Universidad Nacional Autónoma de México

Facultad de Ciencias

Estudio de la topología de los algoritmos aleatorios y distribuidos.

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

Matemático

PRESENTA:

Rolando Corona Jiménez

TUTOR

Dr. Armando Castañeda Rojano

Ciudad Universitaria, Ciudad de México, 2018.

1. **Datos del alumno**

   Corona

   Jiménez

   Rolando

   5529514008

   Universidad Nacional Autónoma de México

   Facultad de Ciencias

   Matemáticas

   411004244

2. **Datos del tutor**

   Dr.

   Armando

   Castañeda

   Rojano

3. **Datos del sinodal 1**

   Dr.

   Sergio

   Rajsbaum

   Gorodezky

4. **Datos del sinodal 2**

   Dr.

   Javier

   Bracho

   Carpizo

5. **Datos del sinodal 3**

   Dr.

   José David

   Flores

   Peñaloza

6. **Datos del sinodal 4**

   Dra.

   Natalia

   Jonard

   Pérez

7. **Datos del trabajo escrito**

   Estudio de la topología de los algoritmos aleatorios y distribuidos

   80 p.

   2018

Agradecimientos

A mis padres y mi familia, por su apoyo incondicional.

A Armando, mi tutor, por su paciencia y sus consejos siempre oportunos.

A Karina, mi tutora en el SBEI, por escucharme y ofrecerme su apoyo a lo largo de mi trayectoria escolar.

A mis amigos y compañeros de la licenciatura.

A mis profesores, por su entusiasmo y compromiso con la promoción del conocimiento.

Al Instituto de Matemáticas de la UNAM, por brindarme un espacio de trabajo para el desarrollo de esta tesis durante el tiempo en el que fui becario del mismo.

A la Facultad de Ciencias y la UNAM, por ser espacios propicios para el desarrollo integral de sus estudiantes.

Al Programa Universitario de Estudios de la Diversidad Cultural Interculturalidad (PUIC) de la UNAM, que a través de su Sistema de becas para estudiantes indígenas (SBEI) me otorgó una beca durante los ocho semestres que dura el plan de estudios de la Licenciatura en Matemáticas, además de dos semestres adicionales en los que trabajé en el desarrollo de esta tesis.

Investigación realizada gracias al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) de la UNAM IA102417. Agradezco a la DGAPA-UNAM la beca recibida.

Índice general

Capítulo 1

Introducción

Un sistema distribuido es un conjunto de dispositivos de cómputo que se comunican entre sí para resolver un problema. Esta definición abarca un amplio rango de sistemas de cómputo que va desde microprocesadores cuyos núcleos se comunican a través de una memoria compartida hasta redes que cubren grandes áreas geográficas como el Internet. El cómputo distribuido tiene muchas aplicaciones y está presente siempre que se involucren procedimientos en los que se tenga que compartir recursos o información, como compartir una impresora en una red local o al hacer una transferencia bancaria; sin embargo, coordinar a distintas entidades de cómputo no es una tarea fácil, existen muchos factores que pueden afectar el funcionamiento de un algoritmo distribuido y fundamentalmente estos factores tienen que ver con algunos atributos que dan como resultado distintos modelos de cómputo, entre dichos factores se encuentran la *interfaz de comunicación*, el *modelo de sincronización* y la *tolerancia a fallos* del sistema. Los procesos tienen que usar un medio de comunicación, dos de los métodos más comunes son el *paso de mensajes* y el uso de una *memoria compartida*. El modelo de sincronización se refiere a las suposiciones que se hacen acerca de cómo se suscitan los eventos a lo largo del tiempo, en un extremo está el modelo *sincrónico* o *simultáneo* en el que todos los procesos ejecutan las operaciones al mismo tiempo, mientras que en el otro extremo los procesos pueden ser *asincrónicos*, de modo que los procesos ejecutan operaciones a distintas velocidades en un orden arbitrario, en medio de ellos está el modelo *semisincrónico*, en el que los procesos poseen cierta información acerca de la sincronización de los eventos. Con respecto a la tolerancia a fallos, se puede

pedir que el hardware sobre el que los algoritmos se ejecutan sea completamente confiable, o bien que el algoritmo pueda terminar satisfactoriamente aún cuando algunos procesos muestren un comportamiento defectuoso ya sea porque simplemente se detengan o incluso si muestran un comportamiento inesperado, el comportamiento defectuoso también incluye fallas en el proceso de comunicación debido a perdidas de mensajes o duplicados de los mismos. No todos los modelos de cómputo pueden resolver la misma clase de problemas, por eso es importante tener en cuenta qué suposiciones se hacen al estudiar un problema de cómputo distribuido.

En las últimas décadas se han desarrollado técnicas que permiten estudiar una clase de problemas distribuidos llamados *tareas de decisión* con la ayuda de la topología combinatoria, en una tarea de decisión los procesos participantes inician con un valor de entrada y al terminar de ejecutar el algoritmo distribuido, cada proceso decide un valor de salida. Con el lenguaje de la topología combinatoria se pueden representar nociones como las de *tarea*, *protocolo* y *solubilidad*; esta representación es independiente del modelo de cómputo que se suponga y establece una relación entre algoritmos distribuidos y complejos simpliciales, a través de la cual se puede averiguar si una tarea es soluble en cierto modelo de cómputo estudiando las propiedades topológicas del complejo simplicial asociado al protocolo de comunicación usado en el modelo de cómputo en cuestión, por ejemplo, para el modelo de cómputo asincrónico basado en *instantáneas* en una memoria compartida, el *Teorema de computabilidad asincrónica* caracteriza, en términos topológicos, a las tareas que se pueden resolver en dicho modelo [7], y es una muestra de lo útil que resulta la topología en el estudio de problemas distribuidos, pues se puede echar mano de muchos resultados que se han desarollado en el área de topología algebraica y combinatoria.

Esta manera de estudiar problemas distribuidos permite estudiar de forma global a todas las posibles *configuraciones* del sistema que se pueden obtener después de que cada proceso ejecuta el protocolo de comunicación, una configuración representa el estado del sistema en un momento dado, este último se describe a través de los procesos que participan en el algoritmo junto con su estado local, que típicamente será la *vista* de cada

proceso, que contiene la información que cada proceso *aprende* durante la ejecución del algoritmo. Una configuración será representada mediante un simplejo cuyos vértices son los procesos junto con sus respectivas vistas, en conjunto las distintas configuraciones forman un complejo simplicial, cuyas propiedades topológicas determinarán si una tarea dada puede ser resuelta a través del protocolo en cuestión.

El estudio de cómputo distribuido a través de la topología combinatoria se ha centrado en la descripción de algortimos distribuidos y deterministas, sin embargo existe una amplia clase de algoritmos distribuidos y aleatorios que han sido desarollados para resolver tareas que no se pueden solucionar de forma determinista o bien para hacer más eficientes algunos procedimientos [13], sobre estos algoritmos no se ha realizado un estudio que permita entender a dichos algoritmos a través de la topología combinatoria.

El objetivo de esta tesis es hacer una contribución al estudio de algoritmos distribuidos y aleatorios a través de su topología. Para ello, primero es necesario entender cómo se ha estudiado la clase de algoritmos deterministas para después pasar a analizar el caso aleatorio. El texto está divido en dos partes, en la primera parte (capítulos 2 y 3), primero se presenta el lenguaje de topología que será necesario para describir la relación que ésta tiene con el cómputo distribuido, la intención de este texto es ser autocontenido por lo que se enuncian todas las definiciones que se consideran necesarias, solamente se omiten algunas definiciones de topología básica (vea [14]), después en el capítulo 3 se describe el *modelo iterado de instantáneas inmediatas* en su versión computacional y combinatoria, estos dos capítulos pueden ser de utilidad para aquellos que quieran comenzar a explorar esta área del conocimiento en la que la computación distribuida y la topología parecen tener una estrecha relación. La segunda parte del texto (capítulos $4, 5$ y 6), comienza con el capítulo 4, en el que se establece un modelo de cómputo aleatorio basado en el modelo de instantáneas inmediatas, para lograr el comportamiento aleatorio los procesos son equipados con un procedimiento que les permite obtener valores aleatorios a través de lanzar una moneda con cierta distribución de probabilidad, en ese mismo capítulo se describe la topología del complejo formado por las posibles configuraciones que se obtienen después de que los procesos ejecutan el *protocolo aleatorio*; finalmente en el capítulo 5 y 6 se

estudian algunas propiedades topológicas del modelo aleatorio, mientras que en el capítulo 7 se enuncian las conclusiones de este trabajo y las direcciones en las que el mismo podría continuar.

Capítulo 2

Elementos de topología combinatoria

En este capítulo se presentan conceptos básicos de topología que serán necesarios para describir y modelar algunas nociones de cómputo distribuido que se presentarán en el siguiente capítulo, estos conceptos corresponden a un área de la topología denominada *topología combinatoria* y han sido transcritos de [6, 9]. Los espacios topológicos que serán de interés para este estudio son aquellos denominados *complejos simpliciales*, que son espacios cuya estructura se puede describir a partir de piezas mas simples, denominados *simplejos*, y de saber cómo estas piezas están integradas para conformar a uno de estos complejos simpliciales. El hecho de poder describir a un espacio a través de piezas mas simples y de las relaciones entre ellas permitirá que el estudio de propiedades topológicas, como la conexidad, pueda ser analizado de una forma sencilla.

2.1. Complejos Simpliciales

Definición 2.1.1 (Complejo Simplicial Abstracto)**.** Dado un conjunto S y una familia $\mathcal{A}$ de subconjuntos finitos de S, se dice que $\mathcal{A}$ es un *complejo simplicial abstracto* si satisface:

(1) Si $X \in \mathcal{A}$ y $Y \subseteq X$, entonces $Y \in \mathcal{A}$.

(2) $\{v\} \in \mathcal{A}$, para todo $v \in S$.

Un elemento de $v \in S$ es llamado *vértice*, mientras que un elemento $\sigma \in \mathcal{A}$ es llamado *simplejo*. El conjunto de vértices de $\mathcal{A}$ se denota por $V(\mathcal{A})$. La *dimensión* de un simplejo $\sigma \in \mathcal{A}$ es $|\sigma| - 1$.

Usualmente se usarán letras latinas minúsculas para denotar a los vértices (u,v,w,...), letras griegas minúsculas para denotar a los simplejos $(\sigma, \tau, \alpha, ...)$ y letras con estilo caligráfico para denotar a los complejos simpliciales $(\mathcal{A}, \mathcal{B}, ...)$.

Un complejo $\mathcal{B}$ es un *subcomplejo* de $\mathcal{A}$ si cada simplejo de $\mathcal{B}$ es también un simplejo de $\mathcal{A}$.

Un simplejo τ es una *cara* de σ si $\tau \subseteq \sigma$, y es una *cara propia* si $\tau \subsetneq \sigma$. Un simplejo $\sigma \in \mathcal{A}$ es una *faceta* si no es una cara propia de algún otro simplejo en $\mathcal{A}$. El conjunto de las facetas de un complejo $\mathcal{A}$ se denotará por $facet(\mathcal{A})$. La *dimensión* de un complejo $\mathcal{A}$ es la máxima dimensión de cualesquiera de sus facetas. Un complejo es *puro* si todas sus facetas tienen la misma dimensión. La *codimensión* de un simplejo σ de dimensión m en un complejo puro de dimensión n (codim σ) es $n - m$.

En el estudio de mapeos entre complejos simpliciales, serán de gran importancia aquellos que preserven estructura, que serán denominados *mapeos simpliciales*.

Definición 2.1.2. Dados dos complejos simpliciales $\mathcal{A}$ y $\mathcal{B}$, y un mapeo $\mu : V(\mathcal{A}) \to V(\mathcal{B})$; se dice que μ es un *mapeo simplicial* entre $\mathcal{A}$ y $\mathcal{B}$, denotado por $\mu : \mathcal{A} \to \mathcal{B}$, si μ envía simplejos de $\mathcal{A}$ en simplejos de $\mathcal{B}$, es decir, si σ es un simplejo de $\mathcal{A}$, entonces $\mu(\sigma)$ es un simplejo de $\mathcal{B}$.

Definición 2.1.3. Dados dos complejos simpliciales $\mathcal{A}$ y $\mathcal{B}$, se dice que $\mathcal{A}$ es *isomorfo* a $\mathcal{B}$ y se denota por $\mathcal{A} \cong \mathcal{B}$, si existen mapeos simpliciales $\phi : \mathcal{A} \to \mathcal{B}$ y $\psi : \mathcal{B} \to \mathcal{A}$, tales que para todo vértice $a \in \mathcal{A}$, $a = \psi(\phi(a))$, y para todo $b \in \mathcal{B}$, $b = \phi(\psi(b))$.

Definición 2.1.4. Dados dos complejos simpliciales $\mathcal{A}$ y $\mathcal{B}$. Un mapeo simplicial $\varphi : \mathcal{A} \to \mathcal{B}$ es *rígido* si la imagen de cada simplejo σ tiene la misma dimensión que σ, es decir, $|\varphi(\sigma)| = |\sigma|$.

2.1.1. Complejos Geométricos

Se usará la notación $[m:n]$, con $n \geq m$, como abreviación del conjunto $\{m, m+1, ...n\}$, y $[n]$ para abreviar $[0:n]$. Un punto y en $\mathbb{R}^d$ es *combinación afín* de un conjunto finito de puntos $X = \{x_0, x_1, ..., x_n\}$ en $\mathbb{R}^d$, si éste puede ser expresado como una suma

$$y = \sum_{i=0}^{n} t_i \cdot x_i, \tag{2.1}$$

donde la suma los coeficientes t_i es igual a uno, si además cada uno de los coeficientes es positivo, se dice que y es una *combinación convexa* de X. La *cerradura convexa* de X, denotada por $convX$, es el conjunto de combinaciones convexas de X. El conjunto X es *afínmente independiente* si ningún punto del conjunto puede ser expresado como una combinación afín de los demás.

El *simplejo estándar de dimensión* n Δ^n, es la cerradura convexa del conjunto de $n+1$ puntos en $\mathbb{R}^{n+1}$ que tienen coordenadas $(1, 0, ..., 0), (0, 1, ..., 0), ..., (0, 0, ..., 1)$. De forma más general, un *simplejo geométrico de dimensión* n o un *n-simplejo geométrico*, es la cerradura convexa de cualquier conjunto de $n+1$ puntos afínmente independientes en $\mathbb{R}^d$ $(d \geq n)$. Cuando $v_0, v_1, ..., v_n \in \mathbb{R}^d$ son afínmente independientes, se dice que esos puntos son los *vértices* del $n-$simplejo $\sigma = conv\{v_0, v_1, ..., v_n\}$. En este caso, para cualquier $S \subseteq [n]$, el $(|S|-1)-$simplejo $\tau = conv\{v_s | s \in S\}$ es una *cara*, o una $(|S|-1)-$cara de σ; se llama *cara propia* si además $S \neq [n]$.

Definición 2.1.5 (Complejo Simplicial Geométrico)**.** Un *complejo simplicial geométrico* $\mathcal{K}$ en $\mathbb{R}^d$ es una colección de simplejos geométricos tales que:

(1) Cualquier cara de un elemento $\sigma \in \mathcal{K}$ también es elemento de $\mathcal{K}$.

(2) Para cualesquiera $\sigma, \tau \in \mathcal{K}$, su intersección $\sigma \cap \tau$ es una cara de cada uno de ellos.

Para cualquier simplejo geométrico $\sigma = conv(v_0, v_1, ..., v_n)$ de dimensión n con un orden fijo en el conjunto de sus vértices, se tiene un único mapeo afín $\varphi : \Delta^n \to \sigma$ que envía el $i-$ésimo vértice de Δ^n a v_i. Este mapeo es llamado el *mapeo característico* de σ.
Dado un complejo simplicial geométrico $\mathcal{K}$, se puede definir un complejo simplicial abstracto subyacente $\mathcal{C}(\mathcal{K})$ como sigue: los vértices de $\mathcal{C}(\mathcal{K})$ son los que se obtienen al unir

todos los conjuntos de vértices de los simplejos de $\mathcal{K}$; además, para cada simplejo $\sigma = conv\{v_0, v_1, ..., v_n\}$ de $\mathcal{K}$, el conjunto $\{v_0, v_1, ..., v_n\}$ es un simplejo de $\mathcal{C}(\mathcal{K})$. En la dirección contraria, dado un complejo simplicial abstacto $\mathcal{A}$ con una cantidad finita de vértices, existen varios complejos simpliciales geométricos $\mathcal{K}$, tales que $\mathcal{C}(\mathcal{K}) = \mathcal{A}$. La construcción mas sencilla es la siguiente: Suponga que $\mathcal{A}$ tiene $d+1$ vértices; tome el simplejo estándar Δ^d en $\mathbb{R}^{d+1}$, y tome el subcomplejo de Δ^d que consiste de los complejos geométricos que corresponden a los conjuntos de vértices de la familia $\mathcal{A}$.

2.1.2. Realización geométrica

Dado un complejo simplicial geométrico $\mathcal{K}$ en $\mathbb{R}^d$, se denota por $|\mathcal{K}|$ a la unión de todos sus simplejos. Este espacio con la topología inducida como subespacio de $\mathbb{R}^d$ se llama la *realización geométrica* de $\mathcal{K}$. Si $\mathcal{A}$ es un complejo simplicial abstracto, se puede construir $\mathcal{K}$, tal que $\mathcal{C}(\mathcal{K}) = \mathcal{A}$, y después definir la realización geométrica de $\mathcal{A}$, como $|\mathcal{A}| := |\mathcal{K}|$. Esta definición no depende de la elección $\mathcal{K}$, sino solamente de $\mathcal{A}$.
Sean $\mathcal{A}$ y $\mathcal{B}$ complejos simpliciales abstactos. Recuerde que un mapeo simplicial $\mu : V(\mathcal{A}) \to V(\mathcal{B})$ mapea cada vértice de $\mathcal{A}$ en algún vértice de $\mathcal{B}$ y también mapea simplejos de $\mathcal{A}$ en simplejos de $\mathcal{B}$. A partir de un mapeo simplicial μ es posible definir una aplicación continua $|\mu|$ entre las realizaciones geométricas de la siguiente manera:

Para cualquier $n-$simplejo $\sigma = \{s_0, s_1, ..., s_n\}$ en $\mathcal{A}$, el mapeo $|\mu|$ se define en los puntos de $|\sigma|$ extendiendo de la siguiente forma:

$$|\mu|\Big(\sum_{i=0}^{n} t_i s_i\Big) = \sum_{i=0}^{n} t_i \mu(s_i). \tag{2.2}$$

Observación 2.1.6. (Sobre el uso del término *simplejo*) La palabra *simplejo* se usará de forma indistinta para referirse tanto a un elemento de un complejo simplicial abstracto como a un complejo simplicial abstracto que consta de todos los subconjuntos finitos de cierto conjunto finito. Si uno tiene un simplejo en el primer sentido, el conjunto potencia de dicho conjunto constituye un simplejo en el segundo sentido.

2.2. Mapeos Portadores

Definición 2.2.1. Dados dos complejos simpliciales $\mathcal{A}$ y $\mathcal{B}$, un *mapeo portador* Φ de $\mathcal{A}$ en $\mathcal{B}$ es un mapeo que lleva a cada simplejo $\sigma \in \mathcal{A}$ a un subcomplejo $\Phi(\sigma)$ de $\mathcal{B}$ ($\Phi : \mathcal{A} \to 2^{\mathcal{B}}$), de modo que para cualesquiera $\sigma, \tau \in \mathcal{A}$ tales que $\sigma \subseteq \tau$, se tiene que $\Phi(\sigma) \subseteq \Phi(\tau)$.

Usualmente los mapeos portadores serán denotados por letras griegas mayúsculas $(\Delta, \Phi, \Xi, ...)$.

Una consecuencia inmediata de la definición anterior es que para cualesquiera $\sigma, \tau \in \mathcal{A}$, $\Phi(\sigma \cap \tau) \subseteq \Phi(\sigma) \cap \Phi(\tau)$, pues por definición $\Phi(\sigma \cap \tau) \subseteq \Phi(\sigma)$ y $\Phi(\sigma \cap \tau) \subseteq \Phi(\tau)$.

Observación 2.2.2. Para un subcomplejo $\mathcal{K} \subseteq \mathcal{A}$, se usará la notación $\Phi(\mathcal{K}) := \bigcup_{\sigma \in \mathcal{K}} \Phi(\sigma)$. Note que la condición de monotonía de los mapeos portadores permite que la unión descrita anteriormente pueda ser tomada únicamente sobre las facetas de $\mathcal{K}$, es decir, $\Phi(\mathcal{K}) = \bigcup_{\sigma \in facet(\mathcal{K})} \Phi(\sigma)$.

Definición 2.2.3. Dados dos complejos simpliciales abstractos $\mathcal{A}$ y $\mathcal{B}$ y un mapeo portador $\Phi : \mathcal{A} \to 2^{\mathcal{B}}$.

(1) El mapeo Φ es *rígido* si para cada simplejo $\sigma \in \mathcal{A}$ de dimensión d, el subcomplejo $\Phi(\sigma)$ es puro de dimensión d.

(2) Si la igualdad $\Phi(\sigma \cap \tau) = \Phi(\sigma) \cap \Phi(\tau)$ se cumple, se dice que que el mapeo Φ es *estricto*.

Es posible componer mapeos simpliciales con mapeos portadores.

Definición 2.2.4. Dados tres complejos simpliciales abstractos $\mathcal{A}, \mathcal{B}$ y $\mathcal{C}$ y un mapeo portador Φ de $\mathcal{A}$ en $\mathcal{B}$.

Si $\varphi : \mathcal{B} \to \mathcal{C}$ es un mapeo simplicial, entonces se puede definir un mapeo portador $\varphi \circ \Phi$ de $\mathcal{A}$ en $\mathcal{C}$ como $(\varphi \circ \Phi)(\sigma) := \varphi(\Phi(\sigma))$ para todo $\sigma \in \mathcal{A}$, donde $\varphi(\Phi(\sigma)) = \bigcup_{\tau \in \Phi(\sigma)} \varphi(\tau)$.

Definición 2.2.5. Dados dos mapeos portadores $\Phi : \mathcal{A} \to 2^{\mathcal{B}}$ y $\Psi : \mathcal{B} \to 2^{\mathcal{C}}$, donde $\mathcal{A}, \mathcal{B}$ y $\mathcal{C}$ son complejos simpliciales, se define un mapeo portador $\Psi \circ \Phi : \mathcal{A} \to 2^{\mathcal{C}}$ tomando $(\Psi \circ \Phi)(\sigma) := \bigcup_{\tau \in \Phi(\sigma)} \Psi(\tau)$, es decir, $(\Psi \circ \Phi)(\sigma) = \Psi(\Phi(\sigma))$ para todo $\sigma \in \mathcal{A}$.

Proposición 2.2.6. *Dados dos mapeos portadores $\Phi : \mathcal{A} \to 2^{\mathcal{B}}$ y $\Psi : \mathcal{B} \to 2^{\mathcal{C}}$, donde $\mathcal{A}, \mathcal{B}$ y $\mathcal{C}$ son complejos simpliciales.*

(1) Si Φ y Ψ son mapeos portadores rígidos, entonces también lo es $\Psi \circ \Phi$.

(2) Si Φ y Ψ son mapeos portadores estrictos, entonces también lo es $\Psi \circ \Phi$.

Demostración. La demostración fue transcrita de [9].

(1) Sea σ un simplejo de dimensión d de $\mathcal{A}$. Como Φ es rígido, entonces $\Phi(\sigma)$ es puro de dimensión d, de modo cualquier faceta τ de $\Phi(\sigma)$ es de dimensión d, y al ser Ψ rígido, se cumple que $\Psi(\tau)$ es un complejo puro de dimensión d. La unión de complejos puros de la misma dimensión es también un complejo puro de la misma dimensión, pues las facetas de todos los complejos son las facetas del complejo que se obtiene al unirlos. Lo anterior prueba que $\bigcup_{\tau \in facet(\Phi(\sigma))} \Psi(\tau)$ es puro de dimensión d, pero este complejo no es mas que $(\Psi \circ \Phi)(\sigma) = \Psi(\Phi(\sigma))$ (Observación 2.2.2). Lo que prueba que $\Psi \circ \Phi$ es rígido.

