

# Relacionando el muestreo (casi) uniforme con la existencia de un FPRAS

IIC3810

Marcelo Arenas y Luis Alberto Croquevielle

# La noción de $p$ -relación

Será conveniente ver las funciones de  $\#P$  como relaciones.

## Definición

Una relación  $R \subseteq \Sigma^* \times \Sigma^*$  es una  $p$ -relación si:

- ▶ Existe un polinomio  $q$  tal que si  $(x, y) \in R$ , entonces  $|y| \leq q(|x|)$
- ▶  $R \in PTIME$ , vale decir, existe un algoritmo de tiempo polinomial que, dado  $(x, y) \in \Sigma^* \times \Sigma^*$ , verifica si  $(x, y) \in R$

# Cada $p$ -relación representa una función en $\#P$

Dada una relación  $R \subseteq \Sigma^* \times \Sigma^*$ , defina la función  $f_R : \Sigma^* \rightarrow \mathbb{N}$  como:

$$f_R(x) = \begin{cases} |\{y \mid (x, y) \in R\}| & \text{si } \{y \mid (x, y) \in R\} \text{ es finito} \\ 0 & \text{en otro caso} \end{cases}$$

# Cada $p$ -relación representa una función en $\#P$

Dada una relación  $R \subseteq \Sigma^* \times \Sigma^*$ , defina la función  $f_R : \Sigma^* \rightarrow \mathbb{N}$  como:

$$f_R(x) = \begin{cases} |\{y \mid (x, y) \in R\}| & \text{si } \{y \mid (x, y) \in R\} \text{ es finito} \\ 0 & \text{en otro caso} \end{cases}$$

## Proposición

*Si  $R$  es una  $p$ -relación, entonces  $f_R \in \#P$*

# Cada $p$ -relación representa una función en $\#P$

Dada una relación  $R \subseteq \Sigma^* \times \Sigma^*$ , defina la función  $f_R : \Sigma^* \rightarrow \mathbb{N}$  como:

$$f_R(x) = \begin{cases} |\{y \mid (x, y) \in R\}| & \text{si } \{y \mid (x, y) \in R\} \text{ es finito} \\ 0 & \text{en otro caso} \end{cases}$$

## Proposición

*Si  $R$  es una  $p$ -relación, entonces  $f_R \in \#P$*

## Ejercicio

Demuestre la proposición.

# Cada función en $\#P$ puede ser representada como una $p$ -relación

Sea  $f : \Sigma^* \rightarrow \mathbb{N}$  una función en  $\#P$

- ▶ Existe una MT no determinista  $M$  tal que para todo  $x \in \Sigma^*$  se tiene que  $f(x) = \text{accept}_M(x)$

# Cada función en $\#P$ puede ser representada como una $p$ -relación

Sea  $f : \Sigma^* \rightarrow \mathbb{N}$  una función en  $\#P$

- ▶ Existe una MT no determinista  $M$  tal que para todo  $x \in \Sigma^*$  se tiene que  $f(x) = \text{accept}_M(x)$

Cada ejecución de  $M$  se puede codificar usando el alfabeto  $\Sigma$

# Cada función en $\#P$ puede ser representada como una $p$ -relación

Sea  $f : \Sigma^* \rightarrow \mathbb{N}$  una función en  $\#P$

- Existe una MT no determinista  $M$  tal que para todo  $x \in \Sigma^*$  se tiene que  $f(x) = \text{accept}_M(x)$

Cada ejecución de  $M$  se puede codificar usando el alfabeto  $\Sigma$

Utilizando las codificaciones de las ejecuciones de  $M$  definimos:

$$R_f = \{(x, y) \in \Sigma^* \times \Sigma^* \mid y \text{ codifica una ejecución de } M \\ \text{con entrada } x \text{ que termina en un estado final}\}$$

Cada función en  $\#P$  puede ser representada como una  $p$ -relación

### Proposición

*Si  $f$  está en  $\#P$ , entonces  $R_f$  es una  $p$ -relación*

Cada función en  $\#P$  puede ser representada como una  $p$ -relación

### Proposición

*Si  $f$  está en  $\#P$ , entonces  $R_f$  es una  $p$ -relación*

**Demostración:** Como la MT no determinista  $M$  en la transparencia anterior es de tiempo polinomial, para una entrada  $x$  se puede:

- ▶ Codificar cualquier ejecución de  $M$  que acepta usando un string de largo polinomial en  $|x|$
- ▶ Verificar si una ejecución termina en estado final (simulando el funcionamiento de  $M$ ) en tiempo polinomial

# Funciones en $\#P$ y $p$ -relaciones

Por lo tanto, de ahora en adelante trabajamos con  $p$ -relaciones.

Estudiaremos los problemas de conteo y de generación uniforme asociados a  $p$ -relaciones, sabiendo que los resultados se extienden de manera natural a funciones en  $\#P$

# Un generador uniforme para una $p$ -relación

Dada una  $p$ -relación  $R \subseteq \Sigma^* \times \Sigma^*$ , sea:

$$N_R(x) = |\{y \in \Sigma^* \mid (x, y) \in R\}|$$

Además, suponga que  $\perp$  es un símbolo reservado que no es usado en  $\Sigma$

# Un generador uniforme para una $p$ -relación

Dada una  $p$ -relación  $R \subseteq \Sigma^* \times \Sigma^*$ , sea:

$$N_R(x) = |\{y \in \Sigma^* \mid (x, y) \in R\}|$$

Además, suponga que  $\perp$  es un símbolo reservado que no es usado en  $\Sigma$

Un algoritmo aleatorizado  $\mathcal{G} : \Sigma^* \rightarrow \Sigma^* \cup \{\perp\}$  es un generador uniforme para  $R$  si para todo  $x, y \in \Sigma^*$ :

- ▶ si  $(x, y) \notin R$ , entonces  $\Pr(\mathcal{G}(x) = y) = 0$
- ▶ si  $(x, y) \in R$ , entonces  $\Pr(\mathcal{G}(x) = y) = \frac{1}{N_R(x)}$

# Un generador casi uniforme para una $p$ -relación

Las herramientas que veremos más adelante no nos permitirán obtener generadores uniformes, sino que una versión más débil

- Consideramos nuevamente una  $p$ -relación  $R \subseteq \Sigma^* \times \Sigma^*$

# Un generador casi uniforme para una $p$ -relación

Las herramientas que veremos más adelante no nos permitirán obtener generadores uniformes, sino que una versión más débil

- Consideramos nuevamente una  $p$ -relación  $R \subseteq \Sigma^* \times \Sigma^*$

## Definición

*Un algoritmo aleatorizado  $\mathcal{G} : \Sigma^* \times (0, 1) \rightarrow \Sigma^* \cup \{\perp\}$  es un generador casi uniforme para  $R$  si para todo  $x, y \in \Sigma^*$  y  $\varepsilon \in (0, 1)$ :*

# Un generador casi uniforme para una $p$ -relación

Las herramientas que veremos más adelante no nos permitirán obtener generadores uniformes, sino que una versión más débil

- Consideramos nuevamente una  $p$ -relación  $R \subseteq \Sigma^* \times \Sigma^*$

## Definición

Un algoritmo aleatorizado  $\mathcal{G} : \Sigma^* \times (0, 1) \rightarrow \Sigma^* \cup \{\perp\}$  es un generador casi uniforme para  $R$  si para todo  $x, y \in \Sigma^*$  y  $\varepsilon \in (0, 1)$ :

- si  $(x, y) \notin R$ , entonces  $\Pr(\mathcal{G}(x, \varepsilon) = y) = 0$

# Un generador casi uniforme para una $p$ -relación

Las herramientas que veremos más adelante no nos permitirán obtener generadores uniformes, sino que una versión más débil

- Consideramos nuevamente una  $p$ -relación  $R \subseteq \Sigma^* \times \Sigma^*$

## Definición

Un algoritmo aleatorizado  $\mathcal{G} : \Sigma^* \times (0, 1) \rightarrow \Sigma^* \cup \{\perp\}$  es un generador casi uniforme para  $R$  si para todo  $x, y \in \Sigma^*$  y  $\varepsilon \in (0, 1)$ :

- si  $(x, y) \notin R$ , entonces  $\Pr(\mathcal{G}(x, \varepsilon) = y) = 0$
- si  $(x, y) \in R$ , entonces:

$$\frac{1}{(1 + \varepsilon) \cdot N_R(x)} \leq \Pr(\mathcal{G}(x, \varepsilon) = y) \leq (1 + \varepsilon) \cdot \frac{1}{N_R(x)}$$

¿Por qué utilizamos  $\frac{1}{(1 + \varepsilon)}$  en lugar de  $(1 - \varepsilon)$ ?

Si reemplazamos  $\frac{1}{(1 + \varepsilon)}$  por  $(1 - \varepsilon)$  en la definición anterior obtenemos una definición equivalente.

