



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

EL PROBLEMA DE LA CONEXIDAD
EN EL MODELO COMPUTACIONAL NUMBER-IN-HAND

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MATEMÁTICO

ANTONIO ANDRÉS LIZAMA ORELLANA

PROFESOR GUÍA:
IVÁN RAPAPORT ZIMERMANN.

MIEMBROS DE LA COMISIÓN:
MARTÍN MATAMALA VÁSQUEZ
JOSÉ SOTO SAN MARTÍN

SANTIAGO DE CHILE
2013

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL MATEMÁTICO
POR: ANTONIO LIZAMA ORELLANA
FECHA: 18/12/2013
PROF. GUÍA: IVÁN RAPAPORT Z.

EL PROBLEMA DE LA CONEXIDAD
EN EL MODELO COMPUTACIONAL NUMBER-IN-HAND

La presente memoria se enmarca en el contexto de la computación distribuida. Esta es un área de las ciencias de la computación relativamente reciente, que surge ante la necesidad de un nuevo paradigma de computación, capaz de trabajar con cantidades masivas de datos, como son, por ejemplo, las redes sociales.

En particular, se estudia la complejidad del problema CONEXIDAD de un grafo $G = (V, E)$ en el modelo computacional *number-in-hand*. Este problema consiste en decidir si un grafo G es o no conexo. Por otro lado, a grandes rasgos, la complejidad que se considera aquí mide el largo de los mensajes (en bits) que los nodos deben comunicar para decidir CONEXIDAD.

En primer lugar, se demuestra que la complejidad, en el caso en que el grafo G es arbitrario, es al menos $f(n)$, donde $f(n) = \log n - (\log \log n + 1 + \log n/n)$. Sin embargo, esta fórmula no aporta información alguna cuando el grafo G posee $n < 27$ nodos, pues $f(n) \leq 1$ para tales valores. Es decir, indica que la cantidad de bits que se tienen que comunicar es al menos ≤ 1 , lo que es evidente. Por esta razón, se profundiza el estudio analizando grafos pequeños, y se demuestra que 1 bit no es suficiente para decidir el problema en tal caso.

Por otro lado, la cota expuesta se obtiene a partir de una reducción que construye un grafo de grado no acotado. Por lo tanto, $f(n)$ puede ser poco ajustada para la familia de grafos de grado acotado. Así pues, se complementa el trabajo restringiéndose a esta clase de grafos, con el fin de saber si en tal caso existe una mejor cota para la complejidad de CONEXIDAD en el modelo *number-in-hand*. Usando técnicas de complejidad comunicacional se encuentra una cota inferior de $\Omega(\log n)$. Más aún, se demuestra que esta cota es ajustada para esta clase de grafos.

AGRADECIMIENTOS

En primer lugar, quiero agradecer a mi familia. A mis hermanas por la paciencia, a mi sobrino Martín por la alegría que ha brindado a nuestro hogar desde su llegada, y en forma muy especial a mis padres por su incondicional apoyo, confianza y amor.

Agradezco también a los miembros de mi comisión examinadora. A los profesores Martín Matamala y José Soto por sus muy útiles comentarios brindados durante la confección de este trabajo. Agradezco particularmente al profesor Iván Rapaport por ayudarme a mantener la calma durante todo el proceso, y por todas las ideas en los momentos donde la inspiración se hacía escasa.

Doy las gracias a mis amigos de la carrera. Al Carastu, Pablopems y el Tish por haberse dado el tiempo de sentarse a mirar la pizarra junto a mi. A Marrano y Orly por ayudarme en el primer paso del trabajo. A Benjamin Button por haber estado siempre ahí. En particular, me gustaría darle las gracias a quien fuera un pilar fundamental para estar aquí hoy, por acompañarme durante el momento más complicado de la carrera, el Bambino.

Agradezco al Benja Palacios, Alf, Johnson, Figura y Bambino por los buenos viajes que compartimos. Al Kiko, Elestric, Papo, Angel y a todos los mencionados previamente por todas las risas y buenos momentos que compartimos. Al DIM United por la excelente campaña que vivimos llegando a la final.

Me gustaría agradecer también a todos los integrantes del Desastre F.C. por todos esos Domingos de pichangas, e inolvidables Algarrobo. En particular, al Pelao por la gran amistad que hemos construido y al Rod, mi compañero de viaje. También agradezco a Nelsón por ser un gran amigo.

Agradezco a Eterin y Sandra Gaez por toda su buena onda y disposición, Oscar Mori por paciencia con tantas impresiones, y a Gladys por toda la paciencia y buena voluntad. Además agradezco al profe Lema por haberme permitido ser su auxiliar por tanto tiempo.

Finalmente, agradezco a las instituciones que financiaron este trabajo: Programa Basal Centro de Modelamiento Matemático y Proyecto Fondecyt 1130061.

Tabla de contenido

1. Introducción	1
1.1. Motivación	1
1.2. Organización	1
2. Preliminares	3
2.1. Modelos computacionales para procesar gran número de datos	3
2.1.1. Modelo <i>semi-streaming</i>	3
2.1.2. Modelo <i>number in hand</i>	4
2.1.3. Comparación de los modelos	5
2.2. El problema CONEXIDAD	6
2.2.1. Estado del arte	7
3. Caso general	9
3.1. Particiones de $[n]$ v/s $\#CC$	10
3.2. Reducción de $\#CC$ al problema de CONEXIDAD	10
3.3. Una cota para el caso general	12
4. Grafos de tamaño pequeño	14
4.1. Grafos de $n = 4$ nodos	14
4.2. Grafos de $n = 5, 6, \dots, 26$ nodos	29
5. Grafos de grado máximo acotado	31
5.1. Complejidad comunicacional	31
5.2. Grado máximo $d = 2$	32
5.2.1. Caso general	33
5.2.2. Caso particular	36
5.3. Grado máximo k fijo	38
6. Conclusiones y trabajo futuro	39
Bibliografía	40

Capítulo 1

Introducción

Este capítulo introductorio tiene dos objetivos. En primer lugar, en la Sección 1.1 se presenta la motivación tras este trabajo. En segundo lugar, la organización del mismo es presentada en la Sección 1.2.

1.1. Motivación

Uno de los principales objetos matemáticos utilizados para la modelación corresponde a los grafos. Su uso resulta natural para modelar redes, refiriéndose por esto a un conjunto de elementos que tienen algún tipo de conexión.

En los últimos tiempos, ha surgido una gran cantidad de redes masivas, es decir, de gran escala. Como ejemplo característico de esto se encuentra la *web*, donde cada nodo representa una página web, y si entre dos de ellas existe un hipervínculo, entonces hay una arista en el grafo que los une. Otro clásico ejemplo es la guía de números telefónicos; en este caso cada número de teléfono constituye un nodo, y la realización de una llamada entre dos de ellos se representa mediante una arista en el grafo.

El tipo de redes descrito en el párrafo anterior se presenta cada vez con mayor frecuencia, y en diversos tipos de aplicaciones/situaciones. Es por esto que su estudio ya no corresponde sólo a uno de carácter teórico, sino que también práctico, a través de implementación computacional. Sin embargo, su gran tamaño imposibilita mantener toda la estructura en memoria principal durante todo el proceso de cálculo. Por esta razón, ha sido necesario introducir nuevos paradigmas de computación que evadan esta problemática, como son la computación distribuida ó el modelo streaming, entre otros.

El presente trabajo se enmarca en el primero de estos, en particular, en el modelo denominado *number-in-hand*. El objetivo que se busca alcanzar es encontrar una cota inferior para la cantidad mínima de recursos necesarios para poder decidir el problema de conexidad de un grafo G en este modelo. En este contexto, la cantidad de recursos se entiende como el número de bits que son comunicados desde un procesador de la red al resto.

1.2. Organización

En primer lugar, en el Capítulo 2 se introducen los conceptos básicos necesarios para el entendimiento del trabajo realizado. Esto es, se define el modelo computacional *number-in-*

hand. Adicionalmente, se presenta de manera formal el problema estudiado, denominado CONEXIDAD, y el estado del arte del mismo. Luego, en el Capítulo 3 se analiza el problema para un grafo G cualquiera. Se demuestra que la complejidad del problema en el modelo *number-in-hand* en tal caso es al menos $f(n)$, donde $f(n) = \log n - (\log \log n + 1 + \log n/n)$.

Esta fórmula posee dos debilidades: lo primero, es que no brinda información útil cuando se estudian grafos pequeños, refiriéndose por esto a grafos de tamaño a $n < 27$, pues en tal caso se tiene que $f(n) \leq 1$. En segundo lugar, para obtener esta cota, se emplea una reducción que genera un grafo de grado máximo no acotado, por lo que esta cota puede no ser buena para grafos de grado acotado. Así pues, surge la necesidad de estudiar estas dos deficiencias en profundidad. La primera es cubierta en el Capítulo 4, donde se demuestra que, en realidad, 1 bit no es suficiente para decidir el problema. Por otro lado, el Capítulo 5 se enfoca en el estudio en la familia de grafos de grado máximo acotado. Haciendo uso de una herramienta muy útil para encontrar cotas en computación distribuida, denominada complejidad comunicacional, se concluye que efectivamente $f(n)$ no es la mejor cota. Se demuestra que la complejidad del problema CONEXIDAD en el modelo *number-in-hand*, restringido a la clase de grafos de grado máximo acotado, es $\Omega(\log n)$.

Capítulo 2

Preliminares

En este capítulo se introducen los conceptos necesarios para el desarrollo del presente trabajo. En primer lugar, en la Sección 2.1, se introducen dos de los principales modelos de computación para trabajar con grandes cantidades de datos. Adicionalmente, se realiza una breve comparación de los mismos.

Posteriormente, en la Sección 2.2, se presenta formalmente el problema a estudiar. Este consiste en decidir si un grafo G es conexo o no en uno de los modelos aludidos en el párrafo anterior, denominado *number-in-hand*. Además, se muestran los principales resultados relacionados con este problema existentes en la literatura.

2.1. Modelos computacionales para procesar gran número de datos

En lo que sigue, una red es representada mediante un grafo $G = (V, E)$. Se asume que los nodos del grafo están etiquetados por el conjunto $V = \{1, \dots, n\}$, y las aristas por el conjunto $E = \{e_1, \dots, e_m\}$. Además, como parámetro de los modelos, se encuentra el tamaño $|V|$ del grafo.

2.1.1. Modelo *semi-streaming*

Este modelo ha sido introducido por Feigenbaum *et al.* en [11]. Como su nombre lo indica, procesa la información de manera *streaming*. Esto es, la entrada no es un gran conjunto de datos que es recibido al mismo tiempo, sino que se recibe por segmentos. Cuando los problemas que se estudian tienen como entrada un grafo $G = (V, E)$, estos segmentos corresponden a las aristas del grafo. Adicionalmente, se dispone de un espacio de memoria global, la que es modificable por todos los nodos durante todo el proceso.

A medida que la entrada es procesada, los nodos, que disponen de un poder de cálculo infinito, pueden realizar operaciones internas para luego modificar la memoria global. De esta manera, al finalizar la computación, la información escrita en la memoria global debe corresponder al valor de $f(x)$, donde x es la entrada (conjunto de aristas del grafo), y f una función (en general relacionada a la topología de la red).

Más rigurosamente, el modelo se define como sigue.

Definición 2.1. Sea $G = (V, E)$ un grafo, donde $E = \{e_1, \dots, e_m\}$. Un **graph stream** es

una secuencia de arcos $e_{i_1}, e_{i_2}, \dots, e_{i_m}$, tal que $e_{i_j} \in E$, e i_1, i_2, \dots, i_m es una permutación cualquiera de $[m]$.

Definición 2.2. Un algoritmo en el modelo *semi-streaming* es tal que su entrada es un graph stream, es decir, los arcos son revelados uno a la vez. A medida que estos son procesados, los nodos pueden realizar computos internos para luego modificar la memoria global.

La complejidad de un algoritmo en este modelo tiene 3 aristas, las cuales son:

1. La cantidad de bits $S(n)$ utilizados de memoria global.
2. La cantidad de pasadas $P(n)$ que puede realizar de manera secuencial sobre el stream de entrada.
3. El tiempo $T(n)$ requerido para procesar cada arista.

En la literatura este modelo es de interés cuando se considera $S(n) = O(n \log^k n)$.

2.1.2. Modelo *number in hand*

El modelo estudiado en la presente memoria, conocido en la literatura mediante el nombre de *number in hand* [3, 4, 19], se enmarca dentro de la computación distribuida (ver Definición 2.3). El nombre nace del hecho que en este modelo cada nodo/procesador tiene conocimiento únicamente de su propia entrada y desconoce la del resto de los miembros de la red.

Definición 2.3. La computación distribuida [18] es un área de las ciencias de la computación que estudia los sistemas distribuidos. Un sistema distribuido es un programa compuesto por varias componentes ubicadas en distintos computadores conectados mediante una red. Estos se coordinan mediante el traspaso de mensajes, con el fin de calcular cierta función $f(x_1, \dots, x_n)$ en conjunto, donde x_i es la entrada de cada uno de los computadores que conforman la red, $i \in \{1, \dots, n\}$.

