Computação Quântica
Aula 12

Murilo V. G. da Silva

DINF/UFPR

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0,1\}^n \to \{0,1\}^n$ com $s \in \{0,1\}^n$ tal que se $x \neq y$ e $f(x) = f(y)$, então $x \oplus s = y$ (i.e., $f(x \oplus s) = f(x)$).

Saida: Retornar a string s (obs: assumimos $s \neq 0^n$)

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0,1\}^n \to \{0,1\}^n$ com $s \in \{0,1\}^n$ tal que se $x \neq y$ e $f(x) = f(y)$, então $x \oplus s = y$ (i.e., $f(x \oplus s) = f(x)$).

Saida: Retornar a string s (obs: assumimos $s \neq 0^n$)

Passos do Algoritmo

- Para algum r aleatório, criar o estado $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$
  (Aula passada)
- A partir do estado acima, obter string y aleatória tal que $y \cdot s = 0 \pmod 2$
  (Aula passada)
- Repetir isso para obter $n-1$ strings y com tal propriedade
  (Aula de hoje)

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0,1\}^n \to \{0,1\}^n$ com $s \in \{0,1\}^n$ tal que se $x \neq y$ e $f(x) = f(y)$, então $x \oplus s = y$ (i.e., $f(x \oplus s) = f(x)$).

Saida: Retornar a string s (obs: assumimos $s \neq 0^n$)

Passos do Algoritmo

- Para algum r aleatório, criar o estado $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$
  (Aula passada)
- A partir do estado acima, obter string y aleatória tal que $y \cdot s = 0 \pmod 2$
  (Aula passada)
- Repetir isso para obter $n-1$ strings y com tal propriedade
  (Aula de hoje)

Ideia central do algoritmo:

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0,1\}^n \to \{0,1\}^n$ com $s \in \{0,1\}^n$ tal que se $x \neq y$ e $f(x) = f(y)$, então $x \oplus s = y$ (i.e., $f(x \oplus s) = f(x)$).

Saida: Retornar a string s (obs: assumimos $s \neq 0^n$)

Passos do Algoritmo

- Para algum r aleatório, criar o estado $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$
  (Aula passada)
- A partir do estado acima, obter string y aleatória tal que $y \cdot s = 0$ (MOD 2)
  (Aula passada)
- Repetir isso para obter $n-1$ strings y com tal propriedade
  (Aula de hoje)

Ideia central do algoritmo:

- cada string y nos dá a equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$ (MOD 2)
  (incógnitas $s_1, ..., s_n$ e coeficientes $y_1, ..., y_n$)

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0,1\}^n \to \{0,1\}^n$ com $s \in \{0,1\}^n$ tal que se $x \neq y$ e $f(x) = f(y)$, então $x \oplus s = y$ (i.e., $f(x \oplus s) = f(x)$).

Saida: Retornar a string s (obs: assumimos $s \neq 0^n$)

Passos do Algoritmo

- Para algum r aleatório, criar o estado $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$
  (Aula passada)
- A partir do estado acima, obter string y aleatória tal que $y \cdot s = 0$ (MOD 2)
  (Aula passada)
- Repetir isso para obter $n-1$ strings y com tal propriedade
  (Aula de hoje)

Ideia central do algoritmo:

- cada string y nos dá a equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$ (MOD 2)
  (incógnitas $s_1, ..., s_n$ e coeficientes $y_1, ..., y_n$)
- Se obtivermos $n-1$ equações, obtemos os valores $s_1, ..., s_n$ (ou seja os bits de s)

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0,1\}^n \to \{0,1\}^n$ com $s \in \{0,1\}^n$ tal que se $x \neq y$ e $f(x) = f(y)$, então $x \oplus s = y$ (i.e., $f(x \oplus s) = f(x)$).

Saida: Retornar a string s (obs: assumimos $s \neq 0^n$)

Passos do Algoritmo

- Para algum r aleatório, criar o estado $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$
  (Aula passada)
- A partir do estado acima, obter string y aleatória tal que $y \cdot s = 0 \pmod 2$
  (Aula passada)
- Repetir isso para obter $n-1$ strings y com tal propriedade
  (Aula de hoje)

Ideia central do algoritmo:

- cada string y nos dá a equação $y_1 s_1 + y_2 s_2 + \ldots + y_n s_n = 0 \pmod 2$
  (incógnitas $s_1, \ldots, s_n$ e coeficientes $y_1, \ldots, y_n$)
- Se obtivermos $n-1$ equações, obtemos os valores $s_1, \ldots, s_n$ (ou seja os bits de s)

Nota: como estamos trabalhando $(\mathrm{MOD}\ 2)$, há duas soluções: a solução trivial 0^n e a solução s, cujos bits são das variáveis $s_1, \ldots, s_n$ e que pode ser obtida usando eliminação gaussiana módulo 2 que roda em tempo $\mathcal{O}(n^3)$

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0,1\}^n \to \{0,1\}^n$ com $s \in \{0,1\}^n$ tal que se $x \neq y$ e $f(x) = f(y)$, então $x \oplus s = y$ (i.e., $f(x \oplus s) = f(x)$).

