Computação Quântica
Aula 10

Murilo V. G. da Silva

DINF/UFPR

Conceitos Preliminares

Relembrando conceitos de Teoria da Computação:

- **Problema de Decisão:** Uma linguagem sobre o alfabeto $\{0, 1\}$

Conceitos Preliminares

Relembrando conceitos de Teoria da Computação:

- **Problema de Decisão:** Uma linguagem sobre o alfabeto $\{0, 1\}$
  - $L_p = \{w \in \{0, 1\}^* \ ; \ w$ é a representação binária de um número primo$\}$

Conceitos Preliminares

Relembrando conceitos de Teoria da Computação:

- **Problema de Decisão:** Uma linguagem sobre o alfabeto $\{0, 1\}$
  - $L_p = \{w \in \{0,1\}^* \ ; \ w$ é a representação binária de um número primo$\}$
  - $L_{\text{SAT}} = \{\llcorner\phi\lrcorner \in \{0,1\}^* \ ; \ \phi$ é uma fórmula em CNF satisfazível$\}$

Conceitos Preliminares

Relembrando conceitos de Teoria da Computação:

- **Problema de Decisão:** Uma linguagem sobre o alfabeto $\{0, 1\}$
  - $L_p = \{w \in \{0,1\}^* \ ; \ w$ é a representação binária de um número primo$\}$
  - $L_{\text{SAT}} = \{\llcorner\phi\lrcorner \in \{0,1\}^* \ ; \ \phi$ é uma fórmula em CNF satisfazível$\}$
- Um problema de decisão L admite solução se existe uma MT que decida L

Conceitos Preliminares

Relembrando conceitos de Teoria da Computação:

- **Problema de Decisão:** Uma linguagem sobre o alfabeto $\{0,1\}$
  - $L_p = \{w \in \{0,1\}^* \ ; \ w$ é a representação binária de um número primo$\}$
  - $L_{\mathrm{SAT}} = \{\llcorner\phi\lrcorner \in \{0,1\}^* \ ; \ \phi$ é uma fórmula em CNF satisfazível$\}$
- Um problema de decisão L admite solução se existe uma MT que decida L

- **Problema de Busca:** Uma linguagem L sobre o alfabeto $\{0,1\}$ juntamente com uma função de solução $s_{\mathrm{L}} : L \to \mathcal{P}(\{0,1\}^*)$

Conceitos Preliminares

Relembrando conceitos de Teoria da Computação:

- **Problema de Decisão:** Uma linguagem sobre o alfabeto $\{0,1\}$
  - $L_p = \{w \in \{0,1\}^* \ ; \ w$ é a representação binária de um número primo$\}$
  - $L_{\text{SAT}} = \{\llcorner\phi\lrcorner \in \{0,1\}^* \ ; \ \phi$ é uma fórmula em CNF satisfazível$\}$
- Um problema de decisão L admite solução se existe uma MT que decida L
- **Problema de Busca:** Uma linguagem L sobre o alfabeto $\{0,1\}$ juntamente com uma função de solução $s_{\text{L}} : L \to \mathcal{P}(\{0,1\}^*)$
  - $L_{\text{COR}} = \{\llcorner G\lrcorner \in \{0,1\}^* \ ; \ G$ é um grafo$\}$ e $\forall x \in L_{\text{COR}}$, $s_{L_{\text{COR}}}(x)$ é o conjunto de strings representado colorações mínimas de $G\}$

Conceitos Preliminares

Relembrando conceitos de Teoria da Computação:

