

Computação Quântica

Aula 11

Murilo V. G. da Silva

DINF/UFPR

O Algoritmo de Simon

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

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $s \neq 0^n$)

Nota: novamente estamos no “modelo *black box*”

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $s \neq 0^n$)

Nota: novamente estamos no “modelo *black box*”

Nota: $N = 2^n$.

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $s \neq 0^n$)

Nota: novamente estamos no “modelo *black box*”

Nota: $N = 2^n$.

Pergunta:

- Com um algoritmo clássico, quantas *queries* precisamos fazer?

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $s \neq 0^n$)

Nota: novamente estamos no “modelo *black box*”

Nota: $N = 2^n$.

Pergunta:

- Com um algoritmo clássico, quantas *queries* precisamos fazer?
- Podemos fazer melhor usando um algoritmo quântico?

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $s \neq 0^n$)

Nota: novamente estamos no “modelo *black box*”

Nota: $N = 2^n$.

Pergunta:

- Com um algoritmo clássico, quantas *queries* precisamos fazer?
- Podemos fazer melhor usando um algoritmo quântico?

Algoritmos clássicos:

- Determinístico: $\mathcal{O}(N) = \mathcal{O}(2^n)$

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $s \neq 0^n$)

Nota: novamente estamos no “modelo *black box*”

Nota: $N = 2^n$.

Pergunta:

- Com um algoritmo clássico, quantas *queries* precisamos fazer?
- Podemos fazer melhor usando um algoritmo quântico?

Algoritmos clássicos:

- Determinístico: $\mathcal{O}(N) = \mathcal{O}(2^n)$
- Aleatorizado: $\mathcal{O}(\sqrt{N})$, lembrando que $\sqrt{N} = 2^{n/2}$

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$
(Assunto de hoje)

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$
(Assunto de hoje)
- A partir do estado acima, obter string y aleatória tal que $y \cdot s = 0 \pmod{2}$
(Assunto de hoje)

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$
(Assunto de hoje)
- A partir do estado acima, obter string y aleatória tal que $y \cdot s = 0 \pmod{2}$
(Assunto de hoje)
- Repetir isso para obter $n - 1$ strings y com tal propriedade
(Aula que vem)

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$
(Assunto de hoje)
- A partir do estado acima, obter string y aleatória tal que $y \cdot s = 0 \pmod{2}$
(Assunto de hoje)
- Repetir isso para obter $n - 1$ strings y com tal propriedade
(Aula que vem)

Ideia central do algoritmo:

O Algoritmo de Simon

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

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

Passos do Algoritmo

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

Ideia central do algoritmo:

- cada string y nos dá a equação $y_1 s_1 + y_2 s_2 + \dots + y_n s_n = 0 \pmod{2}$
(incógnitas $s_1, \dots, s_n$ e coeficientes $y_1, \dots, y_n$)

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$
(Assunto de hoje)
- A partir do estado acima, obter string y aleatória tal que $y \cdot s = 0 \pmod{2}$
(Assunto de hoje)
- Repetir isso para obter $n - 1$ strings y com tal propriedade
(Aula que vem)

Ideia central do algoritmo:

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

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$
(Assunto de hoje)
- A partir do estado acima, obter string y aleatória tal que $y \cdot s = 0 \pmod{2}$
(Assunto de hoje)
- Repetir isso para obter $n - 1$ strings y com tal propriedade
(Aula que vem)

Ideia central do algoritmo:

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

Nota: como estamos trabalhando $\pmod{2}$, há duas soluções: a solução trivial 0^n e a solução s , cujos bits são das variáveis $s_1, \dots, s_n$ 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 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$
(Assunto de hoje)
- A partir do estado acima, obter string y aleatória tal que $y \cdot s = 0 \pmod{2}$
(Assunto de hoje)
- Repetir isso para obter $n - 1$ strings y com tal propriedade
(Aula que vem)

Ideia central do algoritmo:

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

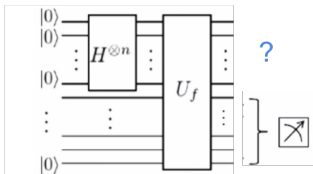
Nota: como estamos trabalhando $\pmod{2}$, há duas soluções: a solução trivial 0^n e a solução s , cujos bits são das variáveis $s_1, \dots, s_n$ e que pode ser obtida usando eliminação gaussiana módulo 2 que roda em tempo $\mathcal{O}(n^3)$

- Cuidado extra: as equações obtidas devem ser linearmente independentes

(1) Como obter o estado $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$?

