, 2004; Herlihy et al., 2013a), and then shown to be collapsible (one-round) to the IS complex (Kozlov, 2015). We use ideas from (Kozlov, 2015), together with distributed computing techniques of partial orders in our development. The IIS (multi-round) version was shown to be collapsible too (Goubault et al., 2015), where a category of colored simplicial complexes is developed.

There are several wait-free implementations of atomic snapshots starting with (Afek et al., 1993), but we are aware of only two algorithms that implement immediate snapshots; the original of Borowsky and Gafni (1993b), and its recursive version (Gafni and Rajsbaum, 2010).

Discrete Morse theory is a combinatorial version of the highly developed Morse theory used to analyze the topology of a manifold by studying differentiable functions on that manifold. The theory, developed by Forman (1995, 1998), has applications in diverse fields of mathematics and computer science. See also (Kozlov, 2008). We were inspired by the study of barycentric subidivsions Zhukova (2016). There have been applications to computer science, see e.g. (Biserka, 2015) and references herein, but to the best of our knowledge, our work is the first to use discrete Morse in distributed computing.

Outline of the thesis This thesis is organized in seven chapters. In the Chapter 1 we give a short introduction to computability in distributed computing and we describe a basic read/write model as well as some snapshot sub-models. In the Chapter 2 we describe two simple distributed algorithms that run on the basic read/write memory model, and produce process views simulating immediate snapshot operations. These algorithms will be important in the analyzing of simplicial complex associated to read/write model. The original algorithm was designed by Borowsky and Gafni (1993b) however we use its recursive version due to Gafni and Rajsbaum (2010) because it is built on top of iterated read/write model. In the iterated model the executions of recursive algorithm can be seen as sequences of executions of one-round read/write model. Chapter 3 is devoted to introduce distributed computing through combinatorial topology, however first we give some basic concepts from topology and explain the important collapsibility procedure. Also in this chapter we describe the simplicial complexes associated to all possible executions of the recursive implementation of Gafni and Rajsbaum (2010). In order to make the thesis self-contained and to understand some proofs of the our main contributions we include in the Chapter 4 the discrete Morse theory notions and results needed to analyze the topology of the protocol complexes of read/write models. Discrete Morse theory is applied to analyze the connectivity of the read/write protocol and immediate snapshot protocol in Chapters 5 and 6 respectively. In each chapter we study the protocol complexes of one-round and multiple rounds. Finally our conclusions can be found in the Chapter 7.

Introducción

Una *tarea de decisión* es el equivalente distribuido de una función, donde cada proceso conoce únicamente parte de la entrada, y después de comunicarse con los demás procesos computa parte de la salida. Por ejemplo, en la tarea k-*set agreement* (Chaudhuri, 1993) los valores de salida de los procesos deben coincider en a lo más k de sus valores de entrada, cuando $k = 1$ se tiene la tarea del *consenso* (Lamport et al., 1982).

Uno de los problemas centrales en el cómputo distribuido es determinar cuales tareas son solubles en un modelo de cómputo distribuido dado, el cual está definido principalmente por el mecanismo de comunicación y la confialibilidad de los procesos. En este sentido se conoce que consenso no es soluble si solamente un proceso falla por crashing, los procesos son asíncronos y ellos se comunican por paso de mensages (Fischer et al., 1985) o por medio de la escritura y lectura de una memoria compartida (Loui and Abu-Amara, 1987). Tiempo después Biran et al. (1990) dieron una caraterización mediante gráficas de aquellas tareas solubles en las cuales a lo más un proceso falla.

El *Teorema de Computabilidad Asncrona* (TCA) (Herlihy and Shavit, 1999) caracteriza aquellas tareas de decisión wait-free solubles por procesos asíncronos y las cuales toleran cualquier número de fallas. La caraterización se realiza mediantes complejos simpliciales los cuales son la version generalizada de las gráficas. En particular, la tarea n-set agreement no es wait-free soluble, con $n+1$ procesos (Borowsky and Gafni, 1993a; Herlihy and Shavit, 1999; Saks and Zaharoglou, 2000).

La teoría de la computabilidad a través de la topología combinatoria ha evoluciolnado hasta comprender fallas de procesos no-independientes, fallas maliciosas , procesos asíncronos y parcialmente asíncronos y varios mecanismos de comunicación (Herlihy et al., 2013a). Hasta hoy en dia el Teorema de Computabilidad Asíncrona para tareas wait-free en las cuales los procesos fallan por crashing y su comunicación es mediante lecturas y escrituras de una memoria compartida es fundamental. Diferentes técnicas topológicas se han derivado de este modelo y extendido a otros modelos de cómputo, p.ej. Herlihy and Rajsbaum (2013). En este sentido la solubilidad de tareas en otros modelos (p.ej. modelos donde t procesos fallan por crashing, para $1 \leq t \leq n$) puede ser en muchos casos reducido al caso wait-free ($t = n$, respectivey) como lo probaron Borowsky et al. (2001) y Herlihy and Rajsbaum (2012).

En el *modelo AS* de Herlihy et al. (2013a) cada proceso puede escribir en su propio registro de una memoria compartida y puede leer la memoria compartida en una sola operación atómica denominada *snapshot*. La caracterización se fundamenta en el complejo simplicial asociado al protocolo de cómputo denominado el *complejo del protocolo*, el cual es una representación geométrica de todas las posibles ejecuciones del protocolo. Algunas simples variaciones de este modelo han sido también consideradas. Por ejemplo en la versión *Immediate Snapshot* (IS) (Attiya and Rajsbaum, 2002; Borowsky and Gafni, 1993a; Saks and Zaharoglou, 2000), los procesos pueden ejecutar la operación combinada de lectura y

escritura. En el modelo *Iterated Immediate Snapshot* (IIS) (Borowsky and Gafni, 1997) es aún más simple de analizar y puede ser extendido (IRIS) con el fin de estudiar modelos parcialmente asíncronos (Rajsbaum et al., 2008). En el modelo IIS los procesos se comunican accesando una secuencia de memorias compartidas a través de las operaciones de immediate snapshot, una operación por cada memoria. El exito de este enfoque se fundamenta en que la topología del complejo del protocolo de un modelo contiene información fundamental acerca de la solución de una tarea y de la complejidad de la solución (Hoest and Shavit, 1997).

Todos los modelos AS, IS, IIS y IRIS pueden solucionar el mismo conjunto de tareas. Sin embargo los complejos simpliciales asociados a cada uno de estos modelos son estructuralmente diferentes. Las propiedades topológicas combinatorias de estos complejos han sido estudiadas en detalle suministrando información del porque algunas tareas son solubles y otras no en un modelo de cómputo dado.

Resultados En realidad los sistemas concurrentes no suministran procesos con operaciones de snapshot. En su lugar, estas operaciones son implementadas (por wait-free protocolos) usando operaciones de lectura y escritura en memorias compartidas (Afek et al., 1993). De ahí que los protocolos de lectura y escritura son computacionalmente equivalentes a los protocolos snapshot. Sin embargo, la estructura de los protocolos de lectura y escritura no han sido estudiados. En esta tesis mostramos que la Teoría Discreta de Morse (Forman, 1995) puede ser usada de una manera natural para estudiar los complejos de protocolos de cómputo distribuido y derivar los siguientes resultados:

1. El complejo del protocolo de lectura y escritura de una ronda es colapsable al complejo del protocolo IS, es decir es colapsable a la subdivisión cromática del complejo de entrada. Los colapsos se realizan por orbitas de la acción del grupo simétrico.

2. Además, el protocolo distribuido de Borowsky and Gafni (1993b) que implementa de una manera wait-free las operaciones de immediate snapshot realiza los colapsos.

3. De igual manera el complejo del protocolo de lectura y escritura de multiples rondas es colapsable.

4. Finalmente, se describe una nueva prueba puramente combinatoria de la colapsabilidad de la subdivisioón cromática de un complejo simplicial Δ siempre y cuando Δ sea colapsable. En otras palabras, probamos que la subdivision cromtica se preserva bajo el procedimiento de colapsabilidad.

Trabajos relacionados El teorema de computabilidad asíncrona (Herlihy and Shavit, 1999) caracteriza las tareas que son wait-free solubles por $n+1$ procesos, los cuales se comunican escribiendo y tomando snapshots de una memoria compartida. En su demostración se prueba que el complejo del protocolo es n-conexo, lo cual en nuestro caso particular es implicado por la colapsabilidad del complejo de lectura y escritura. Esta condición es necesaria para probar resultados de imposibilidad.

El complejo del protocolo immediate snapshot es el más simple de analizar dado su elegante representación combinatoria. Por otro lado el complejo es una subdivisón del complejo de entrada (Herlihy et al., 2013a; Kozlov, 2012), lo cual también se cumple con el complejo del protocolo IIS (Borowsky and Gafni, 1997). En el caso del complejo del protocolo no-iterado immediate snapshot se ha probado que es una pseudomanifold (Attiya

and Rajsbaum, 2002) a pesar de la dificultad de su análisis. Por otro lado los protocolo IS e IIS son heomorfos al complejo de entrada. En general el complejo del protocol AS no es homeomorfo al complejo de entrada pero si son homotópicamente equivalentes (Havlicek, 2004). El mapeo span descrito por Herlihy and Shavit (1999) suministra una equivalencia homotópica entre el complejo del protocolo AS y el complejo de entrada (Havlicek, 2004), con lo cual se da mayor claridad a la base del método de obstrucción (Havlicek, 2000) para detectar la imposibilidad de tareas de decisión.

Tiempo después resultados mas fuertes acerca de la colapsabilidad de los complejos fueron probados. El complejo del protocolo IS de una ronda es colapsable (Kozlov, 2014) y homeomórfico a bolas cerradas. Inicialmente Havlicek (2004) y Herlihy et al. (2013a) probaron que complejo AS es contraible y posteriormente Kozlov (2015) mostró que este es colapsable al complejo IS. Goubault et al. (2015) muestran que el complejo del protocol IIS es colapsable, en su prueba ellos definen la categoria de complejos simpliciales cromáticos.

Existen varias implementaciones de las operaciones snapshot, por ejemplo la implementación de Afek et al. (1993). Pero hasta donde conocemos solo existen dos implementaciones de las operaciones immediate snapshot; la original de Borowsky and Gafni (1993b), y su versión recursiva de Gafni and Rajsbaum (2010).

La teoría de Morse discreta es una versión combinatoria de la teoría de Morse continua usada para el estudio de funciones diferenciables sobre manifolds. La teoría, desarrollada por Forman (1995, 1998), tiene diversas aplicaciones en distintos campos de la matemática y ciencias de la computación, p. eje. (Biserka, 2015). Sin embargo hasta donde sabemos nuestro trabajo es el primero en usar teoría de Morse disctreta en el cónputo distribuido.

Contenido de la tesis La presente tesis está organizada en siete capitulos. En el Capitulo 1 damos una corta introducción a computabilidad en sistemas distribuidos y describimos un modelo básico de lectura y escritura así como también algunos sub-modelos snapshot. En el Capitulo 2 describimos dos algoritmos los cuales están diseados en el modelo básico de lectura y escritura, y las vistas de los procesos que producen simulan operaciones immediate snapshot. Estos algoritmos serán importantes en el análisis del complejo simplicial asociado al modelo de lectura y escritura. El algoritmo original fue diseado por Borowsky and Gafni (1993b) sin embargo nosotros usamos su versión recursiva de Gafni and Rajsbaum (2010) ya que está construido sobre el modelo iterado de lectura y escritura. En el modelo iterado las ejecuciones pueden ser vistas como sucesiones de ejecuciones del modelo de lectura y escritura de una ronda. El Capitulo 3 es dedicado a introducir el cómputo distribuido a través de la topología combinatoria, sin embargo primero damos algunos conceptos básicos de la topología y explicamos el importante procedimiento de colapsabilidad. En este capitulo también describimos los complejos simpliciales asociados a las ejecuciones de la versión recursiva de Gafni and Rajsbaum (2010). Con el fin que la tesis sea auto-contenida y entender algunas demostraciones de nuestras principales contribuciones en el Capitulo 4 incluimos conceptos y resultados propios de la teoría de Morse discreta necesarios para analizar la topología de los modelos de lectura y escritura. La teoría de Morse discreta es aplicada para analizar la conectividad de los protocolos de lectura y escritura e immediate snapshot en los Capitulos 5 y 6 respectivamente. En cada capitulo estudiamos los complejos de protocolo de una ronda y multiples rondas. Finalmente nuestras conclusiones pueden ser encontradas en el Capitulo 7.

Contents

Chapter 1

Read/Write Distributed Models

The aim of this chapter is not to try to represent faithfully how a distributed model works. Instead, we want to give a clean and basic abstraction of different full-information distributed computing models and give a short introduction to computability in distributed computing. In this chapter we describe a basic wait-free read/write model. We also include other important read/write models which provide an atomic operation called *snapshot* and other iterated and non-iterated models.

1.1 Brief description of computability in Distributed Computing

Computability in distributed computing

Deciding if a *function* is computable by a Turing machine is a central concern in sequential computing. A *task* is the equivalent of a function in distributed computing. In a task T each of the $n+1$ processes is assigned an *input value* and each non-faulty process must decide an *output value*, both taken from finite sets. This assignment determines the *specification* of the task and initially each process is unaware of the others' input values. An *asynchronous distributed computing model* M consists of a set of entities, like processes, that communicate through some communication medium without restrictions on the speed of the processes, and satisfies some failures assumptions. For instance, in the WR, AS and IS models, processes communicate by *single-writer/multi-reader memories* and they can fail by *crashing*, see Sections 1.2 and 1.3. A process crashes if it halts and takes no further steps. In such case the process is called *faulty*. In this document we consider models in which at least one process is non-faulty. This kind of models are called *wait-free*. Therefore, a central question in distributed computing is: Is a task T solvable in an asynchronous distributed computing model M ? Being solvable means that each non-faulty process p_i, after communicating with the others, decides an output value which satisfies the specification of the task.

Some distributed tasks

In the literature we can find basic and important distributed tasks such as *consensus* (Lamport et al., 1982), *k-set agreement* (Chaudhuri, 1993), *renaming* (Attiya et al., 1990), *weak symmetry breaking* (Gafni et al., 2006), etc.

Consensus. In the *consensus* task every non-faulty process must agree on a single value (*agreement*) and the value decided is one of the input values proposed by a participating process (*validity*). For instance, in the *binary consensus* task the processes p and q can propose 0 or 1. Therefore, there exist only two possible output values, namely 0 and 1. Notice that if p proposes 0 and it runs solo then p must decide 0. Therefore, if q runs after p finishes the task then q must decide 0.

k-set agreement. A natural generalization of the consensus task is *k-set agreement*. In this task every non-faulty process must agree in at most k different values and these are inputs values proposed by participating processes. For instance, in the 2-set agreement task, suppose processes p, q and r propose the values 0, 1 and 2 respectively. Then if p and q run concurrently, they can decide its corresponding input values. But if r runs after p and q finishes the task then r must decide 0 or 1.

K-Renaming. In the *renaming* task each process is assigned a unique initial name, i.e. an integer in the interval $[1, N]$ for some large N. In order to reduce the size of the namespace, unique names must be chosen in the interval $[1, K]$ where K is smaller than N. In addition, the new names must satisfy the following conditions: (1) no two processes obtain the same name and (2) the new name obtained by a process is independent of its id.

Weak symmetry breaking. In this task the processes have no input values and they must decide on the possible output values of 0 or 1. In every execution in which all processes decide an output value, not all processes agree on the same one.

There exist some variants of these tasks. For instance, *long-lived renaming* (Moir and Anderson, 1995; Moir, 1998) and *group renaming* (Gafni, 2004; Afek et al., 2008). In the former each process can repeatedly acquire a new name and then release it. In the latter each process belongs to a group, knows the original name of its group and has to choose a new name for its group in such a way that two processes belonging to distinct groups choose distinct new names. Generalized symmetry breaking tasks are a family of tasks that include renaming and weak symmetry breaking (Imbs et al., 2011).

Solvability of some distributed tasks

Determining the question if a task T is solvable in a distributed computing model is a complex problem since this depends on the communication, timing and failure model. However, there are some important results in this direction. For instance, the well known *asynchronous computability theorem*, due to Herlihy and Shavit (1993, 1999), fully characterizes the solvability of decision tasks which can be solved in a wait-free manner where processes can write and take snapshots of a shared memory, see Section 1.3. Herlihy and Shavit (1993, 1999) also proved that k-set agreement task does not have a wait-free protocol in this model for $k \leq n$, where $n+1$ processes participate in the task. At the same time this impossibility result was proved by Borowsky and Gafni (1993a) and Saks and Zaharoglou (1993) in different computing models. Later on, it was also proved by Attiya and Rajsbaum (2002) in the non-iterated immediate snapshot model, see Section 1.4. In the case of K-renaming task Castañeda and Rajsbaum (2010, 2012) proved, in the immediate snapshot model, that there exists a wait-free renaming protocol for $K < 2n$ if and only if $n+1$ is not a prime power. They used the equivalence between $(2n-1)$-renaming and WSB for $n+1$

processes. Then WSB is solvable in the immediate snapshot model if and only if $n+1$ is a prime power.

Many mathematical tools have been used to get these results. For instance, since 1993 algebraic topology has been used to reason about asynchronous computations. The framework consists basically of modelling tasks and distributed models by means of simplicial complexes, and simplicial and carrier maps, see Section 3.2. Thus, different combinatorial and topological techniques and tools like connectivity, homotopy type, Sperner's lemma, etc are used to get impossibility results or bounds. This implies that the structure of the simplicial complexes associated to different distributed models plays an important role in this approach.

1.2 A Basic Wait-Free Read/Write Model

One of the basic distributed models is the *read/write* model (WR), e.g. (Attiya and Welch, 2004). It consists of $n+1$ processes denoted by the numbers $[n] = \{0, 1, \ldots, n\}$. A process is a deterministic (possibly infinite) state machine. Processes communicate through a shared memory array $\mathsf{mem}[0 \ldots n]$ which consists of $n+1$ single-writer/multi-reader atomic registers. Each process accesses the shared memory by invoking the atomic operations $\mathsf{write}(x)$ or $\mathsf{read}(j)$, $0 \leq j \leq n$. The $\mathsf{write}(x)$ operation is used by process i to write value x to register i, and process i can invoke $\mathsf{read}(j)$ to read register $\mathsf{mem}[j]$ for any $0 \leq j \leq n$. In general, the read/write model does not impose restrictions on the number of operations to execute by the processes. However, we restrict this number because we are interested in analyzing iterated models, which will be explained in Section 1.4. Therefore, in the one-round read/write model each process i has an input value, which may be its own id i. In its first operation, process i writes its input to $\mathsf{mem}[i]$, then it reads each of the $n+1$ registers in an arbitrary order. Such a sequence of operations, consisting of a write followed by all the reads, is abbreviated by $\mathsf{WScan}(x)$. For instance, in Figure 1.1 we consider three processes in which the values of $\mathsf{mem}[0]$ and $\mathsf{mem}[2]$ were initialized $\perp$ and the value of $\mathsf{mem}[1]$ in y. Then after the $\mathsf{write}(x)$ operation of process 0 the value of $\mathsf{mem}[0]$ changes to x.