Saida: Retornar a string s (obs: assumimos $s \neq 0^n$)

Passos do Algoritmo

- Para algum r aleatório, criar o estado $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$
  (Aula passada)
- A partir do estado acima, obter string y aleatória tal que $y \cdot s = 0 \pmod 2$
  (Aula passada)
- Repetir isso para obter $n-1$ strings y com tal propriedade
  (Aula de hoje)

Ideia central do algoritmo:

- cada string y nos dá a equação $y_1 s_1 + y_2 s_2 + \ldots + y_n s_n = 0 \pmod 2$
  (incógnitas $s_1, \ldots, s_n$ e coeficientes $y_1, \ldots, y_n$)
- Se obtivermos $n-1$ equações, obtemos os valores $s_1, \ldots, s_n$ (ou seja os bits de s)

Nota: como estamos trabalhando $(\mathrm{MOD}\ 2)$, há duas soluções: a solução trivial 0^n e a solução s, cujos bits são das variáveis $s_1, \ldots, s_n$ e que pode ser obtida usando eliminação gaussiana módulo 2 que roda em tempo $\mathcal{O}(n^3)$

- Cuidado extra: as equações obtidas devem ser linearmente independentes

O Algoritmo de Simon

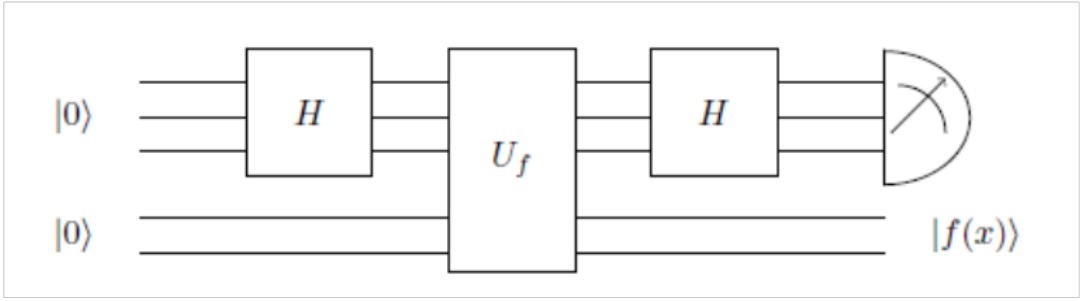
Relembrando como obtivemos a string aleatória y com a propriedade

$$y \cdot s = 0 \pmod 2:$$

O Algoritmo de Simon

Relembrando como obtivemos a string aleatória y com a propriedade

$$y \cdot s = 0 \pmod 2:$$

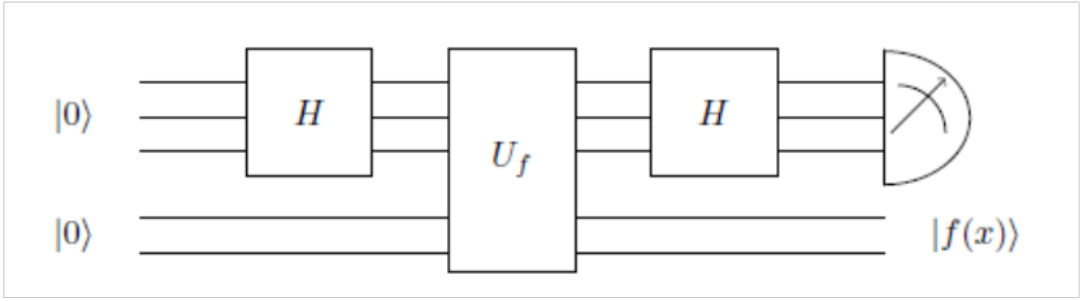


- Primeiro medimos o segundo registrador (superposição de valores $f(x)$)

O Algoritmo de Simon

Relembrando como obtivemos a string aleatória y com a propriedade

$$y \cdot s = 0 \pmod 2:$$

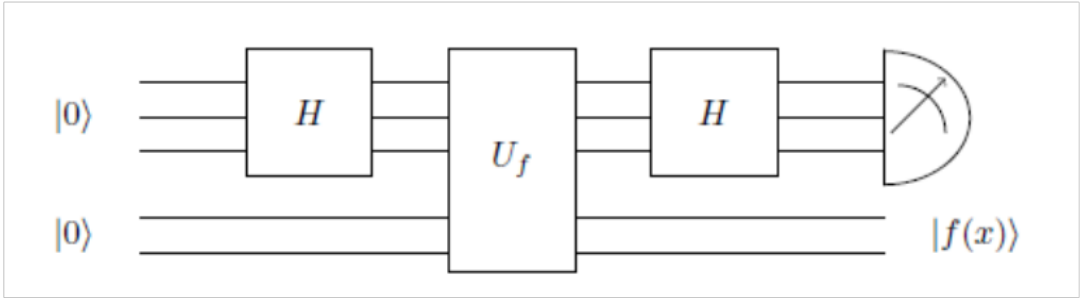


- Primeiro medimos o segundo registrador (superposição de valores $f(x)$)
- Obtivemos no primeiro registrador os valores x consistentes com $f(x)$:

O Algoritmo de Simon

Relembrando como obtivemos a string aleatória y com a propriedade

$$y \cdot s = 0 \pmod 2:$$

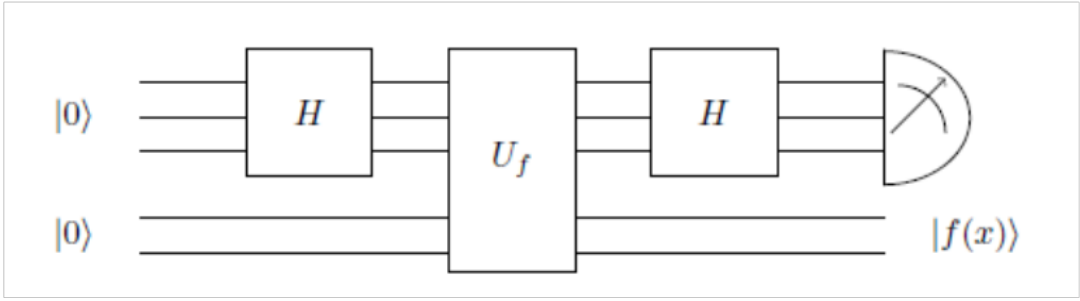


- Primeiro medimos o segundo registrador (superposição de valores $f(x)$)
- Obtivemos no primeiro registrador os valores x consistentes com $f(x)$: a superposição $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$, para um r aleatório

O Algoritmo de Simon

Relembrando como obtivemos a string aleatória y com a propriedade

$$y \cdot s = 0 \pmod 2:$$

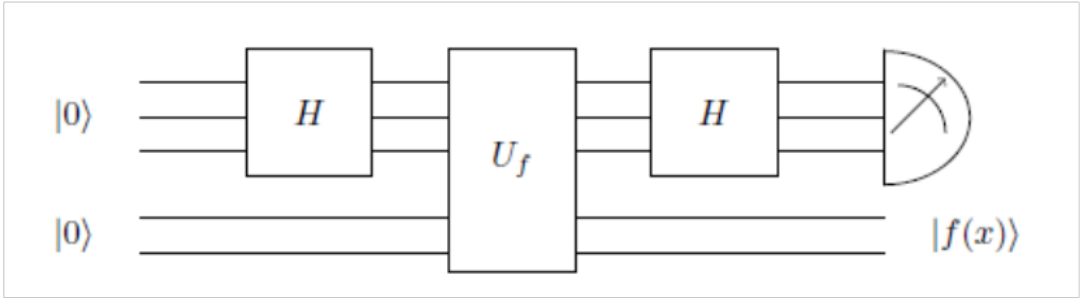


- Primeiro medimos o segundo registrador (superposição de valores $f(x)$)
- Obtivemos no primeiro registrador os valores x consistentes com $f(x)$: a superposição $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$, para um r aleatório
- Fizemos uma amostragem de Fourier em $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$

O Algoritmo de Simon

Relembrando como obtivemos a string aleatória y com a propriedade

$$y \cdot s = 0 \pmod 2:$$

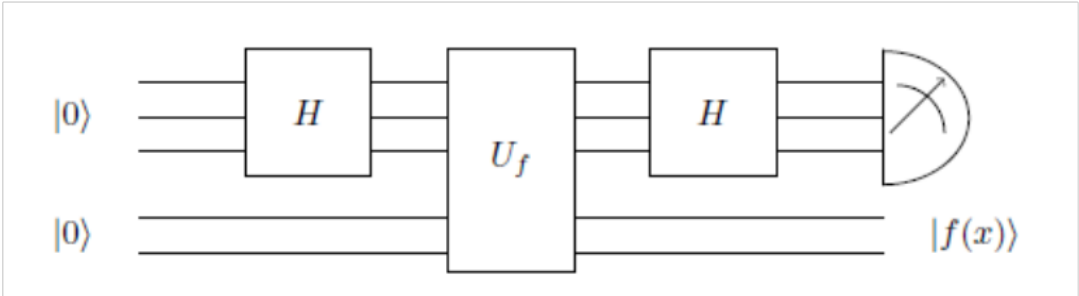


- Primeiro medimos o segundo registrador (superposição de valores $f(x)$)
- Obtivemos no primeiro registrador os valores x consistentes com $f(x)$: a superposição $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$, para um r aleatório
- Fizemos uma amostragem de Fourier em $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$
- Resultado: uma superposição de estados $|y\rangle$, onde cada $|y\rangle$ tem amplitude β_y

O Algoritmo de Simon

Relembrando como obtivemos a string aleatória y com a propriedade

$$y \cdot s = 0 \pmod 2:$$

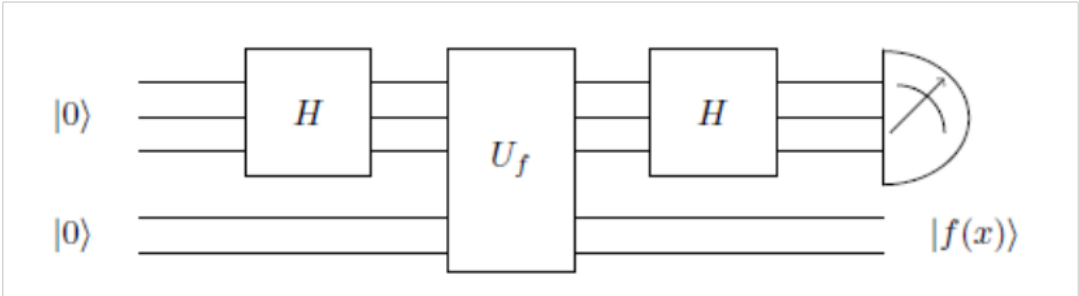


- Primeiro medimos o segundo registrador (superposição de valores $f(x)$)
- Obtivemos no primeiro registrador os valores x consistentes com $f(x)$: a superposição $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$, para um r aleatório
- Fizemos uma amostragem de Fourier em $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$
- Resultado: uma superposição de estados $|y\rangle$, onde cada $|y\rangle$ tem amplitude β_y
- Mostramos que os estados $|y\rangle$ indesejados tem amplitude $\beta_y = 0$ (lembrando que os estados indesejados são os que tem $y \cdot s \neq 0 \pmod 2$)

