

Protocolos de demostración interactivos

IIC3810

Un protocolo de demostración interactivo para SAT

Tenemos un protocolo interactivo para demostrar que $\varphi \in \text{SAT}$

Un protocolo de demostración interactivo para SAT

Tenemos un protocolo interactivo para demostrar que $\varphi \in \text{SAT}$

- ▶ El protocolo tiene dos participantes: un verificador **V** y un demostrador **D**

Un protocolo de demostración interactivo para SAT

Tenemos un protocolo interactivo para demostrar que $\varphi \in \text{SAT}$

- ▶ El protocolo tiene dos participantes: un verificador **V** y un demostrador **D**
- ▶ **V** trata de demostrar que $\varphi \in \text{SAT}$ haciendo preguntas a **D**

Un protocolo de demostración interactivo para SAT

Tenemos un protocolo interactivo para demostrar que $\varphi \in \text{SAT}$

- ▶ El protocolo tiene dos participantes: un verificador **V** y un demostrador **D**
- ▶ **V** trata de demostrar que $\varphi \in \text{SAT}$ haciendo preguntas a **D**
- ▶ **D** tiene poder de computación ilimitado

Un protocolo de demostración interactivo para SAT

Tenemos un protocolo interactivo para demostrar que $\varphi \in \text{SAT}$

- ▶ El protocolo tiene dos participantes: un verificador **V** y un demostrador **D**
- ▶ **V** trata de demostrar que $\varphi \in \text{SAT}$ haciendo preguntas a **D**
- ▶ **D** tiene poder de computación ilimitado
 - ▶ Puede tratar de engañar a **V** dando información que indica que $\varphi \in \text{SAT}$ cuando φ es inconsistente

Un protocolo de demostración interactivo para SAT

Con entrada φ , el protocolo funciona de la siguiente forma:

Un protocolo de demostración interactivo para SAT

Con entrada φ , el protocolo funciona de la siguiente forma:

1. **V** pregunta a **D** por una valuación σ que satisfaga a φ

Un protocolo de demostración interactivo para SAT

Con entrada φ , el protocolo funciona de la siguiente forma:

1. **V** pregunta a **D** por una valuación σ que satisfaga a φ
2. **D** responde con una valuación σ que satisfaga la condición anterior

Un protocolo de demostración interactivo para SAT

Con entrada φ , el protocolo funciona de la siguiente forma:

1. **V** pregunta a **D** por una valuación σ que satisfaga a φ
2. **D** responde con una valuación σ que satisfaga la condición anterior
3. **V** chequea si $\sigma(\varphi) = 1$, y si es así acepta

Un protocolo de demostración interactivo para SAT

Con entrada φ , el protocolo funciona de la siguiente forma:

1. **V** pregunta a **D** por una valuación σ que satisfaga a φ
2. **D** responde con una valuación σ que satisfaga la condición anterior
3. **V** chequea si $\sigma(\varphi) = 1$, y si es así acepta

¿Puede engañar **D** a **V** en este protocolo?

Un protocolo de demostración interactivo para SAT

Con entrada φ , el protocolo funciona de la siguiente forma:

1. **V** pregunta a **D** por una valuación σ que satisfaga a φ
2. **D** responde con una valuación σ que satisfaga la condición anterior
3. **V** chequea si $\sigma(\varphi) = 1$, y si es así acepta

¿Puede engañar **D** a **V** en este protocolo?

- ▶ No por la verificación realizada en el paso 3

Una noción de protocolo más general

El protocolo mostrado en las transparencias anteriores puede ser extendido a cualquier lenguaje $L \in \text{NP}$

▶ ¿Cómo?

Una noción de protocolo más general

El protocolo mostrado en las transparencias anteriores puede ser extendido a cualquier lenguaje $L \in \text{NP}$

▶ ¿Cómo?

Es posible extender esta noción de protocolo en dos direcciones:

Una noción de protocolo más general

El protocolo mostrado en las transparencias anteriores puede ser extendido a cualquier lenguaje $L \in \text{NP}$

- ▶ ¿Cómo?

Es posible extender esta noción de protocolo en dos direcciones:

- ▶ Permitir varias rondas de pregunta y respuesta

Una noción de protocolo más general

El protocolo mostrado en las transparencias anteriores puede ser extendido a cualquier lenguaje $L \in \text{NP}$

- ▶ ¿Cómo?

Es posible extender esta noción de protocolo en dos direcciones:

- ▶ Permitir varias rondas de pregunta y respuesta
- ▶ Permitir que haya una probabilidad de error asociada a la respuesta final de **V**

Una noción de protocolo más general

El protocolo mostrado en las transparencias anteriores puede ser extendido a cualquier lenguaje $L \in \text{NP}$

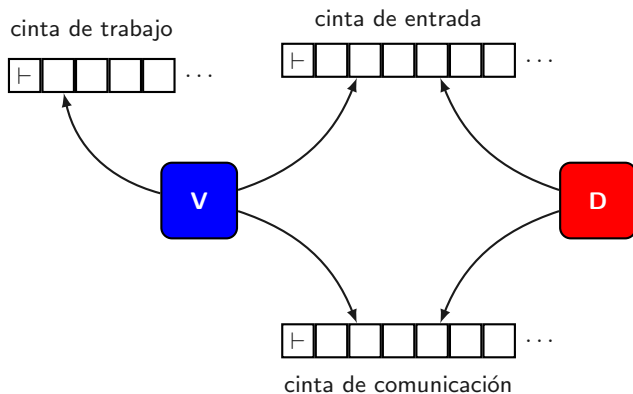
- ▶ ¿Cómo?