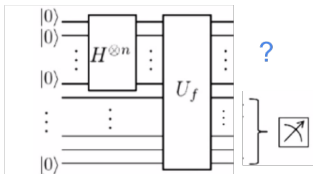
O Algoritmo de Simon

Considere o circuito abaixo:



O Algoritmo de Simon

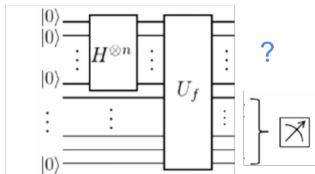
Considere o circuito abaixo:



Qual a saída dos n qubits da parte de cima do circuito?

O Algoritmo de Simon

Considere o circuito abaixo:

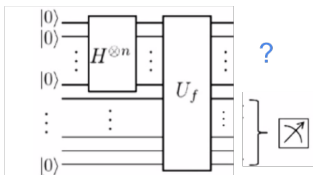


Qual a saída dos n qubits da parte de cima do circuito?

- Estado dos $2n$ qubits saindo de U_f antes da medida: $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle$

O Algoritmo de Simon

Considere o circuito abaixo:

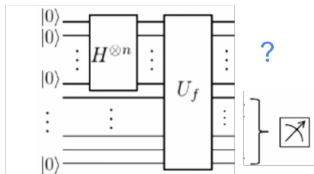


Qual a saída dos n qubits da parte de cima do circuito?

- Estado dos $2n$ qubits saindo de U_f antes da medida: $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle$
- Observe que a parte de cima do circuito contém x e a parte de baixo $f(x)$

O Algoritmo de Simon

Considere o circuito abaixo:

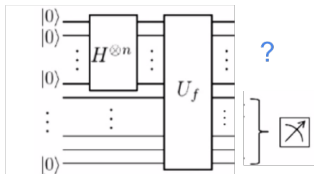


Qual a saída dos n qubits da parte de cima do circuito?

- Estado dos $2n$ qubits saindo de U_f antes da medida: $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle$
- Observe que a parte de cima do circuito contém x e a parte de baixo $f(x)$
- Medindo os n qubits da parte de baixo, estes qubits colapsam em algum valor y

O Algoritmo de Simon

Considere o circuito abaixo:

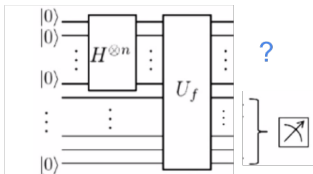


Qual a saída dos n qubits da parte de cima do circuito?

- Estado dos $2n$ qubits saindo de U_f antes da medida: $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle$
- Observe que a parte de cima do circuito contém x e a parte de baixo $f(x)$
- Medindo os n qubits da parte de baixo, estes qubits colapsam em algum valor y
- Observe que $y = f(r)$, para algum valor aleatório r

O Algoritmo de Simon

Considere o circuito abaixo:

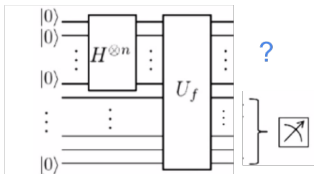


Qual a saída dos n qubits da parte de cima do circuito?