► Vamos a demostrar esto

¿Por qué utilizamos  $\frac{1}{(1 + \varepsilon)}$  en lugar de  $(1 - \varepsilon)$ ?

Si reemplazamos  $\frac{1}{(1 + \varepsilon)}$  por  $(1 - \varepsilon)$  en la definición anterior obtenemos una definición equivalente.

► Vamos a demostrar esto

Pero para establecer una relación entre muestreo (casi) uniforme y la existencia de un FPRAS es más conveniente utilizar  $\frac{1}{(1 + \varepsilon)}$  en lugar de  $(1 - \varepsilon)$

# Una definición alternativa de generador casi uniforme

Consideramos una definición alternativa de generador casi uniforme para una  $p$ -relación  $R \subseteq \Sigma^* \times \Sigma^*$

# Una definición alternativa de generador casi uniforme

Consideramos una definición alternativa de generador casi uniforme para una  $p$ -relación  $R \subseteq \Sigma^* \times \Sigma^*$

Un algoritmo aleatorizado  $\mathcal{H} : \Sigma^* \times (0, 1) \rightarrow \Sigma^* \cup \{\perp\}$  es un generador casi uniforme para  $R$  si para todo  $x, y \in \Sigma^*$  y  $\delta \in (0, 1)$ :

- ▶ si  $(x, y) \notin R$ , entonces  $\Pr(\mathcal{H}(x, \delta) = y) = 0$
- ▶ si  $(x, y) \in R$ , entonces:

$$(1 - \delta) \cdot \frac{1}{N_R(x)} \leq \Pr(\mathcal{H}(x, \delta) = y) \leq (1 + \delta) \cdot \frac{1}{N_R(x)}$$

# Las definiciones son equivalentes

Vamos a demostrar que las definiciones dadas en las dos transparencias anteriores son equivalentes.

De manera precisa, dada una  $p$ -relación  $R \subseteq \Sigma^* \times \Sigma^*$ , vamos a demostrar que:

Si hay un generador casi uniforme para  $R$  bajo la primera definición, entonces hay un generador casi uniforme para  $R$  bajo la segunda definición, y viceversa

# La primera definición implica la segunda

Suponga que  $\mathcal{G} : \Sigma^* \times (0, 1) \rightarrow \Sigma^* \cup \{\perp\}$  es un generador casi uniforme para  $R$  según la primera definición.

► Y sea  $(x, y) \in R$

Para todo  $\varepsilon \in (0, 1)$  tenemos que:

$$\frac{1}{(1 + \varepsilon) \cdot N_R(x)} \leq \mathbf{Pr}(\mathcal{G}(x, \varepsilon) = y) \leq (1 + \varepsilon) \cdot \frac{1}{N_R(x)} \quad (\dagger)$$

# La primera definición implica la segunda

Suponga que  $\mathcal{G} : \Sigma^* \times (0, 1) \rightarrow \Sigma^* \cup \{\perp\}$  es un generador casi uniforme para  $R$  según la primera definición.

► Y sea  $(x, y) \in R$

Para todo  $\varepsilon \in (0, 1)$  tenemos que:

$$\frac{1}{(1 + \varepsilon) \cdot N_R(x)} \leq \Pr(\mathcal{G}(x, \varepsilon) = y) \leq (1 + \varepsilon) \cdot \frac{1}{N_R(x)} \quad (\dagger)$$

En este caso consideramos  $\mathcal{H} = \mathcal{G}$

# La primera definición implica la segunda

Suponga que  $\mathcal{G} : \Sigma^* \times (0, 1) \rightarrow \Sigma^* \cup \{\perp\}$  es un generador casi uniforme para  $R$  según la primera definición.

► Y sea  $(x, y) \in R$

Para todo  $\varepsilon \in (0, 1)$  tenemos que:

$$\frac{1}{(1 + \varepsilon) \cdot N_R(x)} \leq \Pr(\mathcal{G}(x, \varepsilon) = y) \leq (1 + \varepsilon) \cdot \frac{1}{N_R(x)} \quad (\dagger)$$

En este caso consideramos  $\mathcal{H} = \mathcal{G}$

Dado  $\delta \in (0, 1)$ , tenemos que demostrar que:

$$(1 - \delta) \cdot \frac{1}{N_R(x)} \leq \Pr(\mathcal{H}(x, \delta) = y) \leq (1 + \delta) \cdot \frac{1}{N_R(x)}$$

# La primera definición implica la segunda

Pero tenemos que:

$$\begin{aligned}\delta > 0 &\Rightarrow \delta^2 > 0 \\ &\Rightarrow -\delta^2 < 0 \\ &\Rightarrow 1 - \delta^2 < 1 \\ &\Rightarrow (1 - \delta) \cdot (1 + \delta) < 1 \\ &\Rightarrow (1 - \delta) < \frac{1}{(1 + \delta)}\end{aligned}$$

# La primera definición implica la segunda

Pero tenemos que:

$$\begin{aligned}\delta > 0 &\Rightarrow \delta^2 > 0 \\ &\Rightarrow -\delta^2 < 0 \\ &\Rightarrow 1 - \delta^2 < 1 \\ &\Rightarrow (1 - \delta) \cdot (1 + \delta) < 1 \\ &\Rightarrow (1 - \delta) < \frac{1}{(1 + \delta)}\end{aligned}$$

Por lo tanto considerando  $\varepsilon = \delta$  en  $(\dagger)$  y el hecho que  $\mathcal{H}(x, \delta) = \mathcal{G}(x, \delta) = \mathcal{G}(x, \varepsilon)$  obtenemos:

$$(1 - \delta) \cdot \frac{1}{N_R(x)} < \frac{1}{(1 + \varepsilon) \cdot N_R(x)} \leq \Pr(\mathcal{H}(x, \delta) = y) \leq (1 + \varepsilon) \cdot \frac{1}{N_R(x)} = (1 + \delta) \cdot \frac{1}{N_R(x)}$$

# La segunda definición implica la primera

Suponga que  $\mathcal{H} : \Sigma^* \times (0, 1) \rightarrow \Sigma^* \cup \{\perp\}$  es un generador casi uniforme para  $R$  según la segunda definición.

► Y sea  $(x, y) \in R$

Para todo  $\delta \in (0, 1)$  tenemos que:

$$(1 - \delta) \cdot \frac{1}{N_R(x)} \leq \mathbf{Pr}(\mathcal{H}(x, \delta) = y) \leq (1 + \delta) \cdot \frac{1}{N_R(x)} \quad (\ddagger)$$

# La segunda definición implica la primera

Suponga que  $\mathcal{H} : \Sigma^* \times (0, 1) \rightarrow \Sigma^* \cup \{\perp\}$  es un generador casi uniforme para  $R$  según la segunda definición.

► Y sea  $(x, y) \in R$

Para todo  $\delta \in (0, 1)$  tenemos que:

$$(1 - \delta) \cdot \frac{1}{N_R(x)} \leq \mathbf{Pr}(\mathcal{H}(x, \delta) = y) \leq (1 + \delta) \cdot \frac{1}{N_R(x)} \quad (\ddagger)$$

En este caso consideramos  $\mathcal{G}$  definido como  $\mathcal{G}(x, \varepsilon) = \mathcal{H}\left(x, \frac{\varepsilon}{(1 + \varepsilon)}\right)$

# La segunda definición implica la primera

Suponga que  $\mathcal{H} : \Sigma^* \times (0, 1) \rightarrow \Sigma^* \cup \{\perp\}$  es un generador casi uniforme para  $R$  según la segunda definición.