O Algoritmo de Simon

Relembrando como obtivemos a string aleatória y com a propriedade

$$y \cdot s = 0 \pmod 2:$$

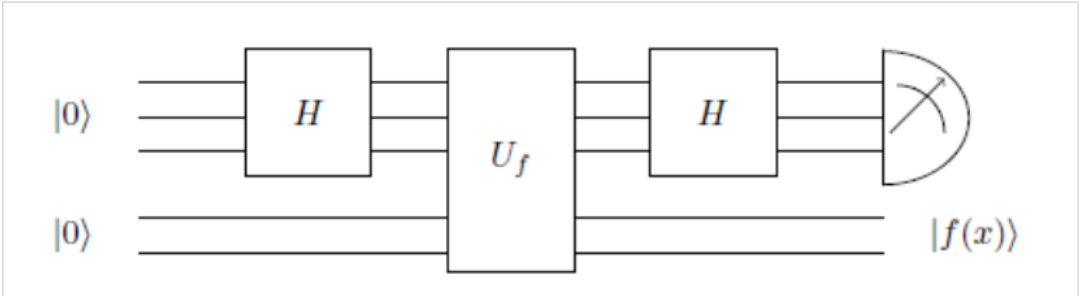


- Primeiro medimos o segundo registrador (superposição de valores $f(x)$)
- Obtivemos no primeiro registrador os valores x consistentes com $f(x)$: a superposição $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$, para um r aleatório
- Fizemos uma amostragem de Fourier em $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$
- Resultado: uma superposição de estados $|y\rangle$, onde cada $|y\rangle$ tem amplitude β_y
- Mostramos que os estados $|y\rangle$ indesejados tem amplitude $\beta_y = 0$ (lembrando que os estados indesejados são os que tem $y \cdot s \neq 0 \pmod 2$)
- Dado um y, temos $\beta_y =$

O Algoritmo de Simon

Relembrando como obtivemos a string aleatória y com a propriedade

$$y \cdot s = 0 \pmod 2:$$

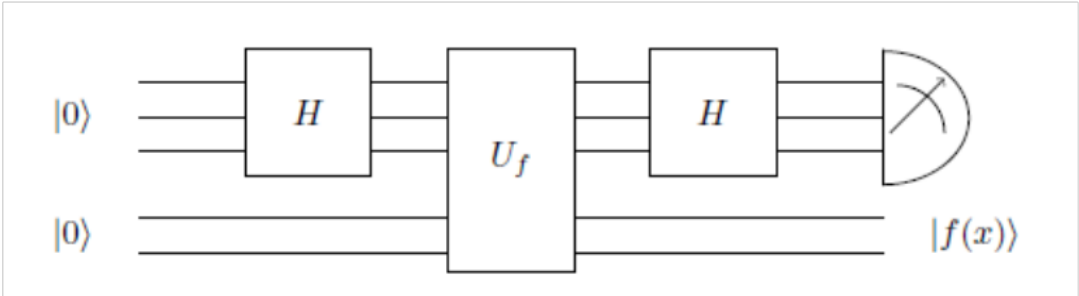


- Primeiro medimos o segundo registrador (superposição de valores $f(x)$)
- Obtivemos no primeiro registrador os valores x consistentes com $f(x)$: a superposição $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$, para um r aleatório
- Fizemos uma amostragem de Fourier em $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$
- Resultado: uma superposição de estados $|y\rangle$, onde cada $|y\rangle$ tem amplitude β_y
- Mostramos que os estados $|y\rangle$ indesejados tem amplitude $\beta_y = 0$ (lembrando que os estados indesejados são os que tem $y \cdot s \neq 0 \pmod 2$)
- Dado um y, temos $\beta_y = \frac{1}{\sqrt{2}}\frac{(-1)^{r\cdot y}}{2^{n/2}} + \frac{1}{\sqrt{2}}\frac{(-1)^{(r\oplus s)\cdot y}}{2^{n/2}} =$

