Protocolos de demostración interactivos

IIC3810

Un protocolo de demostración interactivo para SAT

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

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 $\varphi \in$ SAT

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

- ¿Cómo?

Una noción de protocolo más general

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

- ¿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 $L \in \mathrm{NP}$

- ¿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 $L \in \mathrm{NP}$

- ¿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 $L \in \mathrm{NP}$

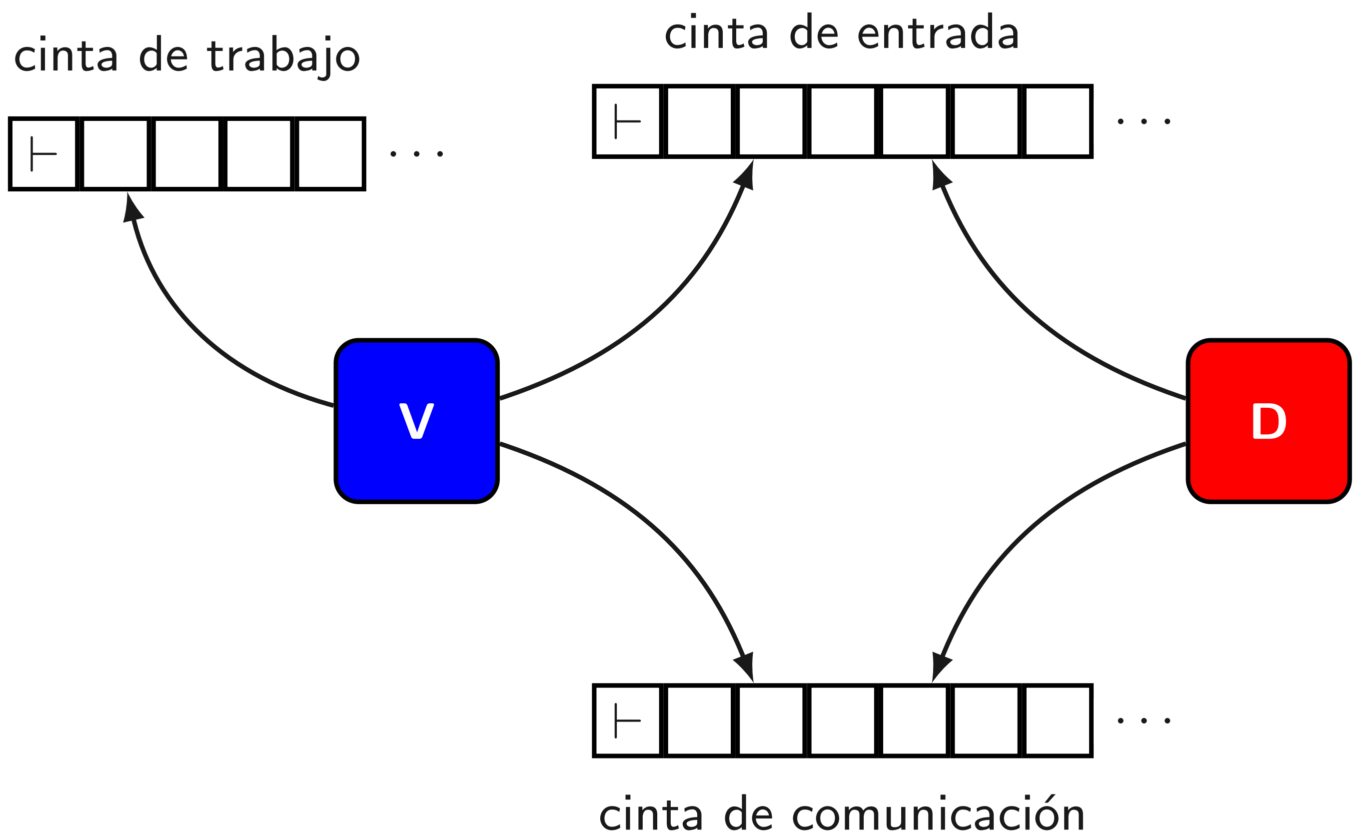
- ¿Cómo?