En este modelo cada procesador v posee un identificador $ID(v)$ (que se asume como un número en $\{1, \dots, n\}$). La entrada x_i del nodo v_i corresponde a su identificador, y los de sus procesadores vecinos, es decir, $x_i = \{ID(u) \mid u \in N_G(v_i)\} \cup \{ID(v_i)\}$, donde $N_G(v_i)$ es el conjunto de vecinos del nodo v_i en G .

El objetivo es computar una función $f(x_1, \dots, x_n)$ (al igual que antes, típicamente alguna propiedad topológica de la red). Para poder lograr esto, se lleva a cabo un proceso de comunicación entre los miembros de la red, dividido en una serie de rondas. En cada una de ellas, un nodo v_i puede realizar tareas internas, y obtener un $output_i$. Tal como en el modelo anterior, el poder de cálculo interno de cada nodo es considerado infinito.

Existen tres variaciones de este modelo, dadas por el modo en que se lleva a cabo la comunicación de las salidas de cada:

1. *Modelo de pizarra*: todo procesador escribe su $output$ en un espacio de memoria global, la pizarra, visible para todos.
2. *Modelo de traspaso de mensaje*: un procesador v_i envía su mensaje $output_i$ a un destinatario específico v_j .

3. *Modelo con referee*: en este caso la red posee un nodo externo, etiquetado por $n + 1$, quien no recibe entrada. Todos los otros procesadores mandan su mensaje a este procesador adicional, denominado *referee*.

A este proceso se le denomina *protocolo*.

Nota.- Si se permite sólo una ronda de comunicación, y todos los nodos comunican su salida simultáneamente, entonces la primera y tercera variante del modelo son equivalentes.

Definición 2.4. En este último modelo son de interés las siguientes dos medidas de complejidad:

1. Número de rondas.
2. Tamaño de mensaje;

entendiendo por esto último, al máximo de los largos escritos en la pizarra durante un protocolo.

Es claro que si no se impone una restricción en el tamaño del mensaje que cada nodo puede escribir en la pizarra, basta con que cada uno escriba su vecindad completa para poder reconstruir cualquier red. De este modo, calcular el valor $f(x_1, \dots, x_n)$ se vuelve trivial. Lo interesante es encontrar la menor cantidad de bits que deben comunicar los nodos para obtener $f(x_1, \dots, x_n)$.

En lo que sigue, se trabaja con la primera de estas variantes, permitiendo sólo una ronda de comunicación. Por comodidad, se habla de un agente externo, léase *referee*, que posee acceso a todos los mensajes en la pizarra. Este agente, quién también posee un poder de cálculo ilimitado, es el encargo de computar la función f .

2.1.3. Comparación de los modelos

Una pregunta natural que surge al momento de estudiar diversos modelos es cómo se comparan entre ellos. Sin embargo, los modelos descritos en las Secciones 2.1.1 y 2.1.2 no son trivialmente comparables, es decir, no hay uno claramente más débil que otro. En esta sección se busca ejemplificar esto mediante dos problemas: uno que puede ser resuelto con cierta cantidad de bits en el modelo *semi-streaming*, y que, si bien, no se ha probado que no se pueda en el modelo *number-in-hand* con la misma cantidad de bits, se cree que no es factible. El segundo muestra el caso inverso.

G bipartito

En el modelo *semi-streaming* es posible determinar si un grafo es bipartito o no utilizando tan sólo $O(n \log n)$ bits, como lo muestran Fiegenbaum *et al.* en [11]. Ellos proponen un algoritmo para encontrar un emparejamiento en un grafo bipartito, para lo que, en particular, primero verifican que el grafo pertenezca a esta clase. Para lograr esto, es fundamental la manera en como es procesada la entrada, esto es, arista por arista.

A medida que se procesa la entrada, se mantiene registro de las componentes conexas del grafo, asociando a cada nodo un número. Además, se le agrega un signo que codifica la pertenencia del vértice al conjunto A ó B , donde $V = A \cup B$ en el caso en que G sea bipartito. Así, si es posible finalizar el proceso asignando signo a todos los nodos,

de manera que dos nodos adyacentes no posean el mismo signo, se dice que el grafo es bipartito. En caso contrario, se dice que no lo es.

Ahora bien, aunque no se ha demostrado que en el modelo *number-in-hand* este problema no se pueda decidir permitiendo a cada nodo enviar un mensaje de tamaño $O(\log n)$, es decir, que el *referee* disponga de $O(n \log n)$ bits, se cree que no es factible [3].

Por otro lado, en el modelo *number-in-hand* se puede reconstruir grafos de degenerancia acotada (ver Definición 2.5), permitiendo a cada nodo escribir un mensaje de largo $O(\log n)$.

Definición 2.5. Un grafo G se dice k -degenerado si todo subgrafo posee un nodo de grado a lo más k . La degenerancia es el menor k tal que G es k -degenerado.

G k -degenerado

Becker *et al.* demuestran que, dado un k fijo, es posible reconstruir un grafo k -degenerado en el modelo *number-in-hand* en una ronda, cuando los nodos pueden escribir un mensaje de largo $O(\log n)$ [4]. Su demostración hace uso del siguiente teorema [24]:

Teorema 2.6. *El siguiente sistema de ecuaciones simultáneas no posee soluciones enteras no-triviales, es decir, $(i_1, i_2, \dots, i_k) = (j_1, j_2, \dots, j_k)$ salvo permutación;*

$$i_1^p + i_2^p + \dots + i_k^p = j_1^p + j_2^p + \dots + j_k^p, \quad \forall 1 \leq p \leq k$$

La información que cada nodo v envía al *referee*, corresponde a la siguiente $k+2$ -tupla:

- Su identificador $ID(v)$.
- Su grado $d_G(v)$.
- Para cada entero p , $1 \leq p \leq k$, $\sum_{w \in N_G(v)} ID(w)^p$.

Luego, gracias al Teorema 2.6 sigue que, con estos mensajes, existe una decodificación única de los mensajes por parte del *referee*. De esta forma, él puede reconstruir el grafo. Nuevamente, aunque no se haya probado que en el modelo *semi-streaming* esto no pueda realizarse con una capacidad de memoria $O(n \log n)$, se cree fuertemente que no lo es.

2.2. El problema CONEXIDAD

Sea $G = (V, E)$ una red, donde $V = \{1, \dots, n\}$. Recordar que el modelo en el que se desenvuelve el trabajo, es el denominado *number-in-hand*. Además, salvo que se explicita de otra forma, se trabaja con protocolos de un ronda. Por otro lado, la información que cada nodo v_i recibe como entrada es $x_i = \{ID(u) \mid u \in N_G(v_i)\} \cup \{ID(v_i)\}$, es decir, v_i conoce su identificación y la de todos sus vecinos. Luego, cada nodo calcula su propio *output* (como antes, se supone poder de cálculo interno ilimitado), el cual escribe en una pizarra. Posteriormente, el *referee* recibe toda esta información, y con ella busca computar una cierta función f .

En este trabajo se estudia el problema CONEXIDAD. Esto es, decidir si un grafo G es conexo o no. La función característica de este problema está definida por:

$$f_{con}(x_1, \dots, x_n) := \begin{cases} 1 & \text{si } G \text{ es conexo} \\ 0 & \text{si no} \end{cases}$$

El objetivo es estudiar cuál es la cantidad mínima de recursos necesarios para que el *referee* pueda calcular esta función. Por cantidad de recursos necesarios se refiere al máximo de los tamaños de los *output* que los nodos escriben en la pizarra. En otras palabras, lo que se busca es conocer la complejidad del problema CONEXIDAD de un grafo G en el modelo computacional *number-in-hand*.

Este problema es de gran interés debido a sus diversas aplicaciones en múltiples áreas. A modo de ejemplo, se mencionan dos de ellas: White y Harary vinculan, de manera natural, el problema de conectividad con el de cohesión social [23]. En su artículo, los autores consideran a cada miembro de una red (grupo de personas) como un nodo, y si existe una relación entre ellos, se dice que la arista correspondiente se encuentra en el grafo construido. Como resulta natural esperar, sus resultados empíricos muestran que a mayor conectividad del grafo, mayor es la cohesión social, entendiendo a ésta como la contribución hecha por un miembro de la red, ya sea al sumarse o restarse a esta, con respecto a la permanencia del grupo como tal.

Por otro lado, Navigli y Mirella realizan un estudio experimental en el que, a través de medidas de conectividad de un grafo, atacan el problema de *Desambiguación del sentido de una palabra* [17]. Esto es, detectar el sentido de una palabra mediante el contexto. En su estudio, cada posible significado de la palabra en contexto constituye un nodo, y los arcos están definidos por dependencia entre ellos (e.g. sinónimos o antónimos). Luego, la estructura del grafo es esencial para asignar un nivel de “importancia” a cada nodo. Esta variable es medida con respecto a la conectividad del grafo, y su papel en ella. Sus resultados son comparados con sus pares, evidenciando una gran similitud.

2.2.1. Estado del arte

Este es un problema clásico dentro de la teoría de grafos. Como tal, ha sido ampliamente estudiado en modelos clásicos de computación, tanto de manera determinista [9, 10, 12, 13, 22], como probabilista [5]. Sin embargo, estos resultados no son aplicables en el marco del presente estudio ya que requieren almacenar el grafo completo en memoria durante toda la computación. Esto corresponde precisamente a lo que se pretende evitar al introducir nuevos paradigmas de computación como es la computación distribuida.

Ahora bien, en el marco de los modelos a gran escala (que procesan grandes cantidades de datos), también se ha desarrollado un interés por conocer la complejidad del problema en cuestión. Sin embargo, la mayoría de los resultados se enmarcan en el contexto del modelo *semi-streaming*. Una excepción es *forzada*¹ más adelante, al referirse a un modelo probabilista.

Un algoritmo en el modelo *semi-streaming* es presentado por Feigenbaum *et al.* [11]. Este necesita tan sólo una lectura de la entrada y requiere espacio $O(n \log n)$. En efecto, a

¹Originalmente también está planteado en modelos *streaming*, pero se puede adaptar al modelo *number-in-hand*.

medida que se procesa la entrada, la única información que se debe mantener almacenada en la memoria global, es la componente conexa a la que pertenece cada nodo. De esta manera, si una arista $e = uv$ es leída y las componentes conexas de estos nodos son distintas, entonces pasan a ser la misma. Al finalizar el proceso, si todo nodo pertenece a la misma componente conexa, entonces se dice que el grafo es conexo. Es claro que las medidas de complejidad, tanto de lectura de la entrada como de espacio, son las señaladas.

Por otro lado, Ahn *et al.* proponen un algoritmo probabilista para modelos *streaming* [1]. La novedad de esta propuesta corresponde a la incorporación de un aspecto hasta ese entonces no considerado en los análisis: la eliminación de aristas. Recordar que los grafos estudiados buscan modelar redes de gran tamaño como son, por ejemplo, las redes sociales. Ahora, este tipo de grafos tienen un carácter dinámico. Siguiendo con el ejemplo, una relación de amistad puede ser deshecha en el tiempo. Sin embargo, esta situación no había sido tomada en cuenta por trabajos previos a los de Jin Ahn *et al.* (salvo por Cormode & Muthukrishnan [8]). Su algoritmo requiere de espacio $O(n \text{ polylog } n)$ para decidir conexidad con alta probabilidad.

El modelo introducido en [1] hace uso de un método de reducción de dimensión. Para ello definen el concepto *esquema de un grafo* (simplemente esquema en lo que sigue), como una representación lineal de éste. Los autores prueban la existencia de una proyección del grafo en un esquema d -dimensional, donde $d \ll n^2$, tal que éste permite conocer propiedades del grafo con alta probabilidad.

Además, prueban que sus algoritmos son adaptables al modelo *semi-streaming*. Más aún, debido a que sus esquemas satisfacen linealidad, es posible adaptar sus algoritmos a otro modelo denominado *distributed streaming*. Este último consiste en segmentar la entrada en bloques, de manera que a cada procesador de la red le corresponde uno de ellos, el que es procesado de manera *streaming*. De esta manera, los resultados que se obtienen a través de los esquemas introducidos por Ahn *et al.*, aplican en el modelo *number-in-hand* (pues el modelo *distributed streaming* es más débil²).

Finalmente, el algoritmo que proponen para decidir conexidad usando esquemas, es análogo al presentado por Feigenbaum *et al.* [11]. En efecto, inicialmente se construye un bosque formado por todos los nodos del grafo. Luego, se procede como sigue: mediante un muestreo, se obtiene una arista. En caso de conectar dos componentes conexas distintas, se colapsan. De esta manera, con alta probabilidad el algoritmo entrega la respuesta correcta.

La adaptación al modelo *distributed streaming* induce un algoritmo de orden $O(n \log^3 n)$. Esto permite concluir, la existencia de un algoritmo probabilista en el modelo *number-in-hand*, para decidir conexidad de orden $O(n \log^3 n)$.

²Basta considerar como entrada de cada nodo en el modelo *number-in-hand* el segmento completo que será recibido de manera *streaming*.

Capítulo 3

Caso general

Este trabajo tiene como objetivo encontrar cotas para la complejidad del problema CONEXIDAD de un grafo G en el modelo *number-in-hand*. En primer lugar, se encuentra una cota para el tamaño del mensaje más largo escrito en la pizarra para poder decidir el problema en cuestión en el caso en que G es un grafo cualquiera. Esto se realiza en este capítulo, donde se concluye que la complejidad de CONEXIDAD en el modelo *number-in-hand* es al menos $\log n - (\log \log n + 1 + \log n/n)$.