O Algoritmo de Simon

Relembrando como obtivemos a string aleatória y com a propriedade

$$y \cdot s = 0 \pmod 2:$$

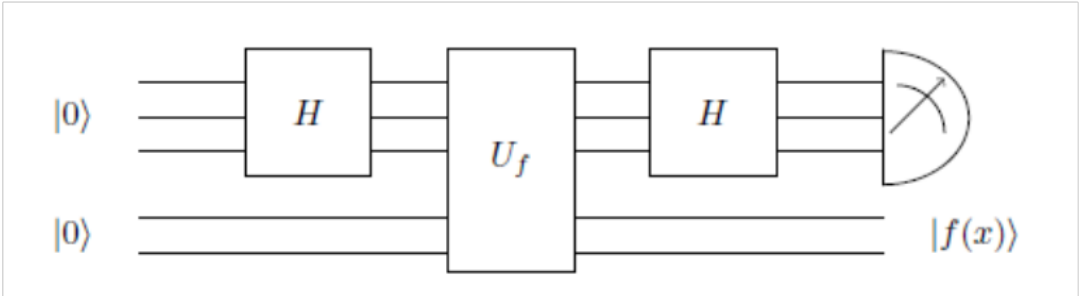


- Primeiro medimos o segundo registrador (superposição de valores $f(x)$)
- Obtivemos no primeiro registrador os valores x consistentes com $f(x)$: a superposição $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$, para um r aleatório
- Fizemos uma amostragem de Fourier em $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$
- Resultado: uma superposição de estados $|y\rangle$, onde cada $|y\rangle$ tem amplitude β_y
- Mostramos que os estados $|y\rangle$ indesejados tem amplitude $\beta_y = 0$ (lembrando que os estados indesejados são os que tem $y \cdot s \neq 0 \pmod 2$)
- Dado um y, temos $\beta_y = \frac{1}{\sqrt{2}}\frac{(-1)^{r\cdot y}}{2^{n/2}} + \frac{1}{\sqrt{2}}\frac{(-1)^{(r\oplus s)\cdot y}}{2^{n/2}} = \frac{(-1)^{r\cdot y}}{2^{(n+1)/2}}[1 + (-1)^{s\cdot y}]$

O Algoritmo de Simon

Relembrando como obtivemos a string aleatória y com a propriedade

$$y \cdot s = 0 \pmod 2:$$

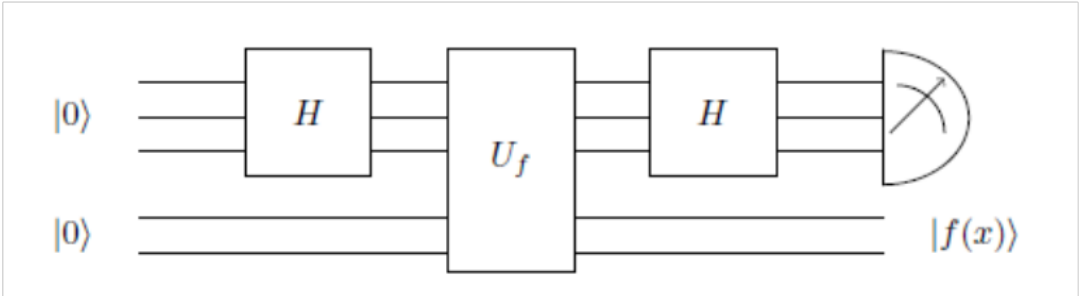


- Primeiro medimos o segundo registrador (superposição de valores $f(x)$)
- Obtivemos no primeiro registrador os valores x consistentes com $f(x)$: a superposição $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$, para um r aleatório
- Fizemos uma amostragem de Fourier em $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$
- Resultado: uma superposição de estados $|y\rangle$, onde cada $|y\rangle$ tem amplitude β_y
- Mostramos que os estados $|y\rangle$ indesejados tem amplitude $\beta_y = 0$ (lembrando que os estados indesejados são os que tem $y \cdot s \neq 0 \pmod 2$)
- Dado um y, temos $\beta_y = \frac{1}{\sqrt{2}}\frac{(-1)^{r\cdot y}}{2^{n/2}} + \frac{1}{\sqrt{2}}\frac{(-1)^{(r\oplus s)\cdot y}}{2^{n/2}} = \frac{(-1)^{r\cdot y}}{2^{(n+1)/2}}[1 + (-1)^{s\cdot y}]$
  - Caso 1: $y \cdot s = 0 \pmod 2 \longrightarrow \beta_y = \frac{(-1)^{r\cdot y}}{2^{(n-1)/2}}$

O Algoritmo de Simon

Relembrando como obtivemos a string aleatória y com a propriedade

$$y \cdot s = 0 \pmod 2:$$



- Primeiro medimos o segundo registrador (superposição de valores $f(x)$)
- Obtivemos no primeiro registrador os valores x consistentes com $f(x)$: a superposição $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$, para um r aleatório
- Fizemos uma amostragem de Fourier em $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$
- Resultado: uma superposição de estados $|y\rangle$, onde cada $|y\rangle$ tem amplitude β_y
- Mostramos que os estados $|y\rangle$ indesejados tem amplitude $\beta_y = 0$ (lembrando que os estados indesejados são os que tem $y \cdot s \neq 0 \pmod 2$)
- Dado um y, temos $\beta_y = \frac{1}{\sqrt{2}}\frac{(-1)^{r\cdot y}}{2^{n/2}} + \frac{1}{\sqrt{2}}\frac{(-1)^{(r\oplus s)\cdot y}}{2^{n/2}} = \frac{(-1)^{r\cdot y}}{2^{(n+1)/2}}[1 + (-1)^{s\cdot y}]$
  - Caso 1: $y \cdot s = 0 \pmod 2 \longrightarrow \beta_y = \frac{(-1)^{r\cdot y}}{2^{(n-1)/2}}$
  - Caso 2: $y \cdot s = 1 \pmod 2 \longrightarrow \beta_y = 0$

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0,1\}^n \to \{0,1\}^n$ com $s \in \{0,1\}^n$ tal que se $x \neq y$ e $f(x) = f(y)$, então $x \oplus s = y$ (i.e., $f(x \oplus s) = f(x)$).

Saida: Retornar a string s (obs: assumimos $s \neq 0^n$)

Ideia do Algoritmo

- Para algum r aleatório, obter o estado $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$ (OK)
- Obter string aleatória y com a propriedade $y \cdot s = 0 \pmod 2$ (OK)
- Repetir isso para obter $n-1$ strings y com tal propriedade Agora!