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 esta 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 $\vdash x\mathrm{BB}\cdots$ 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 $\vdash x \mathrm{BB} \cdots$ 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 **V** a **D** por una consulta, y luego de **D** a **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 **V** a **D** por una consulta, y luego de **D** a **V** por la respuesta a la consulta

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 **V** a **D** por una consulta, y luego de **D** a **V** por la respuesta a la consulta

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

Un protocolo determinista

Algo importante a considerar sobre **D** es cuáles son sus entradas cuando es vista como una función

Un protocolo determinista

Algo importante a considerar sobre **D** es cuáles son sus entradas cuando es vista como una función

Suponga que la entrada del protocolo es w, vale decir, queremos verificar si w está en el lenguaje reconocido por el protocolo

Un protocolo determinista

Algo importante a considerar sobre **D** es cuáles son sus entradas cuando es vista como una función

Suponga que la entrada del protocolo es w, vale decir, queremos verificar si w está en el lenguaje reconocido por el protocolo

Además suponga que **V** en la ronda $2k+1$ decide enviar la pregunta x_{2k+1} a **D**, y **V** había enviado las preguntas x_1, ..., x_{2k-1} a **D** en las rondas anteriores

Un protocolo determinista

Algo importante a considerar sobre **D** es cuáles son sus entradas cuando es vista como una función

Suponga que la entrada del protocolo es w, vale decir, queremos verificar si w está en el lenguaje reconocido por el protocolo

Además suponga que **V** en la ronda $2k + 1$ decide enviar la pregunta x_{2k+1} a **D**, y **V** había enviado las preguntas x_1, ..., x_{2k-1} a **D** en las rondas anteriores

- La respuesta de **D** en la ronda $2k + 2$ depende de w, x_1, ..., x_{2k-1}, x_{2k+1}

Un protocolo determinista

Algo importante a considerar sobre **D** es cuáles son sus entradas cuando es vista como una función

Suponga que la entrada del protocolo es w, vale decir, queremos verificar si w está en el lenguaje reconocido por el protocolo

Además suponga que **V** en la ronda $2k + 1$ decide enviar la pregunta x_{2k+1} a **D**, y **V** había enviado las preguntas $x_1, \ldots, x_{2k-1}$ a **D** en las rondas anteriores

- La respuesta de **D** en la ronda $2k + 2$ depende de $w, x_1, \ldots, x_{2k-1}, x_{2k+1}$
- **D** es una función que dependen de toda la información que ha visto al momento de ser consultado

La clase dIP$[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

La clase dIP$[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

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

La clase dIP[$f(n)$]

Sea L un lenguaje sobre un alfabeto Σ

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

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

La clase dIP[$f(n)$]

Sea L un lenguaje sobre un alfabeto Σ

L está en 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 dIP[$f(n)$]

Sea L un lenguaje sobre un alfabeto Σ

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

- Para cada demostrador **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 **D** tal que (**V**,**D**) acepta w
- Si $w \notin L$, entonces para todo demostrador **D'** se tiene que (**V**,**D'**) rechaza w

La clase dIP[k]