Es posible extender esta noción de protocolo en dos direcciones:

- ▶ Permitir varias rondas de pregunta y respuesta
- ▶ Permitir que haya una probabilidad de error asociada a la respuesta final de **V**

Vamos a ver las clases de complejidad que definen estas condiciones

Un protocolo determinista



Un protocolo determinista

V es una MT determinista que funciona en tiempo $f(|w|)$, donde w es la entrada

- ▶ En cada ronda **V** realiza a lo más $f(|w|)$ pasos

Un protocolo determinista

V es una MT determinista que funciona en tiempo $f(|w|)$, donde w es la entrada

- ▶ En cada ronda **V** realiza a lo más $f(|w|)$ pasos

D es determinista y tiene poder de computación ilimitado

Un protocolo determinista

V es una MT determinista que funciona en tiempo $f(|w|)$, donde w es la entrada

- ▶ En cada ronda **V** realiza a lo más $f(|w|)$ pasos

D es determinista y tiene poder de computación ilimitado

- ▶ **D** es simplemente una función

Un protocolo determinista

V es una MT determinista que funciona en tiempo $f(|w|)$, donde w es la entrada

- ▶ En cada ronda **V** realiza a lo más $f(|w|)$ pasos

D es determinista y tiene poder de computación ilimitado

- ▶ **D** es simplemente una función
- ▶ **D** puede incluso decidir un problema indecidible

Un protocolo determinista

V y **D** comparten dos cintas

Un protocolo determinista

V y **D** comparten dos cintas

- ▶ Una cinta de entrada donde se coloca el string w
 - ▶ Esta cinta es sólo de lectura

Un protocolo determinista

V y **D** comparten dos cintas

- ▶ Una cinta de entrada donde se coloca el string w
 - ▶ Esta cinta es sólo de lectura
- ▶ Una cinta de comunicación donde **V** puede colocar una consulta que es respondida por **D**

Un protocolo determinista

V y **D** comparten dos cintas

- ▶ Una cinta de entrada donde se coloca el string w
 - ▶ Esta cinta es sólo de lectura
- ▶ Una cinta de comunicación donde **V** puede colocar una consulta que es respondida por **D**
 - ▶ Colocar una pregunta o respuesta x en la cinta significa colocar $\vdash xBB \dots$ en ella

Un protocolo determinista

V y **D** comparten dos cintas

- ▶ Una cinta de entrada donde se coloca el string w
 - ▶ Esta cinta es sólo de lectura
- ▶ Una cinta de comunicación donde **V** puede colocar una consulta que es respondida por **D**
 - ▶ Colocar una pregunta o respuesta x en la cinta significa colocar $\vdash xBB \dots$ en ella
 - ▶ La respuesta de **D** a cada consulta de **V** debe tener tamaño acotado por $f(|w|)$

Un protocolo determinista

V además tiene una cinta a la cual **D** no tiene acceso

- ▶ Una cinta de trabajo que es de lectura y escritura

Un protocolo determinista

V además tiene una cinta a la cual **D** no tiene acceso

- ▶ Una cinta de trabajo que es de lectura y escritura

Inicialmente el protocolo entrega el control a **V**

Un protocolo determinista

V además tiene una cinta a la cual **D** no tiene acceso

- ▶ Una cinta de trabajo que es de lectura y escritura

Inicialmente el protocolo entrega el control a **V**

- ▶ Este control permanece en el poder de **V**, hasta que **V** realiza una consulta a **D** y le pasa el control

Un protocolo determinista

V además tiene una cinta a la cual **D** no tiene acceso

- ▶ Una cinta de trabajo que es de lectura y escritura

Inicialmente el protocolo entrega el control a **V**

- ▶ Este control permanece en el poder de **V**, hasta que **V** realiza una consulta a **D** y le pasa el control
- ▶ Una vez que la consulta ha sido respondida **D** le devuelve el control a **V**

Un protocolo determinista

V además tiene una cinta a la cual **D** no tiene acceso

- ▶ Una cinta de trabajo que es de lectura y escritura

Inicialmente el protocolo entrega el control a **V**

- ▶ Este control permanece en el poder de **V**, hasta que **V** realiza una consulta a **D** y le pasa el control
- ▶ Una vez que la consulta ha sido respondida **D** le devuelve el control a **V**
- ▶ **V** es quien decide si aceptar el string de entrada w

Un protocolo determinista

El número de rondas realizadas por el protocolo (\mathbf{V}, \mathbf{D}) con entrada w se define como el número de veces que el control cambia de dueño

Un protocolo determinista

El número de rondas realizadas por el protocolo (\mathbf{V}, \mathbf{D}) con entrada w se define como el número de veces que el control cambia de dueño

- ▶ Por ejemplo, decimos que tenemos 2 rondas si el control pasa de \mathbf{V} a \mathbf{D} por una consulta, y luego de \mathbf{D} a \mathbf{V} por la respuesta a la consulta

Un protocolo determinista

El número de rondas realizadas por el protocolo (\mathbf{V}, \mathbf{D}) con entrada w se define como el número de veces que el control cambia de dueño

- ▶ Por ejemplo, decimos que tenemos 2 rondas si el control pasa de \mathbf{V} a \mathbf{D} por una consulta, y luego de \mathbf{D} a \mathbf{V} por la respuesta a la consulta

\mathbf{V} debe tener el control al momento de decidir si acepta el string de entrada

Un protocolo determinista

El número de rondas realizadas por el protocolo (\mathbf{V}, \mathbf{D}) con entrada w se define como el número de veces que el control cambia de dueño

- ▶ Por ejemplo, decimos que tenemos 2 rondas si el control pasa de \mathbf{V} a \mathbf{D} por una consulta, y luego de \mathbf{D} a \mathbf{V} por la respuesta a la consulta

\mathbf{V} debe tener el control al momento de decidir si acepta el string de entrada

- ▶ Como esta operación termina la ejecución del protocolo, el número de rondas debe ser par

La clase $\text{dIP}[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

La clase $\text{dIP}[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

L está en $\text{dIP}[f(n)]$ si existe un verificador \mathbf{V} que funciona en tiempo polinomial tal que para cada $w \in \Sigma^*$:

La clase $\text{dIP}[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

L está en $\text{dIP}[f(n)]$ si existe un verificador \mathbf{V} que funciona en tiempo polinomial tal que para cada $w \in \Sigma^*$:

- ▶ Para cada demostrador \mathbf{D} , el protocolo (\mathbf{V}, \mathbf{D}) con entrada w realiza un número de rondas acotado por $f(|w|)$

La clase $\text{dIP}[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

L está en $\text{dIP}[f(n)]$ si existe un verificador \mathbf{V} que funciona en tiempo polinomial tal que para cada $w \in \Sigma^*$:

- ▶ Para cada demostrador \mathbf{D} , el protocolo (\mathbf{V}, \mathbf{D}) con entrada w realiza un número de rondas acotado por $f(|w|)$
- ▶ Si $w \in L$, entonces existe demostrador \mathbf{D} tal que (\mathbf{V}, \mathbf{D}) acepta w