Sin embargo, esta cota no aporta información para n pequeño (< 27), ya que en tal caso señala que la cantidad mínima de bits que debe escribir cada nodo es al menos una cantidad negativa ó 1. La primera situación carece de sentido, mientras que la segunda es obvia. Por otro lado, el procedimiento realizado en el caso general, requiere una reducción que construye un grafo de grado máximo no acotado. Así pues, es necesario profundizar el estudio enfocándose en éstas dos situaciones. Esto se lleva a cabo en los Capítulos 4 y 5, respectivamente.

Para estudiar el caso general, donde G es un grafo cualquiera, se comienza por mostrar una inyección entre las particiones de un conjunto con n elementos, y la cantidad de componentes conexas de un grafo con n nodos. Luego, se presenta una reducción del problema de conexidad al problema #CC (ver Definición 3.1).

Ahora bien, como el número de particiones de $[n] := \{1, 2, \dots, n\}$, conocido como Número de Bell (B_n), se ha estudiado ampliamente en la literatura [6, 7, 21], se obtiene de manera directa una cota inferior para el tamaño de mensaje más largo escrito en la pizarra para poder conocer la cantidad de componentes conexas de un grafo.

Finalmente, la reducción de CONEXIDAD a #CC, permite conocer una cota para la complejidad del problema CONEXIDAD en el modelo *number-in-hand*.

Es importante destacar que en todos los casos, tanto general como los 2 particulares, se demuestra que ésta cantidad crece con el tamaño del grafo n .

Definición 3.1. Se define el problema #CC como aquel que dado un grafo G , determina la cantidad de componentes conexas que éste posee.

Un argumento similar al expuesto en este capítulo es empleado por Salomon [20]. Sin embargo, en tal trabajo, el modelo de computación considerado es el modelo *semi-streaming*, por lo que sus resultados no se aplican en el marco del presente.

3.1. Particiones de $[n]$ v/s $\#CC$

En esta sección se prueba la existencia de una inyección entre las particiones de un conjunto con n elementos, y las componentes conexas de un grafo de n nodos.

La idea consiste en demostrar que dada una partición de $[n]$, se puede construir un grafo G asociado a dicha partición, de manera que la cantidad de componentes conexas de G sea igual al tamaño de la partición. Sigue de esta forma que para poder decidir $\#CC_n$ en el modelo *number-in-hand*, existe un nodo que escribe un mensaje de largo al menos $(\log B_n)/n$, donde $\#CC_n$ es el problema de determinar la cantidad de componentes conexas para grafos con n nodos, y B_n es el número de Bell.

Luego, en la siguiente sección se presenta el argumento clave de la demostración del caso general. Este corresponde a una reducción del problema CONEXIDAD al de conocer la cantidad de componentes conexas de un grafo G , es decir, al problema $\#CC$. Esto es, se demuestra que si existe un protocolo que decide CONEXIDAD en el modelo *number-in-hand*, entonces existe otro, del mismo orden, que resuelve $\#CC$. Ahora bien, como en esta sección se encuentra una cota inferior para la complejidad de este último, ésta será válida también para el problema CONEXIDAD.

El siguiente lema muestra una inyección entre las particiones de $[n]$ y las componentes conexas de un grafo con n nodos.

Lema 3.2. *Existe una inyección entre las particiones de un conjunto $A = \{1, 2, \dots, n\}$ y las componentes conexas de grafos con n vértices.*

Demostración. -

Sea $\mathcal{A} := \{A_1 = \{a_1^1, a_2^1, \dots, a_{n_1}^1\}, \dots, A_k = \{a_1^k, a_2^k, \dots, a_{n_k}^k\}\}$ una partición del conjunto $[n]$. A partir de \mathcal{A} se construye un grafo $G_{\mathcal{A}} = (V, E)$, con $|V| = n$, tal que las componentes conexas de G correspondan precisamente a los elementos de la partición \mathcal{A} . Así, es claro que por cada partición \mathcal{A} del conjunto $[n]$, se puede construir un grafo $G_{\mathcal{A}}$, de manera que la cantidad k de conjuntos que componen a la partición, sea igual a la cantidad de componentes conexas que posee el grafo $G_{\mathcal{A}}$. Este grafo se define como sigue:

- $V = \{a_1^1, \dots, a_{n_1}^1, \dots, a_1^k, \dots, a_{n_k}^k\}$ ($\sum_i n_i = n$)
- $E = \{a_j^i a_{j+1}^i \mid i \in [k], j \in [n_i - 1]\}$, donde $n_i = |A_i|$.

En la Figura 3.1 se ilustra la construcción del grafo $G_{\mathcal{A}}$. □

En consecuencia, se concluye que la complejidad de $\#CC$ en el modelo *number-in-hand* es al menos $\log B_n/n$.

3.2. Reducción de $\#CC$ al problema de conexidad

Dado el resultado de la sección anterior, es decir, que la complejidad de $\#CC$ en el modelo *number-in-hand* es al menos $(\log B_n)/n$, basta mostrar que si existe un protocolo \mathcal{P}_1 que resuelve CONEXIDAD, entonces existe otro protocolo \mathcal{P}_2 , del mismo orden, que resuelve $\#CC$. De esta manera, $(\log B_n)/n$ sería también una cota inferior de la complejidad de CONEXIDAD. El siguiente teorema muestra la existencia de tal reducción.

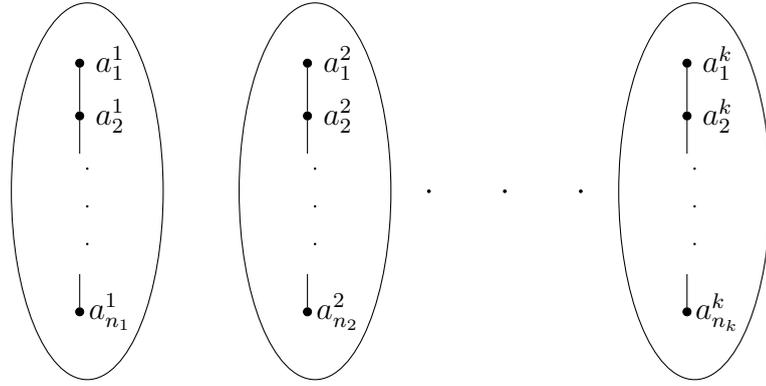


Figura 3.1: Inyección Particiones en CC.

Lema 3.3 ([26]). *Si el problema CONEXIDAD puede resolverse en el modelo number-in-hand con un largo de mensaje máximo $f(n)$, con $f(n) = O(n^c \text{polylog } n)$, entonces, también se puede resolver el problema de #CC en el mismo modelo utilizando $2f(n+2)$ bits.*

Demostración. -

Sea $G = (V, E)$ un grafo y \mathcal{P} un protocolo que decide el problema CONEXIDAD utilizando $f(n)$ bits en el modelo *number-in-hand*, donde $n = |V|$. Esto es, el largo máximo de los mensajes escritos por los nodos en la pizarra es $f(n)$. A partir de G se construye la familia de grafos

$$\mathcal{F} := \{G'_A = (V', E') \mid A \subseteq V, V' = V \cup \{a, b\}, E' = E \cup \{au, bw \mid u \in A, w \in V \setminus A\}\}$$

esto es, agregar dos nuevos vértices a y b , y conectar todo $v \in V$ a uno y sólo uno de ellos. Así, cada grafo G'_A dentro de la familia \mathcal{F} queda definido mediante la elección del conjunto $A \subseteq V$.

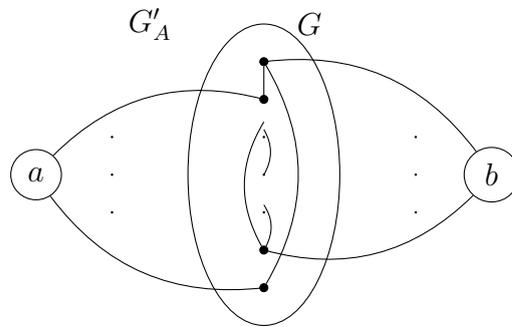


Figura 3.2: Grafo resultante de la reducción.

En base a esta familia de grafos \mathcal{F} , se construye un nuevo protocolo \mathcal{P}' que, dado un grafo $G = (V, E)$, encuentra la cantidad de componentes conexas que éste posee. El nuevo protocolo \mathcal{P}' es como sigue: cada nodo $v \in V$ escribe en la pizarra el mensaje $m_v = m_v^1 m_v^2$, donde m_v^1 corresponde al mensaje que escribiría en el protocolo \mathcal{P} sobre G'_A (ver Figura 3.2) en el caso de estar conectado con a , *i.e.*, $v \in A$, y m_v^2 es la respuesta que daría cuando $vb \in E'$, *i.e.*, $v \in V \setminus A$.

Sigue de la definición de \mathcal{P}' , que el largo del mensaje de cada nodo es $2f(n+2)$. Con esta información, el *referee* es capaz de saber si cada grafo $G'_A \in \mathcal{F}$ es conexo o no. En efecto, como se menciona previamente, un grafo $G'_A \in \mathcal{F}$ queda determinado de manera única por la elección de $A \subseteq V$. Luego, el *referee* debe barrer sobre todos los posibles subconjuntos A de nodos del grafo G . Para cada uno de ellos, considera como mensaje del nodo v m_v^1 si $v \in A$ y m_v^2 si $v \in V \setminus A$. Con esta secuencia de mensajes, y gracias a la existencia del protocolo \mathcal{P} , es posible determinar si G'_A es conexo o no.

Analizando la respuesta sobre todos los grafos de la familia \mathcal{F} es posible determinar el número de componentes conexas que posee el grafo G original. Para esto, basta notar que si G es un grafo con k componentes conexas, V_1, \dots, V_k , entonces G'_A es no conexo si y sólo si $A = \{\cup_{i \in I} V_i \mid I \subseteq \{1, \dots, k\}\}$. Por lo tanto, todo lo que debe hacer el *referee* es contar la cantidad de grafos G'_A tales que $\mathcal{P}(G'_A) = 0$ para poder encontrar k . \square

3.3. Una cota para el caso general

El Lema 3.2 permite concluir que si existe un protocolo \mathcal{P} que encuentre la cantidad de componentes conexas de un grafo con n nodos, entonces la complejidad de tal protocolo en el modelo *number-in-hand* es al menos $\log B_n/n$. En [20] Salomon prueba que $\forall n \geq 2$, $\log B_n > n \log n - n(\log \log n + 1 + \log n/n)$. Conectando esto al Lema 3.3, se concluye que cualquier protocolo \mathcal{P} en el modelo *number-in-hand* que resuelva CONEXIDAD, en el caso de un grafo G cualquiera, tiene complejidad al menos $\log n - (\log \log n + 1 + \log n/n)$.

Teorema 3.4. *La complejidad del problema CONEXIDAD, para un grafo G cualquiera, en el modelo computacional number-in-hand es al menos $\log n - (\log \log n + 1 + \log n/n)$.* \square

Es importante rescatar la importancia de este resultado. Esta cota señala que la complejidad de CONEXIDAD en el modelo *number-in-hand* crece con el tamaño del grafo, n . Más aún, de manera asintótica, el comportamiento es similar al de la función $g(n) := \log n$.

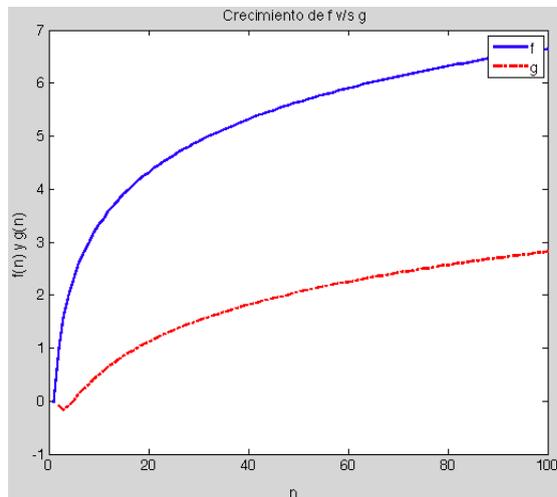


Figura 3.3: f v/s g - n pequeño.

Luego, para poder decidir este problema se requiere que al menos un nodo escriba un mensaje de largo al menos $\log n - h(n)$, donde $h(n)$ es $o(\log n)$. En la Figura 3.3 se compara el crecimiento de la función $f(n) := \log n - (\log \log n + 1 + \log n/n)$ con el de la función $g(n) = \log n$, para el caso n pequeño. Se puede apreciar que en tal caso existe una considerable diferencia entre la magnitud de ambas funciones.

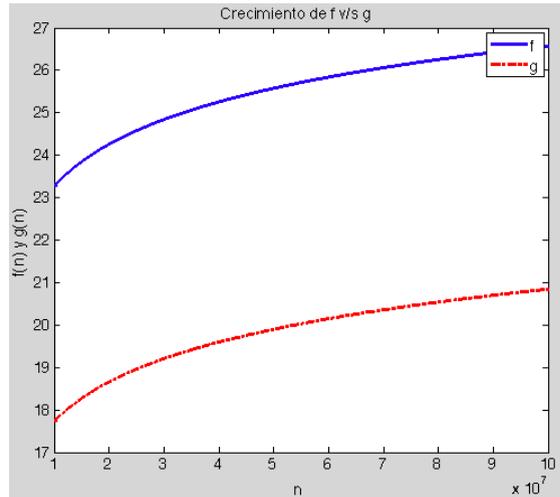


Figura 3.4: f v/s g - n grande.