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0,1\}^n \to \{0,1\}^n$ com $s \in \{0,1\}^n$ tal que se $x \neq y$ e $f(x) = f(y)$, então $x \oplus s = y$ (i.e., $f(x \oplus s) = f(x)$).

Saida: Retornar a string s (obs: assumimos $s \neq 0^n$)

Ideia do Algoritmo

- Para algum r aleatório, obter o estado $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$ (OK)
- Obter string aleatória y com a propriedade $y \cdot s = 0 \pmod 2$ (OK)
- Repetir isso para obter $n-1$ strings y com tal propriedade Agora!

Lembrando que cada $y \cdot s = 0$ nos dá uma equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$ para resolvermos um sistema linear (mod 2)

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0,1\}^n \to \{0,1\}^n$ com $s \in \{0,1\}^n$ tal que se $x \neq y$ e $f(x) = f(y)$, então $x \oplus s = y$ (i.e., $f(x \oplus s) = f(x)$).

Saida: Retornar a string s (obs: assumimos $s \neq 0^n$)

Ideia do Algoritmo

- Para algum r aleatório, obter o estado $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$ (OK)
- Obter string aleatória y com a propriedade $y \cdot s = 0 \pmod 2$ (OK)
- Repetir isso para obter $n-1$ strings y com tal propriedade Agora!

Lembrando que cada $y \cdot s = 0$ nos dá uma equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$ para resolvermos um sistema linear (mod 2)

- Problema: queremos que as $n-1$ equações sejam linearmente independentes

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0,1\}^n \to \{0,1\}^n$ com $s \in \{0,1\}^n$ tal que se $x \neq y$ e $f(x) = f(y)$, então $x \oplus s = y$ (i.e., $f(x \oplus s) = f(x)$).

Saida: Retornar a string s (obs: assumimos $s \neq 0^n$)

Ideia do Algoritmo

- Para algum r aleatório, obter o estado $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$ (OK)
- Obter string aleatória y com a propriedade $y \cdot s = 0 \pmod 2$ (OK)
- Repetir isso para obter $n-1$ strings y com tal propriedade Agora!

Lembrando que cada $y \cdot s = 0$ nos dá uma equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$ para resolvermos um sistema linear (mod 2)

- Problema: queremos que as $n-1$ equações sejam linearmente independentes
- Veremos que a probabilidade de que o algoritmo gere $n-1$ equações linearmente independentes é $\geq 1/4$, portanto basta usar tentativas repetidas:

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0,1\}^n \to \{0,1\}^n$ com $s \in \{0,1\}^n$ tal que se $x \neq y$ e $f(x) = f(y)$, então $x \oplus s = y$ (i.e., $f(x \oplus s) = f(x)$).

Saida: Retornar a string s (obs: assumimos $s \neq 0^n$)

Ideia do Algoritmo

- Para algum r aleatório, obter o estado $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$ (OK)
- Obter string aleatória y com a propriedade $y \cdot s = 0 \pmod 2$ (OK)
- Repetir isso para obter $n-1$ strings y com tal propriedade Agora!

Lembrando que cada $y \cdot s = 0$ nos dá uma equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$ para resolvermos um sistema linear (mod 2)

- Problema: queremos que as $n-1$ equações sejam linearmente independentes
- Veremos que a probabilidade de que o algoritmo gere $n-1$ equações linearmente independentes é $\geq 1/4$, portanto basta usar tentativas repetidas:
  - Resolvendo o sistema e obtendo s, apenas testamos se $f(x) = f(x \oplus s)$

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0,1\}^n \to \{0,1\}^n$ com $s \in \{0,1\}^n$ tal que se $x \neq y$ e $f(x) = f(y)$, então $x \oplus s = y$ (i.e., $f(x \oplus s) = f(x)$).

Saida: Retornar a string s (obs: assumimos $s \neq 0^n$)

Ideia do Algoritmo

- Para algum r aleatório, obter o estado $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$ (OK)
- Obter string aleatória y com a propriedade $y \cdot s = 0 \pmod 2$ (OK)
- Repetir isso para obter $n-1$ strings y com tal propriedade Agora!

Lembrando que cada $y \cdot s = 0$ nos dá uma equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$ para resolvermos um sistema linear (mod 2)

- Problema: queremos que as $n-1$ equações sejam linearmente independentes
- Veremos que a probabilidade de que o algoritmo gere $n-1$ equações linearmente independentes é $\geq 1/4$, portanto basta usar tentativas repetidas:
  - Resolvendo o sistema e obtendo s, apenas testamos se $f(x) = f(x \oplus s)$
  - Caso o teste falhe, rodamos novamente, digamos 100 vezes.

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0,1\}^n \to \{0,1\}^n$ com $s \in \{0,1\}^n$ tal que se $x \neq y$ e $f(x) = f(y)$, então $x \oplus s = y$ (i.e., $f(x \oplus s) = f(x)$).

Saida: Retornar a string s (obs: assumimos $s \neq 0^n$)

Ideia do Algoritmo

- Para algum r aleatório, obter o estado $\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$ (OK)
- Obter string aleatória y com a propriedade $y \cdot s = 0 \pmod 2$ (OK)
- Repetir isso para obter $n-1$ strings y com tal propriedade Agora!