- **Problema de Decisão:** Uma linguagem sobre o alfabeto $\{0,1\}$
  - $L_p = \{w \in \{0,1\}^* \ ; \ w$ é a representação binária de um número primo$\}$
  - $L_{\text{SAT}} = \{\llcorner\phi\lrcorner \in \{0,1\}^* \ ; \ \phi$ é uma fórmula em CNF satisfazível$\}$
- Um problema de decisão L admite solução se existe uma MT que decida L
- **Problema de Busca:** Uma linguagem L sobre o alfabeto $\{0,1\}$ juntamente com uma função de solução $s_{\text{L}} : L \to \mathcal{P}(\{0,1\}^*)$
  - $L_{\text{COR}} = \{\llcorner G\lrcorner \in \{0,1\}^* \ ; \ G$ é um grafo$\}$ e $\forall x \in L_{\text{COR}}$, $s_{L_{\text{COR}}}(x)$ é o conjunto de strings representado colorações mínimas de $G\}$
  - $L_{\text{A}} = \{\llcorner f_a, 1^n\lrcorner \in \{0,1\}^n \ ; \ f_a : \{0,1\}^n \to \{0,1\}$ é a função booleana $f_a(x) = x \cdot a$, para a string "secreta" $a \in \{0,1\}^n\}$ e $\forall x \in L_{\text{A}}$, onde $x = \llcorner f_a, 1^n\lrcorner$, a solução para x é $s_{L_{\text{A}}}(x) = \{a\}$

Conceitos Preliminares

Relembrando conceitos de Teoria da Computação:

- **Problema de Decisão:** Uma linguagem sobre o alfabeto $\{0,1\}$
  - $L_p = \{w \in \{0,1\}^* \ ; \ w$ é a representação binária de um número primo$\}$
  - $L_{\text{SAT}} = \{\llcorner\phi\lrcorner \in \{0,1\}^* \ ; \ \phi$ é uma fórmula em CNF satisfazível$\}$
- Um problema de decisão L admite solução se existe uma MT que decida L

- **Problema de Busca:** Uma linguagem L sobre o alfabeto $\{0,1\}$ juntamente com uma função de solução $s_{\text{L}} : L \to \mathcal{P}(\{0,1\}^*)$
  - $L_{\text{COR}} = \{\llcorner G\lrcorner \in \{0,1\}^* \ ; \ G$ é um grafo$\}$ e $\forall x \in L_{\text{COR}}$, $s_{L_{\text{COR}}}(x)$ é o conjunto de strings representado colorações mínimas de $G\}$
  - $L_{\text{A}} = \{\llcorner f_a, 1^n\lrcorner \in \{0,1\}^n \ ; \ f_a : \{0,1\}^n \to \{0,1\}$ é a função booleana $f_a(x) = x \cdot a$, para a string "secreta" $a \in \{0,1\}^n\}$ e $\forall x \in L_{\text{A}}$, onde $x = \llcorner f_a, 1^n\lrcorner$, a solução para x é $s_{L_{\text{A}}}(x) = \{a\}$
- Um problema de busca L admite solução se existe uma MT que decida recebe $x \in L$ e retorne alguma string de $s_L(x)$.

Conceitos Preliminares

No slide anterior vimos problemas no modelo de computação “tradicional”, ou seja, no modelo de Máquinas de Turing

Conceitos Preliminares

No slide anterior vimos problemas no modelo de computação "tradicional", ou seja, no modelo de Máquinas de Turing

Modelo de computação Caixa Preta

Dada uma função $f : \{0,1\}^n \to \{0,1\}$, uma solução para um problema computacional no modelo caixa preta é uma MT M_f que tenha a habilidade de em um passo computacional obter o resultado para uma "query" para a função f para uma string qualquer de tamanho n.

Conceitos Preliminares

No slide anterior vimos problemas no modelo de computação "tradicional", ou seja, no modelo de Máquinas de Turing

Modelo de computação Caixa Preta

Dada uma função $f : \{0,1\}^n \to \{0,1\}$, uma solução para um problema computacional no modelo caixa preta é uma MT M_f que tenha a habilidade de em um passo computacional obter o resultado para uma "query" para a função f para uma string qualquer de tamanho n.

Interpretação alternativa do modelo (a mais comum): Você recebe código que computa f (mais precisamente $\llcorner M \lrcorner$, mas você pode apenas executar $M(x)$ (usando uma MT universal, por exemplo), mas sem ter permissão para examinar $\llcorner M \lrcorner$.