La clase $\text{dIP}[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

L está en $\text{dIP}[f(n)]$ si existe un verificador \mathbf{V} que funciona en tiempo polinomial tal que para cada $w \in \Sigma^*$:

- ▶ Para cada demostrador \mathbf{D} , el protocolo (\mathbf{V}, \mathbf{D}) con entrada w realiza un número de rondas acotado por $f(|w|)$
- ▶ Si $w \in L$, entonces existe demostrador \mathbf{D} tal que (\mathbf{V}, \mathbf{D}) acepta w
- ▶ Si $w \notin L$, entonces para todo demostrador \mathbf{D}' se tiene que $(\mathbf{V}, \mathbf{D}')$ rechaza w

La clase $\text{dIP}[k]$

Ejercicio

1. Demuestre que $\text{SAT} \in \text{dIP}[2]$ y $\text{GRAPH-ISO} \in \text{dIP}[2]$
2. ¿Es cierto que $\overline{\text{SAT}} \in \text{dIP}[p(n)]$ o $\overline{\text{GRAPH-ISO}} \in \text{dIP}[p(n)]$, para algún polinomio $p(n)$?

La clase dIP

Sea

$$\text{dIP} = \bigcup_{k \in \mathbb{N}} \text{dIP}[n^k]$$

La clase dIP

Sea

$$\text{dIP} = \bigcup_{k \in \mathbb{N}} \text{dIP}[n^k]$$

Proposición

$$\text{dIP} = \text{NP}$$

La clase dIP

Sea

$$\text{dIP} = \bigcup_{k \in \mathbb{N}} \text{dIP}[n^k]$$

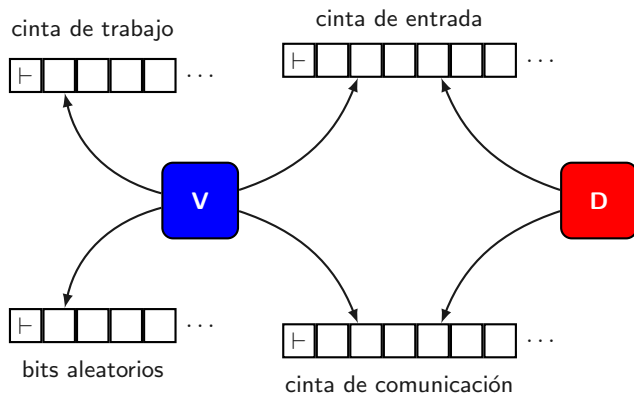
Proposición

$$\text{dIP} = \text{NP}$$

Ejercicio

Demuestre la proposición

Un protocolo aleatorizado



Un protocolo aleatorizado

V es una MT **probabilística** que funciona en tiempo $f(|w|)$, donde w es la entrada

- ▶ En cada ronda **V** realiza a lo más $f(|w|)$ pasos

Un protocolo aleatorizado

V es una MT **probabilística** que funciona en tiempo $f(|w|)$, donde w es la entrada

- ▶ En cada ronda **V** realiza a lo más $f(|w|)$ pasos

D es determinista y tiene poder de computación ilimitado

Un protocolo aleatorizado

V y **D** comparten dos cintas

Un protocolo aleatorizado

V y **D** comparten dos cintas

- ▶ Una cinta de entrada donde se coloca el string w
 - ▶ Esta cinta es sólo de lectura

Un protocolo aleatorizado

V y **D** comparten dos cintas

- ▶ Una cinta de entrada donde se coloca el string w
 - ▶ Esta cinta es sólo de lectura
- ▶ Una cinta de comunicación donde **V** puede colocar una consulta que es respondida por **D**

Un protocolo aleatorizado

V y **D** comparten dos cintas

- ▶ Una cinta de entrada donde se coloca el string w
 - ▶ Esta cinta es sólo de lectura
- ▶ Una cinta de comunicación donde **V** puede colocar una consulta que es respondida por **D**
 - ▶ Colocar una pregunta o respuesta x en la cinta significa colocar $\vdash xBB \dots$ en ella

Un protocolo aleatorizado

V y **D** comparten dos cintas

- ▶ Una cinta de entrada donde se coloca el string w
 - ▶ Esta cinta es sólo de lectura
- ▶ Una cinta de comunicación donde **V** puede colocar una consulta que es respondida por **D**
 - ▶ Colocar una pregunta o respuesta x en la cinta significa colocar $\vdash x\mathbf{B}\mathbf{B}\dots$ en ella
 - ▶ La respuesta de **D** a cada consulta de **V** debe tener tamaño acotado por $f(|w|)$

Un protocolo aleatorizado

V además tiene dos cintas a las cual **D** no tiene acceso

Un protocolo aleatorizado

V además tiene dos cintas a las cual **D** no tiene acceso

- ▶ Una cinta de trabajo que es de lectura y escritura

Un protocolo aleatorizado

V además tiene dos cintas a las cual **D** no tiene acceso

- ▶ Una cinta de trabajo que es de lectura y escritura
- ▶ Una cinta con bits aleatorios que es sólo de lectura y cuya cabeza sólo se mueve hacia la derecha

Un protocolo aleatorizado

V además tiene dos cintas a las cual **D** no tiene acceso

- ▶ Una cinta de trabajo que es de lectura y escritura
- ▶ Una cinta con bits aleatorios que es sólo de lectura y cuya cabeza sólo se mueve hacia la derecha
 - ▶ Está cinta tiene suficientes bits para todas las rondas

Un protocolo aleatorizado

V además tiene dos cintas a las cual **D** no tiene acceso

- ▶ Una cinta de trabajo que es de lectura y escritura
- ▶ Una cinta con bits aleatorios que es sólo de lectura y cuya cabeza sólo se mueve hacia la derecha
 - ▶ Está cinta tiene suficientes bits para todas las rondas

Inicialmente el protocolo entrega el control a **V**

Un protocolo aleatorizado

V además tiene dos cintas a las cual **D** no tiene acceso

- ▶ Una cinta de trabajo que es de lectura y escritura
- ▶ Una cinta con bits aleatorios que es sólo de lectura y cuya cabeza sólo se mueve hacia la derecha
 - ▶ Está cinta tiene suficientes bits para todas las rondas