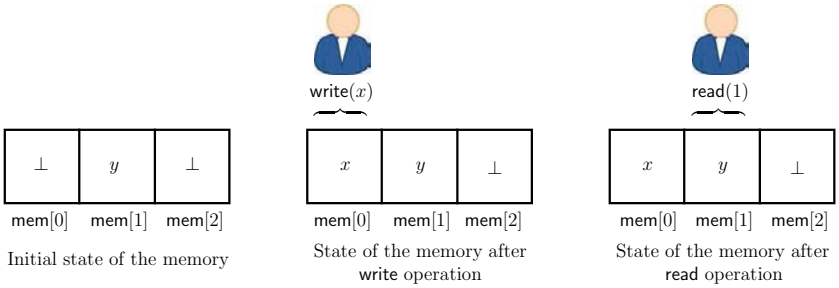


Figure 1.1: read and write operations of process 0

An *execution* consists of a (possible infinite) sequence of operations by the processes and we assume any sequence of the operations is a possible execution. Since processes can fail by crashing we consider also executions where possibly a proper subset of processes participate, consisting of an interleaving of the operations of those processes. These assumptions represent a wait-free model where any number of processes may fail by crashing. It is *wait-free* in the sense that each process is guaranteed to complete its operations regardless of the relative speeds of other processes even if some of them fail. The set of values read in an

execution α by process i is called the *local view* of i which is denoted by $view_i(\alpha)$. It consists of pairs (j, v), indicating that the value v was read from the j-th register. The set of all local views in the execution α is called the *view* of α and is denoted by $view(\alpha)$. For instance, in Figure 1.1 after the read(1) operation the local view of process 0 is $view_0(\alpha) = \{(1, \perp)\}$. Let $\mathcal{E}$ be a set of executions of the WR model. Then $\mathcal{E}$ can be partitioned into equivalence classes according to the following relation: $\alpha \sim \alpha'$ if $view(\alpha) = view(\alpha')$.

1.3 Snapshot Models

Here we consider two one-round read/write sub-models of WR called *atomic snapshot* (AS) (Herlihy and Shavit, 1999) and *immediate snapshot* (IS) (Borowsky and Gafni, 1993a; Saks and Zaharoglou, 2000) models which are subsets of executions of the WR one round model. In particular, the operations of immediate snapshot executions can be organized in *concurrency classes*, each one consisting of a set of writes by the set of processes participating in the concurrency class, followed by a read to all registers by each of these processes. Each process in the AS model accesses the shared memory by invoking the atomic operations write(x) or snapshot(). The snapshot() operation is used by process i to read the whole shared memory mem$[0 \ldots n]$ in a single atomic step. In its first operation, process i writes its input value to mem$[i]$, it then takes a snapshot of the memory. On the other hand, each process in the IS model accesses the shared memory by invoking the atomic operation wsnap(x). In this operation the process i can write its input value x to register i and can take a snapshot of the memory in one atomic step. This operation is called an *immediate snapshot* (Borowsky and Gafni, 1993b).

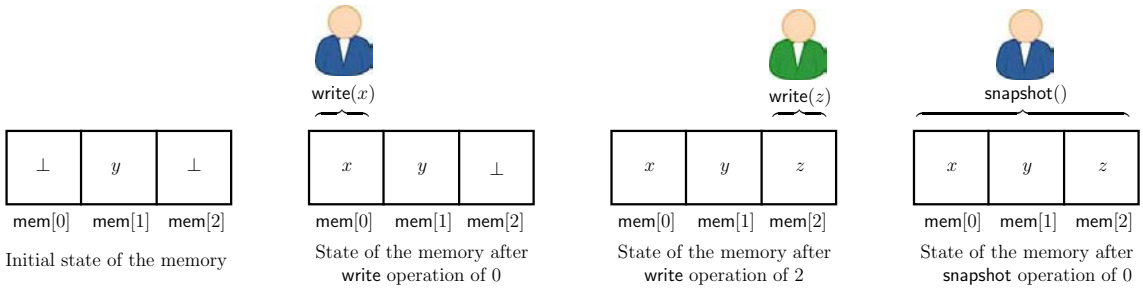


Figure 1.2: snapshot and write operations of process 0

For instance, Figure 1.2 corresponds to an execution α in the AS model. In this case notice that $view_0(\alpha) = \{(0, x), (1, y), (2, z)\}$. While in Figure 1.3, which corresponds to an execution α' in the IS model, $view_0(\alpha') = \{(0, x), (1, y), (2, \perp)\}$. Notice in Figure 1.2, process 2 executed write(z) after and and before operations write(x) and snapshot() of process 0, respectively.

1.4 Iterated and Non-Iterated Models

Here we consider full-information protocols, that is, one in which every process writes its complete state. Two other models can be derived from the WR model (or AS and IS): Iterated and Non-Iterated models. In the *iterated* WR model (IWR), processes communicate through a sequence of arrays. Computations proceed in a *round-based* pattern and in each

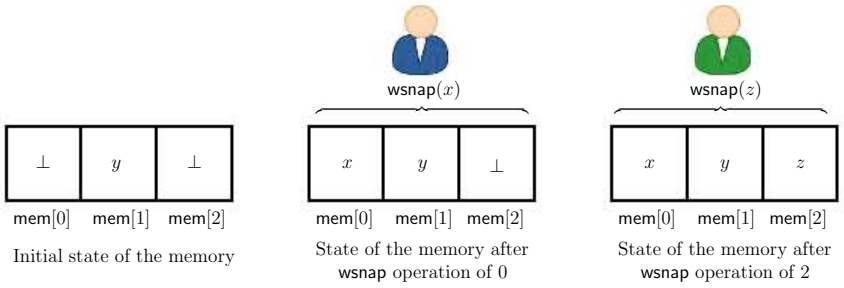


Figure 1.3: wsnap operation of process 0

iteration of the system each array is used at most once. They all go through the sequence of arrays mem_0, $\mathsf{mem}_1 \ldots$ in the same order, and in the r-th round, they access the r-th array, mem_r, exactly as in the one-round version of the WR model. Then every execution α in the IWR model can be seen as a sequence $\alpha = \alpha_0, \alpha_1, \ldots$ where α_r is an execution in the r-th round. Also note that the local view $view_i(\alpha)$ in the IWR can be represented as a sequence of local views in the WR model. In other words, $view_i(\alpha) = view_i(\alpha_0), view_i(\alpha_1), \ldots$ where $view_i(\alpha_r)$ is the local view of the process i in the execution α_r. Namely, process i executes $\mathsf{write}(x)$ to $\mathsf{mem}_r[i]$, where $x = view_i(\alpha_{r-1})$, and then reads one by one all entries j, $\mathsf{mem}_r[j]$, in arbitrary order. For instance, consider the execution $\alpha = \alpha_0, \alpha_1$ of Figure 1.4. It shows two rounds of the process 0 in the IIS. Notice that $view_0(\alpha_0) = \{(0, x), (1, y), (2, \bot)\}$ and $view_0(\alpha_1) = \{(0, view_0(\alpha_0)), (1, \bot), (2, \bot)\}$.

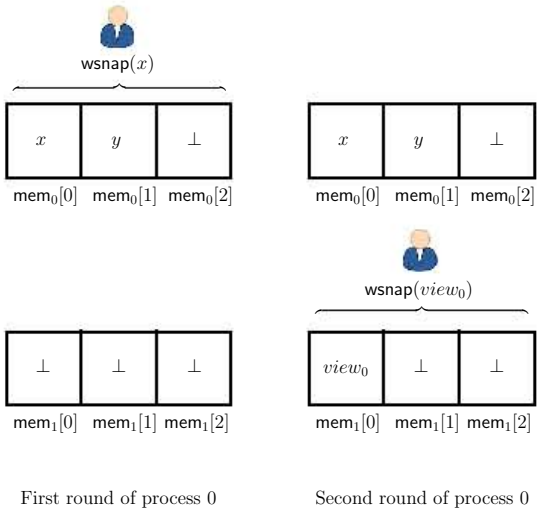


Figure 1.4: Two rounds of process 0

In the *non-iterated, multi-round* version of the WR (NIWR) model there is only one array mem, but processes can execute several rounds of writing and then reading one by one the entries of the array. Describing the local view of a process i in this model is more difficult, however, in the following example in the NIIS model we can get an idea of how it works. First, notice that the local view of a process depends on the round. Suppose that $view_i^r(\alpha)$ represents the r-th local view of the process i in the execution α. Then, in Figure 1.5 we have $view_0^1(\alpha) = \{(0, x), (1, y), (2, \bot)\}$ and $view_0^2(\alpha) = \{(0, view_0^1(\alpha)), (1, y), (2, \bot)\}$.

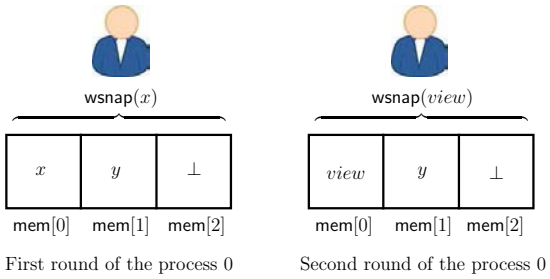


Figure 1.5: Two rounds of process 0 in NIIS

1.5 Chapter Summary

A distributed computing model is defined basically by type of communication mechanism, failures, and timing. For instance, in *shared memory* models each process communicates by executing operations of shared objects like *single-writer/multi-reader memories*, *atomic snapshot* and *immediate snapshot* objects (Rajsbaum, 2010; Herlihy et al., 2013b). One of the main concerns in distributed computing is to provide wait-free algorithms which solve distributed tasks. In this direction Herlihy and Shavit (1993, 1999) described a complete characterization of the distributed tasks for which there is a wait-free algorithm where $n+1$ processes can write and take a snapshot of a shared memory and up to n processes can fail. On the other hand, we know about the impossibility of k-set agreement in the snapshot models (Herlihy and Shavit, 1999; Attiya and Rajsbaum, 2002) or the existence of a wait-free algorithm for K-renaming in the model where $n+1$ processes can take immediate snapshots (Castañeda and Rajsbaum, 2010, 2012).

In the ACM Symposium on Theory of Computing Herlihy and Shavit (1993), Borowsky and Gafni (1993a) and Saks and Zaharoglou (1993) presented different papers in which they proved the impossibility of k-set agreement using combinatorial topology. Since then, the application of concepts from topology to distributed computing has been fruitful in characterizing synchronous and asynchronous distributed computing tasks and their solvability (Herlihy et al., 2013a). This relation is based on the fact that distributed models and tasks can be represented by simplicial complexes. For instance, in the impossibility result of k-set agreement, the study of connectivity of simplicial complex associated to the used model was highly important. Simplicial complexes associated to snapshot models have an elegant and nice combinatorial representation which permits the understanding of their connectivity. However, the simplicial complex associated to read/write models has not been studied since its combinatorial description is more complicated.

Chapter 2

Immediate Snapshot Object

In this chapter we describe two implementations of immediate snapshot object, the original by Borowsky and Gafni (1993b) and its recursive version due to Gafni and Rajsbaum (2010). We consider these implementations since this relates the models WR and IS. We also describe in detail the executions of the recursive implementation. This description will be fundamental in the topological analysis of the WR model.

2.1 Participating Set Problem

Problem

Participating set problem was introduced by Borowsky and Gafni (1993b) in order to describe the immediate snapshot object, see Section 1.3. In this problem a process i writes its id into a shared memory and returns the set of processes which wrote before or concurrent with i into the memory. Formally, the problem is defined by the conditions:

1. *Self-containment*: $i \in view_i$.
2. *Atomic snapshot*: For all i, j, either $view_i \subseteq view_j$ or $view_j \subseteq view_i$.
3. *Immediacy*: For all i, j, if $i \in view_j$ then $view_i \subseteq view_j$.

Notice that the first two conditions determine the specification of the atomic snapshot operation (Attiya et al., 1995b). However, the immediacy condition is not necessary for an atomic snapshot. Consider the execution of Figure 2.1, where first of all, the process 0 writes its id in the shared memory followed by process 2 and it by process 1. Notice that between the snapshot operations of the processes 0 and 2 the process 1 writes its id. In this execution the immediacy condition is not satisfied since $view_0$ is not contained in $view_2$. However, the execution of Figure 2.2 satisfies the requirements of the participating set problem. In this execution process 1 writes its id after process 0 executes a snapshot to the memory.

The atomic condition implies there exists a linear order $view_{i_0} \subset view_{i_1} \subset \cdots \subset view_{i_n}$. In addition, by the immediacy condition it is concluded that if $j \in T_k = view_{i_k} - view_{i_{k-1}}$ then $view_j = view_{i_k}$. Hence, an immediate snapshot is an operation in which $\mathsf{write}(i)$ and $\mathsf{snap}(i)$ are fused in a single operation $\mathsf{wsnap}(i)$, see Section 1.3.

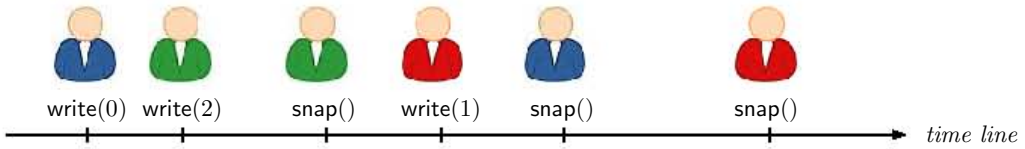


Figure 2.1: Atomic snapshot execution

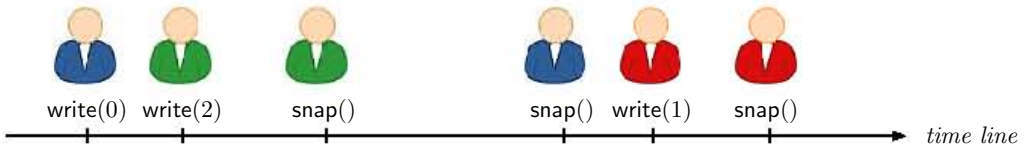


Figure 2.2: Immediate atomic snapshot execution

Solution

By means of the algorithm in Figure 2.3 Borowsky and Gafni (1993b) proved that participating set problem is solvable in the WR model. Notice that this algorithm is non-recursive and it is built on top of the non-iterated multi-round read/write model, see Sections 1.3 and 1.4. In this algorithm $n+1$ processes communicate by a shared memory $\mathsf{Level}[0 \dots n]$ which consists of $n+1$ single-writer/multi-reader atomic registers. Processes start in the level $n+2$ (line (1)) and continue moving down through the levels (line (3)). In the line (4) process i executes $\mathsf{WScan}(l_i)$ to the memory $\mathsf{Level}[0 \dots n]$, by performing first $\mathsf{write}(l_i)$, followed by $\mathsf{read}(j)$ for each $j \in [n]$ in any order. The set of values read S_i (each one with its location) is what the invocation of $\mathsf{WScan}(l_i)$ returns. Then $view_i$ contains all values $(j, l_j) \in S_i$ such that the process j is at the same level of i or below (line (5)). Notice that process i moves to level $l_i - 1$ if there are fewer than l_i processes at level l_i or below (line (6)).

Algorithm participating-set(i)
(1) $l_i \leftarrow n+2$
(2) **repeat**
(3) $\quad l_i \leftarrow l_i - 1$
(4) $\quad S_i \leftarrow \mathsf{WScan}(l_i)$
(5) $\quad view_i \leftarrow \{j \ : (j, l_j) \in S_i,\ l_j \leq l_i\}$
(6) **until** $|view_i| \geq l_i$
(7) return $view_i$.

Figure 2.3: Participating set code for process i

Lemma 1. *Let l be an integer such that $1 \leq l \leq n+2$. In the* participating-set *algorithm of Figure 2.3 it is satisfied*

$$|Set_l| = |\{j \ : \ \mathsf{Level}[j] \leq l\}| \leq l$$

Proof. Suppose there exists $1 \leq l' \leq n+2$ such that $|Set_{l'}| > l'$. Then according to line (3) of the algorithm 2.3 at least $l'+1$ processes must have moved from level $l'+1$ to l'. Let i

be one of the last processes on level $l'+1$ to execute $\mathsf{WScan}(l'+1)$. Then process i reads at least $l'+1$ processes at its own level or below which implies according to line (6) that process i will not move to level l'. This contradicts the assumption. □

From now on let l_i denote the last of $\mathsf{Level}[i]$ when process i returns.

Corollary 1. *For every process i, $|view_i| = l_i$.*

Proof. Consider $j \in view_i$. Then $\mathsf{Level}[j] \leq l_j \leq l_i$, hence $j \in Set_{l_i}$. In consequence $view_i \subseteq Set_{l_i}$. Therefore, by Lemma 1 and line (6) of Algorithm 2.3 $|view_i| = |Set_{l_i}| = l_i$. □

Proposition 1. *Algorithm 2.3 solves participating set problem.*

Proof. Self-containment condition is clear from definition of $\mathsf{WScan}()$. For the atomic condition let i, j be processes such that $l_i \leq l_j$. Then by Lemma 1 $view_i = S_{l_i} \subseteq S_{l_j} = view_j$. Now if $i \in view_j$ then $l_i \leq l_j$. Thus $view_i \subseteq view_j$. □

2.2 Implementations of Immediate Snapshot Object

Non-recursive Implementation

Borowsky and Gafni (1993b) introduced the *immediate snapshot* model which provides one-shot immediate snapshot object, see Section 1.3. An object of this kind can be accessed by a process invoking wsnap(). Recall that in this operation a process writes its input value into a shared memory and takes a snapshot of the memory in one atomic step. Formally, one-shot immediate snapshot object is defined by the conditions *self-containment*, *atomic snapshot*, and *immediacy* except that each local view is a set of pairs (i, v_i) where v_i is the input value of process i. Then participating set problem is a particular instance of the one-shot immediate snapshot problem. Figure 2.4 shows the non-recursive implementation designed by Borowsky and Gafni (1993b) in which participating-set algorithm is used as subroutine.

Algorithm wsnap(v_i)
(1) write(v_i)
(2) $part_i \leftarrow$ participating-set(i)
(3) **foreach** $j \in part_i$ **do** $view_i \leftarrow view_i \cup \{(j, \mathsf{mem}[j])\}$
(4) return $view_i$.

Figure 2.4: Non-recursive code for process i

In the wsnap algorithm $n+1$ processes communicate by a shared memory $\mathsf{mem}[0 \ldots n]$ which consists of $n+1$ single-writer/multi-reader atomic registers. In line (1) process i writes its input value v_i and in line (2) the algorithm participating-set is executed by process i. Then $view_i$ consists of all pairs $(j, \mathsf{mem}[j])$ for each $j \in part_i$ (line (3)) where $part_i$ is what participating-set returns.

Proposition 2. *Algorithm 2.4 solves one-shot immediate snapshot problem.*

Proof. First note that each process writes its input value into the shared memory (line (1)) before to invoke participating-set algorithm (line (2)). Then specification of participating-set algorithm proves that conditions of one-shot immediate snapshot problem are satisfied. □

Recursive Implementation

Gafni and Rajsbaum (2010) described simple and elegant recursive distributed algorithms for some important tasks such as atomic snapshot, immediate snapshot and renaming, thus showing the benefits of designing distributed algorithms using recursion. Consider their recursive version algorithm IS for the iterated WR model, presented in Figure 2.5. Processes go through a series of disjoint shared memory arrays $\mathsf{mem}_0, \mathsf{mem}_1, \ldots, \mathsf{mem}_n$. Each array mem_k is accessed by process i invoking $\mathsf{WScan}(v_i)$ in the recursive call $\text{IS}(n+1-k)$. Then, in line (1), process i executes $\text{WScan}(v_i)$ to the memory mem_k and the set of values read $view_i$ (each one with its location) is what the invocation of $\mathsf{WScan}(v_i)$ returns. In line (2) the process i checks if $view_i$ contains $n+1-k$ id's, else $\text{IS}(n-k)$ is again invoked on the next shared memory in line (3). It is important to note that in each recursive call $\text{IS}(n+1-k)$ at least one process returns with $|view_i| = n+1-k$, given that $n+1-k$ processes invoked IS. For instance, in the first recursive call $\text{IS}(n+1)$ the last process to write reads $n+1$ values and terminates the algorithm.

Algorithm $\text{IS}(n+1)$
(1) $view_i \leftarrow \mathsf{WScan}(v_i)$
(2) **if** $|view_i| = n+1$ **then** return $view_i$
(3) **else** return $\text{IS}(n)$.