No slide anterior vimos problemas no modelo de computação "tradicional", ou seja, no modelo de Máquinas de Turing

Modelo de computação Caixa Preta

Dada uma função $f : \{0,1\}^n \to \{0,1\}$, uma solução para um problema computacional no modelo caixa preta é uma MT M_f que tenha a habilidade de em um passo computacional obter o resultado para uma "query" para a função f para uma string qualquer de tamanho n.

Interpretação alternativa do modelo (a mais comum): Você recebe código que computa f (mais precisamente $\llcorner M \lrcorner$, mas você pode apenas executar $M(x)$ (usando uma MT universal, por exemplo), mas sem ter permissão para examinar $\llcorner M \lrcorner$.

Considere o problema do slide anterior $L_{\mathrm{A}} = \{\llcorner f_a, 1^n \lrcorner \in \{0,1\}^n \; ; \; f_a : \{0,1\}^n \to \{0,1\}$ é a função booleana $f_a(x) = x \cdot a$, para a string "secreta" $a \in \{0,1\}^n\}$ e $\forall x \in L_{\mathrm{A}}$, onde $x = \llcorner f_a, 1^n \lrcorner$, a solução para x é $s_{L_{\mathrm{A}}}(x) = \{a\}$.

No slide anterior vimos problemas no modelo de computação "tradicional", ou seja, no modelo de Máquinas de Turing

Modelo de computação Caixa Preta

Dada uma função $f : \{0,1\}^n \to \{0,1\}$, uma solução para um problema computacional no modelo caixa preta é uma MT M_f que tenha a habilidade de em um passo computacional obter o resultado para uma "query" para a função f para uma string qualquer de tamanho n.

Interpretação alternativa do modelo (a mais comum): Você recebe código que computa f (mais precisamente $\llcorner M \lrcorner$, mas você pode apenas executar $M(x)$ (usando uma MT universal, por exemplo), mas sem ter permissão para examinar $\llcorner M \lrcorner$.

Considere o problema do slide anterior $L_{\mathrm{A}} = \{\llcorner f_a, 1^n \lrcorner \in \{0,1\}^n \; ; \; f_a : \{0,1\}^n \to \{0,1\}$ é a função booleana $f_a(x) = x \cdot a$, para a string "secreta" $a \in \{0,1\}^n\}$ e $\forall x \in L_{\mathrm{A}}$, onde $x = \llcorner f_a, 1^n \lrcorner$, a solução para x é $s_{L_{\mathrm{A}}}(x) = \{a\}$.

Qual é a complexidade de tempo deste problema no modelo **Caixa Preta**?

O Problema de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

O Problema de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

Saida: Retornar a string a

O Problema de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

Saida: Retornar a string a

Obs: Neste problema assumimos que estamos no "modelo *black box*"

(na prática podemos pensar que a entrada é um circuito ou um algoritmo que computa f, mas não podemos "inspecioná-los", sim apenas fazer *queries*.)

O Problema de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

Saida: Retornar a string a

Obs: Neste problema assumimos que estamos no "modelo *black box*"

(na prática podemos pensar que a entrada é um circuito ou um algoritmo que computa f, mas não podemos "inspecioná-los", sim apenas fazer *queries*.)

Pergunta:

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

O Problema de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

Saida: Retornar a string a

Obs: Neste problema assumimos que estamos no "modelo *black box*"

(na prática podemos pensar que a entrada é um circuito ou um algoritmo que computa f, mas não podemos "inspecioná-los", sim apenas fazer *queries*.)

Pergunta:

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

O Problema de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

Saida: Retornar a string a

Obs: Neste problema assumimos que estamos no "modelo *black box*"

(na prática podemos pensar que a entrada é um circuito ou um algoritmo que computa f, mas não podemos "inspecioná-los", sim apenas fazer *queries*.)