Inicialmente el protocolo entrega el control a **V**

- ▶ Este control permanece en el poder de **V**, hasta que **V** realiza una consulta a **D** y le pasa el control

Un protocolo aleatorizado

V además tiene dos cintas a las cual **D** no tiene acceso

- ▶ Una cinta de trabajo que es de lectura y escritura
- ▶ Una cinta con bits aleatorios que es sólo de lectura y cuya cabeza sólo se mueve hacia la derecha
 - ▶ Está cinta tiene suficientes bits para todas las rondas

Inicialmente el protocolo entrega el control a **V**

- ▶ Este control permanece en el poder de **V**, hasta que **V** realiza una consulta a **D** y le pasa el control
- ▶ Una vez que la consulta ha sido respondida **D** le devuelve el control a **V**

Un protocolo aleatorizado

V además tiene dos cintas a las cual **D** no tiene acceso

- ▶ Una cinta de trabajo que es de lectura y escritura
- ▶ Una cinta con bits aleatorios que es sólo de lectura y cuya cabeza sólo se mueve hacia la derecha
 - ▶ Está cinta tiene suficientes bits para todas las rondas

Inicialmente el protocolo entrega el control a **V**

- ▶ Este control permanece en el poder de **V**, hasta que **V** realiza una consulta a **D** y le pasa el control
- ▶ Una vez que la consulta ha sido respondida **D** le devuelve el control a **V**
- ▶ **V** es quien decide si aceptar el string de entrada w

Un protocolo aleatorizado

El número de rondas realizadas por el protocolo (\mathbf{V}, \mathbf{D}) con entrada w se define como el número de veces que el control cambia de dueño

Un protocolo aleatorizado

El número de rondas realizadas por el protocolo (\mathbf{V}, \mathbf{D}) con entrada w se define como el número de veces que el control cambia de dueño

\mathbf{V} debe tener el control al momento de decidir si acepta el string de entrada

La clase $\text{IP}[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

La clase $\text{IP}[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

L está en $\text{IP}[f(n)]$ si existe un verificador \mathbf{V} que funciona en tiempo polinomial (MT aleatorizada de tiempo polinomial) tal que para cada $w \in \Sigma^*$:

La clase $\text{IP}[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

L está en $\text{IP}[f(n)]$ si existe un verificador \mathbf{V} que funciona en tiempo polinomial (MT aleatorizada de tiempo polinomial) tal que para cada $w \in \Sigma^*$:

- ▶ Para cada demostrador \mathbf{D} , el protocolo (\mathbf{V}, \mathbf{D}) con entrada w realiza un número de rondas acotado por $f(|w|)$

La clase $\text{IP}[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

L está en $\text{IP}[f(n)]$ si existe un verificador \mathbf{V} que funciona en tiempo polinomial (MT aleatorizada de tiempo polinomial) tal que para cada $w \in \Sigma^*$:

- ▶ Para cada demostrador \mathbf{D} , el protocolo (\mathbf{V}, \mathbf{D}) con entrada w realiza un número de rondas acotado por $f(|w|)$
- ▶ Si $w \in L$, entonces existe demostrador \mathbf{D} tal que

$$\Pr((\mathbf{V}, \mathbf{D}) \text{ acepte } w) \geq \frac{3}{4}$$

La clase $IP[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

L está en $IP[f(n)]$ si existe un verificador \mathbf{V} que funciona en tiempo polinomial (MT aleatorizada de tiempo polinomial) tal que para cada $w \in \Sigma^*$:

- ▶ Para cada demostrador \mathbf{D} , el protocolo (\mathbf{V}, \mathbf{D}) con entrada w realiza un número de rondas acotado por $f(|w|)$
- ▶ Si $w \in L$, entonces existe demostrador \mathbf{D} tal que

$$\Pr((\mathbf{V}, \mathbf{D}) \text{ acepte } w) \geq \frac{3}{4}$$

- ▶ Si $w \notin L$, entonces para todo demostrador \mathbf{D}' se tiene que

$$\Pr((\mathbf{V}, \mathbf{D}') \text{ acepte } w) \leq \frac{1}{4}$$

La clase IP

Sea

$$\text{IP} = \bigcup_{k \in \mathbb{N}} \text{IP}[n^k]$$

La clase IP

Sea

$$IP = \bigcup_{k \in \mathbb{N}} IP[n^k]$$

Proposición

$$IP \subseteq PSPACE$$

La clase IP

Sea

$$IP = \bigcup_{k \in \mathbb{N}} IP[n^k]$$

Proposición

$$IP \subseteq PSPACE$$

Ejercicio

Demuestre la proposición

- ▶ Piense primero cómo se demuestra que $BPP \subseteq PSPACE$

¿Por qué nos interesan $IP[k]$ y IP ?

No sabemos si $\overline{\text{GRAPH-ISO}} \in NP$

¿Por qué nos interesan $IP[k]$ y IP ?

No sabemos si $\overline{GRAPH-ISO} \in NP$

Pero sí podemos demostrar que existe un protocolo aleatorizado para aceptar grafos no isomorfos:

Proposición

$\overline{GRAPH-ISO} \in IP[4]$

Una demostración de que $\overline{\text{GRAPH-ISO}} \in \text{IP}[4]$

Con entrada (G_1, G_2) el protocolo funciona de la siguiente forma:

Una demostración de que $\overline{\text{GRAPH-ISO}} \in \text{IP}[4]$

Con entrada (G_1, G_2) el protocolo funciona de la siguiente forma:

1. **V** primero revisa si G_1 y G_2 tienen distinto número de nodos. Si es así acepta, si no va al paso 2

Una demostración de que $\overline{\text{GRAPH-ISO}} \in \text{IP}[4]$

Con entrada (G_1, G_2) el protocolo funciona de la siguiente forma:

1. **V** primero revisa si G_1 y G_2 tienen distinto número de nodos. Si es así acepta, si no va al paso 2
2. Sea m el número de nodos de G_1 y G_2

Una demostración de que $\overline{\text{GRAPH-ISO}} \in \text{IP}[4]$

Con entrada (G_1, G_2) el protocolo funciona de la siguiente forma:

1. **V** primero revisa si G_1 y G_2 tienen distinto número de nodos. Si es así acepta, si no va al paso 2
2. Sea m el número de nodos de G_1 y G_2
3. **V** repite 2 veces los pasos 3.1 – 3.5

Una demostración de que $\overline{\text{GRAPH-ISO}} \in \text{IP}[4]$

Con entrada (G_1, G_2) el protocolo funciona de la siguiente forma:

1. **V** primero revisa si G_1 y G_2 tienen distinto número de nodos. Si es así acepta, si no va al paso 2
2. Sea m el número de nodos de G_1 y G_2
3. **V** repite 2 veces los pasos 3.1 – 3.5
 - 3.1 **V** escoge con distribución uniforme un número $i \in \{1, 2\}$ y una permutación $f : \{1, \dots, m\} \rightarrow \{1, \dots, m\}$

Una demostración de que $\overline{\text{GRAPH-ISO}} \in \text{IP}[4]$

Con entrada (G_1, G_2) el protocolo funciona de la siguiente forma:

1. **V** primero revisa si G_1 y G_2 tienen distinto número de nodos. Si es así acepta, si no va al paso 2
2. Sea m el número de nodos de G_1 y G_2
3. **V** repite 2 veces los pasos 3.1 – 3.5
 - 3.1 **V** escoge con distribución uniforme un número $i \in \{1, 2\}$ y una permutación $f : \{1, \dots, m\} \rightarrow \{1, \dots, m\}$
 - 3.2 Sea $H = f(G_i)$

Una demostración de que $\overline{\text{GRAPH-ISO}} \in \text{IP}[4]$

- 3.3 **V** pone H en la cinta de comunicación y pregunta a **D** si es isomorfo a G_1

Una demostración de que $\overline{\text{GRAPH-ISO}} \in \text{IP}[4]$

3.3 **V** pone H en la cinta de comunicación y pregunta a **D** si es isomorfo a G_1

3.4 **D** responde **sí** si H y G_1 son isomorfos, y **no** en caso contrario

Una demostración de que $\overline{\text{GRAPH-ISO}} \in \text{IP}[4]$

- 3.3 **V** pone H en la cinta de comunicación y pregunta a **D** si es isomorfo a G_1
- 3.4 **D** responde **sí** si H y G_1 son isomorfos, y **no** en caso contrario
- 3.5 Si $i = 1$ y **D** respondió **no**, o si $i = 2$ y **D** respondió **sí**, entonces **V** rechaza

Una demostración de que $\overline{\text{GRAPH-ISO}} \in \text{IP}[4]$

3.3 **V** pone H en la cinta de comunicación y pregunta a **D** si es isomorfo a G_1

3.4 **D** responde **sí** si H y G_1 son isomorfos, y **no** en caso contrario

3.5 Si $i = 1$ y **D** respondió **no**, o si $i = 2$ y **D** respondió **sí**, entonces **V** rechaza

4. **V** acepta

Una demostración de que $\overline{\text{GRAPH-ISO}} \in \text{IP}[4]$

El protocolo tiene 4 rondas

Además, tenemos que:

Una demostración de que $\overline{\text{GRAPH-ISO}} \in \text{IP}[4]$

El protocolo tiene 4 rondas

Además, tenemos que:

- ▶ Si G_1 y G_2 no son isomorfos:

$$\Pr((V, D) \text{ acepte } (G_1, G_2)) = 1$$

Una demostración de que $\overline{\text{GRAPH-ISO}} \in \text{IP}[4]$

El protocolo tiene 4 rondas

Además, tenemos que:

- ▶ Si G_1 y G_2 no son isomorfos:

$$\Pr((\mathbf{V}, \mathbf{D}) \text{ acepte } (G_1, G_2)) = 1$$

- ▶ Si G_1 y G_2 son isomorfos, entonces para todo \mathbf{D}' :

$$\Pr((\mathbf{V}, \mathbf{D}') \text{ acepte } (G_1, G_2)) = \frac{1}{4}$$



Podemos disminuir el número de rondas para $\overline{\text{GRAPH-ISO}}$

Corolario

$\overline{\text{GRAPH-ISO}} \in IP[2]$

Podemos disminuir el número de rondas para $\overline{\text{GRAPH-ISO}}$

Corolario

$\overline{\text{GRAPH-ISO}} \in IP[2]$

Ejercicio

Demuestre el corolario

IP contiene a co-NP

Teorema

$$\overline{CNF-SAT} \in IP[2n]$$

IP contiene a co-NP

Teorema

$$\overline{CNF-SAT} \in IP[2n]$$

Corolario

$$co-NP \subseteq IP$$

IP contiene a co-NP

Teorema

$$\overline{CNF-SAT} \in IP[2n]$$

Corolario

$$co-NP \subseteq IP$$

Ejercicio

Demuestre el corolario

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Sea $\varphi = C_1 \wedge \cdots \wedge C_m$ una fórmula en CNF cuyas variables son x_1, \dots, x_n

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Sea $\varphi = C_1 \wedge \cdots \wedge C_m$ una fórmula en CNF cuyas variables son x_1, \dots, x_n

Para cada literal ℓ , defina

$$\tau_\ell = \begin{cases} (1 - x_i) & \ell = x_i \\ x_i & \ell = \neg x_i \end{cases}$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Sea $\varphi = C_1 \wedge \cdots \wedge C_m$ una fórmula en CNF cuyas variables son x_1, \dots, x_n

Para cada literal ℓ , defina

$$\tau_\ell = \begin{cases} (1 - x_i) & \ell = x_i \\ x_i & \ell = \neg x_i \end{cases}$$

Para cada cláusula $C = (\ell_1 \vee \cdots \vee \ell_k)$, defina

$$\tau_C = 1 - \prod_{i=1}^k \tau_{\ell_i}$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Finalmente defina

$$g(x_1, \dots, x_n) = \prod_{i=1}^m \tau_{C_i}$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Finalmente defina

$$g(x_1, \dots, x_n) = \prod_{i=1}^m \tau_{C_i}$$

Por ejemplo, si $\varphi = (x \vee y) \wedge (\neg x \vee z \vee w) \wedge (\neg y \vee \neg w)$, entonces

$$g(x, y, z, w) = (1 - (1 - x) \cdot (1 - y)) \cdot (1 - x \cdot (1 - z) \cdot (1 - w)) \cdot (1 - y \cdot w)$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Para cada valuación $\sigma : \{x_1, \dots, x_n\} \rightarrow \{0, 1\}$, tenemos que:

- ▶ Si $\sigma(\varphi) = 1$, entonces $g(\sigma(x_1), \dots, \sigma(x_n)) = 1$
- ▶ Si $\sigma(\varphi) = 0$, entonces $g(\sigma(x_1), \dots, \sigma(x_n)) = 0$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Para cada valuación $\sigma : \{x_1, \dots, x_n\} \rightarrow \{0, 1\}$, tenemos que:

- ▶ Si $\sigma(\varphi) = 1$, entonces $g(\sigma(x_1), \dots, \sigma(x_n)) = 1$
- ▶ Si $\sigma(\varphi) = 0$, entonces $g(\sigma(x_1), \dots, \sigma(x_n)) = 0$

Para demostrar que φ no es satisfacible, **D** debe demostrar a **V** que:

$$\sum_{(a_1, \dots, a_n) \in \{0, 1\}^n} g(a_1, \dots, a_n) = 0$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Para cada valuación $\sigma : \{x_1, \dots, x_n\} \rightarrow \{0, 1\}$, tenemos que:

- ▶ Si $\sigma(\varphi) = 1$, entonces $g(\sigma(x_1), \dots, \sigma(x_n)) = 1$
- ▶ Si $\sigma(\varphi) = 0$, entonces $g(\sigma(x_1), \dots, \sigma(x_n)) = 0$

Para demostrar que φ no es satisfacible, **D** debe demostrar a **V** que:

$$\sum_{(a_1, \dots, a_n) \in \{0, 1\}^n} g(a_1, \dots, a_n) = 0$$

A continuación vamos a ver un protocolo de demostración interactivo para $\overline{\text{CNF-SAT}}$ que utiliza esta propiedad

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Con entrada φ el protocolo funciona de la siguiente forma:

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Con entrada φ el protocolo funciona de la siguiente forma:

1. **V** le indica a **D** que el protocolo ha comenzado

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Con entrada φ el protocolo funciona de la siguiente forma:

1. **V** le indica a **D** que el protocolo ha comenzado
2. **D** le devuelve a **V** un polinomio $h_1(x_1)$ tal que

$$h_1(x_1) = \sum_{(a_2, \dots, a_n) \in \{0,1\}^{n-1}} g(x_1, a_2, \dots, a_n)$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Con entrada φ el protocolo funciona de la siguiente forma:

1. **V** le indica a **D** que el protocolo ha comenzado
2. **D** le devuelve a **V** un polinomio $h_1(x_1)$ tal que

$$h_1(x_1) = \sum_{(a_2, \dots, a_n) \in \{0,1\}^{n-1}} g(x_1, a_2, \dots, a_n)$$

3. Si el grado de $h_1(x_1)$ es mayor que m entonces **V** rechaza

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Con entrada φ el protocolo funciona de la siguiente forma:

1. **V** le indica a **D** que el protocolo ha comenzado
2. **D** le devuelve a **V** un polinomio $h_1(x_1)$ tal que

$$h_1(x_1) = \sum_{(a_2, \dots, a_n) \in \{0,1\}^{n-1}} g(x_1, a_2, \dots, a_n)$$

3. Si el grado de $h_1(x_1)$ es mayor que m entonces **V** rechaza
4. **V** verifica que $h_1(0) + h_1(1) = 0$, y si no es así entonces rechaza

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Con entrada φ el protocolo funciona de la siguiente forma:

1. **V** le indica a **D** que el protocolo ha comenzado
2. **D** le devuelve a **V** un polinomio $h_1(x_1)$ tal que

$$h_1(x_1) = \sum_{(a_2, \dots, a_n) \in \{0,1\}^{n-1}} g(x_1, a_2, \dots, a_n)$$

3. Si el grado de $h_1(x_1)$ es mayor que m entonces **V** rechaza
4. **V** verifica que $h_1(0) + h_1(1) = 0$, y si no es así entonces rechaza
5. **V** genera al azar con distribución uniforme un número entero $r_1 \in \{0, \dots, 2^{nm} - 1\}$, y se lo envía a **D**

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

6. Los siguientes pasos se repiten para $i = 2, \dots, n$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

6. Los siguientes pasos se repiten para $i = 2, \dots, n$

6.1 **D** le devuelve a **V** un polinomio $h_i(x_i)$ tal que

$$h_i(x_i) = \sum_{(a_{i+1}, \dots, a_n) \in \{0,1\}^{n-i}} g(r_1, \dots, r_{i-1}, x_i, a_{i+1}, \dots, a_n)$$

CNF-SAT está en $IP[2n]$

6. Los siguientes pasos se repiten para $i = 2, \dots, n$

6.1 **D** le devuelve a **V** un polinomio $h_i(x_i)$ tal que

$$h_i(x_i) = \sum_{(a_{i+1}, \dots, a_n) \in \{0,1\}^{n-i}} g(r_1, \dots, r_{i-1}, x_i, a_{i+1}, \dots, a_n)$$

6.2 Si el grado de $h_i(x_i)$ es mayor que m entonces **V** rechaza

CNF-SAT está en $IP[2n]$

6. Los siguientes pasos se repiten para $i = 2, \dots, n$

6.1 **D** le devuelve a **V** un polinomio $h_i(x_i)$ tal que

$$h_i(x_i) = \sum_{(a_{i+1}, \dots, a_n) \in \{0,1\}^{n-i}} g(r_1, \dots, r_{i-1}, x_i, a_{i+1}, \dots, a_n)$$

6.2 Si el grado de $h_i(x_i)$ es mayor que m entonces **V** rechaza

6.3 **V** verifica que $h_{i-1}(r_{i-1}) = h_i(0) + h_i(1)$, y si no es así entonces rechaza

CNF-SAT está en $IP[2n]$

6. Los siguientes pasos se repiten para $i = 2, \dots, n$

6.1 **D** le devuelve a **V** un polinomio $h_i(x_i)$ tal que

$$h_i(x_i) = \sum_{(a_{i+1}, \dots, a_n) \in \{0,1\}^{n-i}} g(r_1, \dots, r_{i-1}, x_i, a_{i+1}, \dots, a_n)$$