(2) Sean σ_1, σ_2, elementos de $\mathcal{A}$. Se tiene que

$$\Psi(\Phi(\sigma_1)) \cap \Psi(\Phi(\sigma_2)) = \Big(\bigcup_{\tau_1 \in \Phi(\sigma_1)} \Psi(\tau_1) \Big) \cap \Big(\bigcup_{\tau_2 \in \Phi(\sigma_2)} \Psi(\tau_2) \Big) \quad =$$

$$\bigcup_{\tau_1 \in \Phi(\sigma_1), \tau_2 \in \Phi(\sigma_2)} \big(\Psi(\tau_1) \cap \Psi(\tau_2) \big) = \bigcup_{\tau_1 \in \Phi(\sigma_1), \tau_2 \in \Phi(\sigma_2)} \Psi(\tau_1 \cap \tau_2) \quad =$$

$$\bigcup_{\tau \in \Phi(\sigma_1) \cap \Phi(\sigma_2)} \Psi(\tau) = \bigcup_{\tau \in \Phi(\sigma_1 \cap \sigma_2)} \Psi(\tau) \quad = \Psi(\Phi(\sigma_1 \cap \sigma_2)).$$

De lo que se concluye que $\Psi \circ \Phi$ es un mapeo estricto. □

2.3. Complejos Cromáticos

Definición 2.3.1. Un *m-etiquetado* o simplemente *etiquetado* de un complejo $\mathcal{A}$ es una función $\varphi : V(\mathcal{A}) \to D$, donde D es algún conjunto con m elementos.

Definición 2.3.2. Una *m-coloración* o simplemente *coloración* de un complejo $\mathcal{A}$ es un m-etiquetado $\chi : V(\mathcal{A}) \to \Pi$ que es inyectivo en los vértices de cada simplejo $\sigma \in \mathcal{A}$, es decir, si $s_0, s_1 \in \sigma$, entonces $\chi(s_0) \neq \chi(s_1)$. Un complejo simplicial junto con una m-coloración

χ es llamado un *complejo cromático* (o *complejo m-cromático*), y se denota por $(\mathcal{A}, \chi_{\mathcal{A}})$ y por $(\mathcal{A}, \chi)$, cuando el contexto no dé lugar a confusiones.

Definición 2.3.3. Dados dos complejos m-cromáticos $(\mathcal{A}, \chi_{\mathcal{A}})$ y $(\mathcal{B}, \chi_{\mathcal{B}})$, se dice que un mapeo simplicial $\phi : \mathcal{A} \to \mathcal{B}$ *preserva color* si para todo vértice $v \in V(A)$, $\chi_{\mathcal{A}}(v) = \chi_{\mathcal{B}}(\phi(v))$.

Definición 2.3.4. Dados dos complejos cromáticos $\mathcal{A}$ y $\mathcal{B}$ y un mapeo portador $\Phi : \mathcal{A} \to 2^{\mathcal{B}}$. Se dice que Φ es *cromático* si Φ es rígido y si para todo $\sigma \in \mathcal{A}$ se tiene que $\chi_{\mathcal{A}}(\sigma) = \chi_{\mathcal{B}}(\Phi(\sigma))$, donde $\chi_{\mathcal{B}}(\Phi(\sigma)) := \{\chi_{\mathcal{B}}(v) | v \in V(\Phi(\sigma))\}$.

Proposición 2.3.5. *Dados dos mapeos portadores $\Phi : \mathcal{A} \to 2^{\mathcal{B}}$ y $\Psi : \mathcal{B} \to 2^{\mathcal{C}}$, donde $\mathcal{A}, \mathcal{B}$ y $\mathcal{C}$ son complejos simpliciales cromáticos. Si Φ y Ψ son mapeos portadores cromáticos, entonces también lo es $\Psi \circ \Phi$.*

Demostración. En primer lugar, la composición $\Psi \circ \Phi$ es un mapeo portador rígido, de acuerdo con la proposición 2.2.6. Resta probar que $\chi_{\mathcal{A}}(\sigma) = \chi_{\mathcal{C}}((\Psi \circ \Phi)(\sigma))$. Para ello, note que

$$\begin{aligned} \chi_{\mathcal{C}}((\Psi \circ \Phi)(\sigma)) = \chi_{\mathcal{C}}(\bigcup_{\alpha \in \Phi(\sigma)} \Psi(\alpha)) \quad &= \bigcup_{\alpha \in \Phi(\sigma)} \chi_{\mathcal{C}}(\Psi(\alpha)) \quad = \bigcup_{\alpha \in \Phi(\sigma)} \chi_{\mathcal{B}}(\alpha) = \\ = \chi_{\mathcal{B}}(\bigcup_{\alpha \in \Phi(\sigma)} \alpha) \quad &= \chi_{\mathcal{B}}(\Phi(\sigma)) \quad = \chi_{\mathcal{A}}(\sigma). \end{aligned}$$

Como se quería demostrar. □

2.4. Complejos Shellables

Definición 2.4.1. Un complejo simplicial $\mathcal{C}$ es *shellable* si sus facetas pueden formar una sucesión $\phi_0, \phi_1, ..., \phi_t$, a la que se denominará *sucesión shellable*, de modo que el complejo $(\bigcup_{i=0}^{k-1} \phi_i) \cap \phi_k$ es unión de caras de dimensión $dim(\phi_k) - 1$ de ϕ_k, para $0 < k \leq t$. En lo que sigue, todos los complejos shellables serán considerados puros.

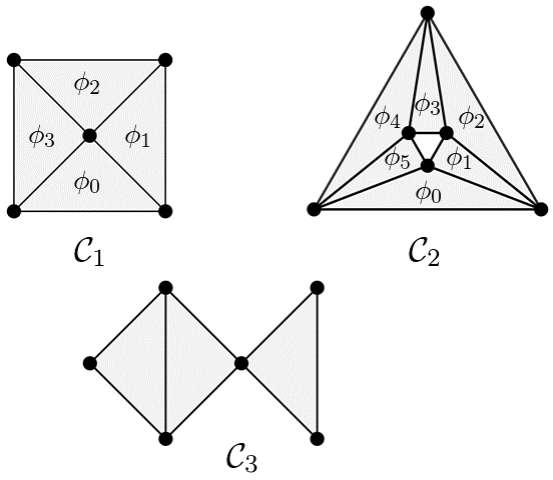

Figura 2.1: Los complejos $\mathcal{C}_1$ y $\mathcal{C}_2$ con shellables, mientras $\mathcal{C}_3$ no lo es.

El siguiente lema será de utilidad mas adelante.

Lema 2.4.2. *La unión de caras de dimensión $n-1$ de un simplejo de dimensión n es un complejo shellable y puro de dimensión $n-1$.*

Demostración. Si $\tau_1, \tau_2, ..., \tau_r$ son caras (distintas) de dimensión $n-1$ de σ de dimensión n, entonces $\tau_i \cap \tau_j$ es una cara de dimensión $n-2 = dim(\tau_i) - 1 = dim(\tau_j) - 1$, si $i \neq j$. De modo que $\tau_1, \tau_2, ..., \tau_r$ es una sucesión shellable para el complejo $\bigcup_{j=1}^{r} \tau_j$, pues para $k \in \{1, 2, ..., r\}$ se cumple que $(\bigcup_{i=1}^{k-1} \tau_i) \cap \tau_k$ es unión de caras de dimensión $dim(\tau_k) - 1 = n-2$ de τ_k, ya que $(\bigcup_{i=1}^{k-1} \tau_i) \cap \tau_k = \bigcup_{i=1}^{k-1} (\tau_i \cap \tau_k)$ y como ya se mencionó, $\tau_i \cap \tau_j$ es una cara de dimensión $n-2$ de τ_k.

Para ver que el comlejo es puro, basta notar que las facetas de dicho complejo son $\tau_1, \tau_2, ..., \tau_r$, todas ellas de dimensión $n-1$. □

2.5. Conexidad

Definición 2.5.1. Sea $\mathcal{K}$ un complejo simplicial. Un *camino* (o *trayectoria*) entre dos vértices u y v en $\mathcal{K}$, es una sucesión de vértices $u = v_0, v_1, ..., v_l = v$, tal que cada pareja $\{v_i, v_{i+1}\}$ es una arista de $\mathcal{K}$, para $0 \leq i < l$. Un camino es *simple* si los vertices son distintos.

Definición 2.5.2. Un complejo simplicial $\mathcal{K}$ es *conectable por trayectorias* si existe un camino entre cualesquiera dos vértices de $\mathcal{K}$. Los mayores subcomplejos de $\mathcal{K}$ que son conectables por trayectorias son las *componentes conexas* de $\mathcal{K}$.

Definición 2.5.3. Sea k cualquier entero no negativo. El complejo $\mathcal{K}$ es *k-conexo* si para toda $0 \leq l \leq k$, cualquier mapeo continuo $f : S^l \to |\mathcal{K}|$ puede ser extendido continuamente a $F : D^{l+1} \to |\mathcal{K}|$, donde la esfera S^l es la frontera de D^{l+1}, y D^{l+1} consta de todos los puntos en R^{l+1} cuya distancia al origen es menor o igual que uno.

Si $k = -1$, se dirá que un complejo $\mathcal{K}$ es $-1-$conexo si $\mathcal{K}$ es no vacío.

La siguiente observación habla sobre la conexidad de un simplejo de dimensión n, dicho resultado se sustenta en el hecho que la realización geométrica de un simplejo de dimensión n es isomorfa al disco de dimensión n, D^n.

Observación 2.5.4. Si σ es un simplejo de dimensión n, entonces σ es $(n-1)-$conexo.

Definición 2.5.5. Suponga que $\mathcal{K}$ y $\mathcal{L}$ son complejos simpliciales, tales que $\mathcal{K}$ es puro. Un *mapeo portador q-conexo* $\Phi : \mathcal{K} \to 2^{\mathcal{L}}$ es un mapeo portador rígido y estricto tal que para cada $\sigma \in \mathcal{K}$, el complejo $\Phi(\sigma)$ es $(q - \operatorname{codim} \sigma)-$conexo.

Una herramienta útil para determinar la conexidad de un complejo simplicial es el *Lema del Nervio*, que se presenta a continuación, en este texto se enuncia sin su demostración, pero ésta puede ser consultada en [5].

Definición 2.5.6 (Complejo del Nervio)**.** Dado un complejo simplicial $\mathcal{K}$ y $(\mathcal{K}_i)_{i\in I}$ una familia de subcomplejos no vacíos que cubren a $\mathcal{K}$, es decir, $\mathcal{K} = \bigcup_{i\in I} \mathcal{K}_i$. El *nervio* de la cubierta $(\mathcal{K}_i)_{i\in I}$, es el complejo simplicial abstracto $\mathcal{N}(\mathcal{K}_i | i \in I)$, cuyos vértices son las componentes $\mathcal{K}_i$ y cuyos simplejos son conjuntos de componentes $\{\mathcal{K}_j | j \in J\}$, de los cuales la intersección $\bigcap_{j\in J} \mathcal{K}_j$ es no vacía.

Proposición 2.5.7 (Lema del Nervio)**.** *Sea $\{\mathcal{K}_i | i \in I\}$ una cubierta de un complejo simplicial $\mathcal{K}$, y sea k un entero positivo fijo. Para cualquier subconjunto de índices $J \subseteq I$, se define $\mathcal{K}_J = \bigcap_{j\in J} \mathcal{K}_j$. Si se cumple que $\mathcal{K}_J$ es vacío o $(k - |J| + 1)-$ conexo, para cualquier $J \subseteq I$, entonces $\mathcal{K}$ es $k-$conexo si y sólo si el complejo del nervio $\mathcal{N}(\mathcal{K}_i | i \in I)$ es $k-$conexo.*

Corolario 2.5.8. *Si $\mathcal{K}_1$ y $\mathcal{K}_2$ son complejos simpliciales $k-$conexos, tales que $\mathcal{K}_1 \cap \mathcal{K}_2$ es $(k-1)-$conexo, entonces el complejo simplicial $\mathcal{K}_1 \cup \mathcal{K}_2$ es $k-$conexo.*

Demostración. Una cubierta para $\mathcal{K}_1 \cup \mathcal{K}_2$ es $\{\mathcal{K}_1, \mathcal{K}_2\}$. En este caso el nervio de la cubierta es una arista con vértices $\mathcal{K}_1$ y $\mathcal{K}_2$. Siguiendo la notación de la proposición anterior, se cumple que para cualquier subconjunto J de $I = \{1, 2\}$, $\mathcal{K}_J = \bigcap_{j \in J} \mathcal{K}_j$ es vacío o $(k - |J| + 1)-$ conexo, ya que si $J = \{1\}$, entonces $\mathcal{K}_J$ es $\mathcal{K}_1$, y éste cumple con ser $k = k - (1) + 1 = (k - |J| + 1)-$ conexo. El argumento anterior también es válido si $J = \{2\}$. Si $J = \{0, 1\}$, entonces $\mathcal{K}_J = \mathcal{K}_1 \cap \mathcal{K}_2$ es por hipótesis $k - 1 = k - (2) + 1 = (k - |J| + 1)-$ conexo. En estas condiciones el Lema del Nervio asegura que $\mathcal{K}_1 \cup \mathcal{K}_2$ es $k-$conexo si el complejo $\mathcal{N}(\mathcal{K}_1, \mathcal{K}_2)$ es $k-$conexo. Como ya se dijo antes el nervio de la cubierta es simplemente una arista, que en efecto es $k-$conexa, por lo que $\mathcal{K}$ también es $k-$conexo. □

2.6. Subdivisiones

Definición 2.6.1. (Subdivisión Cromática) Sea $(\mathcal{A}, \chi)$ un complejo simplicial abstracto cromático y $\sigma \in \mathcal{A}$. Se define la *subdivisión cromática estándar* de σ, Chσ, como el complejo cuyos vértices de son de la forma (c, τ), donde $c \in [n]$, τ es una cara no vacía de σ y $c \in \chi(\tau)$; además, un conjunto con $d + 1$ vértices $\{(c_0, \tau_0), ..., (c_d, \tau_d)\}$ es un $d-$simplejo de Chσ si cumple las siguientes condiciones:

(1) Para cualquier $i, j \in \{0, ..., d\}$, se tiene que $\tau_i \subseteq \tau_j$ o $\tau_j \subseteq \tau_i$.

(2) Para cualquier $i, j \in \{0, ..., d\}$, si $i \in \chi(\tau_j)$, entonces $\tau_i \subseteq \tau_j$.

Finalmente, para que Chσ sea cromático se define $\chi(v, \tau) = v$.

Para un complejo simplicial abstracto cromático $(\mathcal{A}, \chi)$, el complejo Ch$\mathcal{A}$ es igual a $\bigcup_{\sigma \in \mathcal{A}} \text{Ch}\sigma$.

Observación 2.6.2. (Propiedades de la subdivisión cromática) Sea σ cualquier simplejo de un complejo simplicial $\mathcal{I}$. El complejo Ch(σ) cumple que es $(dim(\sigma) - 1)-$conexo y además es shellable [9, 10].

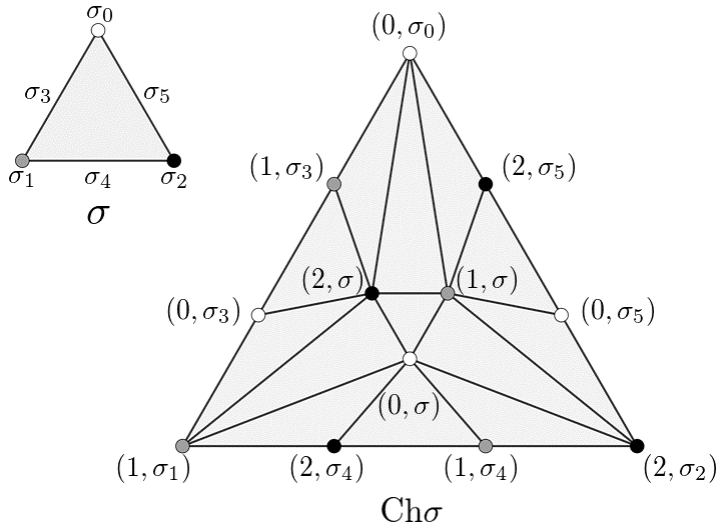

Figura 2.2: El simplejo σ y su subdivisión cromática Chσ.
Los vértices de σ son σ_0, σ_1 y σ_2, mientras que sus aristas son σ_3, σ_4 y σ_5.

2.7. Construcciones

2.7.1. Unión

Definición 2.7.1. Dados dos complejos simpliciales $\mathcal{A}$ y $\mathcal{B}$ cuyos conjuntos de vértices $V(\mathcal{A})$ y $V(\mathcal{B})$ son ajenos, el *complejo unión* [1] de $\mathcal{A}$ y $\mathcal{B}$, denotado por $\mathcal{A} * \mathcal{B}$, es el complejo simplicial abstracto cuyo conjunto de vértices es $V(\mathcal{A}) \cup V(\mathcal{B})$ y cuyos simplejos son las uniones $\alpha \cup \beta$, donde $\alpha \in \mathcal{A}$ y $\beta \in \mathcal{B}$.

De la definición de complejo simplicial abstracto se tiene que $\emptyset \in \mathcal{A}$ y $\emptyset \in \mathcal{B}$, de modo que $\mathcal{A}$ y $\mathcal{B}$ son subcomplejos de $\mathcal{A} * \mathcal{B}$. La figura 2.3 muestra algunos de ejemplos de la unión de complejos recién definida.

Proposición 2.7.2. *Para cualesquiera dos complejos simpliciales $\mathcal{A}$ y $\mathcal{B}$, se cumple que*

$$facet(\mathcal{A} * \mathcal{B}) = \{\alpha \cup \beta | \alpha \in facet(\mathcal{A}), \beta \in facet(\mathcal{B})\}.$$

Demostración. Sea γ una faceta de $\mathcal{A} * \mathcal{B}$. Al ser γ un simplejo de $\mathcal{A} * \mathcal{B}$, γ es de la forma $\alpha \cup \beta$, donde α y β son simplejos de $\mathcal{A}$ y $\mathcal{B}$, respectivamente. Si alguno de ellos no es una faceta de su respectivo complejo, por ejemplo si α no es una faceta de $\mathcal{A}$, se tiene que α

[1] No debe confundirse al complejo unión de $\mathcal{A}$ y $\mathcal{B}$ ($\mathcal{A} * \mathcal{B}$) con la unión de $\mathcal{A}$ y $\mathcal{B}$ ($\mathcal{A} \cup \mathcal{B}$).

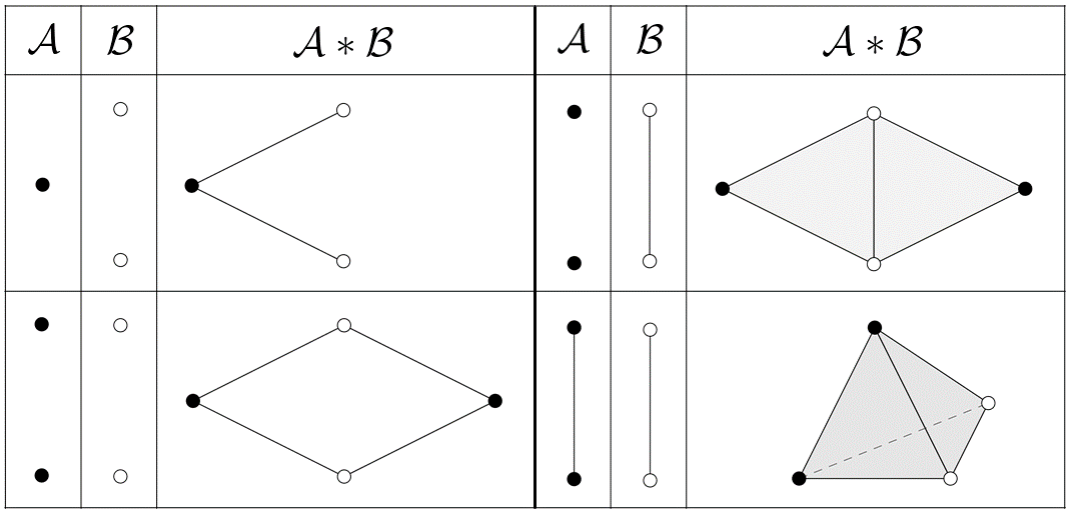

Figura 2.3: Ejemplos de la unión de complejos.

está contenido propiamente en algún otro simplejo de $\mathcal{A}$, digamos α', de lo que se sigue que γ está contenido propiamente en $\alpha' \cup \beta$, contradiciendo el hecho que γ es una faceta de $\mathcal{A} * \mathcal{B}$. El mismo argumento es válido si se supone que β no es una faceta de $\mathcal{B}$, así que tanto α y β son facetas de sus respectivos complejos.

Ahora considere a α una faceta de $\mathcal{A}$ y β una faceta de $\mathcal{B}$, su unión $\gamma = \alpha \cup \beta$, es un simplejo de $\mathcal{A} * \mathcal{B}$. Si γ no es una faceta de $\mathcal{A} * \mathcal{B}$, entonces está contenido propiamente en otro simplejo $\gamma' \in \mathcal{A} * \mathcal{B}$, donde $\gamma' = \alpha' \cup \beta'$ y, α' y β' son simplejos de $\mathcal{A}$ y $\mathcal{B}$, respectivamente. Considere la intersección

$$\alpha \cap (\alpha' \cup \beta') = (\alpha \cap \alpha') \cup (\alpha \cap \beta') = \alpha \cap \alpha'.$$

La última igualdad se debe a que $\alpha \cap \beta' = \emptyset$, pues $\mathcal{A}$ y $\mathcal{B}$ no tienen vértices en común. Además, como $\alpha \subseteq \alpha' \cup \beta'$, entonces $\alpha = \alpha \cap (\alpha' \cup \beta') = \alpha \cap \alpha'$, por lo que $\alpha \subseteq \alpha'$; de forma similar también se sigue que $\beta \subseteq \beta'$. Una de las inclusiones anteriores debe de ser propia, de lo contrario γ sería igual a γ', pero con esto se tendría que $\alpha \subsetneq \alpha'$ o $\beta \subsetneq \beta'$, contradiciendo el hecho que α y β son facetas de los complejos $\mathcal{A}$ y $\mathcal{B}$, respectivamente. De modo que γ es una faceta de $\mathcal{A} * \mathcal{B}$. Esto concluye la prueba de la proposición. □

2.7.2. Pseudoesferas

Definición 2.7.3 (Esfera combinatoria)**.** Sea $\sigma = \{v_0, v_1, ..., v_n\}$ un simplejo de dimensión n. La *esfera combinatoria* (también llamada *pseudoesfera*) de dimensión n, denotada por $\Psi(\sigma)$, es el complejo definido como sigue:

(1) Cada pareja (v_i, v) es un vértice, donde $v \in \{0, 1\}$, $i \in \{0, ..., n\}$.

(2) Para cualquier subconjunto $J \subseteq \{0, ..., n\}$, el conjunto $\{(v_j, v) | j \in J, v \in \{0, 1\}\}$ es un simplejo si los v_j son distintos.

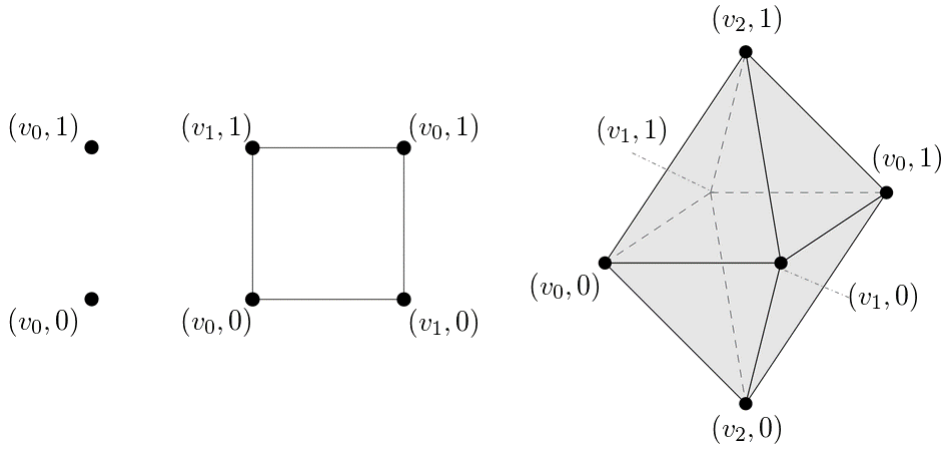

Figura 2.4: Pseudoesferas de dimensión 0, 1 y 2.

