Fast Quantum Algorithms
Lectures 1 and 2

Gábor Ivanyos
MTA SZTAKI

3rd de Brún Workshop, Galway 7-10 December, 2009.

Contents

Simon's problem

- $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ $(m \geq n)$

Simon's problem

- $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ $(m \geq n)$
  given by oracle performing $|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$

Simon's problem

- $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ $(m \geq n)$

  given by oracle performing $|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$
- Promise:

Simon's problem

- $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m \ (m \geq n)$

  given by oracle performing $|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$
- Promise:
  - either $\exists u$ s.t.

$$f(x) = f(x') \Leftrightarrow x' = x \text{ or } x' = x + u$$

Simon's problem

- $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ $(m \geq n)$

  given by oracle performing $|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$
- Promise:
  - either $\exists u$ s.t.

    $$f(x) = f(x') \Leftrightarrow x' = x \text{ or } x' = x + u$$

  - or f injective

Simon's problem

- $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ $(m \geq n)$

  given by oracle performing $|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$
- Promise:
  - either $\exists u$ s.t.

$$f(x) = f(x') \Leftrightarrow x' = x \text{ or } x' = x + u$$

  - or f injective
- Task: decide which is the case

Simon's problem

- $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ $(m \geq n)$

  given by oracle performing $|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$
- Promise:
  - either $\exists u$ s.t.

$$f(x) = f(x') \Leftrightarrow x' = x \text{ or } x' = x + u$$

  - or f injective
- Task: decide which is the case
  - and find u if $\exists$

Simon's problem

- $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ $(m \geq n)$

  given by oracle performing $|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$
- Promise:
  - either $\exists u$ s.t.

$$f(x) = f(x') \Leftrightarrow x' = x \text{ or } x' = x + u$$

  - or f injective
- Task: decide which is the case
  - and find u if $\exists$
  - Remark: reducible to the decision version

Simon's problem

- $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ $(m \geq n)$
  given by oracle performing $|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$
- Promise:
  - either $\exists u$ s.t.
    $$f(x) = f(x') \Leftrightarrow x' = x \text{ or } x' = x + u$$
  - or f injective
- Task: decide which is the case
  - and find u if $\exists$
  - Remark: reducible to the decision version
- Classically difficult:

Simon's problem

- $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ $(m \geq n)$

  given by oracle performing $|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$
- Promise:
  - either $\exists u$ s.t.

$$f(x) = f(x') \Leftrightarrow x' = x \text{ or } x' = x + u$$

  - or f injective
- Task: decide which is the case
  - and find u if $\exists$
  - Remark: reducible to the decision version
- Classically difficult:

  with $2^{\frac{n}{4}}$ queries can guess the case only with probability $\leq \frac{1}{2} + \frac{1}{2^{n/2}}$

Simon's algorithm

- $|0\rangle|0\rangle$

Simon's algorithm

- $|0\rangle|0\rangle$

  $\downarrow$ Hadamard$^{\otimes n}$

Simon's algorithm

- $|0\rangle|0\rangle$

  $\downarrow$ Hadamard$^{\otimes n}$

- $\sum_{x \in \mathbb{Z}_2^n} |x\rangle|0\rangle$

Simon's algorithm

- $|0\rangle|0\rangle$

  $\downarrow$ Hadamard$^{\otimes n}$

- $\sum_{x\in\mathbb{Z}_2^n}|x\rangle|0\rangle$

  $\downarrow$ f-oracle

Simon's algorithm

- $|0\rangle|0\rangle$

  $\downarrow$ Hadamard$^{\otimes n}$

- $\sum_{x\in\mathbb{Z}_2^n}|x\rangle|0\rangle$

  $\downarrow$ f-oracle

- $\sum_{x\in\mathbb{Z}_2^n}|x\rangle|f(x)\rangle$

Simon's algorithm

- $|0\rangle|0\rangle$

  $\downarrow$ Hadamard$^{\otimes n}$

- $\sum_{x\in\mathbb{Z}_2^n}|x\rangle|0\rangle$

  $\downarrow$ f-oracle

- $\sum_{x\in\mathbb{Z}_2^n}|x\rangle|f(x)\rangle$

  $\downarrow$ measure $f(x)$, drop it

Simon's algorithm

- $|0\rangle|0\rangle$

  $\downarrow$ Hadamard$^{\otimes n}$

- $\sum_{x\in\mathbb{Z}_2^n} |x\rangle|0\rangle$

  $\downarrow$ f-oracle

- $\sum_{x\in\mathbb{Z}_2^n} |x\rangle|f(x)\rangle$

  $\downarrow$ measure $f(x)$, drop it

- $|x\rangle + |x+u\rangle$

Simon's algorithm

- $|0\rangle|0\rangle$

  $\downarrow$ Hadamard$^{\otimes n}$

- $\sum_{x\in\mathbb{Z}_2^n}|x\rangle|0\rangle$

  $\downarrow$ f-oracle

- $\sum_{x\in\mathbb{Z}_2^n}|x\rangle|f(x)\rangle$

  $\downarrow$ measure $f(x)$, drop it

- $|x\rangle+|x+u\rangle$

  $\downarrow$ Hadamard$^{\otimes n}$

Simon's algorithm

- $|0\rangle|0\rangle$

  $\downarrow$ Hadamard$^{\otimes n}$

- $\sum_{x\in\mathbb{Z}_2^n}|x\rangle|0\rangle$

  $\downarrow$ f-oracle

- $\sum_{x\in\mathbb{Z}_2^n}|x\rangle|f(x)\rangle$

  $\downarrow$ measure $f(x)$, drop it

- $|x\rangle+|x+u\rangle$

  $\downarrow$ Hadamard$^{\otimes n}$

- $\sum_{y\in\mathbb{Z}_2^n}\left((-1)^{(x,y)}+(-1)^{(x,y)+(u,y)}\right)|y\rangle$

Simon's algorithm

- $|0\rangle|0\rangle$

  $\downarrow$ Hadamard$^{\otimes n}$

- $\sum_{x\in\mathbb{Z}_2^n}|x\rangle|0\rangle$

  $\downarrow$ f-oracle

- $\sum_{x\in\mathbb{Z}_2^n}|x\rangle|f(x)\rangle$

  $\downarrow$ measure $f(x)$, drop it

- $|x\rangle+|x+u\rangle$

  $\downarrow$ Hadamard$^{\otimes n}$

- $\sum_{y\in\mathbb{Z}_2^n}\left((-1)^{(x,y)}+(-1)^{(x,y)+(u,y)}\right)|y\rangle$

  $\downarrow$ $=$

Simon's algorithm

- $|0\rangle|0\rangle$

$\downarrow$ Hadamard$^{\otimes n}$

- $\sum_{x\in\mathbb{Z}_2^n} |x\rangle|0\rangle$

$\downarrow$ f-oracle

- $\sum_{x\in\mathbb{Z}_2^n} |x\rangle|f(x)\rangle$

$\downarrow$ measure $f(x)$, drop it

- $|x\rangle + |x+u\rangle$

$\downarrow$ Hadamard$^{\otimes n}$

- $\sum_{y\in\mathbb{Z}_2^n} \left((-1)^{(x,y)} + (-1)^{(x,y)+(u,y)}\right) |y\rangle$

$\downarrow$ $=$

- $\sum_{y\in u^\perp} (-1)^{(x,y)}|y\rangle$

Simon's algorithm 2

- $\sum_{y \in u^\perp} (-1)^{xy} |y\rangle$

Simon's algorithm 2

- $\sum_{y \in u^{\perp}} (-1)^{xy} |y\rangle$
  $\downarrow$ measure

Simon's algorithm 2

- $\sum_{y\in u^\perp}(-1)^{xy}|y\rangle$

  $\downarrow$ measure

- random $y \in u^\perp$

Simon's algorithm 2

- $\sum_{y \in u^\perp} (-1)^{xy} |y\rangle$

  $\downarrow$ measure

- random $y \in u^\perp$

- $\ell = O(n)$ iteration gives $y_1, \ldots, y_\ell$:

Simon's algorithm 2

- $\sum_{y \in u^\perp} (-1)^{xy} |y\rangle$

  $\downarrow$ measure

- random $y \in u^\perp$

- $\ell = O(n)$ iteration gives $y_1, \ldots, y_\ell$:
  $\{y_1, \ldots, y_\ell\}^\perp = \{0, u\}$ (probably).

Simon's algorithm 2

- $\sum_{y \in u^{\perp}} (-1)^{xy} |y\rangle$

  $\downarrow$ measure

- random $y \in u^{\perp}$

- $\ell = O(n)$ iteration gives $y_1, \ldots, y_\ell$:
  $\{y_1, \ldots, y_\ell\}^{\perp} = \{0, u\}$ (probably).

- Remarks:

Simon's algorithm 2

- $\sum_{y \in u^\perp} (-1)^{xy} |y\rangle$

  $\downarrow$ measure
- random $y \in u^\perp$

- $\ell = O(n)$ iteration gives $y_1, \ldots, y_\ell$:
  $\{y_1, \ldots, y_\ell\}^\perp = \{0, u\}$ (probably).

- Remarks:
  - measurements only for simplification

Simon's algorithm 2

- $\sum_{y \in u^\perp} (-1)^{xy} |y\rangle$
  $\downarrow$ measure
- random $y \in u^\perp$
- $\ell = O(n)$ iteration gives $y_1, \ldots, y_\ell$:
  $\{y_1, \ldots, y_\ell\}^\perp = \{0, u\}$ (probably).
- Remarks:
  - measurements only for simplification
  - $\exists$ exact method with $O(n)$ rounds (Høyer 97)

Simon's algorithm 2

- $\sum_{y \in u^\perp} (-1)^{xy} |y\rangle$

  $\downarrow$ measure

- random $y \in u^\perp$

- $\ell = O(n)$ iteration gives $y_1, \ldots, y_\ell$:
  $\{y_1, \ldots, y_\ell\}^\perp = \{0, u\}$ (probably).

- Remarks:
  - measurements only for simplification
  - $\exists$ exact method with $O(n)$ rounds (Høyer 97)
    uses Grover's techniques

Contents

QFT mod 2^ℓ

- Fourier transform mod 2^ℓ:

$$\Phi_{2^\ell} : |j\rangle \mapsto \sum_{k=0}^{2^\ell-1} \omega^{kj}|k\rangle, \qquad \text{where } \omega = \sqrt[2^\ell]{1} \; (= e^{\frac{2\pi i}{2^\ell}})$$

QFT mod 2^ℓ

- Fourier transform mod 2^ℓ:

$$\Phi_{2^\ell} : |j\rangle \mapsto \sum_{k=0}^{2^\ell-1} \omega^{kj}|k\rangle, \qquad \text{where } \omega = \sqrt[2^\ell]{1} \ \left(= e^{\frac{2\pi i}{2^\ell}}\right)$$

- Transformation of basis change:

QFT mod 2^ℓ

- Fourier transform mod 2^ℓ:

$$\Phi_{2^\ell} : |j\rangle \mapsto \sum_{k=0}^{2^\ell - 1} \omega^{kj} |k\rangle, \qquad \text{where } \omega = \sqrt[2^\ell]{1} \ \left(= e^{\frac{2\pi i}{2^\ell}}\right)$$

- Transformation of basis change:

  standard basis $\rightarrow$ eigenvectors of shift mod 2^n.