Figure 2.5: Recursive code for process i

Proposition 3. *Algorithm 2.5 solves one-shot immediate snapshot problem.*

Proof. The proof is by induction on the number of processes. Base case two processes can be verified considering the views of processes. Now suppose that algorithm 2.5 solves the one-shot immediate snapshot problem for k processes with $k \leq n$. For $n+1$ processes let S be the set of processes that terminate the algorithm when $view_i = [n]$ for each $i \in S$. By the induction hypothesis algorithm 2.5 solves the one-shot immediate snapshot problem for the processes in $[n] \setminus S$. Hence, self-contained, atomic snapshot and immediacy conditions are satisfied since $view_j \subseteq [n]$ for all $j \in [n] \setminus S$. Therefore, algorithm 2.5 solves the one-shot immediate snapshot problem. □

Description of the Executions of the Recursive Implementation

As we mentioned in Section 1.4 every execution in the IWR model can be seen as a sequence of executions in the one-round version of WR model. This facilitates the process of analyzing the recursive implementation of Gafni and Rajsbaum (2010). For convenience, from now on, the input value of each process is its id. Since algorithm 2.5 is built on top of the iterated multi-round read/write model every execution can be represented by a finite sequence $\alpha = \alpha_0, \alpha_1, \ldots, \alpha_l$ with α_k an execution of the WR one round model. Every process that takes a step in α_k invokes the recursive call with $\text{IS}(n+1-k)$. Since at least one process terminates the algorithm the length $l(\alpha) = l+1$ is at most $n+1$. The last returned local view in execution α for process i is denoted $view_i(\alpha)$, and the set of all local views is denoted by $view(\alpha)$. For instance, figure 2.6 shows an execution of the algorithm IS in the first memory for three processes where each of them invokes IS(3). In this execution $view_0 = \{0,1,2\}$, $view_1 = \{1,2\}$ and $view_2 = \{0,2\}$.

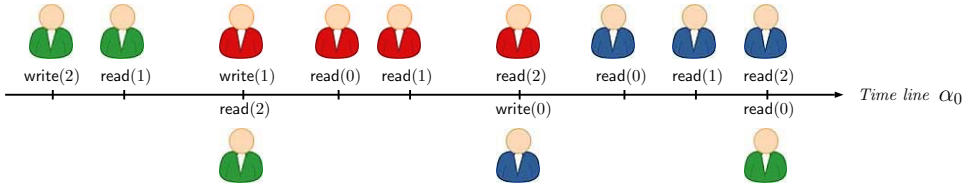


Figure 2.6: Execution first memory of recursive implementation

Therefore, the processes 1 and 2 have to invoke IS(2) while 0 returns its view. Now if the execution in the second memory is given by figure 2.7 then 1 and 2 return the views $view_1 = view_2 = \{1, 2\}$. Notice that if $\mathcal{E}_l$ denotes the set of views of all executions α with $l(\alpha) = l + 1$ then $\mathcal{E}_n \subseteq \cdots \subseteq \mathcal{E}_0$. In particular, $\mathcal{E}_0$ corresponds to the views of executions of the one round WR of Section 1.2. Also, $\mathcal{E}_n$ corresponds to the views of the immediate snapshot model, see Proposition 3 (Theorem 1 of Gafni and Rajsbaum (2010)).

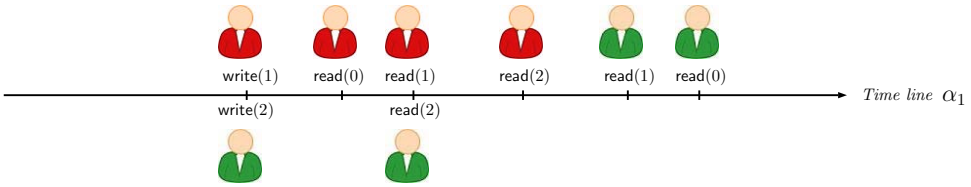


Figure 2.7: Execution second memory of recursive implementation

2.3 Properties of Immediate Snapshot executions

In this section we describe some additional properties of executions of recursive implementation in Subsection 2.2. Recall from Section 1.2 that an execution can be seen as a linear order on the set of write and read operations. For a subset $I \subseteq [n]$ let

$$\mathcal{O}_I = \{w_i, r_i(j) \ : \ i \in I, \ j \in [n]\},$$

with $I = \mathcal{O}_I = \emptyset$.

Definition 1. A *wr-execution on* I is a pair $\alpha = (\mathcal{O}_I, \rightarrow_\alpha)$ with $\rightarrow_\alpha$ a linear order on $\mathcal{O}_I$ such that $w_i \rightarrow_\alpha r_i(j)$ for all $j \in [n]$. The set I is called the *id* set of α which is denoted by $id(\alpha)$.

Hence the local view of i in α is

$$view_i(\alpha) = \{j \in I \ : \ w_j \rightarrow_\alpha r_i(j)\},$$

and the view of α is $view(\alpha) = \{(i, view_i(\alpha)) \ : \ i \in I\}$. Notice that the chain $w_i \rightarrow_\alpha r_i(j_0) \rightarrow_\alpha \cdots \rightarrow_\alpha r_i(j_n)$ represents the invoking of WScan by process i in the *wr*-execution α. Consider a *wr*-execution α and suppose that the order in which the process i reads the array $\mathsf{mem}[0 \ldots n]$ is given by $r_i(j_0) \rightarrow_\alpha \cdots \rightarrow_\alpha r_i(j_n)$. If every write operation w_k satisfies $w_k \rightarrow_\alpha r_i(j_0)$ or $r_i(j_n) \rightarrow_\alpha w_k$ then $view_i(\alpha)$ corresponds to an atomic snapshot.

As a consequence, every execution in the snapshot model and immediate snapshot model corresponds to an execution in the read/write model. For instance, in the wr-execution

$$\alpha : w_2 \to r_2(0) \to w_0 \to r_0(0) \to r_0(1) \to r_0(2) \to w_1 \to \\ \to r_1(0) \to r_2(1) \to r_1(1) \to r_2(2) \to r_1(2)$$

$view_0(\alpha) = \{0, 2\}$ and $view_1(\alpha) = [2]$ are immediate snapshots. This means processes 0 and 2 could have read the array instantaneously. In contrast, $view_2(\alpha) = \{1, 2\}$ does not correspond to a snapshot.

Proposition 4. *Let α be a wr-execution on I. Then there exists $j \in I$ such that $view_j(\alpha) = I$.*

Proof. Consider the process j such that $w_i \to_\alpha w_j$ for all i. □

Let α be a wr-execution. For $0 \leq k \leq n$, define

$$id_k(\alpha) = \{j \in id(\alpha) \ : \ |view_j(\alpha)| = n + 1 - k\}.$$

Definition 2. An *IS-execution* is a finite sequence $\alpha = \alpha_0, \ldots, \alpha_l$ such that

1. α_0 is a wr-execution on $[n]$.
2. α_{k+1} is a wr-execution on $id(\alpha_k) - id_k(\alpha_k)$.

Given an IS-execution α, Proposition 4 implies that $l(\alpha) = l + 1 \leq n + 1$. Moreover $id(\alpha_{k+1}) \subseteq id(\alpha_k)$ for all $0 \leq k \leq l - 1$. Then $|id(\alpha_k)| \leq n + 1 - k$. In addition, according to the protocol in Figure 2.5, the local view of i is

$$view_i(\alpha) = view_i(\alpha_k)$$

if $i \in id(\alpha_k) - id(\alpha_{k+1})$ and $view_i(\alpha) = view_i(\alpha_l)$ for $k = l$. Hence the view of α is defined as

$$view(\alpha) = \{(i, view_i(\alpha)) \ : \ i \in [n]\}.$$

Two IS-executions α, α' are *equivalent* if $view(\alpha) = view(\alpha')$, denoted $\alpha \sim \alpha'$.

Proposition 5. *Let α and α' be IS-executions with $l(\alpha) = l(\alpha)$. Given $0 \leq k \leq l$,*

1. *If $\alpha \sim \alpha'$ then $id(\alpha_k) = id(\alpha'_k)$.*
2. *If $\alpha_k \sim \alpha'_k$ then $\alpha \sim \alpha'$.*

Proof. To prove 1 note that $id_k(\alpha_k) = id_k(\alpha'_k)$ since $view(\alpha) = view(\alpha')$. Hence $id(\alpha_k) = id(\alpha'_k)$. Now, 2 is a consequence of the equality $view(\alpha_k) = view(\alpha'_k)$. □

Proposition 6. *Let $\alpha = \alpha_0, \ldots, \alpha_{l+1}$ be an IS-execution, $l(\alpha) = l + 2$. Then $view(\alpha) = view(\alpha')$ for some IS-execution α' such that $l(\alpha') = l + 1$.*

Under the conditions of the previous proposition, $\alpha' = \alpha_0, \ldots, \alpha_{l-1}, \alpha'_l$, where

$$view_i(\alpha'_l) = \begin{cases} view_i(\alpha_l), & \text{if } i \in id_l(\alpha_l) \\ view_i(\alpha_{l+1}), & \text{if } i \notin id_l(\alpha_l) \end{cases}$$

It follows that $\mathcal{E}_l = \{view(\alpha) \ : \ \alpha = \alpha_0, \ldots, \alpha_l\}$. Thus, Proposition 6 implies $\mathcal{E}_{l+1} \subseteq \mathcal{E}_l$. For instance, consider the IS-execution $\alpha = \alpha_0, \alpha_1, \alpha_2$ where

$$\alpha_0 : w_0 \to r_0(0) \to r_0(1) \to r_0(2) \to w_1 \to r_1(0) \to r_1(1) \to r_1(2) \to w_2 \to r_2(0) \to r_2(1) \to r_2(2)$$

$$\alpha_1 : w_0 \to r_0(0) \to r_0(1) \to r_0(2) \to w_1 \to r_1(0) \to r_1(1) \to r_1(2)$$

$$\alpha_2 : w_0 \to r_0(0) \to r_0(1) \to r_0(2)$$

So $view(\alpha) = \{(0, \{0\}), (1, \{0, 1\}), (2, \{0, 1, 2\})\} \in \mathcal{E}_2 \subseteq \mathcal{E}_1 \subseteq \mathcal{E}_0$. Let $S_{[n]}$ denote the permutation group of $[n]$. Notice that if the id's of processes in a wr-execution α on I are permuted according to $\pi \in S_{[n]}$, we obtain a new linear order on $\pi(I)$. In other words, $\alpha' = \pi(\alpha)$ is the wr-execution defined by

1. $w_i \to_\alpha w_j$ if and only if $w_{\pi(i)} \to_{\alpha'} w_{\pi(j)}$.
2. $w_j \to_\alpha r_i(j)$ if and only if $w_{\pi(j)} \to_{\alpha'} r_{\pi(i)}(\pi(j))$.

This implies the following proposition.

Proposition 7. *Let α be a wr-execution on I and π be a permutation of $S_{[n]}$. Then,*

1. *$\alpha' = \pi(\alpha)$ is a wr-execution.*
2. *if $\sigma = view(\alpha)$ then $\pi(\sigma) = view(\pi(\alpha))$.*

2.4 Chapter Summary

Although immediate atomic snapshot executions are a special class of read/write executions, they have the same computational power. However, proving impossibility results is one of benefits of immediate snapshot model introduced by Borowsky and Gafni (1993b). In the literature, there are several wait-free implementations of atomic snapshot objects, however, to the best of our knowledge there are only two implementations of immediate snapshot object. The original version due to Borowsky and Gafni (1993b) and its recursive version by Gafni and Rajsbaum (2010). Maybe one of the advantages of the recursive version, in contrast with its original version, lies on the fact that it is built on top of the iterated multi-round read/write model. Then every execution of the recursive implementation can be expressed by a sequence of executions of one-round read/write model. Moreover, the executions of recursive implementations can be formalized to a better understanding. These implementations relate the basic one-round read/write model and immediate snapshot model. In consequence, the study from a topological point of view of read/write models can be done by means of immediate snapshot models.

Chapter 3

Combinatorial Topology and Distributed Computing

In the previous chapters we have talked about using topology in distributed computing. Here we introduce distributed computing by means combinatorial topology, see (Herlihy et al., 2013a). But first we give some standard technical definitions from combinatorial topology, see (Jonsson, 2008; Kozlov, 2015). Also we define simplicial complexes associated to executions of recursive implementation of immediate snapshot object, see Figure 2.5.

3.1 Elements of Combinatorial Topology and Collapsibility

Definition 3. An (abstract) *simplicial complex* Δ on a finite set V is a collection of non empty subsets of V such that for any $v \in V$, $\{v\} \in \Delta$, and if $\sigma \in \Delta$ and $\tau \subseteq \sigma$ then $\tau \in \Delta$.

The elements of V are called *vertices* and the elements of Δ *simplices*. Every subset τ of σ with $\sigma \in \Delta$ is called a *face* of σ, we will denote this with $\tau < \sigma$. Also if ρ contains σ with $\rho \in \Delta$ then ρ is called a *co-face* of σ. The *dimension* of a simplex σ is $\dim(\sigma) = k$ if σ contains $k+1$ vertices. Sometimes, if it is necessary, we will indicate this with a superscript $\sigma^{(k)}$. For instance the vertices are 0-simplices. Given a non empty set S, Δ^S denotes the standard simplicial complex whose vertex set is S and every non empty subset of S is a simplex. In particular for a positive integer n, Δ^n denotes the simplicial complex $\Delta^{[n]}$. From now on we identify a complex Δ with its collection of simplices. A simplicial complex Δ is pure if all its maximal simplices have the same dimension. The dimension of a simplicial complex Δ is the largest dimension of any of its simplices. For instance Figure 3.1 shows a simplicial complex Δ of dimension 2 in which $\sigma_0 = \{v_0, v_1, v_2\}$, $\sigma_1 = \{v_1, v_2, v_3\}$, $\tau_5 = \{v_3, v_4\}$ and $\tau_6 = \{v_1, v_4\}$ are the maximal simplices.

Definition 4. Let Δ_1 and Δ_2 be simplicial complexes. A map $f : V_1 \to V_2$ is a simplicial map if $\sigma \in \Delta_1$ then $f(\sigma) \in \Delta_2$.

Consider the simplicial complexes Δ of Figure 3.1 and Δ_3 of Figure 3.2. Then the map given by $f(v_0) = f(v_2) = f(v_3) = v_3$, $f(v_1) = v_1$ and $f(v_4) = v_4$ is a simplicial map.

For every pair of simplices $\tau < \sigma$ we denote by

$$I(\tau, \sigma) = \{\rho \in \Delta \ : \ \tau \leq \rho \leq \sigma\},$$

or in general $I(\tau) = \{\rho \in \Delta \ : \ \tau \leq \rho\}$.

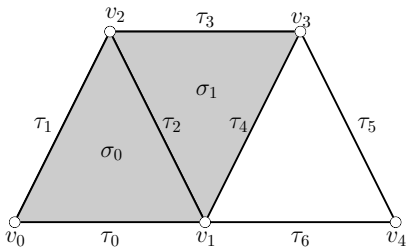


Figure 3.1: Simplicial complex Δ

Definition 5. A simplex τ of Δ is called *free* if there exists a maximal simplex σ such that $\tau < \sigma$ and no other maximal simplex contains τ.

Definition 6. Let τ be a free simplex of a simplicial complex Δ. Then the procedure of removing every simplex of $I(\tau)$ is called a *collapse*. In particular if $\dim \sigma = \dim \tau + 1$, the collapse is called *elementary*.

Let Δ and Γ be simplicial complexes, Δ *is collapsible to* Γ if there exists a sequence of collapses leading from Δ to Γ. The corresponding procedure is denoted by $\Delta \searrow \Gamma$. In particular, if the collapse is made with a free simplex τ, it is denoted by $\Delta \searrow_\tau \Gamma$. If Γ is a simplex then Δ is *collapsible*. Since, each elementary collapse is a deformation retract[1], so if Δ is collapsible to Γ, the complexes Δ and Γ are homotopy equivalent. In particular, every collapsible complex is contractible[2] (the converse is false (Zeeman, 1963)). Now, consider the simplicial complex Δ of Figure 3.1. Then $c_1 = (\tau_0, v_0, \tau_2, v_2)$ is a sequence of collapses and so,

$$\Delta \searrow_{\tau_0} \Delta_1 \searrow_{v_0} \Delta_2' \searrow_{\tau_2} \Delta_2 \searrow_{v_2} \Delta_3$$

There are other sequences of collapses from Δ to Δ_3 like $c_2 = (\tau_0, \tau_2, v_0, v_2)$ and $c_3 = (v_0, v_2)$, see Figure 3.2. Notice that c_1 and c_2 are sequences of elementary collapses and there exists a permutation of free faces v_0 and τ_2. In other words, $I(v_0) \cap I(\tau_2) = \emptyset$ in the simplicial complex Δ_1. In consequence we can can collapse first v_0 and then τ_2 or viceversa in order to obtain Δ_2. This implies that we can collapse them in a parallel way.

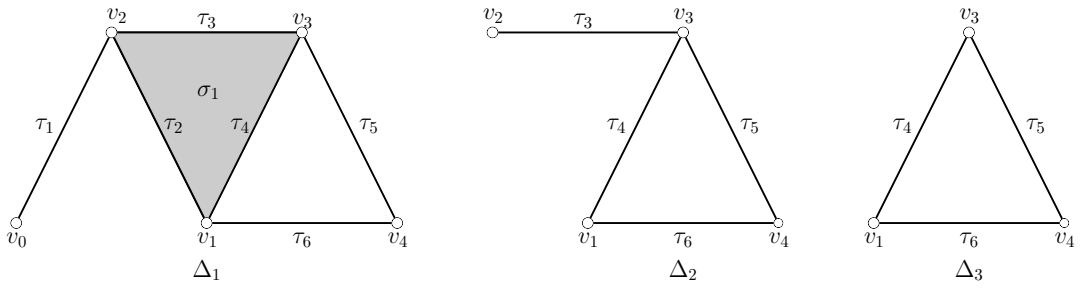


Figure 3.2: Collapsibility procedure of Δ

Definition 8 (Kozlov, 2015) is a generalization of procedure of collapsing several free faces in a parallel way. Specifically it gives a procedure to collapse a simplicial complex, by collapsing simultaneously by entire orbits of a group action on the vertex set.

[1] If $\Delta \searrow_\tau \Gamma$ then there exists a continuous deformation $\varphi : |\Delta| \to |\Gamma|$ such that $\varphi(x) = x$ for all $x \in |\Gamma|$ and φ and $id_{|\Delta|}$ are homotopic.

[2] A simplicial complex Δ is contractible if $|\Delta|$ is homotopic equivalent to a space with a single point.

Definition 7. Let G be a finite group. G acts on a simplicial complex Δ if there exists a function $\varphi : G \times V \to V$ such that for every $g \in G$ the function $g : V \to V$ given by $g(v) = \varphi(g, v)$ is a simplicial map.

Definition 8. Let Δ be a simplicial complex with a simplicial action of a finite group G. A simplex τ is called *G-free* if

1. τ is free and
2. for all $g \in G$ such that $g(\tau) \neq \tau$, $I(\tau) \cap I(g(\tau)) = \emptyset$

If τ is G-free then the procedure of removing every simplex $\rho \in \bigcup_{g \in G} I(g(\tau))$ is called a *G-collapse of* Δ.