La definición anterior se puede presentar de forma mas general (ver [9]) permitiendo que la segunda coordenada de los vértices tome cualquier valor de un conjunto finito determinado, sin embargo para los fines que se requieren en este trabajo, la definición anterior es suficiente.

Proposición 2.7.4. *Para cualesquiera simplejos σ, τ, tales que $\sigma \subseteq \tau$, se tiene que $\Psi(\sigma) \subseteq \Psi(\tau)$.*

Demostración. Suponga que $\sigma = \{v_0, ..., v_m\}$ y $\tau = \{v_0, ..., v_m, ..., v_n\}$. Los vértices $\Psi(\sigma)$ son de la forma (v_i, v), con $v_i \in \sigma$ y $v \in \{0, 1\}$. Como cada $v_i \in \sigma$ también es un elemento de τ, entonces (v_i, v) también es un vértice de $\Psi(\tau)$.
Un simplejo de $\Psi(\sigma)$ es de la forma $\{(v_j, v)|j \in J, v \in \{0, 1\}\}$, donde $J \subseteq \{0, ..., m\}$ y los v_j son distintos. Si $J \subseteq \{0, ..., m\}$, entonces también $J \subseteq \{0, ..., m, ..., n\}$, así que todo simplejo de $\Psi(\sigma)$ también es un simplejo de $\Psi(\tau)$. □

Proposición 2.7.5. *Si $\sigma = \{v_0, ..., v_n\}$, entonces $\Psi(\sigma) = \Psi(\sigma - \{v_i\}) * \Psi(\{v_i\})$, para cualquier $v_i \in \sigma$.*

Demostración. Los vértices de $\Psi(\sigma - \{v_i\}) * \Psi(\{v_i\})$ se conforman de la unión de los vértices de $\Psi(\sigma - \{v_i\})$ con los de $\Psi(\{v_i\})$, que son de la forma (v_j, v), $v_j \in \sigma - \{v_i\}$ con $v \in \{0, 1\}$, y (v_i, v), con $v \in \{0, 1\}$, respectivamente. Así que los vértices $\Psi(\sigma - \{v_i\}) * \Psi(\{v_i\})$ son los mismos que los de $\Psi(\sigma)$.
Ahora se probará que todo simplejo de $\Psi(\sigma - \{v_i\}) * \Psi(\{v_i\})$ es un simplejo de $\Psi(\sigma)$. De la proposición 2.7.4 se sigue que tanto los simplejos de $\Psi(\sigma - \{v_i\})$ como los de $\Psi(\{v_i\})$, son simplejos de $\Psi(\sigma)$. Por otra parte, note que $\Psi(\{v_i\}) = \big\{\emptyset, \{(v_i, 0)\}, \{(v_i, 1)\}\big\}$. Así que los simplejos de $\Psi(\sigma - \{v_i\}) * \Psi(\{v_i\})$ a los que falta considerar son los de la forma $\tau \cup \{(v_i, 0)\}, \tau \cup \{(v_i, 1)\}\big\}$, donde τ es un simplejo de $\Psi(\sigma - \{v_i\})$. Los simplejos anteriores también son simplejos de $\Psi(\sigma)$, pues τ es de la forma $\{(v_j, v)|j \in J', v \in \{0, 1\}\}$, donde $J' \subseteq \{0, ..., n\} - \{i\}$ y los v_j son distintos. De modo que $\tau \cup \{(v_i, 0)\}$ es un simplejo de la forma $\{(v_j, v)|j \in J' \cup \{i\}, v \in \{0, 1\}\}$, donde $J' \cup \{i\} \subseteq \{0, ..., n\}$ y los v_j son distintos, que es lo que se requiere para que un conjunto sea un simplejo de $\Psi(\sigma)$.
Resta ver que todo simplejo de $\Psi(\sigma)$ es también un simplejo de $\Psi(\sigma - \{v_i\}) * \Psi(\{v_i\})$. Los simplejos de $\Psi(\sigma)$ son de la forma $\{(v_j, v)|j \in J, v \in \{0, 1\}\}$ donde $J \subseteq \{0, ..., n\}$ y los v_j son distintos. Si $J \subseteq \{0, ..., n\} - \{i\}$ o $J = \{i\}$, entonces τ es un simplejo de $\Psi(\sigma - \{v_i\})$ o de $\Psi(\{v_i\})$, respectivamente; en cualesquiera de los casos, τ es un simplejo de $\Psi(\sigma - \{v_i\}) * \Psi(\{v_i\})$. Finalmente, si $i \in J$, pero $\{i\} \neq J$, entonces τ se puede considerar como la unión de dos simplejos τ_1, τ_2, que tienen la forma $\{(v_j, v)|j \in J', v \in \{0, 1\}\}$, donde

$J' \subseteq \{0, ..., n\} - \{i\}$ y los v_j son distintos, y $\{(v_i, v) | v \in \{0, 1\}\}$, donde v_i solo aparece una vez como primer coordenada, respectivamente. Como τ_1 es un simplejo de $\Psi(\sigma - \{v_i\})$ y τ_2 es un simplejo de $\Psi(\{v_i\})$, entonces $\tau = \tau_1 \cup \tau_2$ es un simplejo de $\Psi(\sigma - \{v_i\}) * \Psi(\{v_i\})$, como se quería demostrar.

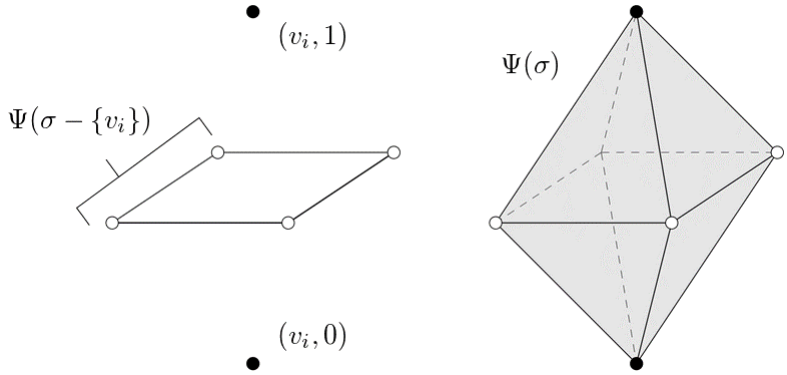

Figura 2.5: La psuedoesfera de dimensión n es el complejo unión de las pseudoesferas de dimensión $n - 1$ y dimensión 0.

□

Proposición 2.7.6. *Si σ es un simplejo de dimensión n, entonces $\Psi(\sigma)$ tiene 2^{n+1} facetas.*

Demostración. Por inducción sobre n.

Si $n = 0$ y $\sigma = \{v_0\}$, entonces $\Psi(\{v_0\}) = \big\{\emptyset, \{(v_0, 0)\}, \{(v_0, 1)\}\big\}$. En este caso, $\Psi(\sigma)$ tiene $2^{0+1} = 2$ facetas que son $\{(v_0, 0)\}$ y $\{(v_0, 1)\}$.

Suponga ahora que para cualquier simplejo α de dimensión n, $\Psi(\alpha)$ tiene 2^{n+1} facetas.

Sea $\sigma = \{v_0, ..., v_n, v_{n+1}\}$ un simplejo de dimensión $n + 1$. De la proposición 2.7.5 se sigue que $\Psi(\sigma) = \Psi(\sigma - \{v_{n+1}\}) * \Psi(\{v_{n+1}\})$. Al ser $\sigma - \{v_{n+1}\}$ un simplejo de dimensión n, se puede suponer que $\Psi(\sigma - \{v_{n+1}\})$ tiene 2^{n+1} facetas. Las facetas de $\Psi(\sigma - \{v_{n+1}\}) * \Psi(\{v_{n+1}\})$, según la proposición 2.7.2, son de la forma $\tau \cup \{(v_i, 0)\}$, $\tau \cup \{(v_i, 1)\}\big\}$, donde τ es una faceta de $\Psi(\sigma - \{v_{n+1}\})$, así que $\Psi(\sigma) = \Psi(\sigma - \{v_{n+1}\}) * \Psi(\{v_{n+1}\})$ tiene $2(2^{n+1}) = 2^{(n+1)+1}$ facetas, como se quería demostrar. □

Observación 2.7.7. Si $\mathcal{K}$ es un complejo simplicial, entonces $\Psi(\mathcal{K})$ denotará al complejo $\bigcup_{\tau \in \mathcal{K}} \Psi(\tau)$.

Proposición 2.7.8. *Si $\mathcal{K}$ es un complejo simplicial, entonces $\Psi(\mathcal{K}) = \bigcup_{\tau \in facet(\mathcal{K})} \Psi(\tau)$.*

Demostración. Esto se debe a que para cualesquiera simplejos τ_1, τ_2 tales que $\tau_1 \subseteq \tau_2$, se tiene que $\Psi(\tau_1) \subseteq \Psi(\tau_2)$(Proposición 2.7.4). Así que para cualquier simplejo σ de $\mathcal{K}$, existe $\tau \in facet(\mathcal{K})$ tal que $\sigma \subseteq \tau$, de modo que $\Psi(\sigma) \subseteq \Psi(\tau)$, lo que prueba que $\Psi(\mathcal{K}) = \bigcup_{\tau \in facet(\mathcal{K})} \Psi(\tau)$. □

La intersección de dos esferas combinatorias es también una esfera combinatoria.

Proposición 2.7.9 (Intersección de esferas combinatorias)**.** *Para cualesquiera simplejos σ y τ, se cumple que*

$$\Psi(\sigma \cap \tau) = \Psi(\sigma) \cap \Psi(\tau). \tag{2.3}$$

Demostración. Sean $\sigma = \{v_0, ..., v_n\}$ y $\tau = \{u_0, ..., u_m\}$. Suponga que $\sigma \cap \tau$ es un conjunto con $k+1$ elementos, se puede suponer sin perder generalidad que la intersección de los dos simplejos son los primeros $k+1$ vértices de cada simplejo, es decir, que $v_i = u_i$ para toda i en $\{0, ..., k\}$ y $\sigma \cap \tau = \{v_0, ..., v_k\}$.
La proposición 2.7.4 garantiza que $\Psi(\sigma \cap \tau) \subseteq \Psi(\sigma)$ y que $\Psi(\sigma \cap \tau) \subseteq \Psi(\tau)$, de lo que se sigue que $\Psi(\sigma \cap \tau) \subseteq \Psi(\sigma) \cap \Psi(\tau)$.

Ahora considere a un simplejo $\alpha \in \Psi(\sigma) \cap \Psi(\tau)$, α es de la forma $\alpha = \{(v_j, v) | j \in J, v \in \{0, 1\}\}$, donde $J \subseteq \{0, ..., n\}$ y los v_j son distintos; pero también es de la forma $\alpha' = \{(u_i, v) | i \in I, v \in \{0, 1\}\}$, donde $I \subseteq \{0, ..., m\}$ y los u_i son distintos. Como dichos complejos son iguales, se sigue que los vértices de $(v_j, v) \in \alpha$ también son vértices de α', lo que implica que los v_j son elementos de τ y por lo tanto $v_j \in \sigma \cap \tau$. Esto restringe el rango de valores que toma J, pues el que v_j esté en $\sigma \cap \tau$, implica que j esté en $\{0, ..., k\}$. Por lo que α es de la forma $\alpha = \{(v_j, v) | j \in J, v \in \{0, 1\}\}$, donde $J \subseteq \{0, ..., k\}$ y los v_j son distintos, y esta es precisamente la forma que tienen los simplejos de $\Psi(\sigma \cap \tau)$. Así $\alpha \in \Psi(\sigma \cap \tau)$.
Si la intersección de σ y τ es el conjunto vacío, entonces el único simplejo en común entre $\Psi(\sigma)$ y $\Psi(\tau)$ es el conjunto vacío, pues ya se mostró que para que un vértice de la forma (w, v) esté en $\Psi(\sigma) \cap \Psi(\tau)$, se tiene que cumplir que w esté en $\sigma \cap \tau$. □

Proposición 2.7.10. *Para cualquier simplejo σ, se cumple que $\Psi(\sigma)$ es un complejo shellable.*

Demostración. Por inducción sobre la dimensión de σ.

Caso $n = 1$. $\sigma = \{v_0, v_1\}$. Las facetas de $\Psi(\sigma)$, son:

$\phi_0 = \{(v_0, 0), (v_1, 0)\}$, $\phi_1 = \{(v_1, 0), (v_0, 1)\}$, $\phi_2 = \{(v_0, 1), (v_1, 1)\}$ y $\phi_3 = \{(v_1, 1), (v_0, 0)\}$.

En el orden en el que fueron listadas, las facetas forman una sucesión shellable para $\Psi(\sigma)$. La intersección de ϕ_0 y ϕ_1, es un vértice, que es de dimensión $dim(\phi_1) - 1 = 0$. Así también la intersección de ϕ_2 con la unión de ϕ_0 y ϕ_1. Finalmente, ϕ_3 intersecta a la unión de las facetas anteriores en un par de vértices, que es una unión de dos caras de dimensión $dim(\phi_3) - 1 = 0$, de ϕ_3, como lo requiere la definición de un complejo shellable (Definición 2.4.1).

Suponga ahora que para todo simplejo de dimensión $n - 1$, se cumple que $\Psi(\sigma)$ es un complejo shellable.

Sea σ un simplejo de dimensión n, con vértices $v_0, v_1, ..., v_n$. El complejo $\Psi(\sigma)$ se puede considerar como el complejo unión de $\Psi(\sigma - \{v_n\})$ y $\Psi(v_n)$, es decir $\Psi(\sigma) = \Psi(\sigma - \{v_n\}) * \Psi(v_n)$ (Proposición 2.7.5). Por hipótesis de inducción, el complejo $\Psi(\sigma - \{v_n\})$ es shellable. Sea $\phi_0, \phi_1, ..., \phi_t$ una sucesión shellable para dicho complejo. A partir de la sucesión anterior se definirá una sucesión shellable para $\Psi(\sigma)$. La sucesión que se propone es la siguiente:

$$\{(v_n, 0)\} \cup \phi_0, \{(v_n, 0)\} \cup \phi_1, ..., \{(v_n, 0)\} \cup \phi_t, \{(v_n, 1)\} \cup \phi_0, \{(v_n, 1)\} \cup \phi_1, ..., \{(v_n, 1)\} \cup \phi_t.$$

Observe que en la lista están todas las facetas de $\Psi(\sigma)$, esto se sigue de que $\Psi(\sigma) = \Psi(\sigma - \{v_n\}) * \Psi(v_n)$ y de la proposición 2.7.2. Ahora considere un elemento de la sucesión propuesta, éste puede ser de la forma $\{(v_n, 0)\} \cup \phi_k$ o $\{(v_n, 1)\} \cup \phi_k$.

En el primer caso, en el que la faceta seleccionada es de la forma $\{(v_n, 0)\} \cup \phi_k$, se tiene que

$$\begin{aligned}\Big(\bigcup_{i=0}^{k-1} \big(\{(v_n, 0)\} \cup \phi_i\big)\Big) \cap \big(\{(v_n, 0)\} \cup \phi_k\big) &= \bigcup_{i=0}^{k-1} \Big(\big(\{(v_n, 0)\} \cup \phi_i\big) \cap \big(\{(v_n, 0)\} \cup \phi_k\big)\Big) \\ &= \bigcup_{i=0}^{k-1} \Big(\{(v_n, 0)\} \cup (\phi_i \cap \phi_k)\Big)\end{aligned}$$

$$
\begin{aligned}
&= \{(v_n,0)\} \cup \Big(\bigcup_{i=0}^{k-1} (\phi_i \cap \phi_k) \Big) \\
&= \{(v_n,0)\} \cup \Big((\bigcup_{i=0}^{k-1} \phi_i) \cap \phi_k \Big) \\
&= \{(v_n,0)\} \cup \Big(\bigcup_{j=0}^{r_k} \phi_j \Big) (\phi_0, \phi_1, ..., \phi_t \text{ es una sucesión shellable}) \\
&= \bigcup_{j=0}^{r_k} \big(\{(v_n,0)\} \cup \phi_j \big).
\end{aligned}
$$

De lo que se concluye que

$$
\Big(\bigcup_{i=0}^{k-1} \big(\{(v_n,0)\} \cup \phi_i \big) \Big) \cap \big(\{(v_n,0)\} \cup \phi_k \big) = \bigcup_{j=0}^{r_k} \big(\{(v_n,0)\} \cup \phi_j \big),
$$

donde cada ϕ_j es una cara de dimensión $dim(\phi_k)-1$ de ϕ_k, por lo que cada $\{(v_n,0)\} \cup \phi_j$ es de dimensión $(dim(\phi_k)-1)+1 = dim(\phi_k)$, como la dimensión de $\{(v_n,0)\} \cup \phi_k$ es la dimensión de ϕ_k mas uno, entonces $\{(v_n,0)\} \cup \phi_j$ es una cara de dimensión $dim(\{(v_n,0)\} \cup \phi_k) - 1$ de $\{(v_n,0)\} \cup \phi_k$, como se tenía que demostrar.

Ahora bien, si la faceta por analizar es de la forma $\{(v_n,1)\} \cup \phi_k$, la intersección de $\{(v_n,1)\} \cup \phi_k$ con la unión de las facetas previas en la lista se puede dividir en dos partes: la primera en la que intersecta a todos los $\{(v_n,0)\} \cup \phi_i$, con $i \in \{0,1,...,t\}$; y la segunda, en la que intersecta a los $\{(v_n,1)\} \cup \phi_i$, con $0 \leq i < k$. De modo que dicha intersección queda descrita como

$$
\Big[\Big(\bigcup_{j=0}^{t} \big(\{(v_n,0)\} \cup \phi_j \big) \Big) \cap \big(\{(v_n,1)\} \cup \phi_k \big) \Big] \cup \Big[\Big(\bigcup_{i=0}^{k-1} \big(\{(v_n,1)\} \cup \phi_i \big) \Big) \cap \big(\{(v_n,1)\} \cup \phi_k \big) \Big]. \quad (2.4)
$$

Note que $\big(\{(v_n,0)\} \cup \phi_j \big) \cap \big(\{(v_n,1)\} \cup \phi_k \big) = \phi_j \cap \phi_k$, para $j \in \{0,1,...,t\}$, por lo que el primer uniendo de la expresión 2.4 es igual a $\bigcup_{j=0}^{t} \phi_j \cap \phi_k$, y este último término es igual a ϕ_k, pues para $j = k$, $\phi_j \cap \phi_k = \phi_k$. La unión de los $\phi_j \cap \phi_k$ está contenida en ϕ_k, pero a la vez ϕ_k es uno de los conjuntos que están siendo unidos, esto sólo es posible si dicha unión es igual a ϕ_k, que a su vez, ϕ_k es una cara de dimensión $dim(\{(v_n,1)\} \cup \phi_k) - 1$ de $\{(v_n,1)\} \cup \phi_k$.

Para el segundo uniendo de la expresión 2.4, el razonamiento es similar al que se hizo anteriormente al analizar las facetas de la forma $\{(v_n,0)\}\cup\phi_k$ (considerando $k>0$), es decir, se puede afirmar que

$$\Big(\bigcup_{i=0}^{k-1}\big(\{(v_n,1)\}\cup\phi_i\big)\Big)\cap\big(\{(v_n,1)\}\cup\phi_k\big)=\bigcup_{l=0}^{s_k}\big(\{(v_n,1)\}\cup\phi_l\big),$$

donde ϕ_l es una cara de dimensión $dim(\phi_k)-1$ de ϕ_k y cada $\{(v_n,1)\}\cup\phi_l$ es una cara de dimensión $dim(\{(v_n,1)\}\cup\phi_l)-1$ de $\{(v_n,1)\}\cup\phi_k$. Del análisis anterior se deduce que la expresión 2.4 se puede reescribir como

$$\phi_k\cup\big(\bigcup_{l=0}^{s_k}\big(\{(v_n,1)\}\cup\phi_l\big)\big),$$

donde cada uno de los uniendos es una cara de dimensión $dim(\{(v_n,1)\}\cup\phi_k)-1$ de $\{(v_n,1)\}\cup\phi_k$, que es lo que se quería demostrar.

Si $k=0$, la intersección $\{(v_n,1)\}\cup\phi_k$ con la unión de las facetas anteriores es ϕ_k, que es una cara de dimensión $dim(\{(v_n,1)\}\cup\phi_k)-1$ de $\{(v_n,1)\}\cup\phi_k$. Se ha probado entonces que la sucesión propuesta es, en efecto, una sucesión shellable para $\Psi(\sigma)$. □

Lema 2.7.11. *Si $\mathcal{K}$ es unión de caras de dimensión $n-1$ de un simplejo σ de dimensión n y $v\notin V(\mathcal{K})$, entonces $K*\{v\}$ es $(n-1)-$conexo.*

Demostración. La prueba es por inducción sobre la dimensión de σ.

Si la dimensión de σ es 1, entonces σ es una arista que consta de dos vértices, digamos $\sigma=\{v_0,v_1\}$, si $\mathcal{K}$ es unión de caras de dimensión 0, entonces $\mathcal{K}$ consta de uno o dos vértices. Si $\mathcal{K}$ consta de un vértice, entonces $\mathcal{K}*\{v\}$ es una arista, y ésta es $0-$conexa; por otro lado, si $\mathcal{K}$ consta de dos vértices, entonces $\mathcal{K}*\{v\}$ es la unión de las aristas $\{v,v_0\}$ y $\{v,v_1\}$, como la intersección de estas aristas es no vacía, entonces $\mathcal{K}*\{v\}$ es $0-$conexo.

Suponga que si τ es un simplejo de dimensión n y $\mathcal{L}$ es un complejo que es unión de caras de dimensión $n-1$ de τ, tal que $v\notin V(\mathcal{L})$, entonces el complejo $\mathcal{L}*\{v\}$ es $(n-1)-$conexo.

Sea σ un simplejo de dimensión $n+1$ y $\mathcal{K}$ un complejo que es unión de caras de dimensión n de σ, hay que probar que $\mathcal{K}*\{v\}$ es $n-$conexo, para ello considere el siguiente enunciado:

Si $\phi_0, ..., \phi_t$ son caras de dimensión n de σ entonces $\left(\bigcup_{i=0}^{t} \phi_i\right) * \{v\}$ es $n-$conexo.

El enunciado anterior se prueba por inducción sobre t, con la consideración de que el número de caras de cualquier simplejo es finito.

Si $t = 0$, entonces $\phi_0 * \{v\}$ es un simplejo de dimensión $n+1$, que según la observación 2.5.4, es $n-$conexo. Suponga entonces que si $\phi_0, ..., \phi_t$ son caras de dimensión n de σ, entonces $\left(\bigcup_{i=0}^{t} \phi_i\right) * \{v\}$ es $n-$conexo.