Por otro lado, en la Figura 3.4, se muestra la misma comparación de la Figura 3.3, pero para n grande. Se muestra que, efectivamente en este caso el comportamiento de ambas funciones se asemeja bastante. Sin embargo, la diferencia que existe entre estas dos funciones para n pequeño se arrastra para todo n .

El resultado obtenido en esta sección, posee un par de deficiencias. En primer lugar, no aporta información en el caso n pequeño. Esto pues $f(n) \leq 1$ para $n < 27$. Por otro lado, la cota obtenida puede no ser la mejor en el caso de grafos de grado máximo acotado, pues en la reducción empleada en la Sección 3.2, el grafo G' construido pierde esta propiedad, es decir, es de máximo no acotado. Debido a esto, es que en principio, podría existir una cota que sea más ajustada para esta familia de grafos.

Por esta razón, resulta de interés estudiar cuál es la complejidad del problema en cuestión, cuando se restringe por un lado, a grafos de tamaño pequeño, y por otro a grafos de grado acotado. Esto se profundiza en los siguientes capítulos.

Capítulo 4

Grafos de tamaño pequeño

En el capítulo anterior, la siguiente cota inferior para la complejidad de CONEXIDAD en el modelo *number-in-hand* ha sido expuesta: $f(n)$, donde $f(n) = \log n - (\log \log n + 1 + \log n/n)$. Sin embargo, este resultado es poco útil cuando el tamaño del grafo es pequeño, entendiendo por esto $n < 27$, pues para tales n , $f(n) \leq 1$.

Por esta razón resulta necesario analizar este caso de manera particular. El resultado de este capítulo señala que en realidad 1 bit no es suficiente para poder decidir el problema, incluso cuando el grafo posee pocos nodos. Para hacer esto se considera primero el caso $n_0 = 4$, y luego se generaliza para $n = 5, 6, \dots, 27$.

La decisión de tomar $n_0 = 4$ es simple. Para $n = 3$ existe un protocolo trivial de 1 bit que decide el problema. En efecto, el bit que cada nodo v escriba en la pizarra, es la respuesta a una misma pregunta: ¿es $d_G(v) = 0$? Este protocolo, sin embargo, sólo es efectivo en el caso de $n = 3$, ya que únicamente en tal caso es posible caracterizar a los grafos no conexos como aquellos que poseen algún nodo aislado.

4.1. Grafos de $n = 4$ nodos

Se busca responder la siguiente pregunta: ¿será 1 bit suficiente para poder decidir conexidad de un grafo G con $n = 4$ nodos? ¡La respuesta es NO!, como muestra el Teorema 4.1. Un bit no es suficiente para decidir el problema CONEXIDAD en el modelo *number-in-hand*, ni siquiera cuando se trata de grafos con pocos nodos.

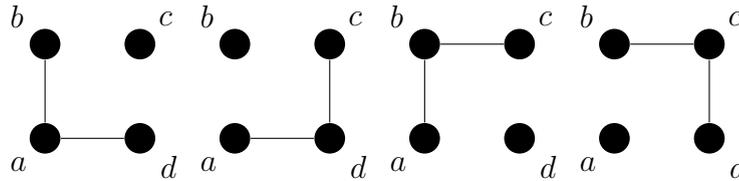
Teorema 4.1. *El problema de CONEXIDAD, restringido a grafos de 4 nodos, no se puede decidir en el modelo number-in-hand, cuando los nodos pueden escribir un mensaje de 1 bit.*

Demostración. -

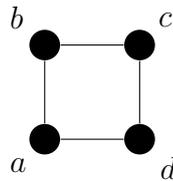
Suponga que existe un protocolo \mathcal{P} tal que decide el problema de conexidad para grafos con $|V| = 4$, y largo de mensaje igual a 1 bit.

Por la fórmula de Cayley's [16] se tiene que existen n^{n-2} árboles distintos con n nodos etiquetados. Luego, en el caso $n = 4$ sigue que hay 16 árboles distintos, exactamente el mismo número de posibles mensajes que pueden ser generados por un grafo G de 4 nodos, donde cada nodo es restringido a enviar un mensaje de un bit. Así, para que pueda existir un tal protocolo \mathcal{P} , debe haber al menos dos árboles que escriban el mismo mensaje

$x_1x_2x_3x_4$, $x_i \in \{0, 1\}$. En caso contrario, trivialmente un grafo no conexo genera un mismo mensaje que uno conexo. Más aún, no puede suceder tampoco que todo grafo no conexo escriba el mismo mensaje, ya que si tal fuese el caso, se tendría que los 4 grafos siguientes escriben el mensaje $abcd$:



Esto implica que el cuadrado entrega la misma respuesta $abcd$.



Por lo tanto, los árboles disponen de a lo más 14 mensajes distintos. Esto significa que o bien al menos dos pares de árboles escriben el mismo mensaje, ó una tripleta de ellos lo hace. Este hecho será utilizado más adelante en la demostración.

A continuación se prueba que si existe un par de árboles distintos que escriben un mismo mensaje, entonces hay un grafo conexo y uno no conexo que dan la misma respuesta. Esto se logra condicionando sobre la cantidad de aristas en común que tienen el par de árboles en cuestión.

Sean T_1 y T_2 dos árboles que bajo el protocolo \mathcal{P} , que resuelve el problema de conectividad con un bit en grafos de 4 nodos, entregan la misma respuesta $abcd$. Se utilizará el siguiente ordenamiento para referirse a los nodos de :

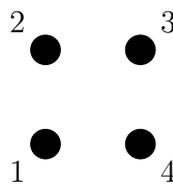


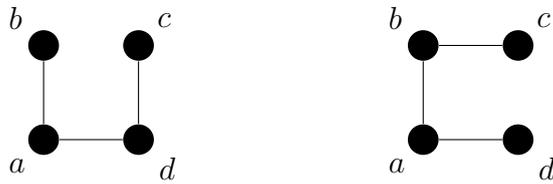
Figura 4.1: Enumeración de nodos.

Además se usará la notación \bar{x} para representar el complemento del bit x .

- $|E(T_1) \cap E(T_2)| = 2$.

Como los únicos árboles posibles que hay con cuatro nodos son caminos y estrellas, en este caso no pueden ambos ser del último tipo.

- Camino-Camino.
Suponga que se tiene



Sigue que:



Ahora bien, como \mathcal{P} separa grafos conexos de no conexos, se debe tener que $c' = \bar{c}$. Finalmente,



Así, existen grafos conexos cuyas respuestas son: $abcd$ y $ab\bar{c}d$. Luego, sin importar el valor de c'' , el mensaje del grafo de la derecha, no-conexo, coincide con el de uno que sí lo es.

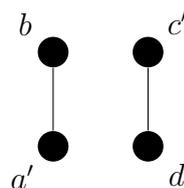
b) Camino-Estrella.



Notando que el nodo aislado del grafo de la izquierda de la siguiente figura debe escribir \bar{d} , se tiene el mensaje del grafo de la derecha, como se muestra a continuación:

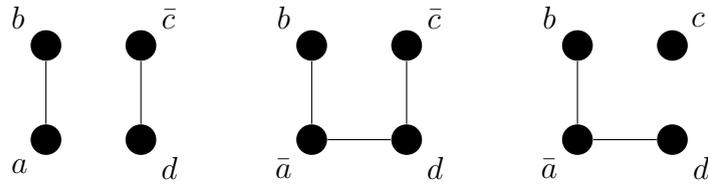


Se estudia ahora el siguiente grafo,



y los dos posibles valores que puede tomar a' . Es claro que si $a' = a \Rightarrow c' = \bar{c}$. Así pues, se tienen los siguientes casos:

1) $a' = a$.



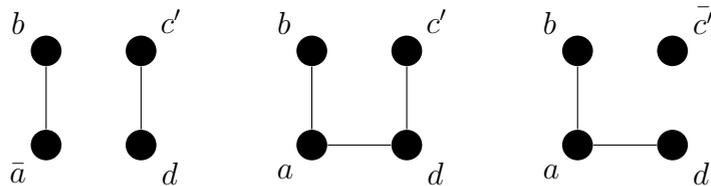
Resulta directo, por el mensaje que escribe el primer grafo, que el mensaje del nodo 1 en el grafo del medio sea \bar{a} . De esta manera, el nodo aislado del último grafo debe enviar c . Finalmente se obtiene,



Es decir, un grafo conexo y uno no conexo tienen el mismo mensaje. Por otro lado, si estudiamos la otra posibilidad:

2) $a' = \bar{a}$.

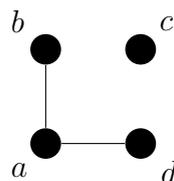
La situación es la siguiente:



Si $c' = c$, se tiene



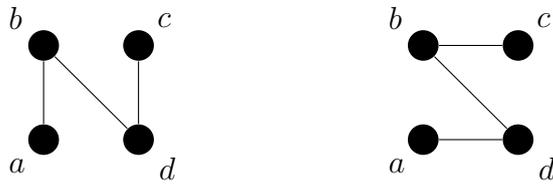
reduciéndose así al caso 1.a) ya tratado. Por otro lado, si $c' = \bar{c}$ entonces,



llegando a una contradicción.

2. $|E(T_1) \cap E(T_2)| = 1$.-

a) Camino-Camino.-



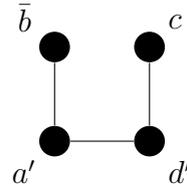
Considerando:



Notar que si $b'd' = bd$, este caso se reduce a uno de tipo 1.a).

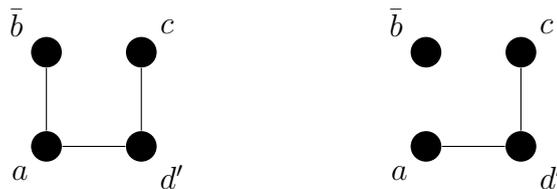
1) $b' = b$.-

En este caso, por lo recién mencionado, debe tenerse $d' = \bar{d}$, y por lo tanto, $b_1 = \bar{b}$ ya que hay grafos conexos cuyos mensajes corresponden a $abcd$ y $abcd\bar{}$. Enfocándose ahora en:



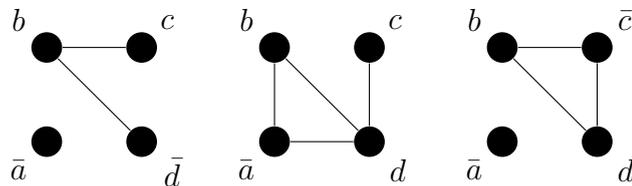
hay dos subcasos que estudiar.

1.1) $a' = a$.-

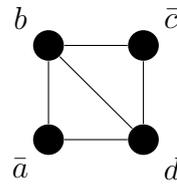


Es sabido que hay grafos conexos que escriben $abcd$ y $abcd\bar{}$. Esto explica el mensaje del nodo 2 en el grafo de la derecha, lo que concluye este escenario.

1.2) $a' = \bar{a}$.-



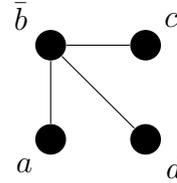
Debido a que en este caso $b'd' = b\bar{d}$, el nodo 1 en el primer grafo de la figura debe enviar \bar{a} , lo que justifica los mensajes de los dos grafos restantes. Finalmente,



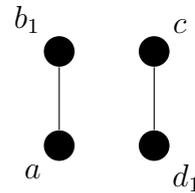
mismo mensaje el último grafo de la figura previa.

2) $b' = \bar{b}$.

2.1) $d' = d$



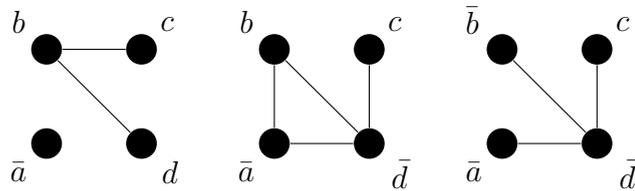
Como antes, se considera:



Si $b_1 = b \Rightarrow d_1 = \bar{d}$.



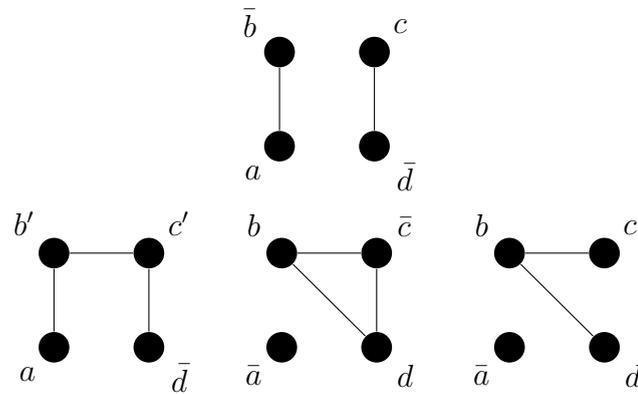
No puede ser que $a' = a$ en el segundo grafo, pues de ser así, si $d' = d$ se tiene un caso tipo 1.a). Por otro lado, si $d = \bar{d}$ los dos grafos previos generan una contradicción. Luego, $a' = \bar{a}$.