► Y sea  $(x, y) \in R$

Para todo  $\delta \in (0, 1)$  tenemos que:

$$(1 - \delta) \cdot \frac{1}{N_R(x)} \leq \Pr(\mathcal{H}(x, \delta) = y) \leq (1 + \delta) \cdot \frac{1}{N_R(x)} \quad (\ddagger)$$

En este caso consideramos  $\mathcal{G}$  definido como  $\mathcal{G}(x, \varepsilon) = \mathcal{H}\left(x, \frac{\varepsilon}{(1 + \varepsilon)}\right)$

Dado  $\varepsilon \in (0, 1)$ , tenemos que demostrar que:

$$\frac{1}{(1 + \varepsilon) \cdot N_R(x)} \leq \Pr(\mathcal{G}(x, \varepsilon) = y) \leq (1 + \varepsilon) \cdot \frac{1}{N_R(x)}$$

# La primera definición implica la segunda

Tenemos que:

$$\begin{aligned}\delta = \frac{\varepsilon}{(1 + \varepsilon)} &\Rightarrow \delta + \delta \cdot \varepsilon = \varepsilon \\ &\Rightarrow 0 = \varepsilon - \delta - \delta \cdot \varepsilon \\ &\Rightarrow 1 = 1 + \varepsilon - \delta - \delta \cdot \varepsilon \\ &\Rightarrow 1 = (1 - \delta) \cdot (1 + \varepsilon) \\ &\Rightarrow \frac{1}{(1 + \varepsilon)} = (1 - \delta)\end{aligned}$$

# La primera definición implica la segunda

Tenemos que:

$$\begin{aligned}\delta = \frac{\varepsilon}{(1 + \varepsilon)} &\Rightarrow \delta + \delta \cdot \varepsilon = \varepsilon \\ &\Rightarrow 0 = \varepsilon - \delta - \delta \cdot \varepsilon \\ &\Rightarrow 1 = 1 + \varepsilon - \delta - \delta \cdot \varepsilon \\ &\Rightarrow 1 = (1 - \delta) \cdot (1 + \varepsilon) \\ &\Rightarrow \frac{1}{(1 + \varepsilon)} = (1 - \delta)\end{aligned}$$

Por lo tanto considerando  $\delta = \frac{\varepsilon}{(1+\varepsilon)}$  en  $(\dagger)$  y los hechos que  $\frac{\varepsilon}{(1+\varepsilon)} \leq \varepsilon$  y  $\mathcal{G}(x, \varepsilon) = \mathcal{H}(x, \frac{\varepsilon}{(1+\varepsilon)}) = \mathcal{H}(x, \delta)$  obtenemos:

$$\begin{aligned}\frac{1}{(1 + \varepsilon) \cdot N_R(x)} &= (1 - \delta) \cdot \frac{1}{N_R(x)} \leq \Pr(\mathcal{G}(x, \varepsilon) = y) \leq \\ &(1 + \delta) \cdot \frac{1}{N_R(x)} \leq (1 + \varepsilon) \cdot \frac{1}{N_R(x)}\end{aligned}$$

# Un esquema de generación casi uniforme

Desde ahora en adelante consideramos la primera definición de generador casi uniforme.

# Un esquema de generación casi uniforme

Desde ahora en adelante consideramos la primera definición de generador casi uniforme.

## Definición

Dada una  $p$ -relación  $R \subseteq \Sigma^* \times \Sigma^*$ , un algoritmo aleatorizado  $\mathcal{G} : \Sigma^* \times (0,1) \rightarrow \Sigma^* \cup \{\perp\}$  es un **fully polynomial almost uniform generator (FPAUG)** para  $R$  si

1.  $\mathcal{G}$  es un generador casi uniforme para  $R$
2. Existe un polinomio  $q(u, v)$  tal que para todo  $x \in \Sigma^*$  y  $\varepsilon \in (0, 1)$ , el número de pasos ejecutados por  $\mathcal{G}(x, \varepsilon)$  es menor o igual a  $q(|x|, \frac{1}{\varepsilon})$

# Una definición alternativa de FPRAS

## Definición

Dada una  $p$ -relación  $R \subseteq \Sigma^* \times \Sigma^*$ , un algoritmo aleatorizado  $\mathcal{A} : \Sigma^* \times (0,1) \rightarrow \mathbb{N}$  es un **fully polynomial randomized approximation scheme (FPRAS)** para  $R$  si existe un polinomio  $q(u, v)$  tal que para cada  $x \in \Sigma^*$  y  $\varepsilon \in (0, 1)$ :

1. El número de pasos ejecutados por  $\mathcal{A}(x, \varepsilon)$  es menor o igual a  $q(|x|, \frac{1}{\varepsilon})$
2.  $\Pr\left(\frac{N_R(x)}{(1 + \varepsilon)} \leq \mathcal{A}(x, \varepsilon) \leq (1 + \varepsilon) \cdot N_R(x)\right) \geq \frac{3}{4}$

# Una definición alternativa de FPRAS

## Definición

Dada una  $p$ -relación  $R \subseteq \Sigma^* \times \Sigma^*$ , un algoritmo aleatorizado  $\mathcal{A} : \Sigma^* \times (0,1) \rightarrow \mathbb{N}$  es un **fully polynomial randomized approximation scheme (FPRAS)** para  $R$  si existe un polinomio  $q(u, v)$  tal que para cada  $x \in \Sigma^*$  y  $\varepsilon \in (0, 1)$ :

1. El número de pasos ejecutados por  $\mathcal{A}(x, \varepsilon)$  es menor o igual a  $q(|x|, \frac{1}{\varepsilon})$
2.  $\Pr\left(\frac{N_R(x)}{(1 + \varepsilon)} \leq \mathcal{A}(x, \varepsilon) \leq (1 + \varepsilon) \cdot N_R(x)\right) \geq \frac{3}{4}$

## Observación

Dado  $f \in \#P$  representado como  $R_f$ , se puede demostrar que esta definición de FPRAS es equivalente a la vista en el capítulo anterior.

- Usando las mismas ideas que en la demostración de que las definiciones de generador casi uniforme son equivalentes

# Un comentario sobre las definiciones anteriores

La noción de algoritmo aleatorizado se formaliza usando MT probabilísticas.

- ▶ Estas máquinas funcionan con cintas de bits, por lo que las probabilidades resultantes son de la forma  $\frac{n}{2^k}$

# Un comentario sobre las definiciones anteriores

La noción de algoritmo aleatorizado se formaliza usando MT probabilísticas.

- ▶ Estas máquinas funcionan con cintas de bits, por lo que las probabilidades resultantes son de la forma  $\frac{n}{2^k}$

Por lo tanto, al describir un algoritmo aleatorizado, en teoría no podemos decir algo como “la probabilidad de error del algoritmo es  $\frac{1}{3}$ ”

# Algunos comentarios sobre las definiciones anteriores

Tratar de tener algoritmos aleatorizados con probabilidades arbitrarias no entrega intuiciones nuevas

- ▶ Y hace mucho más técnicas y complicadas las demostraciones

# Algunos comentarios sobre las definiciones anteriores

Tratar de tener algoritmos aleatorizados con probabilidades arbitrarias no entrega intuiciones nuevas

- Y hace mucho más técnicas y complicadas las demostraciones

## Supuesto

Todas las probabilidades que vamos a considerar (por ejemplo, la probabilidad  $\frac{1}{N_R(x)}$ ) son de la forma  $\frac{n}{2^k}$

# La relación entre FPAUG y FPRAS

Pasaremos ahora a enunciar y demostrar que la existencia de un FPAUG implica la existencia de un FPRAS.

- ▶ Este resultado es válido para una amplia clase de relaciones
- ▶ Esto nos va a permitir utilizar una gran cantidad de herramientas desarrolladas para el muestreo de variables aleatorias en la construcción de FPRAS

# La relación entre FPAUG y FPRAS

Pasaremos ahora a enunciar y demostrar que la existencia de un FPAUG implica la existencia de un FPRAS.

- ▶ Este resultado es válido para una amplia clase de relaciones
- ▶ Esto nos va a permitir utilizar una gran cantidad de herramientas desarrolladas para el muestreo de variables aleatorias en la construcción de FPRAS

Primero debemos formalizar la noción de  $p$ -relación auto-reducible, la cual es necesaria al demostrar la relación entre FPAUG y FPRAS.

# Relaciones auto-reducibles

Intuitivamente, un problema se dice auto-reducible si es que se puede solucionar mediante la resolución de instancias más simples del mismo problema.

# Relaciones auto-reducibles

Intuitivamente, un problema se dice auto-reducible si es que se puede solucionar mediante la resolución de instancias más simples del mismo problema.

## Ejemplo

Sea  $\varphi$  una fórmula proposicional con variables  $x_1, \dots, x_n$

Utilizamos la notación  $\varphi[\frac{x_i}{v}]$  para indicar que la variable  $x_i$  es reemplazada por  $v \in \{0, 1\}$

- Reemplazar por 0 y 1 equivale a reemplazar por una contradicción y una tautología, respectivamente

El problema de determinar si  $\varphi$  es satisfacible se reduce al problema de determinar si  $\varphi[\frac{x_1}{0}]$  o  $\varphi[\frac{x_1}{1}]$  es satisfacible

- Así, una instancia de SAT se reduce a instancias más simples de SAT

# Relaciones auto-reducibles: formalización

## Definición

*Una relación  $R \subseteq \Sigma^* \times \Sigma^*$  es auto-reducible si:*

# Relaciones auto-reducibles: formalización

## Definición

*Una relación  $R \subseteq \Sigma^* \times \Sigma^*$  es auto-reducible si:*

- 1. Existe una función  $g : \Sigma^* \rightarrow \mathbb{N}$  tal que  $g$  es computable en tiempo polinomial y para cada  $(x, y) \in R$  se tiene que  $|y| = g(x)$*

# Relaciones auto-reducibles: formalización

## Definición

Una relación  $R \subseteq \Sigma^* \times \Sigma^*$  es auto-reducible si:

1. Existe una función  $g : \Sigma^* \rightarrow \mathbb{N}$  tal que  $g$  es computable en tiempo polinomial y para cada  $(x, y) \in R$  se tiene que  $|y| = g(x)$
2. Existen funciones  $\psi : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$  y  $\sigma : \Sigma^* \rightarrow \mathbb{N}$  tales que:

# Relaciones auto-reducibles: formalización

## Definición

Una relación  $R \subseteq \Sigma^* \times \Sigma^*$  es auto-reducible si:

1. Existe una función  $g : \Sigma^* \rightarrow \mathbb{N}$  tal que  $g$  es computable en tiempo polinomial y para cada  $(x, y) \in R$  se tiene que  $|y| = g(x)$
2. Existen funciones  $\psi : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$  y  $\sigma : \Sigma^* \rightarrow \mathbb{N}$  tales que:
  - ▶  $\psi$  y  $\sigma$  son computables en tiempo polinomial

# Relaciones auto-reducibles: formalización

## Definición

Una relación  $R \subseteq \Sigma^* \times \Sigma^*$  es auto-reducible si:

1. Existe una función  $g : \Sigma^* \rightarrow \mathbb{N}$  tal que  $g$  es computable en tiempo polinomial y para cada  $(x, y) \in R$  se tiene que  $|y| = g(x)$
2. Existen funciones  $\psi : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$  y  $\sigma : \Sigma^* \rightarrow \mathbb{N}$  tales que:
  - ▶  $\psi$  y  $\sigma$  son computables en tiempo polinomial
  - ▶  $\sigma(x) \in O(\log(|x|))$

# Relaciones auto-reducibles: formalización

## Definición

Una relación  $R \subseteq \Sigma^* \times \Sigma^*$  es auto-reducible si:

1. Existe una función  $g : \Sigma^* \rightarrow \mathbb{N}$  tal que  $g$  es computable en tiempo polinomial y para cada  $(x, y) \in R$  se tiene que  $|y| = g(x)$
2. Existen funciones  $\psi : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$  y  $\sigma : \Sigma^* \rightarrow \mathbb{N}$  tales que:
  - ▶  $\psi$  y  $\sigma$  son computables en tiempo polinomial
  - ▶  $\sigma(x) \in O(\log(|x|))$
  - ▶  $\forall x \in \Sigma^* : \text{si } g(x) > 0, \text{ entonces } 0 < \sigma(x) \leq g(x)$

# Relaciones auto-reducibles: formalización

## Definición

Una relación  $R \subseteq \Sigma^* \times \Sigma^*$  es auto-reducible si:

1. Existe una función  $g : \Sigma^* \rightarrow \mathbb{N}$  tal que  $g$  es computable en tiempo polinomial y para cada  $(x, y) \in R$  se tiene que  $|y| = g(x)$
2. Existen funciones  $\psi : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$  y  $\sigma : \Sigma^* \rightarrow \mathbb{N}$  tales que:
  - ▶  $\psi$  y  $\sigma$  son computables en tiempo polinomial
  - ▶  $\sigma(x) \in O(\log(|x|))$
  - ▶  $\forall x \in \Sigma^* : \text{si } g(x) > 0, \text{ entonces } 0 < \sigma(x) \leq g(x)$
  - ▶  $\forall x, w \in \Sigma^* : |\psi(x, w)| \leq |x|$

# Relaciones auto-reducibles: formalización

## Definición

Una relación  $R \subseteq \Sigma^* \times \Sigma^*$  es auto-reducible si:

1. Existe una función  $g : \Sigma^* \rightarrow \mathbb{N}$  tal que  $g$  es computable en tiempo polinomial y para cada  $(x, y) \in R$  se tiene que  $|y| = g(x)$
2. Existen funciones  $\psi : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$  y  $\sigma : \Sigma^* \rightarrow \mathbb{N}$  tales que:
  - ▶  $\psi$  y  $\sigma$  son computables en tiempo polinomial
  - ▶  $\sigma(x) \in O(\log(|x|))$
  - ▶  $\forall x \in \Sigma^* : \text{si } g(x) > 0, \text{ entonces } 0 < \sigma(x) \leq g(x)$
  - ▶  $\forall x, w \in \Sigma^* : |\psi(x, w)| \leq |x|$
  - ▶  $\forall x, y \in \Sigma^* \text{ con } y = a_1 \cdots a_n :$   
 $(x, y) \in R \text{ si y sólo si } (\psi(x, a_1 \cdots a_{\sigma(x)}), a_{\sigma(x)+1} \cdots a_n) \in R$

# Relaciones auto-reducibles: ejemplos

## Ejercicios

Demuestre que las siguientes relaciones son auto-reducibles:

1.  $R_{\text{SAT}} = \{(\varphi, \sigma) \mid \varphi \text{ es una fórmula proposicional y } \sigma \text{ es una valuación tal que } \sigma(\varphi) = 1\}$
2.  $R_{\text{IS}} = \{(G, S) \mid G \text{ es un grafo y } S \text{ es un conjunto independiente de } G\}$

# Una relación natural no auto-reducible

Dado un grafo  $G = (N, A)$ , decimos que  $S \subseteq N$  es un conjunto independiente **maximal** de  $G$  si:

1.  $S$  es un conjunto independiente de  $G$
2. Para todo conjunto independiente  $S'$  de  $G$ , no se cumple que  $S \subsetneq S'$

# Una relación natural no auto-reducible

Dado un grafo  $G = (N, A)$ , decimos que  $S \subseteq N$  es un conjunto independiente **maximal** de  $G$  si:

1.  $S$  es un conjunto independiente de  $G$
2. Para todo conjunto independiente  $S'$  de  $G$ , no se cumple que  $S \subsetneq S'$

Considere la relación:

$$R_{\text{MIS}} = \{(G, S) \mid G \text{ es un grafo y } S \text{ es un conjunto independiente maximal de } G\}$$

# Una relación natural no auto-reducible

Dado un grafo  $G = (N, A)$ , decimos que  $S \subseteq N$  es un conjunto independiente **maximal** de  $G$  si:

1.  $S$  es un conjunto independiente de  $G$
2. Para todo conjunto independiente  $S'$  de  $G$ , no se cumple que  $S \subsetneq S'$

Considere la relación:

$$R_{\text{MIS}} = \{(G, S) \mid G \text{ es un grafo y } S \text{ es un conjunto independiente maximal de } G\}$$

¿Es  $R_{\text{MIS}}$  auto-reducible?

# Una relación natural no auto-reducible

Dado un grafo  $G = (N, A)$ , decimos que  $S \subseteq N$  es un conjunto independiente **maximal** de  $G$  si:

1.  $S$  es un conjunto independiente de  $G$
2. Para todo conjunto independiente  $S'$  de  $G$ , no se cumple que  $S \subsetneq S'$

Considere la relación:

$$R_{\text{MIS}} = \{(G, S) \mid G \text{ es un grafo y } S \text{ es un conjunto independiente maximal de } G\}$$

¿Es  $R_{\text{MIS}}$  auto-reducible?

- ¿Cómo se puede demostrar que no lo es?

# Una propiedad de las relaciones auto-reducibles

Para demostrar que  $R_{\text{MIS}}$  no es auto-reducible identificamos una propiedad de las relaciones auto-reducibles que no es cumplida por  $R_{\text{MIS}}$

- ▶ Bajo una suposición de complejidad

# Una propiedad de las relaciones auto-reducibles

Para demostrar que  $R_{\text{MIS}}$  no es auto-reducible identificamos una propiedad de las relaciones auto-reducibles que no es cumplida por  $R_{\text{MIS}}$

- ▶ Bajo una suposición de complejidad

Dado un alfabeto  $\Sigma$ , suponga dado un orden lineal en  $\Sigma$

- ▶ Este orden lineal induce un orden lexicográfico  $\leq$  en  $\Sigma^*$

# Una propiedad de las relaciones auto-reducibles

Para demostrar que  $R_{\text{MIS}}$  no es auto-reducible identificamos una propiedad de las relaciones auto-reducibles que no es cumplida por  $R_{\text{MIS}}$

- Bajo una suposición de complejidad

Dado un alfabeto  $\Sigma$ , suponga dado un orden lineal en  $\Sigma$

- Este orden lineal induce un orden lexicográfico  $\leq$  en  $\Sigma^*$

## Definición

Dada una relación  $R \subseteq \Sigma^* \times \Sigma^*$ :

$$\text{Exists}(R) = \{x \mid \exists y : (x, y) \in R\}$$

$$\text{Min}(R) = \{(x, y) \mid x \in \text{Exists}(R) \wedge y = \arg \min_{\leq} \{z \mid (x, z) \in R\}\}$$

$$\text{Max}(R) = \{(x, y) \mid x \in \text{Exists}(R) \wedge y = \arg \max_{\leq} \{z \mid (x, z) \in R\}\}$$