Ahora considere a $\phi_0, ..., \phi_{t+1}$ caras de dimensión n de σ; por hipótesis de inducción (sobre t), se tiene que $\left(\bigcup_{i=0}^{t} \phi_i\right) * \{v\}$ es $n-$conexo, además, usando el mismo argumento que en el caso base, se tiene que $\phi_{t+1} * \{v\}$ también es $n-$conexo. La intersección de estos complejos es

$$\left(\left(\bigcup_{i=0}^{t} \phi_i \right) \cap \phi_{t+1} \right) * \{v\}. \tag{2.5}$$

Recuerde que en la prueba de la proposición 2.4.2 se demuestra que si $\phi_0, ..., \phi_{t+1}$ son caras de dimensión $\dim(\sigma) - 1$ de σ, entonces cualquier orden que se de a dichas facetas es un orden shellable. Así pues, el complejo $\left(\bigcup_{i=0}^{t} \phi_i\right) \cap \phi_{t+1}$, se puede describir como unión de caras de dimensión $n-1$ de ϕ_{k+1}, digamos $\bigcup_{j=1}^{s} \tau_j$. Así que el complejo de la expresión 2.5 se puede describir como

$$\left(\bigcup_{j=1}^{s} \tau_j \right) * \{v\}.$$

Usando la hipótesis de inducción (sobre n) se tiene que este último complejo es $(n-1)-$conexo. Teniendo pues que tanto $\left(\bigcup_{i=0}^{t} \phi_i\right) * \{v\}$ como $\phi_{t+1} * \{v\}$ son $n-$conexos y su intersección es $(n-1)-$conexo, se puede concluir a partir del corolario 2.5.8 del Lema del Nervio, que la unión de dichos complejos es $n-$conexo, o lo que es lo mismo, $\left(\bigcup_{i=0}^{t+1} \phi_i\right) * \{v\}$ es $n-$conexo. De lo anterior se concluye que $\mathcal{K} * \{v\}$ es $n-$conexo, que es lo que se tenía que demostrar. □

Proposición 2.7.12. *Si σ es un simplejo de dimensión n, entonces $\Psi(\sigma)$ es $(n-1)-$conexo.*

Demostración. La prueba es por inducción sobre la dimensión de σ y el enunciado que se probará es el siguiente:

Si $\sigma = \{v_0, ..., v_n\}$ es un simplejo de dimensión n, entonces los complejos $\Psi(\{v_0, ..., v_n\} - \{v_i\}) * \{(v_i, 0)\}$ y $\Psi(\{v_0, ..., v_n\} - \{v_i\}) * \{(v_i, 1)\}$ son $(n-1)-$conexos y $\Psi(\{v_0, ..., v_n\})$ es $(n-1)-$conexo, esto último quiere decir que la pseudoesfera de un simplejo de dimensión n es un complejo $(n-1)-$conexo.

Para $n = 1$ e $i = 1$ (sin perder generalidad), se tiene que $\Psi(\{v_0\}) * \{(v_1, 0)\}$ consta de las aristas $\{(v_0, 0), (v_1, 0)\}$ y $\{(v_0, 1), (v_1, 0)\}$, con sus respectivos vértices, esta gráfica es conexa pues $(v_1, 0)$ es un vértice común, esto prueba que $\Psi(\{v_0\}) * \{(v_1, 0)\}$ y $\Psi(\{v_0\}) * \{(v_1, 1)\}$ son $1 - 1 = 0-$conexos; por otro lado su intersección es $\Psi(\{v_0\})$ y consta de dos vértices $(v_0, 0)$ y $(v_0, 1)$, por lo que es no vacío o $-1-$conexo, así que la unión de $\Psi(\{v_0\}) * \{(v_1, 0)\}$ y $\Psi(\{v_0\}) * \{(v_1, 1)\}$ es una gráfica conexa, que es $\Psi(\{v_0, v_1\})$, como se quería demostrar. Ahora suponga por inducción que el enunciado en cuestión es válido para dimensión n. Sea $\sigma = \{v_0, ..., v_n, v_{n+1}\}$ es un simplejo de dimensión $n + 1$, hay que probar que $\Psi(\{v_0, ..., v_{n+1}\} - \{v_i\}) * \{(v_i, 0)\}$ y $\Psi(\{v_0, ..., v_{n+1}\} - \{v_i\}) * \{(v_i, 1)\}$ son $n-$conexos y $\Psi(\{v_0, ..., v_{n+1}\})$ es $n-$conexo, es decir, que la pseudoesfera de dimensión $n + 1$ es $n-$conexa.

Note que por hipótesis de inducción el complejo $\Psi(\{v_0, ..., v_{n+1}\} - \{v_i\})$ es $(n-1)-$conexo. Ahora se probará que $\Psi(\{v_0, ..., v_{n+1}\} - \{v_i\}) * \{(v_i, 0)\}$ es $n-$conexo, para ello considere a $\phi_0, ..., \phi_t$ un orden shellable para $\Psi(\{v_0, ..., v_{n+1}\} - \{v_i\})$. Para toda $k \leq t$ (inducción sobre k) se probará que $\bigcup_{i=0}^{k} \phi_i$ es tal que $\big(\bigcup_{i=0}^{k} \phi_i\big) * \{(v_i, 0)\}$ es $(n-1)-$conexo, comenzando con $k = 0$, se tiene que $\phi_0 * \{(v_i, 0)\}$ es un simplejo de dimensión $n + 1$, pues ϕ_0 es de dimensión n, así que según la observación 2.5.4, $\phi_0 * \{(v_i, 0)\}$ es $n-$conexo. Ahora suponga que si $0 < k < t$, entonces $\big(\bigcup_{i=0}^{k} \phi_i\big) * \{(v_i, 0)\}$ es $n-$conexo, para $k + 1$ se tiene que

$$\big(\bigcup_{i=0}^{k+1} \phi_i\big) * \{(v_i, 0)\} = \big(\big(\bigcup_{i=0}^{k} \phi_i\big) * \{(v_i, 0)\}\big) \cup \big(\phi_{k+1} * \{(v_i, 0)\}\big).$$

Ya se tiene que tanto $\big(\bigcup_{i=0}^{k} \phi_i\big) * \{(v_i, 0)\}$ como $\phi_{k+1} * \{(v_i, 0)\}$ son $n-$conexos, ahora hay que ver que su intersección es $(n-1)-$conexa para poder usar el corolario 2.5.8 del Lema del Nervio y así concluir que $\big(\bigcup_{i=0}^{k+1} \phi_i\big) * \{(v_i, 0)\}$ es $n-$conexo, dicha intersección es

el complejo $\left(\left(\bigcup_{i=0}^{k}\phi_i\right)\cap\phi_{k+1}\right)*\{(v_i,0)\}=\mathcal{K}*\{(v_i,0)\}$, donde $\mathcal{K}$ es una unión de caras de dimensión $n-1$ de ϕ_{k+1} (pues $\phi_0,...,\phi_t$ es una sucesión shellable), por lo que según el lema 2.7.11, el complejo $\mathcal{K}*\{(v_i,0)\}$ es $(n-1)-$conexo, como se quería probar. De lo anterior se concluye que $\left(\bigcup_{i=0}^{t}\phi_i\right)*\{(v_i,0)\}$ es $n-$conexo, es decir $\Psi(\{v_0,...,v_{n+1}\}-\{v_i\})*(v_i,0)$ es $n-$conexo y así también $\Psi(\{v_0,...,v_{n+1}\}-\{v_i\})*(v_i,1)$, finalmente para calcular la conexidad de la pseudoesfera de dimensión $n+1$, $\Psi(\{v_0,...,v_{n+1}\})$ solo hay que recordar que dicho complejo es la unión de $\Psi(\{v_0,...,v_{n+1}\}-\{v_i\})*\{(v_i,0)\}$ y de $\Psi(\{v_0,...,v_{n+1}\}-\{v_i\})*\{(v_i,1)\}$, además la intersección de dichos complejos es la pseudoesfera de dimensión n, $\Psi(\{v_0,...,v_n\})$ que por hipótesis de inducción es $(n-1)-$conexa, por lo que usando el corolario 2.5.8 del Lema del Nervio se puede concluir que $\Psi(\{v_0,...,v_{n+1}\})$ es $n-$conexa.

□

Capítulo 3

Tareas y Protocolos

En este capítulo se describe la relación que existe entre la topología combinatoria y el estudio de problemas de cómputo distribuido, dicha relación se expresa en términos de dos modelos, un modelo operacional y un modelo combinatorio. Al final del capítulo se presenta una prueba de la imposibilidad del consenso binario en el modelo iterado de instantáneas inmediatas, dicha prueba, aunque elemental, permite entender que los conceptos presentados a lo largo del capítulo son bastante coherentes y brindan una perspectiva diferente al estudiar un problema de cómputo distribuido como lo es el consenso binario, pues permite estudiar de forma estática los distintos escenarios posibles que se suscitan después de que cada proceso ejecuta el protocolo de comunicación.

3.1. Modelo operacional

Una tarea de decisión, o simplemente tarea, es un problema distribuido en el que cada proceso que participa comienza con un valor de entrada y después de ejecutar un protocolo de comunicación decide un valor de salida. En el modelo de cómputo que se estudia a lo largo de este texto se asume que la comunicación entre los procesos es través de una memoria compartida a lo largo de varias rondas de comunicación, en dicha memoria los procesos escriben su estado local que representa el conocimiento que cada proceso tiene del sistema; para conocer el estado de los demás, es necesario que cada proceso lea lo que

hay en la memoria compartida, sin embargo, es posible que los procesos no se comuniquen al mismo tiempo y la comunicación no sea inmediata, de modo que algunos procesos no puedan aprender el estado de los demás procesos en una ronda, esto debido a la *asincronía* del sistema: cada proceso ejecuta instrucciones a cierta velocidad que puede variar y que es independiente de las velocidades de los otros procesos, esta característica impide que los procesos puedan distinguir si un proceso ha fallado o simplemente aún no ha comenzado a ejecutar instrucciones; con respecto a las fallas, se supondrá que si un proceso falla es porque su ejecución se detiene y termina de forma silenciosa.

La memoria compartida se representa con un arreglo y los procesos acceden a ella usando instantáneas inmediatas a lo largo de varias rondas de comunicación; existen distintas formas para tener acceso a la memoria compartida, pero las variaciones más comunes a las instantáneas inmediatas son equivalentes a dicho modelo de cómputo, en el sentido que los resultados sobre computabilidad que sean válidos en el modelo de instantáneas inmediatas también lo serán en los modelos equivalentes, y viceversa [4, 9]. La ventaja de usar instantáneas inmediatas será clara al presentar el modelo combinatorio que representa al protocolo de comunicación, pues dicha representación será un complejo simplicial que resulta ser sencillo de describir, permitiendo así centrar la atención en los aspectos claves (como la asincronía) de los sistemas distribuidos al estudiar la clase de problemas que se pueden resolver con ellos.

3.1.1. Procesos

Cada proceso tiene un único *nombre*, tomado de un universo de nombres Π. Al proceso con nombre $P \in \Pi$ se le denominará como " el proceso P ". El proceso P tiene un conjunto de estados que incluye un conjunto de *estados iniciales* Q^{in} y un conjunto de *estados finales* Q^{fin}. Cada estado q de un proceso incluye un componente *nombre* tomado de Π, que es inmutable y que será denotado por $\mathsf{nombre(q)}$. Si el proceso cambia del estado q al estado $\mathsf{q'}$ en una ejecución, entonces $\mathsf{nombre(q)} = \mathsf{nombre(q')}$.

Además, cada estado de un proceso incluye un componente *vista*, denotado por $\mathsf{vista(q)}$, que típicamente cambia de un estado a otro en una ejecución. Este componente representa lo que cada proceso conoce sobre la ejecución en curso.

Como cada estado queda determinado por su nombre y su vista, el estado q se puede escribir como la pareja (P, v), donde $\mathsf{nombre}(\mathsf{q}) = P$ y $\mathsf{vista}(\mathsf{q}) = v$.

3.1.2. Configuraciones y ejecuciones

Una *configuración* C es un conjunto de estados de procesos correspondiente al estado del sistema en cierto momento. Cada proceso aparece a lo más una vez en cada configuración; si s_0, s_1 son estados distintos en C_i, entonces $\mathsf{nombre}(s_0) \neq \mathsf{nombre}(s_1)$. Una *configuración inicial* C_0 es una configuración en la cual el estado de cada proceso es un estado inicial, mientras que una *configuración final* es aquella en la que el estado de cada proceso es un estado final. Una *ejecución* define el orden en que los procesos se comunican; formalmente se puede definir como una sucesión alternante de configuraciones y conjuntos de nombres de procesos

$$C_0, S_0, C_1, S_1, ..., S_r, C_{r+1},$$

que satisface las siguientes condiciones:

· C_0 es la configuración inicial.

· S_i es el conjunto de nombres de los procesos que cambiaron su estado entre la configuración C_i y la configuración C_{i+1}.

A una terna de la forma C_i, S_i, C_{i+1} se le denominará *paso concurrente*. Si $P \in S_i$, se dice que P *realiza un paso*. Por un paso se entenderá una instantánea inmediata, concepto que se explicará más adelante.

Una configuración final τ es *accesible* desde una configuración inicial σ, si existe una ejecución $C_0, S_0, C_1, S_1, ..., S_r, C_{r+1}$, donde σ corresponde a C_0 y C_{r+1} corresponde a τ.

Si la última configuración de una ejecución no es una configuración final, porque ésta incluye procesos cuyos estados no son finales, entonces se considera que dichos procesos han fallado.

3.1.3. Tareas

Una tarea especifica qué combinaciones de valores de entrada pueden ser asignados a los procesos, a cada proceso se le asigna un valor de un conjunto de *valores de entrada* V^{in}. De forma más precisa, una *asignación de entrada* para un conjunto de procesos Π es un conjunto de pares $\{(P_j, v_j)|P_j \in \Pi, v_j \in V^{in}\}$, donde cada proceso P_j aparece una sola vez, pero los valores v_j no son necesariamente distintos. De la misma forma, una tarea especifica qué combinaciones de valores de salida pueden ser elegidos por los procesos. Cada proceso elige un valor de un conjunto de *valores de salida* V^{out}. De forma similar a una asignación de entrada se puede definir una *asignación de salida* para un conjunto de procesos Π. Informalmente, una tarea está dada por un conjunto de asignaciones de entrada $\mathcal{I}$, un conjunto de asignaciones de salida $\mathcal{O}$, y una relación Δ, que especifica, para cada asignación de entrada, las asignaciones de salida que pueden ser elegidas.

Definición 3.1.1. Una *tarea* es una terna $(\mathcal{I}, \mathcal{O}, \Delta)$, donde:

- $\mathcal{I}$ es un conjunto de asignaciones de entrada,
- $\mathcal{O}$ es un conjunto de asignaciones de salida,
- $\Delta : \mathcal{I} \to 2^{\mathcal{O}}$ es un mapeo que envía a cada asignación de entrada a un conjunto de asignaciones de salida.

Consenso

La tarea del *consenso* consiste en lograr que $n + 1$ procesos en un sistema distribuido elijan un mismo valor de salida. Cada proceso tiene un valor de entrada que toma de cierto rango de valores y eventualmente debe decidir un valor de salida del mismo rango de valores; la decisión de elegir un valor es irrevocable. Cuando el rango de valores de entrada es el conjunto $\{0, 1\}$ el problema se denomina *consenso binario*. Un algoritmo distribuido correcto para el consenso binario debe cumplir las siguientes condiciones:

1. *Acuerdo.* Todos los procesos que deciden un valor, eligen el mismo valor.
2. *Terminación.* Todos los procesos que no fallan deciden eventualmente.

3. *Validez.* El valor común que los procesos deciden es el valor de entrada de algún proceso.

La definición del consenso recién presentada se puede describir conforme a la definición 3.1.1, por simplicidad, se hará el análisis para el consenso binario. Considere una asignación de entrada en la que $n+1$ procesos tienen el mismo valor de entrada y que este es 0, entonces las únicas asignaciones de salida posibles son aquellas en las que todos los procesos tienen como valor de salida al 0. Si $v_0, ..., v_n$, son los procesos en cuestión, entonces la asignación de entrada es el conjunto de parejas $\{(v_0, 0), ..., (v_n, 0)\}$, y las asignaciones de salida válidas para dicha asignación de entrada son de la forma $\{(v_{i_0}, 0), ..., (v_{i_k}, 0)\}$, donde $v_{i_0}, ..., v_{i_k}$ son los procesos que no fallaron, con eso queda descrito $\Delta(\{(v_0, 0), ..., (v_n, 0)\})$. El análisis es similar si el único valor de entrada es 1.

Ahora considere una asignación de entrada en la que aparecen como valores de entrada tanto 0 como 1, entonces $\{(v_{i_0}, 0), ..., (v_{i_k}, 0)\}$ y $\{(v_{i_0}, 1), ..., (v_{i_k}, 1)\}$ son asignaciones de salida válidas si $v_{i_0}, ..., v_{i_k}$ son los procesos que no fallaron y entre los valores de entrada de dichos procesos aparecen tanto el 0 como el 1, lo que describe a la imagen de la asignación de entrada en cuestión bajo Δ.

3.1.4. Protocolos

Un protocolo es un programa que resuelve una tarea. Se considerarán protocolos que pueden ser divididos en dos partes: una parte independiente de la tarea que el protocolo resuelve y otra que es dependiente de la misma. En la parte del protocolo que es independiente de la tarea, cada proceso comunica repetidamente su vista (se asume que cada proceso comparte toda la información que corresponde a su estado) a los demás, obtiene la vista de los otros procesos y actualiza su propio estado para reflejar lo que ha aprendido. Cuando han sucedido suficientes rondas de comunicación, cada proceso selecciona un valor de salida que resulta de aplicar un mapeo, que depende de la tarea en cuestión, a su vista final. De forma específica, en la parte independiente de la tarea, cada proceso ejecuta el *protocolo iterado de instantáneas inmediatas* (a veces sólo se dirá *protocolo iterado*), cuyo pseudocódigo se muestra a continuación:

Algoritmo 3.1 Protocolo iterado de instantáneas inmediatas.

```
shared mem : array[0...N-1][0...n]                    ▷ Hay n + 1 procesos
procedure protocoloGenérico(v_i)                      ▷ Ejecutado por el proceso P_i
    vista := v_i
    for l := 0 to N − 1 do                            ▷ Hay N iteraciones o rondas
        mem[l][i] := vista
        vista:= snapshot(mem[l][*])
    end for
    return δ(vista)
end procedure
```

Para reflejar la estructura multironda de los protocolos, la memoria se representa mediante un arreglo bidimensional $\mathsf{mem}[l][i]$, en el que la fila l es compartida únicamente entre los procesos que participan en la ronda l, y en la columna i, P_i es el único que puede escribir. Las entradas del arreglo $\mathsf{mem}[l][i]$ se suponen inicializadas en algún valor predeterminado, usualmente $\perp$. En la ronda l, el proceso P_i ejecuta una instantánea inmediata: P_i escribe su vista actual *vista* en $\mathsf{mem}[l][i]$ e inmediatamente después toma una instantánea de las entradas $\mathsf{mem}[l][*]$ que corresponden a la ronda l. Después de completar todas las rondas, P_i decide un valor al aplicar un mapeo de decisión determinista a su vista final.

Es importante aclarar que se asumirá que cada proceso tomará una instantánea inmediatamente después de escribir en la memoria compartida, esto quiere decir que si el proceso P_i ejecuta la asignación $\mathsf{mem}[l][i] := \mathsf{vista}$, ningún otro proceso que ejecute el protocolo podrá escribir en su entrada correspondiente de la memoria antes que el proceso P_i tome una instantánea de la memoria $\mathsf{snap}:= \mathsf{snapshot}(\mathsf{mem}[l][*])$. Sin embargo, puede suceder que algún otro proceso que ejecute el protocolo, digamos P_j, ejecute la asignación $\mathsf{mem}[l][j] := \mathsf{vista}$, de forma simultánea con P_i, en dicho caso los dos procesos obtendrán la misma vista al ejecutar la sentencia $\mathsf{snap}:= \mathsf{snapshot}(\mathsf{mem}[l][*])$. Esta consideración hará más sencillo el análisis del modelo combinatorio que se presentará más adelante.

A continuación se muestran algunas configuraciones finales que se pueden obtener a partir de ejecuciones del protocolo iterado de una ronda en el que participan los procesos P, Q y R, con configuración inicial $\{(P,p),(Q,q),(R,r)\}$. Los vértices de la forma $(S,\perp)$ se omiten de la vista de los procesos.

Suponga que la ejecución se da de forma totalmente secuencial, de manera que los tres procesos ejecutan uno tras otro el protocolo iterado de una ronda. Si P es el primero en ejecutar el protocolo, Q el segundo y R el tercero, entonces la configuración final obtenida es:

$$\big\{\big(P,\{(P,p)\}\big),\big(Q,\{(P,p),(Q,q)\}\big),\big(R,\{(P,p),(Q,q),(R,r)\}\big)\big\}.$$

Ahora suponga que P y Q ejecutan primero el protocolo de forma simultánea. Los dos obtienen la misma vista $\{(P,p),(Q,q)\}$. Después de P y Q, R ejecuta el protocolo y obtiene la vista $\{(P,p),(Q,q),(R,r)\}$. Esta ejecución resulta en la configuración final

$$\big\{\big(P,\{(P,p),(Q,q)\}\big),\big(Q,\{(P,p),(Q,q)\}\big),\big(R,\{(P,p),(Q,q),(R,r)\}\big)\big\}.$$

Considere a $\sigma = \{(P,p),(Q,q),(R,r)\}$, y

$$\tau = \big\{\big(P,\{(P,p)\}\big),\big(Q,\{(P,p),(Q,q),(R,r)\}\big),\big(R,\{(P,p),(Q,q),(R,r)\}\big)\big\};$$

en esta situación τ es accesible desde σ a través de una ejecución en la P es el primero en ejecutar el protocolo y después Q y R lo ejecutan de forma simultánea.
Dado un conjunto de asignaciones de entrada $\mathcal{I}$ para el Algoritmo 3.1, se denotará por $\mathcal{P}$ al conjunto de configuraciones finales accesibles desde las configuraciones iniciales definidas por $\mathcal{I}$. Así que si σ es una asignación de entrada de $\mathcal{I}$, en $\mathcal{P}$ están todas las configuraciones finales que son accesibles desde σ, dichas configuraciones serán denotadas por $\Lambda(\sigma)$. La terna $(\mathcal{I},\mathcal{P},\Lambda)$ se denominará *protocolo*. Los protocolos y las tareas están ligados de la siguiente manera: los procesos eligen su valor de salida usando un *mapeo de decisión* δ, que manda a cada proceso con su respectiva vista a un valor de salida. Se dice que el proceso P *elige* o *decide* el valor u, a partir de su vista final v, si $\delta(P,v) = (P,u)$. El mapeo δ se extiende de forma natural a las configuraciones finales de la siguiente forma: $\delta(C) = \{\delta(P,v)|(P,v) \in C\}$. Un protocolo $(\mathcal{I},\mathcal{P},\Lambda)$, con un mapeo de desición δ, *resuelve* una tarea $(\mathcal{I},\mathcal{O},\Delta)$, si para cada asignación de entrada $\sigma \in \mathcal{I}$ y cada configuración final $\tau \in \mathcal{P}$ accesible desde σ, es decir, tal que $\tau \in \Lambda(\sigma)$, se cumple que $\delta(\tau)$ es una asignación de salida O en $\mathcal{O}$ permitida por la especificación del problema, lo que equivale a que $O \in \Delta(\sigma)$.

3.2. Modelo combinatorio

3.2.1. Tareas

La siguiente definición tiene el objetivo de modelar, en términos combinatorios, la noción de tarea establecida en la definición 3.1.1.

Considere un sistema distribuido con $n+1$ procesos con nombres tomados de un conjunto Π, V^{in} es el dominio de los valores de entrada y V^{out} el dominio de valores de salida.

Definición 3.2.1. (Tarea) Una *tarea* es una terna $(\mathcal{I}, \mathcal{O}, \Delta)$, donde:

- $\mathcal{I}$ es un complejo cromático puro, coloreado por Π y etiquetado por V^{in}, tal que cada vértice es identificado únicamente por su color junto con su etiqueta.
- $\mathcal{O}$ es un complejo cromático puro, coloreado por Π y etiquetado por V^{out}, tal que cada vértice es identificado únicamente por su color junto con su etiqueta.
- Δ es un mapeo portador de $\mathcal{I}$ en $\mathcal{O}$ que preserva nombres (cromático).

Si v es un vértice, $\mathsf{nombre}(v)$ denotará el color de v (que usualmente será el nombre de un proceso) y $\mathsf{vista}(v)$ denotará su etiqueta (que usualmente será la vista de ese proceso). Para identificar de manera única a cada vértice de $\mathcal{I}$ y de $\mathcal{O}$ a través de su color junto con etiqueta basta pedir que las funciones $(\mathsf{nombre},\mathsf{vista}) : V(\mathcal{I}) \to \Pi \times V^{in}$ y $(\mathsf{nombre},\mathsf{vista}) : V(\mathcal{O}) \to \Pi \times V^{out}$ sean inyectivas.