Obteniendo de esta manera:



Resta por analizar el caso: $b_1 = \bar{d} \Rightarrow d_1 = \bar{d}$.

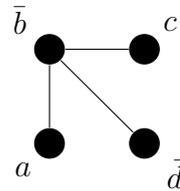


Un argumento análogo al anterior permite concluir $c' = \bar{c} \Rightarrow b' = b$, lo que permite conocer el resto de los mensajes de manera directa. Para concluir se considera el siguiente par de grafos:

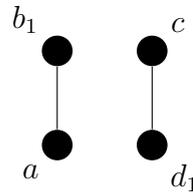


Los dos grafos no-conexos de la figura anterior permiten concluir que $a' = a$. Pero en ese caso se debe tener $c' = c$, llegando así, nuevamente, una situación como es en el caso 1.a).

2.2) $b' = \bar{b}$, $d' = \bar{d}$.

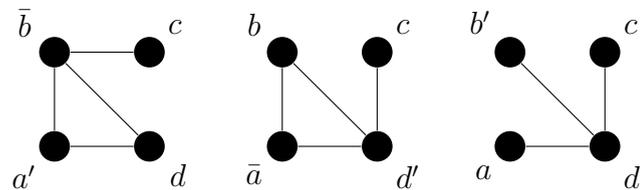


Enfocándose nuevamente en:



sigue que el mismo análisis desarrollado en el caso $b_1 = b$, $d_1 = \bar{d}$ aplica también en esta situación, y por tanto se supone $b_1 = \bar{b}$, $d_1 = d$.

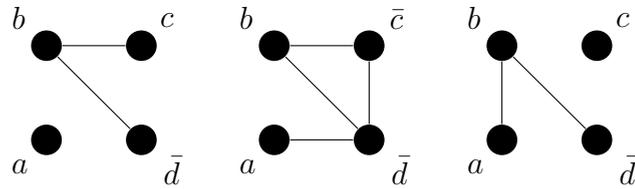
Ahora,



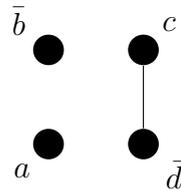
debido al presente caso, $a' = \bar{a}$. Del tercer grafo se deduce $d' = \bar{d}$.

El mensaje que debe enviar el grafo del medio en la figura de arriba, $ab\bar{c}\bar{d}$, justifica el escrito por el grafo de la izquierda a continuación. Los

otros dos siguen directamente.



Con esto,



concluyendo así, que este tampoco es un caso factible.

b) Camino-Estrella.-



Primero se estudia el par:



Ahora, hay tres posibles combinaciones para a' y d' (se descarta $a' = a \wedge d' = d$, pues se reduce a un caso ya visto).

1) $a' = a, d' = \bar{d}$.-

1.1) $c' = c$.



lo que corresponde a un caso 1.a).

1.2) $c' = \bar{c}$.

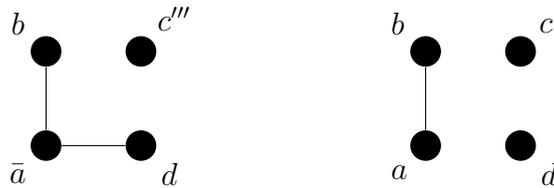


Luego, este caso tampoco es posible.

2) $a' = \bar{a}, d' = d$



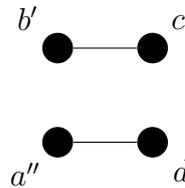
Es claro que $a'' = \bar{a}$. Ahora, como los dos grafos de la figura anterior no pueden escribir el mismo mensaje, $c'' = \bar{c}$. Finalmente:



Pero como existe un grafo conexo cuyo mensaje es $\bar{a}b\bar{c}d \Rightarrow c''' = c$. Se explica así, el mensaje del último grafo, el que genera conflicto en este caso.

3) $a' = \bar{a}, d' = \bar{d}$.

3.1) $c' = c$



$b' = b$ ya que existen grafos conexos cuyos mensajes corresponden a $abcd$ y $\bar{a}bcd$. Si se considera:



y razona de igual manera se concluye $b'' = \bar{b}$. Así pues, los dos grafos previos, uno conexo y uno no, escriben el mismo mensaje.

3.2) $c' = \bar{c}$.



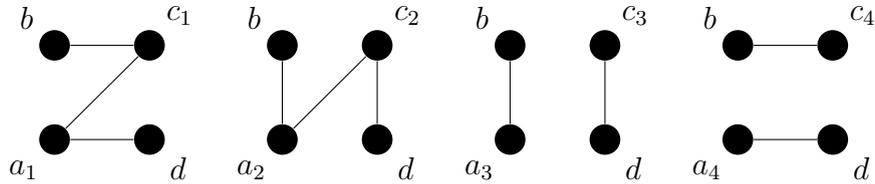
En este caso se tiene un grafo conexo que envía $\bar{a}b\bar{c}d \Rightarrow c'' = c \Rightarrow d'' = \bar{d}$. Esto conlleva a una contradicción plasmada en la siguiente pareja:



c) Estrella-Estrella.-
Se tiene:



Para analizar este caso, más complejo que los anteriores, se consideran los 4 siguientes grafos:



Si $a_i c_i = ac$ para cualquier $i \in \{1, 2, 3, 4\}$, se obtiene una contradicción. En efecto, para $i = 1, 2$ se reduce a un caso tipo 1.b). Por otro lado, para $i = 3, 4$ se tendrían grafos conexos y no conexos tales que escriben el mismo mensaje. De esta manera, $a_i c_i \neq ac \forall i$.

Más aún, si $a_1 c_1 = a_2 c_2$, entonces se cae en el caso 2a). Por último, un protocolo válido debe satisfacer: $a_i c_i \neq a_j c_j, i = 1, 2, j = 3, 4$.

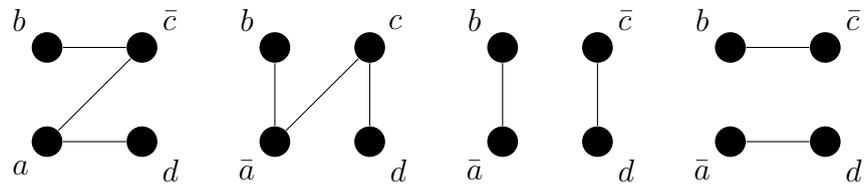
Así pues, que $ac \neq a_i c_i \forall i, a_1 c_1 \neq a_2 c_2, a_i c_i \neq a_j c_j, i = 1, 2, j = 3, 4$. Ahora, como sólo hay tres combinaciones disponibles, y cuatro parejas, dos deben ser iguales. Estas deben ser la 3 y 4, es decir, $a_3 c_3 = a_4 c_4$. Notar que fijando $a_1 c_1$ y $a_2 c_2$, las restantes quedan determinadas.

Incluso más, por simetría basta estudiar los siguientes casos:

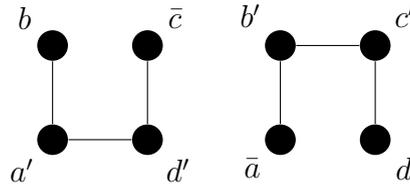
- $a_1 c_1 = a\bar{c} \quad \wedge \quad a_2 c_2 = \bar{a}c$
- $a_1 c_1 = a\bar{c} \quad \wedge \quad a_2 c_2 = \bar{a}\bar{c}$
- $a_1 c_1 = \bar{a}c \quad \wedge \quad a_2 c_2 = \bar{a}\bar{c}$

Pero el análisis de los dos últimos casos es análogo, por lo que en realidad sólo se deben estudiar los dos primeros escenarios.

$$1) \quad a_1 c_1 = a\bar{c} \quad \wedge \quad a_2 c_2 = \bar{a}c \quad \Rightarrow \quad a_3 c_3 = a_4 c_4 = \bar{a}\bar{c}$$

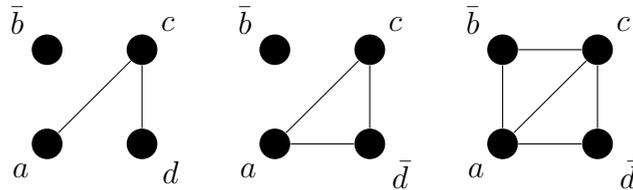


Ahora, mirando

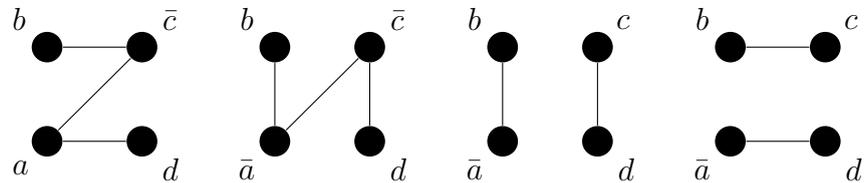


se puede notar que si $d' = d$ (en el primer grafo) se llega a una situación 2.a), o a un grafo conexo y desconexo escribiendo el mismo mensaje. Luego, $d' = \bar{d}$.

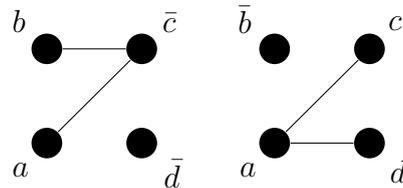
De manera similar, $b' = \bar{b}$. Para terminar:



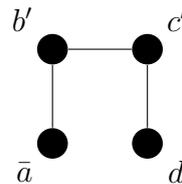
$$2) a_1c_1 = a\bar{c} \wedge a_2c_2 = \bar{a}\bar{c} \Rightarrow a_3c_3 = a_4c_4 = \bar{a}\bar{c}$$



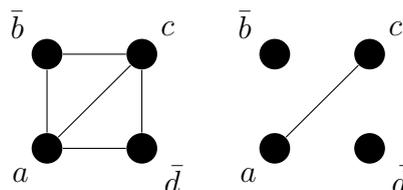
Se estudian:



Como antes, si se incluye

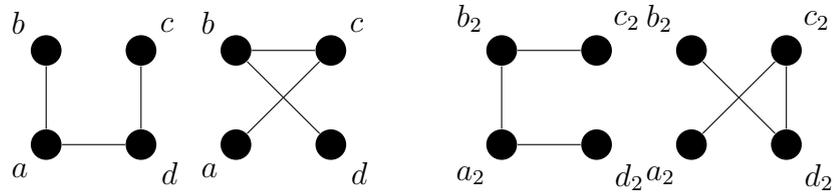


se concluye $b' = \bar{b}$, con lo que:



3. $|E(T_1) \cap E(T_2)| = 0$.-

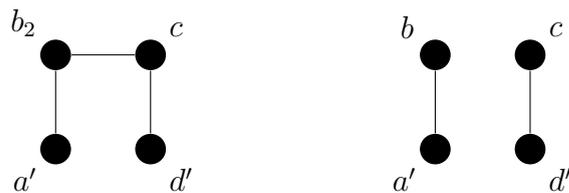
Como fue mencionado en un comienzo, al menos tres árboles escriben el mismo mensaje, o bien dos pares de ellos lo hacen. En el primer caso, necesariamente dos de ellos tienen aristas en común, situación ya estudiada. Si en la segunda alternativa, alguno de los pares de árboles comparte aristas también corresponde a lo ya analizado. Por tanto, se puede suponer, en este último caso, que se tiene dos pares de árboles, cada uno compuesto por árboles aristas-disjuntos, que escriben un mismo mensaje. Esto se grafica a continuación.



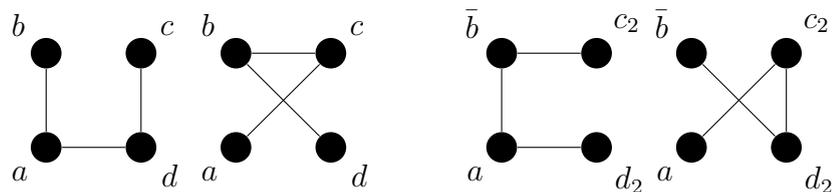
Primero, notar que $a_2 = a$, ya que en el primer grafo de cada pareja, el nodo 1 posee la misma vecindad. Ahora, de



se extrae que $c' \neq c''$. Si $c' = c \Rightarrow b_2 = \bar{b}$. Por otro lado, si $c' = \bar{c}, c'' = c$ y luego

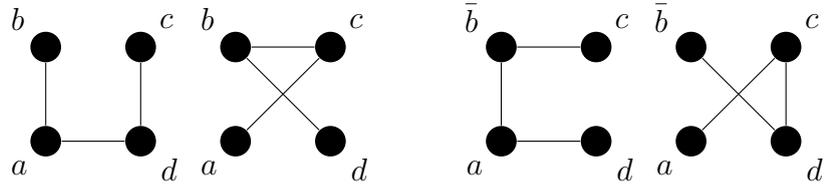


lo que implica que $b_2 = \bar{b}$. Se concluye pues, que en cualquier caso se tiene $b_2 = \bar{b}$. Así, la situación queda descrita por:

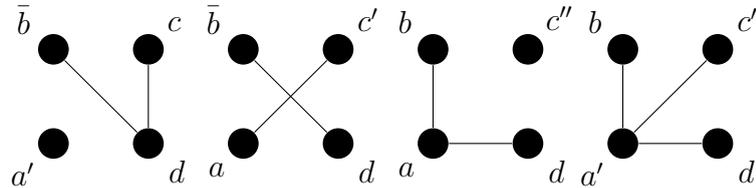


Por lo que se deben estudiar cuatro escenarios.