# Una propiedad de las relaciones auto-reducibles

## Teorema

*Si  $R$  es una  $p$ -relación auto-reducible tal que  $\text{Exists}(R) \in PTIME$ , entonces  $\text{Min}(R) \in PTIME$  y  $\text{Max}(R) \in PTIME$*

# Una propiedad de las relaciones auto-reducibles

## Teorema

*Si  $R$  es una  $p$ -relación auto-reducible tal que  $\text{Exists}(R) \in PTIME$ , entonces  $\text{Min}(R) \in PTIME$  y  $\text{Max}(R) \in PTIME$*

## Ejercicio

Demuestre el teorema

# $R_{\text{MIS}}$ no es auto-reducible

## Proposición

Si  $R_{\text{MIS}}$  es auto-reducible, entonces  $\text{PTIME} = \text{NP}$

# $R_{\text{MIS}}$ no es auto-reducible

## Proposición

Si  $R_{\text{MIS}}$  es auto-reducible, entonces  $\text{PTIME} = \text{NP}$

## Ejercicio

Demuestre las siguientes propiedades:

1.  $R_{\text{MIS}}$  es una  $p$ -relación y  $\text{Exists}(R_{\text{MIS}}) \in \text{PTIME}$
2.  $\text{Min}(R_{\text{MIS}})$  es co-NP-completo

# $R_{\text{MIS}}$ no es auto-reducible

## Proposición

Si  $R_{\text{MIS}}$  es auto-reducible, entonces  $\text{PTIME} = \text{NP}$

## Ejercicio

Demuestre las siguientes propiedades:

1.  $R_{\text{MIS}}$  es una  $p$ -relación y  $\text{Exists}(R_{\text{MIS}}) \in \text{PTIME}$
2.  $\text{Min}(R_{\text{MIS}})$  es co-NP-completo

A partir de estas propiedades y del teorema demuestre la proposición.

# Solucionando el ejercicio: $\overline{\text{Min}(R_{\text{MIS}})}$ es NP-hard

Vamos a mostrar que  $\text{CNF-SAT} \leq_m^p \overline{\text{Min}(R_{\text{MIS}})}$

# Solucionando el ejercicio: $\overline{\text{Min}(R_{\text{MIS}})}$ es NP-hard

Vamos a mostrar que  $\text{CNF-SAT} \leq_m^p \overline{\text{Min}(R_{\text{MIS}})}$

- Dada una formula proposicional  $\varphi$  en CNF, mostramos como construir en tiempo polinomial un grafo  $G = (N, A)$  y un conjunto  $S \subseteq N$  tales que:

$$\varphi \text{ es satisfacible} \quad \text{si y sólo si} \quad (G, S) \in \overline{\text{Min}(R_{\text{MIS}})}$$

# Solucionando el ejercicio: $\overline{\text{Min}(R_{\text{MIS}})}$ es NP-hard

Vamos a mostrar que  $\text{CNF-SAT} \leq_m^p \overline{\text{Min}(R_{\text{MIS}})}$

- Dada una formula proposicional  $\varphi$  en CNF, mostramos como construir en tiempo polinomial un grafo  $G = (N, A)$  y un conjunto  $S \subseteq N$  tales que:

$$\varphi \text{ es satisfacible} \quad \text{si y sólo si} \quad (G, S) \in \overline{\text{Min}(R_{\text{MIS}})}$$

Debemos representar  $S$  como un string sobre un alfabeto  $\Sigma$  fijo.

# Solucionando el ejercicio: $\overline{\text{Min}(R_{\text{MIS}})}$ es NP-hard

Vamos a mostrar que  $\text{CNF-SAT} \leq_m^p \overline{\text{Min}(R_{\text{MIS}})}$

- Dada una formula proposicional  $\varphi$  en CNF, mostramos como construir en tiempo polinomial un grafo  $G = (N, A)$  y un conjunto  $S \subseteq N$  tales que:

$$\varphi \text{ es satisfacible} \quad \text{si y sólo si} \quad (G, S) \in \overline{\text{Min}(R_{\text{MIS}})}$$

Debemos representar  $S$  como un string sobre un alfabeto  $\Sigma$  fijo.

- Esto es necesario porque debemos tener un orden lexicográfico sobre los conjuntos independientes maximales de  $G$

# Solucionando el ejercicio: $\overline{\text{Min}(R_{\text{MIS}})}$ es NP-hard

Vamos a mostrar que  $\text{CNF-SAT} \leq_m^p \overline{\text{Min}(R_{\text{MIS}})}$

- ▶ Dada una formula proposicional  $\varphi$  en CNF, mostramos como construir en tiempo polinomial un grafo  $G = (N, A)$  y un conjunto  $S \subseteq N$  tales que:

$$\varphi \text{ es satisfacible} \quad \text{si y sólo si} \quad (G, S) \in \overline{\text{Min}(R_{\text{MIS}})}$$

Debemos representar  $S$  como un string sobre un alfabeto  $\Sigma$  fijo.

- ▶ Esto es necesario porque debemos tener un orden lexicográfico sobre los conjuntos independientes maximales de  $G$

En estricto rigor también  $G$  debería ser representado como un string sobre  $\Sigma$

- ▶ Aunque esto no es fundamental para la demostración

# Solucionando el ejercicio: $\overline{\text{Min}(R_{\text{MIS}})}$ es NP-hard

Vamos a dar la idea de la demostración con un ejemplo.

- ▶ Dejamos como un ejercicio el generalizar esta idea a cualquier fórmula proposicional en CNF

# Solucionando el ejercicio: $\overline{\text{Min}(R_{\text{MIS}})}$ es NP-hard

Vamos a dar la idea de la demostración con un ejemplo.

- ▶ Dejamos como un ejercicio el generalizar esta idea a cualquier fórmula proposicional en CNF

Suponga que  $\varphi = C_1 \wedge C_2$ , donde  $C_1 = (r \vee t)$  y  $C_2 = (t \vee \neg s \vee \neg t \vee \neg u)$

- ▶ Consideramos  $\Sigma = \{0, 1\}$  en la reducción

# Los nodos del grafo $G$

El conjunto  $N$  de nodos de  $G$  es definido como:

$$N = \{C_1, C_2, r, s, t, u, \neg r, \neg s, \neg t, \neg u, \star\}$$

donde  $\star$  es un símbolo que no es mencionado en  $\varphi$

# Representando un conjunto independiente de $G$

Para representar un conjunto  $S \subseteq N$  usamos un string  $w \in \{0,1\}^*$  de largo 11

- ▶ El primer bit de  $w$  es 1 si  $C_1 \in S$ , y 0 en caso contrario. El segundo bit de  $w$  es 1 si  $C_2 \in S$ , y 0 en caso contrario
- ▶ El tercer bit de  $w$  es 1 si  $\star \in S$ , y 0 en caso contrario
- ▶ Los siguientes bits de  $w$  son contruidos de la misma forma para los nodos  $r, s, t, u, \neg r, \neg s, \neg t, \neg u$