Pergunta:

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

Algoritmo clássico:

- Seja $x_i \in \{0,1\}^n$, onde x_i tem o i-ésimo bit setado para 1 e os demais para 0.

O Problema de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

Saida: Retornar a string a

Obs: Neste problema assumimos que estamos no "modelo *black box*"

(na prática podemos pensar que a entrada é um circuito ou um algoritmo que computa f, mas não podemos "inspecioná-los", sim apenas fazer *queries*.)

Pergunta:

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

Algoritmo clássico:

- Seja $x_i \in \{0,1\}^n$, onde x_i tem o i-ésimo bit setado para 1 e os demais para 0.
- Compute $f(x_i)$, para $i = 1, ..., n$.

O Problema de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

Saida: Retornar a string a

Obs: Neste problema assumimos que estamos no "modelo *black box*"

(na prática podemos pensar que a entrada é um circuito ou um algoritmo que computa f, mas não podemos "inspecioná-los", sim apenas fazer *queries*.)

Pergunta:

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

Algoritmo clássico:

- Seja $x_i \in \{0,1\}^n$, onde x_i tem o i-ésimo bit setado para 1 e os demais para 0.
- Compute $f(x_i)$, para $i = 1, ..., n$. Com isso revela-se o i-ésimo bit de a

O Problema de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

Saida: Retornar a string a

Obs: Neste problema assumimos que estamos no "modelo *black box*"

(na prática podemos pensar que a entrada é um circuito ou um algoritmo que computa f, mas não podemos "inspecioná-los", sim apenas fazer *queries*.)

Pergunta:

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

Algoritmo clássico:

- Seja $x_i \in \{0,1\}^n$, onde x_i tem o i-ésimo bit setado para 1 e os demais para 0.
- Compute $f(x_i)$, para $i = 1, ..., n$. Com isso revela-se o i-ésimo bit de a
- Note que como cada query pode revelar no máximo um bit, o algoritmo é ótimo neste modelo.

O Algoritmo de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

O Algoritmo de Vazirani-Bernstein

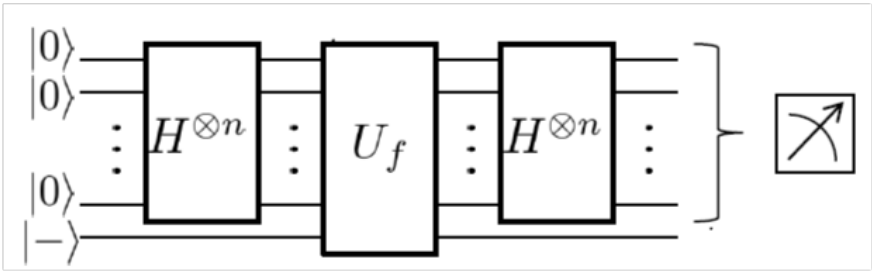
Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