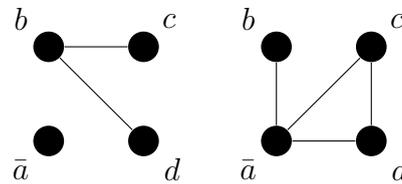
a) $c_2 = c \wedge d_2 = d$:



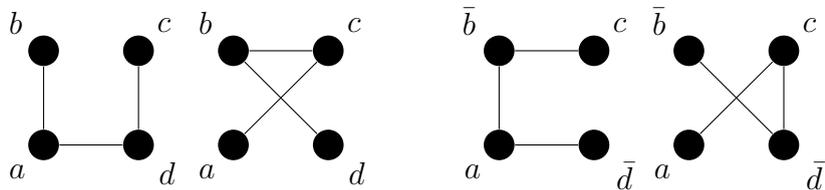
De los siguientes grafos:



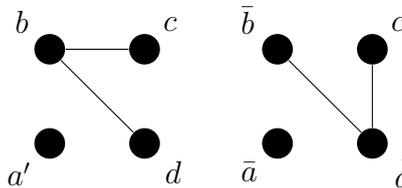
se concluye $a' = \bar{a}$ (del primero). El segundo, por su parte, indica $c' = \bar{c}$, mientras que el tercero $c'' = \bar{c}$. Toda esta información es utilizado en el cuarto, con lo que $a' = \bar{a}$. Sin embargo, de ser así:



b) $c_2 = c \wedge d_2 = \bar{d}$.

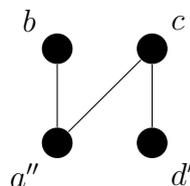


De los grafos desconexos que se ilustran en la siguiente figura

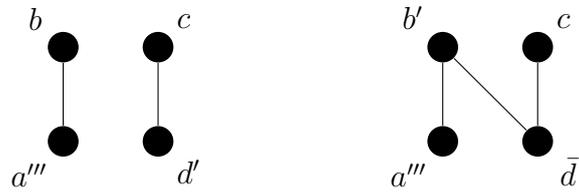


se obtiene que el nodo 1 escribe $a' = \bar{a}$. De esta forma, el mensaje del segundo grafo es $\bar{a}bcd$.

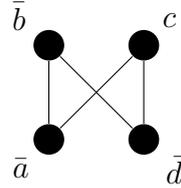
Para continuar, se analizan el siguiente grafo:



1) $a'' = \bar{a}$.



Como $a'' = \bar{a} \Rightarrow a''' = a \Rightarrow d' = \bar{d} \Rightarrow b' = \bar{b}$. Entonces,



lo que no puede ser, ya que un grafo no conexo escribe el mismo mensaje.

2) $a'' = a$.

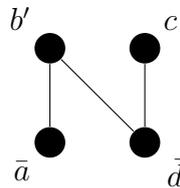
Notar que $d' = d$, corresponde a un caso tipo 1.a). Luego, $d' = \bar{d}$.



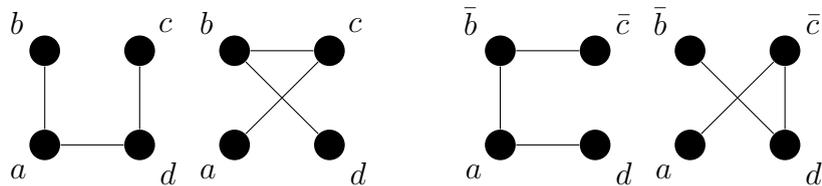
Ahora, los siguientes grafos no-conexos escriben $\bar{a}bc\bar{d}$ y $\bar{a}\bar{b}c\bar{d}$ respectivamente.



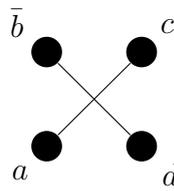
Luego, sin importar el valor que tome b' en la siguiente figura, se llega a una contradicción.



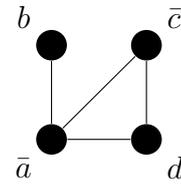
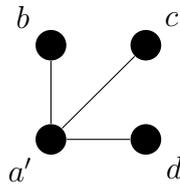
c) $c_2 = \bar{c} \wedge d_2 = d$.



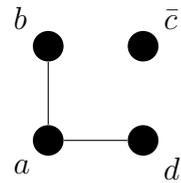
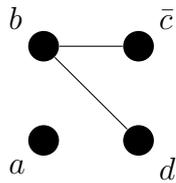
En este caso se trabaja con



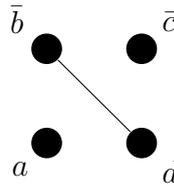
Siguiendo con



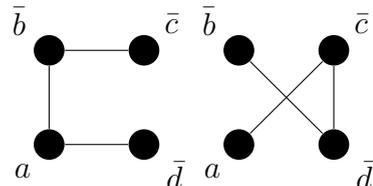
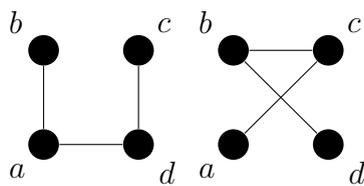
Si $a' = a$, se reduce a un caso 1.b) $\Rightarrow a' = \bar{a}$.



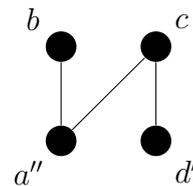
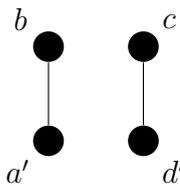
Pero entonces



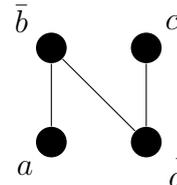
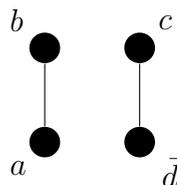
d) $c_2 = \bar{c} \wedge d_2 = \bar{d}$.



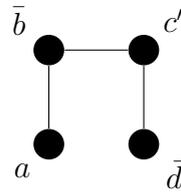
Se consideran



1) $a' = a$.

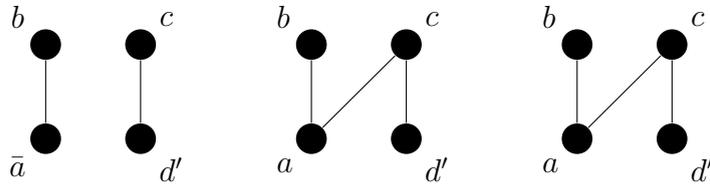


A partir de



se puede concluir que cualquiera sea el valor de c' , se tiene un caso tipo 1.

2) $a' = \bar{a}$.



Nuevamente, si $d' = d$ se tiene una situación tipo 1, por lo que $d' = \bar{d}$. La situación es análoga en el caso $b' = b$. Finalmente,



El argumento es el mismo en el caso $c' = c$ en el grafo de la izquierda. Mirando el de la derecha, a'' no puede ser a ya que un grafo desconexo escribe tal mensaje, pero tampoco puede ser \bar{a} , pues de ser así, se tendría un caso ya estudiado.

□

Así, se ha probado que es imposible decidir CONEXIDAD en grafos de 4 nodos cuando éstos pueden escribir un mensaje de tan solo un bit.

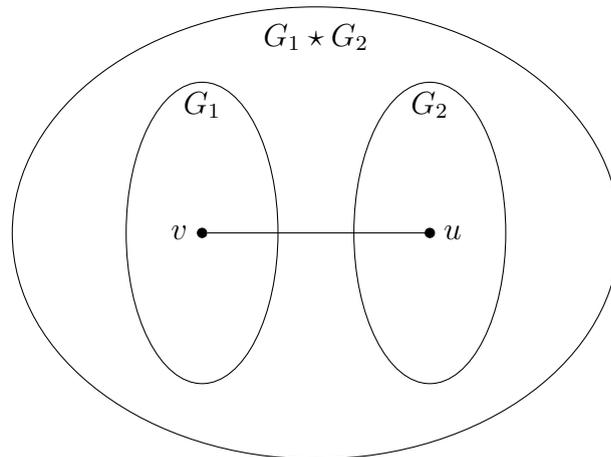
4.2. Grafos de $n = 5, 6, \dots, 26$ nodos

Para poder extender el resultado previo a todo $n \geq 4$, es decir, que un sólo bit por nodo no es suficiente para decidir CONEXIDAD, se hace uso del siguiente lema.

Lema 4.2. *Si existe un protocolo \mathcal{P} tal que decide el problema de CONEXIDAD en el modelo computacional number-in-hand, cuando los nodos pueden escribir tan solo un bit de información, para grafos con $n > 4$ nodos, entonces existe \mathcal{P}' que decide el mismo problema en grafos de 4 nodos.*

La demostración hace uso de la siguiente operación.

Definición 4.3. Se define la operación $G_1 \star G_2$ como aquella que dados dos grafos G_1 y G_2 , construye un nuevo grafo G como sigue: genera una copia de G_1 y una de G_2 , y se conectan mediante dos nodos cualesquiera $v \in G_1$, $u \in G_2$. La siguiente figura ilustra la operación.

Figura 4.2: Operación \star .

Demostración. -

Sea \mathcal{P} un protocolo como el de la hipótesis y G un grafo cualquiera de 4 nodos. Se define el protocolo \mathcal{P}' como aquel que a partir de G construye $G' = G \star P_{n-4}$ y retorna $\mathcal{P}(G')$. Es claro que $|V(G')| = n$. Además, G' es conexo *ssi* G lo es. De esta manera, resulta evidente que \mathcal{P}' decide la conexidad del grafo G . □

Nota.- El Lema 4.2 sigue siendo válido, y la demostración es la misma, si es que en vez de 1 bit, cada nodo puede escribir c bits, donde $c \geq 1$ es una constante.

De esta manera, ha sido completamente abarcado el primero de los dos casos que la fórmula encontrada en el Capítulo 3 no considera, grafos de tamaño pequeño. Se demostró que tal resultado no aporta información para grafos de pocos nodos ($n < 27$), ya que 1 bit no es suficiente para decidir conexidad en el modelo *number-in-hand*.

Capítulo 5

Grafos de grado máximo acotado

La cota inferior para la complejidad de CONEXIDAD en el modelo *number-in-hand*, expuesta en el Capítulo 3, posee dos deficiencias. Una de ellas fue cubierta en el Capítulo 4; el caso en que el grafo G tiene una cantidad de nodos $n < 27$. En este capítulo se analiza la segunda. Esto corresponde a estudiar la familia de grafos de grado máximo acotado.

Una adaptación del Lema 4.2 permite reducir el estudio a grafos G , de grado máximo 2. Haciendo uso de una herramienta muy útil para acotar inferiormente la complejidad de problemas en computación distribuida, denominada complejidad comunicacional, se prueba que en tal caso la complejidad de CONEXIDAD en el modelo *number-in-hand* es $\Omega(\log n)$.

Por otro lado, es claro que en el caso en que G es un grafo de grado máximo k fijo, entonces la complejidad del problema de CONEXIDAD es a lo más $k \log n$. Esto pues, si cada nodo envía toda su vecindad, es posible reconstruir el grafo completo.

5.1. Complejidad comunicacional

La complejidad comunicacional [2, 15] es una herramienta muy útil y bastante empleada en el marco de la computación distribuida, debido a que permite brindar cotas inferiores para la complejidad de los problemas estudiados.

En ella se dispone de dos jugadores, Alice y Bob, ambos con poder computacional ilimitado. Cada uno posee su propia entrada, correspondiente a un vector binario n -dimensional, x e y respectivamente, y ninguno conoce la entrada del otro jugador. Ellos buscan poder computar el valor de $f(x, y)$, donde la función $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ es conocida por ambos. Para ello, intercambian bits de información mediante un proceso de comunicación secuencial, donde en cada paso, un bit es comunicado de un jugador al otro. A este proceso se le denomina protocolo de comunicación. El *costo* de un protocolo corresponde a la mínima cantidad de bits necesarios intercambiar en el peor caso.

De manera formal se define como sigue.

Definición 5.1. Un *protocolo de comunicación* de r -rondas para una función f es una secuencia de pares $(S_1, C_1), (S_2, C_2), \dots, (S_r, C_r), (f_1, f_2)$. La variable S_i consta de una entrada, el patrón de comunicación de las primeras $i - 1$ rondas, y una salida en $\{1, 2\}$ que indica qué jugador envía información en la i -ésima ronda. Por su parte C_i también consta de entrada y salida. La primera es la historia de comunicación más el *input* del jugador

activo, mientras que la última es el bit que éste comunica a su par en la i -ésima ronda. Finalmente f_1 y f_2 , las salidas de cada jugador, deben ser igual a $f(x, y)$.

Así, la complejidad comunicacional de una función f queda definida por

$$C(f) = \min_{P \in \mathcal{P}_f} \max_{x, y} \{\text{Número de bits intercambiados en } P \text{ sobre } x, y\}$$

donde \mathcal{P}_f es la familia de protocolos que resuelven f .

Una observación directa es que $C(f) \leq n + 1$, donde n es el mínimo de los largos de la codificación de las entradas x e y , pues un protocolo trivial corresponde a que un jugador envíe su entrada completa al segundo, de manera que éste pueda computar $f(x, y)$ y responder la salida al primero. Sin embargo, la pregunta de interés es: ¿existe un mejor protocolo? Esto es, un protocolo que permita a Alice y Bob conocer $f(x, y)$ intercambiando menos de $n + 1$ bits.