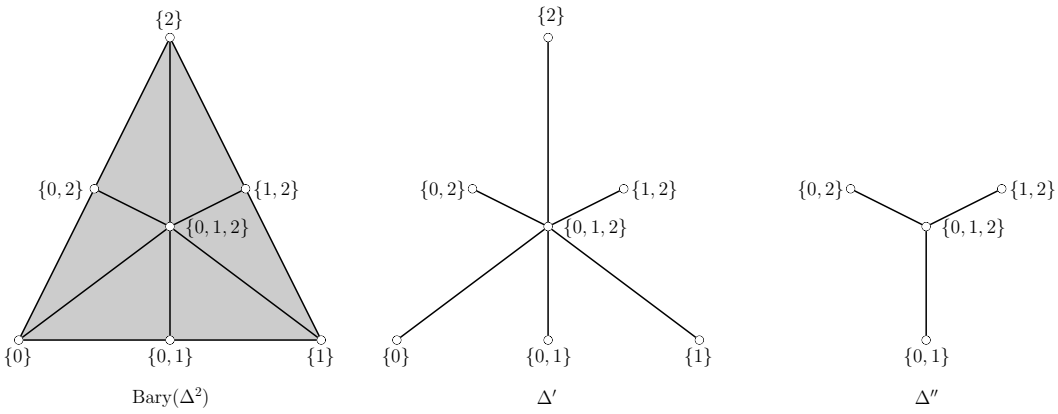


Figure 3.3: Barycentric subdivision of Δ^2

Since, if τ is G-free then $g(\tau)$ is free as well, the previous definition guarantees that all collapses in the orbit of τ can be done in any order *i.e* every G-collapse is a collapse. A simplicial complex Δ is *G-collapsible to* Γ if there exist a sequence of G-collapses leading from Δ to Γ. It is denoted by $\Delta \searrow_G \Gamma$. In a similar way, if the G-collapse is elementary with a G-free simplex τ, the notation $\Delta \searrow_{G(\tau)} \Gamma$ will be used. In the case Γ is a simplex, Δ is called *G-collapsible.* For instance, consider the simplicial complex Δ^2, τ a 1-face of Δ^2 and the action of $S_{[2]}$ over Δ^2 then τ is free but not $S_{[2]}$-free. However, if we consider the barycentric subdivision $\text{Bary}(\Delta^2)$ with the action of $S_{[2]}$ given by $\varphi(f, \sigma) = f(\sigma)$ then $\text{Bary}(\Delta^2)$ is $S_{[2]}$-collapsible, see Figure 3.3. Specifically,

$$\text{Bary}(\Delta^2) \searrow_{S_{[2]}(\tau_0)} \Delta' \searrow_{S_{[2]}(\tau_1)} \Delta'' \searrow_{S_{[2]}(\tau_2)} \{0, 1, 2\}$$

where $\tau_0 = \{\{0\}, \{0, 1\}\}$, $\tau_1 = \{\{0\}\}$, and $\tau_2 = \{\{0, 1\}\}$.

3.2 Computability through Combinatorial Topology

In this section we describe the topological approach to distributed computing given by Herlihy and Shavit (1993). A complete description can be found in (Herlihy et al., 2012) and (Herlihy et al., 2013a).

Simplicial Complexes model Distributed Tasks

Recall from Section 1.1 that in a distributed task each process p_i is assigned an input ini_i value and each non-faulty process must decide an output value out_i. In consequence, if only processes $p_{i_0}, \ldots, p_{i_k}$ participate in the task then $\sigma = \{(p_{i_0}, in_{i_0}), \ldots, (p_{i_k}, in_{i_k})\}$ represents an initial state of the system, σ is called an *initial configuration*. Notice, if σ is a initial configuration then so is every subset $\tau \subseteq \sigma$ because τ represents an initial configuration where fewer processes participate. Therefore the set of all possible initial configurations describes a simplicial complex which is n-chromatic in the sense of definition 9.

Definition 9. An n-dimensional simplicial complex Δ is called *n-chromatic* if $w, w' \in \sigma \in \Delta$ then $w = (i, v_i)$ with $i \in [n]$ and $(w)_1 \neq (w')_1$ where $(\cdot)_i$ is the projection on the i-th component.

Let Δ be a n-chromatic simplicial complex. For convenience we denote

$$id(\Delta) = \bigcup_{\sigma \in \Delta} id(\sigma) \quad \text{and} \quad val(\Delta) = \bigcup_{\sigma \in \Delta} val(\sigma)$$

where $id(\sigma) = (\sigma)_1$ and $val(\sigma) = (\sigma)_2$.

Definition 10. A decision task is a triple $T = (\mathcal{I}, \mathcal{O}, \mathcal{S})$ where $\mathcal{I}$ and $\mathcal{O}$ are n-chromatic complexes and $\mathcal{S}$ is a map which satisfies:

1. $\mathcal{S}(\sigma)$ is a subcomplex of $\mathcal{O}$ for every $\sigma \in \mathcal{I}$.
2. $\mathcal{S}(\tau) \subseteq \mathcal{S}(\sigma)$ if $\tau < \sigma$.
3. $id(\sigma) = id(\mathcal{S}(\sigma))$.

In general a map $\mathcal{S}$ which satisfies the conditions 1 and 2 of previous definition is called a *carrier map*, moreover if $\mathcal{S}$ satisfies the condition 3 then $\mathcal{S}$ is called *chromatic*. The simplicial complexes $\mathcal{I}$ and $\mathcal{O}$ are called the input and output complexes of T respectively and $\mathcal{S}$ the specification map. The input and output complexes are defined on vertex sets $val(\mathcal{I}) = V^{\text{in}}$ and $val(\mathcal{O}) = V^{\text{out}}$ which are the valid input and output values of processes. In consequence in a decision task $T = (\mathcal{I}, \mathcal{O}, \mathcal{S})$ each vertex represents a state (initial or final) of a single process and the simplicial complexes $\mathcal{I}$ and $\mathcal{O}$ define all possible input and output configurations respectively. The specification task describes which outputs can be produced for each initial configuration. In other words, if the system begins in initial configuration $\sigma \in \mathcal{I}$ then every execution must finish in some $\tau \in \mathcal{S}(\sigma)$. In addition, if σ is an initial configuration in which the processes in $\sigma' < \sigma$ finish the task in some execution before the others then the processes in $\sigma \setminus \sigma'$ must decide values compatible with the values already decided by the processes in σ'. For instance in the binary consensus task if $\sigma = \{(p, 0), (q, 1)\}$ then $\mathcal{S}(\sigma) = \mathcal{O}$. In consequence if the process p in a execution runs solo then it must decide 0, so the process q must be able to agree with process p.

Examples of Distributed Tasks

In the following we define some distributed tasks in the sense of Definition 10. These tasks were introduced in Chapter 1. Consider a non empty subset I of $[n]$.

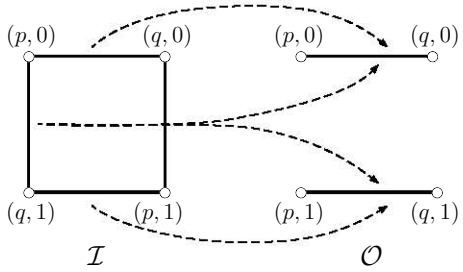


Figure 3.4: The binary consensus task.

Binary Consensus Task. Each process proposes a binary input value and all participating processes must agree on a single value, see Figure 3.4. Hence $V^{in} = V^{out} = V = \{0, 1\}$ and

1. Input Complex: $\sigma = \{(i, v_i) \ : \ i \in I, \ v_i \in \{0, 1\}\} \in \mathcal{I}$.
2. Output Complex: $\tau = \{(i, v_i) \ : \ i \in I, \ v_i \in \mathrm{V}\} \in \mathcal{O}$ if $|val(\tau)| = 1$.
3. Specification Map: $\tau \in \mathcal{S}(\sigma)$ if $val(\tau) \subseteq val(\sigma)$ and $id(\tau) \subseteq id(\sigma)$.

k-set agreement. First recall that consensus task is a particular case of set agreement. In this task $V^{\mathrm{in}} = V^{\mathrm{out}} = V$ and

1. Input Complex: $\sigma = \{(i, v_i) \ : \ i \in I, \ v_i \in \mathrm{V}\} \in \mathcal{I}$.
2. Output Complex: $\tau = \{(i, v_i) \ : \ i \in I, \ v_i \in \mathrm{V}\} \in \mathcal{O}$ if $|val(\tau)| \leq k$.
3. Specification Map: $\tau \in \mathcal{S}(\sigma)$ if $val(\tau) \subseteq val(\sigma)$ and $id(\tau) \subseteq id(\sigma)$.

For instance consider 2-set agreement for three processes. Suppose processes p, q and r propose values 0, 1 and 2 respectively. Then output complex has 21 maximal simplices, see Figure 3.5.

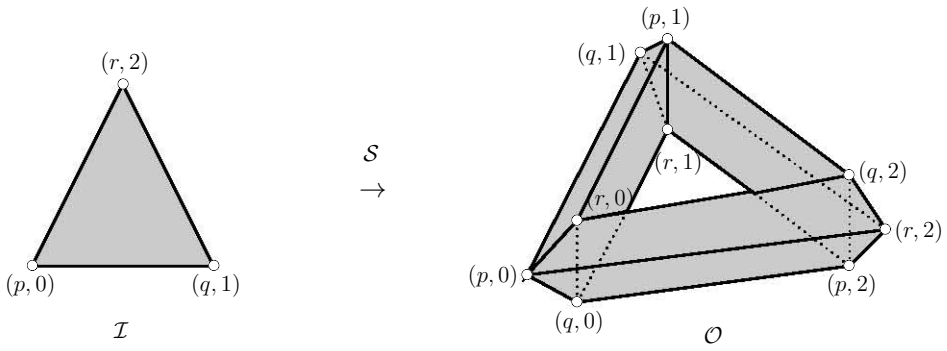


Figure 3.5: 2-set agreement task.

Weak Symmetry Breaking (WSB). In this task each process has not an input value and must decide a binary value. In case of all processes decide an output value in a execution then at least one decides 0 and at least one 1. Then $V^{in} = \{\perp\}$, $V^{out} = \{0, 1\}$ and

1. Input Complex: $\sigma = \{(i, \perp) \ : \ i \in I\} \in \mathcal{I}$.
2. Output Complex: $\tau = \{(i, v_i) \ : \ i \in I, \ v_i \in \{0, 1\}\} \in \mathcal{O}$ and if $\dim \tau = n$ the $val(\sigma) = \{0, 1\}$.
3. Specification Map: $\tau \in \mathcal{S}(\sigma)$ if $id(\tau) \subseteq id(\sigma)$.

In Figure 3.6 is shown the WSB task for three process p, q and r.

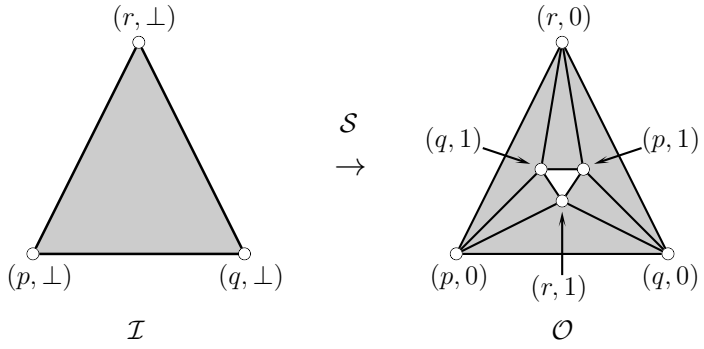


Figure 3.6: Weak Symmetry Breaking task.

K-Renaming. In this task the input value (or $name_i$) of each process is taken from $[N]$. Then they have to choose a new name such that (1) $name_i \in [K]$, (2) no two processes have the same name and (3) the new name is independent of its id. Therefore $V^{in} = [N]$, $V^{out} = [K]$ and

1. Input Complex: $\sigma = \{(i, name_i) \ : \ i \in I, \ name_i \in [N]\} \in \mathcal{I}$.
2. Output Complex: $\tau = \{(i, name_i) \ : \ i \in I, \ name_i \in [K]\} \in \mathcal{O}$ if $|val(\tau)| = \dim \tau + 1$.
3. Specification Map: $\tau \in \mathcal{S}(\sigma)$ if $name_i \neq name_j$ for $i \neq j$ and $id(\tau) \subseteq id(\sigma)$.

Solvability in Distributed Computing

A *protocol* is a finite distributed algorithm in which initially each process knows its input value, but no the other's. Each non faulty process communicates with the others and eventually halts deciding an output value. Therefore, a protocol solves a task T if in each execution the output values form a valid output configuration. In consequence, we have the following definition.

Definition 11. A chromatic carrier map $\mathcal{P}$ *solves a task* $T = (\mathcal{I}, \mathcal{O}, \mathcal{S})$ if there exists a simplicial map δ from $\mathcal{P}(\mathcal{I})$ to $\mathcal{O}$ such that

1. $\delta(\mathcal{P}(\sigma)) \subseteq \mathcal{S}(\sigma)$ for every simplex $\sigma \in \mathcal{I}$.

2. $id(v) = id(\delta(v))$ for every vertex $v \in \mathcal{P}(\mathcal{I})$.

The simplicial complex $\mathcal{P}(\mathcal{I})$ is called the *protocol complex* associated to distributed model and δ is called a *decision map*. Then $\mathcal{P}(\sigma)$ represents all possible executions of the protocol if the system begins in the initial configuration σ. Notice each vertex of $\mathcal{P}(\mathcal{I})$ represents a final state in the model considered. For instance, in the models considered in the Chapter 1 each vertex (or final state) is a pair $(i, view_i)$. In the following we give a short description of solvability of set agreement and renaming in which different topological techniques are used.

k-set agreement. The ACT (Herlihy and Shavit, 1999) gives necessary and sufficient conditions which characterize the decision tasks that can be solved in a wait-free manner by asynchronous processes that communicate by writing and taking atomic snapshots of a shared memory. They proved that the simplicial complex $\mathcal{P}_{AS}(\sigma^{(n)})$ is n-connected where $\mathcal{P}_{AS}$ is the carrier map associated to the distributed model considered by (Herlihy and Shavit, 1999) and σ is an initial configuration. This property is used to prove that there exists a chromatic carrier map sub such that it is a subdivision and $\text{sub}(\sigma)$ is a subcomplex of $\mathcal{P}_{AS}(\sigma)$. Hence the Sperner's Lemma implies that n-set agreement is not solvable. In general, Herlihy and Rajsbaum (2000) proved that if there exists a distributed protocol $\mathcal{P}$ for k-set agreement and acyclic carrier map[3] Σ from $\sigma^{(d)}$ to $\mathcal{P}(\sigma)$ such that

$$val(\delta(\Sigma(\tau))) \subseteq val(\tau)$$

for all simplex $\tau < \sigma$ where δ is the decision map then $k \geq d+1$. In particular if $\mathcal{P}_{AS}$ solves set agreement then considering $\Sigma(\sigma) = \mathcal{P}_{AS}(\sigma)$ where $|val(\sigma)| = n+1$ it is obtained that $k \geq n+1$.

K-renaming. Attiya et al. (1990) described a wait-free protocol for $K \geq 2n$ and proved that there is not solution for $K \leq n+1$ in the message-passing model. All these results can be extended to wait-free read/write shared memory models (Attiya et al., 1995a). Castañeda and Rajsbaum (2010, 2012) proved that there exists an infinite values of n for which $(2n-1)$-renaming is solvable in the immediate snapshot model. In the proof they used the equivalence between WSB and $(2n-1)$-renaming (Gafni et al., 2006) and a particular class of pseudo-manifolds called *divided images*. This class were defined by Attiya and Rajsbaum (2002) in order to study immediate snapshot models. This kind of complexes are orientable and graph-connected. A pseudo-manifold $\mathcal{K}$ is *orientable* if it is possible to give an orientation to each of its n-simplices such that if σ, σ' share an $(n-1)$-face τ then τ gets opposite induced orientations from σ and σ'. On the other hand $\mathcal{K}$ is graph-connected if its i-graph is connected. The i-graph associated to $\mathcal{K}$ is formed by the vertex set of all its i-simplices and two vertices form an edge if they share an $(i-1)$-simplex. They described the number of monochromatic n-simplices $\mathcal{C}$ if the vertices of protocol complex $\mathcal{P}(\sigma)$ considered are colored with 0 and 1. In fact they showed that there exist $k_i \in \mathbb{Z}$ such that

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

It is known that binomial numbers $\binom{n+1}{1}, \ldots, \binom{n+1}{n}$ are relatively prime if and only if $n+1$ is not a prime power (Dickson, 2005). In consequence, there exists a wait-free protocol for $(2n-1)$-renaming if and only if $n+1$ is not a prime power.

[3] A carrier map Σ from Δ to Γ is acyclic if $H_q(\Sigma(\sigma)) = 0$ for all $\sigma \in \Delta$.

3.3 Protocol Complex of IS-Executions

Structure of the Protocol Complex

Let n be a positive integer.

Definition 12. The abstract simplicial complex $\mathsf{WR}_l(\Delta^n)$ with $0 \leq l \leq n$ consists of the set of vertices $V = \{(i, view_i) \ : \ i \in view_i \subseteq [n]\}$. A subset $\sigma \subseteq V$ forms a simplex if only if there exists an IS-execution α such that $l(\alpha) = l + 1$ and $\sigma \subseteq view(\alpha)$.

The complex $\mathsf{WR}_0(\Delta^n)$ is called the *read/write protocol complex*, and it will be denoted by $\mathsf{WR}(\Delta^n)$. Protocol complexes for the particular cases $n = 1$ and $n = 2$ are shown in Figure 3.7. Notice that the protocol complexes $\chi(\Delta^n)$ and View^n associated to IS and AS models respectively are subcomplexes of $\mathsf{WR}(\Delta^n)$. Figure 3.8 shows the complexes $\mathsf{WR}_l(\Delta^2)$.

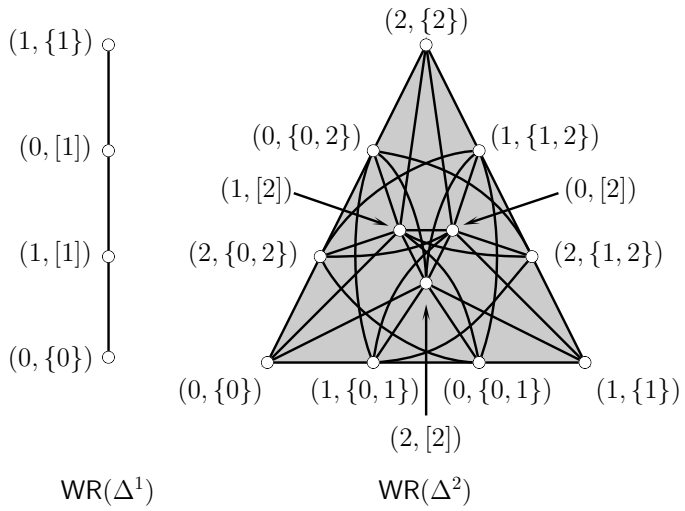


Figure 3.7: Protocol complex for $n = 1$ and $n = 2$.

From now on we will write WR_l instead of $\mathsf{WR}_l(\Delta^n)$ unless we specify the standard complex. Notice that $\mathsf{WR}_{n-1} = \mathsf{WR}_n$. In addition Proposition 6 implies every maximal simplex of WR_{l+1} is a simplex of WR_l, hence WR_{l+1} is a subcomplex of WR_l.

$$\chi(\Delta^n) = \mathsf{WR}_n \subseteq \cdots \subseteq \mathsf{WR}_0 = \mathsf{WR}(\Delta^n).$$

In (Kozlov, 2015) a combinatorial description of the protocol complex View^n associated to the atomic snapshot model is given. There, every simplex of View^n is represented as a matrix. Likewise, every simplex $\sigma \in \mathsf{WR}(\Delta^n)$ can be expressed as a $2 \times t$ matrix of subsets of $[n]$

$$M_\sigma = \begin{pmatrix} [n] & V_1 & \cdots & V_t \\ I_0 & I_1 & \cdots & I_t \end{pmatrix} \tag{3.1}$$

where $I_i \cap I_j = \emptyset$ with $i \neq j$, $I_i \subseteq V_j \neq \emptyset$ for all $j \leq i$ and $V_i \neq V_j$ if $i \neq j$. According to Definition 1, to every matrix M as in (3.1) we can associate a wr-execution α, such that the simplex

$$\sigma_M = \bigcup_{i=0}^{t} \bigcup_{p \in I_i} \{(p, V_i)\}.$$

satisfies $\sigma_M \subseteq view(\alpha)$. Notice that $\dim(\sigma_M) = |I_0| + |I_1| + \cdots + |I_t| - 1$.

Definition 13. Let N and M be matrices as in (3.1). Then N is a subset of M, $N \subset M$, if $\sigma_N \subset \sigma_M$.

In other words $N \subset M$ if for every column (I'_i, V'_i) of N there exists (I_j, V_j) in M such that $I'_i \subseteq I_j$ and $V'_i = V_j$. In spite of the fact that (3.1) is not unique there exists an equivalence relation on the set of all matrices given by: $M \sim M'$ if and only if $\sigma_M = \sigma_{M'}$. Then we have the following proposition.