# Representando un conjunto independiente de $G$

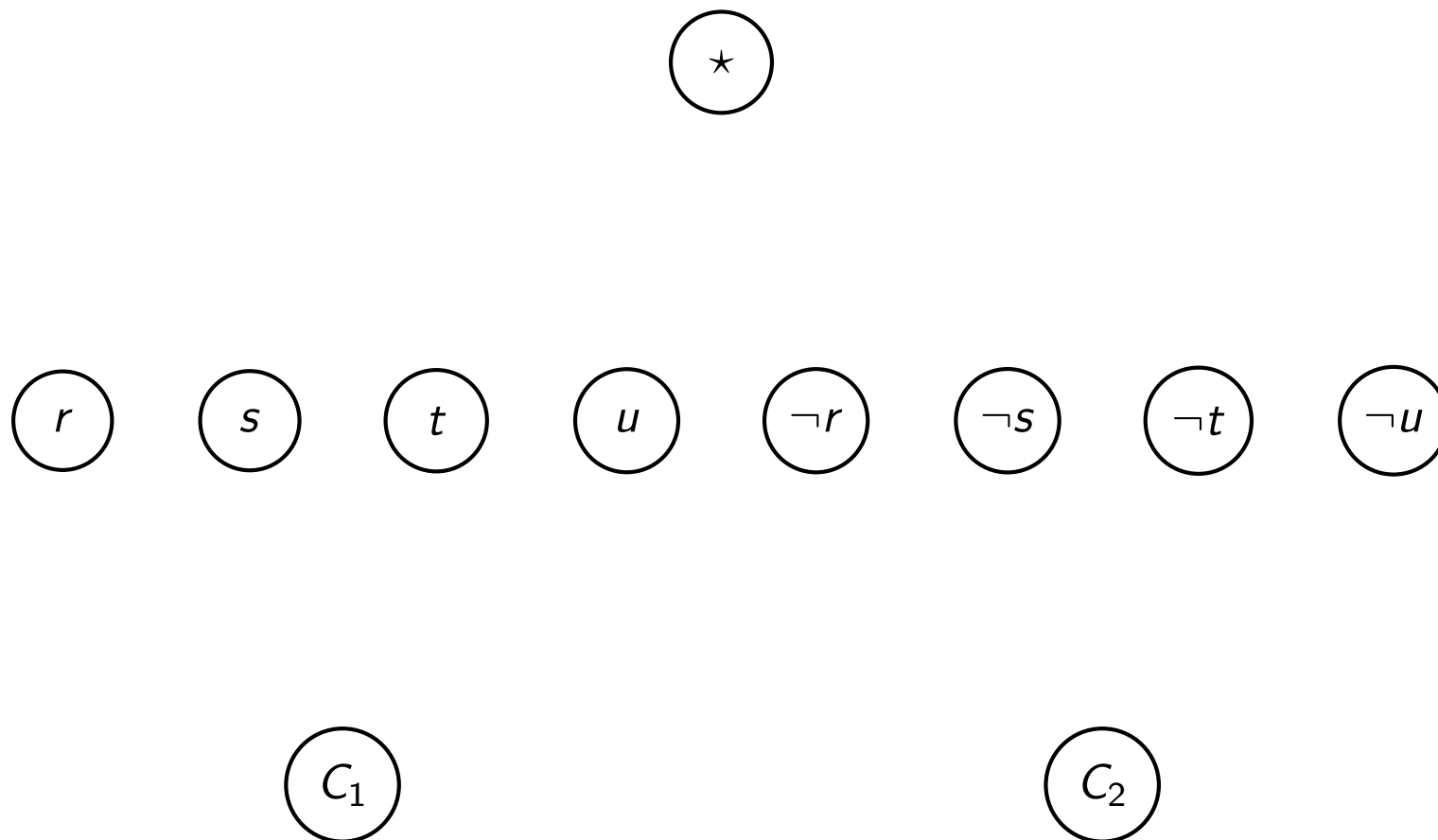
Para representar un conjunto  $S \subseteq N$  usamos un string  $w \in \{0,1\}^*$  de largo 11

- ▶ El primer bit de  $w$  es 1 si  $C_1 \in S$ , y 0 en caso contrario. El segundo bit de  $w$  es 1 si  $C_2 \in S$ , y 0 en caso contrario
- ▶ El tercer bit de  $w$  es 1 si  $\star \in S$ , y 0 en caso contrario
- ▶ Los siguientes bits de  $w$  son contruidos de la misma forma para los nodos  $r, s, t, u, \neg r, \neg s, \neg t, \neg u$

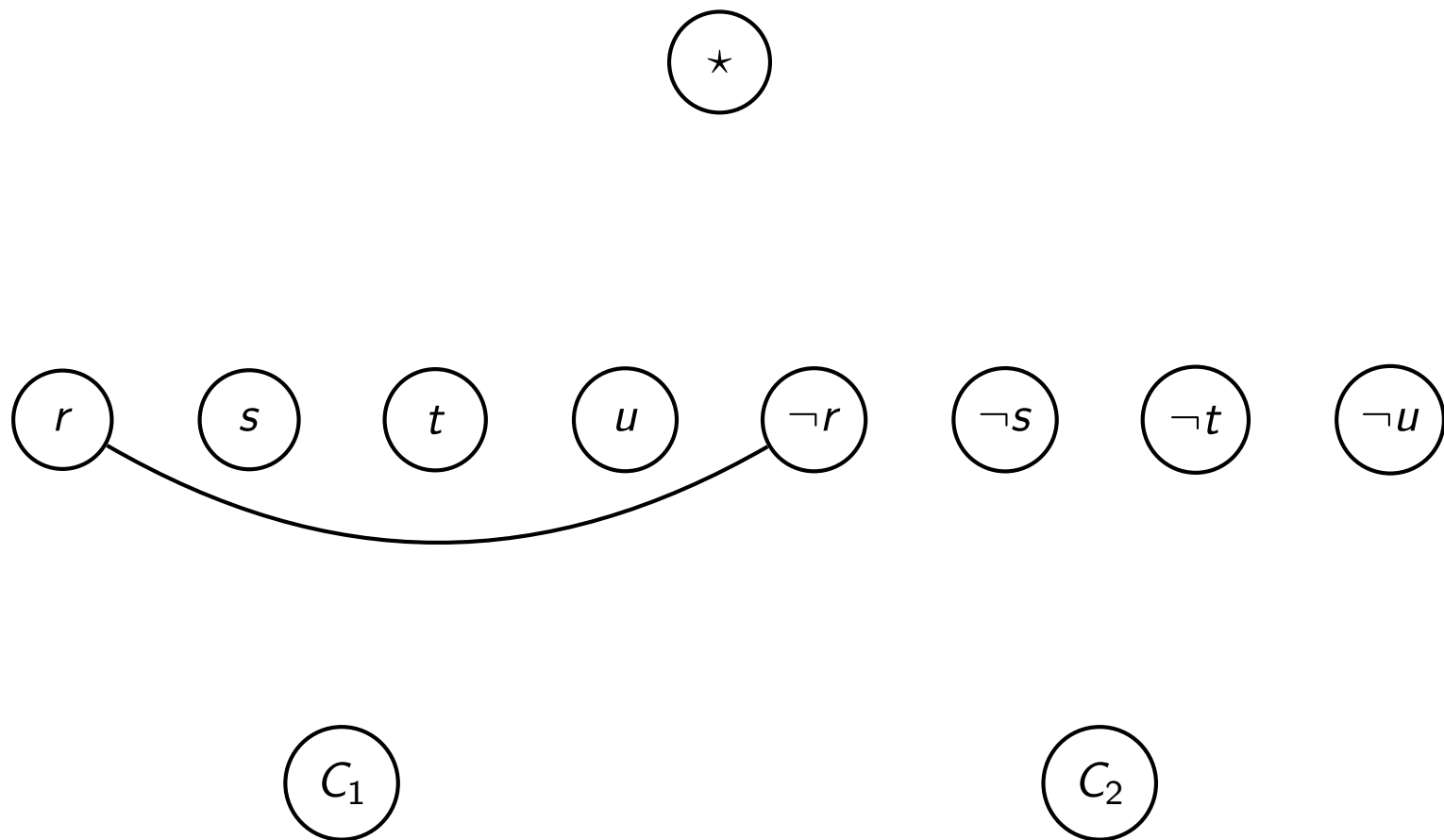
## Ejemplo

El conjunto  $S = \{C_2, \star, u, \neg s, \neg t\}$  es representado por el string 01100010110