Responder a esta pregunta es de gran utilidad cuando se estudian problemas en modelos de computación distribuida, ya que las cotas obtenidas mediante este procedimiento, inducen una cota para la complejidad de los problemas en el modelo *number-in-hand*, mediante una simple reducción. Un ejemplo de esto se presenta a continuación.

Ejemplo 5.2. Se define el problema EQ como sigue:

$$EQ(x, y) = \begin{cases} 1 & \text{si } x = y; \\ 0 & \text{si no.} \end{cases}$$

Suponga que se dispone de una red como en el modelo *number-in-hand* y que se desea saber si los nodos v_1 y v_n poseen la misma vecindad. Luego, una cota inferior para la complejidad de este problema en el modelo *number-in-hand* es $C(EQ)/n$. En efecto, sean x, y vectores indicatrices correspondientes a la vecindades de v_1 y v_n respectivamente, y considere a x e y como las entradas de Alice y Bob. Ahora bien, la complejidad comunicacional dice que se necesitan $C(EQ)$ bits para poder computar $EQ(x, y)$, por lo que cualquier protocolo en el modelo *number-in-hand* capaz de resolver el problema en cuestión requiere que el *referee* disponga de al menos $C(EQ)$ bits. Finalmente, para que el *referee* reciba tal cantidad de información debe existir 1 nodo que envíe un mensaje de al menos $C(EQ)/n$ bits.

Adicionalmente, se define la matriz de computo de una función f , denotada por $M(f)$, com sigue.

Definición 5.3. La matriz de computo de una función f , $M(f)$, es una matriz de $2^n \times 2^n$, tal que el elemento (x, y) corresponde a $f(x, y)$.

Nota.- Se escribe un elemento de la matriz mediante el par (x, y) para dar a entender que las filas se indexan por las entradas de Alice, x , y las columnas por las entradas de Bob, y .

5.2. Grado máximo $d = 2$

De manera análoga a lo realizado en el Capítulo 4 para analizar el caso de grafos pequeños, se estudia primero un caso simple, y luego se generaliza. Así, en este caso se abarca primero el caso en que el grado máximo es $d = 2$. Para poder acotar la complejidad del problema de CONEXIDAD en el modelo *number-in-hand*, para esta familia de grafos, se hace uso de la complejidad comunicacional.

5.2.1. Caso general

Sean Alice y Bob dos sujetos tales que cada uno conoce un grafo completo K_n (n par), cuyos conjuntos de nodos son $\{u_1, \dots, u_n\}$ y $\{v_1, \dots, v_n\}$ respectivamente. Además, ambos disponen de un emparejamiento perfecto en su grafo, M^A y M^B respectivamente. En base a estos emparejamientos se define el siguiente grafo:

Definición 5.4. Sean K_n^1 y K_n^2 dos grafos completos, con n par. Dados dos emparejamientos perfectos M^A, M^B , uno de cada grafo, se define el grafo $G_{M^A, M^B} = (V, E)$, donde $V = \{u_1, \dots, u_n, v_1, \dots, v_n\}$, y el conjunto de aristas es $E = M^A \cup M^B \cup \{u_i v_i, \forall i \in [n]\}$.

Un ejemplo se muestra en la siguiente figura.

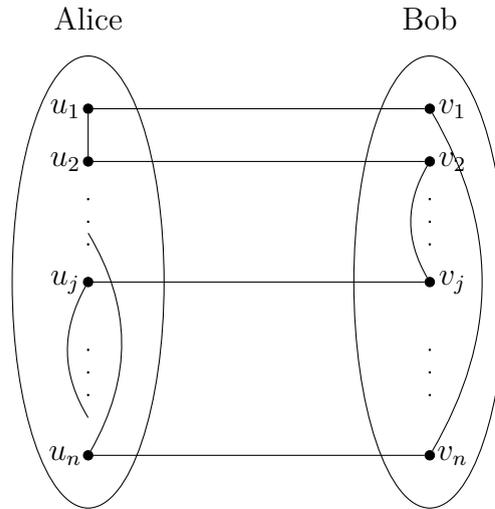


Figura 5.1: Grafo G_{M^A, M^B} .

Luego, lo que Alice y Bob desean saber es si el grafo G_{M^A, M^B} es conexo o no. A este problema se refiere por CON-AB. Para poder decidir este problema, es necesario que uno de ellos conozca ambos emparejamientos para tomar la decisión. Esto se demuestra en el siguiente lema.

Lema 5.5. *La complejidad comunicacional del problema CON-AB es $\Theta(|\langle M \rangle|)$.*

Nota.- En el enunciado, M corresponde a un emparejamiento perfecto dentro de un grafo completo K_n (n par), y $\langle M \rangle$ a su codificación en binario.

Para ver que la complejidad comunicacional es a lo más $|\langle M \rangle|$, basta notar que si Alice envía a Bob su emparejamiento completo, salvo una arista (como se trata de un emparejamiento perfecto, entonces especificar $n/2 - 1$ elementos es suficiente para que éste resulte únicamente definido), Bob es capaz de decidir el problema. Posteriormente él envía la respuesta a Alice y el protocolo finaliza.

Lo interesante, es mostrar la cota inferior. Para ello, se hace uso de dos lemas. El primero, un resultado clásico de complejidad comunicacional. El segundo por otro lado, prueba que para todo par de emparejamientos de Alice, M_1^A y M_2^A , existe un emparejamiento de Bob, M^B , tal que $G_{M_1^A, M^B}$ es no conexo y $G_{M_2^A, M^B}$ sí lo es. Así, para poder saber si G_{M^A, M^B} es conexo, es necesario que uno de ellos envíe una función inyectiva de los emparejamientos al otro.

Lema 5.6. [2] Sea $f : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$ tal que todas las filas de $M(f)$ son distintas. Entonces, $C(f) \geq \log n$.

Lema 5.7. Para todo par de emparejamientos de Alice M_1^A y M_2^A , existe un emparejamiento de Bob, M^B , de manera que $G_{M_1^A, M^B}$ es no conexo y $G_{M_2^A, M^B}$ sí lo es.

Demostración. -

Sean M_1^A, M_2^A dos emparejamientos distintos de Alice. Sea, además, $e^* = u_i u_j \in M_1^A \cap \overline{M_2^A}$, es decir, una arista que se encuentre en sólo uno ellos (existe pues $M_1^A \neq M_2^A$). Lo primero que se debe notar es que si $v_i v_j \in M^B$, un emparejamiento de Bob, entonces $G_{M_1^A, M^B}$ será no conexo, cualesquiera sean el resto de las aristas de M^B . Así, basta con demostrar la existencia de un emparejamiento de Bob que conteniendo la arista $v_i v_j$ haga conexo el grafo $G := G_{M_2^A, M^B}$ para concluir.

En lo que sigue, un tal M^B es construido. Para esto, se reordenan los nodos de Alice de acuerdo a la permutación π . Esta es la permutación que deja en las primeras posiciones a los nodos u_i con su vecino en M_2^A ($u_{i'}$), seguidos por u_j con su vecino M_2^A ($u_{j'}$). Luego, continúan el resto de los nodos emparejados de acuerdo a las aristas de M_2^A . Un ejemplo de esto se ilustra en la Figura 5.2.

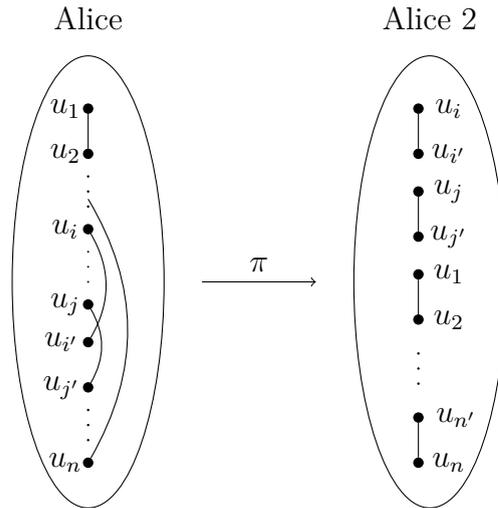


Figura 5.2: Permutación π .

Con este nuevo orden $\pi(u)$, resulta fácil encontrar un emparejamiento de Bob que haga conexo al grafo. En efecto, si dos nodos consecutivos del nuevo orden, llámense $u_{\pi(l)}, u_{\pi(l)+1}$, no están conectados mediante una arista del emparejamiento de Alice, es decir, $u_{\pi(l)} u_{\pi(l)+1} \notin M_2^A$, entonces se agregan sus correspondientes en el grafo de Bob al emparejamiento, esto es, $v_{\pi(l)} v_{\pi(l)+1} \in M^B$. Esto se cumple para todo $4 \leq l \leq n - 1$. Adicionalmente, se deben agregar las aristas $v_i v_j$ y $v_n v_{i'}$ al emparejamiento, donde i' es el índice del vecino de u_i en M_2^A . La Figura 5.3 ejemplifica una situación final.

De esta forma, se ha encontrado un emparejamiento de Bob, M^B , tal que $G_{M_1^A, M^B}$ es no conexo, y $G_{M_2^A, M^B}$ sí lo es. □

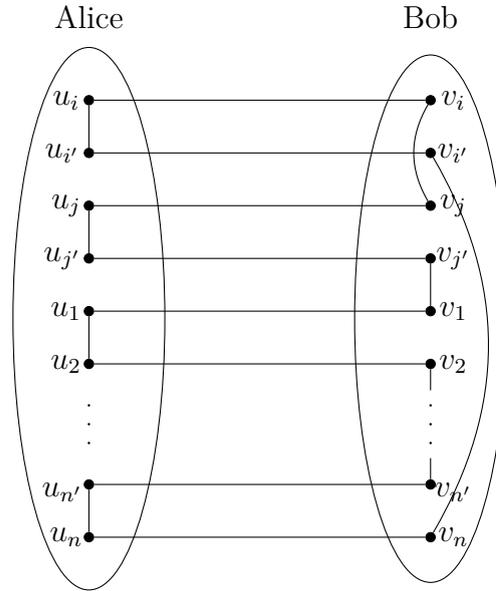


Figura 5.3: Configuración final.

Sea f la función característica del problema CON-AB. El Lema 5.7 permite concluir que todas las filas de la matriz de computo $M(f)$ de f son distintas. En efecto, para todo par de filas (emparejamientos de Alice M_1^A, M_2^A), existe una columna (emparejamiento de Bob M^B) que las distingue (hace conexo a $G_{M_1^A, M^B}$ y no conexo a $G_{M_2^A, M^B}$). Finalmente, complementando este resultado con el Lema 5.6, se concluye como corolario el Lema 5.5.

Para poder entender el alcance de éste, es necesario precisar cuánta es la información necesaria comunicar, es decir, $|\langle M \rangle|$. E. Hynds y M. Cole dan una respuesta a esto, ya que ellos estudian la cantidad de emparejamientos perfecto que posee K_n [14]. Tal cantidad, denotada por $\|M\|$, está dada por la expresión:

$$\|M\| = \prod_{i=1}^{n/2} (2i - 1) \quad (5.1)$$

es decir, es el producto de los impares menores a n (par). Además, $|\langle M \rangle| = \log \|M\|$. Luego, para poder decidir el problema de CONEXIDAD en el modelo *number-in-hand*, el *referee* debe disponer de $\Omega(n \log n)$ bits de información, pues $|\langle M \rangle| = \Omega(n \log n)$. Por lo tanto, existe un nodo cuyo largo de mensaje es $\Omega(\log n)$.

Teorema 5.8. *La complejidad del problema de CONEXIDAD en el modelo number-in-hand, restringido a grafos de grado máximo 2, es $\Omega(\log n)$.*

□

Ahora bien, se ha demostrado así, que para decidir el problema de CONEXIDAD, restringido a la familia de grafos de grado máximo 2, existe un nodo que debe escribir en la pizarra un mensaje de largo al menos $\beta := \left\lceil \log \left(\prod_{i=1}^{n/2} (2i - 1) \right) \right\rceil / n$ bits. Luego, es de interés conocer el valor de

$$\alpha = \beta / n \quad (5.2)$$

esto es, cuántas veces $\log n$ debe ser el tamaño del mensaje más largo escrito en la pizarra.

Para esto, en la Figura 5.4 se grafica el valor de α , en función de $n = 4k$. Si bien el resultado de esta sección permite concluir una cota inferior para el problema de CONEXIDAD para todo grafo de grado máximo 2, el procedimiento realizado es válido para grafos de tamaño $n = 4k$. Esto pues, cada jugador, Alice y Bob, dispone de un $K_{n'}$, con n' par. Así, el grafo completo, G_{M^A, M^B} , está compuesto por $2n' = 4k$ nodos.