Proposition 8. *There exists a bijection between* $\mathsf{WR}(\Delta^n)$ *and the set of all equivalence classes* $M(\Delta^n)$.

Proposition 8 implies that $M(\Delta^n)$ is a simplicial complex. Let σ be a simplex of $\mathsf{WR}(\Delta^n)$, the equivalence class of M_σ is called the matrix representation of σ.

Definition 14. Let M be a matrix of the form (3.1). Then we define the partial snapshot of M, denoted by $psnap(M)$, as the highest number s such that

1. $V_i \subset V_s \subset \cdots \subset V_1 \subset [n]$ for all $i < s$.
2. $I_i \subseteq V_i \setminus V_{i+1}$ for $0 \leq i \leq s$.
3. $I_s \cap (V_{s+1} \cup \cdots \cup V_t) = \emptyset$.

Proposition 9. *Let* M *and* M' *be matrices as in (3.1). If* $M \sim M'$ *then* $psnap(M) = psnap(M')$.

Proof. Suppose

$$M = \begin{pmatrix} [n] & V_1 & \cdots & V_t \\ I_0 & I_1 & \cdots & I_t \end{pmatrix} \text{ and } M' = \begin{pmatrix} [n] & V'_1 & \cdots & V'_t \\ I'_0 & I'_1 & \cdots & I'_t \end{pmatrix}$$

where $psnap(M) = s \leq s' = psnap(M')$. We shall prove by induction $V'_i = V_i$ for all $0 \leq i \leq s$. Base case $i = 0$ is clear. Then suppose $V'_i = V_j$ for some $j > i$ hence $V_i \supset V_j = V'_i \supset V_i$, a contradiction. Therefore $s = s'$. □

Consider the simplex σ given by the matrix

$$M_\sigma = \begin{pmatrix} [4] & \{1,3,4\} & \{3,4\} & \{1,4\} \\ \{0,2\} & \{1\} & \{3\} & \{4\} \end{pmatrix}$$

with $0 = psnap(M_\sigma)$. There exists an IS-execution α with $l = l(\alpha) \leq 2$ such that $\sigma \subseteq view(\alpha)$. In addition if $l = 2$ then we can say processes 0 and 2 terminated the Algorithm 2.5 in the first recursive call, while in the second recursive call no process read 4 processes participating in the second memory layer. Therefore $\sigma \in \mathsf{WR}_2 \subset \mathsf{WR}_1 \subset \mathsf{WR}_0$ but $\sigma \notin \mathsf{WR}_3$.

Additional properties of the protocol complex

From now on we will use the matrix representation to describe the simplices of $\mathsf{WR}(\Delta^n)$ and we suppose that $l < n - 1$.

Theorem 1. *Let* σ *be a simplex of* $\mathsf{WR}(\Delta^n)$ *with* $s = psnap(M_\sigma)$. *Then,* $\sigma \in \mathsf{WR}_l$ *if and only if*

$$|V^1_\sigma| = |V_{s+1} \cup \cdots \cup V_t| \leq n + 1 - l$$

Proof. Suppose that $\sigma \in \mathsf{WR}_l$. Then there exists an IS-execution $\alpha = \alpha_0, \ldots, \alpha_l$ such that $\sigma \subseteq view(\alpha)$. Thus, by Definition 2, $V_i \subseteq id(\alpha_l)$ for all $s < i \leq t$, and hence

$$V_\sigma^1 = V_{s+1} \cup \cdots \cup V_t \subseteq id(\alpha_l).$$

Therefore $|id(\alpha_l)| \leq n+1-l$ proves the first implication. The other direction is clear from Definition 2. □

Let σ be a simplex of $\mathsf{WR}(\Delta^n)$.

Corollary 2. *If σ satisfies $|V_s| \leq n+2-l$ where $s = psnap(M_\sigma)$ then $\sigma \in \mathsf{WR}_l$.*

Proof. By Definition 14, $V_\sigma^1 \subset V_s$ then $|V_\sigma^1| < |V_s| \leq n+2-l$. □

Corollary 3. *$|V_\sigma^1| \geq n+1-l$ if only if $\sigma \notin \mathsf{WR}_{l+1}$.*

For the following, σ will be a simplex such that $|V_\sigma^1| = n+1-l$. For each IS-execution $\alpha = \alpha_0, \ldots, \alpha_l$ such that $\sigma \subseteq view(\alpha)$, it holds $id(\alpha_l) = V_\sigma^1$. Thus, we can suppose $V_{s+1} = V_\sigma^1$, see Proposition 4. We define

$$I_\sigma^s = V_\sigma^1 \setminus V_\sigma^2$$

if $V_\sigma^2 \subset V_\sigma^1$ or

$$I_\sigma^s = V_\sigma^1 \setminus (I_{s+2} \cup \cdots \cup I_t)$$

in the other case, where $V_\sigma^2 = V_{s+2} \cup \cdots \cup V_t$. Note that I_σ^s is non-empty because of $|V_\sigma^1| = n+1-l$. Now consider the simplices σ^- and σ^+ given by

$$\sigma^- = \begin{pmatrix} [n] & \cdots & V_s & V_{s+1} & V_{s+2} & \cdots & V_t \\ I_0 & \cdots & I_s & I_{s+1} \setminus I_\sigma^s & I_{s+2} & \cdots & I_t \end{pmatrix}$$

and

$$\sigma^+ = \begin{pmatrix} [n] & \cdots & V_s & V_{s+1} & V_{s+2} & \cdots & V_t \\ I_0 & \cdots & I_s & I_{s+1} \cup I_\sigma^s & I_{s+2} & \cdots & I_t \end{pmatrix}.$$

Proposition 10. *Simplices σ^- and σ^+ satisfy the following statements:*

1. $\sigma^- \leq \sigma \leq \sigma^+$.
2. $|V_{\sigma^-}^1| = |V_{\sigma^+}^1| = n+1-l$.
3. *If $\sigma^- \leq \tau \leq \sigma^+$ then $\tau^- = \sigma^-$ and $\tau^+ = \sigma^+$.*

Proof. All items are clear from the definitions of σ^- and σ^+ and Definition 13. □

Consider the interval

$$I(\sigma) = \{\tau \in \mathsf{WR}_l \ : \ \sigma^- \subseteq \tau \subseteq \sigma^+\}.$$

Then the previous proposition implies:

Corollary 4. *Let σ, τ be simplices of WR_l*

1. *$I(\sigma) \cap I(\tau) = \emptyset$ or $I(\sigma) = I(\tau)$.*
2. *$I(\sigma) \cap \mathsf{WR}_{l+1} = \emptyset$.*

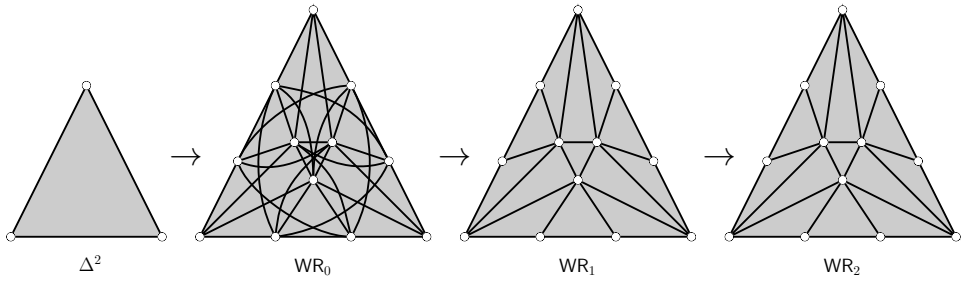


Figure 3.8: Complexes WR_l.

Remark 1. Notice that Proposition 10 and Corollary 4 imply that the set

$$L = \{\sigma \in \mathsf{WR}_l \ : \ |V^1_\sigma| = n + 1 - l \ \}$$

can be partitioned in intervals $I(\sigma)$. In other words there exist simplices $\sigma_1, \ldots, \sigma_k$ such that

$$L = I(\sigma_1) \cup \cdots \cup I(\sigma_k).$$

On the other hand, by Proposition 7 there exists an equivalence relation on WR_l given by: $\sigma \sim \sigma'$ if only if there exists $\pi \in S_{[n]}$ such that $\sigma = \pi(\sigma')$. In addition

$$\pi(\sigma) = \begin{pmatrix} [n] & \pi(V_1) & \cdots & \pi(V_t) \\ \pi(I_0) & \pi(I_1) & \cdots & \pi(I_t) \end{pmatrix}.$$

Example 1. In Figure 3.7, consider the simplex σ given by

$$\sigma = \begin{pmatrix} [2] & \{1,2\} & \{0,2\} \\ \{0\} & \{1\} & \{2\} \end{pmatrix}$$

and the permutation $\pi(0) = 1$, $\pi(1) = 2$ and $\pi(2) = 0$. Then

$$\pi(\sigma) = \begin{pmatrix} [2] & \{0,2\} & \{0,1\} \\ \{1\} & \{2\} & \{0\} \end{pmatrix}.$$

Furthermore $\pi(\sigma) \in \mathsf{WR}(\Delta^2)$ for all $\pi \in S_{[n]}$.

Proposition 11. *Let $\sigma \in \mathsf{WR}_l$ be a simplex. Then*

1. $\pi(\sigma) \in \mathsf{WR}_l$.
2. $\pi(\sigma^-) = \pi(\sigma)^-$ *and* $\pi(\sigma^+) = \pi(\sigma)^+$.

Proof. Equality $\pi(V^1_\sigma) = V^1_{\pi(\sigma)}$ proves the first item. The second item is a consequence of

$$\pi(I_{s+1} \setminus I^s_\sigma) = \pi(I_{s+1}) \setminus \pi(I^s_\sigma)$$

and

$$\pi(I_{s+1} \cup I^s_\sigma) = \pi(I_{s+1}) \cup \pi(I^s_\sigma).$$

□

Remark 2. Propositions 10 and 11 imply that for every $\pi \in S_{[n]}$ such that $\pi(\sigma) \neq \sigma$,

$$I(\sigma) \cap I(\pi(\sigma)) = \emptyset$$

Hence, if $I(\overline{\sigma})$ denotes the set

$$I(\overline{\sigma}) = \bigcup_{\pi \in S_{[n]}} I(\pi(\sigma))$$

there exist simplices $\rho_1, \ldots, \rho_m$ such that $L = I(\overline{\rho}_1) \cup \cdots \cup I(\overline{\rho}_m)$.

3.4 Chapter Summary

One of the most outstanding discoveries in distributed computing is the application of techniques from topology to prove results about computability in resilient distributed systems. Herlihy and Shavit (1999) and Saks and Zaharoglou (2000) shared the 2004 Gödel Prize for applications of topology to the theory of distributed computing. Distributed computing through combinatorial topology basically models decision tasks by simplicial complexes and simplicial and carrier maps. A decision task is a triple $T = (\mathcal{I}, \mathcal{O}, \mathcal{S})$ where $\mathcal{I}$ and $\mathcal{O}$ are simplicial complexes and $\mathcal{S}$ is a carrier map. In this approach a vertex of simplicial complex $\mathcal{I}$ (or $\mathcal{O}$) model a initial state (or final state) of a process. A distributed computing protocol is modeled by a carrier map $\mathcal{P}$ where $\mathcal{P}(\sigma)$ represents all possible executions if the system begins in the initial configuration σ, which is a simplex of $\mathcal{I}$. Then $T = (\mathcal{I}, \mathcal{O}, \mathcal{S})$ is solvable by a distributed protocol $\mathcal{P}$ if there exists a simplicial map δ such that for every initial configuration $\sigma \in \mathcal{I}$, $\delta(\mathcal{P}(\sigma)) \subseteq \mathcal{S}(\sigma)$. This topological approach to distributed computing has been used to prove impossibility and bounds results for important decision tasks, using different techniques taken from combinatorial and algebraic topology. For instance Herlihy and Shavit (1993) considered the distributed protocol $\mathcal{P}_{AS}$, which is computational equivalent to any wait-free protocol in the read/write shared memory model, where the asynchronous processes communicate by writing and taking atomic snapshot of a shared memory. They showed that the simplicial complex $\mathcal{P}(\sigma)$ is n-connected. Then using the Sperner's Lemma they proved that k-set agreement is impossible in the read/write shared memory model. On the other hand, Castañeda and Rajsbaum (2010, 2012) showed that there exists an infinite values of n for which $(2n-1)$-renaming task is solvable considering the protocol $\mathcal{P}_{IS}$ where processes can communicate by taking immediate snapshots. They showed that $\mathcal{P}_{IS}(\sigma)$ is a divided image, a particular class of pseudo-manifolds, and provided a formula for the monochromatic n-simplices. In consequence, they proved that there exists a wait-free protocol for $(2n-1)$-renaming if only if $n+1$ is not a prime power. Extended material about this topic can be found in the works of Conde and Rajsbaum (2012), Herlihy et al. (2012), and Herlihy et al. (2013a). The protocols considered in many of results about computability in distributed computing have been widely studied due to their nice geometric representation, for instance the atomic and immediate snapshot protocols. The recursive algorithm which implements the immediate snapshot operation provides a new class of protocol complexes $\mathsf{WR}_l(\Delta^n)$, $0 \leq l \leq n$, where $\mathsf{WR}_0(\Delta^n)$ and $\mathsf{WR}_n(\Delta^n) = \chi(\Delta^n)$ are the protocol complexes associated to read/write and immediate snapshot models respectively. We can simplify a complicated simplicial complex, eliminating its free faces, in order to study its homotopy type. This procedure is called collapsibility which is a fundamental tool in algebraic and combinatorial topology. However determining the free faces of a simplicial complex can be complicated. Discrete Morse theory is a tool for determining equivalence classes between simplicial complexes. The equivalence classes are provided

by the homotopy type. This theory was developed by Forman (1998, 2001) and for these structures, applications of the discrete theory are often more natural, as well as simpler and more straightforward to apply.

Chapter 4

Notions of Discrete Morse Theory

In this chapter basic notions of discrete Morse theory are reviewed (Forman, 1995, 1998). Theorem 6 is seems new; it will be used in the proof of the collapsibility of $\mathsf{WR}(\Delta^n)$. We include the proof of some important results in this theory, to help in the understanding of the proof of Theorem 6.

4.1 Preliminares

Recall that we use $\tau < \sigma$ to denote that τ is a face of σ. For a simplex σ of dimension k, we sometimes add a superscript to indicate its dimension, writing $\sigma^{(k)}$.

Let Δ be a simplicial complex and $f : \Delta \longrightarrow \mathbb{R}$ be a function. For every simplex $\sigma^{(k)}$, let

1. $N_1(\sigma, f) = \{\tau^{(k-1)} < \sigma \ : \ f(\tau) \geq f(\sigma)\}$.
2. $N_2(\sigma, f) = \{\rho^{(k+1)} > \sigma \ : \ f(\rho) \leq f(\sigma)\}$.

Definition 15 (discrete Morse function)**.** A function $f : \Delta \longrightarrow \mathbb{R}$ is a *discrete Morse function* if for each $\sigma^{(k)}$,

1. $|N_1(\sigma, f)| \leq 1$, and
2. $|N_2(\sigma, f)| \leq 1$.

Lemma 2. *Let f be a discrete Morse function on Δ. The cardinalities in* 1 *and* 2 *of Definition 15 cannot both be equal to* 1 *for a simplex σ.*

Proof. Suppose that $|N_1(\sigma, f)| = |N_2(\sigma, f)| = 1$. Then there exist unique simplices $\tau^{(k-1)}$ and $\rho^{(k+1)}$ such that $\tau < \sigma < \rho$ and $f(\tau) \geq f(\sigma) \geq f(\rho)$. On the other, for a simplex $\sigma'^{(k)}$ such that $\tau < \sigma' < \rho$ with $\sigma' \neq \sigma$, $f(\tau) < f(\sigma') < f(\rho)$. Hence

$$f(\sigma') > f(\tau) \geq f(\rho) > f(\sigma')$$

which is a contradiction. □

In the case $N_1(\sigma, f)$ and $N_2(\sigma, f)$ are empty the simplex σ is called *critical*, and it is called *regular* otherwise. Lemma 2 implies that regular simplices come in pairs. Hence there is a bijection between $N_1(\Delta, f)$ and $N_2(\Delta, f)$ where $N_i(\Delta, f) = \cup_{\sigma \in \Delta} N_i(\sigma, f)$. Now, if $N_0(\Delta, f)$ denotes the set of critical simplices, then Δ can be partitioned as follows,

$$\Delta = N_0(\Delta, f) \cup N_1(\Delta, f) \cup N_2(\Delta, f).$$

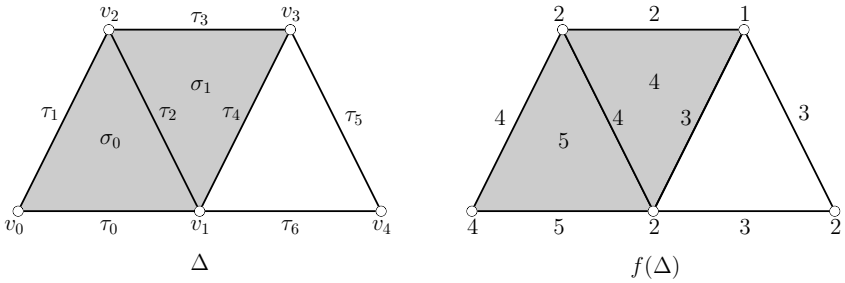


Figure 4.1: A discrete Morse function f.

If σ is a regular simplex, $pair(\sigma)$ will denote the simplex such that $pair(\sigma) \in N_1(\sigma, f)$ or $pair(\sigma) \in N_2(\sigma, f)$. For instance, in the discrete Morse function of Figure 4.1 $\tau_0 \in N_1(\sigma_0, f)$ and $\sigma_0 \in N_2(\tau_0, f)$ hence $\sigma_0 = pair(\tau_0)$.

Lemma 3. *For every discrete Morse function f on Δ, there exists a Morse function f' on Δ such that,*

1. *For every regular pair $(\tau^{(k)}, \sigma^{(k+1)})$, $f(\tau) > f(\sigma)$.*
2. *$N_i(\Delta, f) = N_i(\Delta, f')$ for all $i = 0, 1, 2$.*

Proof. Let $(\tau^{(k)}, \sigma^{(k+1)})$ be a regular pair and suppose $f(\tau) = f(\sigma)$. Consider $\epsilon > 0$ such that

$$\epsilon < \min\{f(\sigma') - f(\tau) \ : \ \sigma' \neq \sigma\}$$

where $\tau < \sigma'^{(k+1)}$. Let

$$f'(\rho) = \begin{cases} f(\rho), & \text{if } \rho \neq \tau; \\ f(\rho) + \epsilon, & \text{if } \rho = \tau. \end{cases}$$

Then by construction $f(\tau) > f(\sigma)$ and $f(\rho) < f(\tau) < f(\sigma')$ for every $\rho^{(k-1)}, \sigma'^{(k+1)}$ such that $\rho < \tau < \sigma'$. Similarly we can define a function f' in which $f'(\sigma) = f(\sigma) - \epsilon$. □

Figure 4.2 shows the discrete function f' associated to function f of Figure 4.1.

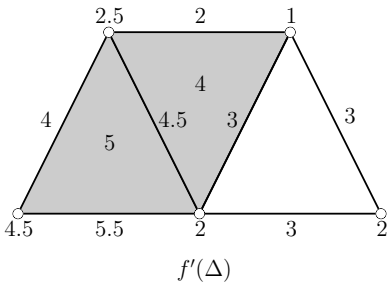


Figure 4.2: Discrete Morse function f' associated to function f of Figure 4.1

4.2 Basic Results

For any $a \in \mathbb{R}$ the *sublevel complex* is defined as,

$$\Delta_a = \bigcup_{f(\sigma)\leq a} \bigcup_{\tau\leq\sigma} \tau.$$

See Figure 4.3. Now we can state the following results which can be found in (Forman, 1998).