QFT mod 2^ℓ

- Fourier transform mod 2^ℓ:

$$\Phi_{2^\ell} : |j\rangle \mapsto \sum_{k=0}^{2^\ell-1} \omega^{kj}|k\rangle, \qquad \text{where } \omega = \sqrt[2^\ell]{1} \;\left(= e^{\frac{2\pi i}{2^\ell}}\right)$$

- Transformation of basis change:

  standard basis $\rightarrow$ eigenvectors of shift mod 2^n.
- Spec. case $\ell = 1$: Hadamard-gate

QFT mod 2^ℓ

- Fourier transform mod 2^ℓ:

$$\Phi_{2^\ell} : |j\rangle \mapsto \sum_{k=0}^{2^\ell-1} \omega^{kj}|k\rangle, \qquad \text{where } \omega = \sqrt[2^\ell]{1} \;\; (= e^{\frac{2\pi i}{2^\ell}})$$

- Transformation of basis change:

  standard basis $\longrightarrow$ eigenvectors of shift mod 2^n.
- Spec. case $\ell = 1$: Hadamard-gate
- qbits of j:

$$|j\rangle = |j_{\ell-1}\rangle|j_{\ell-2}\rangle \ldots |j_1\rangle|j_0\rangle,$$

QFT mod 2^ℓ

- Fourier transform mod 2^ℓ:

$$\Phi_{2^\ell} : |j\rangle \mapsto \sum_{k=0}^{2^\ell-1} \omega^{kj}|k\rangle, \qquad \text{where } \omega = \sqrt[2^\ell]{1} \;\; (= e^{\frac{2\pi i}{2^\ell}})$$

- Transformation of basis change:

  standard basis $\longrightarrow$ eigenvectors of shift mod 2^n.
- Spec. case $\ell = 1$: Hadamard-gate
- qbits of j:

$$|j\rangle = |j_{\ell-1}\rangle|j_{\ell-2}\rangle \ldots |j_1\rangle|j_0\rangle,$$

  where

$$j = j_0 + 2j_1 + \ldots + 2^{\ell-1}j_{\ell-1},$$

QFT mod 2^ℓ

- Fourier transform mod 2^ℓ:

$$\Phi_{2^\ell} : |j\rangle \mapsto \sum_{k=0}^{2^\ell-1} \omega^{kj}|k\rangle, \qquad \text{where } \omega = \sqrt[2^\ell]{1} \ (= e^{\frac{2\pi i}{2^\ell}})$$

- Transformation of basis change:

  standard basis $\rightarrow$ eigenvectors of shift mod 2^n.
- Spec. case $\ell = 1$: Hadamard-gate
- qbits of j:

$$|j\rangle = |j_{\ell-1}\rangle|j_{\ell-2}\rangle \dots |j_1\rangle|j_0\rangle,$$

  where

$$j = j_0 + 2j_1 + \dots + 2^{\ell-1}j_{\ell-1},$$

- induction:

$$|j\rangle = |[j/2]\rangle|j_0\rangle$$

QFT mod 2^ℓ, part 2

$$\Phi_{2^\ell}|j\rangle = \sum_{k'=0}^{2^\ell-1} \omega^{k'j}|k'\rangle$$

QFT mod 2^ℓ, part 2

$$\Phi_{2^\ell}|j\rangle = \sum_{k'=0}^{2^\ell-1} \omega^{k'j}|k'\rangle$$

$$= \sum_{k=0}^{2^{\ell-1}-1} \omega^{2kj}|2k\rangle + \sum_{k=0}^{2^{\ell-1}-1} \omega^{(2k+1)j}|2k+1\rangle$$

QFT mod 2^ℓ, part 2

$$\Phi_{2^\ell}|j\rangle = \sum_{k'=0}^{2^\ell-1} \omega^{k'j}|k'\rangle$$

$$= \sum_{k=0}^{2^{\ell-1}-1} \omega^{2kj}|2k\rangle + \sum_{k=0}^{2^{\ell-1}-1} \omega^{(2k+1)j}|2k+1\rangle$$

$$= \sum_{k=0}^{2^{\ell-1}-1} \left(\omega^{2kj}|2k\rangle + \omega^{(2k+1)j}|2k+1\rangle\right)$$

QFT mod 2^ℓ, part 2

$$\Phi_{2^\ell}|j\rangle = \sum_{k'=0}^{2^\ell-1} \omega^{k'j}|k'\rangle$$

$$= \sum_{k=0}^{2^{\ell-1}-1} \omega^{2kj}|2k\rangle + \sum_{k=0}^{2^{\ell-1}-1} \omega^{(2k+1)j}|2k+1\rangle$$

$$= \sum_{k=0}^{2^{\ell-1}-1} \left(\omega^{2kj}|2k\rangle + \omega^{(2k+1)j}|2k+1\rangle\right)$$

$$= \sum_{k=0}^{2^{\ell-1}-1} \omega^{2kj} \left(|k\rangle|0\rangle + \omega^{j}|k\rangle|1\rangle\right)$$

QFT mod 2^ℓ, part 2

$$\Phi_{2^\ell}|j\rangle = \sum_{k'=0}^{2^\ell-1} \omega^{k'j}|k'\rangle$$

$$= \sum_{k=0}^{2^{\ell-1}-1} \omega^{2kj}|2k\rangle + \sum_{k=0}^{2^{\ell-1}-1} \omega^{(2k+1)j}|2k+1\rangle$$

$$= \sum_{k=0}^{2^{\ell-1}-1} \left(\omega^{2kj}|2k\rangle + \omega^{(2k+1)j}|2k+1\rangle\right)$$

$$= \sum_{k=0}^{2^{\ell-1}-1} \omega^{2kj} \left(|k\rangle|0\rangle + \omega^{j}|k\rangle|1\rangle\right)$$

$$= \left(\Phi_{2^{\ell-1}}|[j/2]\rangle\right) \otimes \left(|0\rangle + \omega^{j}|1\rangle\right)$$

QFT mod 2^ℓ - simple implementation

- $\Phi_{2^\ell} = \Phi_{2^{\ell-1}}$ and cond. phase shift by ω^j

QFT mod 2^ℓ - simple implementation

- $\Phi_{2^\ell} = \Phi_{2^{\ell-1}}$ and cond. phase shift by ω^j

  for $t = 0$ to $\ell - 1$ if $j_t = 1$ then cond phase shift by

$$\omega^{2^t} = e^{\frac{2\pi i}{2^{\ell-t}}} \ (t = 0, \ldots, \ell - 1)$$

QFT mod 2^ℓ - simple implementation

- $\Phi_{2^\ell} = \Phi_{2^{\ell-1}}$ and cond. phase shift by ω^j
  for $t = 0$ to $\ell - 1$ if $j_t = 1$ then cond phase shift by
  $$\omega^{2^t} = e^{\frac{2\pi i}{2^{\ell-t}}} \; (t = 0, \ldots, \ell - 1)$$
- Procedure:

QFT mod 2^ℓ - simple implementation

- $\Phi_{2^\ell} = \Phi_{2^{\ell-1}}$ and cond. phase shift by ω^j
  for $t = 0$ to $\ell - 1$ if $j_t = 1$ then cond phase shift by

$$\omega^{2^t} = e^{\frac{2\pi i}{2^{\ell-t}}} \; (t = 0, \ldots, \ell - 1)$$

- Procedure:
  - $|j\rangle \left|0^{\ell-1}\right\rangle |0\rangle$

QFT mod 2^ℓ - simple implementation

- $\Phi_{2^\ell} = \Phi_{2^{\ell-1}}$ and cond. phase shift by ω^j
  for $t = 0$ to $\ell - 1$ if $j_t = 1$ then cond phase shift by
  $$\omega^{2^t} = e^{\frac{2\pi i}{2^{\ell-t}}} \; (t = 0, \ldots, \ell - 1)$$

- Procedure:
  - $|j\rangle \left|0^{\ell-1}\right\rangle |0\rangle$
    $\downarrow$ $\qquad$ $\mathrm{QFT}_2^{\ell-1}$ on $|[j/2]\rangle$

QFT mod 2^ℓ - simple implementation

- $\Phi_{2^\ell} = \Phi_{2^{\ell-1}}$ and cond. phase shift by ω^j

  for $t = 0$ to $\ell - 1$ if $j_t = 1$ then cond phase shift by

$$\omega^{2^t} = e^{\frac{2\pi i}{2^{\ell-t}}} \; (t = 0, \ldots, \ell - 1)$$

- Procedure:
  - $|j\rangle \left|0^{\ell-1}\right\rangle |0\rangle$

    $\downarrow$ $\quad$ $\text{QFT}_2^{\ell-1}$ on $|[j/2]\rangle$
  - $|j\rangle \otimes (\Phi_{2^{\ell-1}}|[j/2]\rangle) \otimes |0\rangle$

QFT mod 2^ℓ - simple implementation

- $\Phi_{2^\ell} = \Phi_{2^{\ell-1}}$ and cond. phase shift by ω^j
  for $t = 0$ to $\ell - 1$ if $j_t = 1$ then cond phase shift by

$$\omega^{2^t} = e^{\frac{2\pi i}{2^{\ell-t}}} \ (t = 0, \dots, \ell - 1)$$

- Procedure:
  - $|j\rangle \left|0^{\ell-1}\right\rangle |0\rangle$
    $\downarrow$ QFT$_2^{\ell-1}$ on $|[j/2]\rangle$
  - $|j\rangle \otimes (\Phi_{2^{\ell-1}}|[j/2]\rangle) \otimes |0\rangle$
    $\downarrow$ Hadamard

QFT mod 2^ℓ - simple implementation

- $\Phi_{2^\ell} = \Phi_{2^{\ell-1}}$ and cond. phase shift by ω^j
  for $t = 0$ to $\ell - 1$ if $j_t = 1$ then cond phase shift by

$$\omega^{2^t} = e^{\frac{2\pi i}{2^{\ell-t}}} \; (t = 0, \ldots, \ell - 1)$$

- Procedure:
  - $|j\rangle \left|0^{\ell-1}\right\rangle |0\rangle$

    $\downarrow$ QFT$_2^{\ell-1}$ on $|[j/2]\rangle$
  - $|j\rangle \otimes (\Phi_{2^{\ell-1}}|[j/2]\rangle) \otimes |0\rangle$

    $\downarrow$ Hadamard
  - $|j\rangle \otimes (\Phi_{2^{\ell-1}}|[j/2]\rangle) \otimes (|0\rangle + |1\rangle)$

QFT mod 2^ℓ - simple implementation

- $\Phi_{2^\ell} = \Phi_{2^{\ell-1}}$ and cond. phase shift by ω^j

  for $t = 0$ to $\ell - 1$ if $j_t = 1$ then cond phase shift by

$$\omega^{2^t} = e^{\frac{2\pi i}{2^{\ell-t}}} \; (t = 0, \ldots, \ell - 1)$$

- Procedure:
  - $|j\rangle \left|0^{\ell-1}\right\rangle |0\rangle$

    $\downarrow$ QFT$_2^{\ell-1}$ on $|[j/2]\rangle$
  - $|j\rangle \otimes (\Phi_{2^{\ell-1}}|[j/2]\rangle) \otimes |0\rangle$

    $\downarrow$ Hadamard
  - $|j\rangle \otimes (\Phi_{2^{\ell-1}}|[j/2]\rangle) \otimes (|0\rangle + |1\rangle)$

    $\downarrow$ for($t \in [0, l-1]$)

    if($j_t \neq 0$) then do cond. phase shift

QFT mod 2^ℓ - simple implementation

- $\Phi_{2^\ell} = \Phi_{2^{\ell-1}}$ and cond. phase shift by ω^j

  for $t = 0$ to $\ell - 1$ if $j_t = 1$ then cond phase shift by

  $$\omega^{2^t} = e^{\frac{2\pi i}{2^{\ell-t}}} \ (t = 0, \ldots, \ell - 1)$$

- Procedure:
  - $|j\rangle \left|0^{\ell-1}\right\rangle |0\rangle$

    $\downarrow$ $\quad$ $\mathrm{QFT}_2^{\ell-1}$ on $|[j/2]\rangle$
  - $|j\rangle \otimes (\Phi_{2^{\ell-1}}|[j/2]\rangle) \otimes |0\rangle$

    $\downarrow$ $\quad$ Hadamard
  - $|j\rangle \otimes (\Phi_{2^{\ell-1}}|[j/2]\rangle) \otimes (|0\rangle + |1\rangle)$

    $\downarrow$ $\quad$ for$(t \in [0, l-1])$

    if$(j_t \neq 0)$ then do cond. phase shift
  - $|j\rangle \otimes \Phi_{2^\ell}(|j\rangle)$

QFT mod 2^ℓ - simple implementation 2

- So far: $P : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi(|j\rangle)$

QFT mod 2^ℓ - simple implementation 2

- So far: $P : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi(|j\rangle)$
- $\omega \leftrightarrow \overline{\omega}$: $\overline{P} : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi^{-1}(|j\rangle)$