Recuerde que según la definición 3.1.1, dada una tarea $(\mathcal{I}, \mathcal{O}, \Delta)$, los conjuntos $\mathcal{I}$ y $\mathcal{O}$ se conforman de asignaciones de entrada y de salida, respectivamente. Cada asignación de entrada o de salida es un conjunto de parejas que puede ser visto como un simplejo. La definición combinatoria de tarea pide que tanto $\mathcal{I}$ como $\mathcal{O}$ sean complejos simpliciales, las razones de este requerimiento se explican ahora. Suponga que σ es una asignación de entrada de $\mathcal{I}$ y que $\sigma' \subseteq \sigma$; pedir que σ' esté en $\mathcal{I}$, rescata la idea de que debe ser posible para los procesos que aparecen en σ' elegir un valor de salida aún cuando el resto de procesos que están en σ fallen antes de ejecutar cualquier paso (se dice que estos procesos no participan).

De esta manera los procesos que participan en σ' se ejecutarán como si la configuración inicial hubiera sido σ'.

Consenso

En la tarea del *consenso*, cada proceso inicia con un valor de entrada. Todos los procesos deben coincidir en un mismo valor de salida, que debe ser el valor de entrada de algún proceso. En el *consenso binario* los valores de entrada pueden ser 0 o 1. Formalmente, hay $n+1$ procesos. El complejo de entrada $\mathcal{I}$ tiene vértices de la forma (P, v), donde $P \in \Pi$, $v \in \{0, 1\}$. Además, para cualquier subconjunto $S \subseteq \Pi$, $S = \{P_0, ..., P_l\}$, y cualquier colección de valores $v_0, ..., v_l$ tomados de $\{0, 1\}$, los vértices $(P_0, v_0), ..., (P_l, v_l)$ forman un $l - simplejo$ de $\mathcal{I}$, y esos son precisamente todos los simplejos de $\mathcal{I}$.

El complejo de salida $\mathcal{O}$ para el consenso binario consiste de dos $n-$simplejos ajenos. Un simplejo tiene $n+1$ vértices de la forma $(P, 0)$, con $P \in \Pi$, incluyendo a todas sus caras, y el otro tiene $n+1$ vértices de la forma $(P, 1)$, con $P \in \Pi$, incluyendo a todas sus caras. Este complejo es disconexo, con dos componentes conexas. Finalmente, el mapeo portador $\Delta : \mathcal{I} \to 2^{\mathcal{O}}$ se describe a continuación. Para un simplejo $\sigma = \{(P_0, v_0), ..., (P_l, v_l)\}$ de $\mathcal{I}$, el subcomplejo $\Delta(\sigma)$ se define con las siguientes reglas:

1. Si $v_0 = ... = v_l = 0$, entonces $\Delta(\sigma)$ tiene al $l-$simplejo cuyos vértices son $(P_0, 0), ..., (P_l, 0)$, y todas sus caras.

2. Si $v_0 = ... = v_l = 1$, entonces $\Delta(\sigma)$ tiene al $l-$simplejo cuyos vértices son $(P_0, 1), ..., (P_l, 1)$, y todas sus caras.

3. Si $\{v_0, ..., v_l\}$ tiene al 0 y al 1, entonces $\Delta(\sigma)$ tiene dos $l-$simplejos que son ajenos, uno tiene vértices $(P_0, 0), ..., (P_l, 0)$, y el otro tiene vértices $(P_0, 1), ..., (P_l, 1)$, junto con todas sus caras.

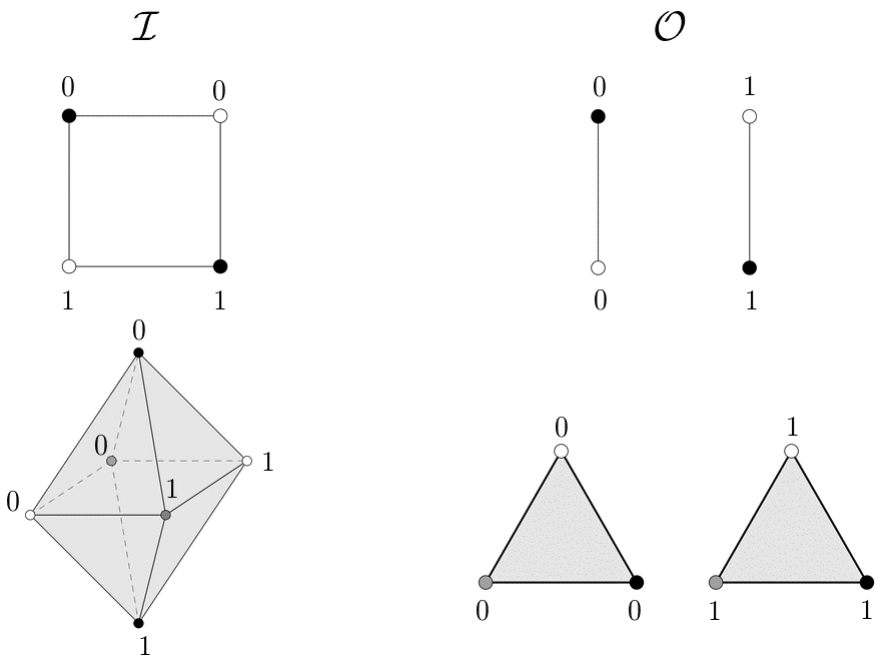

Figura 3.1: Complejos de entrada y de salida de dos y tres procesos para el consenso binario.

Hay que verificar que Δ es un mapeo portador cromático. Suponga $\sigma', \sigma \in \mathcal{I}$, son tales que $\sigma' \subseteq \sigma$. En el caso en el que el conjunto valores de entrada de los procesos de σ sea el conjunto $\{0\}$, entonces $\Delta(\sigma)$ es el complejo formado por σ y todas sus caras. Teniendo que σ' una cara de σ, se deduce que el conjunto de valores de entrada de los procesos que aparecen en σ' es también $\{0\}$. La definición de la tarea indica que $\Delta(\sigma')$ es el complejo formado por σ' y todas sus caras, este complejo está contenido en $\Delta(\sigma)$, pues σ' es una cara de σ y todas las caras de σ' son también caras de σ. El análisis es el mismo cuando el conjunto de valores de entrada es $\{1\}$. Ahora bien, si el conjunto de valores de entrada de los procesos que aparecen en σ es $\{0, 1\}$, entonces, sin importar los valores de entrada que aparezcan en σ', el complejo $\Delta(\sigma')$ consta de simplejos cuyos vértices toman nombres de σ' pero que tienen la misma etiqueta, ya sea 0 o 1; en cualesquiera de los casos, estos simplejos también son caras de un simplejo cuyos vértices tienen los nombres que aparecen en σ y que tienen la misma etiqueta, ya sea 0 o 1, por lo que los simplejos de $\Delta(\sigma')$ son también simplejos de $\Delta(\sigma)$. Lo anterior prueba que Δ es monótono. El hecho que Δ es rígido y cromático se sigue inmediatamente de la definición.

3.2.2. Protocolos

Definición 3.2.2. (Protocolo) Un *protocolo* para $n+1$ procesos es una terna $(\mathcal{I}, \mathcal{P}, \Lambda)$, donde:

- $\mathcal{I}$ es un complejo cromático puro de dimensión n, coloreado con nombres de Π y etiquetado con valores de V^{in}, tal que cada vértice es identificado únicamente por su color junto con su etiqueta.

- $\mathcal{P}$ es un complejo cromático puro de dimensión n, coloreado con nombres de Π y etiquetado con valores de *Vistas*, tal que cada vértice es identificado únicamente por su color junto con su etiqueta.

- $\Lambda : \mathcal{I} \to 2^{\mathcal{P}}$ es un mapeo portador estricto y cromático tal que $\mathcal{P} = \bigcup_{\sigma \in \mathcal{I}} \Lambda(\sigma)$.

Definición 3.2.3. Dada una tarea $(\mathcal{I}, \mathcal{O}, \Delta)$ para $n+1$ procesos y un protocolo $(\mathcal{I}, \mathcal{P}, \Lambda)$, se dice que el protocolo *resuelve* la tarea si existe un mapeo simplicial cromático $\delta : \mathcal{P} \to \mathcal{O}$, llamado *mapeo de decisión* que satisface

$$\delta(\Lambda(\sigma)) \subseteq \Delta(\sigma) \tag{3.1}$$

para todo $\sigma \in \mathcal{I}$.

Definición 3.2.4. El *protocolo de instantáneas inmediatas de una ronda* $(\mathcal{I}, \mathcal{P}, \Lambda)$ para $n+1$ procesos consta de:

- El complejo de entrada $\mathcal{I}$, que puede ser cualquier complejo simplicial cromático puro de dimensión n, coloreado con nombres de Π y etiquetado por V^{in}.

- El mapeo portador $\Lambda : \mathcal{I} \to 2^{\mathcal{P}}$, envía a cada simplejo $\sigma \in \mathcal{I}$ al subcomplejo de posibles configuraciones finales de las ejecuciones del protocolo de instantáneas inmediatas de una ronda (Pseudocódigo 3.1) por cada uno de los procesos (y solamente esos procesos) que participan en σ.

- El complejo de protocolo $\mathcal{P}$, es la unión de $\Lambda(\sigma)$, sobre todos los $\sigma \in \mathcal{I}$.

El procotolo descrito anteriormente es un protocolo según la definición 3.2.2 [9]; en lo que sigue se hace una descripción de la topología de este protocolo, para ello considere a $\sigma \in \mathcal{I}$, tal que $\sigma = \{(P,P),(Q,Q),(R,R)\}$, lo que equivale a que cada proceso use como valor de entrada su propio nombre. El complejo $\Lambda(\sigma)$ se conforma de todas las configuraciones finales obtenidas cuando P, Q y R ejecutan el protocolo iterado de una ronda. Los vértices de $\Lambda(\sigma)$ son de la forma (S, v), donde v es la vista final del S.

La figura 3.2 muestra una representación del complejo $\Lambda(\sigma)$.

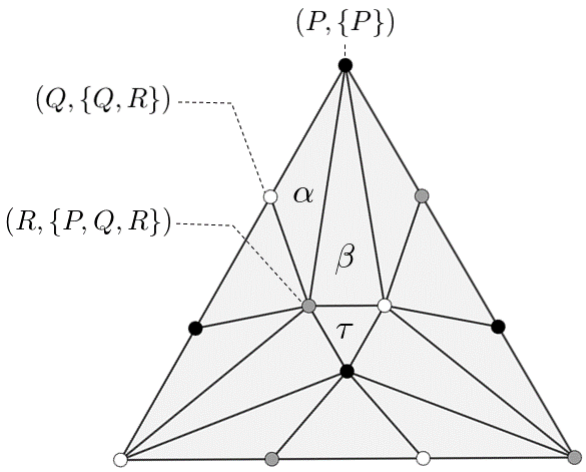

Figura 3.2: $\Lambda(\sigma)$. Las parejas que conforman las vistas se abrevian usando únicamente el nombre de cada proceso, pues el valor de entrada de cada proceso es su nombre.

Los vértices cuyos nombres son P, Q y R, están coloreados de negro, blanco y gris, respectivamente. El simplejo que tiene la etiqueta α, corresponde a la configuración final obtenida de una ejecución totalmente secuencial. El simplejo β corresponde a una ejecución en la que P es el primero en ejecutar el protocolo, y después, Q y R lo hacen de forma simultánea, obteniendo estos dos la misma vista. El último simplejo señalado es τ, y corresponde a una ejecución totalmente concurrente, en la que los tres procesos realizan un paso a la vez. El vértice $(P, \{(P,p)\})$ está tanto en α como en β, en ambos casos P fue el primero en ejecutar el protocolo, por lo que P no puede ver lo que escribieron el resto de los procesos en la memoria; en esta situación las ejecuciones α y β son indistinguibles para P, y en general un vértice no puede distinguir entre las ejecuciones que representan los triángulos a los que pertenece.

Para este caso particular, uno puede enlistar todas las configuraciones finales que son accesibles desde σ para darse cuenta que la representación anterior es correcta. El complejo $\Lambda(\sigma)$, no es más que la subdivisión cromática de σ, Chσ, establecida en la definición 2.6.1. Este hecho es válido en general, para cualquier $\sigma \in \mathcal{I}$, los complejos $\Lambda(\sigma)$ y Chσ son isomorfos [6, 9].

Definición 3.2.5. (Composición de Protocolos) Suponga que $(\mathcal{I}, \mathcal{P}, \Lambda)$ y $(\mathcal{I}', \mathcal{P}', \Lambda')$ son protocolos tales que $\mathcal{P} \subseteq \mathcal{I}'$. La *composición* de dichos protocolos es el protocolo $(\mathcal{I}, \mathcal{P}'', \Lambda'')$, donde Λ'' es la composición de Λ y Λ', $(\Lambda' \circ \Lambda)(\sigma) = \Lambda'(\Lambda(\sigma))$, para σ en $\mathcal{I}$, y $\mathcal{P}'' = \Lambda''(\mathcal{I})$.

La definición anterior es correcta, pues de acuerdo a las proposiciones 2.2.6 y 2.3.5, la composición de mapeos portadores estrictos y cromáticos es también un mapeo estricto y cromático, respectivamente.

Para modelar la estructura multironda del protocolo iterado descrito en la sección 3.1.4, se hará uso de la composición de protocolos recién definida. Después de que el proceso P ejecuta el protocolo de instantáneas inmediatas de una ronda, éste obtiene una vista, digamos v. En la segunda ronda, P vuelve a ejecutar el protocolo de una ronda, pero ahora usando como valor entrada a v. Este procedimiento se puede repetir las veces que sea necesario, cada proceso usa su vista final como valor de entrada para la siguiente ejecución del protocolo. De esta manera, el resultado de componer N veces el protocolo de instantáneas inmediatas de una ronda es equivalente al protocolo iterado de N rondas.

Definición 3.2.6. El *protocolo de instantáneas inmediatas de r rondas* es el protocolo que se obtiene al componer r veces el protocolo de instantáneas inmediatas de una ronda descrito en la definición 3.2.4.

Con el fin de analizar la estructura combinatoria de los complejos de los protocolos multironda, considere la ejecución del protocolo de dos rondas por los procesos P y Q, de tal modo que cada proceso usa su nombre como valor de entrada. El simplejo $\sigma = \{(P, P), (Q, Q)\}$ representa a la configuración inicial descrita anteriormente. El complejo del protocolo de una ronda para σ es la imagen de σ bajo Λ, que equivale a la subdivisión cromática de σ, Ch(σ). El complejo $\Lambda^2(\sigma)$ se calcula como la unión de las imágenes de

los simplejos de $\Lambda(\sigma)$ bajo Λ (basta tomar la unión sobre las facetas, según la observación 2.2.2), esto refleja la idea de que cada proceso inicia la segunda ronda usando como valor de entrada la vista que obtuvo en la primera ronda, pues los vértices en $\Lambda(\sigma)$ están etiquetados con las vistas obtenidas por los procesos. De acuerdo con la figura 3.3, las facetas de $\Lambda(\sigma)$ son las aristas α_1, α_2 y α_3; al aplicar el mapeo Λ a cualesquiera de estas aristas, obtenemos la subdivisión cromática de la arista en cuestión. Así que se puede decir que $\Lambda^2(\sigma)$ se obtiene al aplicar dos veces el mapeo Ch a σ (Ch calcula la subdivisión cromática de un complejo), es decir, $\Lambda^2(\sigma)$ es el complejo $\mathrm{Ch}^2(\sigma) = \mathrm{Ch}(\mathrm{Ch}(\sigma))$. Al complejo $\mathrm{Ch}^2(\sigma)$ se denominará la *segunda subdivisión cromática de* σ. (De forma similar se puede definir la $r-$*ésima subdivisión cromática de* σ). Usando la nueva terminología, se puede decir que el complejo $\Lambda^2(\sigma)$ no es otro sino la segunda subdivisión cromática de σ.

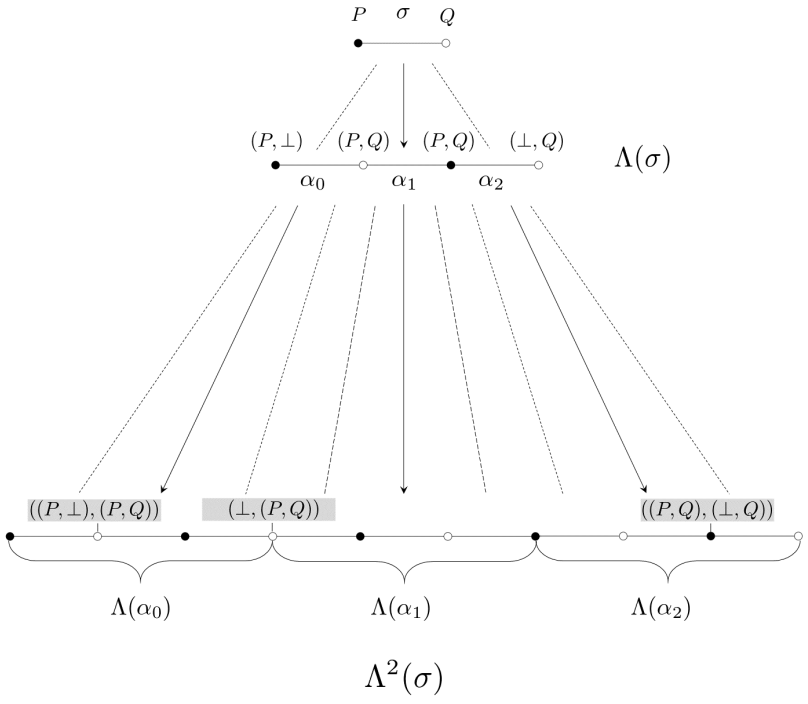

Figura 3.3: Los vértices de σ están etiquetados con sus valores de entrada. Los vértices de los complejos $\Lambda(\sigma)$ y $\Lambda^2(\sigma)$ están etiquetados con sus respectivas vistas, abreviadas de modo que la etiqueta $(P, \perp)$ representa a la vista $\{(P, P), (Q, \perp)\}$.

3.2.3. Imposibilidad del consenso binario

Para finalizar este capítulo, se presenta una sencilla prueba del porqué el consenso binario no se puede resolver en el modelo de cómputo presentado en este capítulo, aunque esta prueba se presenta para el caso de dos procesos, el resultado es válido en general y el argumento es similar al que aquí se presenta y esencialmente es que no es posible definir un mapeo de decisión que resuelva el consenso binario porque el complejo de protocolo es conexo.

Proposición 3.2.7. *(Imposibilidad del consenso binario) El consenso binario descrito en la sección 3.2.1, no se puede resolver mediante el protocolo iterado de instantáneas inmediatas.*

Demostración. La prueba se hará para dos procesos. Primero se explicará porque no se puede definir el mapeo de decisión descrito en la definición 3.2.3 después de la primera ronda y después se analiza el caso general.

Sea $\mathcal{I}$ el complejo de entrada para dos procesos P_0 y P_1 que están coloreados de negro y blanco, respectivamente; $\mathcal{I}$ consta de cuatro aristas que representan a las configuraciones iniciales y que son a, b, c y d, según la figura 3.4.

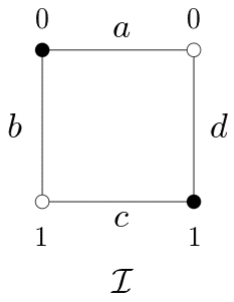

Figura 3.4: Complejo de entrada $\mathcal{I}$.

Para la demostración basta con analizar una arista y en este caso en análisis se hará con la arista b. Al terminar la primera ronda el complejo de protocolo $\Lambda(b)$ es una gráfica que consta de tres aristas e_0, e_1 y e_2, note que dicha gráfica es conexa. Si existe un mapeo sim-

plicial $\delta : \Lambda(b) \to \mathcal{O}$ tal que $\delta(\Lambda(b)) \subseteq \Delta(b)$, entonces al ser δ simplicial, $|\delta| : |\Lambda(\sigma)| \to |\mathcal{O}|$ es un mapeo continuo, y como $\Lambda(b)$ es conexa, entonces $|\delta|(|\Lambda(b)|)$ es conexo, o lo que es lo mismo la gráfica $\delta(\Lambda(b))$ es conexa, lo que implica que dicha gráfica tiene que estar contenida en una componente conexa de $\Delta(b)$, pues $\delta(\Lambda(b)) \subseteq \Delta(b)$, como las únicas componentes conexas de $\Delta(b)$ son f_0 y f_1, entonces $\delta(\Lambda(b))$ está contenido en una de esas aristas y como δ es rígido entonces $\delta(\Lambda(d))$ es exactamente una de las dos aristas ya mencionadas, lo que equivale a decir que después de ejecutar el protocolo de comunicación los dos procesos eligieron un mismo valor, si dicho valor es 1, entonces $\delta(\Lambda(b)) = f_1$, en particular se cumple que $\delta(e_0) = \delta(e_1) = \delta(e_2) = f_1$; recuerde que estas aristas representan ejecuciones del protocolo de comunicación, por ejemplo, e_0 es una ejecución en la que P_0 escribió y leyó en la memoria sin conocer el valor de entrada de P_1, según lo dicho anteriormente, en esta ejecución P_0 decide 1, lo cual es incorrecto, pues si P_0 ejecutó el protocolo de comunicación antes que P_1, entonces para P_0 la ejecución e_0 es indistinguible de aquella en la que P_0 es el único que ejecuta el protocolo de comunicación con valor de entrada 0 y en la que necesariamente tiene que decidir su mismo valor de entrada, pues el valor de salida del consenso tiene que ser el valor de entrada de alguno de los procesos que participan en el algoritmo, y al ser P_0 el único participante, decide 0, en contradicción a lo que se había dicho anteriormente; formalmente, el vértice coloreado de negro de la arista e_0 corresponde a la ejecución en solo de P_0, $\delta(\Lambda(v_0))$, este vértice es enviado por δ al vértice en f_1 coloreado de negro, violando así la restricción que Δ establece para v_0, pues $\Delta(v_0)$ es el vértice en f_0 de color negro. La misma contradicción surge al suponer que el mapeo de decisión δ envía a $\Lambda(b)$ a f_0. Con esto queda demostrado que es imposible definir tal mapeo de decisión δ, debido a que $\Lambda(b)$ es conexo.

Para el caso general, es decir cuando el protocolo se ejecuta r veces, el análisis es muy similar, ya que $\Lambda^r(b)$ es una gráfica conexa para toda r, por lo que la imagen de dicha gráfica bajo un mapeo simplicial y rígido δ tiene que ser f_0 o f_1, a partir de lo cual se puede repetir el argumento presentado anteriormente para concluir que no es posible definir un mapeo δ tal que $\delta(\Lambda^r(b)) \subseteq \Delta(b)$.

□

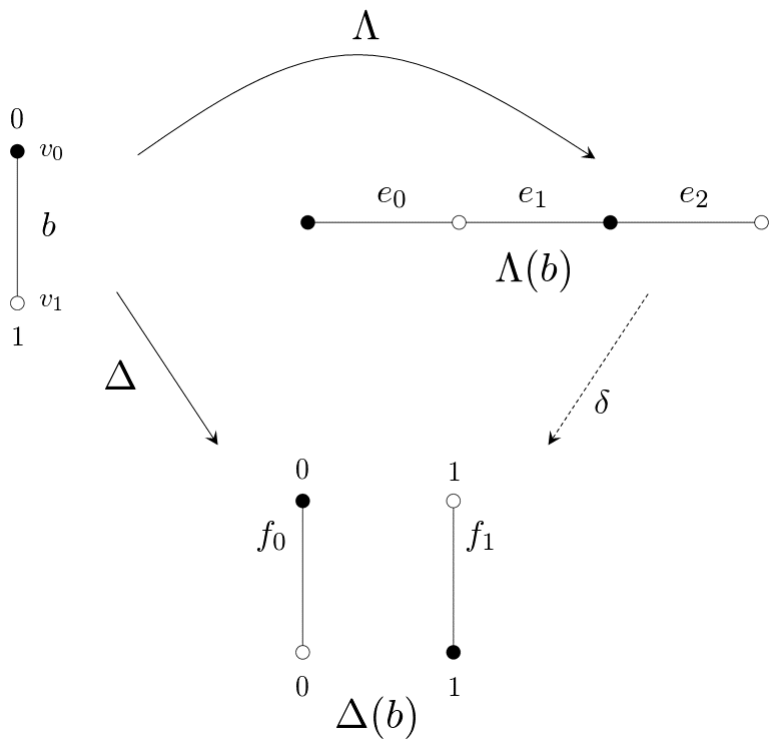

Figura 3.5: No es posible definir δ tal que $\delta(\Lambda(d)) \subseteq \Delta(d)$.