Theorem 2. *If $f^{-1}(a,b]$ contains no critical simplices then Δ_b collapses to Δ_a.*

Proof. First we will prove $\rho \in \Delta_b$ if only if $\sigma = pair(\rho) \in \Delta_b$. Consider the following cases:

1. If $\sigma < \rho$ then $\sigma \in \Delta_b$.
2. If $\rho < \sigma$ with $\sigma \notin \Delta_b$ then $f(\rho) \geq f(\sigma) > b$. On the other hand, there exists $\sigma'^{(k)}$ with $\sigma' \neq \sigma$ such that $\rho < \sigma'$ and $f(\sigma') \leq b$. Hence $b < f(\rho) < f(\sigma') \leq b$ which is a contradiction.

So, if $\rho \in \Delta_b$ then $\sigma \in \Delta_b$. The other direction is deduced from the fact $pair(pair(\rho)) = \rho$. Now suppose $f^{-1}(a,b] = \{\rho_1, \ldots, \rho_l\}$. By Lemma 3 we can suppose $f(\rho_l) > \cdots > f(\rho_1)$. Then the interval $(a,b]$ can be partitioned into smaller subintervals $(a_i, b_i]$ such that there is a single regular simplex ρ_i with $f(\rho_i) \in (a_i, b_i]$. Therefore suppose that $l = 1$. If $\rho_1 \in \Delta_a$ then $\Delta_b = \Delta_a$. In the other case $\Delta_b = \Delta_a \cup \{\rho_1, pair(\rho_1)\}$. This implies either ρ or $pair(\rho)$ is a free face. Hence, $\Delta_b \searrow \Delta_a$. □

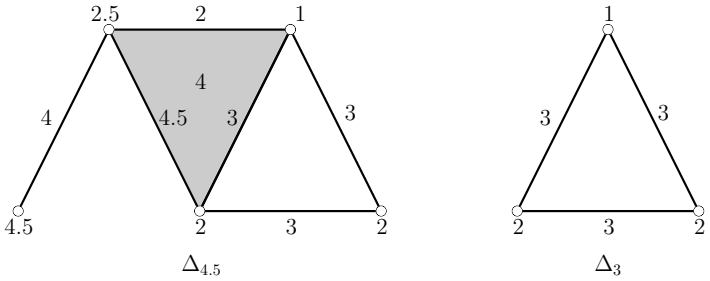


Figure 4.3: Examples of sublevel complexes Δ_b.

Corollary 5. *If f has one critical point then Δ is collapsible.*

Notice that the set of regular simplices of Δ are given in pairs $(\tau^{(k-1)}, \sigma^{(k)})$ where $f(\tau) \geq f(\sigma)$. In fact given two pairs v, v' of regular simplices, $v \cap v' = \emptyset$.

Definition 16 (Discrete vector field)**.** A *discrete vector field* of a simplicial complex Δ is a set W of pairs $v = (\tau^{(k-1)}, \sigma^{(k)})$ such that:

1. For every $\rho \in \Delta$, there exists at most one $v \in W$ such that $\rho \in v$.
2. For each $v = (\tau^{(k-1)}, \sigma^{(k)})$, $\tau^{(k-1)} < \sigma^{(k)}$.

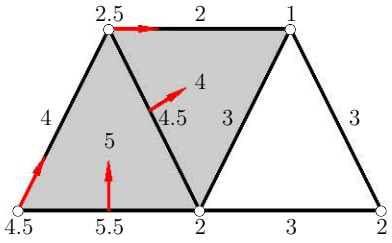


Figure 4.4: Gradient vector field of f'.

So every discrete Morse function f of Δ defines a discrete vector field W. In this case the discrete vector field is called the *gradient vector field* of f. Figure 4.4 shows the gradient vector of the discrete Morse function f given in Figure 4.1. Lemma 3 implies the gradient vector field of f and f' are the same.

A non-trivial *path* of dimension k in a discrete vector field W is a sequence of simplices,

$$\tau_0^{(k-1)}, \sigma_0^{(k)}, \tau_1^{(k-1)}, \sigma_1^{(k)}, \ldots, \tau_m^{(k-1)}, \sigma_m^{(k)}, \tau_{m+1}^{(k-1)} \tag{4.1}$$

with $k \geq 1$, such that for every i, $0 \leq i \leq m$, $v = (\tau_i^{(k-1)}, \sigma_i^{(k)}) \in W$ and $\tau_{i+1} < \sigma_i$. In the case $\tau_0 = \tau_{m+1}$, the path is called *closed*. Notice if W is a gradient vector field of a Morse function f then a sequence of simplices as in (4.1) satisfies

$$f(\tau_0) \geq f(\sigma_0) > f(\tau_1) \geq f(\sigma_1) > \cdots > f(\tau_m) \geq f(\sigma_m) > f(\tau_{m+1}).$$

In Proposition 12 we present a construction of a discrete vector field for a simplicial complex Δ. Consider the simplices $\tau, \sigma \in \Delta$ such that $\tau < \sigma$ and a fix vertex $x \in \sigma \setminus \tau$. For every subset $s \subseteq \sigma \setminus \tau$ with $x \in s$ let

$$\begin{aligned} \tau' &= \tau'(\tau, s, x) = \tau \cup (s \setminus x) \\ \sigma' &= \sigma'(\tau, s, x) = \tau \cup s. \end{aligned}$$

We define $W(\tau, \sigma, x)$ as the set of all pairs $v = (\tau', \sigma')$. Notice that $I(\tau, \sigma)$ is the union of all vectors $v \in W(\tau, \sigma, x)$.

Proposition 12. *If there exist simplices $\tau_1, \sigma_1, \ldots, \tau_k, \sigma_k$ such that*

1. $\tau_i \subset \sigma_i$.
2. $I(\tau_i, \sigma_i) \cap I(\tau_j, \sigma_j) = \emptyset$.

then

$$W = W(\tau_1, \sigma_1, x_1) \cup \cdots \cup W(\tau_k, \sigma_k, x_k)$$

is a discrete vector field for Δ.

Proof. By construction the second condition of Definition 16 is true. On the other hand, since $I(\tau_i, \sigma_i) \cap I(\tau_j, \sigma_j) = \emptyset$ for every simplex $\rho \in \Delta$ there exist at most one $v \in W$ such that $\rho \in v$. □

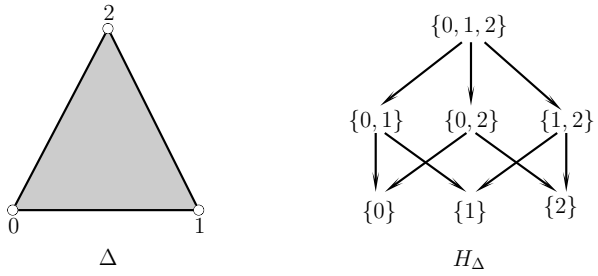


Figure 4.5: Hasse diagram.

Theorem 5 gives necessary and sufficient conditions of when a discrete vector field W is the gradient vector field of a Morse function f. In the following, we introduce some combinatorial tools to prove the Theorem 5. Given a simplicial complex Δ, the *Hasse Diagram* associated to Δ is the directed graph $H_\Delta = (V, E')$ whose vertex set V is the set of simplices and $(\sigma^{(k+1)}, \tau^{(k)}) \in E'$ if $\tau < \sigma$. For instance Figure 4.5 shows the Hasse diagram of Δ^2.

For a discrete vector field W of Δ we associated the directed graph $G_\Delta = (V, E)$ which is a modification of H_Δ. In the graph H_Δ, every arrow $(\sigma, \tau) \in E'$ is reversed if $(\tau, \sigma) \in W$, see Figure 4.6. Theorem 3 is the bridge that connects discrete vector fields and directed graphs.

Theorem 3. *Let W be a discrete vector field of simplicial complex Δ. If there are no closed path in W then the directed graph G_Δ has no cycles.*

Proof. Suppose there exists a directed cycle in G_Δ

$$c : \rho_0, \rho_1, \ldots, \rho_r$$

with $\rho_1 = \rho_r$. Let $\rho_i^{(k)}$ be a vertex of c. Note if $\dim(\rho_{i+2}) > \dim(\rho_i)$ then $(\rho_i^{(k)}, \rho_{i+1}^{(k+1)})$ and $(\rho_{i+1}^{(k+1)}, \rho_{i+2}^{(k+2)})$ are elements of W which cannot be possible since W is a discrete vector field. Hence $\dim(\rho_{i+2}) = \dim(\rho_i)$ and the cycle c is the form:

$$c : \tau_0^{(k-1)}, \sigma_0^{(k)}, \ldots, \tau_m^{(k-1)}, \sigma_m^{(k)}, \tau_{m+1}^{(k-1)}$$

Therefore c is a closed path in W. □

Theorem 4, which is a standard result in graph theory, implies the Theorem 5. In addition, the function f mentioned in the first theorem is a discrete Morse function of Δ with W its gradient vector field.

Theorem 4. *Let G be a finite directed graph. There exists a real-valued function f of the vertices of $G = (V, E)$ that is strictly decreasing along each directed path if only if G has no directed cycles.*

Proof. First implication is clear. Suppose G is finite and acyclic. Then every path starting in a vertex $v \in W$ is finite. Consider the function f given by:

$$f(v) = \max\{l(p_v) \ : \ p_v \text{ is a path starting in } v\}.$$

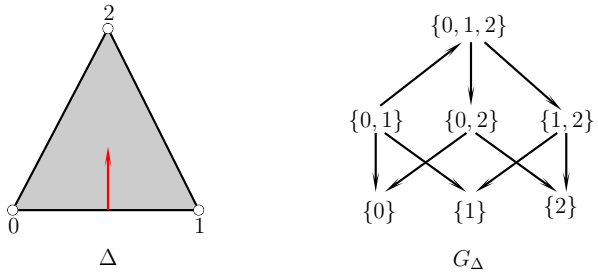


Figure 4.6: Modified Hasse diagram.

where $l(p_v)$ is the length of p_v. Hence $f(v) = 1$ if only if

$$\{u \in V \ : \ (v, u) \in E\} = \emptyset.$$

Therefore f is strictly decreasing along each directed path. □

Theorem 5. *A discrete vector field W of Δ is the gradient vector field of a Morse function if only if there are no closed paths in W.*

Proof. Consider the directed graph G_Δ and apply Theorem 4. □

Theorem 6. *Let W be a discrete vector field of Δ with no closed paths, and consider $\Delta' = \Delta \backslash \cup_{v \in W} v$. Then, Δ' is a subcomplex of Δ if only if there exists a Morse function F such that:*

1. *W is the gradient vector field of F.*
2. *$F^{-1}(a, b] = \bigcup\limits_{v \in W} v$ where $a = \max F(\Delta')$.*

Proof. Let G_Δ be the directed graph associated to Δ. Note that $G_{\Delta'}$ is a subgraph of G_Δ. Consider the Morse function on Δ constructed in the proof of Theorem 4 and let

$$a = \max f(\Delta')$$

Then we define a real-valued function on Δ as follows:

$$F(\sigma) = \begin{cases} f(\sigma) + a, & \text{if } \sigma \in L \\ f(\sigma), & \text{if } \sigma \in \Delta'. \end{cases}$$

where $L = \cup_{v \in W} v$. Now we shall prove that F is a Morse function on Δ. Consider a simplex σ of Δ then:

1. If $N_1(\sigma, f) = \emptyset$ then $\sigma \in \Delta'$. Thus $F(\tau) < F(\sigma) < F(\rho)$ for all $\tau^{(k-1)} < \sigma^{(k)} < \rho^{(k+1)}$.

2. If there exists $\tau \in N_1(\sigma, f)$ then $(\tau, \sigma) \in W$. Hence it is clear that $F(\tau) \geq F(\sigma)$.

Therefore $N_i(\sigma, F) = N_i(\sigma, f)$ for $0 \leq i \leq 2$ and F and f have the same gradient vector field W. This construction implies that $F^{-1}(a, b] = \cup_{v \in W} v$ where $b = \max F(\Delta)$.

For the other implication, consider a simplex $\tau^{(k-1)} < \sigma^{(k)}$ with σ a simplex of Δ'. Suppose that $\tau \notin \Delta'$ then

$$F(\tau) > a \geq F(\sigma)$$

hence $\sigma \notin \Delta'$. Therefore Δ' is a subcomplex of Δ. □

Corollary 6. *Let W be a discrete vector field of Δ with no closed paths, and consider $\Delta' = \Delta \setminus \cup_{v \in W} v$. If Δ' is a subcomplex of Δ then $\Delta \searrow \Delta'$.*

Example 2. Consider the simplicial complex and the discrete vector field W given by Figure 4.6. In this case, it is clear that Δ' is a subcomplex of Δ. In Figure 4.7 is shown the Morse functions f and F mentioned in the proof of Theorem 6.

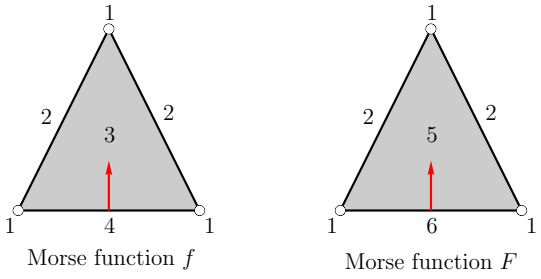


Figure 4.7: Morse function F of Δ.

4.3 Chapter Summary

Discrete Morse theory was developed by Forman (1995, 1998) as a combinatorial analog to Morse theory, developed by Morse (1929, 1930). The original theory analyzes the topology of a manifold by studying differentiable functions on that manifold, while discrete Morse theory provides similar methods of analysis for topological spaces endowed with additional, discrete structure. The main idea in this theory is to simplify a simplicial complex Δ reducing the number of simplices without modifying the homotopy type. This procedure consists in removing some simplices which are determined by a function $f : \Delta \to \mathbb{R}$. These simplices are called *regular* and they come in pairs (τ, σ) with (1) $\tau < \sigma$, (2) $\dim \sigma = \dim \tau + 1$ and $f(\tau) \geq f(\sigma)$. A simplex which is not regular is called *critical*. The sublevel complex Δ_c at the value c consists of all simplices with value less than c together with their faces. One of the fundamental results of this theory establishes that Δ_b can be collapsed to Δ_a if $f^{-1}(a, b]$ contains no critical simplices. Each Morse function f defines a set called the *gradient vector field* of f which consists of all vectors $v = (\tau, \sigma)$ where v is a regular pair. A particular property of the gradient vector field is that it does not contain closed k-path. A set of vectors $v = (\tau, \sigma)$ such that (1) $\tau < \sigma \in \Delta$, (2) $\dim \sigma = \dim \tau + 1$ and (3) every simplex $\rho \in \Delta$ belongs at most one vector is called a *discrete vector field*. Then every discrete vector field without closed k-paths defines a Morse function. It is clear that if W is a discrete vector field without closed paths and $\Delta' = \Delta \setminus \cup_{v \in W} v$ is a subcomplex of Δ then Δ can be collapsed to Δ'. This basic result will be used in the following chapter.

Chapter 5

Read/Write Distributed Model

In this chapter we use the discrete Morse theory in order to describe a gradient vector field G^k_{wr} for the simplicial complex $\mathsf{WR}^{(k)}(\Delta^n)$ which is the protocol complex associated to the read/write model with k rounds. First we build the gradient vector field for $k = 1$ then we generalize this construction to $k > 1$.

5.1 One round

Let σ be a simplex of L and $i_\sigma \in I^s_\sigma$ be a fix process. Consider

$$W_\rho = W(\rho^-, \rho^+, i_\rho) = W(\rho^-, \rho^+, x)$$

where $x = (i_\sigma, V^1_\sigma)$. This construction implies the following proposition.

Proposition 13. *Let σ be a simplex of L. Then W_σ does not contain non-trivial closed paths.*

Proposition 14. *For every $\sigma \in L$, $i_{\pi(\sigma)} = \pi(i_\sigma)$ if*

$$W_{\pi(\sigma)} = \pi(W_\sigma) = \{\pi(v) \ : \ v \in W_\sigma\}.$$

Proof. The claim is a consequence of Proposition 7 and definition of W_σ. □

Let us fix $\rho_1, \ldots, \rho_r$ representatives of equivalence classes of the relation given in Remark 1. Let

$$G_l = \bigcup_{j=1}^{r} \bigcup_{\pi \in S_{[n]}} W_{\pi(\rho_j)}.$$

Corollary 7. *G_l is a discrete vector field for WR_l.*

Proof. The claim is a consequence of Proposition 12. □

Theorem 7. *G_l is the gradient vector field of a Morse function.*

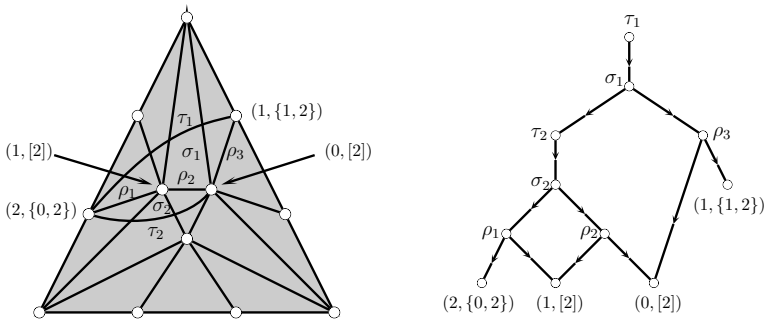
Proof. Consider the non-trivial closed k-path in G_l

$$c : \tau_0, \sigma_0, \tau_1, \sigma_1, \ldots, \tau_m, \sigma_m, \tau_{m+1}.$$

From the definition of G_l, $V^k_{\sigma_{i+1}} = V^k_{\tau_{i+1}} \subseteq V^k_{\sigma_i}$ for $k = 1, 2$. Then $\tau_0 = \tau_{m+1}$ implies $V^k_{\sigma_m} = \cdots = V^k_{\sigma_0}$. Therefore $I^s_{\sigma_m} = \cdots = I^s_{\sigma_0}$. Now suppose, for all $0 \leq i \leq m$, $\tau_i = \tau_i(\rho_i, s_i)$ and $\sigma_i = \tau_i(\rho_i, s_i)$. Then

$$\rho^-_{i+1} = \tau^-_{i+1} < \sigma^-_i = \rho^-_i$$

Hence $\rho^-_0 = \cdots = \rho^-_m$ and therefore $\rho_0 = \cdots = \rho_m = \rho$. In consequence, by Proposition 13 c is trivial. □



σ	τ_1	σ_1	τ_2	σ_2	ρ_1	ρ_2	ρ_3
$f(\sigma)$	6	5	4	3	2	2	2
$F(\sigma)$	9	8	7	6	5	5	5

Figure 5.1: Morse function on WR_0.

Corollary 8. *For every $0 \leq l \leq n$,*

1. *WR_l is collapsible to WR_{l+1}.*
2. *WR_l is $S_{[n]}$-collapsible to WR_{l+1}.*

Proof. Proposition 7 and Theorem 6 imply item 1. By construction, if $v = (\tau, \sigma) \in G_l$ and $\pi \in S_{[n]}$ then $\pi(v) = (\pi(\tau), \pi(\sigma)) \in G_l$. In addition $F(\pi(\tau)) = F(\tau)$ where F is the Morse function considered in the proof of Theorem 6. This implies that every simplex in

$$S_{[n]}(\tau) = \{\pi(\tau) \ : \ \pi \in S_{[n]}\}$$

can be collapsed in a parallel way. □

Corollary 9. *The set $G_{wr} = \bigcup_{l=0}^{n-1} G_l$ is a gradient vector field of a discrete Morse function for the complex $\mathsf{WR}(\Delta^n)$.*

Corollary 10. *For every natural number n, the simplicial complex $\mathsf{WR}(\Delta^n)$ is $S_{[n]}$-collapsible to $\chi(\Delta^n)$.*

Example 3. For $n = 2$, WR_0 contains 6 more maximal simplices than WR_1 and

$$L = I(\overline{\tau_1}) \cup I(\overline{\tau_2}),$$

where

$$\tau_1 = \begin{pmatrix} [2] & \{1,2\} & \{0,2\} \\ \emptyset & \{1\} & \{2\} \end{pmatrix} \text{ and } \tau_2 = \begin{pmatrix} [2] & \{0,2\} \\ \{0\} & \{2\} \end{pmatrix}.$$

Then there are 12 regular pairs in G_0, see Figure 3.8. In Figure 5.1, we present the regular pairs (τ_1, σ_1) and (τ_2, σ_2) and the value of its corresponding Morse functions f and F. According to the Morse function F, first we can collapse in a parallel way $S_{[n]}(\tau_1)$ and then $S_{[n]}(\tau_2)$. See Figure 5.2.