Ejercicio

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

La clase dIP

Sea

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

La clase dIP

Sea

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

Proposición

$dIP = NP$

La clase dIP

Sea

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

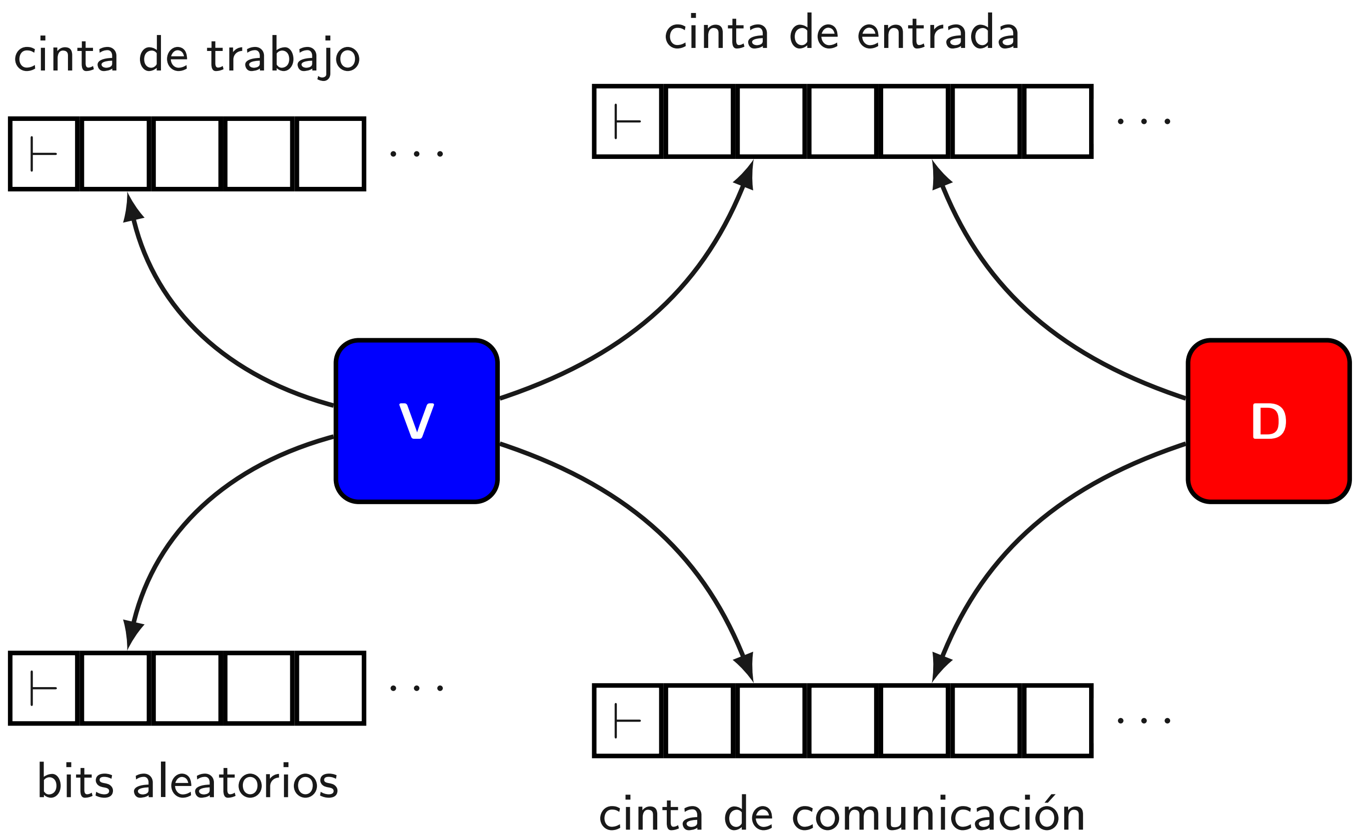
Proposición

$dIP = NP$

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 $\vdash x\mathrm{BB}\cdots$ 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 $\vdash x\text{BB}\cdots$ 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

V debe tener el control al momento de decidir si acepta el string de entrada

La clase IP$[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

La clase IP$[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

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

La clase IP$[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

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

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

La clase IP$[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

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

- Para cada demostrador **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 **D** tal que

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

La clase IP$[f(n)]$

Sea L un lenguaje sobre un alfabeto Σ

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

- Para cada demostrador **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 **D** tal que

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

- Si $w \notin L$, entonces para todo demostrador **D'** se tiene que

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

La clase IP

Sea

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

La clase IP

Sea

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

Vamos a ver ejemplos de protocolos interactivos que nos permiten entender el poder de IP

La clase IP

Sea

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

Vamos a ver ejemplos de protocolos interactivos que nos permiten entender el poder de IP

- Y vamos a caracterizar IP en términos de las clases de complejidad usuales

¿Por qué nos interesan IP[k] y IP?