6.2 Si el grado de $h_i(x_i)$ es mayor que m entonces **V** rechaza

6.3 **V** verifica que $h_{i-1}(r_{i-1}) = h_i(0) + h_i(1)$, y si no es así entonces rechaza

6.4 **V** genera al azar con distribución uniforme un número entero $r_i \in \{0, \dots, 2^{nm} - 1\}$. Si $i < n$, entonces le envía r_i a **D**

CNF-SAT está en $IP[2n]$

6. Los siguientes pasos se repiten para $i = 2, \dots, n$

6.1 **D** le devuelve a **V** un polinomio $h_i(x_i)$ tal que

$$h_i(x_i) = \sum_{(a_{i+1}, \dots, a_n) \in \{0,1\}^{n-i}} g(r_1, \dots, r_{i-1}, x_i, a_{i+1}, \dots, a_n)$$

6.2 Si el grado de $h_i(x_i)$ es mayor que m entonces **V** rechaza

6.3 **V** verifica que $h_{i-1}(r_{i-1}) = h_i(0) + h_i(1)$, y si no es así entonces rechaza

6.4 **V** genera al azar con distribución uniforme un número entero $r_i \in \{0, \dots, 2^{nm} - 1\}$. Si $i < n$, entonces le envía r_i a **D**

7. **V** verifica si $h_n(r_n) = g(r_1, \dots, r_n)$. Si es así entonces acepta, y en caso contrario rechaza

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

El protocolo tiene $2n$ rondas

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

El protocolo tiene $2n$ rondas

Si φ no es satisfacible, entonces considerando un demostrador **D** que utiliza el polinomio $g(x_1, \dots, x_n)$ obtenemos que:

$$\Pr((\mathbf{V}, \mathbf{D}) \text{ acepte } \varphi) = 1$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Suponga que φ es satisfacible.

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Suponga que φ es satisfacible. Nos falta demostrar que para cualquier demostrador \mathbf{D}' :

$$\Pr((\mathbf{V}, \mathbf{D}') \text{ acepta } \varphi) \leq \frac{1}{4}$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Suponga que φ es satisfacible. Nos falta demostrar que para cualquier demostrador \mathbf{D}' :

$$\Pr((\mathbf{V}, \mathbf{D}') \text{ acepta } \varphi) \leq \frac{1}{4}$$

Vamos a hacer la demostración considerando $n \geq 2$ y $m \geq 2$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Suponga que φ es satisfacible. Nos falta demostrar que para cualquier demostrador \mathbf{D}' :

$$\Pr((\mathbf{V}, \mathbf{D}') \text{ acepta } \varphi) \leq \frac{1}{4}$$

Vamos a hacer la demostración considerando $n \geq 2$ y $m \geq 2$

- ▶ ¿Cómo manejamos los casos en que $n = 1$ o $m = 1$?

CNF-SAT está en $IP[2n]$

Suponga que φ es satisfacible. Nos falta demostrar que para cualquier demostrador D' :

$$\Pr((V, D') \text{ acepta } \varphi) \leq \frac{1}{4}$$

Vamos a hacer la demostración considerando $n \geq 2$ y $m \geq 2$

- ▶ ¿Cómo manejamos los casos en que $n = 1$ o $m = 1$?

Además, suponemos que cada cláusula en φ no tiene literales complementarios ni repetidos

- ▶ ¿Por qué podemos suponer esto?

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Suponga que \mathbf{D}' está tratando de engañar a \mathbf{V}

- ▶ \mathbf{D}' está tratando de que \mathbf{V} acepte φ , aunque esta fórmula es satisfacible

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Suponga que \mathbf{D}' está tratando de engañar a \mathbf{V}

- ▶ \mathbf{D}' está tratando de que \mathbf{V} acepte φ , aunque esta fórmula es satisfacible

Sean $h'_i(x_i)$ los polinomios generados por \mathbf{D}'

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Suponga que \mathbf{D}' está tratando de engañar a \mathbf{V}

- ▶ \mathbf{D}' está tratando de que \mathbf{V} acepte φ , aunque esta fórmula es satisfacible

Sean $h'_i(x_i)$ los polinomios generados por \mathbf{D}'

Tenemos que $h'_1(x_1) \neq h_1(x_1)$

- ▶ Puesto que $h_1(0) + h_1(1) > 0$ y \mathbf{D}' está tratando de engañar a \mathbf{V}

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Si $h'_1(r_1) = h_1(r_1)$, entonces \mathbf{D}' puede definir $h'_2(x_2) = h_2(x_2)$, y desde ahí puede engañar a \mathbf{V}

► Puesto que $h'_2(0) + h'_2(1) = h_2(0) + h_2(1) = h_1(r_1) = h'_1(r_1)$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Si $h'_1(r_1) = h_1(r_1)$, entonces \mathbf{D}' puede definir $h'_2(x_2) = h_2(x_2)$, y desde ahí puede engañar a \mathbf{V}

- ▶ Puesto que $h'_2(0) + h'_2(1) = h_2(0) + h_2(1) = h_1(r_1) = h'_1(r_1)$

Pero si $h'_1(r_1) \neq h_1(r_1)$, entonces se debe tener que $h'_2(x_2) \neq h_2(x_2)$

- ▶ Puesto que $h'_1(r_1)$ debe ser igual a $h'_2(0) + h'_2(1)$ para que \mathbf{D}' pueda engañar a \mathbf{V}

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Si continuamos con este razonamiento vemos que \mathbf{D}' logra engañar a \mathbf{V} si la siguiente condición es cierta:

$$\bigvee_{i=1}^n h'_i(r_i) = h_i(r_i)$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Si continuamos con este razonamiento vemos que \mathbf{D}' logra engañar a \mathbf{V} si la siguiente condición es cierta:

$$\bigvee_{i=1}^n h'_i(r_i) = h_i(r_i)$$

En particular, la condición $h'_n(r_n) = h_n(r_n)$ es equivalente a pedir que $h'_n(r_n) = g(r_1, \dots, r_n)$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Si continuamos con este razonamiento vemos que \mathbf{D}' logra engañar a \mathbf{V} si la siguiente condición es cierta:

$$\bigvee_{i=1}^n h'_i(r_i) = h_i(r_i)$$

En particular, la condición $h'_n(r_n) = h_n(r_n)$ es equivalente a pedir que $h'_n(r_n) = g(r_1, \dots, r_n)$