Lembrando que cada $y \cdot s = 0$ nos dá uma equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$ para resolvermos um sistema linear (mod 2)

- Problema: queremos que as $n-1$ equações sejam linearmente independentes
- Veremos que a probabilidade de que o algoritmo gere $n-1$ equações linearmente independentes é $\geq 1/4$, portanto basta usar tentativas repetidas:
  - Resolvendo o sistema e obtendo s, apenas testamos se $f(x) = f(x \oplus s)$
  - Caso o teste falhe, rodamos novamente, digamos 100 vezes.
  - A Probabilidade de falhar 100 vezes é $\leq 0.75^{100}$, ou seja, nula para efeitos práticos.

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)

  Probabilidade de "dar problema" $(y = y_1 y_2 ... y_n = 0)$: $\frac{1}{2^{n-1}}$

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)

  Probabilidade de "dar problema" $(y = y_1 y_2 ... y_n = 0)$: $\frac{1}{2^{n-1}}$

- A segunda equação deve ser diferente da primeira e $y \neq 0$

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)

  Probabilidade de "dar problema" $(y = y_1 y_2 ... y_n = 0)$: $\frac{1}{2^{n-1}}$

- A segunda equação deve ser diferente da primeira e $y \neq 0$

  Probabilidade de "dar problema": $\frac{2}{2^{n-1}}$

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)

  Probabilidade de "dar problema" $(y = y_1 y_2 ... y_n = 0)$: $\frac{1}{2^{n-1}}$

- A segunda equação deve ser diferente da primeira e $y \neq 0$

  Probabilidade de "dar problema": $\frac{2}{2^{n-1}}$

- A terceira não pode ser combinação linear duas primeiras equações

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)

  Probabilidade de "dar problema" $(y = y_1 y_2 ... y_n = 0)$: $\frac{1}{2^{n-1}}$

- A segunda equação deve ser diferente da primeira e $y \neq 0$

  Probabilidade de "dar problema": $\frac{2}{2^{n-1}}$

- A terceira não pode ser combinação linear duas primeiras equações

  Probabilidade de "dar problema": $\frac{4}{2^{n-1}}$

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)

  Probabilidade de "dar problema" $(y = y_1 y_2 ... y_n = 0)$: $\frac{1}{2^{n-1}}$

- A segunda equação deve ser diferente da primeira e $y \neq 0$

  Probabilidade de "dar problema": $\frac{2}{2^{n-1}}$

- A terceira não pode ser combinação linear duas primeiras equações

  Probabilidade de "dar problema": $\frac{4}{2^{n-1}}$ (são 4 c.l., pois estamos em $\mathbb{Z}_2$)

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)

  Probabilidade de "dar problema" $(y = y_1 y_2 ... y_n = 0)$: $\frac{1}{2^{n-1}}$

- A segunda equação deve ser diferente da primeira e $y \neq 0$

  Probabilidade de "dar problema": $\frac{2}{2^{n-1}}$

- A terceira não pode ser combinação linear duas primeiras equações

  Probabilidade de "dar problema": $\frac{4}{2^{n-1}}$ (são 4 c.l., pois estamos em $\mathbb{Z}_2$)
  (tem que ser $\neq 0$, $\neq$ primeira equação, $\neq$ segunda equação, $\neq$ soma das duas)

  $\vdots$

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)
  Probabilidade de "dar problema" $(y = y_1 y_2 ... y_n = 0)$: $\frac{1}{2^{n-1}}$
- A segunda equação deve ser diferente da primeira e $y \neq 0$
  Probabilidade de "dar problema": $\frac{2}{2^{n-1}}$
- A terceira não pode ser combinação linear duas primeiras equações
  Probabilidade de "dar problema": $\frac{4}{2^{n-1}}$ (são 4 c.l., pois estamos em $\mathbb{Z}_2$)
  (tem que ser $\neq 0$, $\neq$ primeira equação, $\neq$ segunda equação, $\neq$ soma das duas)
  $\vdots$
- A $n-2$-ésima equação não pode combinação linear das equações anteriores

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)
  Probabilidade de "dar problema" $(y = y_1 y_2 ... y_n = 0)$: $\frac{1}{2^{n-1}}$
- A segunda equação deve ser diferente da primeira e $y \neq 0$
  Probabilidade de "dar problema": $\frac{2}{2^{n-1}}$
- A terceira não pode ser combinação linear duas primeiras equações
  Probabilidade de "dar problema": $\frac{4}{2^{n-1}}$ (são 4 c.l., pois estamos em $\mathbb{Z}_2$)
  (tem que ser $\neq 0$, $\neq$ primeira equação, $\neq$ segunda equação, $\neq$ soma das duas)
  $\vdots$
- A $n-2$-ésima equação não pode combinação linear das equações anteriores
  Probabilidade de "dar problema": $\frac{2^{n-3}}{2^{n-1}} = \frac{1}{4}$

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)
  Probabilidade de "dar problema" $(y = y_1 y_2 ... y_n = 0)$: $\frac{1}{2^{n-1}}$
- A segunda equação deve ser diferente da primeira e $y \neq 0$
  Probabilidade de "dar problema": $\frac{2}{2^{n-1}}$
- A terceira não pode ser combinação linear duas primeiras equações
  Probabilidade de "dar problema": $\frac{4}{2^{n-1}}$ (são 4 c.l., pois estamos em $\mathbb{Z}_2$)
  (tem que ser $\neq 0$, $\neq$ primeira equação, $\neq$ segunda equação, $\neq$ soma das duas)
  $\vdots$
- A $n-2$-ésima equação não pode combinação linear das equações anteriores
  Probabilidade de "dar problema": $\frac{2^{n-3}}{2^{n-1}} = \frac{1}{4}$
- A $(n-1)$-ésima equação não pode ser combinação linear das equações anteriores