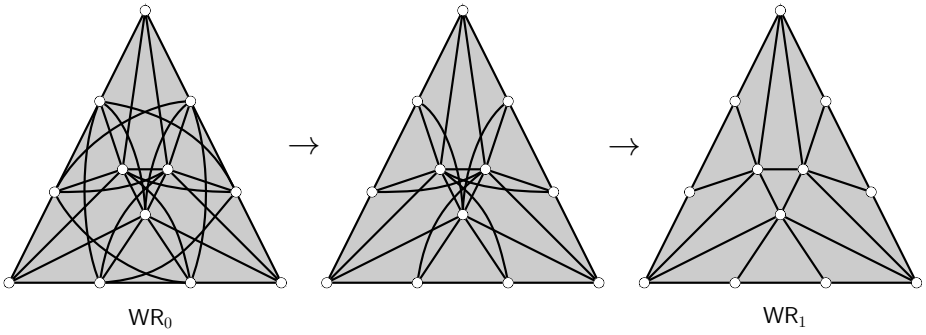


Figure 5.2: Collapsibility from WR_0 to WR_1.

5.2 Multiple rounds

Here we describe the gradient vector field G^k_{wr} with $k > 1$. In Figure 5.3 we present the complex $\mathsf{WR}^{(2)}(\Delta^2)$ in which only $\mathsf{WR}(\sigma_1)$ is depicted. Then the protocol complex of the iterated read/write model of k rounds is defined as:

$$\mathsf{WR}^{(k)}(\Delta^n) = \bigcup_{\sigma \in \mathsf{WR}^{(k-1)}(\Delta^n)} \mathsf{WR}(\sigma).$$

where WR is the carrier map defined in the Subsection 3.3. Notice that $\mathsf{WR}^{(k)}(\Delta^n)$ is well defined since

1. $\mathsf{WR}(\tau) \subseteq \mathsf{WR}(\sigma)$ if $\tau \subseteq \sigma$.
2. $\mathsf{WR}(\sigma_1) \cap \mathsf{WR}(\sigma_2) = \mathsf{WR}(\sigma_1 \cap \sigma_2)$.

For each simplex $\sigma \in \mathsf{WR}^{(k-1)}(\Delta^n)$ let $G_{wr}(\sigma)$ be the gradient vector field described in Subsection 5.1.

Proposition 15. *G_{wr} satisfies the following properties.*

1. *If $\tau \subseteq \sigma$ then $G_{wr}(\tau) \subseteq G_{wr}(\sigma)$.*
2. *$G_{wr}(\sigma_1) \cap G_{wr}(\sigma_2) = G_{wr}(\sigma_1 \cap \sigma_2)$.*

Now consider

$$G^k_{wr} = \bigcup_{\sigma \in \mathsf{WR}^{(k-1)}(\Delta^n)} G_{wr}(\sigma).$$

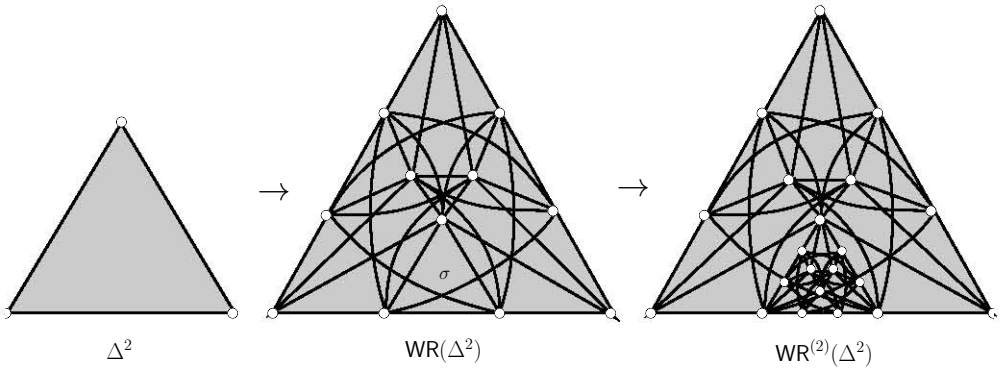


Figure 5.3: Complexes of the iterated model $\mathsf{WR}^{(2)}(\Delta^2)$.

Theorem 8. *G^k_{wr} is a gradient vector field for $\mathsf{WR}^{(k)}(\Delta^n)$.*

Proof. Let

$$\alpha_0, \beta_0, \ldots, \alpha_m, \beta_m, \alpha_{m+1}$$

be a non-trivial closed k-path c in G^k_{wr}. Suppose $(\alpha_i, \beta_i) \in G_{wr}(\sigma_i)$ where $\sigma_i = carrier(\beta_i) = carrier(\alpha_i)$ for all i, $0 \leq i \leq m$. Since $\alpha_{i+1} < \beta_i$, we have $\sigma_{i+1} < \sigma_i$. Then $\sigma = \sigma_0 = \cdots = \sigma_m$ and so c is a closed path in $G_{wr}(\sigma)$. Therefore Corollary 9 implies that c is trivial. □

Corollary 11. *For every natural number n, the protocol complex $\mathsf{WR}^{(k)}(\Delta^n)$ is collapsible to $\chi^{(k)}(\Delta^n)$.*

Proof. The proof is by induction on k. Corollary 9 proves the base case. On the other hand Theorem 8 proves that $\mathsf{WR}^{(k)}(\Delta^n)$ is collapsible to $\chi(\mathsf{WR}^{(k-1)}(\Delta^n))$. Therefore Corollary 16 and the induction hypothesis imply the claim. □

5.3 Chapter Summary

Studying the topology of the protocol complex $\mathsf{WR}(\Delta^n)$ where WR is the carrier map associated to read/write model of one round can be complicated in comparison with the protocol complexes $\chi(\Delta^n)$ and $View^n$. This problem is caused by its complex structure. However, using discrete Morse theory and the Algorithm 2.5 we can build a gradient vector field G_l of a Morse function of WR_l for each $0 \leq l \leq n-1$ such that every simplex σ of WR_{l+1} is critical. Therefore, by Theorem 2, WR_l is collapsible to WR_{l+1}. In addition, this procedure can be done in entire orbits of the natural symmetric group action. In consequence,

$$\mathsf{WR}(\Delta^n) = \mathsf{WR}_0 \searrow_{S_{[n]}} \mathsf{WR}_1 \searrow_{S_{[n]}} \cdots \searrow_{S_{[n]}} \mathsf{WR}_n = \chi(\Delta^n).$$

In the Chapter 6, we prove that chromatic subdivision χ is preserved by collapsibility. In other words, $\chi(\Delta)$ is collapsible to $\chi(\Delta')$ if Δ is collapsible to Δ'. Then we prove that $\mathsf{WR}^{(k)}(\Delta^n)$ is collapsible to $\chi(\mathsf{WR}^{(k-1)}(\Delta^n))$ describing a gradient vector field G^k_{wr} of a Morse function of $\mathsf{WR}^{(k)}(\Delta^n)$. Hence, using induction on k, $\mathsf{WR}^{(k)}(\Delta^n)$ is collapsible to $\chi^{(k)}(\Delta^n)$ which implies that they have the same homotopy type.

Chapter 6

Immediate Snapshot Distributed Model

Goubault et al. (2015) proved that simplicial complex $\chi^{(k)}(\Delta^n)$ associated to iterated immediate snapshot protocol of k rounds is collapsible. They studied the combinatorial structure present in the category of colored simplicial complexes. In this chapter we obtain the same result by using discrete Morse theory. Specifically, we describe a gradient vector field G^k_χ for the simplicial complex $\chi^{(k)}(\Delta^n)$. First we consider the case $k = 1$ and then we generalize to $k > 1$.

6.1 One round

In this case, the construction of discrete vector field G_χ is similar to G_{wr}, see Chapter 5. A simplex $\sigma \in \chi(\Delta^n) = \mathsf{WR}_n$ if $|V^1_\sigma| \leq 1$ or equivalently if $s = psnap(\sigma) = t$. Kozlov (2015) defines

$$\sigma^- = \begin{pmatrix} [n] & V_1 & \cdots & V_t \\ I_0 \cap V_1 & I_1 & \cdots & I_t \end{pmatrix}$$

and

$$\sigma^+ = \begin{pmatrix} [n] & V_1 & \cdots & V_t \\ I_0 \cup ([n] \setminus V_1) & I_1 & \cdots & I_t \end{pmatrix}$$

for all $\sigma \in \chi(\Delta^n)$. Notice if we define $psnap(\sigma) = -1$ then $V^1_\sigma = [n]$ and $V^2_\sigma = V_1$, hence $I^s_\sigma = [n] \setminus V_1$. Therefore our definitions of σ^- and σ^+ are equivalente to Kozlov's definitions. In (Kozlov, 2015) the following properties are proved.

Proposition 16. *Let n be an positive integer.*

1. $\sigma^- \subseteq \sigma \subseteq \sigma^+$.
2. *If $\sigma^- \subseteq \tau \subseteq \sigma^+$ then $\sigma^- = \tau^-$ and $\sigma^+ = \tau^+$.*
3. *For all $\pi \in S_{[n]}$, $\pi(\sigma)^- = \pi(\sigma^-)$ and $\pi(\sigma)^+ = \pi(\sigma^+)$.*

In addition, notice the equivalence relation on WR_l (see Remark 1), is preserved on $\chi(\Delta^n)$. That is: $\sigma \sim \sigma'$ if there exists a permutation $\pi \in S_{[n]}$ such that $\sigma = \pi(\sigma')$. Hence there exist simplices $\sigma_1, \ldots, \sigma_m$ such that

$$\chi(\Delta^n) = I(\overline{\sigma_1}) \cup \cdots \cup I(\overline{\sigma_m}).$$

Thus we can use the construction of Section 5.1 to get the following theorem.

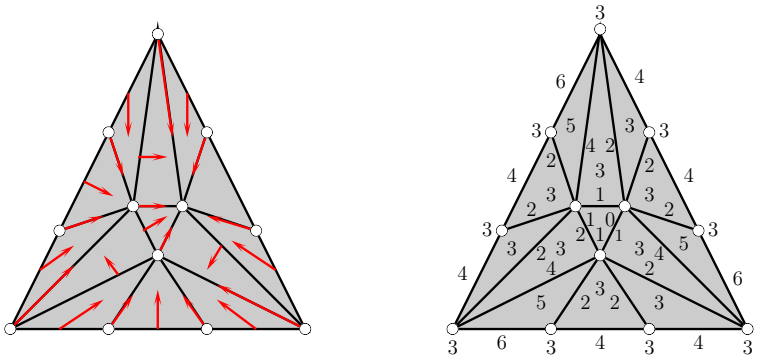
Theorem 9. *Let n be an positive integer. Then there exists a discrete vector field G_χ on $\chi(\Delta^n)$ with no non-trivial closed path such that*

$$(\pi(\tau), \pi(\tau)) \in G_\chi$$

for all $(\tau, \sigma) \in G_\chi$ $\pi \in S_{[n]}$.

Corollary 12. *For all positive integers $n \geq 1$, $\chi(\Delta)$ is $S_{[n]}$-collapsible.*

In Figure 6.1 we present a gradient vector field and its corresponding Morse function f for the complex $\chi(\Delta^2)$.



G_χ $\qquad$ $f(\chi(\Delta^2))$

Figure 6.1: Gradient vector field and Morse function on $\chi(\Delta^2)$.

6.2 Multiple rounds

Suppose G_χ^{k-1} is the gradient vector field for the complex $\chi^{(k-1)}(\Delta^n)$. In this section, following ideas from (Zhukova, 2016), we associate to each regular pair $v = (\tau^{(k-1)}, \sigma^{(k)}) \in G_\chi^{k-1}$ the set $G(v)$ which is formed by pairs (α, β) with $\dim \beta = \dim \alpha + 1$. For convenience we can suppose that $\tau = \Delta^{n-1}$ and $\sigma = \Delta^n$. Let

$$\Lambda^n = \Delta_0^n \cup \cdots \cup \Delta_i^{n-1}$$

where $\Delta_i^n = \Delta^{[n]\setminus\{i\}}$ and

$$\gamma = \begin{pmatrix} [n] & V_1 & \cdots & V_t \\ I_0 & I_1 & \cdots & I_t \end{pmatrix}$$

be a simplex of $\chi(\sigma)$ with $1 \leq |I_i|$, $1 \leq i \leq t$. Suppose $k = \max\{j : n \in V_j\}$. Consider the following cases:

1. If $n \in I_k$ and $|V_k \setminus V_{k+1}| = 1$. Let

$$\gamma^- = \begin{pmatrix} [n] & \cdots & V_{k-1} & V_{k+1} & \cdots & V_t \\ I_0 & \cdots & I_{k-1} & I_{k+1} & \cdots & I_t \end{pmatrix}$$

and $\gamma^+ = \gamma$. In the case $k = 0$,

$$\gamma^- = \begin{pmatrix} [n] & V_1 & \cdots & V_t \\ \emptyset & I_1 & \cdots & I_t \end{pmatrix}$$

2. If $n \in I_k$ and $|V_k \setminus V_{k+1}| \geq 2$. Let

$$\gamma^- = \begin{pmatrix} [n] & \cdots & V_k & \cdots & V_t \\ I_0 & \cdots & \{n\} & \cdots & I_t \end{pmatrix} \text{ and } \gamma^+ = \begin{pmatrix} [n] & \cdots & V_k & \cdots & V_t \\ I_0 & \cdots & I'_k & \cdots & I_t \end{pmatrix}.$$

where $I'_k = V_k \setminus V_{k+1}$.

3. If $n \notin I_k$ and $|V_k \setminus V_{k+1}| \geq 2$. Let $\gamma^- = \gamma$ and

$$\gamma^+ = \begin{pmatrix} [n] & \cdots & V_k & V_{k+1} \cup \{n\} & V_{k+1} & \cdots & V_t \\ I_0 & \cdots & I_k & \{n\} & I_{k+1} & \cdots & I_t \end{pmatrix}.$$

The following results are a consequence of the previous construction. In addition they are the analogous versions of Proposition 10 and Corollary 4.

Proposition 17. *Simplices γ^- and γ^+ satisfy the following properties.*

1. $\gamma^- \subseteq \gamma \subseteq \gamma^+$.
2. *If $\gamma^- \subseteq \rho \subseteq \gamma^+$ then $\rho^- = \gamma^-$ and $\rho^+ = \gamma^+$.*

Corollary 13. *Let γ, ρ be simplices of $\chi(\sigma)$.*

1. *$I(\gamma) \cap I(\rho) = \emptyset$ or $I(\gamma) = I(\rho)$.*
2. *$I(\gamma) \cap \chi(\Lambda^n) = \emptyset$ if $\gamma \notin \chi(\Lambda^n)$.*

Proof. Item 1 follows from the construction of γ^- and γ^+. To prove 2 notice that

$$\gamma = \begin{pmatrix} [n] & V_1 & \cdots & V_t \\ I_0 & I_1 & \cdots & I_t \end{pmatrix}$$

is not a simplex of $\chi(\Lambda^n)$ if only if $|I_0| \geq 1$ or $V_1 = [n-1]$. Then by definition, if γ satisfies the previous conditions then γ^- and γ^+ also satisfy them. □

Now consider

$$G(\tau, \sigma) = \bigcup_{\gamma \notin \chi(\Lambda^n)} W(\gamma^-, \gamma^+, x)$$

Proposition 18. *For each $(\tau, \sigma) \in G_\chi^{k-1}$, $G(\tau, \sigma)$ is a gradient vector field for $\chi(\sigma)$.*

Proof. By Proposition 12 and Corollary 13, $G(\tau, \sigma)$ is a discrete vector field for $\chi(\sigma)$. Let

$$\alpha_0, \beta_0, \ldots, \alpha_m, \beta_m$$

be a non-trivial path and suppose $\alpha_i = \alpha(\gamma_i^-, \gamma_i^+, p_i)$ and $\beta_i = \beta(\gamma_i^-, \gamma_i^+, p_i)$. Notice that by construction each vector $v_i = (\alpha_i, \beta_i)$ can be classified as follows:

1. v_i is the type 1 if γ_i satisfies the conditions of the case 1 or 3.
2. v_i is the type 2 if γ_i satisfies the conditions of the case 2.

Suppose

$$\beta_i = \begin{pmatrix} [n] & \cdots & V_k & \cdots & V_t \\ I_0 & \cdots & I_k & \cdots & I_t \end{pmatrix}$$

Hence if β_i is the type 1 then $|I_{k+1}| \geq 2$ if only if $\alpha_{i+1} = \alpha_i$. Therefore $|I_{k+1}| = 1$ and so β_{i+1} is the type 2. Now if β_i is the type 2 then $\{n, x_i\} \subseteq I_k \subseteq V_k \setminus V_{k+1}$. In the case $I_k = V_k \setminus V_{k+1}$ we have

$$\alpha_{i+1} = \begin{pmatrix} [n] & \cdots & V_k & \cdots & V_t \\ I_0 & \cdots & I_k \setminus \{n\} & \cdots & I_t \end{pmatrix}$$

and so β_{i+1} is the type 1. On the other case we can get α_{i+1} if only if we remove a process of $I_0 \cup \cdots \cup I_{k-1} \cup I_{k+1} \cup \cdots \cup I_t$ or remove n. In the first case β_{i+1} is the type 2 with

$$\beta_{i+1} = \begin{pmatrix} [n] & \cdots & V_k & \cdots & V_t \\ I'_0 & \cdots & I'_k & \cdots & I'_t \end{pmatrix}$$

such that $I_k \subset I'_k$. Therefore, in every step of the path the entry n travels to the right side of each simplex during the path. Hence, there are no closed paths inside of $G(\tau, \sigma)$. □

Corollary 14. *For all natural numbers n, $\chi(\Delta^n)$ collapses to $\chi(\Lambda^n)$.*

In Figure 6.2 we illustrate the gradient vector field described in the proof of Proposition 18 for the case $n = 2$. On the other hand, according to the proof of Proposition 18,

$$G'_\chi = \bigcup_{\gamma \in \chi(\Delta^n)} W(\gamma^-, \gamma^+, x)$$

is a gradient vector field, different from G_χ, which shows that $\chi(\Delta^n)$ is collapsible; see Subsection 6.1.

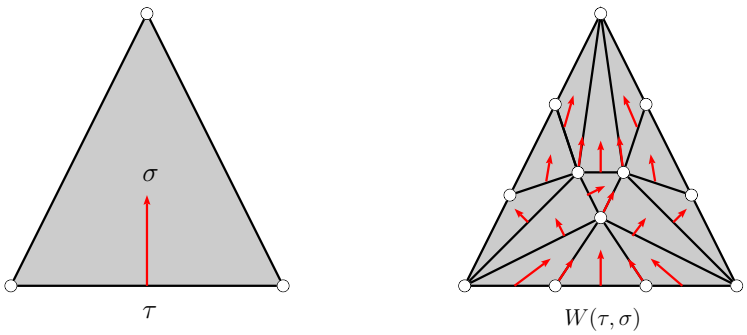


Figure 6.2: $\chi(\Delta^2)$ collapses to $\chi(\Lambda^2)$.

Let (α, β) be a regular pair of $G(\tau, \sigma)$. Then, β is a *border simplex* if there exists a face $\alpha' < \beta$ such that $\dim \beta = \dim \alpha' + 1$ and $\alpha' \in \partial(\chi(\sigma))$.

Notice that if $(\alpha, \beta) \in G(\tau, \beta)$ then

$$\beta = \begin{pmatrix} [n] & V_1 & \cdots & V_t \\ I_0 & I_1 & \cdots & I_t \end{pmatrix}$$

with $|I_i| \geq 1$ for all i, $0 \leq i \leq t$. On the other hand if β is a border simplex then it satisfies one of the following conditions:

1. $I_0 = \{n\}$ and $V_1 = [n-1]$.
2. $|I_0| = 1$ and $n \in I_1$.