- Estado dos $2n$ qubits saindo de U_f antes da medida: $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle$
- Observe que a parte de cima do circuito contém x e a parte de baixo $f(x)$
- Medindo os n qubits da parte de baixo, estes qubits colapsam em algum valor y
- Observe que $y = f(r)$, para algum valor aleatório r
- Na realidade, pela definição de f , dado $y = f(r) = f(r \oplus s)$, para um r aleatório

O Algoritmo de Simon

Considere o circuito abaixo:

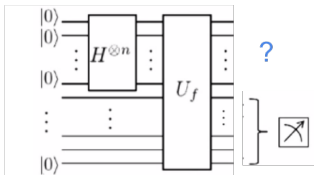


Qual a saída dos n qubits da parte de cima do circuito?

- Estado dos $2n$ qubits saindo de U_f antes da medida: $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle$
- Observe que a parte de cima do circuito contém x e a parte de baixo $f(x)$
- Medindo os n qubits da parte de baixo, estes qubits colapsam em algum valor y
- Observe que $y = f(r)$, para algum valor aleatório r
- Na realidade, pela definição de f , dado $y = f(r) = f(r \oplus s)$, para um r aleatório
- Depois da medida parcial o estado será condizente com os bits medidos, e portanto os $2n$ qubits serão $\frac{1}{\sqrt{2}} \sum_{x \in \{r, r \oplus s\}} |x\rangle |f(x)\rangle = \frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle |y\rangle$

O Algoritmo de Simon

Considere o circuito abaixo:



Qual a saída dos n qubits da parte de cima do circuito?

- Estado dos $2n$ qubits saindo de U_f antes da medida: $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle$
- Observe que a parte de cima do circuito contém x e a parte de baixo $f(x)$
- Medindo os n qubits da parte de baixo, estes qubits colapsam em algum valor y
- Observe que $y = f(r)$, para algum valor aleatório r
- Na realidade, pela definição de f , dado $y = f(r) = f(r \oplus s)$, para um r aleatório
- Depois da medida parcial o estado será condizente com os bits medidos, e portanto os $2n$ qubits serão $\frac{1}{\sqrt{2}} \sum_{x \in \{r, r \oplus s\}} |x\rangle |f(x)\rangle = \frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle |y\rangle$
- Com isso o estado dos n qubits de cima é $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$

O Algoritmo de Simon

Entrada: Uma função 2-para-1 $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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 $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$ (Resolvido!!)
- A partir do estado acima, obter string y aleatória tal que $y \cdot s = 0 \pmod{2}$ (Agora)
- Repetir isso para obter $n - 1$ strings y com tal propriedade (Aula que vem)

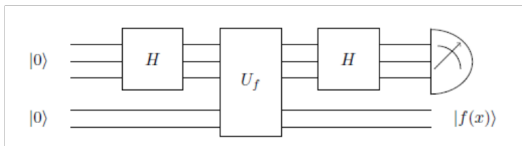
(2) Como obter string aleatória y , tal que $y \cdot s = 0 \pmod{2}$?

O Algoritmo de Simon

Obtendo string aleatória y com a propriedade $y \cdot s = 0 \pmod{2}$:

O Algoritmo de Simon

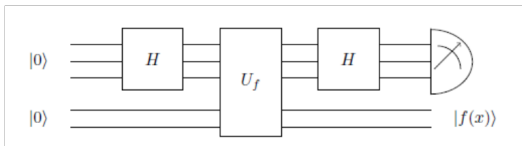
Obtendo string aleatória y com a propriedade $y \cdot s = 0 \pmod{2}$:



- Basta fazer uma amostragem de Fourier em $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$

O Algoritmo de Simon

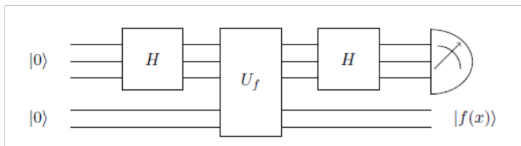
Obtendo string aleatória y com a propriedade $y \cdot s = 0 \pmod{2}$:



- Basta fazer uma amostragem de Fourier em $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$
- Nos qubits de cima, depois de $H^{\otimes n}$ e antes da medida, temos uma superposição de estados $|y\rangle$, onde cada estado $|y\rangle$ tem amplitude β_y