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

¿Por qué nos interesan IP[k] y IP?

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

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

Proposición

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

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 $f : \{1, \ldots, m\} \to \{1, \ldots, m\}$

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 $f : \{1, \ldots, m\} \to \{1, \ldots, m\}$

    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

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:

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:

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

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:

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

- Si G_1 y G_2 son isomorfos, entonces para todo **D'**:

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:

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

- Si G_1 y G_2 son isomorfos, entonces para todo **D'**:

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

$\square$

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

Corolario

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

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

Corolario

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

Ejercicio

Demuestre el corolario

IP contiene a co-NP

Teorema

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

IP contiene a co-NP

Teorema

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

Corolario

$\mathit{co\text{-}NP} \subseteq \mathit{IP}$

IP contiene a co-NP

Teorema

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

Corolario

$\mathit{co\text{-}NP} \subseteq \mathit{IP}$

Ejercicio

Demuestre el corolario

Un resultado más fuerte

Defina el siguiente lenguaje:

$$\text{COUNT-CNF-SAT} \;=\; \{(\varphi, k) \mid \varphi \text{ es una fórmula en CNF y} \\ \text{el número de valuaciones que satisface a } \varphi \text{ es } k\}$$

Un resultado más fuerte

Defina el siguiente lenguaje:

$$\text{COUNT-CNF-SAT} = \{(\varphi, k) \mid \varphi \text{ es una fórmula en CNF y} \\ \text{el número de valuaciones que satisface a } \varphi \text{ es } k\}$$

Teorema

COUNT-CNF-SAT $\in$ *IP*$[2n]$

Un resultado más fuerte

Defina el siguiente lenguaje:

$$\text{COUNT-CNF-SAT} = \{(\varphi, k) \mid \varphi \text{ es una fórmula en CNF y el número de valuaciones que satisface a } \varphi \text{ es } k\}$$

Teorema

COUNT-CNF-SAT $\in$ *IP*$[2n]$

Ejercicio

Demuestre usando el teorema que $\overline{\text{CNF-SAT}} \in \text{IP}[2n]$

COUNT-CNF-SAT está en IP[$2n$]

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

COUNT-CNF-SAT está en IP[$2n$]

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

Suponemos que cada cláusula en φ no tiene literales complementarios ni repetidos

- ¿Por qué podemos suponer esto?

COUNT-CNF-SAT está en IP[$2n$]

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

Suponemos que cada cláusula en φ no tiene literales complementarios ni repetidos

- ¿Por qué podemos suponer esto?

Además, para la definición del protocolo interactivo suponemos que $n \geq 2$ y $m \geq 2$

COUNT-CNF-SAT está en IP[$2n$]

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

Suponemos que cada cláusula en φ no tiene literales complementarios ni repetidos

- ¿Por qué podemos suponer esto?

Además, para la definición del protocolo interactivo suponemos que $n \geq 2$ y $m \geq 2$

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

COUNT-CNF-SAT está en IP[$2n$]

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

Suponemos que cada cláusula en φ no tiene literales complementarios ni repetidos

- ¿Por qué podemos suponer esto?