QFT mod 2^ℓ - simple implementation 2

- So far: $P : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi(|j\rangle)$
- $\omega \leftrightarrow \overline{\omega}$: $\overline{P} : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi^{-1}(|j\rangle)$
- Simple implementation of $|j\rangle \mapsto \otimes\Phi(j)$ (using aux qbits):

QFT mod 2^ℓ - simple implementation 2

- So far: $P : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi(|j\rangle)$
- $\omega \leftrightarrow \overline{\omega}$: $\overline{P} : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi^{-1}(|j\rangle)$
- Simple implementation of $|j\rangle \mapsto \otimes\Phi(j)$ (using aux qbits):
  - $|j\rangle \otimes |0\rangle$

QFT mod 2^ℓ - simple implementation 2

- So far: $P : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi(|j\rangle)$
- $\omega \leftrightarrow \overline{\omega}$: $\overline{P} : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi^{-1}(|j\rangle)$
- Simple implementation of $|j\rangle \mapsto \otimes\Phi(j)$ (using aux qbits):
  - $|j\rangle \otimes |0\rangle$
    $\downarrow$ P

QFT mod 2^ℓ - simple implementation 2

- So far: $P : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi(|j\rangle)$
- $\omega \leftrightarrow \overline{\omega}$: $\overline{P} : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi^{-1}(|j\rangle)$
- Simple implementation of $|j\rangle \mapsto \otimes\Phi(j)$ (using aux qbits):
  - $|j\rangle \otimes |0\rangle$
    $\downarrow$ P
  - $|j\rangle \otimes \Phi(|j\rangle)$

QFT mod 2^ℓ - simple implementation 2

- So far: $P : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi(|j\rangle)$
- $\omega \leftrightarrow \overline{\omega}$: $\overline{P} : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi^{-1}(|j\rangle)$
- Simple implementation of $|j\rangle \mapsto \otimes\Phi(j)$ (using aux qbits):
  - $|j\rangle \otimes |0\rangle$

    $\downarrow$ P
  - $|j\rangle \otimes \Phi(|j\rangle)$

    $\downarrow$ swap

QFT mod 2^ℓ - simple implementation 2

- So far: $P : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi(|j\rangle)$
- $\omega \leftrightarrow \overline{\omega}$: $\overline{P} : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi^{-1}(|j\rangle)$
- Simple implementation of $|j\rangle \mapsto \otimes\Phi(j)$ (using aux qbits):
  - $|j\rangle \otimes |0\rangle$
    $\downarrow$ P
  - $|j\rangle \otimes \Phi(|j\rangle)$
    $\downarrow$ swap
  - $\Phi(|j\rangle) \otimes |j\rangle$

QFT mod 2^ℓ - simple implementation 2

- So far: $P : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi(|j\rangle)$
- $\omega \leftrightarrow \overline{\omega}$: $\overline{P} : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi^{-1}(|j\rangle)$
- Simple implementation of $|j\rangle \mapsto \otimes\Phi(j)$ (using aux qbits):
  - $|j\rangle \otimes |0\rangle$

    $\downarrow$ P
  - $|j\rangle \otimes \Phi(|j\rangle)$

    $\downarrow$ swap
  - $\Phi(|j\rangle) \otimes |j\rangle$

    $\downarrow$ $\overline{P}^{-1}$

QFT mod 2^ℓ - simple implementation 2

- So far: $P : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi(|j\rangle)$
- $\omega \leftrightarrow \overline{\omega}$: $\overline{P} : |j\rangle \otimes |0\rangle \mapsto |j\rangle \otimes \Phi^{-1}(|j\rangle)$
- Simple implementation of $|j\rangle \mapsto \otimes\Phi(j)$ (using aux qbits):
  - $|j\rangle \otimes |0\rangle$
    $\downarrow$ P
  - $|j\rangle \otimes \Phi(|j\rangle)$
    $\downarrow$ swap
  - $\Phi(|j\rangle) \otimes |j\rangle$
    $\downarrow$ $\overline{P}^{-1}$
  - $\Phi(|j\rangle) \otimes |0\rangle$

QFT over 2^{ℓ} - remark

Remark: QFT in the literature

QFT over 2^ℓ - remark

Remark: QFT in the literature

-reorganized in a clever way

QFT over 2^ℓ - remark

Remark: QFT in the literature

-reorganized in a clever way

-more "efficient"

QFT over 2^ℓ - remark

Remark: QFT in the literature

-reorganized in a clever way

-more "efficient"

-has nice circuit description

QFT over 2^ℓ - remark

Remark: QFT in the literature

-reorganized in a clever way

-more "efficient"

-has nice circuit description

-computes $|j\rangle \mapsto \Phi(|j\rangle)$ without auxiliary qbits

QFT over 2^ℓ - remark

Remark: QFT in the literature

-reorganized in a clever way

-more "efficient"

-has nice circuit description

-computes $|j\rangle \mapsto \Phi(|j\rangle)$ without auxiliary qbits

Details in *Cleve, Ekert, Macchiavello, Mosca (1998)*

Phase estimation (eigenvalue estimation)

- Given:

Phase estimation (eigenvalue estimation)

- Given:

  state (vector) ψ, an *eigenvector*

Phase estimation (eigenvalue estimation)

- Given:

  state (vector) ψ, an *eigenvector*

  of the unitary U

Phase estimation (eigenvalue estimation)

- Given:

  state (vector) ψ, an *eigenvector*

  of the unitary U

  oracles for $U, U^2, U^4, \ldots$

Phase estimation (eigenvalue estimation)

- Given:

  state (vector) ψ, an *eigenvector*

  of the unitary U

  oracles for $U, U^2, U^4, \ldots$

- Task: approximate (*phase* of) the eigenvalue

Phase estimation (eigenvalue estimation)

- Given:

  state (vector) ψ, an *eigenvector*

  of the unitary U

  oracles for $U, U^2, U^4, \ldots$

- Task: approximate (*phase* of) the eigenvalue

  $U\psi = e^{\alpha \cdot 2\pi i}\psi$

Phase estimation (eigenvalue estimation)

- Given:

  state (vector) ψ, an *eigenvector*

  of the unitary U

  oracles for $U, U^2, U^4, \ldots$

- Task: approximate (*phase* of) the eigenvalue

  $U\psi = e^{\alpha \cdot 2\pi i}\psi$

  compute the ℓ most significant bits of *phase* α.

Phase estimation (eigenvalue estimation)

- Given:

  state (vector) ψ, an *eigenvector*

  of the unitary U

  oracles for $U, U^2, U^4, \ldots$

- Task: approximate (*phase* of) the eigenvalue

  $$U\psi = e^{\alpha \cdot 2\pi i}\psi$$

  compute the ℓ most significant bits of *phase* α.

- Operation for task:

Phase estimation (eigenvalue estimation)

- Given:

  state (vector) ψ, an *eigenvector*

  of the unitary U

  oracles for $U, U^2, U^4, \ldots$

- Task: approximate (*phase* of) the eigenvalue

  $U\psi = e^{\alpha \cdot 2\pi i}\psi$

  compute the ℓ most significant bits of *phase* α.

- Operation for task:

  $\psi \otimes \left|0^\ell\right\rangle \mapsto \psi \otimes |k\rangle,$

Phase estimation (eigenvalue estimation)

- Given:

  state (vector) ψ, an *eigenvector*

  of the unitary U

  oracles for $U, U^2, U^4, \ldots$

- Task: approximate (*phase* of) the eigenvalue

  $U\psi = e^{\alpha \cdot 2\pi i}\psi$

  compute the ℓ most significant bits of *phase* α.

- Operation for task:

  $$\psi \otimes \left|0^\ell\right\rangle \mapsto \psi \otimes |k\rangle,$$

  where

  $$\left|\alpha - \frac{k}{2^\ell}\right| \leq \frac{1}{2^{\ell+1}}.$$

Phase estimation - algorithm idea

- Assume $\alpha = \frac{k}{2^\ell}$

Phase estimation - algorithm idea

- Assume $\alpha = \frac{k}{2^\ell}$
- Then

$$\sum_{j=0}^{\ell-1} U^j \psi \otimes |j\rangle = \psi \otimes \sum_{j=0}^{\ell-1} e^{\alpha \cdot 2\pi i j} |j\rangle$$

Phase estimation - algorithm idea

- Assume $\alpha = \frac{k}{2^\ell}$
- Then

$$\sum_{j=0}^{\ell-1} U^j \psi \otimes |j\rangle = \psi \otimes \sum_{j=0}^{\ell-1} e^{\alpha \cdot 2\pi i j} |j\rangle$$

$$= \psi \otimes \sum_{j=0}^{\ell-1} e^{\frac{2\pi i}{\ell} \cdot kj} |j\rangle$$

Phase estimation - algorithm idea

- Assume $\alpha = \frac{k}{2^\ell}$
- Then

$$\sum_{j=0}^{\ell-1} U^j \psi \otimes |j\rangle = \psi \otimes \sum_{j=0}^{\ell-1} e^{\alpha \cdot 2\pi i j} |j\rangle$$

$$= \psi \otimes \sum_{j=0}^{\ell-1} e^{\frac{2\pi i}{\ell} \cdot kj} |j\rangle$$

$$= \psi \otimes \Phi_{2^\ell}(|k\rangle)$$

Phase estimation - algorithm idea

- Assume $\alpha = \frac{k}{2^\ell}$
- Then

$$\sum_{j=0}^{\ell-1} U^j \psi \otimes |j\rangle = \psi \otimes \sum_{j=0}^{\ell-1} e^{\alpha \cdot 2\pi i j} |j\rangle$$

$$= \psi \otimes \sum_{j=0}^{\ell-1} e^{\frac{2\pi i}{\ell} \cdot kj} |j\rangle$$

$$= \psi \otimes \Phi_{2^\ell}(|k\rangle)$$

- Apply $\Phi_{2^\ell}^{-1}$, obtain

Phase estimation - algorithm idea

- Assume $\alpha = \frac{k}{2^\ell}$
- Then

$$\sum_{j=0}^{\ell-1} U^j \psi \otimes |j\rangle = \psi \otimes \sum_{j=0}^{\ell-1} e^{\alpha \cdot 2\pi i j} |j\rangle$$

$$= \psi \otimes \sum_{j=0}^{\ell-1} e^{\frac{2\pi i}{\ell} \cdot kj} |j\rangle$$

$$= \psi \otimes \Phi_{2^\ell}(|k\rangle)$$

- Apply $\Phi_{2^\ell}^{-1}$, obtain

$$\psi \otimes |k\rangle$$

Phase estimation - the algorithm

$$r = O(\log \frac{1}{\epsilon})$$

Phase estimation - the algorithm

$r = O(\log \frac{1}{\epsilon}))$

Init: $\psi \otimes |0\rangle \mapsto \psi \otimes \sum_{j=0}^{\ell+r-1} |j\rangle$

Phase estimation - the algorithm

$r = O(\log \frac{1}{\epsilon}))$

Init: $\psi \otimes |0\rangle \mapsto \psi \otimes \sum_{j=0}^{\ell+r-1} |j\rangle$

for $(d = 0, d < \ell + r, d := d + 1)$:

Phase estimation - the algorithm

$r = O(\log \frac{1}{\epsilon}))$

Init: $\psi \otimes |0\rangle \mapsto \psi \otimes \sum_{j=0}^{\ell+r-1} |j\rangle$

for $(d = 0, d < \ell + r, d := d + 1)$:

if $(j_d = 1)$:

Phase estimation - the algorithm

$r = O(\log \frac{1}{\epsilon}))$

Init: $\psi \otimes |0\rangle \mapsto \psi \otimes \sum_{j=0}^{\ell+r-1} |j\rangle$

for $(d = 0, d < \ell + r, d := d + 1)$:

if $(j_d = 1)$:

apply $U^{2^{d-1}}$ to ψ

Phase estimation - the algorithm

$r = O(\log \frac{1}{\epsilon}))$

Init: $\psi \otimes |0\rangle \mapsto \psi \otimes \sum_{j=0}^{\ell+r-1} |j\rangle$

for $(d = 0, d < \ell + r, d := d + 1)$:

if $(j_d = 1)$:

apply $U^{2^{d-1}}$ to ψ

Last step: apply $\Phi_{2^{\ell+r}}^{-1}$ to the second part

Phase estimation - the algorithm

$r = O(\log \frac{1}{\epsilon}))$

Init: $\psi \otimes |0\rangle \mapsto \psi \otimes \sum_{j=0}^{\ell+r-1} |j\rangle$

for $(d = 0, d < \ell + r, d := d + 1)$:

if $(j_d = 1)$:

apply $U^{2^{d-1}}$ to ψ

Last step: apply $\Phi_{2^{\ell+r}}^{-1}$ to the second part

Return the first ℓ bits

Phase estimation - the algorithm

$r = O(\log \frac{1}{\epsilon}))$

Init: $\psi \otimes |0\rangle \mapsto \psi \otimes \sum_{j=0}^{\ell+r-1} |j\rangle$

for $(d = 0, d < \ell + r, d := d + 1)$:

if $(j_d = 1)$:

apply $U^{2^{d-1}}$ to ψ

Last step: apply $\Phi_{2^{\ell+r}}^{-1}$ to the second part

Return the first ℓ bits

State before Φ^{-1}:

$$\text{State} = \sum_{j=0}^{\ell+r-1} U^j \psi \otimes |j\rangle = \psi \otimes \sum_{j=0}^{\ell+r-1} e^{\alpha \cdot 2\pi ij} |j\rangle$$

Phase estimation - the algorithm

$r = O(\log \frac{1}{\epsilon}))$

Init: $\psi \otimes |0\rangle \mapsto \psi \otimes \sum_{j=0}^{\ell+r-1} |j\rangle$

for $(d = 0, d < \ell + r, d := d + 1)$:

if $(j_d = 1)$:

apply $U^{2^{d-1}}$ to ψ

Last step: apply $\Phi^{-1}_{2^{\ell+r}}$ to the second part

Return the first ℓ bits

State before Φ^{-1}:

$$\text{State} = \sum_{j=0}^{\ell+r-1} U^j \psi \otimes |j\rangle = \psi \otimes \sum_{j=0}^{\ell+r-1} e^{\alpha \cdot 2\pi i j} |j\rangle$$

If $\alpha = \frac{k}{2^{\ell+r}}$:

$$\text{State} = \psi \otimes \Phi(|k\rangle)$$

Phase estimation - algorithm analysis

- Before inverse QFT (if $\alpha = \frac{k}{2^{\ell+r}}$)

$$\text{State} = \psi \otimes \Phi(|k\rangle)$$

Phase estimation - algorithm analysis

- Before inverse QFT (if $\alpha = \frac{k}{2^{\ell+r}}$)

$$\text{State} = \psi \otimes \Phi(|k\rangle)$$

- After inverse QFT

Phase estimation - algorithm analysis

- Before inverse QFT (if $\alpha = \frac{k}{2^{\ell+r}}$)

$$\text{State} = \psi \otimes \Phi(|k\rangle)$$

- After inverse QFT

  if $\alpha = \frac{k}{2^{\ell+r}}$:

$$\text{State} = \psi \otimes |k\rangle$$

Phase estimation - algorithm analysis

- Before inverse QFT (if $\alpha = \frac{k}{2^{\ell+r}}$)

$$\text{State} = \psi \otimes \Phi(|k\rangle)$$

- After inverse QFT

  if $\alpha = \frac{k}{2^{\ell+r}}$:

$$\text{State} = \psi \otimes |k\rangle$$

  if $\alpha \approx \frac{k}{2^{\ell+r}}$:

$$\text{State} \approx \psi \sum_{|k'-k|<2^r} c_{k'} |k'\rangle.$$

Phase estimation - algorithm analysis

- Before inverse QFT (if $\alpha = \frac{k}{2^{\ell+r}}$)

$$\text{State} = \psi \otimes \Phi(|k\rangle)$$

- After inverse QFT

  if $\alpha = \frac{k}{2^{\ell+r}}$:

$$\text{State} = \psi \otimes |k\rangle$$

  if $\alpha \approx \frac{k}{2^{\ell+r}}$:

$$\text{State} \approx \psi \sum_{|k'-k|<2^r} c_{k'} |k'\rangle.$$

- Details: *In: e.g., Cleve, Ekert, Macchiavello, Mosca (1998).*

Period finding

- Given: $f : \mathbb{Z} \to \{\text{strings}\}$

Period finding

- Given: $f : \mathbb{Z} \to \{\text{strings}\}$
  by oracle $U_f : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$

Period finding

- Given: $f : \mathbb{Z} \to \{\text{strings}\}$

  by oracle $U_f : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$

  spec. case: $x \mapsto f(x)$ by classical algorithm

Period finding

- Given: $f : \mathbb{Z} \rightarrow \{\text{strings}\}$

  by oracle $U_f : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$

  spec. case: $x \mapsto f(x)$ by classical algorithm
- Promise: $f(x) = f(y) \Leftrightarrow x \equiv y \pmod{r}$

Period finding

- Given: $f : \mathbb{Z} \to \{\text{strings}\}$

  by oracle $U_f : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$

  spec. case: $x \mapsto f(x)$ by classical algorithm
- Promise: $f(x) = f(y) \Leftrightarrow x \equiv y \pmod r$
- Task: find r

Period finding

- Given: $f : \mathbb{Z} \rightarrow \{\text{strings}\}$

  by oracle $U_f : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$

  spec. case: $x \mapsto f(x)$ by classical algorithm
- Promise: $f(x) = f(y) \Leftrightarrow x \equiv y \pmod{r}$
- Task: find r
- Gadget: quantum graph (diagram) of f:

$$|f_N\rangle = \sum_{x=0}^{N-1} |x\rangle|f(x)\rangle$$

Period finding 2

Computing $|f_N\rangle$ for $N = 2^\ell$:

Period finding 2

Computing $|f_N\rangle$ for $N = 2^\ell$:

- $|0\rangle|0\rangle$

Period finding 2

Computing $|f_N\rangle$ for $N = 2^\ell$:

- $|0\rangle|0\rangle$

$\downarrow$ Hadamard$^{\otimes \ell}$

Period finding 2

Computing $|f_N\rangle$ for $N = 2^\ell$:

- $|0\rangle|0\rangle$

$\downarrow$ Hadamard$^{\otimes \ell}$

- $\sum_{x=0}^{N-1} |x\rangle|0\rangle$

Period finding 2

Computing $|f_N\rangle$ for $N = 2^\ell$:

- $|0\rangle|0\rangle$

$\downarrow$ Hadamard$^{\otimes \ell}$

- $\sum_{x=0}^{N-1} |x\rangle|0\rangle$

$\downarrow$ U_f

Period finding 2

Computing $|f_N\rangle$ for $N = 2^\ell$:

- $|0\rangle|0\rangle$

  $\downarrow$ Hadamard$^{\otimes\ell}$

- $\sum_{x=0}^{N-1}|x\rangle|0\rangle$

  $\downarrow$ U_f

- $\sum_{x=0}^{N-1}|x\rangle|f(x)\rangle = |f_N\rangle$

Period finding 3

Decomposition of $|f_N\rangle$:

$$|f_N\rangle = \sum_{x=0}^{N-1} |x\rangle|f(x)\rangle \approx \sum_{y=0}^{r-1} \left(\sum_{z=0}^{[\frac{N}{r}]-1} |rz+y\rangle \right) |f(y)\rangle$$

Period finding 3

Decomposition of $|f_N\rangle$:

$$|f_N\rangle = \sum_{x=0}^{N-1} |x\rangle|f(x)\rangle \approx \sum_{y=0}^{r-1} \left(\sum_{z=0}^{[\frac{N}{r}]-1} |rz+y\rangle \right) |f(y)\rangle$$

- Measure $f(y)$ (i.e., take term for fixed y):

$$\sum_{z=0}^{[\frac{N}{r}]-1} |rz+y\rangle$$

Period finding 3

Decomposition of $|f_N\rangle$:

$$|f_N\rangle = \sum_{x=0}^{N-1} |x\rangle|f(x)\rangle \approx \sum_{y=0}^{r-1} \left(\sum_{z=0}^{[\frac{N}{r}]-1} |rz+y\rangle \right) |f(y)\rangle$$

- Measure $f(y)$ (i.e., take term for fixed y):

$$\sum_{z=0}^{[\frac{N}{r}]-1} |rz+y\rangle$$

- Decompose into eigenvectors of shift mod $r[\frac{N}{r}]$ (QFT "in mind"):

$$\sum_{j=0}^{r[\frac{N}{r}]-1} c_{yj} u_j, \qquad \text{where } u_j = \sum_{k=0}^{r[\frac{N}{r}]-1} \omega^{-kj}|k\rangle.$$

Period finding 4

Up to normalization:

Period finding 4

Up to normalization:

$$c_{yj} = \sum_{z=0}^{[\frac{N}{r}]-1} \omega^{j(rz+y)} = \omega^{jy} \sum_{z=0}^{[\frac{N}{r}]-1} \omega^{jrz} =$$

Period finding 4

Up to normalization:

$$c_{yj} = \sum_{z=0}^{[\frac{N}{r}]-1} \omega^{j(rz+y)} = \omega^{jy} \sum_{z=0}^{[\frac{N}{r}]-1} \omega^{jrz} =$$

$$\omega^{jy} \begin{cases} [\frac{N}{r}] & \text{if } j \equiv 0 \pmod{[\frac{N}{r}]} \\ 0 & \text{otherwise} \end{cases}$$

where $\omega = \sqrt[r[N/r]]{1}$.

Period finding 5

- Have state

$$\sum_{j=0}^{r[\frac{N}{r}]-1} c_{yj} u_j$$

Period finding 5

- Have state

$$\sum_{j=0}^{r[\frac{N}{r}]-1} c_{yj} u_j$$

  - $c_{yj} = 0$ if j is not a multiple of $[\frac{N}{r}]$,

Period finding 5

- Have state

$$\sum_{j=0}^{r[\frac{N}{r}]-1} c_{yj} u_j$$

  - $c_{yj} = 0$ if j is not a multiple of $[\frac{N}{r}]$,
  - $|c_{yj}|$ the same for $j = \ell[\frac{N}{r}]$, $\ell = 0, \ldots, r-1$.

Period finding 5

- Have state

$$\sum_{j=0}^{r[\frac{N}{r}]-1} c_{yj} u_j$$

  - $c_{yj} = 0$ if j is not a multiple of $[\frac{N}{r}]$,
  - $|c_{yj}|$ the same for $j = \ell[\frac{N}{r}]$, $\ell = 0, \ldots, r-1$.
- State:

$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]}$$

Period finding 5

- Have state

$$\sum_{j=0}^{r[\frac{N}{r}]-1} c_{yj} u_j$$

  - $c_{yj} = 0$ if j is not a multiple of $[\frac{N}{r}]$,
  - $|c_{yj}|$ the same for $j = \ell[\frac{N}{r}]$, $\ell = 0, \ldots, r-1$.
- State:

$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]}$$

- comb. of eigenvectors of shift with

Period finding 5

- Have state

$$\sum_{j=0}^{r[\frac{N}{r}]-1} c_{yj} u_j$$

  - $c_{yj} = 0$ if j is not a multiple of $[\frac{N}{r}]$,
  - $|c_{yj}|$ the same for $j = \ell[\frac{N}{r}]$, $\ell = 0, \ldots, r-1$.
- State:

$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]}$$

- comb. of eigenvectors of shift with eigenvalues

$$\omega^{\ell[N/r]} = (\sqrt[r[N/r]]{1})^{\ell[N/r]} = (\sqrt[r]{1})^{\ell}$$

Period finding 6

- Have state

$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]}$$

Period finding 6

- Have state
$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]}$$
- apply phase estimation

Period finding 6

- Have state

$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]}$$

- apply phase estimation

$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]} |\ell/r\rangle$$

Period finding 6

- Have state

$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]}$$

- apply phase estimation

$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]} |\ell/r\rangle$$

- continued fraction approx. gives r

Period finding 6

- Have state
$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]}$$
- apply phase estimation
$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]} |\ell/r\rangle$$
- continued fraction approx. gives r
  if $gcd(\ell, r) = 1$.