- ▶ Esta es la última condición que se necesita para que \mathbf{V} acepte

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Por definición del protocolo y dado que ninguna cláusula de φ tiene literales repetidos o complementarios, el grado de cada $h_i(x_i)$ y $h'_i(x'_i)$ es a lo más m

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Por definición del protocolo y dado que ninguna cláusula de φ tiene literales repetidos o complementarios, el grado de cada $h_i(x_i)$ y $h'_i(x'_i)$ es a lo más m

Por lo tanto tenemos que:

CNF-SAT está en $\text{IP}[2n]$

Por definición del protocolo y dado que ninguna cláusula de φ tiene literales repetidos o complementarios, el grado de cada $h_i(x_i)$ y $h'_i(x'_i)$ es a lo más m

Por lo tanto tenemos que:

$$\Pr((\mathbf{V}, \mathbf{D}') \text{ acepta } \varphi) =$$

CNF-SAT está en $\text{IP}[2n]$

Por definición del protocolo y dado que ninguna cláusula de φ tiene literales repetidos o complementarios, el grado de cada $h_i(x_i)$ y $h'_i(x'_i)$ es a lo más m

Por lo tanto tenemos que:

$$\Pr((\mathbf{V}, \mathbf{D}') \text{ acepta } \varphi) = \Pr\left(\bigvee_{i=1}^n h'_i(r_i) = h_i(r_i)\right)$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Por definición del protocolo y dado que ninguna cláusula de φ tiene literales repetidos o complementarios, el grado de cada $h_i(x_i)$ y $h'_i(x'_i)$ es a lo más m

Por lo tanto tenemos que:

$$\begin{aligned}\Pr((\mathbf{V}, \mathbf{D}') \text{ acepta } \varphi) &= \Pr\left(\bigvee_{i=1}^n h'_i(r_i) = h_i(r_i)\right) \\ &= \Pr\left(\bigvee_{i=1}^n \left[h'_i(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right]\right)\end{aligned}$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Por definición del protocolo y dado que ninguna cláusula de φ tiene literales repetidos o complementarios, el grado de cada $h_i(x_i)$ y $h'_i(x'_i)$ es a lo más m

Por lo tanto tenemos que:

$$\begin{aligned}\Pr((\mathbf{V}, \mathbf{D}') \text{ acepta } \varphi) &= \Pr\left(\bigvee_{i=1}^n h'_i(r_i) = h_i(r_i)\right) \\ &= \Pr\left(\bigvee_{i=1}^n \left[h'_i(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right]\right) \\ &= \sum_{i=1}^n \Pr\left(h'_i(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right)\end{aligned}$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Por definición del protocolo y dado que ninguna cláusula de φ tiene literales repetidos o complementarios, el grado de cada $h_i(x_i)$ y $h'_i(x'_i)$ es a lo más m

Por lo tanto tenemos que:

$$\begin{aligned}\Pr((\mathbf{V}, \mathbf{D}') \text{ acepta } \varphi) &= \Pr\left(\bigvee_{i=1}^n h'_i(r_i) = h_i(r_i)\right) \\&= \Pr\left(\bigvee_{i=1}^n \left[h'_i(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right]\right) \\&= \sum_{i=1}^n \Pr\left(h'_i(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right) \\&\leq \sum_{i=1}^n \Pr\left(h'_i(r_i) = h_i(r_i) \mid \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right)\end{aligned}$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Por definición del protocolo y dado que ninguna cláusula de φ tiene literales repetidos o complementarios, el grado de cada $h_i(x_i)$ y $h'_i(x'_i)$ es a lo más m

Por lo tanto tenemos que:

$$\begin{aligned}\Pr((\mathbf{V}, \mathbf{D}') \text{ acepta } \varphi) &= \Pr\left(\bigvee_{i=1}^n h'_i(r_i) = h_i(r_i)\right) \\&= \Pr\left(\bigvee_{i=1}^n \left[h'_i(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right]\right) \\&= \sum_{i=1}^n \Pr\left(h'_i(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right) \\&\leq \sum_{i=1}^n \Pr\left(h'_i(r_i) = h_i(r_i) \mid \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right) \\&\leq \sum_{i=1}^n \frac{m}{2^{nm}}\end{aligned}$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Por definición del protocolo y dado que ninguna cláusula de φ tiene literales repetidos o complementarios, el grado de cada $h_i(x_i)$ y $h'_i(x'_i)$ es a lo más m

Por lo tanto tenemos que:

$$\begin{aligned}\Pr((\mathbf{V}, \mathbf{D}') \text{ acepta } \varphi) &= \Pr\left(\bigvee_{i=1}^n h'_i(r_i) = h_i(r_i)\right) \\&= \Pr\left(\bigvee_{i=1}^n \left[h'_i(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right]\right) \\&= \sum_{i=1}^n \Pr\left(h'_i(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right) \\&\leq \sum_{i=1}^n \Pr\left(h'_i(r_i) = h_i(r_i) \mid \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right) \\&\leq \sum_{i=1}^n \frac{m}{2^{nm}} = \frac{nm}{2^{nm}}\end{aligned}$$

$\overline{\text{CNF-SAT}}$ está en $\text{IP}[2n]$

Por definición del protocolo y dado que ninguna cláusula de φ tiene literales repetidos o complementarios, el grado de cada $h_i(x_i)$ y $h'_i(x'_i)$ es a lo más m

Por lo tanto tenemos que:

$$\begin{aligned}\Pr((\mathbf{V}, \mathbf{D}') \text{ acepta } \varphi) &= \Pr\left(\bigvee_{i=1}^n h'_i(r_i) = h_i(r_i)\right) \\&= \Pr\left(\bigvee_{i=1}^n \left[h'_i(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right]\right) \\&= \sum_{i=1}^n \Pr\left(h'_i(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right) \\&\leq \sum_{i=1}^n \Pr\left(h'_i(r_i) = h_i(r_i) \mid \bigwedge_{j=1}^{i-1} h'_j(r_j) \neq h_j(r_j) \right) \\&\leq \sum_{i=1}^n \frac{m}{2^{nm}} = \frac{nm}{2^{nm}} \leq \frac{1}{4}\end{aligned}$$

□