In the first case $\alpha' = \alpha$, while in the second case $\alpha' \neq \alpha$. Consider

$$G_\chi^k = \bigcup_{v \in G_\chi^{k-1}} G(v)$$

Theorem 10. *G_χ^k is a gradient vector field for $\chi^{(k)}(\Delta^n)$.*

Proof. Let ρ be a simplex of $\chi^{(k)}(\Delta^n)$ and suppose there exist $v_1, v_2 \in G_\chi^k$ such that $\rho \in v_1 \cap v_2$ with $v_i \in G(\tau_i, \sigma_i)$ for $i = 1, 2$. Then

$$\rho \in \chi(\sigma_1) \cap \chi(\sigma_2) = \chi(\sigma_1 \cap \sigma_2).$$

Therefore Corollary 13 implies that

$$\rho \in \chi(\tau_1) \cap \chi(\tau_2) = \chi(\tau_1 \cap \tau_2).$$

and thus $\tau_1 = \tau_2$, which is a contradiction. In consequence G_χ^k is a discrete vector field. Let

$$\alpha_0, \beta_0, \ldots, \alpha_m, \beta_m, \alpha_{m+1}$$

be a non-trivial closed k-path in G_χ^k. Then there exists a partition

$$[m] = [a_0, b_0] \cup \cdots \cup [a_l, b_l]$$

with $a_j < b_j$ and $a_j = b_{j-1} + 1$ such that

$$(\alpha_{a_j}, \beta_{a_j}), \ldots, (\alpha_{b_j}, \beta_{b_j}) \in G(\tau_j, \sigma_j)$$

for all j, $0 \leq j \leq l$. Thus we have a closed sequence c of regular pairs

$$\tau_0, \sigma_0, \ldots, \tau_l, \sigma_l, \tau_{l+1}$$

in G_χ^{k-1}. Note that every β_{b_j} is a border simplex in $G(\tau_j, \sigma_j)$. Hence

$$\beta_{b_j} = \begin{pmatrix} [n] & V_1 & \cdots & V_t \\ I_0 & I_1 & \cdots & I_t \end{pmatrix}$$

with $|I_0| = 1$, $n \in I_1$ and

$$\alpha_{a_{j+1}} = \begin{pmatrix} [n] & V_1 & \cdots & V_t \\ \emptyset & I_1 & \cdots & I_t \end{pmatrix}$$

This implies that $\dim(\tau_{j+1}) = |V_1| - 1 = \dim(\sigma_{j+1}) - 1$ and therefore

$$n = \dim(\sigma_j) \geq \dim(\sigma_{j+1}).$$

In consequence,

$$\dim(\sigma_0) \geq \dim(\sigma_1) \geq \cdots \geq \dim(\sigma_m) \geq \dim(\sigma_0)$$

and so c is a closed path in G_χ^{k-1}. □

In fact, notice that the previous construction implies the following corollary. See Figure 6.3.

Corollary 15. *If G' is a gradient vector field for a simplicial complex Δ then*

$$G'_\chi = \bigcup_{(\tau,\sigma)\in G'} G(\tau,\sigma)$$

is a gradient vector field for $\chi(\Delta)$.

Corollary 16. *If Δ is collapsible to Δ' then $\chi(\Delta)$ is collapsible to $\chi(\Delta')$*

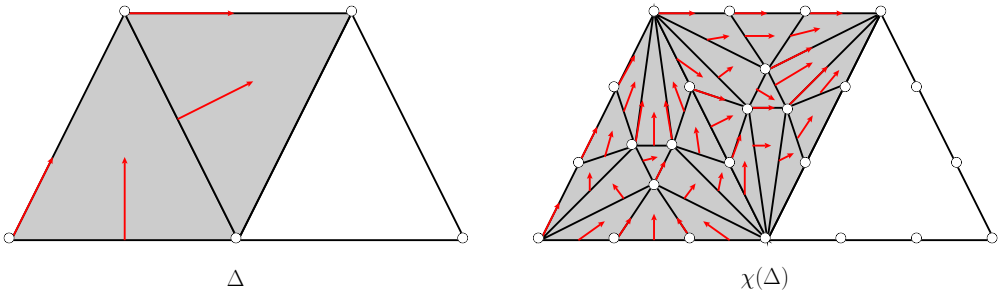


Figure 6.3: Gradient vector field for $\chi(\Delta)$.

6.3 Chapter Summary

The structure of the protocol complex $\chi^{(k)}(\Delta^n)$ has a nice combinatorial representation, which permits to study its connectivity. In the case $k = 1$ we proved that $\chi(\Delta^n)$ is collapsible by using discrete Morse theory. The construction of the gradient vector field G_χ is similar to G_{wr}. Since χ is a subdivision map we can define inductively a gradient vector field $G_\chi^{(k)}$ of protocol complex $\chi^{(k)}(\Delta^n)$. In other words, for each vector $v = (\tau,\sigma) \in G_\chi^{(k-1)}$ we associated a gradient vector field $G(\tau,\sigma)$ of $\chi(\sigma)$ such that every simplex $\sigma \in \chi(\Lambda)$ is critical where $\Lambda = \sigma \setminus \tau$. Hence $\chi(\sigma)$ is collapsible to $\chi(\Lambda)$. In consequence, we proved that $G_\chi^{(k)} = \cup_{v\in G_\chi^{(k-1)}} G(v)$ is a gradient vector field of $\chi^{(k)}(\Delta^n)$. In general, the construction of $G(\tau,\sigma)$ permit us to prove that $\chi(\Delta)$ is collapsible to $\chi(\Delta')$ if Δ is collapsible to Δ'.

Chapter 7

Conclusions and Future Work

Conclusions

Since the discovery of the intimate relation between topology and fault-tolerant distributed computing in 1993, various approaches combining algorithmic, topological, and combinatorial techniques have been used, in order to understand the structure of protocol complexes associated to different distributed computing models. The structure of a protocol complex in turns determines the decision tasks solvable in the model, and the time complexity of the solvable cases. In this paper we have used, for the first time, discrete Morse theory, in order to study the connectivity of the protocol complex $\mathsf{WR}^{(k)}(\Delta^n)$ associated to the iterated wait-free read/write model. In fact, we have proved that $\mathsf{WR}^{(k)}(\Delta^n)$ is collapsible, and that the Algorithm 2.5 performs the collapses. One of the keys in the collapsibility procedure is that the simplicial complex $\mathsf{WR}^{(k)}(\Delta^n)$ is defined recursively. The Algorithm 2.5 collapses WR_l to WR_{l+1} where $0 \leq l \leq n-1$ and WR_l is the simplicial complex associated to shared memory mem_l in Algorithm 2.5. Moreover, since readings operations of process i are executed to the memory mem_l in any order, the collapsibility procedure can be performed by entire orbits of the symmetric group. It is the first time read/write operations are considered directly, in previous studies about the structure of protocol complexes only atomic snapshots were considered, as a means for reading the shared memory. In addition, we used discrete Morse theory to give new combinatorial proofs about of collapsibility of the iterated immediate snapshot complex $\chi^{(k)}(\Delta^n)$. In general, we have proved that chromatic subdivision is a construction which preserves collapsibility. In other words, if Δ is collapsible to Δ' then $\chi(\Delta)$ is collapsible to $\chi(\Delta')$.

The study of protocol complexes can be complicated due to its structure, however in the basic method of proving impossibility results we only need to show that the protocol complex has a certain topological property. In the asynchronous computability theorem, Herlihy and Shavit (1999) characterized the tasks that are solvable wait-free by $n+1$ processes, communicating by writing and taking snapshots of a shared memory. At the core of the necessity part of the theorem, is shown that the protocol complex in n-connected (Lemma 4.12), which is implied by our collapsibility result, for the more complicated read/write protocol complex. The necessity part of the asynchronous computability theorem is what is used to prove impossibility results. For instance, it is known that if $T = (\mathcal{I}, \mathcal{O}, \mathcal{S})$ is an $(n+1)$-process k-set agreement task and $\mathcal{P}$ is a protocol complex such that $\mathcal{P}(\sigma)$ is $(k-1)$-connected for all $\sigma \in \mathcal{I}$ then $\mathcal{P}$ cannot solve k-set agreement task T. In consequence, since $\mathsf{WR}^{(k)}(\Delta^n)$ is n-connected the set agreement task is not solvable in the iterated wait-free read/write model of k rounds.

Future Work

As we have mentioned, different topology and combinatorial tools have been used in order to study the topology of simplicial complexes associated to different models. In our case, we found discrete Morse theory very natural to use in the distributed computing setting, and hope that it can be applied to analyze other full information distributed computing protocols. For instance, in the study of non-iterated protocols like non-iterated immediate snapshot model (Attiya and Rajsbaum, 2002). Studying the topology of non-iterated models can be complicated since, as we mentioned in the Chapter 1, describing the local view of a process in a non-iterated model is more difficult than in the iterated case. Moreover, the protocol complexes associated to non-iterated immediate snapshot model of k and $k+1$ rounds are not related, which does not happen in the iterated case. To be exact, the protocol complex associated to non-iterated immediate snapshot model of k rounds does not have a recursive definition like in the iterated case. This increases the difficulty level in its topological study. On the other hand, discrete Morse theory can be applied in the topological study of partial snapshots objects which are a generalization of atomic snapshot (Attiya et al., 2008) or in the study of other synchronization primitives like test&set, compare&swap, etc.

Finally, to the best of our knowledge there are only two algorithms that implement immediate snapshot, the original of Borowsky and Gafni (1993b) and its recursive version of Gafni and Rajsbaum (2010). Then a natural research direction which rises from the collapsibility procedure and the analysis of Algorithm 2.5 is to study the relation between collapsibility procedure and step complexity of immediate snapshot object in order to give lower bounds.

Bibliography

Yehuda Afek, Hagit Attiya, Danny Dolev, Eli Gafni, Michael Merritt, and Nir Shavit. Atomic snapshots of shared memory. *J. ACM*, 40(4):873–890, Sep. 1993. ISSN 0004-5411.

Yehuda Afek, Iftah Gamzu, Irit Levy, Michael Merritt, and Gadi Taubenfeld. *Group Renaming*, pages 58–72. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-92221-6.

Hagit Attiya and Sergio Rajsbaum. The combinatorial structure of wait-free solvable tasks. *SIAM J. Comput.*, 31(4):1286–1313, April 2002. ISSN 0097-5397.

Hagit Attiya and Jennifer Welch. *Distributed Computing: Fundamentals, Simulations, and Advanced Topics*. Woley, 2 edition, 2004. ISBN 978-0-471-45324-6.

Hagit Attiya, Amotz Bar-Noy, Danny Dolev, David Peleg, and Rüdiger Reischuk. Renaming in an asynchronous environment. *J. ACM*, 37(3):524–548, jul 1990. ISSN 0004-5411.

Hagit Attiya, Amotz Bar-Noy, and Danny Dolev. Sharing memory robustly in message-passing systems. *J. ACM*, 42(1):124–142, January 1995a. ISSN 0004-5411.

Hagit Attiya, Maurice Herlihy, and Ophir Rachman. Atomic snapshots using lattice agreement. *Distributed Computing*, 8(3):121–132, 1995b. ISSN 0178-2770.

Hagit Attiya, Rachid Guerraoui, and Eric Ruppert. Partial snapshot objects. In *Proceedings of the Twentieth Annual Symposium on Parallelism in Algorithms and Architectures*, SPAA '08, pages 336–343, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-973-9.

Ofer Biran, Shlomo Moran, and Shmuel Zaks. A combinatorial characterization of the distributed 1-solvable tasks. *J. Algorithms*, 11(3):420–440, 1990.

Cvetkovska Biserka. Predicting weather phenomena using discrete morse theory. Master's thesis, University of Ljubljana, Ljubljana, 2015.

Elizabeth Borowsky and Eli Gafni. Generalized flp impossibility result for t-resilient asynchronous computations. In *Proc. 25th Annual ACM Symp. on Theory of Computing*, STOC, pages 91–100, New York, NY, USA, 1993a. ACM. ISBN 0-89791-591-7.

Elizabeth Borowsky and Eli Gafni. Immediate atomic snapshots and fast renaming. In *Proc. 12th ACM Symp. on Principles of Distributed Computing*, PODC, pages 41–51, New York, NY, USA, 1993b. ACM. ISBN 0-89791-613-1.

Elizabeth Borowsky and Eli Gafni. A simple algorithmically reasoned characterization of wait-free computation (extended abstract). In *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '97, pages 189–198, New York, NY, USA, 1997. ACM. ISBN 0-89791-952-1.

Elizabeth Borowsky, Eli Gafni, Nancy Lynch, and Sergio Rajsbaum. The BG distributed simulation algorithm. *Distributed Computing*, 14(3):127–146, 2001.

Armando Castañeda and Sergio Rajsbaum. New combinatorial topology bounds for renaming: the lower bound. *Distributed Computing*, 22(5-6):287–301, 2010. ISSN 0178-2770.

Armando Castañeda and Sergio Rajsbaum. New combinatorial topology bounds for renaming: The upper bound. *J. ACM*, 59(1):3:1–3:49, March 2012. ISSN 0004-5411.

Soma Chaudhuri. More choices allow more faults: set consensus problems in totally asynchronous systems. *Information and Computation*, 105(1):132–158, 1993.

Rodolfo Conde and Sergio Rajsbaum. An introduction to the topological theory of distributed computing with safe-consensus. *Electronic Notes in Theoretical Computer Science*, 283:29 – 51, 2012. ISSN 1571-0661.

Leonard Eugene Dickson. *History of the theory of numbers*. Dover books on mathematics. Dover Publ., Mineola, NY, 2005.

M. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed commit with one faulty process. *Journal of the ACM*, 32(2), apr 1985.

Robin Forman. A discrete morse theory for cell complexes. In *in Geometry, Topology 6 Physics for Raoul Bott*. International Press, 1995.

Robin Forman. Morse theory for cell complexes. *Advances in Mathematics*, 134(1):90 – 145, 1998. ISSN 0001-8708.

Robin Forman. A user's guide to discrete morse theory. In *Proc. of the 2001 Internat. Conf. on Formal Power Series and Algebraic Combinatorics, A special volume of Advances in Applied Mathematics*, page 48, 2001.

Eli Gafni. *Group-Solvability*, pages 30–40. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-30186-8.

Eli Gafni and Sergio Rajsbaum. Recursion in distributed computing. In Shlomi Dolev, Jorge Cobb, Michael Fischer, and Moti Yung, editors, *Stabilization, Safety, and Security of Distributed Systems*, volume 6366 of *Lecture Notes in Computer Science*, pages 362–376. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-16022-6.

Eli Gafni, Sergio Rajsbaum, and Maurice Herlihy. *Subconsensus Tasks: Renaming Is Weaker Than Set Agreement*, pages 329–338. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-44627-9.

Eric Goubault, Samuel Mimram, and Christine Tasson. Iterated chromatic subdivisions are collapsible. *Applied Categorical Structures*, 23(6):777–818, 2015.

John Havlicek. Computable obstructions to wait-free computability. *Distributed Computing*, 13(2):59–83, 2000. ISSN 0178-2770.

John Havlicek. A note on the homotopy type of wait-free atomic snapshot protocol complexes. *SIAM J. Comput.*, 33(5):1215–1222, 2004.

Maurice Herlihy and Sergio Rajsbaum. Algebraic spans. *Mathematical. Structures in Comp. Sci.*, 10(4):549–573, August 2000. ISSN 0960-1295.

Maurice Herlihy and Sergio Rajsbaum. Simulations and reductions for colorless tasks. In *Proceedings of the 2012 ACM symposium on Principles of distributed computing*, PODC '12, pages 253–260, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1450-3.

Maurice Herlihy and Sergio Rajsbaum. The topology of distributed adversaries. *Distributed Computing*, 26(3):173–192, 2013. ISSN 0178-2770.

Maurice Herlihy and Nir Shavit. The asynchronous computability theorem for t-resilient tasks. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 111–120, New York, NY, USA, 1993. ACM. ISBN 0-89791-591-7.

Maurice Herlihy and Nir Shavit. The topological structure of asynchronous computability. *J. ACM*, 46(6):858–923, November 1999. ISSN 0004-5411.

Maurice Herlihy, Sergio Rajsbaum, and Michel Raynal. Computability in distributed computing: A tutorial. *SIGACT News*, 43(3):88–110, August 2012. ISSN 0163-5700.

Maurice Herlihy, Dmitry Kozlov, and Sergio Rajsbaum. *Distributed Computing Through Combinatorial Topology*. Elsevier, Imprint Morgan Kaufmann, 2013a. ISBN 978-0-12-404578-1.

Maurice Herlihy, Sergio Rajsbaum, and Michel Raynal. Power and limits of distributed computing shared memory models. *Theoretical Computer Science*, 509:3 – 24, 2013b. ISSN 0304-3975.

Gunnar Hoest and Nir Shavit. Towards a topological characterization of asynchronous complexity. In *Proc. 16th ACM Symp. Principles of distributed computing*, PODC, pages 199–208, New York, NY, USA, 1997. ACM. ISBN 0-89791-952-1.

Damien Imbs, Sergio Rajsbaum, and Michel Raynal. *The Universe of Symmetry Breaking Tasks*. Springer Berlin Heidelberg, 2011.

Jakob Jonsson. *Simplicial Complexes of Graphs*. Lecture Notes in Mathematics. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

Dimitry Kozlov. *Combinatorial Algebraic Topology*. Springer, Berlin Heidelberg, 1 edition, 2008. ISBN 978-3-540-73051-4.

Dmitry Kozlov. Chromatic subdivision of a simplicial complex. *Homology Homotopy Appl.*, 14(2):197–209, 2012.

Dmitry Kozlov. Topology of the immediate snapshot complexes. *Topology Appl.*, 178: 160–184, 2014. ISSN 0166-8641.

Dmitry Kozlov. Topology of the view complex. *Homology Homotopy Appl.*, 17(1):307–319, 2015. ISSN 1532-0073.

Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982. ISSN 0164-0925.

M. C. Loui and H. H. Abu-Amara. *Memory requirements for agreement among unreliable asynchronous processes*, volume 4, pages 163–183. JAI press, 1987.

Mark Moir. Fast, long-lived renaming improved and simplified. *Science of Computer Programming*, 30(3):287 – 308, 1998. ISSN 0167-6423.

Mark Moir and James H. Anderson. Wait-free algorithms for fast, long-lived renaming. *Science of Computer Programming*, 25(1):1 – 39, 1995. ISSN 0167-6423.

Marston Morse. The foundations of the calculus of variations in the large in m-space (first paper). *Transactions of the American Mathematical Society*, 31(3):379–404, 1929.

Marston Morse. The foundations of a theory of the calculus of variations in the large in m-space (second paper). *Transactions of the American Mathematical Society*, 32(4): 599–631, 1930.

Sergio Rajsbaum. Iterated shared memory models. In Alejandro López-Ortiz, editor, *LATIN 2010: Theoretical Informatics: 9th Latin American Symposium, Oaxaca, Mexico, April 19-23, 2010. Proceedings*, pages 407–416. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

Sergio Rajsbaum, Michel Raynal, and Corentin Travers. The iterated restricted immediate snapshot model. In *COCOON*, volume 5092 of *Lecture Notes in Computer Science*, pages 487–497. Springer, 2008. ISBN 978-3-540-69732-9.

Michael Saks and Fotios Zaharoglou. Wait-free k-set agreement is impossible: the topology of public knowledge. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC '93, pages 101–110, New York, NY, USA, 1993. ACM. ISBN 0-89791-591-7.

Michael Saks and Fotios Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. *SIAM J. Comput.*, 29(5):1449–1483, 2000.

E.C. Zeeman. On the dunce hat. *Topology*, 2(4):341 – 358, 1963. ISSN 0040-9383.

A. M Zhukova. Discrete morse theory for the barycentric subdivision. *ArXiv e-prints*, May 2016.