Period finding 6

- Have state
$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]}$$
- apply phase estimation
$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]} |\ell/r\rangle$$
- continued fraction approx. gives r
  if $gcd(\ell, r) = 1$.
- Details: *In: Cleve, Ekert, Macchiavello, Mosca (1998).*

Period finding 6

- Have state

$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]}$$

- apply phase estimation

$$\sum_{\ell=0}^{r-1} c_{y\ell[N/r]} u_{[\ell[N/r]]} |\ell/r\rangle$$

- continued fraction approx. gives r

  if $gcd(\ell, r) = 1$.

- Details: *In: Cleve, Ekert, Macchiavello, Mosca (1998).*
- Original: *Shor 1994*

QFT modulo m

- $|j\rangle|0\rangle$

QFT modulo m

- $|j\rangle|0\rangle$

  $\downarrow \qquad \approx \text{Hadamard}^{\otimes \log_m} + \ldots$

QFT modulo m

- $|j\rangle|0\rangle$

  $\downarrow$ $\qquad \approx \text{Hadamard}^{\otimes \log_m} + \ldots$

- $|j\rangle\sum_{k=0}^{m-1}|k\rangle$

QFT modulo m

- $|j\rangle|0\rangle$

  $\downarrow$ $\approx$ Hadamard$^{\otimes \log_m} + \ldots$

- $|j\rangle \sum_{k=0}^{m-1} |k\rangle$

  dyadic approx of jk/m in *aux*

  $\downarrow$ cond. phase shifts, bitwise

  uncompute *aux*

QFT modulo m

- $|j\rangle|0\rangle$

  $\downarrow$ $\approx$ Hadamard$^{\otimes \log_m} + \ldots$

- $|j\rangle \sum_{k=0}^{m-1} |k\rangle$

  $\downarrow$ dyadic approx of jk/m in *aux*
  cond. phase shifts, bitwise
  uncompute *aux*

- $\approx |j\rangle \sum_{k=0}^{m-1} \omega^{jk}|k\rangle$

QFT modulo m

- $|j\rangle|0\rangle$

  $\downarrow$ $\approx \text{Hadamard}^{\otimes \log_m} + \ldots$

- $|j\rangle \sum_{k=0}^{m-1} |k\rangle$

  dyadic approx of jk/m in *aux*

  $\downarrow$ cond. phase shifts, bitwise

  uncompute *aux*

- $\approx |j\rangle \sum_{k=0}^{m-1} \omega^{jk}|k\rangle$

  $\downarrow$ inverse phase estimation

QFT modulo m

- $|j\rangle|0\rangle$

  $\downarrow$ $\approx$ Hadamard$^{\otimes \log_m} + \ldots$

- $|j\rangle \sum_{k=0}^{m-1} |k\rangle$

  dyadic approx of jk/m in *aux*

  $\downarrow$ cond. phase shifts, bitwise

  uncompute *aux*

- $\approx |j\rangle \sum_{k=0}^{m-1} \omega^{jk}|k\rangle$

  $\downarrow$ inverse phase estimation

- $|0\rangle \sum_{k=0}^{m-1} \omega^{jk}|k\rangle$

QFT modulo m

- $|j\rangle|0\rangle$

  $\downarrow$ $\approx$ Hadamard$^{\otimes \log_m} + \ldots$

- $|j\rangle \sum_{k=0}^{m-1} |k\rangle$

  $\downarrow$ dyadic approx of jk/m in *aux*
  cond. phase shifts, bitwise
  uncompute *aux*

- $\approx |j\rangle \sum_{k=0}^{m-1} \omega^{jk}|k\rangle$

  $\downarrow$ inverse phase estimation

- $|0\rangle \sum_{k=0}^{m-1} \omega^{jk}|k\rangle$

  $\downarrow$ swap

QFT modulo m

- $|j\rangle|0\rangle$

  $\downarrow$ $\approx$ Hadamard$^{\otimes \log_m} + \ldots$

- $|j\rangle \sum_{k=0}^{m-1} |k\rangle$

  $\downarrow$ dyadic approx of jk/m in *aux*
  cond. phase shifts, bitwise
  uncompute *aux*

- $\approx |j\rangle \sum_{k=0}^{m-1} \omega^{jk}|k\rangle$

  $\downarrow$ inverse phase estimation

- $|0\rangle \sum_{k=0}^{m-1} \omega^{jk}|k\rangle$

  $\downarrow$ swap

- $\sum_{k=0}^{m-1} \omega^{jk}|k\rangle|0\rangle$

QFT of $\mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_n}$

- Tensor product of QFT's for $\mathbb{Z}_{m_1}, \ldots, \mathbb{Z}_{m_n}$.

QFT of $\mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_n}$

- Tensor product of QFT's for $\mathbb{Z}_{m_1}, \ldots, \mathbb{Z}_{m_n}$.
- For $\mathbb{Z}_m^n$:

$$|u\rangle \mapsto \sum_{v \in \mathbb{Z}_m^n} \omega^{(u,v)} |v\rangle,$$

QFT of $\mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_n}$

- Tensor product of QFT's for $\mathbb{Z}_{m_1}, \ldots, \mathbb{Z}_{m_n}$.
- For $\mathbb{Z}_m^n$:

$$|u\rangle \mapsto \sum_{v \in \mathbb{Z}_m^n} \omega^{(u,v)} |v\rangle,$$

where $\omega = \sqrt[m]{1}$

QFT of $\mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_n}$

- Tensor product of QFT's for $\mathbb{Z}_{m_1}, \ldots, \mathbb{Z}_{m_n}$.
- For $\mathbb{Z}_m^n$:

$$|u\rangle \mapsto \sum_{v \in \mathbb{Z}_m^n} \omega^{(u,v)} |v\rangle,$$

where $\omega = \sqrt[m]{1}$
and $(u, v) = \sum_{i=1}^{n} u_i v_i \pmod{m}$.

QFT of $\mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_n}$

- Tensor product of QFT's for $\mathbb{Z}_{m_1}, \ldots, \mathbb{Z}_{m_n}$.
- For $\mathbb{Z}_m^n$:

$$|u\rangle \mapsto \sum_{v \in \mathbb{Z}_m^n} \omega^{(u,v)} |v\rangle,$$

  where $\omega = \sqrt[m]{1}$
  and $(u, v) = \sum_{i=1}^n u_i v_i \pmod{m}$.
- In terms of characters:

$$|g\rangle \mapsto \sum_{\chi \in \hat{G}} \chi(g) |\chi\rangle,$$

QFT of $\mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_n}$

- Tensor product of QFT's for $\mathbb{Z}_{m_1}, \ldots, \mathbb{Z}_{m_n}$.
- For $\mathbb{Z}_m^n$:

$$|u\rangle \mapsto \sum_{v \in \mathbb{Z}_m^n} \omega^{(u,v)} |v\rangle,$$

where $\omega = \sqrt[m]{1}$
and $(u, v) = \sum_{i=1}^n u_i v_i \pmod{m}$.

- In terms of characters:

$$|g\rangle \mapsto \sum_{\chi \in \hat{G}} \chi(g) |\chi\rangle,$$

G finite abelian group, $\hat{G} = \mathrm{Hom}(G, \mathbb{C}^*)$

Abelian QFT – interpretations

- Characters

Abelian QFT – interpretations

- Characters
  - $\hat{G} = \mathrm{Hom}(G, \mathbb{C}^*)$

$$|g\rangle \mapsto \sum_{\chi \in \hat{G}} \chi(g) |"\chi"\rangle,$$

Abelian QFT – interpretations

- Characters
  - $\hat{G} = \mathrm{Hom}(G, \mathbb{C}^*)$

$$|g\rangle \mapsto \sum_{\chi \in \hat{G}} \chi(g) |"\chi"\rangle,$$

  - $"\chi"$ string encoding χ

Abelian QFT – interpretations

- Characters
  - $\hat{G} = \mathrm{Hom}(G, \mathbb{C}^*)$

$$|g\rangle \mapsto \sum_{\chi \in \hat{G}} \chi(g) |"\chi"\rangle,$$

  - $"\chi"$ string encoding χ
  - in $\mathbb{Z}_m^n$, v may encode $\chi_v : u \mapsto \omega^{(u,v)}$

Abelian QFT – interpretations

- Characters
  - $\hat{G} = \mathrm{Hom}(G, \mathbb{C}^*)$

$$|g\rangle \mapsto \sum_{\chi \in \hat{G}} \chi(g)|"\chi"\rangle,$$

  - $"\chi"$ string encoding χ
  - in $\mathbb{Z}_m^n$, v may encode $\chi_v : u \mapsto \omega^{(u,v)}$
- Basis change of $\mathbb{C}G$

Abelian QFT – interpretations

- Characters
  - $\hat{G} = \mathrm{Hom}(G, \mathbb{C}^*)$

$$|g\rangle \mapsto \sum_{\chi \in \hat{G}} \chi(g) |''\chi''\rangle,$$

  - "χ" string encoding χ
  - in $\mathbb{Z}_m^n$, v may encode $\chi_v : u \mapsto \omega^{(u,v)}$
- Basis change of $\mathbb{C}G$
  - standard basis: $|g\rangle$, $g \in G$

Abelian QFT – interpretations

- Characters
  - $\hat{G} = \mathrm{Hom}(G, \mathbb{C}^*)$

$$|g\rangle \mapsto \sum_{\chi \in \hat{G}} \chi(g)|"\chi"\rangle,$$

  - $"\chi"$ string encoding χ
  - in $\mathbb{Z}_m^n$, v may encode $\chi_v : u \mapsto \omega^{(u,v)}$
- Basis change of $\mathbb{C}G$
  - standard basis: $|g\rangle$, $g \in G$
  - $|\chi\rangle \in \mathbb{C}G$ common eigenvector:

$$g|\chi\rangle = \chi(g)|\chi\rangle$$

Abelian QFT – interpretations

- Characters
  - $\hat{G} = \text{Hom}(G, \mathbb{C}^*)$

$$|g\rangle \mapsto \sum_{\chi \in \hat{G}} \chi(g)|"\chi"\rangle,$$

  - "χ" string encoding χ
  - in $\mathbb{Z}_m^n$, v may encode $\chi_v : u \mapsto \omega^{(u,v)}$
- Basis change of $\mathbb{C}G$
  - standard basis: $|g\rangle$, $g \in G$
  - $|\chi\rangle \in \mathbb{C}G$ common eigenvector:

$$g|\chi\rangle = \chi(g)|\chi\rangle$$

$$|\chi\rangle = \sum_{g \in G} \overline{\chi}(g)|g\rangle$$

Contents

HSP - the hidden subgroup problem

- G (finite) group

HSP - the hidden subgroup problem

- G (finite) group
- $f : G \to \{\text{objects}\}$ **hides** the subgroup $H \leq G$, if

HSP - the hidden subgroup problem

- G (finite) group
- $f : G \longrightarrow \{\text{objects}\}$ **hides** the subgroup $H \leq G$, if

$$f(x) = f(y) \Leftrightarrow xH = yH$$

x and y are in the same left coset of H

HSP - the hidden subgroup problem

- G (finite) group
- $f : G \longrightarrow \{\text{objects}\}$ **hides** the subgroup $H \leq G$, if

  $f(x) = f(y) \Leftrightarrow xH = yH$

  x and y are in the same left coset of H

  f is constant on the left cosets of H
  and takes different values on different cosets

HSP - the hidden subgroup problem

- G (finite) group
- $f : G \longrightarrow \{\text{objects}\}$ **hides** the subgroup $H \leq G$, if

  $$f(x) = f(y) \Leftrightarrow xH = yH$$

  x and y are in the same left coset of H

  f is constant on the left cosets of H
  and takes different values on different cosets

- f given by an oracle (or an efficient algorithm) performing

  $$|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$$

HSP - the hidden subgroup problem

- G (finite) group
- $f : G \longrightarrow \{\text{objects}\}$ **hides** the subgroup $H \leq G$, if

  $f(x) = f(y) \Leftrightarrow xH = yH$

  x and y are in the same left coset of H

  f is constant on the left cosets of H
  and takes different values on different cosets

- f given by an oracle (or an efficient algorithm) performing

  $|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$

  often: classical algorithm $x \mapsto f(x)$

HSP - the hidden subgroup problem

- G (finite) group
- $f : G \longrightarrow \{\text{objects}\}$ **hides** the subgroup $H \leq G$, if

  $f(x) = f(y) \Leftrightarrow xH = yH$

  x and y are in the same left coset of H

  f is constant on the left cosets of H
  and takes different values on different cosets

- f given by an oracle (or an efficient algorithm) performing

  $|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$

  often: classical algorithm $x \mapsto f(x)$

- Task: find (generators for) H.