Además, para la definición del protocolo interactivo suponemos que $n \geq 2$ y $m \geq 2$

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

COUNT-CNF-SAT está en IP[$2n$]

Para cada literal ℓ, defina

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

COUNT-CNF-SAT está en IP[$2n$]

Para cada literal ℓ, defina

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

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

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

COUNT-CNF-SAT está en IP$[2n]$

Finalmente defina

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

COUNT-CNF-SAT está en IP[$2n$]

Finalmente defina

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

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

$$\begin{aligned} g(x, y, z, w) \ &= \\ &(1 - (1 - x) \cdot (1 - y)) \cdot (1 - x \cdot (1 - z) \cdot (1 - w)) \cdot (1 - y \cdot w) \end{aligned}$$

COUNT-CNF-SAT está en IP[$2n$]

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

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

COUNT-CNF-SAT está en IP[$2n$]

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

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

Para demostrar que $(\varphi, k) \in$ COUNT-CNF-SAT, **D** debe demostrar a **V** que:

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

COUNT-CNF-SAT está en IP[$2n$]

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

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

Para demostrar que $(\varphi, k) \in$ COUNT-CNF-SAT, **D** debe demostrar a **V** que:

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

A continuación vamos a ver un protocolo de demostración interactivo para COUNT-CNF-SAT que utiliza esta propiedad

COUNT-CNF-SAT está en IP[$2n$]

Con entrada φ el protocolo funciona de la siguiente forma:

COUNT-CNF-SAT está en IP[$2n$]

Con entrada φ el protocolo funciona de la siguiente forma:

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

COUNT-CNF-SAT está en 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

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

COUNT-CNF-SAT está en 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

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

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

COUNT-CNF-SAT está en 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

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

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

COUNT-CNF-SAT está en 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

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

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

COUNT-CNF-SAT está en IP[$2n$]

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

COUNT-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

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

COUNT-CNF-SAT está en IP[$2n$]

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

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

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

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

COUNT-CNF-SAT está en IP[2n]

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

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

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

   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

COUNT-CNF-SAT está en IP[$2n$]

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

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

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

   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 $r_i \in \{0, \ldots, 2^{nm} - 1\}$. Si $i < n$, entonces le envía r_i a **D**

COUNT-CNF-SAT está en IP[$2n$]

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

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

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

   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 $r_i \in \{0, \ldots, 2^{nm} - 1\}$. Si $i < n$, entonces le envía r_i a **D**

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

COUNT-CNF-SAT está en IP[$2n$]

El protocolo tiene $2n$ rondas

COUNT-CNF-SAT está en IP[$2n$]

El protocolo tiene $2n$ rondas

Si $(\varphi, k) \in$ COUNT-CNF-SAT, entonces considerando un demostrador **D** que utiliza el polinomio $g(x_1, \ldots, x_n)$ obtenemos que:

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

COUNT-CNF-SAT está en IP[$2n$]

El protocolo tiene $2n$ rondas

Si $(\varphi, k) \in$ COUNT-CNF-SAT, entonces considerando un demostrador **D** que utiliza el polinomio $g(x_1, \ldots, x_n)$ obtenemos que:

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

Suponga que $(\varphi, k) \notin$ COUNT-CNF-SAT.

COUNT-CNF-SAT está en IP$[2n]$

El protocolo tiene $2n$ rondas

Si $(\varphi, k) \in$ COUNT-CNF-SAT, entonces considerando un demostrador **D** que utiliza el polinomio $g(x_1, \ldots, x_n)$ obtenemos que:

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

Suponga que $(\varphi, k) \notin$ COUNT-CNF-SAT. Nos falta demostrar que para cualquier demostrador **D'**:

$$\mathbf{Pr}\big((\mathbf{V}, \mathbf{D'}) \text{ acepte } (\varphi, k)\big) \quad \leq \quad \frac{1}{4}$$

COUNT-CNF-SAT está en IP[$2n$]

Suponga que **D'** está tratando de engañar a **V**

- **D'** está tratando de que **V** acepte (φ, k), aunque el número de valuaciones que satisfacen a φ no es k

COUNT-CNF-SAT está en IP[$2n$]

Suponga que **D'** está tratando de engañar a **V**

- **D'** está tratando de que **V** acepte (φ, k), aunque el número de valuaciones que satisfacen a φ no es k

Sean $h_i'(x_i)$ los polinomios generados por **D'**

COUNT-CNF-SAT está en IP[$2n$]