O Algoritmo de Simon

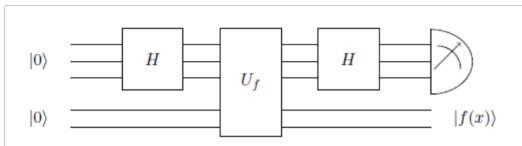
Obtendo string aleatória y com a propriedade $y \cdot s = 0 \pmod{2}$:



- Basta fazer uma amostragem de Fourier em $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$
- Nos qubits de cima, depois de $H^{\otimes n}$ e antes da medida, temos uma superposição de estados $|y\rangle$, onde cada estado $|y\rangle$ tem amplitude β_y
- Ideia: mostrar 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

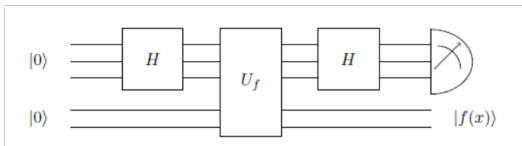
Obtendo string aleatória y com a propriedade $y \cdot s = 0 \pmod{2}$:



- Basta fazer uma amostragem de Fourier em $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$
- Nos qubits de cima, depois de $H^{\otimes n}$ e antes da medida, temos uma superposição de estados $|y\rangle$, onde cada estado $|y\rangle$ tem amplitude β_y
- Ideia: mostrar 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

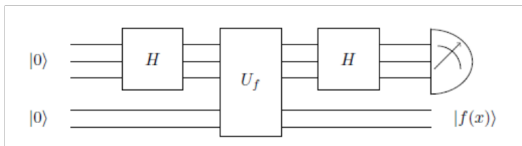
Obtendo string aleatória y com a propriedade $y \cdot s = 0 \pmod{2}$:



- Basta fazer uma amostragem de Fourier em $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$
- Nos qubits de cima, depois de $H^{\otimes n}$ e antes da medida, temos uma superposição de estados $|y\rangle$, onde cada estado $|y\rangle$ tem amplitude β_y
- Ideia: mostrar 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

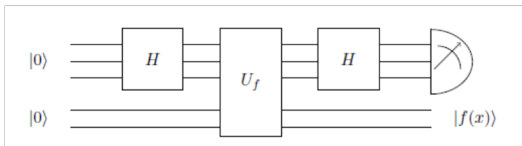
Obtendo string aleatória y com a propriedade $y \cdot s = 0 \pmod{2}$:



- Basta fazer uma amostragem de Fourier em $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$
- Nos qubits de cima, depois de $H^{\otimes n}$ e antes da medida, temos uma superposição de estados $|y\rangle$, onde cada estado $|y\rangle$ tem amplitude β_y
- Ideia: mostrar 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

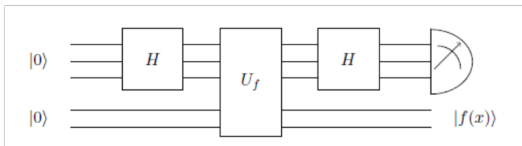
Obtendo string aleatória y com a propriedade $y \cdot s = 0 \pmod{2}$:



- Basta fazer uma amostragem de Fourier em $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$
- Nos qubits de cima, depois de $H^{\otimes n}$ e antes da medida, temos uma superposição de estados $|y\rangle$, onde cada estado $|y\rangle$ tem amplitude β_y
- Ideia: mostrar 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

Obtendo string aleatória y com a propriedade $y \cdot s = 0 \pmod{2}$:



- Basta fazer uma amostragem de Fourier em $\frac{1}{\sqrt{2}} |r\rangle + \frac{1}{\sqrt{2}} |r \oplus s\rangle$
- Nos qubits de cima, depois de $H^{\otimes n}$ e antes da medida, temos uma superposição de estados $|y\rangle$, onde cada estado $|y\rangle$ tem amplitude β_y
- Ideia: mostrar 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: $y \cdot s = 1 \pmod{2} \longrightarrow \beta_y = 0$