HSP - the hidden subgroup problem

- G (finite) group
- $f : G \longrightarrow \{\text{objects}\}$ **hides** the subgroup $H \leq G$, if

  $$f(x) = f(y) \Leftrightarrow xH = yH$$

  x and y are in the same left coset of H

  f is constant on the left cosets of H
  and takes different values on different cosets
- f given by an oracle (or an efficient algorithm) performing

  $$|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$$

  often: classical algorithm $x \mapsto f(x)$
- Task: find (generators for) H.
- Examples:

HSP - the hidden subgroup problem

- G (finite) group
- $f : G \longrightarrow \{\text{objects}\}$ **hides** the subgroup $H \leq G$, if

  $f(x) = f(y) \Leftrightarrow xH = yH$

  x and y are in the same left coset of H

  f is constant on the left cosets of H
  and takes different values on different cosets

- f given by an oracle (or an efficient algorithm) performing

  $|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$

  often: classical algorithm $x \mapsto f(x)$

- Task: find (generators for) H.
- Examples:

  Period $G = \mathbb{Z}$, f r-periodical, $H = r\mathbb{Z}$.

HSP - the hidden subgroup problem

- G (finite) group
- $f : G \to \{\text{objects}\}$ **hides** the subgroup $H \leq G$, if

  $f(x) = f(y) \Leftrightarrow xH = yH$

  x and y are in the same left coset of H

  f is constant on the left cosets of H
  and takes different values on different cosets

- f given by an oracle (or an efficient algorithm) performing

  $$|x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$$

  often: classical algorithm $x \mapsto f(x)$

- Task: find (generators for) H.
- Examples:

  Period $G = \mathbb{Z}$, f r-periodical, $H = r\mathbb{Z}$.

  Discrete log $G = Z_n \times Z_n$, $f(k, \ell) = u^k v^{-\ell}$,
  $H = \{(k, \ell) | u^k = v^\ell\}$.

Graph automorphism

- permuted graph

Graph automorphism

- permuted graph
  - Γ graph on $\{1, \ldots, n\}$, $\sigma \in S_n$,

Graph automorphism

- permuted graph
  - Γ graph on $\{1, \ldots, n\}$, $\sigma \in S_n$,
  - permuted graph Γ^σ, with edges:

Graph automorphism

- permuted graph
  - Γ graph on $\{1, \ldots, n\}$, $\sigma \in S_n$,
  - permuted graph Γ^σ, with edges: $(\sigma(i), \sigma(j))$ where (i, j) edge of Γ.

Graph automorphism

- permuted graph
  - Γ graph on $\{1,\ldots,n\}$, $\sigma \in S_n$,
  - permuted graph Γ^σ, with edges: $(\sigma(i),\sigma(j))$ where (i,j) edge of Γ.
- Graph automorphism as HSP

Graph automorphism

- permuted graph
  - Γ graph on $\{1,\ldots,n\}$, $\sigma \in S_n$,
  - permuted graph Γ^σ, with edges: $(\sigma(i),\sigma(j))$ where (i,j) edge of Γ.
- Graph automorphism as HSP

$$G = S_n \;\; f(\sigma) = \Gamma^\sigma.$$

Graph automorphism

- permuted graph
  - Γ graph on $\{1, \ldots, n\}$, $\sigma \in S_n$,
  - permuted graph Γ^σ, with edges: $(\sigma(i), \sigma(j))$ where (i, j) edge of Γ.
- Graph automorphism as HSP

  $G = S_n$ $f(\sigma) = \Gamma^\sigma$.

  hidden subgroup $= Aut(G)$

Graph automorphism

- permuted graph
  - Γ graph on $\{1, \ldots, n\}$, $\sigma \in S_n$,
  - permuted graph Γ^σ, with edges: $(\sigma(i), \sigma(j))$ where (i, j) edge of Γ.
- Graph automorphism as HSP

  $G = S_n$ $f(\sigma) = \Gamma^\sigma$.

  hidden subgroup $= Aut(G)$

  In general: stabilizers in large permutation actions

Graph automorphism

- permuted graph
  - Γ graph on $\{1, \ldots, n\}$, $\sigma \in S_n$,
  - permuted graph Γ^σ, with edges: $(\sigma(i), \sigma(j))$ where (i, j) edge of Γ.
- Graph automorphism as HSP

  $G = S_n$ $f(\sigma) = \Gamma^\sigma$.
  hidden subgroup $= Aut(G)$
  In general: stabilizers in large permutation actions
- Graph iso $\leftarrow$ Graph auto

Graph automorphism

- permuted graph
  - Γ graph on $\{1, \ldots, n\}$, $\sigma \in S_n$,
  - permuted graph Γ^σ, with edges: $(\sigma(i), \sigma(j))$ where (i, j) edge of Γ.
- Graph automorphism as HSP

  $G = S_n$ $f(\sigma) = \Gamma^\sigma$.
  hidden subgroup $= Aut(G)$
  In general: stabilizers in large permutation actions
- Graph iso $\leftarrow$ Graph auto

  Γ_1, Γ_2 connected.

Graph automorphism

- permuted graph
  - Γ graph on $\{1,\ldots,n\}$, $\sigma \in S_n$,
  - permuted graph Γ^σ, with edges: $(\sigma(i),\sigma(j))$ where (i,j) edge of Γ.
- Graph automorphism as HSP

  $G = S_n$ $f(\sigma) = \Gamma^\sigma$.
  hidden subgroup $=$ $Aut(G)$
  In general: stabilizers in large permutation actions
- Graph iso $\leftarrow$ Graph auto

  Γ_1, Γ_2 connected.
  $\Gamma_1 \cong \Gamma_2$ iff

$$\left|\mathrm{Aut}(\Gamma_1 \dot{\bigcup} \Gamma_2)\right| = 2 \cdot |\mathrm{Aut}(\Gamma_1)| \cdot |\mathrm{Aut}(\Gamma_2)|$$

.

Coset states

- $|1_G\rangle|0\rangle$

Coset states

- $|1_G\rangle|0\rangle$

  $\downarrow$ (usually easy)

Coset states

- $|1_G\rangle|0\rangle$

  $\downarrow$ (usually easy)

- $\sum_{x\in G}|x\rangle|0\rangle$

Coset states

- $|1_G\rangle|0\rangle$

  $\downarrow$ (usually easy)

- $\sum_{x\in G}|x\rangle|0\rangle$

  $\downarrow$ f-oracle

Coset states

- $|1_G\rangle|0\rangle$

  $\downarrow$ (usually easy)

- $\sum_{x\in G}|x\rangle|0\rangle$

  $\downarrow$ f-oracle

- $\sum_{x\in G}|x\rangle|f(x)\rangle$

Coset states

- $|1_G\rangle|0\rangle$

  $\downarrow$ (usually easy)

- $\sum_{x\in G}|x\rangle|0\rangle$

  $\downarrow$ f-oracle

- $\sum_{x\in G}|x\rangle|f(x)\rangle$

  $\downarrow$ (equality)

Coset states

- $|1_G\rangle|0\rangle$

  $\downarrow$ (usually easy)

- $\sum_{x\in G}|x\rangle|0\rangle$

  $\downarrow$ f-oracle

- $\sum_{x\in G}|x\rangle|f(x)\rangle$

  $\downarrow$ (equality)

- $\sum_s\sum_{f(x)=s}|x\rangle|s\rangle = \sum_{a\in T}\sum_{x\in H}|ax\rangle|f(a)\rangle$

Coset states

- $|1_G\rangle|0\rangle$

  $\downarrow$ (usually easy)

- $\sum_{x\in G}|x\rangle|0\rangle$

  $\downarrow$ f-oracle

- $\sum_{x\in G}|x\rangle|f(x)\rangle$

  $\downarrow$ (equality)

- $\sum_s \sum_{f(x)=s}|x\rangle|s\rangle = \sum_{a\in T}\sum_{x\in H}|ax\rangle|f(a)\rangle$

T: left transversal of H
$=$ a set of left coset representatives by H

Coset states 2

$$\sum_{a \in T} \sum_{x \in H} |ax\rangle |f(a)\rangle$$

Coset states 2

$\sum_{a \in T} \sum_{x \in H} |ax\rangle|f(a)\rangle$

$\downarrow$ (equality)

Coset states 2

$$\sum_{a\in T}\sum_{x\in H}|ax\rangle|f(a)\rangle$$

$\downarrow$ (equality)

$$\sum_{a\in T}\left(\sum_{x\in H}|ax\rangle\right)|f(a)\rangle$$

Coset states 2

$$\sum_{a\in T}\sum_{x\in H}|ax\rangle|f(a)\rangle$$

$\downarrow$ (equality)

$$\sum_{a\in T}\left(\sum_{x\in H}|ax\rangle\right)|f(a)\rangle$$

$\downarrow$ measure/ignore $f(a)$

Coset states 2

$\sum_{a\in T}\sum_{x\in H}|ax\rangle|f(a)\rangle$

$\downarrow$ (equality)

$\sum_{a\in T}\left(\sum_{x\in H}|ax\rangle\right)|f(a)\rangle$

$\downarrow$ measure/ignore $f(a)$

coset state

$$|aH\rangle := \sum_{x\in H}|ax\rangle \quad \text{for random } a\in T$$

Coset states 2

$\sum_{a\in T}\sum_{x\in H}|ax\rangle|f(a)\rangle$

$\downarrow$ (equality)

$\sum_{a\in T}\left(\sum_{x\in H}|ax\rangle\right)|f(a)\rangle$

$\downarrow$ measure/ignore $f(a)$

coset state

$$|aH\rangle := \sum_{x\in H}|ax\rangle \quad \text{for random } a \in T$$

$\Leftrightarrow$for random $a \in G$

Coset states - summary

Coset state (with random $a \in T$ (random $a \in G$))

$$|aH\rangle = \frac{1}{\sqrt{|H|}} \sum_{x \in H} |ax\rangle$$

(normalizing factor included)

Abelian Fourier sampling

- $\sum_{x \in H} |ax\rangle$

Abelian Fourier sampling

- $\sum_{x \in H} |ax\rangle$

  $\downarrow$ QFT

Abelian Fourier sampling

- $\sum_{x \in H} |ax\rangle$

  $\downarrow$ QFT

- $\sum_{x \in H} \sum_{\chi \in \hat{G}} \chi(ax) |\chi\rangle$

Abelian Fourier sampling

- $\sum_{x \in H} |ax\rangle$

$\downarrow$ QFT

- $\sum_{x \in H} \sum_{\chi \in \hat{G}} \chi(ax)|\chi\rangle$

$\downarrow$ (equality)

Abelian Fourier sampling

- $\sum_{x \in H} |ax\rangle$

  $\downarrow$ QFT

- $\sum_{x \in H} \sum_{\chi \in \hat{G}} \chi(ax)|\chi\rangle$

  $\downarrow$ (equality)

  $\sum_{\chi \in \hat{G}} \left(\chi(a) \sum_{x \in H} \chi(x)\right) |\chi\rangle$

Abelian Fourier sampling

- $\sum_{x\in H} |ax\rangle$

  $\downarrow$ QFT

- $\sum_{x\in H}\sum_{\chi\in\hat{G}} \chi(ax)|\chi\rangle$

  $\downarrow$ (equality)