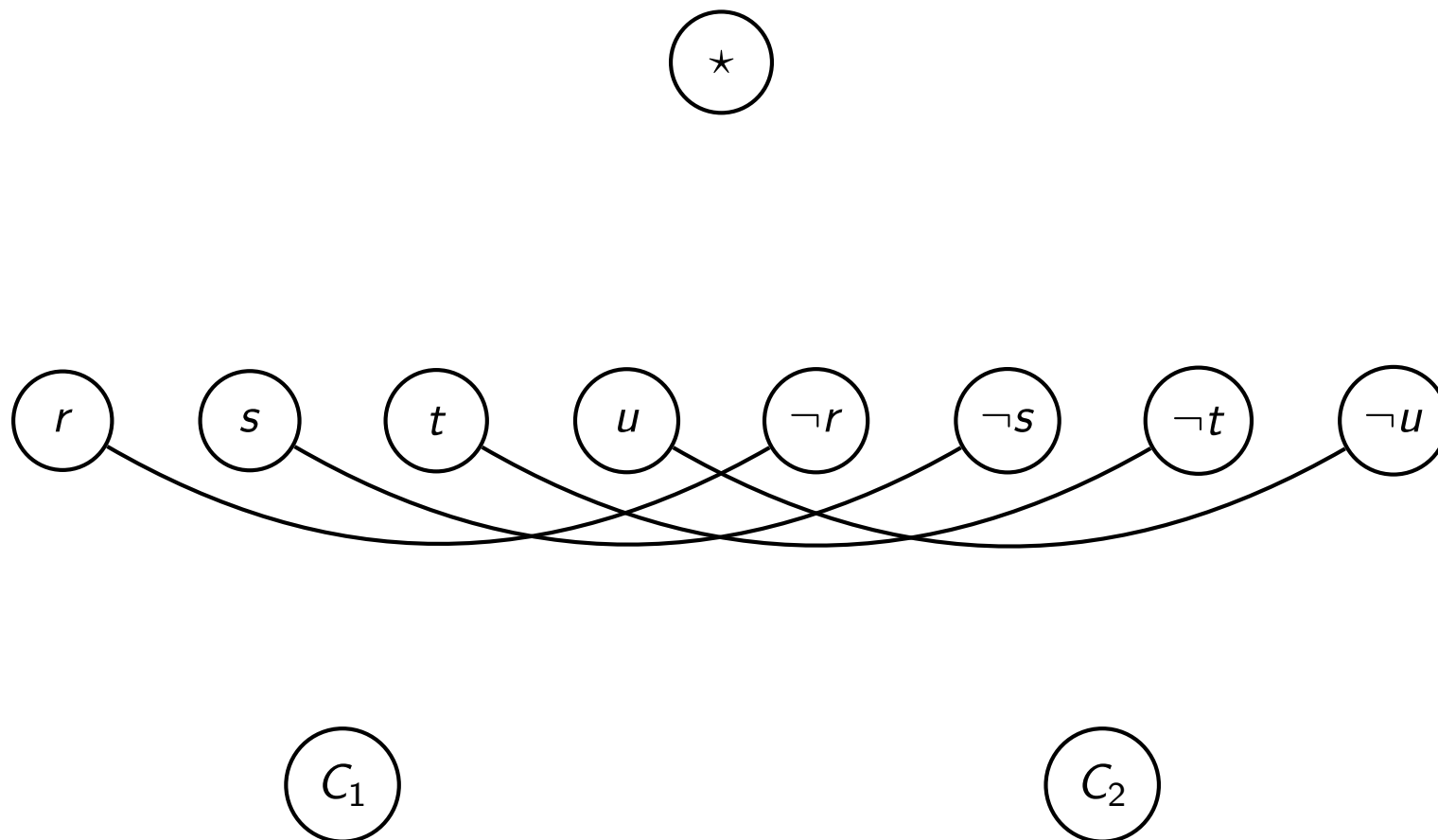
# Los arcos del grafo $G$



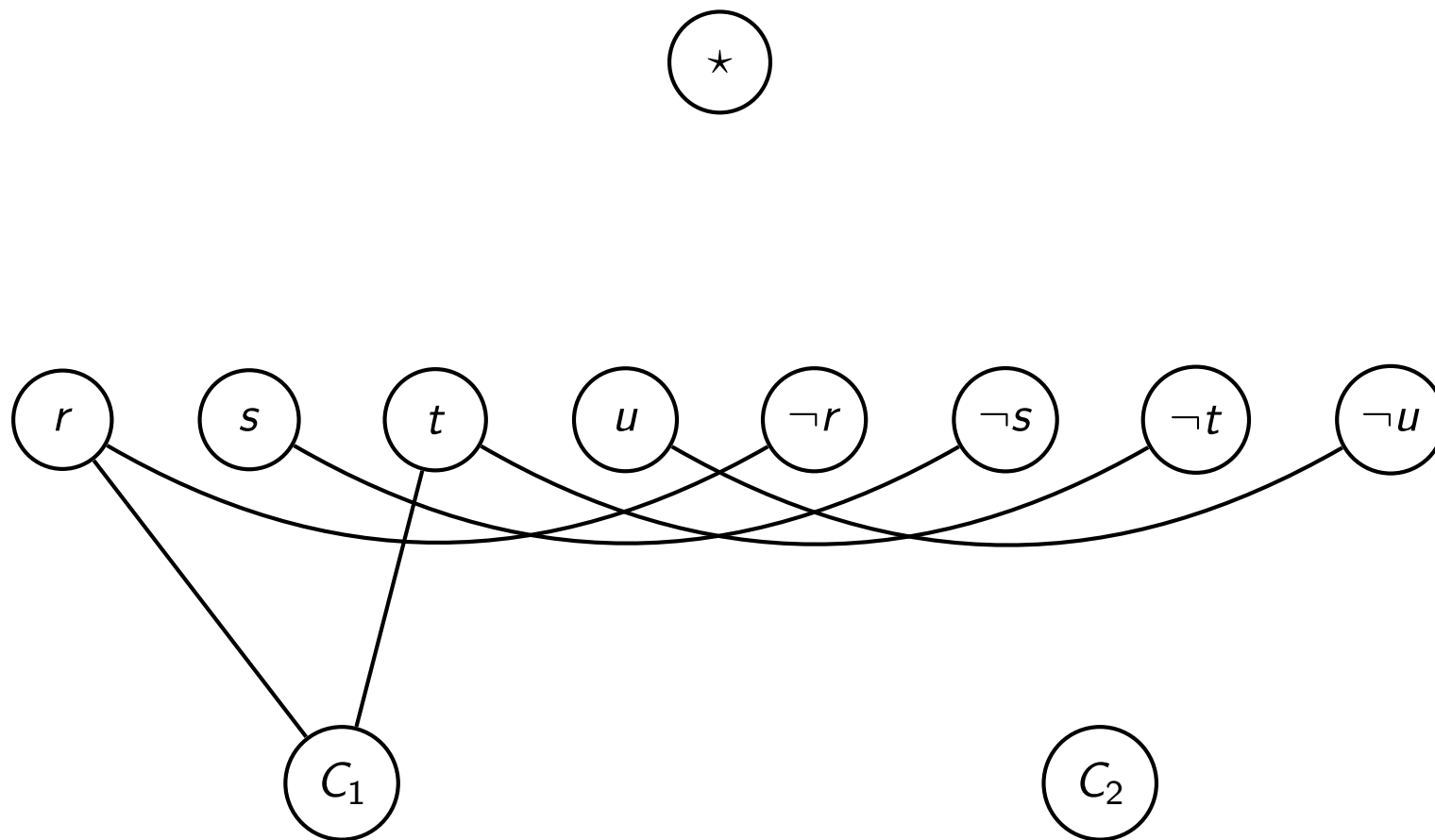
# Los arcos del grafo $G$



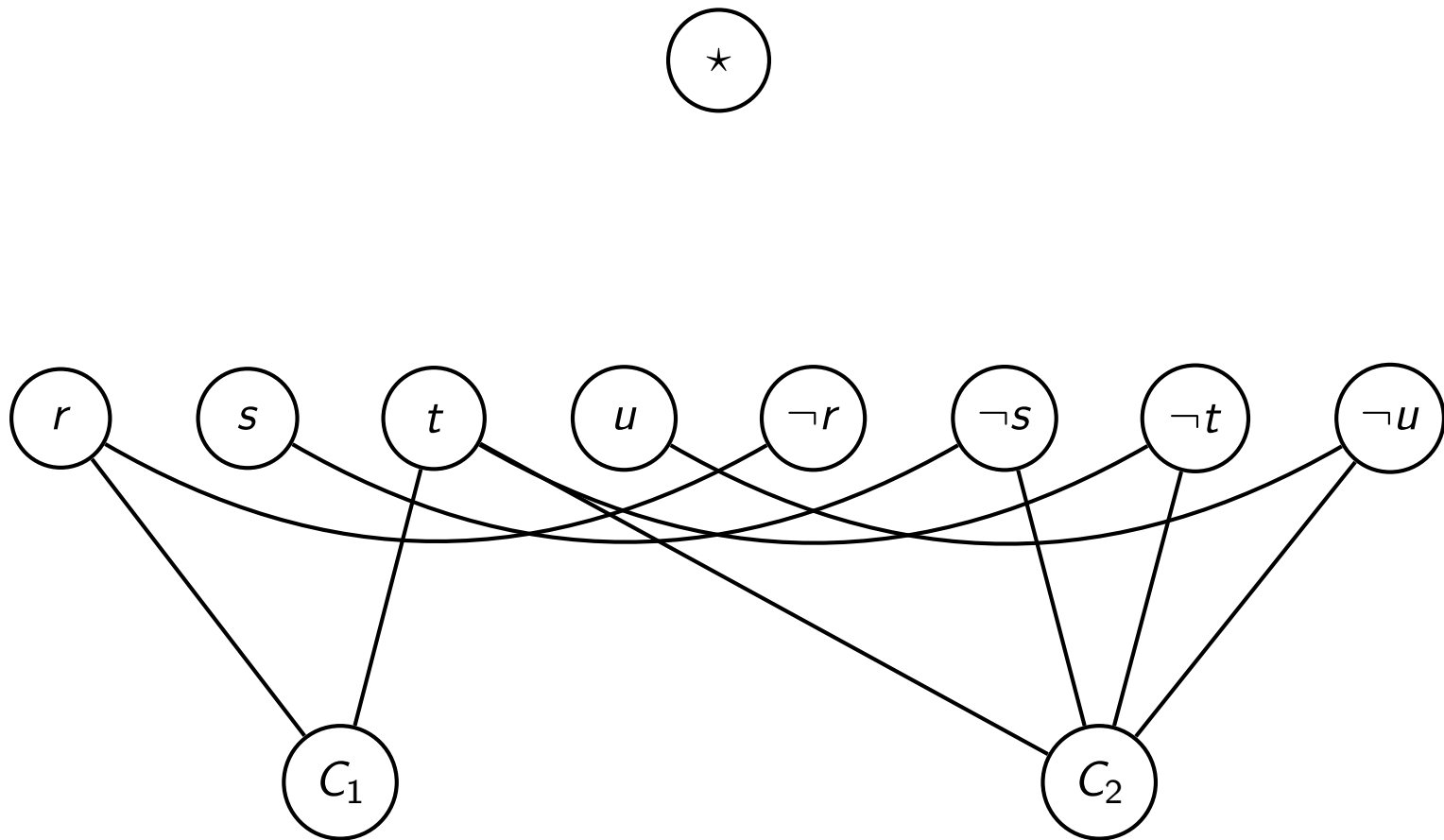
# Los arcos del grafo $G$



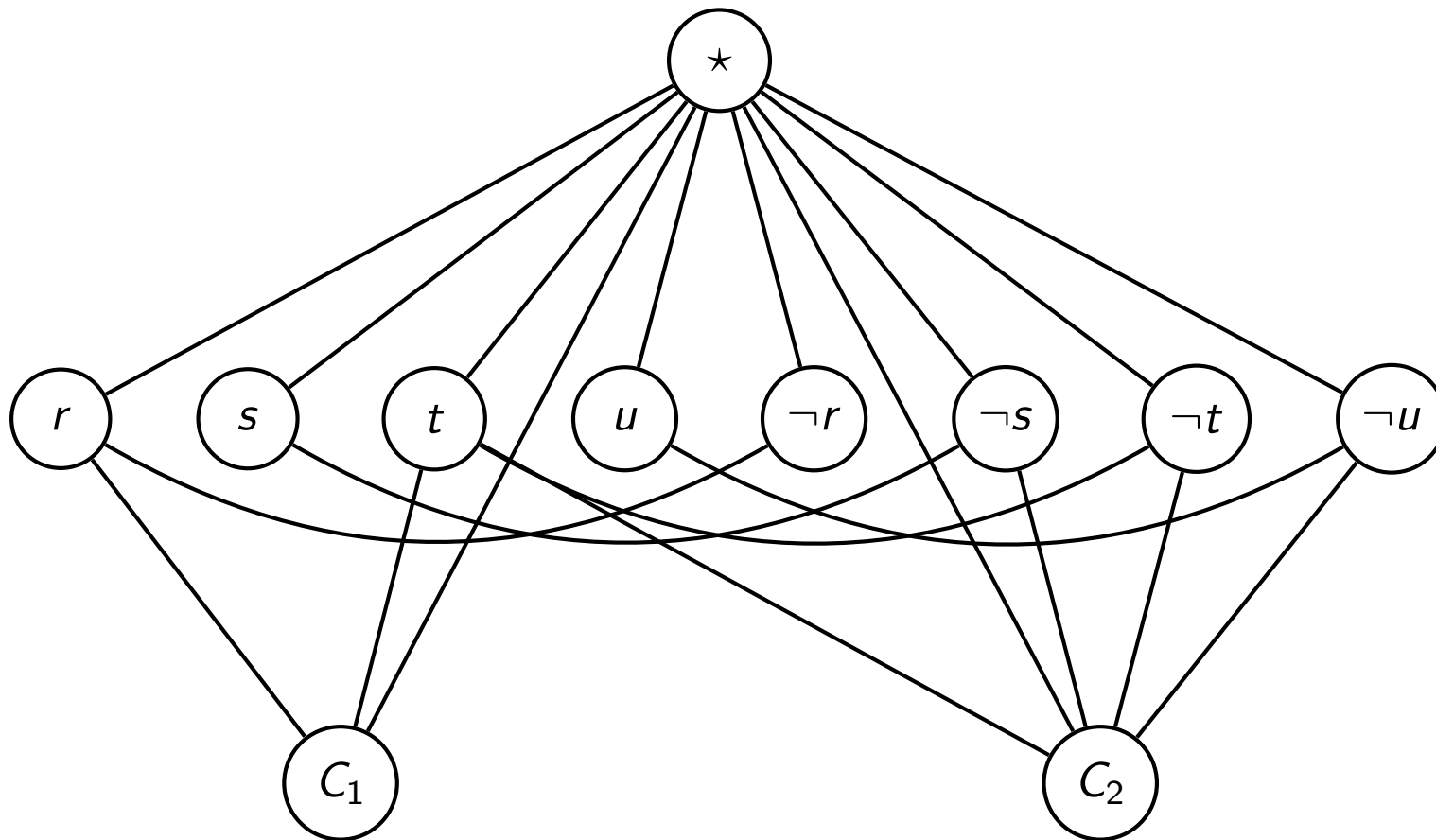
## Los arcos del grafo $G$



## Los arcos del grafo $G$



## Los arcos del grafo $G$



# La equivalencia entre los problemas

$S = \{\star\}$  es un conjunto independiente maximal de  $G$

- Es representado por el string 00100000000

# La equivalencia entre los problemas

$S = \{\star\}$  es un conjunto independiente maximal de  $G$

- Es representado por el string 00100000000

Tenemos que  $\varphi$  es satisfacible si y sólo si  $(G, S) \in \overline{\text{Min}(R_{\text{MIS}})}$

- ¿Por qué se cumple esto en general?

# La equivalencia entre los problemas

$S = \{\star\}$  es un conjunto independiente maximal de  $G$

- Es representado por el string 00100000000

Tenemos que  $\varphi$  es satisfacible si y sólo si  $(G, S) \in \overline{\text{Min}(R_{\text{MIS}})}$

- ¿Por qué se cumple esto en general?

$S' = \{r, \neg s\}$  también es un conjunto independiente maximal de  $G$

# La equivalencia entre los problemas

$S = \{\star\}$  es un conjunto independiente maximal de  $G$

- Es representado por el string 00100000000

Tenemos que  $\varphi$  es satisfacible si y sólo si  $(G, S) \in \overline{\text{Min}(R_{\text{MIS}})}$

- ¿Por qué se cumple esto en general?

$S' = \{r, \neg s\}$  también es un conjunto independiente maximal de  $G$

- $S'$  representa a una valuación  $\sigma$  que satisface  $\varphi$ :  $\sigma(r) = 1$  y  $\sigma(s) = 0$

# La equivalencia entre los problemas

$S = \{\star\}$  es un conjunto independiente maximal de  $G$

- ▶ Es representado por el string 00100000000

Tenemos que  $\varphi$  es satisfacible si y sólo si  $(G, S) \in \overline{\text{Min}(R_{\text{MIS}})}$