Suponga que **D'** está tratando de engañar a **V**

- **D'** está tratando de que **V** acepte (φ, k), aunque el número de valuaciones que satisfacen a φ no es k

Sean $h_i'(x_i)$ los polinomios generados por **D'**

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

- Puesto que $h_1(0) + h_1(1) \neq k$ y **D'** está tratando de engañar a **V**

COUNT-CNF-SAT está en 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 $h_2'(0) + h_2'(1) = h_2(0) + h_2(1) = h_1(r_1) = h_1'(r_1)$

COUNT-CNF-SAT está en 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 **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 **D'** pueda engañar a **V**

COUNT-CNF-SAT está en IP[$2n$]

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

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

COUNT-CNF-SAT está en IP[$2n$]

Si continuamos con este razonamiento vemos que **D'** logra engañar a **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 $h'_n(r_n) = g(r_1, \ldots, r_n)$

COUNT-CNF-SAT está en IP[$2n$]

Si continuamos con este razonamiento vemos que **D’** logra engañar a **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 $h'_n(r_n) = g(r_1, \ldots, r_n)$

- Esta es la última condición que se necesita para que **V** acepte

COUNT-CNF-SAT está en 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

COUNT-CNF-SAT está en 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:

COUNT-CNF-SAT está en 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:

$$\mathbf{Pr}\big((\mathbf{V}, \mathbf{D'}) \text{ acepte } (\varphi, k)\big) \quad =$$

COUNT-CNF-SAT está en 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:

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

COUNT-CNF-SAT está en 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:

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

COUNT-CNF-SAT está en 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:

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

COUNT-CNF-SAT está en 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:

$$\begin{aligned}
\mathbf{Pr}\big((\mathbf{V},\mathbf{D'})\text{ acepte }(\varphi,k)\big) &= \mathbf{Pr}\left(\bigvee_{i=1}^{n} h_i'(r_i) = h_i(r_i)\right) \\
&= \mathbf{Pr}\left(\bigvee_{i=1}^{n}\left[h_i'(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h_j'(r_j) \neq h_j(r_j)\right]\right) \\
&= \sum_{i=1}^{n}\mathbf{Pr}\left(h_i'(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h_j'(r_j) \neq h_j(r_j)\right) \\
&\leq \sum_{i=1}^{n}\mathbf{Pr}\left(h_i'(r_i) = h_i(r_i) \;\middle|\; \bigwedge_{j=1}^{i-1} h_j'(r_j) \neq h_j(r_j)\right)
\end{aligned}$$

COUNT-CNF-SAT está en 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:

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

COUNT-CNF-SAT está en 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:

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

COUNT-CNF-SAT está en 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:

$$
\begin{aligned}
\mathbf{Pr}\big((\mathbf{V}, \mathbf{D'}) \text{ acepte } (\varphi, k)\big) \quad &= \quad \mathbf{Pr}\Big(\bigvee_{i=1}^{n} h_i'(r_i) = h_i(r_i)\Big) \\
&= \quad \mathbf{Pr}\Big(\bigvee_{i=1}^{n} \Big[h_i'(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h_j'(r_j) \neq h_j(r_j)\Big]\Big) \\
&= \quad \sum_{i=1}^{n} \mathbf{Pr}\Big(h_i'(r_i) = h_i(r_i) \wedge \bigwedge_{j=1}^{i-1} h_j'(r_j) \neq h_j(r_j)\Big) \\
&\leq \quad \sum_{i=1}^{n} \mathbf{Pr}\Big(h_i'(r_i) = h_i(r_i) \;\Big|\; \bigwedge_{j=1}^{i-1} h_j'(r_j) \neq h_j(r_j)\Big) \\
&\leq \quad \sum_{i=1}^{n} \frac{m}{2^{nm}} \;=\; \frac{nm}{2^{nm}} \;\leq\; \frac{1}{4} \qquad \square
\end{aligned}
$$