  $\sum_{\chi\in\hat{G}} \left(\chi(a)\sum_{x\in H}\chi(x)\right)|\chi\rangle$

- with normalizing factors:

$$\sum_{\chi\in\hat{G}} \left(\frac{\chi(a)}{|G|^{\frac{1}{2}}|H|^{\frac{1}{2}}}\sum_{x\in H}\chi(x)\right)|\chi\rangle$$

Abelian Fourier sampling 2

- Coefficient of χ

Abelian Fourier sampling 2

- Coefficient of χ

$$\frac{\chi(a)}{\sqrt{|G:H|}}\frac{1}{|H|}\sum_{x\in H}\chi(x) = \begin{cases} \frac{\chi(a)}{\sqrt{|G:H|}} & \text{if } \chi_H = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Abelian Fourier sampling 2

- Coefficient of χ

$$\frac{\chi(a)}{\sqrt{|G:H|}}\frac{1}{|H|}\sum_{x\in H}\chi(x) = \begin{cases} \frac{\chi(a)}{\sqrt{|G:H|}} & \text{if } \chi_H = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Proof:

Abelian Fourier sampling 2

- Coefficient of χ

$$\frac{\chi(a)}{\sqrt{|G:H|}}\frac{1}{|H|}\sum_{x\in H}\chi(x) \quad = \quad \begin{cases} \frac{\chi(a)}{\sqrt{|G:H|}} & \text{if } \chi_H = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Proof:
orthogonality of 1_H and χ_H

Abelian Fourier sampling 2

- Coefficient of χ

$$\frac{\chi(a)}{\sqrt{|G:H|}}\frac{1}{|H|}\sum_{x\in H}\chi(x) \quad = \quad \begin{cases} \frac{\chi(a)}{\sqrt{|G:H|}} & \text{if } \chi_H = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Proof:
orthogonality of 1_H and χ_H
$\frac{1}{|H|}\sum_{x\in H}\chi(x) = \begin{cases} 1 & \text{if } \chi_H = 1, \\ 0 & \text{otherwise} \end{cases}$

Abelian Fourier sampling 2

- Coefficient of χ

$$\frac{\chi(a)}{\sqrt{|G:H|}}\frac{1}{|H|}\sum_{x\in H}\chi(x) = \begin{cases} \frac{\chi(a)}{\sqrt{|G:H|}} & \text{if } \chi_H = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Proof:
orthogonality of 1_H and χ_H
$\frac{1}{|H|}\sum_{x\in H}\chi(x) = \begin{cases} 1 & \text{if } \chi_H = 1, \\ 0 & \text{otherwise} \end{cases}$

- Probability of observing χ

Abelian Fourier sampling 2

- Coefficient of χ

$$\frac{\chi(a)}{\sqrt{|G:H|}}\frac{1}{|H|}\sum_{x\in H}\chi(x) = \begin{cases} \frac{\chi(a)}{\sqrt{|G:H|}} & \text{if } \chi_H = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Proof:
orthogonality of 1_H and χ_H
$\frac{1}{|H|}\sum_{x\in H}\chi(x) = \begin{cases} 1 & \text{if } \chi_H = 1, \\ 0 & \text{otherwise} \end{cases}$

- Probability of observing χ

$$= \begin{cases} \frac{1}{|G:H|} & \text{if } \chi \in H^{\perp}, \\ 0 & \text{otherwise} \end{cases} .$$

Computing H

- $H^{\perp} = \{\chi \in \hat{G} \mid \chi_H = 1\}$ subgroup of $\hat{G}$.

Computing H

- $H^{\perp} = \{\chi \in \hat{G} \mid \chi_H = 1\}$ subgroup of $\hat{G}$.
- generating set Γ of $H^{\perp}$ collected in

expectedly $O(\log |G|)$ repetitions.

Computing H

- $H^{\perp} = \{\chi \in \hat{G} \mid \chi_H = 1\}$ subgroup of $\hat{G}$.
- generating set Γ of $H^{\perp}$ collected in

  expectedly $O(\log |G|)$ repetitions.
- $H = \{x \in G \mid \chi(x) = 1 \text{ for every } \chi \in \Gamma\}$.

Computing H

- $H^\perp = \{\chi \in \hat{G} \mid \chi_H = 1\}$ subgroup of $\hat{G}$.
- generating set Γ of $H^\perp$ collected in

  expectedly $O(\log |G|)$ repetitions.
- $H = \{x \in G \mid \chi(x) = 1 \text{ for every } \chi \in \Gamma\}$.
- computing H: system of linear congruences.

Remarks on Abelian Fourier Sampling

- No need of measuring the value of f

Remarks on Abelian Fourier Sampling

- No need of measuring the value of f
- f can be quantum-state valued:

Remarks on Abelian Fourier Sampling

- No need of measuring the value of f
- f can be quantum-state valued:

  $f : G \to \mathbb{C}^X$ hides H if:

Remarks on Abelian Fourier Sampling

- No need of measuring the value of f
- f can be quantum-state valued:

  $f : G \to \mathbb{C}^X$ hides H if:

  - f constant on left cosets of H

Remarks on Abelian Fourier Sampling

- No need of measuring the value of f
- f can be quantum-state valued:

  $f : G \to \mathbb{C}^X$ hides H if:
  - f constant on left cosets of H
  - $f(a) \perp f(b)$ if $aH \neq bH$

Remarks on Abelian Fourier Sampling

- No need of measuring the value of f
- f can be quantum-state valued:

  $f : G \to \mathbb{C}^X$ hides H if:
  - f constant on left cosets of H
  - $f(a) \perp f(b)$ if $aH \neq bH$

  Fourier sampling finds H efficiently if G abelian and f hides H.

Remarks on Abelian Fourier Sampling

- No need of measuring the value of f
- f can be quantum-state valued:

  $f : G \to \mathbb{C}^X$ hides H if:
  - f constant on left cosets of H
  - $f(a) \perp f(b)$ if $aH \neq bH$

  Fourier sampling finds H efficiently if G abelian and f hides H.
- Even the function f can be different in different steps,

Remarks on Abelian Fourier Sampling

- No need of measuring the value of f
- f can be quantum-state valued:

  $f : G \to \mathbb{C}^X$ hides H if:
  - f constant on left cosets of H
  - $f(a) \perp f(b)$ if $aH \neq bH$

  Fourier sampling finds H efficiently if G abelian and f hides H.
- Even the function f can be different in different steps,

  they only must hide *the same* H.

Applications

- Group element order finding: $f(k) = a^k$ (A group, $a \in A$)

Applications

- Group element order finding: $f(k) = a^k$ (A group, $a \in A$)
- Discrete logarithm

Applications

- Group element order finding: $f(k) = a^k$ (A group, $a \in A$)
- Discrete logarithm

$\longrightarrow$ breaking many cryptosystems

Applications

- Group element order finding: $f(k) = a^k$ (A group, $a \in A$)
- Discrete logarithm

  $\longrightarrow$ breaking many cryptosystems
- Generalized discrete log:

Applications

- Group element order finding: $f(k) = a^k$ (A group, $a \in A$)
- Discrete logarithm

  $\longrightarrow$ breaking many cryptosystems
- Generalized discrete log:

  a.k.a. constructive membership in abelian black box groups

Applications

- Group element order finding: $f(k) = a^k$ (A group, $a \in A$)
- Discrete logarithm

  $\longrightarrow$ breaking many cryptosystems
- Generalized discrete log:

  a.k.a. constructive membership in abelian black box groups

  $a_1, \ldots, a_n, b \in A$, express b as:

$$b = \prod_{i=1}^{n} a_i^{k_i}$$

Applications

- Group element order finding: $f(k) = a^k$ (A group, $a \in A$)
- Discrete logarithm

  $\rightarrow$ breaking many cryptosystems
- Generalized discrete log:

  a.k.a. constructive membership in abelian black box groups

  $a_1, \ldots, a_n, b \in A$, express b as:

$$b = \prod_{i=1}^{n} a_i^{k_i}$$

Classically difficult, even in the exponent 2 case

Generalized discrete log as HSP

- $a_1, \ldots, a_n, b \in A$,

Generalized discrete log as HSP

- $a_1, \ldots, a_n, b \in A$,
- m_i : order of a_i, m : order of b

Generalized discrete log as HSP

- $a_1, \ldots, a_n, b \in A$,
- m_i : order of a_i, m : order of b
- group

$$G = \mathbb{Z}_m \oplus \bigoplus_{i=1}^{n} \mathbb{Z}_{m_i}$$

Generalized discrete log as HSP

- $a_1, \ldots, a_n, b \in A$,
- m_i : order of a_i, m : order of b
- group

$$G = \mathbb{Z}_m \oplus \bigoplus_{i=1}^{n} \mathbb{Z}_{m_i}$$

- hiding function

$$f(\ell, \ell_1, \ldots, \ell_n) = b^{-\ell} \prod_{i=1}^{n} a_i^{\ell_i}$$

Generalized discrete log as HSP

- $a_1, \ldots, a_n, b \in A$,
- m_i : order of a_i, m : order of b
- group

$$G = \mathbb{Z}_m \oplus \bigoplus_{i=1}^{n} \mathbb{Z}_{m_i}$$

- hiding function

$$f(\ell, \ell_1, \ldots, \ell_n) = b^{-\ell} \prod_{i=1}^{n} a_i^{\ell_i}$$

- hidden subgroup

$$H = \begin{cases} \langle 1, k_1, \ldots, k_n \rangle & \text{if } b = \prod_{i=1}^{n} a_i^{k_i} \\ \{0\} & \text{otherwise.} \end{cases}$$

Applications 2

- Computing the structure of finite abelian black box groups (Cheung and Mosca 2001)

Applications 2

- Computing the structure of finite abelian black box groups (Cheung and Mosca 2001)

  classically even approximating the order is difficult

Applications 2

- Computing the structure of finite abelian black box groups (Cheung and Mosca 2001)

  classically even approximating the order is difficult
- Computing with solvable (and more) black box groups

Applications 2

- Computing the structure of finite abelian black box groups (Cheung and Mosca 2001)

    classically even approximating the order is difficult
- Computing with solvable (and more) black box groups
    - Watrous (solvable)

Applications 2

- Computing the structure of finite abelian black box groups (Cheung and Mosca 2001)

  classically even approximating the order is difficult
- Computing with solvable (and more) black box groups
  - Watrous (solvable)

    Based on uniform superposition $|G\rangle = \sum_{g \in G} |g\rangle$

Applications 2

- Computing the structure of finite abelian black box groups (Cheung and Mosca 2001)

  classically even approximating the order is difficult
- Computing with solvable (and more) black box groups
  - Watrous (solvable)

    Based on uniform superposition $|G\rangle = \sum_{g \in G} |g\rangle$
  - Beals and Babai (solvable+more)

Applications 2

- Computing the structure of finite abelian black box groups (Cheung and Mosca 2001)

  classically even approximating the order is difficult
- Computing with solvable (and more) black box groups
  - Watrous (solvable)

    Based on uniform superposition $|G\rangle = \sum_{g \in G} |g\rangle$
  - Beals and Babai (solvable+more)

    Classical, with oracles for factoring and

Applications 2

- Computing the structure of finite abelian black box groups (Cheung and Mosca 2001)

  classically even approximating the order is difficult
- Computing with solvable (and more) black box groups
  - Watrous (solvable)

    Based on uniform superposition $|G\rangle = \sum_{g \in G} |g\rangle$
  - Beals and Babai (solvable+more)

    Classical, with oracles for factoring and constructive membership in abelian subgroups

Applications 2

- Computing the structure of finite abelian black box groups (Cheung and Mosca 2001)

  classically even approximating the order is difficult

- Computing with solvable (and more) black box groups
  - Watrous (solvable)

    Based on uniform superposition $|G\rangle = \sum_{g \in G} |g\rangle$
  - Beals and Babai (solvable+more)

    Classical, with oracles for factoring and constructive membership in abelian subgroups