- ▶ ¿Por qué se cumple esto en general?

$S' = \{r, \neg s\}$  también es un conjunto independiente maximal de  $G$

- ▶  $S'$  representa a una valuación  $\sigma$  que satisface  $\varphi$ :  $\sigma(r) = 1$  y  $\sigma(s) = 0$
- ▶  $S'$  es representado por el string 00010000100

# La equivalencia entre los problemas

$S = \{\star\}$  es un conjunto independiente maximal de  $G$

- ▶ Es representado por el string 00100000000

Tenemos que  $\varphi$  es satisfacible si y sólo si  $(G, S) \in \overline{\text{Min}(R_{\text{MIS}})}$

- ▶ ¿Por qué se cumple esto en general?

$S' = \{r, \neg s\}$  también es un conjunto independiente maximal de  $G$

- ▶  $S'$  representa a una valuación  $\sigma$  que satisface  $\varphi$ :  $\sigma(r) = 1$  y  $\sigma(s) = 0$
- ▶  $S'$  es representado por el string 00010000100
- ▶ Tenemos que  $(G, S) \in \overline{\text{Min}(R_{\text{MIS}})}$  puesto que 00010000100 es menor que 00100000000 en orden lexicográfico

# Comentarios finales

## Ejercicios

1. Para  $\varphi = (r \vee t) \vee (t \vee \neg s \vee \neg t \vee \neg u)$ , encuentre  $S''$  tal que  $(G, S'') \in \text{Min}(R_{\text{MIS}})$
2. Generalice la construcción mostrada para cualquier fórmula proposicional  $\varphi$  en CNF
  - ▶ Si  $\varphi$  menciona  $m$  cláusulas y  $n$  variables proposicionales, entonces el grafo  $G$  debe tener  $m + 2 \cdot n + 1$  nodos
  - ▶ Como en el ejemplo, se debe tener que  $S = \{\star\}$
3. Para la construcción realizada en 2, demuestre que  $\varphi$  es satisfacible si y sólo si  $(G, S) \in \overline{\text{Min}(R_{\text{MIS}})}$

# Una herramienta fundamental

## Teorema (Jerrum, Valiant & Vazirani)

*Sea  $R$  una  $p$ -relación auto-reducible. Si existe un FPAUG para  $R$ , entonces existe un FPRAS para  $R$ .*