Saida: Retornar a string a

O Algoritmo de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

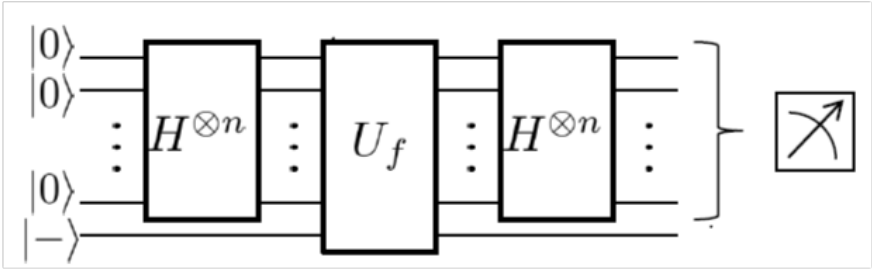
Saida: Retornar a string a



O Algoritmo de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

Saida: Retornar a string a

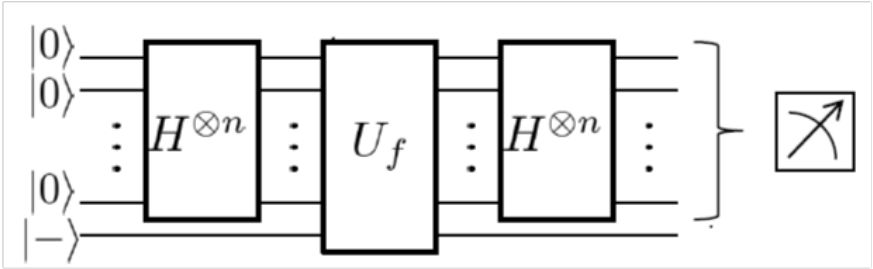


- Estado dos n qubits saindo da primeira transormada de Hadamard: $\frac{1}{2^{\frac{n}{2}}} \sum |x\rangle$

O Algoritmo de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

Saida: Retornar a string a

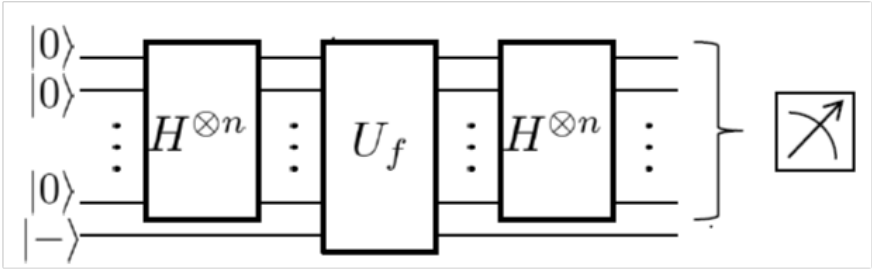


- Estado dos n qubits saindo da primeira transormada de Hadamard: $\frac{1}{2^{\frac{n}{2}}} \sum |x\rangle$
- Estado dos $n+1$ qubits depois de U_f: $\frac{1}{2^{\frac{n}{2}}} \sum (-1)^{f(x)} |x\rangle |-\rangle$

O Algoritmo de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

Saida: Retornar a string a

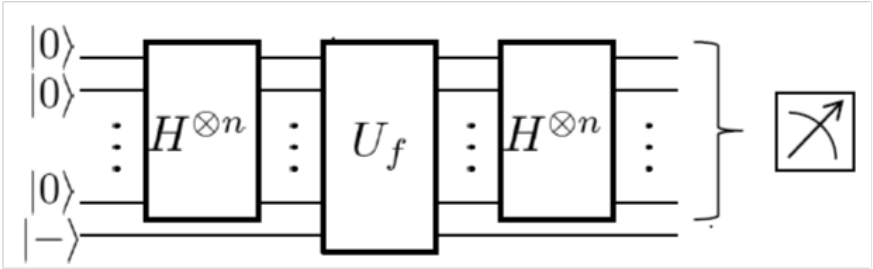


- Estado dos n qubits saindo da primeira transormada de Hadamard: $\frac{1}{2^{\frac{n}{2}}} \sum |x\rangle$
- Estado dos $n+1$ qubits depois de U_f: $\frac{1}{2^{\frac{n}{2}}} \sum (-1)^{f(x)} |x\rangle |-\rangle$
- Estado dos n qubits saindo da segunda transformada de Hadamard: $|a\rangle$

O Algoritmo de Vazirani-Bernstein

Entrada: Uma função $f : \{0,1\}^n \to \{0,1\}$ que computa $f(x) = x \cdot a$, para string a desconhecida.

Saida: Retornar a string a



- Estado dos n qubits saindo da primeira transormada de Hadamard: $\frac{1}{2^{\frac{n}{2}}} \sum |x\rangle$
- Estado dos $n+1$ qubits depois de U_f: $\frac{1}{2^{\frac{n}{2}}} \sum (-1)^{f(x)} |x\rangle |-\rangle$
- Estado dos n qubits saindo da segunda transformada de Hadamard: $|a\rangle$
- Medindo $|a\rangle$, obtém-se a com probabilidade 1.