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)
  Probabilidade de "dar problema" $(y = y_1 y_2 ... y_n = 0)$: $\frac{1}{2^{n-1}}$
- A segunda equação deve ser diferente da primeira e $y \neq 0$
  Probabilidade de "dar problema": $\frac{2}{2^{n-1}}$
- A terceira não pode ser combinação linear duas primeiras equações
  Probabilidade de "dar problema": $\frac{4}{2^{n-1}}$ (são 4 c.l., pois estamos em $\mathbb{Z}_2$)
  (tem que ser $\neq 0$, $\neq$ primeira equação, $\neq$ segunda equação, $\neq$ soma das duas)
  $\vdots$
- A $n-2$-ésima equação não pode combinação linear das equações anteriores
  Probabilidade de "dar problema": $\frac{2^{n-3}}{2^{n-1}} = \frac{1}{4}$
- A $(n-1)$-ésima equação não pode ser combinação linear das equações anteriores
  Probabilidade de "dar problema": $\frac{2^{n-2}}{2^{n-1}} = \frac{1}{2}$

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)
  Probabilidade de "dar problema" $(y = y_1 y_2 ... y_n = 0)$: $\frac{1}{2^{n-1}}$
- A segunda equação deve ser diferente da primeira e $y \neq 0$
  Probabilidade de "dar problema": $\frac{2}{2^{n-1}}$
- A terceira não pode ser combinação linear duas primeiras equações
  Probabilidade de "dar problema": $\frac{4}{2^{n-1}}$ (são 4 c.l., pois estamos em $\mathbb{Z}_2$)
  (tem que ser $\neq 0$, $\neq$ primeira equação, $\neq$ segunda equação, $\neq$ soma das duas)
  $\vdots$
- A $n-2$-ésima equação não pode combinação linear das equações anteriores
  Probabilidade de "dar problema": $\frac{2^{n-3}}{2^{n-1}} = \frac{1}{4}$
- A $(n-1)$-ésima equação não pode ser combinação linear das equações anteriores
  Probabilidade de "dar problema": $\frac{2^{n-2}}{2^{n-1}} = \frac{1}{2}$

Prob. de "dar problema" até penúltima equação é uma soma geométrica $\leq 1/2$

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)
  Probabilidade de "dar problema" $(y = y_1 y_2 ... y_n = 0)$: $\frac{1}{2^{n-1}}$
- A segunda equação deve ser diferente da primeira e $y \neq 0$
  Probabilidade de "dar problema": $\frac{2}{2^{n-1}}$
- A terceira não pode ser combinação linear duas primeiras equações
  Probabilidade de "dar problema": $\frac{4}{2^{n-1}}$ (são 4 c.l., pois estamos em $\mathbb{Z}_2$)
  (tem que ser $\neq 0$, $\neq$ primeira equação, $\neq$ segunda equação, $\neq$ soma das duas)
  $\vdots$
- A $n-2$-ésima equação não pode combinação linear das equações anteriores
  Probabilidade de "dar problema": $\frac{2^{n-3}}{2^{n-1}} = \frac{1}{4}$
- A $(n-1)$-ésima equação não pode ser combinação linear das equações anteriores
  Probabilidade de "dar problema": $\frac{2^{n-2}}{2^{n-1}} = \frac{1}{2}$

Prob. de "dar problema" até penúltima equação é uma soma geométrica $\leq 1/2$
Prob. de "dar problema" na última equação é $1/2$

O Algoritmo de Simon

Probabilidade de que as equações obtidas sejam L.I é $\geq 1/4$:

- A primeira equação $y_1 s_1 + y_2 s_2 + ... + y_n s_n = 0$
  (qualquer uma serve, exceto $y_i = 0$, para todo i)

  Probabilidade de "dar problema" $(y = y_1 y_2 ... y_n = 0)$: $\frac{1}{2^{n-1}}$

- A segunda equação deve ser diferente da primeira e $y \neq 0$

  Probabilidade de "dar problema": $\frac{2}{2^{n-1}}$

- A terceira não pode ser combinação linear duas primeiras equações

  Probabilidade de "dar problema": $\frac{4}{2^{n-1}}$ (são 4 c.l., pois estamos em $\mathbb{Z}_2$)
  (tem que ser $\neq 0$, $\neq$ primeira equação, $\neq$ segunda equação, $\neq$ soma das duas)

  $\vdots$

- A $n-2$-ésima equação não pode combinação linear das equações anteriores

  Probabilidade de "dar problema": $\frac{2^{n-3}}{2^{n-1}} = \frac{1}{4}$

- A $(n-1)$-ésima equação não pode ser combinação linear das equações anteriores

  Probabilidade de "dar problema": $\frac{2^{n-2}}{2^{n-1}} = \frac{1}{2}$

Prob. de "dar problema" até penúltima equação é uma soma geométrica $\leq 1/2$
Prob. de "dar problema" na última equação é $1/2$

Probabilidade geral de dar certo (todas eq. L.I.) é $\geq \frac{1}{4}$