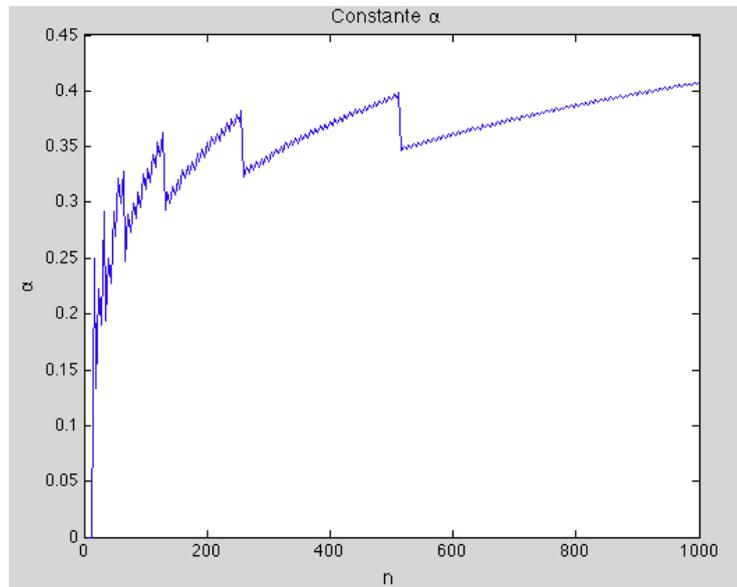


Figura 5.4: α v/s n .

Por otro lado, es directo que si cada nodo escribe un mensaje de largo $2 \log n$ bits (la codificación propia y de sus dos vecinos), se puede reconstruir el grafo, y por tanto, decidir si es conexo o no. De esta forma se debe tener que $\alpha \leq 2$, donde α es el definido en 5.2.

5.2.2. Caso particular

Para poder entender el alcance del Teorema 5.8, refiriéndose por esto al comportamiento de la cota en el caso $n \neq 4k$, se estudia la separación de dos familias particulares de grafos, ambas de grado máximo 2, pero con $n = 6$ nodos. Las familias son:

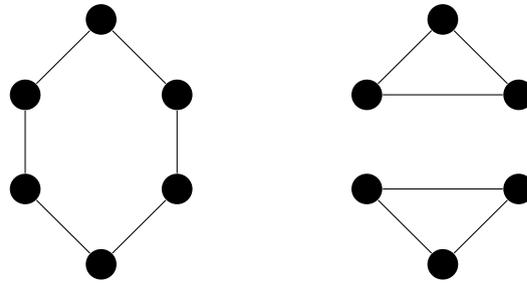
$$\mathcal{F}_1 = \{G \mid G \text{ está formado por 2 triángulos}\} \text{ y } \mathcal{F}_2 = \{G \mid G \text{ es un anillo de 6 nodos}\}$$

(ver Figura 5.5). Es claro que si el tamaño de los mensajes es $2 \lceil \log n \rceil = 6$, entonces es fácil distinguir ambas clases. Más aún, en tal caso es posible reconstruir el grafo G por completo.

En esta sección se presenta un protocolo \mathcal{P} , en el que cada nodo escribe un mensaje de 2 bits, que permite separar estas dos familias en una sola ronda [25].

Protocolo \mathcal{P} : Dos bits - una ronda

El protocolo se basa en el hecho que $G = (V, E)$ es un grafo simple (sin *buckles*), $|V| = 6$, y $d_G(v) = 2$, $\forall v \in V$.

Figura 5.5: C_6 v/s $C_3 - C_3$.

Primero se debe notar que cada nodo posee 2 vecinos, dentro de 5 posibles. Luego, quienes se conecten con el nodo v^* ; $ID(v^*) = 6$, pueden identificar al otro vecino que poseen con 2 bits, ya que sólo restan 4 posibilidades. Algo similar ocurre con v^* , quien puede identificar a su vecino con mayor ID (no puede ser el nodo v tal que $ID(v) = 1$). Así, estos tres nodos bastan para decidir el problema en este caso particular. Si los dos vecinos de v^* , son vecinos entre ellos, entonces se trata de un grafo compuesto por dos triángulos. En caso contrario es un ciclo de largo 6.

Un protocolo \mathcal{P} que realiza lo previamente descrito se presenta a continuación.

Algoritmo 1 Pseudo-código \mathcal{P}

Entrada: $\{ID(v)\}_{v \in V}$.

Salida: Decidir si G posee triángulo o no.

```

for  $v \in V$  do
  if  $ID(v) == 6$  then
    return  $\langle ID(v^*) \rangle$ , donde  $v^* = \max_{u \in N(v)} ID(u)$ 
  else if  $6 \notin \{ID(u) | u \in N(v)\}$  then
    return 00
  else
    return  $\langle ID(v^*) \rangle$ , donde  $v^* = \min_{u \in N(v)} ID(u)$ 
  end if
end for

```

Nota.- $\langle ID(v) \rangle$ denota la codificación binaria del identificador del nodo v .

Correctitud de \mathcal{P}

Sea G un grafo compuesto por dos triángulos disjuntos T_1 y T_2 . Sin pérdida de generalidad, $v^* \in T_1$. Luego, $m(u) = 00$, $\forall u \in T_2$. Sea $T_1 = u_1 u_2 v^*$. Así, u_1 envía la codificación del ID de u_2 y viceversa, *i.e.*, $m(u_1) = \langle ID(u_2) \rangle$, $m(u_2) = \langle ID(u_1) \rangle$. Si $\langle ID(u_i) \rangle \neq 00$, $i = 1, 2$, la correctitud del algoritmo sigue de manera directa.

Se supone pues, que $ID(u_1) > ID(u_2)$, y $\langle ID(u_2) \rangle = 00$. En tal caso, $m(u_2) = m(v^*) = \langle ID(u_1) \rangle$, permitiendo identificar de esta manera la vecindad completa de u_1 . Además,

$m(u_2) \neq 00 \Rightarrow u_2v^* \in E(G)$. Se concluye que $u_2v^* \in E(G)$, $u_1u_2 \in E(G)$ and $u_1v^* \in E(G)$, *i.e.*, G es la unión de dos triángulos.

5.3. Grado máximo k fijo

Para poder generalizar al caso de un grado máximo acotado fijo $k \geq 2$, se hará uso de una operación $G_1 \star G_2$ (ver definición 4.3).

Lema 5.9. *Suponga que existe un protocolo \mathcal{P} tal que decide el problema de CONEXIDAD en el modelo computacional number-in-hand, cuando los nodos pueden escribir $O(f(n))$ bits, para grafos de grafo máximo $k > 2$, y $f(n) = O(n^c \text{polylog } n)$. Entonces, existe un protocolo \mathcal{P}' , del mismo orden, que decide el mismo problema en grafos de grado máximo 2.*

Demostración. -

Sea \mathcal{P} un protocolo como el de la hipótesis, y G un grafo de n nodos y grado máximo 2. Sea además, $G' = G \star K_{k-1}$. Así, $|V(G')| = n + k - 1$. Ahora, para asegurarse que el grado máximo de G' sea k , se debe modificar levemente la operación \star ; el nodo escogido en G es cualquiera, pero dentro de los tienen grado 2. Es directo que G es conexo *ssi* G' lo es.

Luego, se define el protocolo \mathcal{P}' como aquel que dado un grafo G , construye G' , y da como respuesta, la salida que obtiene tras correr $\mathcal{P}(G')$. De esta manera, \mathcal{P}' decide el problema de CONEXIDAD en grafos cuyo grado máximo es 2. Este protocolo es de orden $O(f(n + k - 1)) = O(f(n))$. \square

Corolario 5.10. *La complejidad del problema de CONEXIDAD en el modelo number-in-hand es $\Omega(\log n)$ cuando se restringe a la clase de grafos de grado máximo acotado.*

\square

Este resultado muestra que, en realidad, la cota que se obtiene en el Capítulo 3 no es ajustada para esta clase de grafos. Se obtiene así, el orden preciso del problema de CONEXIDAD para la familia de grafos de grado máximo acotado: $\Theta(\log n)$.

Capítulo 6

Conclusiones y trabajo futuro

La presente memoria se ha enfocado en el estudio de la complejidad del problema CONEXIDAD, es decir, decidir si un grafo G es conexo o no, en el marco de la computación distribuida, en particular, en el modelo computacional *number-in-hand*.

El análisis del problema en el caso general, es decir, en el caso de un grafo G arbitrario, permite concluir que la complejidad de CONEXIDAD en el modelo *number-in-hand* es al menos $\log n - (\log \log n + 1 + \log n/n)$.

Esta cota, sin embargo, posee dos debilidades. En primer lugar, carece de sentido para grafos de $n < 27$ nodos, pues en tales casos señala que el largo de mensaje de los nodos es al menos ≤ 1 . En segundo lugar, para obtener la cota se emplea una reducción que genera un grafo de grado no acotado.

Restringiendo el estudio a estos ambos casos, se encuentran mejores cotas en ambas situaciones. Para grafos pequeños ($n < 27$) se demuestra que 1 bit no es suficiente para decidir CONEXIDAD en el modelo *number-in-hand*. Por otro lado, para la familia de grafos de grado acotado se muestra que la complejidad de CONEXIDAD es $\Theta(\log n)$.

Como trabajo futuro se plantea encontrar protocolos eficientes para resolver el problema CONEXIDAD en el caso general. Se sabe que si el grafo G es de grado máximo acotado k , entonces $k \log n$ bits por nodo son suficientes para decidir el problema. Sin embargo, en el caso general no se conoce una mejor cota superior que $n \log n$ (proveniente de enviar la vecindad completa). Por otro lado, es necesario verificar si la cota inferior expuesta es ajustada en el caso general. En caso que no lo sea, una tarea futura es mejorarla.

Adicionalmente, un aporte sería estudiar el problema de CONEXIDAD, en el modelo *number-in-hand*, incorporando un factor aleatorio al análisis. Si bien los resultados de Ahn *et al.* [1] pueden ser adaptados a éste modelo, el hecho de que su trabajo no se enfoque en el modelo *number-in-hand* permite pensar que puede existir una cota más ajustada.

Bibliografía

- [1] K. J. Ahn, S. Guha, and A. McGregor. Analyzing graph structure via linear measurements. In Y. Rabani, editor, *SODA*, pages 459–467. SIAM, 2012.
- [2] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*, chapter Communication Complexity. Cambridge University Press, New York, NY, USA, 1st edition, 2009.
- [3] F. Becker, A. Kosowski, N. Nisse, I. Rapaport, and K. Suchan. Allowing each node to communicate only once in a distributed system: shared whiteboard models. In G. E. Blelloch and M. Herlihy, editors, *SPAA*, pages 11–17. ACM, 2012.
- [4] F. Becker, M. Matamala, N. Nisse, I. Rapaport, K. Suchan, and I. Todinca. Adding a referee to an interconnection network: What can(not) be computed in one round. In *IPDPS*, pages 508–514. IEEE, 2011.
- [5] M. Becker, W. Degenhardt, J. Doenhardt, S. Hertel, G. Kaninke, W. Kerber, K. Mehlhorn, S. Näher, H. Rohnert, and T. Winter. A probabilistic algorithm for vertex connectivity of graphs. *Inf. Process. Lett.*, 15(3):135–136, 1982.
- [6] D. Berend and T. Tassa. Improved bounds on bell numbers and on moments of sums of random variables. *Probability and Mathematical Statistics*, 30(2):185–205, 2010.
- [7] S. Bouroubi. Bell numbers and engel’s conjecture, 2007.
- [8] G. Cormode and S. Muthukrishnan. Space efficient mining of multigraph streams. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS ’05, pages 271–282, New York, NY, USA, 2005. ACM.
- [9] A.-H. Esfahanian and S. L. Hakimi. On computing the connectivities of graphs and digraphs. *Networks*, 14(2):355–366, 1984.
- [10] S. Even and R. E. Tarjan. Network flow and testing graph connectivity. *SIAM J. Comput.*, 4(4):507–518, 1975.
- [11] J. Feigenbaum, S. Kannan, A. McGregor, and J. Zhang. On graph problems in a semi-streaming model. In *In 31st International Colloquium on Automata, Languages and Programming*, pages 531–543, 2004.
- [12] M. R. Henzinger and S. Rao. Faster vertex connectivity algorithms. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*.

-
- [13] M. R. Henzinger, S. Rao, and H. N. Gabow. Computing vertex connectivity: New bounds from old techniques. *J. Algorithms*, 34(2):222–250, 2000.
- [14] E. Hynds and M. B. Cole. Perfect matchings: Enumeration and properties of extendability. *Alabama Journal of Mathematics*, 29, 2005.
- [15] E. Kushilevitz and N. Nisan. *Communication complexity*. Cambridge University Press, New York, NY, USA, 1997.
- [16] J. W. Moon. *Various proofs of cayley’s formula for counting trees*, chapter 11, pages 70–78. Holt, Rinehart and Winston, New York, 1967.
- [17] R. Navigli and M. Lapata. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(4):678–692, Apr. 2010.
- [18] D. Peleg. *Distributed computing: a locality-sensitive approach*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.
- [19] J. M. Phillips, E. Verbin, and Q. Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’12, pages 486–501. SIAM, 2012.
- [20] A. Z. Salamon. Streaming bounds from difference ramification. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:183, 2012.
- [21] Z.W. Sun and D. Zagier. On a curious property of bell numbers. *Bulletin of the Australian Mathematical Society*, 84:153–158, 8 2011.
- [22] R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972.
- [23] D. R. White and F. Harary. The cohesiveness of blocks in social networks: Node connectivity and conditional density. *Sociological Methodology*, 31:305–359, 2001.
- [24] E. Wright. Equal sums of like powers. *Bull. Amer. Math. Soc.*, 54(8):755–757, 1948.
- [25] Conversación personal con Andrés Gomberoff.
- [26] Conversación personal con Ioan Todinca.