Capítulo 4

Protocolo aleatorio de instantáneas inmediatas

En la primera parte de este documento, se explicó como a través de la topología combinatoria es posible modelar nociones como son las tareas y protocolos, junto con una definición de solubilidad; en particular se presentó el protocolo iterado de instantáneas inmediatas, dicho protocolo ha sido ampliamente estudiado y se sabe con exactitud cuáles son las tareas que se pueden resolver en ese modelo de cómputo [9]. Sin embargo, toda la teoría desarollada hasta el momento se ha centrado en el estudio de algoritmos distribuidos deterministas, dejando lugar a una pregunta que surge forma natural y es saber si es posible modelar algoritmos aleatorios usando topología combinatoria. En primer lugar cabe decir que existen distintas formas de implementar algoritmos aleatorios en un sistema distribuido [1], una de ellas es extender el modelo de cómputo dotando a los procesos con una operación que les permita obtener valores aleatorios, por ejemplo, lanzando una moneda con cierta distribución de probabilidad. En el capítulo anterior se presentó una demostración del hecho que el consenso binario no tiene solución en el modelo de instantáneas inmediatas; el resultado de imposibilidad es aún más general y se puede consultar en [11] y en [9], este último usando a la topología como herramienta para la prueba. El problema del consenso ha sido ampliamente estudiado, en [1] se puede consultar una serie de algoritmos aleatorios que abordan el problema del consenso binario sin violar el resultado

de imposibilidad [11], cambiando algunas condiciones como pedir que el requerimiento de terminación suceda con cierta probabilidad, por ejemplo con probabilidad 1 o mayor que cero [3].

El objetivo de esta segunda parte del documento es extender el modelo de instantáneas inmediatas agregando una operación que dé como resultado valores aleatorios, que serán las caras de una moneda (0 y 1) y partir de ello describir las configuraciones finales que se obtienen después de que los procesos ejecuten este nuevo protocolo extendido al que se llamará protocolo aleatorio. Con lo dicho al inicio del capítulo resulta comprensible que la extensión se haga sobre el modelo de instantáneas inmediatas. En este capítulo se presentan los modelos operacional y combinatorio del protocolo aleatorio y en los capítulos siguientes se estudian propiedades del modelo combinatorio, principalmente relacionadas con la conexidad del complejo asociado al protocolo.

4.1. Modelo operacional

Para dar al protocolo de instantáneas inmediatas un comportamiento aleatorio, se optará por equipar a los procesos con una operación llamada coin-flip() que regresa valores aleatorios, en principio estos valores serán 0 y 1, y la probabilidad de obtener uno de dichos valores es la misma para ambos.

Algoritmo 4.1 Protocolo aleatorio de instantáneas inmediatas.

```
shared mem : array[0...n]                          ▷ Hay n + 1 procesos
procedure PROTOCOLOALEATORIO(v_i)                  ▷ Ejecutado por el proceso P_i
    mem[i] := v_i
    vista:= snapshot(mem[*])
    coinValue := coin-flip()                       ▷ El valor obtenido es 0 o 1
end procedure
```

El resultado de la operación coin-flip() permite a los procesos decidir un valor cuando no es posible hacerlo únicamente a partir de sus vistas.

En el capítulo anterior se presentó el protocolo de instantáneas inmediatas y se observó que el complejo de protocolo $\Lambda(\sigma)$ es isomorfo a la subdivisión cromática de σ, Chσ. Al agregar la operación coin-flip() surge como una pregunta natural saber cómo describir

la topología del complejo de configuraciones finales que se pueden obtener después de la ejecución del protocolo aleatorio.

4.2. Topología del protocolo aleatorio para dos procesos

Sea σ una arista cualquiera. Para describir a las ejecuciones del protocolo aleatorio considere el experimento que consiste en lanzar una moneda justa que toma valores 0 y 1 por cada uno de los vértices de cada una de las aristas de $\mathrm{Ch}(\sigma)$, en cada vértice se lanzan tantas monedas como aristas incidan en él y cada una de estas monedas estará asociada con una arista en particular; además suponga que cada uno de estos lanzamientos es independiente del resto. Al asociar a cada vértice con el valor que resulta al lanzar una moneda, se obtiene una gráfica como el de la figura 4.1 (note que hay vértices asociados con más de una moneda pero estas monedas a su vez están asociadas con una sola arista), si se denota por $\Gamma(\sigma)$ a una de estas posibles gráficas, entonces los vértices de $\Gamma(\sigma)$ son de la forma $(v, coin)$, donde v es un vértice de $\mathrm{Ch}(\sigma)$ y $coin$ es el valor que se obtuvo al lanzar una moneda.

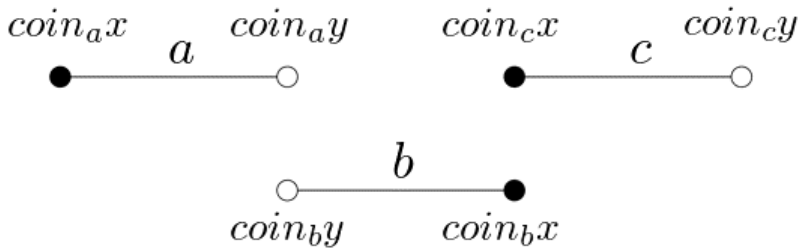

Figura 4.1: Representación de $\Gamma(\sigma)$.
Las aristas de $\mathrm{Ch}(\sigma)$ son a, b y c.

Distintas combinaciones de monedas dan lugar a distintas gráficas, y éstas pueden tener distintas propiedades topológicas; por ejemplo, si las monedas $coin_a y$ y $coin_b y$ son distintas, la gráfica obtenida es disconexa, esto sin importar los valores del resto de las monedas. Suponga que $coin_a y = 1$ y $coin_b y = 0$, y el resto de monedas es igual a 0, la gráfica que se obtiene es la que muestra en la figura 4.2.

Note que para que $\Gamma(\sigma)$ sea conexa es necesario que $coin_a y = coin_b y$ y $coin_b x = coin_c x$, mientras que para que sea disconexa basta que alguna de esas igualdades no se cumpla.

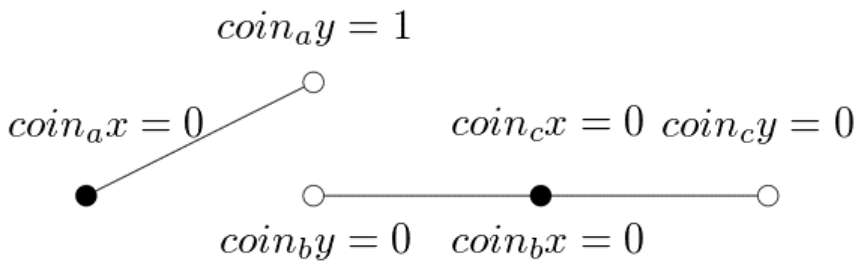

Figura 4.2: La gráfica es disconexa porque $coin_a y \neq coin_b y$.

Como el resultado de lanzar las monedas es independiente, la probabilidad de que dadas cualesquiera dos monedas $coin_1$ y $coin_2$ coincidan en su valor es $\frac{1}{2}$. Así también, como los lanzamientos son independientes, la probabilidad de que $\Gamma(\sigma)$ sea conexa se puede calcular como el producto de $\Pr(coin_a y = coin_b y)$ y $\Pr(coin_b x = coin_c x)$, que en este caso es $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$. Para cada una de las aristas en $\mathrm{Ch}\sigma$, hay cuatro posibles resultados a obtener, por ejemplo para la arista a las posibilidades son $coin_a x = 0 = coin_a y$, $coin_a x = 0$ y $coin_a y = 1$, $coin_a x = 1$ y $coin_a y = 0$, y $coin_a x = 1 = coin_a y$; cada una de estas posibilidades representa a una arista y la probabilidad de obtener alguna de ellas es la misma para cada una, que es $\frac{1}{4}$.

4.3. Un algoritmo aleatorio para el consenso binario

En el capítulo anterior se explicó porque el consenso binario no se puede resolver de forma determinista en el modelo de instantáneas inmediatas, en dicha prueba fue crucial el hecho que el complejo de protocolo $\Lambda^r(\sigma)$ es conexo para cualquier r. A continuación se presenta un algoritmo aleatorio para el consenso binario y después se hace un estudio de la topología *asociada* a dicho algoritmo.

El algoritmo 4.2 es una adaptación de la versión presentada en [16]. Si bien este algoritmo no tiene la forma genérica del protocolo aleatorio descrito al inicio del capítulo, resulta conveniente hacer el análisis en esta forma no genérica para ilustrar de forma clara la estructura topológica asociada al algoritmo.

Algoritmo 4.2 Consenso binario.

```
1: shared propuesta : array[r][0...n] acuerdo : array[r][0...n]        ▷ Hay n + 1 procesos
2: procedure CONSENSO(pref := input)                                   ▷ Ejecutado por el proceso P_i
3:     decide := false; r:= 0;
4:     while !decide do
5:         propuesta[r][i] := pref;
6:         vista:= snapshot(propuesta[r][*]);
7:         if 0 y 1 aparecen como valores propuestos en vista then
8:             acuerdo := false;
9:         else
10:            acuerdo := true;
11:        end if
12:        acuerdo[r][i] := acuerdo;
13:        check:= snapshot(acuerdo[r][*]);
14:        if false aparece en check then
15:            coin := coin-flip();
16:            if hay una j tal que check[r][j] = true then
17:                pref := vista[r][j];
18:            else
19:                pref := coin;
20:            end if
21:        else
22:            decide := true;
23:        end if
24:        r := r+1;
25:    end while
26:    return pref;
27: end procedure
```

En la ronda r, el proceso P_i *anuncia* el valor que propone escribiéndolo en su entrada correspondiente de la memoria compartida propuesta[r][i] (línea 5), inmediatamente después, P_i toma una instantánea de la memoria propuesta[r][*] (línea 6) para después escribir true o false en una segunda memoria compartida acuerdo[r][i], dependiendo de si todos los procesos propusieron el mismo valor (línea 12); nuevamente P_i toma una instantánea, pero ahora de la memoria acuerdo[r][*], a la que se denomina check(línea 13), y después procede de la siguiente manera: 1) Si ve que todos los procesos en check tienen el valor true, entonces P_i termina la ronda y decide su propio valor; 2) Si ve al menos un false y al menos un true, entonces cambia su preferencia por la de algún proceso que tiene el valor true; 3) Si sólo ve el valor false, entones cambia su preferencia por el resultado que se obtiene al lanzar una moneda. Aquí es donde las monedas juegan un papel crucial, pues permiten a

los procesos cambiar de preferencia de forma no determinista para así conseguir acordar en la misma preferencia.

Al igual que en el capítulo anterior se asumirá que inmediatamente después de escribir en la memoria, cada proceso toma una una instantánea de la misma. Al finalizar la ronda r, el estado del proceso P_i se puede caracterizar por las instantáneas que tomó (vista y check) y su preferencia ($pref$), dicho estado será representado por la tupla $(P_i, vista_i, check_i, pref_i)$.

Recuerde que el orden en que los procesos realizan las operaciones da lugar a distintas ejecuciones. Para este algoritmo en particular, el orden en que los procesos toman las instantáneas puede hacer que un proceso cambie o no su preferencia en una ronda. Por ejemplo, suponga que los procesos P_0, P_1 y P_2 (coloreados de negro, blanco y gris, respectivamente) ejecutan el algoritmo con valores de entrada 1, 1 y 0, respectivamente, y que en la primera ronda P_1 es el primero en anunciar su preferencia, seguido de P_0 y P_2, que lo hacen de forma concurrente, de manera que la variable local *acuerdo* de P_0, P_1 y P_2 tendrá el valor false, true y false, respectivamente. Luego cada uno de los procesos tiene que escribir el valor de su variable *acuerdo* en la memoria acuerdo y después tomar una instantánea de la misma para saber si deben o no cambiar su preferencia. Uno puede sentirse tentado a pensar que el hecho que la variable *acuerdo* de P_1 tenga el valor true es suficiente para que el resto de procesos cambien su preferencia por la de P_1, pero esto no es suficiente, pues debido a la asincronía del sistema, es posible que alguno de los demás procesos anuncie primero el valor de su variable *acuerdo* sin ver el anuncio correspondiente de P_1, esto sin importar que P_1 haya sido el primero en anunciar su preferencia. Primero considere el caso en el que P_2 es el primero en anunciar el valor de *acuerdo*, seguido de P_0 y P_1 que lo hacen de forma concurrente. En la instantánea check que P_2 obtiene, el único valor que ve es false, por lo que P_2 cambia su preferencia utilizando una moneda. Después P_0 y P_1 obtienen su respectiva instantánea check, ambos ven los valores true y false, P_1 conserva su preferencia, pues su variable *acuerdo* es true, y P_0 cambia su preferencia por la de P_1; al final de la primera ronda las preferencias de los procesos P_0, P_1 y P_2 son 1, 1 y *coin*, donde *coin* es el valor que P_2 obtuvo al lanzar una moneda. Ahora considere el caso en el que P_0 y P_2 anuncian su valor de *acuerdo*, seguidos de P_1; estos dos procesos sólo

verán el valor false en check y cada uno decidirá su preferencia al lanzar una moneda. Por su parte P_1 conservará su preferencia y al final de la ronda las preferencias serán $coin_{P_0}$, 1 y $coin_{P_2}$. Los dos ejemplos anteriores muestran que hay que considerar tanto el orden en que los procesos anuncian su preferencia como el orden en que anuncian su acuerdo, junto con el resultado de las monedas que lanzan.

Para cada posible ejecución del algoritmo 4.2 por los procesos P_0, P_1 y P_2, la configuración del sistema al terminar la primera ronda se puede representar con un simplejo de dimensión dos que consta de tres vértices de la forma $(P_i, vista_i, check_i, pref_i)$, donde cada vértice describe el estado final de un proceso al terminar la ronda; para describir al conjunto de posibles configuraciones finales después de la primera ronda, considere a $\sigma = \{(P_0, 1), (P_1, 1), (P_2, 0)\}$ como la configuración inicial; las distintas ejecuciones de la primera instantánea por los tres procesos determinan una subdivisión de σ, que es la subdivisión cromática de σ; después los procesos tienen que realizar otra instantánea, y nuevamente las distintas ejecuciones de esta instantánea determinan una subdivisión cromática de cada una de las configuraciones del sistema obtenidas después de la primera instantánea, que están representadas por los simplejos de $\mathrm{Ch}(\sigma)$, así que las posibles configuraciones del sistema después de que cada proceso ejecuta las dos instantáneas (sin considerar aún a la componente $pref$) corresponden a las facetas de la segunda subdivisión cromática de σ. La última coordenada de los vértices que describen el estado final de un proceso es la componente $pref$, que es la preferencia de cada proceso y que puede ser 0 o 1, y ésta dependerá de la información obtenida de las instantáneas o del lanzamiento de una moneda.

Siguiendo con el ejemplo en el que los procesos P_0, P_1 y P_2 ejecutan el algoritmo 4.2 con los valores de entrada $1, 1$ y 0, note que no es posible que todos decidan algún valor en la primera ronda porque al menos un proceso verá que fueron propuestos los valores 0 y 1, y entrará al cuerpo del **if** de la línea 7, por lo que no podrá retornar valor alguno hasta la siguiente ronda; sin embargo, en la segunda ronda es posible que todos los procesos decidan un valor y para ello es necesario (y suficiente), que al final de la primera ronda todos los procesos tengan la misma preferencia. Suponga que sin importar cuál fue el orden

de ejecución de las operaciones de los procesos, al final de la primera ronda todos tuvieron la misma preferencia, que puede ser 0 o 1, y que puede variar por cada ejecución posible. La figura 4.3 representa a un posible complejo (denotado por $\mathcal{R}$) de configuraciones finales en el que sin importar en qué orden se hayan ejecutado las dos instantáneas, todos los procesos terminan con la misma preferencia.

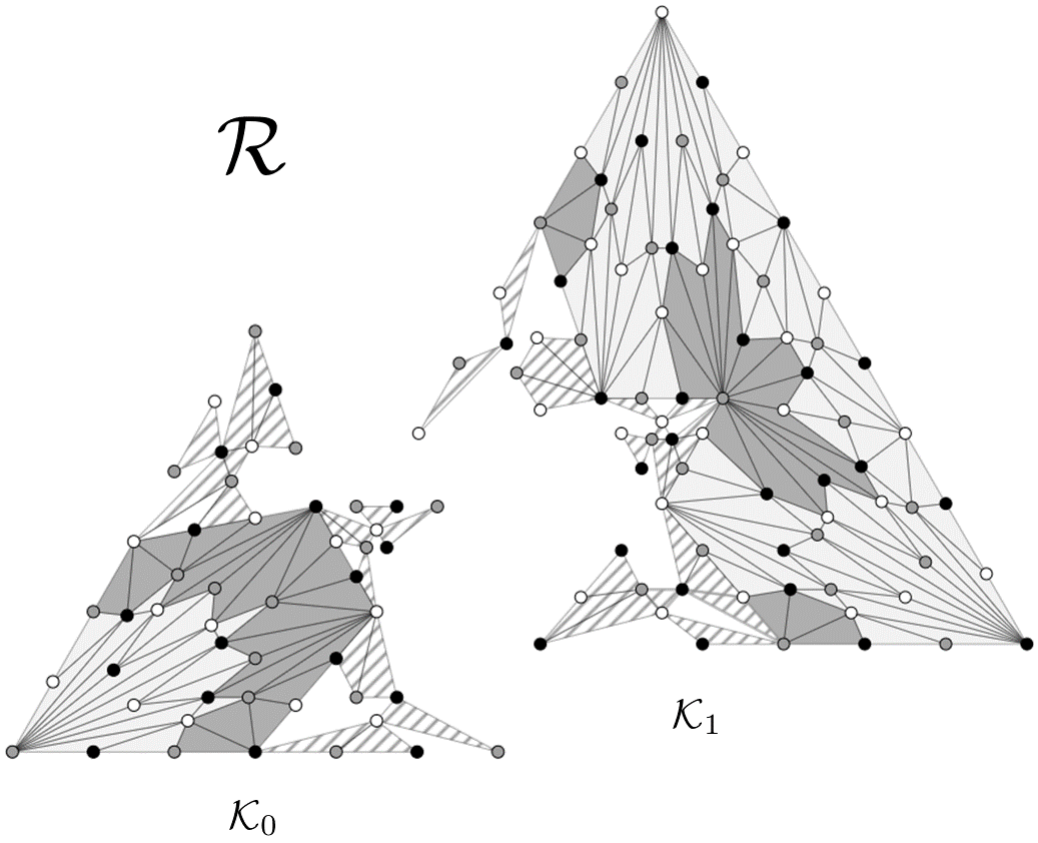

Figura 4.3: Complejo de configuraciones finales $\mathcal{R}$.
$\mathcal{R}$ es la unión de $\mathcal{K}_0$ y $\mathcal{K}_1$.

Los simplejos iluminados con gris claro representan ejecuciones en las que las preferencias de los procesos están determinadas y tienen que ser 0 (si están en $\mathcal{K}_0$) o 1 (si están en $\mathcal{K}_1$), en estas ejecuciones todos los procesos vieron el anuncio de un proceso cuya variable *acuerdo* era `true` e imitaron esa preferencia si la propia era distinta. En los simplejos iluminados con gris oscuro, hubo al menos un proceso cuyo *acuerdo* fue `true`, y un proceso que no vio el anuncio de *acuerdo* del primero; los procesos que no vieron el acuerdo tienen

que lanzar una moneda, y para que todos tengan la misma preferencia el resultado de dicha moneda tiene que ser la preferencia de algún proceso cuyo *acuerdo* fue true. El resto de los simplejos representan ejecuciones en las que ningún proceso tuvo *acuerdo* igual a true y por lo tanto, cada uno de los procesos tiene que lanzar una moneda y el resultado de estas tres monedas debe ser el mismo para que todos terminen la ronda con la misma preferencia; en la figura 4.3 estos simplejos aparecen iluminados con una textura de rayas, si están en $\mathcal{K}_0$ o $\mathcal{K}_1$ todos obtuvieron 0 o 1 al lanzar las monedas, respectivamente. El complejo $\mathcal{R}$ recién descrito es disconexo, las ejecuciones en las que todos los procesos terminan con 0 o 1 como preferencia están contenidas en $\mathcal{K}_0$ o $\mathcal{K}_1$ respectivamente; así pues todos los vértices que están en una misma componente conexa tienen la misma preferencia, por ejemplo, la ejecución en la que P_0 es el primero en ejecutar las dos instantáneas y P_1 y P_2 imitan la preferencia de P_0 (que es 1) no puede estar en la misma componente conexa que la ejecución en la que P_2 ejecuta primero las dos instantáneas y luego P_0 y P_1 siguen su preferencia (que es 0).

Sin la ayuda de las monedas no habría sido posible que el complejo $\mathcal{R}$ fuera disconexo; recuerde que en el capítulo anterior el argumento que se usó para probar que no existe un mapeo de decisión para resolver el consenso binario fue que el complejo de protocolo para una configuración inicial en la que los valores de entrada son distintos es conexo, dicho argumento no tiene validez en este caso pues $\mathcal{R}$ no es conexo, incluso se podría hablar de un mapeo simplicial δ, tal que $\delta(\mathcal{R}) \subseteq \Delta(\sigma)$, que es el que envía a cada vértice a su respectiva preferencia.

4.4. Modelo combinatorio

Recuerde que en la sección 4.2 se hizo una descripción de los posibles complejos que se obtienen después de que dos procesos ejecutan el protocolo alteatorio, esta descripción se hizo a través de un experimento que consiste en lanzar monedas por cada vértice de las aristas de $\mathrm{Ch}(\sigma)$, donde σ es cualquier configuración inicial. Para cada arista a de $\mathrm{Ch}(\sigma)$, lanzar un par de monedas equivale a elegir de forma aleatoria una arista de la pseudoesfera $\Psi(a)$ (figura 4.4), así que lanzar monedas en los vértices de las aristas de $\mathrm{Ch}(\sigma)$ es lo mismo

que elegir una arista de forma aleatoria por cada pseudoesfera correspondiente.

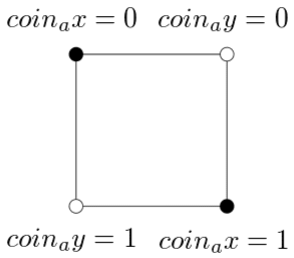

Figura 4.4: La pseudoesfera $\Psi(a)$.

Con lo dicho anteriormente, es posible describir de forma alternativa a los posibles complejos de configuraciones finales que se obtienen cuando $n + 1$ procesos ejecutan el protocolo aleatorio; si σ es una configuración inicial, entonces los complejos recién mencionados se obtienen al elegir aleatoriamente una faceta de cada una de las psuedoesferas $\Psi(\tau)$, donde $\tau \in facet(\Lambda(\sigma)) = facet(\mathrm{Ch}(\sigma))$ (figura 4.5).

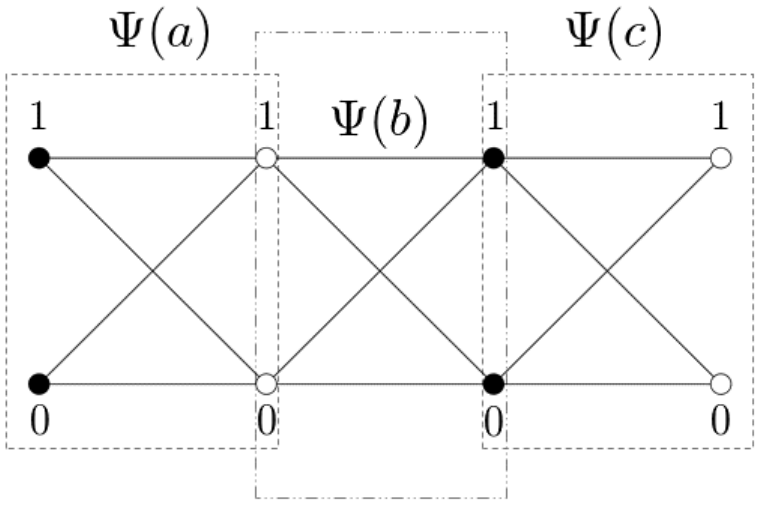

Figura 4.5: El complejo $\Psi(\Lambda(\sigma))$, es unión de tres pseudoesferas.
Las aristas de $\Lambda(\sigma)$ son a, b y c.

Si se considera a la unión de todos los posibles complejos de configuraciones finales se obtiene un complejo al que se denotará por $\Lambda_{\mathcal{R}}(\sigma)$

$$\Lambda_{\mathcal{R}}(\sigma) = \bigcup_{\tau \in \Lambda(\sigma)} \Psi(\tau).$$

En el siguiente capítulo se estudiarán algunas propiedades topológicas de $\Lambda_{\mathcal{R}}(\sigma)$.

Capítulo 5

Propiedades de $\Lambda_{\mathcal{R}}$

Recuerde que al final del capítulo anterior se definió al complejo $\Lambda_{\mathcal{R}}(\sigma)$ como el resultado de aplicar el operador Ψ a cada una de las facetas del complejo $\Lambda(\sigma)$. En este capítulo se estudian algunas propiedades referentes a la conexidad de dicho complejo, en particular se presentan dos proposiciones referentes a la conexidad del complejo de protocolo $\Lambda_{\mathcal{R}}(\sigma)$ y también $\Lambda_{\mathcal{R}}(\mathcal{I})$, cuando σ es un simplejo de dimensión n y $\mathcal{I}$ es un complejo shellable y puro de dimensión n, que establecen que ambos complejos son $(n-1)-$conexos. En el capítulo 3 se hizo notar que la conexidad del complejo de protocolo $\Lambda(\sigma)$ fue determinante para el resultado de imposibilidad allí presentado; la conexidad en dimensiones altas también es importante al estudiar otras tareas, por eso es de interés estudiar la conexidad del complejo $\Lambda_{\mathcal{R}}(\sigma)$ aún cuando no esté en el alcance de este trabajo discutir resultados de imposibilidad en el modelo de cómputo aleatorio.

A continuación se presentan dos lemas que serán de utilidad para probar que el complejo $\Lambda_{\mathcal{R}}(\sigma)$ es $(n-1)-$conexo si σ es de dimensión n.

Lema 5.0.1. *Sean $\mathcal{K}$ y $\mathcal{L}$ complejos simpliciales, entonces se cumple*

$$\Psi(\mathcal{K} \cap \mathcal{L}) = \Psi(\mathcal{K}) \cap \Psi(\mathcal{L}).$$

Demostración. $\Psi(\mathcal{K}\cap\mathcal{L}) = \bigcup_{\tau\in\mathcal{K}\cap\mathcal{L}} \Psi(\tau) = (\bigcup_{\tau\in\mathcal{K}} \Psi(\tau))\cap(\bigcup_{\tau\in\mathcal{L}} \Psi(\tau)) = \Psi(\mathcal{K})\cap\Psi(\mathcal{L})$. $\square$

Lema 5.0.2. *Si $\mathcal{K}$ es un complejo shellable, puro y de dimensión n, entonces $\Psi(\mathcal{K}) = \bigcup_{\tau\in\mathcal{K}} \Psi(\tau)$ es $(n-1)-$ conexo.*

Demostración. Por inducción sobre n.

Si $n = 1$, entonces $\mathcal{K}$ es una gráfica shellable y pura. Si las aristas de K son $e_1, e_2, ..., e_t$, entonces hay que demostrar que $\Psi(\mathcal{K}) = \bigcup_{i=1}^{t} e_i$ es 0-conexo, lo que se hará por inducción sobre el número de aristas de K.

Si $t = 1$, entonces $\Psi(e_1)$ es $dim(e_1) - 1 = 0-$ conexo (Proposición 2.7.12).

Suponga que toda gráfica $\mathcal{G}$ shellable y pura con $t-1$ aristas es tal que $\Psi(\mathcal{G})$ es $0-$ conexo.

Sea $\mathcal{K}$ una gráfica shellable y pura con t aristas y $e_1, e_2, ..., e_t$ una sucesión shellable para $\mathcal{K}$.

Defina $\mathcal{K}_1 = \Psi(\bigcup_{i=1}^{t-1} e_i)$ y $\mathcal{K}_2 = \Psi(e_t)$.

Note que $\bigcup_{i=1}^{t-1} e_i$ es shellable ($e_1, e_2, ..., e_{t-1}$ es una sucesión shellable) y que por hipótesis de inducción, $\Psi(\bigcup_{i=1}^{t-1} e_i)$ es $0-$ conexo, de manera que $\mathcal{K}_1$ y $\mathcal{K}_2$ son $0-$ conexos; también es cierto que $\mathcal{K}_1 \cup \mathcal{K}_2 = \Psi(\mathcal{K})$, esto último es porque $\Psi(\bigcup_{i=1}^{t-1} e_i) = \bigcup_{i=1}^{t-1} \Psi(e_i)$, así que para concluir que $\Psi(\mathcal{K})$ es conexo, basta probar que $\mathcal{K}_1 \cap \mathcal{K}_2$ es no vacío; como $\mathcal{K}$ es shellable, entonces $(\bigcup_{i=1}^{t-1} e_i) \cap e_t$ es unión de caras de dimensión $0 = dim(e_t) - 1$ de e_t, es decir, dicha intersección es unión de vértices de e_t y por lo tanto es no vacía, así que $\Psi\big((\bigcup_{i=1}^{t-1} e_i) \cap e_t\big)$ es no vacío, o lo que es lo mismo $\mathcal{K}_1 \cap \mathcal{K}_2$ es no vacío (Proposición 5.0.1), lo que permite concluir que $\Psi(\mathcal{K}) = \mathcal{K}_1 \cup \mathcal{K}_2$ es $0-$ conexo.

Suponga que todo complejo $\mathcal{C}$ shellable, puro y de dimensión $n-1$ es tal que $\Psi(\mathcal{C})$ es $(n-2)-$ conexo.

Sea $\mathcal{K}$ shellable, puro y de dimensión n, hay que probar que $\Psi(\mathcal{K}) = \bigcup_{\tau \in \mathcal{K}} \Psi(\tau) = \bigcup_{\tau \in facet(\mathcal{K})} \Psi(\tau)$ es $(n-1)-$ conexo.

Si el número de facetas de $\mathcal{K}$ es t, entonces por inducción sobre t se mostrará que $\Psi(\mathcal{K})$ es $(n-1)-$ conexo.

En caso en que $t = 1$, se tiene $\Psi(\phi_1)$ es $(dim(\phi_1) - 1)-$ conexo (Proposición 2.7.12), y como $dim(\phi_1) = n$, entonces $\Psi(\phi_1)$ es $(n-1)-$ conexo.

Suponga que si $\mathcal{L}$ es un complejo shellabe, puro y de dimensión n, con $t-1$ facetas, entonces $\Psi(\mathcal{L})$ es $(n-1)-$ conexo.

Sea $\mathcal{K}$ shellable y puro de dimensión n con t facetas y $\phi_1, \phi_2, ..., \phi_t$ una sucesión shellable para $\mathcal{K}$. El complejo $\bigcup_{i=0}^{t-1} \phi_i$ es shellable, puro y de dimensión n, con $t-1$ facetas,

además, por hipótesis de inducción se cumple que $\Psi(\bigcup_{i=0}^{t-1}\phi_i)$ es $(n-1)-$ conexo. Sean $\mathcal{K}_1 = \Psi(\bigcup_{i=0}^{t-1}\phi_i)$ y $\mathcal{K}_2 = \Psi(\phi_t)$, note que estos complejos son $(n-1)-$ conexos ($\mathcal{K}_2$ es $(n-1)-$conexo según la proposición 2.7.12) y que además $\mathcal{K}_1 = \bigcup_{i=1}^{t-1}\Psi(\phi_i)$, por lo que $\Psi(\mathcal{K}) = \mathcal{K}_1 \cup \mathcal{K}_2$.

Por otro lado, $\mathcal{K}_1 \cap \mathcal{K}_2$ es $(n-2)-$ conexo, ya que dicho complejo es igual a $\Psi(\bigcup_{i=0}^{t-1}\phi_i) \cap \Psi(\phi_t)$, que a su vez es igual a $\Psi((\bigcup_{i=1}^{t-1}\phi_i) \cap \phi_t)$ (Lema 5.0.1), y como $(\bigcup_{i=1}^{t-1}\phi_i) \cap \phi_t$ es unión de caras de dimensión $n-1 = dim(\phi_t) - 1$ de ϕ_t, pues $\phi_1, ..., \phi_t$ es una sucesión shellable para $\mathcal{K}$, entonces $(\bigcup_{i=1}^{t-1}\phi_i) \cap \phi_t$ es shellable, puro y de dimensión $n-1$ (Lema 2.4.2), la hipótesis de inducción sobre n, asegura que $\Psi((\bigcup_{i=1}^{t-1}\phi_i) \cap \phi_t)$ es $(n-2)-$ conexo, como se afirmó anteriormente.

Finalmente, teniendo que $\mathcal{K}_1$ y $\mathcal{K}_2$ son $(n-1)-$ conexos y tales que $\mathcal{K}_1 \cap \mathcal{K}_2$ es $(n-2)-$ conexo, el corolario 2.5.8 del Lema del Nervio garantiza que $\Psi(\mathcal{K}) = \mathcal{K}_1 \cup \mathcal{K}_2$ es $(n-1)-$ conexo, como se quería demostrar. □

El lema anterior se pudo haber probado con la ayuda del lema 5.0.6 y del hecho de conocer la conexidad de las pseudoesferas; sin embargo, aquí se presentó una demostración alternativa que hace uso de algunas propiedades que son exclusivas de las pseudoesferas.

Proposición 5.0.3. *Sea σ un simplejo de dimensión n, entonces el complejo $\Lambda_{\mathcal{R}}(\sigma)$ es $(n-1)-$ conexo.*

Demostración. Recuerde que por definición $\Lambda_{\mathcal{R}}(\sigma) = \Psi(\Lambda(\sigma))$, como $\Lambda(\sigma)$ es shellable y puro de dimensión n (Observación 2.6.2), el lema 5.0.2 asegura que $\Psi(\Lambda(\sigma))$ es $(n-1)-$ conexo, como lo afirma la proposición. □

El siguiente lema será útil para determinar la conexidad del complejo $\Lambda^r_{\mathcal{R}}(\sigma)$, que es el resultado de componer el protocolo aleatorio r veces.

Lema 5.0.4. *Si $\mathcal{K}$ es un complejo shellable, puro y de dimensión n, entonces $\Psi(\mathcal{K})$ es shellable, puro y de dimensión n.*

Demostración. En primer lugar observe que por definición, para cualquier faceta σ de $\mathcal{K}$, la pseudoesfera $\Psi(\sigma)$ es un complejo puro de dimensión n, así pues $\Psi(\mathcal{K})$ es también un

complejo puro de dimensión n.

Ahora bien, sea $\phi_1, \phi_2, ..., \phi_t$ una sucesión shellable para $\mathcal{K}$. Para cada $i \in \{1, 2, ..., t\}$, sea $\tau_{i_1}, \tau_{i_2}, ..., \tau_{i_{2^{n+1}}}$ una sucesión shellable para $\Psi(\phi_i)$ (Proposición 2.7.10 y proposición 2.7.6).

Ahora considere la sucesión

$$\tau_{1_1}, \tau_{1_2}, ..., \tau_{1_{2^{n+1}}}, ..., \tau_{i_1}, \tau_{i_2}, ..., \tau_{i_{2^{n+1}}}, ..., \tau_{t_1}, \tau_{t_2}, ..., \tau_{t_{2^{n+1}}}.$$

Se afirma que esta sucesión es una sucesión shellable para $\Psi(\mathcal{K})$. Es necesario demostrar que para cualquier elemento τ_{i_k} de la lista anterior, se cumple que $(\bigcup_{(1\leq j<i)\vee(j=i\wedge 1\leq r<k)} \tau_{j_r}) \cap \tau_{i_k}$ es unión de caras de dimensión $n-1$ de τ_{i_k} (si no se hace alguna restricción sobre r, entonces r toma todos los valores entre 1 y 2^{n+1}, i.e. $\bigcup_{1\leq j<i} \tau_{j_r} = \bigcup_{1\leq j<i, 1\leq r\leq 2^{n+1}} \tau_{j_r}$). Separando las τ_{j_r} en las que $j = i$, el complejo mencionado anteriormente es el mismo que

$$\big[(\bigcup_{1\leq j<i} \tau_{j_r}) \cap \tau_{i_k}\big] \cup \big[(\bigcup_{1\leq r<k} \tau_{i_r}) \cap \tau_{i_k}\big]. \tag{5.1}$$

Es conveniente analizar a los dos miembros de la unión por separado, comenzando por el uniendo del lado izquierdo.

Observe que $\bigcup_{1\leq j<i} \tau_{j_r} = \Psi(\bigcup_{j=1}^{i-1} \phi_j)$ y que $\tau_{i_k} \subseteq \Psi(\phi_i)$, por lo que

$$\begin{aligned}
(\bigcup_{1\leq j<i} \tau_{j_r}) \cap \tau_{i_k} &= \Psi(\bigcup_{j=1}^{i-1} \phi_j) \cap (\Psi(\phi_i) \cap \tau_{i_k}) \quad \big(\tau_{i_k} = (\Psi(\phi_i) \cap \tau_{i_k}\big) \\
&= \big(\Psi(\bigcup_{j=1}^{i-1} \phi_j) \cap \Psi(\phi_i)\big) \cap \tau_{i_k} \\
&= \Psi\big((\bigcup_{j=1}^{i-1} \phi_j) \cap \phi_i\big) \cap \tau_{i_k} \qquad \text{Lema 5.0.1.}
\end{aligned}$$

Como $\phi_1, \phi_2, ..., \phi_t$ es una sucesión shellable para $\mathcal{K}$, entonces $(\bigcup_{j=1}^{i-1} \phi_j) \cap \phi_i = \bigcup_{s=1}^{m_i} \phi_{i_s}$; donde cada ϕ_{i_s} es una cara de dimensión $n-1$ de ϕ_i.

Con lo dicho anteriormente, se concluye que

$$(\bigcup_{1\leq j<i} \tau_{j_r}) \cap \tau_{i_k} = \Psi(\bigcup_{s=1}^{m_i} \phi_{i_s}) \cap \tau_{i_k}.$$

Distribuyendo se tiene que

$$(\bigcup_{1\leq j<i} \tau_{j_r}) \cap \tau_{i_k} = \bigcup_{s=1}^{m_i} (\Psi(\phi_{i_s}) \cap \tau_{i_k}),$$

donde cada $\Psi(\phi_{i_s}) \cap \tau_{i_k}$ es una cara de dimensión $n-1$ de τ_{i_k}, hecho que se demostrará a continuación.

Hay que tener en cuenta que ϕ_{i_s} es una cara de dimensión $n-1$ de ϕ_i y que τ_{i_k} es una faceta de $\Psi(\phi_i)$, de modo que τ_{i_k} es un simplejo de dimensión n que puede ser descrito como

$$\tau_{i_k} = \{(N_{i_0}, v_{i_0}), (N_{i_1}, v_{i_1}), ..., (N_{i_n}, v_{i_n}) | V_{i_l} \in \{0,1\}\},$$

donde $\phi_i = \{N_{i_0}, N_{i_1}, ..., N_{i_n}\}$. Al ser ϕ_{i_s} una cara de dimensión $n-1$ de ϕ_i, toma n vértices de ϕ_i; sin perder generalidad, se puede suponer que dichos vértices son $N_{i_0}, N_{i_1}, ..., N_{i_{n-1}}$, con lo que los simplejos de $\Psi(\phi_{i_s})$ quedan descritos de la siguiente manera (Definición 2.7.3):

Para cualquier subconjunto J de $\{0, 1, ..., n-1\}$, el conjunto:

$\{(N_{i_j}, v_j) | j \in J, v_j \in \{0,1\}\}$, es un simplejo de $\Psi(\phi_{i_s})$ si los N_{i_j} son distintos.

Así, la intersección de $\Psi(\phi_{i_s})$ y τ_{i_k} es

$$\Psi(\phi_{i_s}) \cap \tau_{i_k} = \{(N_{i_0}, v_{i_0}), (N_{i_1}, v_{i_1}), ..., (N_{i_{n-1}}, v_{i_{n-1}})\},$$

pues por definición $\{(N_{i_0}, v_{i_0}), (N_{i_1}, v_{i_1}), ..., (N_{i_{n-1}}, v_{i_{n-1}})\}$ es una cara de $\Psi(\phi_{i_s})$, que a su vez, está contenido en τ_{i_k}. La intersección no puede ser más grande porque dicha intersección es una faceta de $\Psi(\phi_{i_s})$, ya que las parejas no pueden repetir nombres y ϕ_{i_s} sólo tiene n vértices. Lo que prueba que efectivamente, la intersección de $\Psi(\phi_{i_s})$ y τ_{i_k} es una cara de dimensión $n-1$ de τ_{i_k}, como se afirmó anteriormente, por lo que el uniendo del lado izquierdo de la expresión 5.1 es unión de caras de dimensión $n-1$ de τ_{i_k}.

Para el uniendo del lado derecho de 5.1 recuerde que $\tau_{i_1}, \tau_{i_2}, ..., \tau_{i_{2^{n+1}}}$ es una sucesión shellable para $\Psi(\phi_i)$, de modo que $(\bigcup_{1\leq r<k} \tau_{i_r}) \cap \tau_{i_k}$ es unión de caras de dimensión $n-1$ de τ_{i_k}.

Se ha demostrado que $\left[(\bigcup_{1\leq j<i} \tau_{j_r}) \cap \tau_{i_k}\right] \cup \left[(\bigcup_{1\leq r<k} \tau_{i_r}) \cap \tau_{i_k}\right]$ es unión de caras de dimensión $n-1$ de τ_{i_k}, lo que permite concluir que la sucesión propuesta previamente, es una sucesión shellable para $\Psi(\mathcal{K})$, con lo que concluye la prueba de la proposición. □

Proposición 5.0.5. *Sea σ un simplejo de dimensión n, entonces el complejo $\Lambda^r_{\mathcal{R}}(\sigma)$ es $(n-1)-$ conexo.*

Demostración. Por inducción sobre r se probará que el complejo $\Lambda^r_{\mathcal{R}}(\sigma)$ es $(n-1)-$ conexo y que además es shellable y puro de dimensión n.
Si $r = 1$, la afirmación referente a la conexidad es válida por la proposición 5.0.3; por otra parte, según la observación 2.6.2 (esta observación es sobre propiedades de la subdivisión cromática Ch, pero debido a que los complejos $\mathrm{Ch}(\sigma)$ y $\Lambda(\sigma)$ son isomorfos, las propiedades también son válidas para Λ), $\Lambda(\sigma)$ es shellable y puro de dimensión n, de lo que se sigue que $\Psi(\Lambda(\sigma))$, es decir $\Lambda_{\mathcal{R}}(\sigma)$ es shellable y puro de dimensión n, esto último por el lema 5.0.4.
Suponga que para cualquier simplejo σ de dimensión n, el complejo $\Lambda^r_{\mathcal{R}}(\sigma)$ es $(n-1)-$ conexo, shellable y puro de dimensión n.
Ahora bien, $\Lambda^{r+1}_{\mathcal{R}}(\sigma) = \Lambda_{\mathcal{R}}(\Lambda^r_{\mathcal{R}}(\sigma))$; por hipótesis de inducción $\Lambda^r_{\mathcal{R}}(\sigma)$ es shellable y puro de dimensión n, por lo que según la observación 2.6.2, $\Lambda(\Lambda^r_{\mathcal{R}}(\sigma))$ es shellable y puro de dimensión n, una vez mas del lema 5.0.4 se sigue que $\Psi(\Lambda(\Lambda^r_{\mathcal{R}}(\sigma))) = \Lambda^{r+1}_{\mathcal{R}}(\sigma)$ es $(n-1)-$ conexo, shellable y puro de dimensión n, como se quería demostrar. □

Ahora se enuncia un lema sin demostración, misma que puede consultar en [9], que permitirá determinar la conexidad del complejo $\Lambda_{\mathcal{R}}(\mathcal{I})$, al suponer que $\mathcal{I}$ es shellable.

Lema 5.0.6. *Si $\mathcal{K}$ es un complejo simplicial shellable y puro, y $\Phi : \mathcal{K} \to 2^{\mathcal{L}}$ es un mapeo portador $q-$conexo (Definición 2.5.5), entonces el complejo simplicial $\Phi(\mathcal{K})$ es $q-$conexo.*

Proposición 5.0.7. *Si $\mathcal{I}$ es un complejo shellable, puro y de dimensión n, entonces $\Lambda_{\mathcal{R}}(\mathcal{I})$ es $(n-1)-$conexo.*

Demostración. Considere el mapeo $\Lambda_{\mathcal{R}} : \mathcal{I} \to 2^{\Lambda_{\mathcal{R}}(\mathcal{I})}$, este mapeo es $(n-1)-$conexo, pues si $\sigma \in \mathcal{I}$ es de dimensión m, entonces cod $\sigma = n - m$, y se cumple que $\Lambda_{\mathcal{R}}(\sigma)$ es $(n-1)-(n-m) = (m-1)-$conexo, esto último es cierto por la proposición 5.0.3, además el mapeo es rígido y estricto, ya que tanto Ψ y Λ son mapeos rígidos y estrictos, y la composición de mapeos portadores preserva dichas propiedades (Proposición 2.2.6). Así pues, del lema 5.0.6 se concluye que el complejo $\Lambda_{\mathcal{R}}(\mathcal{I})$ es $(n-1)-$conexo. □

Capítulo 6

Conexidad esperada en rondas subsecuentes

Al final del capítulo 3 se demostró que el problema del consenso binario no se puede resolver en el modelo de instantáneas inmediatas, y el argumento clave en dicha prueba fue que el complejo de protocolo $\Lambda(\sigma)$ es conexo; después, en el capítulo 4 se introdujo el protocolo aleatorio de instantáneas inmediatas, que permite *alterar* la topología del complejo de protocolo estándar, esto cuando se considera a una de las posibles gráficas que se pueden obtener después de que en cada ejecución los procesos lancen una moneda. Como se vio anteriormente, distintas combinaciones de monedas dan lugar a distintas gráficas, con distintas propiedades topológicas. Es de gran interés saber qué tan probable es que se obtenga una gráfica disconexa al hacer el experimento descrito en el capítulo anterior, pues obtener una gráfica disconexa es una condición necesaria (más no suficiente) para poder definir un mapeo similar a un mapeo de decisión que permita solucionar en un caso particular por ejemplo, el consenso binario para dos procesos. Si después de la primera ronda no se obtiene un complejo disconexo, el experimento se puede repetir en cada arista del complejo obtenido y así obtener la representación de los posibles complejos que se obtienen en las rondas subsecuentes del protocolo aleatorio. Al repetir este experimento es posible demostrar que el número esperado de rondas en el que el complejo seleccionado a lo largo de las rondas sea disconexo, es finito. Este capítulo está dedicado a probar dicha afirmación.

La figura 6.1 muestra una representación de la estructura de dichas gráficas, note que hay algunos vértices que están asociados con dos monedas, una de ellas corresponde a la moneda lanzada en la primera parte del experimento.

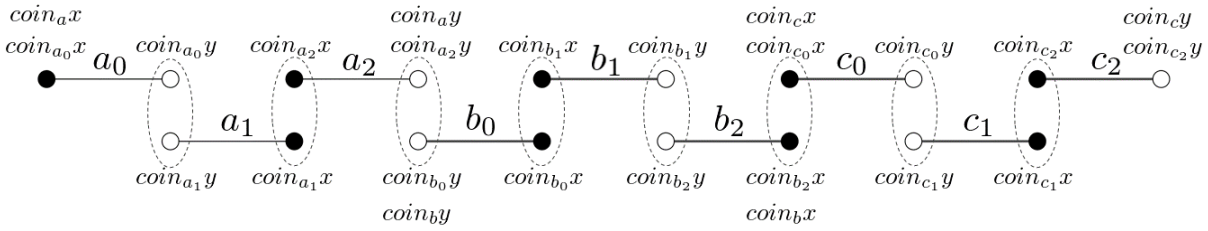

Figura 6.1: Representación de $\Gamma^2(\sigma)$.
Las parejas encerradas se denominarán parejas de vértices *internos*, en este caso son 8 parejas.

Si $\mathcal{A}$ es un gráfica, $\Gamma(\mathcal{A})$ denotará a una posible gráfica que se obtiene al realizar el experimento descrito anteriormente en cada una de las aristas de $\mathcal{A}$, y $\Gamma^2(\sigma)$ denotará a una gráfica que se obtiene al realizar el experimento en las aristas de $\Gamma(\sigma)$, de forma similar se define $\Gamma^r(\sigma)$ como una gráfica que se obtiene al realizar el experimento en las aristas de $\Gamma^{r-1}(\sigma)$, donde σ es cualquier arista. Es importante tener en cuenta que la gráfica $\Gamma(\sigma)$ representa a los posibles resultados del experimento y no se trata de una gráfica fija o determinada.

Lema 6.0.1. *Sea σ una arista cualquiera y Γ^r_C el evento en el que el complejo $\Gamma^r(\sigma)$ es conexo, entonces se cumple que*

$$\Pr(\Gamma^r_C) = \left(\frac{1}{2}\right)^{\sum_{k=1}^{r} 3^k - 1}.$$

Demostración. Primero se hará la prueba para los casos $r = 1, 2$ e inmediatamente después, para el caso general.
Observe que para $r = 1$ (6.2), es necesario que las dos parejas de vértices internos obtengan el mismo valor al lanzar sus correspondientes monedas; lo que los vértices en los extremos obtengan no afecta a la conexidad del complejo $\Gamma(\sigma)$. La probabilidad de que una pareja de vértices coincidan en sus valores al lanzar sus respectivas monedas es $\frac{1}{2}$. La probabilidad

de que $\Gamma(\sigma)$ sea conexo es la intersección del evento en el que la primera pareja de vértices internos obtiene el mismo valor al lanzar las monedas y el evento en el que a la segunda pareja le sucede lo mismo. Como estos eventos son independientes, la probabilidad de la intersección se calcula como el producto de las respectivas probabilidades, que en este caso es la misma y es $\frac{1}{2}$, por lo que la probabilidad de que $\Gamma(\sigma)$ sea conexo es $\frac{1}{4} = \left(\frac{1}{2}\right)^{3^1-1}$.

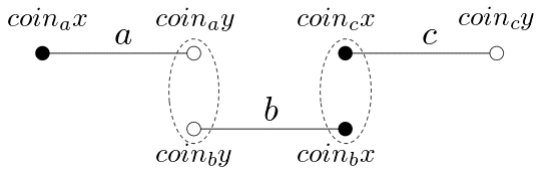

Figura 6.2: Para que $\Gamma(\sigma)$ sea conexa es necesario que las dos parejas de vértices internos coincidan en el valor de sus monedas, respectivamente.

Ahora bien, para $r = 2$ es necesario que $\Gamma(\sigma)$ sea conexo para que $\Gamma^2(\sigma)$ también lo sea, lo que implica que $\Gamma^2_C \subseteq \Gamma^1_C$, por lo que $\Gamma^2_C = \Gamma^2_C \cap \Gamma^1_C$ y $\Pr(\Gamma^2_C) = \Pr(\Gamma^2_C|\Gamma^1_C) \cdot \Pr(\Gamma^1_C)$. Por un lado, se tiene que $\Pr(\Gamma^1_C) = \frac{1}{4}$, así que resta calcular $\Pr(\Gamma^2_C|\Gamma^1_C)$, que es la probabilidad de que $\Gamma^2(\sigma)$ sea conexo asumiendo que $\Gamma^1(\sigma)$ lo es; para que esto suceda, es necesario y suficiente que al igual que en el caso anterior, cada pareja de vértices internos obtenga el mismo valor. El número de parejas de vértices internos es 8, por lo que $\Pr(\Gamma^2_C|\Gamma^1_C)$ es la probabilidad de la intersección de 8 eventos que son aquellos en los que cada pareja de vértices obtiene el mismo valor al lanzar las monedas correspondientes; como estos eventos son independientes, el calculo de la probabilidad de su intersección está dada por el producto de sus probabilidades, que es $\frac{1}{2}$ para cada uno. De lo anterior se sigue que $\Pr(\Gamma^2_C|\Gamma^1_C) = \left(\frac{1}{2}\right)^8$, por lo que $\Pr(\Gamma^2_C) = \left(\frac{1}{2}\right)^8 \cdot \frac{1}{4} = \left(\frac{1}{2}\right)^{3^2-1} \cdot \left(\frac{1}{2}\right)^{3^1-1} = \left(\frac{1}{2}\right)^{(3^1-1)+(3^2-1)}$.

Para el caso general el análisis es muy similar a lo dicho anteriormente, pues para que $\Gamma^r(\sigma)$ sea conexo, se tiene que cumplir que $\Gamma^1(\sigma), ..., \Gamma^{r-1}(\sigma)$ son conexos, esto implica que

$\Gamma^r_C \subseteq \Gamma^1_C \cap .... \cap \Gamma^{r-1}_C$ y $\Gamma^r_C = \Gamma^1_C \cap .... \cap \Gamma^{r-1}_C \cap \Gamma^r_C$, así pues,

$$\begin{aligned}
\Pr(\Gamma^r_C) &= \Pr(\Gamma^r_C \cap \Gamma^{r-1}_C \cap ... \cap \Gamma^1_C) \\
&= \Pr(\Gamma^r_C | \Gamma^{r-1}_C \cap ... \cap \Gamma^1_C) \cdot \Pr(\Gamma^{r-1}_C \cap ... \cap \Gamma^1_C) \\
&= \Pr(\Gamma^r_C | \Gamma^{r-1}_C) \cdot \Pr(\Gamma^{r-1}_C \cap ... \cap \Gamma^1_C) \\
&= \Pr(\Gamma^r_C | \Gamma^{r-1}_C) \cdot \Pr(\Gamma^{r-1}_C | \Gamma^{r-2}_C \cap ... \cap \Gamma^1_C) \cdot \Pr(\Gamma^{r-2}_C \cap ... \cap \Gamma^1_C) \\
&= \Pr(\Gamma^r_C | \Gamma^{r-1}_C) \cdot \Pr(\Gamma^{r-1}_C | \Gamma^{r-2}_C) \cdot \Pr(\Gamma^{r-2}_C \cap ... \cap \Gamma^1_C) \\
&= \Pr(\Gamma^r_C | \Gamma^{r-1}_C) \cdot \Pr(\Gamma^{r-1}_C | \Gamma^{r-2}_C) \cdot ... \cdot \Pr(\Gamma^2_C | \Gamma^1_C) \cdot \Pr(\Gamma^1_C).
\end{aligned}$$

Para terminar la prueba, note que si k es tal que $2 \leq k \leq r$, entonces $\Pr(\Gamma^k_C | \Gamma^{k-1}_C) = \left(\frac{1}{2}\right)^{3^k-1}$, donde $3^k - 1$ es el número de parejas de vértices internos de $\Gamma^k(\sigma)$, pues como ya se dijo antes, si uno asume que $\Gamma^{k-1}(\sigma)$ es conexo, basta con asegurar que cada pareja de vértices internos de $\Gamma^k(\sigma)$ obtiene el mismo valor al lanzar una moneda para que $\Gamma^k(\sigma)$ sea conexo, y como evento lo anterior equivale a la intersección de $3^k - 1$ eventos independientes, cada uno de probabilidad $\frac{1}{2}$. Así que

$$\Pr(\Gamma^r_C) = \Big(\prod_{k=2}^{r} \Pr(\Gamma^k_C | \Gamma^{k-1}_C) \Big) \cdot \Pr(\Gamma^1_C) = \left(\frac{1}{2}\right)^{\sum_{k=2}^{r} 3^k - 1} \cdot \left(\frac{1}{2}\right)^{3^1 - 1} = \left(\frac{1}{2}\right)^{\sum_{k=1}^{r} 3^k - 1}.$$

□

Proposición 6.0.2. *Si σ es una arista cualquiera y Z es una variable aleatoria discreta tal que $Z = r$ si $\Gamma^{r-1}(\sigma)$ es conexo pero $\Gamma^r(\sigma)$ es disconexo ($\Gamma^0(\sigma)$ se define como σ), entonces el valor esperado de Z es menor o igual que 3, es decir,*

$$E[Z] \leq 3.$$

Demostración. Utilizando la notación del lema anterior, se tiene que $\Gamma^{r-1}_C \cap \Gamma^r_D$ es el evento en el que $\Gamma^{r-1}(\sigma)$ es conexo pero $\Gamma^r(\sigma)$ es disconexo (Γ^r_D es el evento: $\Gamma^r(\sigma)$ es disconexo); por lo que si $2 \leq r$, entonces

$$\begin{aligned}
\Pr[Z=r] &= \Pr(\Gamma_C^{r-1} \cap \Gamma_D^r) \\
&= \Pr(\Gamma_D^r | \Gamma_C^{r-1}) \cdot \Pr(\Gamma_C^{r-1}) \\
&= \Pr(\Gamma_D^r | \Gamma_C^{r-1}) \cdot \left(\frac{1}{2}\right)^{\sum_{k=1}^{r-1} 3^k - 1} && \text{Lema 6.0.1} \\
&= \Pr((\Gamma_C^r)^{\mathsf{c}} | \Gamma_C^{r-1}) \cdot \left(\frac{1}{2}\right)^{\sum_{k=1}^{r-1} 3^k - 1} && (\Gamma_C^r)^{\mathsf{c}} = \Gamma_D^r \\
&= \left(1 - \Pr(\Gamma_C^r | \Gamma_C^{r-1})\right) \cdot \left(\frac{1}{2}\right)^{\sum_{k=1}^{r-1} 3^k - 1} \\
&= \left(1 - \left(\frac{1}{2}\right)^{3^r - 1}\right) \cdot \left(\frac{1}{2}\right)^{\sum_{k=1}^{r-1} 3^k - 1}. && \text{Lema 6.0.1}
\end{aligned}$$

Para simplificar la última expresión observe que $1 - \left(\frac{1}{2}\right)^{3^r-1} = 1 - \frac{1}{2^{3^r-1}} = \frac{2^{3^r-1}-1}{2^{3^r-1}}$, y además

$$\begin{aligned}
\sum_{k=1}^{r-1} 3^k - 1 = \left(\sum_{k=1}^{r-1} 3^k\right) - (r-1) \quad &= \left(\sum_{k=0}^{r-1} 3^k\right) - [(r-1) + 3^0] \\
= \left(\frac{3^r - 1}{2}\right) - r \qquad &= \frac{3^r - 2r - 1}{2} \qquad \text{(progresión geométrica).}
\end{aligned}$$

Al sustituir estos valores en el cálculo de $\Pr[Z=r]$, se obtiene que

$$\begin{aligned}
\Pr[Z=r] &= \left(1 - \left(\frac{1}{2}\right)^{3^r-1}\right) \cdot \left(\frac{1}{2}\right)^{\sum_{k=1}^{r-1} 3^k - 1} \\
&= \left(\frac{2^{3^r-1}-1}{2^{3^r-1}}\right) \cdot \left(\frac{1}{2}\right)^{\frac{3^r - 2r - 1}{2}}.
\end{aligned}$$

La última expresión se reduce aplicando operaciones elementales para así concluir que si $2 \leq r$, entonces

$$\Pr[Z=r] = \frac{1}{2^{3^r - 2r - 1}}. \tag{6.1}$$

Si $r = 1$, entonces $\Pr[Z=1] = \Pr(\Gamma_D^1) = \frac{3}{4}$.

El valor esperado de Z, es por definición $E[Z] = \sum r \cdot p(r)$, donde $p(r) = \Pr[Z=r]$. En lugar de calcular el valor de $E[Z]$, se optará por dar una cota superior, para dar dicha cota es necesario probar el siguiente par de afirmaciones :

(1) $\sum_{r=2}^{\infty} \frac{r}{2^r} = \frac{3}{2}$.

(2) Para toda $r \in \mathbb{N}$ tal que $2 \leq r$, se cumple que $\frac{1}{2^{3^r-2r-1}} \leq \frac{1}{2^r}$.

Prueba de (1). Para probar que $\sum_{r=2}^{\infty} \frac{r}{2^r} = \frac{3}{2}$, primero se probará que $\sum_{r=1}^{\infty} \frac{r}{2^r} = 2$. Para ello considere las siguientes igualdades

$$\begin{aligned} S &= \sum_{r=0}^{\infty} \frac{r}{2^r} \\ &= \sum_{r=1}^{\infty} \frac{r}{2^r} \\ &= \sum_{r=0}^{\infty} \frac{r+1}{2^{r+1}}. \end{aligned}$$

Por lo que

$$\begin{aligned} 2S - S &= \sum_{r=0}^{\infty} \frac{r+1}{2^r} - \sum_{r=0}^{\infty} \frac{r}{2^r} \\ &= \sum_{r=0}^{\infty} \frac{1}{2^r} = 2. \end{aligned}$$

Esta última igualdad es bastante conocida por lo que se omite su prueba.
Teniendo que $\sum_{r=1}^{\infty} \frac{r}{2^r} = 2$, se concluye que $\sum_{r=2}^{\infty} \frac{r}{2^r} = 2 - \frac{1}{2} = \frac{3}{2}$.

Prueba de (2). Considere la función $f(r) = 3^r - 3r - 1$, f cumple que $0 \leq f(r)$ si $2 \leq r$, pues note que $0 \leq f(2) = 2$ y f es creciente, ya que si $2 \leq r$, se tiene que

$$\begin{aligned} 3 &\leq 2 \cdot 3^r \\ 0 &\leq 2 \cdot 3^r - 3 \\ 0 &\leq 3 \cdot 3^r - 3^r - 3 \\ 0 &\leq 3^{r+1} - 3^r - 3 \\ 0 &\leq 3^{r+1} - 3^r - 3 \\ -3r - 1 &\leq 3^{r+1} - 3^r - 3 - 3r - 1 \\ 3^r - 3r - 1 &\leq 3^{r+1} - 3 - 3r - 1 \\ 3^r - 3r - 1 &\leq 3^{r+1} - 3(r+1) - 1 \\ f(r) &\leq f(r+1), \end{aligned}$$

lo que prueba que f es creciente (basta con hacer la prueba únicamente con naturales), y como $2 \leq f(2)$, entonces $0 \leq f(r)$ si $2 \leq r$; de esto se sigue que $r \leq 3^r - 2r - 1$, y como $\frac{1}{2} < 1$, entonces $\frac{1}{2^{3^r-2r-1}} \leq \frac{1}{2^r}$, como se afirmó.

Finalmente, se tiene que

$$E[Z] = \sum r \cdot p(r) = \frac{3}{4} + \sum_{r=2}^{\infty} \frac{r}{2^{3^r-2r-1}} \leq \frac{3}{4} + \sum_{r=2}^{\infty} \frac{r}{2^r} = \frac{3}{4} + \frac{3}{2} = \frac{9}{4} \leq 3.$$

□

Capítulo 7

Conclusiones

7.1. Conclusiones

Este trabajo representa un primer paso en el estudio de los algoritmos distribuidos y aleatorios a través de la topología combinatoria. En el capítulo 4 se dio la definición del modelo operacional y combinatorio del protocolo aleatorio de instantáneas inmediatas. Después en el capítulo 5 se estudiaron algunas propiedades de conexidad del complejo de protocolo asociado al modelo combinatorio y en capítulo 6 se explicó como el complejo (al que se denominó $\Gamma(\sigma)$) que resulta al elegir una faceta por cada una de las pseudoesferas de las facetas del complejo del modelo determinista se desconecta en un número finito de rondas.

La descripción y el estudio de las propiedades topológicas del complejo de protocolo del modelo aleatorio de instantáneas inmediatas resultaron ser sencillos gracias al conocimiento que se tiene de antemano sobre el modelo determinista, sin embargo hay algunas nociones, como la de solubilidad, que no fueron analizadas en este trabajo, aunque se habló de relajar algunas condiciones (como la terminación con probabilidad mayor que cero), no se llegó a una definición que involucre una relación directa entre el complejo de protocolo y el complejo de salida.

7.2. Trabajo futuro

Quedan muchas interrogantes después de presentar este primer acercamiento al estudio de algoritmos aleatorios como lo es, por ejemplo, establecer una noción de solubilidad parecida a la del caso determinista. En el capítulo 4 se suposo que el resultado de lanzar una moneda era justo y que los únicos valores que se podían obtener eran 0 y 1, estas condiciones se podrían extender de la siguiente manera, en primer lugar, en [2] se describe un tipo de moneda que cumple la condición que dado un *paramétro de acuerdo* δ, la probabilidad de que todos los procesos obtengan el mismo valor (0 o 1) es al menos δ, este parámetro resulta útil para hacer más eficiente un algoritmo como el consenso y también para enfrentar a un *adversario fuerte*, posiblemente reduciendo el número de rondas esperado para solucionar una tarea; para aumentar el número de valores aleatorios que se pueden obtener al lanzar una moneda se podría recurrir a la definición general de pseudoesfera, sin embargo, la pseudoesfera en general no es isomorfa a la esfera combinatoria, por lo que se tendría que analizar con cuidado las repercusiones de esta modificación.

Finalmente, en el capítulo 6 se hace un estudio de la conexidad esperada en rondas subsecuentes para dos procesos, hay dos direcciones en las que se puede continuar dicho estudio, una de ellas es aumentar el número de procesos y por lo tanto la dimensión del complejo de protocolo, y preguntarse por la probabilidad de que dicho complejo se desconecte en un número finito de rondas; la otra dirección está relacionada con la conexidad en dimensiones superiores, es decir, preguntarse cuál es la probabilidad de que el complejo de protocolo sea $k-$conexo en una cierta ronda r.

Apéndice A

Nociones básicas de probabilidad.

En este apéndice se presentan algunos conceptos de la teoría de probabilidad que se usan a lo largo del texto y que han sido transcritos de [15].

Definición A.0.1. (Espacio de probabilidad) Un espacio de probabilidad es una terna $(\Omega, \mathscr{F}, P)$ en donde Ω es un conjunto arbitrario, $\mathscr{F}$ es una familia de subconjuntos de Ω y P es una función definida sobre $\mathscr{F}$, que cumplen las siguientes condiciones
$\mathscr{F}$ debe cumplir

(S1). $\Omega \in \mathscr{F}$.

(S2). Si $A \in \mathscr{F}$ entonces $A^c \in \mathscr{F}$.

(S3). Si $A_1, A_2, ... \in \mathscr{F}$, entonces $\bigcup_{n=1}^{\infty} A_n \in \mathscr{F}$.

Mientas que P satisface

(P1) $P(A) \geq 0$.

(P2) $P(\Omega) = 1$.

(P3) $P(\bigcup_{k=1}^{\infty} A_k) = \sum_{k=1}^{\infty} P(A_k)$ cuando $A_1, A_2, ...$ son ajenos dos a dos.

A los elementos de $\mathscr{F}$ se les llama *eventos.*

Definición A.0.2. (Probabilidad condicional) Sean A y B dos eventos y supongamos que B tiene probabilidad estrictamente positiva. La probabilidad condicional del evento A, dado el evento B, se denota por el símbolo $P(A|B)$ y se define como el cociente

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

El concepto de independencia es una forma de incorporar al cálculo de probabilidades la no afectación de la ocurrencia de un evento sobre la probabilidad de otro. Es un concepto importante que se deriva de observaciones de situaciones reales y su utilización reduce considerablemente el cálculo de probabilidades.

Definición A.0.3. (Independencia de eventos) Se dice que los eventos A y B son independientes si se cumple la igualdad

$$P(A \cap B) = P(A)P(B). \tag{A.1}$$

Bajo la hipótesis adicional de que $P(B) > 0$, la identidad A.1 puede escribirse como $P(A|B) = P(A)$ y esto significa que la ocurrencia del evento B no afecta a la probabilidad de A. Análogamente, cuando $P(A) > 0$ la identidad A.1 se puede escribir como $P(B|A) = P(B)$, es decir, la ocurrencia del evento A no cambia a la probabilidad de B.

Consideremos que tenemos un experimento aleatorio cualquiera junto con un espacio de probabilidad asociado $(\Omega, \mathscr{F}, P)$.

Definición A.0.4. (Variable aleatoria) Una variable aleatoria es una transformación X del espacio de resultados Ω al conjunto de números reales, esto es,

$$X : \Omega \to \mathbb{R},$$

tal que para cualquier número real x,

$$\{\omega \in \Omega | X(\omega)\} \in \mathscr{F}.$$

Definición A.0.5. (Variable aleatoria discreta) Decimos que una variable aleatoria es *discreta* cuando el conjunto de valores que ésta toma es un conjunto discreto, es decir, un conjunto finito o numerable.

Definición A.0.6. (Función de probabilidad) Sea X una variable aleatoria discreta con valores $x_0, x_1, ...$ La *función de probabilidad* de X, denotada por $f(x) : \mathbb{R} \to \mathbb{R}$, se define como sigue

$$f(x) = \begin{cases} P(X = x) & \text{si } x = x_0, x_1, ... \\ 0 & \text{otro caso.} \end{cases}$$

En palabras, la función de probabilidad es simplemente aquella función que indica la probabilidad en los distintos valores que toma la variable aleatoria.

Definición A.0.7. (Esperanza de una variable aleatoria discreta) Sea X una variable aleatoria discreta con función de probabilidad $f(x)$. La esperanza de X se define como

$$E(X) = \sum_x f(x)x,$$

suponiendo que esta suma es absolutamente convergente, es decir, cuando la suma de los valores absolutos es convergente.

La esperanza de una variable aleatoria es entonces un número que indica el promedio ponderado de los diferentes valores que la variable puede tomar. A la esperanza se le conoce también con los nombre de *media, valor esperado* o *valor promedio.*

Bibliografía

[1] Aspnes James. Randomized protocols for asyncronous consensus. Distributed Computing, 16(2-3):165–176, Sept. 2003.

[2] Aspnes James, Attiya Hagit, Censor Keren. Combining Shared Coin Algorithms. Journal of Parallel and Distributed Computing 70(3):317–322, March 2010.

[3] Attiya Hagit, Welch Jennifer. Distributed Computing: Fundamentals, Simulations, and Advanced Topics. Second Edition, John Wiley and Sons, 2004.

[4] Borowsky Elizabeth, Gafni Eli. A Simple Algorithmically Reasoned Characterization of Wait-free Computation. In Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing, pages 189–198, 1997.

[5] Kozlov Dmitry N. Combinatorial algebraic topology. Algorithms and computation in Mathematics, vol. 21. New York, Heidelberg: Springer; 2007.

[6] Kozlov Dmitry N. Chromatic subdivision of a simplicial complex. Homology, homotopy and applications. 2012;14(2):197–209.

[7] Herlihy Maurice, Shavit Nir. The topological structure of asynchronous computability. J ACM 1999;46(6):858–923.

[8] Herlihy Maurice, Shavit Nir. The art of multiprocessor programming. New York, NY, USA: Morgan Kaufmann; 2008.

[9] Herlihy Maurice, Kozlov Dmitry N, Rajsbaum Sergio. Distributed Computing Through Combinatorial Topology. San Francisco, CA, USA: Morgan Kaufmann; 2013.

[10] Herlihy Maurice, Rajsbaum Sergio. The topology of distributed adversaries. Distributed Computing 2013; 26(3):173–192.

[11] Fischer Michael J., Lynch Nancy A., Paterson Michael S. Impossibility of distributed consensus with one faulty process. Journal of the ACM, 32(2):374-382, April 1985.

[12] Lynch Nancy A. Distributed Algorithms. San Francisco, CA, USA: Morgan Kaufmann; 1996.

[13] Norman Gethin. Analysing Randomized Distributed Algorithms. In: Baier C., Haverkort B.R., Hermanns H., Katoen JP., Siegle M. (eds) Validation of Stochastic Systems. Lecture Notes in Computer Science, vol 2925. Springer, Berlin, Heidelberg, 2004.

[14] Prieto Carlos. Topología básica. Segunda Edición, Fondo de Cultura Económica, 2013.

[15] Rincón Luis. Introducción a la probabilidad. Las Prensas de Ciencias, 2014.

[16] Saks Michael, Shavit Nir, Woll Heather. Optimal time randomized consensus—making resilient algorithms fast in practice. In Proceedings of the Second Annual ACM-SIAM Symposium on Discrete Algorithms, pages 351–362, San Francisco, California, 28–30 January 1991.