$\downarrow$ (noticed in $\sim$, Magniez, Santha 2001)

hidden normal subgroups in such groups

Applications 2

- Computing the structure of finite abelian black box groups (Cheung and Mosca 2001)

  classically even approximating the order is difficult
- Computing with solvable (and more) black box groups
  - Watrous (solvable)

    Based on uniform superposition $|G\rangle = \sum_{g \in G} |g\rangle$
  - Beals and Babai (solvable+more)

    Classical, with oracles for factoring and constructive membership in abelian subgroups

  $\downarrow$ (noticed in $\sim$, Magniez, Santha 2001)

  hidden normal subgroups in such groups
  - Probably normal HSP in other cases (Ákos)

Factoring $\longleftarrow$ order finding

- Order finding in $\mathbb{Z}_n^*$: $a \in \mathbb{Z}_n^*$

Factoring $\longleftarrow$ order finding

- Order finding in $\mathbb{Z}_n^*$: $a \in \mathbb{Z}_n^*$

  $o(a)$=smallest r: $a^r \equiv 1 \pmod{n}$

Factoring $\longleftarrow$ order finding

- Order finding in $\mathbb{Z}_n^*$: $a \in \mathbb{Z}_n^*$

  $o(a)$=smallest r: $a^r \equiv 1 \pmod{n}$
- Assume n odd, not a prime power

Factoring $\longleftarrow$ order finding

- Order finding in $\mathbb{Z}_n^*$: $a \in \mathbb{Z}_n^*$

  $o(a)$=smallest r: $a^r \equiv 1 \pmod{n}$
- Assume n odd, not a prime power
- For random $a \in \mathbb{Z}_n^*$, with probability $\geq \frac{1}{4}$

Factoring $\longleftarrow$ order finding

- Order finding in $\mathbb{Z}_n^*$: $a \in \mathbb{Z}_n^*$

  $o(a)$=smallest r: $a^r \equiv 1 \pmod{n}$
- Assume n odd, not a prime power
- For random $a \in \mathbb{Z}_n^*$, with probability $\geq \frac{1}{4}$
  - $o(a)$ even,

Factoring $\longleftarrow$ order finding

- Order finding in $\mathbb{Z}_n^*$: $a \in \mathbb{Z}_n^*$

  $o(a)$=smallest r: $a^r \equiv 1 \pmod{n}$
- Assume n odd, not a prime power
- For random $a \in \mathbb{Z}_n^*$, with probability $\geq \frac{1}{4}$
  - $o(a)$ even,
  - $b = a^{\frac{o(a)}{2}} \not\equiv \pm 1 \pmod{n}$, but $b^2 \equiv 1$:

Factoring $\longleftarrow$ order finding

- Order finding in $\mathbb{Z}_n^*$: $a \in \mathbb{Z}_n^*$

  $o(a)$=smallest r: $a^r \equiv 1 \pmod{n}$
- Assume n odd, not a prime power
- For random $a \in \mathbb{Z}_n^*$, with probability $\geq \frac{1}{4}$
  - $o(a)$ even,
  - $b = a^{\frac{o(a)}{2}} \not\equiv \pm 1 \pmod{n}$, but $b^2 \equiv 1$:

$\Downarrow$

Factoring $\longleftarrow$ order finding

- Order finding in $\mathbb{Z}_n^*$: $a \in \mathbb{Z}_n^*$

  $o(a)$=smallest r: $a^r \equiv 1 \pmod n$
- Assume n odd, not a prime power
- For random $a \in \mathbb{Z}_n^*$, with probability $\geq \frac{1}{4}$
  - $o(a)$ even,
  - $b = a^{\frac{o(a)}{2}} \not\equiv \pm 1 \pmod n$, but $b^2 \equiv 1$:

$$\Downarrow$$

$\gcd(b-1, n)$ a proper factor of n.

Discrete log - limitations

- No efficient equality-test based discrete log

Discrete log - limitations

- No efficient equality-test based discrete log
  - $A = \mathbb{Z}_p \times \mathbb{Z}_p$

Discrete log - limitations

- No efficient equality-test based discrete log
  - $A = \mathbb{Z}_p \times \mathbb{Z}_p$
  - Secret: $u \in \mathbb{Z}_p^*$

Discrete log - limitations

- No efficient equality-test based discrete log
  - $A = \mathbb{Z}_p \times \mathbb{Z}_p$
  - Secret: $u \in \mathbb{Z}_p^*$
  - Subgroup $U = \{(x, ux) | x \in \mathbb{Z}_p\}$

Discrete log - limitations

- No efficient equality-test based discrete log
  - $A = \mathbb{Z}_p \times \mathbb{Z}_p$
  - Secret: $u \in \mathbb{Z}_p^*$
  - Subgroup $U = \{(x, ux) | x \in \mathbb{Z}_p\}$
  - $G = A/U$: encoding and $+$ in A;

Discrete log - limitations

- No efficient equality-test based discrete log
  - $A = \mathbb{Z}_p \times \mathbb{Z}_p$
  - Secret: $u \in \mathbb{Z}_p^*$
  - Subgroup $U = \{(x, ux) | x \in \mathbb{Z}_p\}$
  - $G = A/U$: encoding and $+$ in A;
    equality test (membership in U) by black box

Discrete log - limitations

- No efficient equality-test based discrete log
  - $A = \mathbb{Z}_p \times \mathbb{Z}_p$
  - Secret: $u \in \mathbb{Z}_p^*$
  - Subgroup $U = \{(x, ux) | x \in \mathbb{Z}_p\}$
  - $G = A/U$: encoding and $+$ in A;
    equality test (membership in U) by black box
  - Property:

$$\log_{(0,1)}(-1, 0) = \ell \Leftrightarrow (1, 0) + \ell(0, 1) \in U \Leftrightarrow \ell = u$$

Discrete log - limitations

- No efficient equality-test based discrete log
  - $A = \mathbb{Z}_p \times \mathbb{Z}_p$
  - Secret: $u \in \mathbb{Z}_p^*$
  - Subgroup $U = \{(x, ux) | x \in \mathbb{Z}_p\}$
  - $G = A/U$: encoding and $+$ in A;
    equality test (membership in U) by black box
  - Property:

$$\log_{(0,1)}(-1, 0) = \ell \Leftrightarrow (1, 0) + \ell(0, 1) \in U \Leftrightarrow \ell = u$$

  - Gives reduction from (quantum) search

Discrete log - limitations

- No efficient equality-test based discrete log
  - $A = \mathbb{Z}_p \times \mathbb{Z}_p$
  - Secret: $u \in \mathbb{Z}_p^*$
  - Subgroup $U = \{(x, ux) | x \in \mathbb{Z}_p\}$
  - $G = A/U$: encoding and $+$ in A;
    equality test (membership in U) by black box
  - Property:

$$\log_{(0,1)}(-1,0) = \ell \Leftrightarrow (1,0) + \ell(0,1) \in U \Leftrightarrow \ell = u$$

  - Gives reduction from (quantum) search
    lower bound $\Omega(\sqrt{p})$ quantum queries

Discrete log - limitations

- No efficient equality-test based discrete log
  - $A = \mathbb{Z}_p \times \mathbb{Z}_p$
  - Secret: $u \in \mathbb{Z}_p^*$
  - Subgroup $U = \{(x, ux) | x \in \mathbb{Z}_p\}$
  - $G = A/U$: encoding and $+$ in A;
    equality test (membership in U) by black box
  - Property:

$$\log_{(0,1)}(-1, 0) = \ell \Leftrightarrow (1, 0) + \ell(0, 1) \in U \Leftrightarrow \ell = u$$

  - Gives reduction from (quantum) search
    lower bound $\Omega(\sqrt{p})$ quantum queries
- Open: Complexity of equality-test–based *order finding*?

Contents

HSP in $\mathbb{Z}^n$

- $G = \mathbb{Z}^n$, $f : G \to \{0, 1\}^s$

HSP in $\mathbb{Z}^n$

- $G = \mathbb{Z}^n$, $f : G \to \{0,1\}^s$
  $$f(x) = f(y) \Leftrightarrow x - y \in H$$

HSP in $\mathbb{Z}^n$

- $G = \mathbb{Z}^n$, $f : G \to \{0,1\}^s$

  $$f(x) = f(y) \Leftrightarrow x - y \in H$$

- $|G : H| \leq 2^s \Rightarrow H \geq$ rectangular lattice K,

HSP in $\mathbb{Z}^n$

- $G = \mathbb{Z}^n$, $f : G \to \{0,1\}^s$

  $$f(x) = f(y) \Leftrightarrow x - y \in H$$

- $|G : H| \leq 2^s \Rightarrow H \geq$ rectangular lattice K,

  basis vectors for K have length at most 2^s.

HSP in $\mathbb{Z}^n$

- $G = \mathbb{Z}^n$, $f : G \to \{0,1\}^s$

  $f(x) = f(y) \Leftrightarrow x - y \in H$
- $|G : H| \leq 2^s \Rightarrow H \geq$ rectangular lattice K,

  basis vectors for K have length at most 2^s.
  - $H_1 = \{x \in \mathbb{Z} | (x, 0, \ldots, 0) \in H\}$

HSP in $\mathbb{Z}^n$

- $G = \mathbb{Z}^n$, $f : G \to \{0,1\}^s$

  $$f(x) = f(y) \Leftrightarrow x - y \in H$$
- $|G : H| \leq 2^s \Rightarrow H \geq$ rectangular lattice K,

  basis vectors for K have length at most 2^s.
  - $H_1 = \{x \in \mathbb{Z} | (x, 0, \ldots, 0) \in H\}$
    hidden by $f(x, 0, \ldots, 0)$

HSP in $\mathbb{Z}^n$

- $G = \mathbb{Z}^n$, $f : G \to \{0,1\}^s$

  $f(x) = f(y) \Leftrightarrow x - y \in H$
- $|G : H| \leq 2^s \Rightarrow H \geq$ rectangular lattice K,

  basis vectors for K have length at most 2^s.
  - $H_1 = \{x \in \mathbb{Z} | (x, 0, \ldots, 0) \in H\}$
    hidden by $f(x, 0, \ldots, 0)$
  - $H_2, \ldots, H_n$ similar

HSP in $\mathbb{Z}^n$

- $G = \mathbb{Z}^n$, $f : G \rightarrow \{0,1\}^s$
  $f(x) = f(y) \Leftrightarrow x - y \in H$
- $|G : H| \leq 2^s \Rightarrow H \geq$ rectangular lattice K,
  basis vectors for K have length at most 2^s.
  - $H_1 = \{x \in \mathbb{Z} | (x, 0, \ldots, 0) \in H\}$
    hidden by $f(x, 0, \ldots, 0)$
  - $H_2, \ldots, H_n$ similar
  - Find $H_1, H_2, \ldots, H_n$ by Shor's algorithm/phase estimation

HSP in $\mathbb{Z}^n$

- $G = \mathbb{Z}^n$, $f : G \to \{0,1\}^s$
  $$f(x) = f(y) \Leftrightarrow x - y \in H$$
- $|G : H| \leq 2^s \Rightarrow H \geq$ rectangular lattice K,
  basis vectors for K have length at most 2^s.
  - $H_1 = \{x \in \mathbb{Z} | (x, 0, \ldots, 0) \in H\}$
    hidden by $f(x, 0, \ldots, 0)$
  - $H_2, \ldots, H_n$ similar
  - Find $H_1, H_2, \ldots, H_n$ by Shor's algorithm/phase estimation
- $H \geq K = H_1 \times H_2 \times \cdots \times H_n$

HSP in $\mathbb{Z}^n$

- $G = \mathbb{Z}^n$, $f : G \to \{0,1\}^s$

  $$f(x) = f(y) \Leftrightarrow x - y \in H$$

- $|G : H| \leq 2^s \Rightarrow H \geq$ rectangular lattice K,

  basis vectors for K have length at most 2^s.

  - $H_1 = \{x \in \mathbb{Z} | (x, 0, \ldots, 0) \in H\}$
    hidden by $f(x, 0, \ldots, 0)$
  - $H_2, \ldots, H_n$ similar
  - Find $H_1, H_2, \ldots, H_n$ by Shor's algorithm/phase estimation

- $H \geq K = H_1 \times H_2 \times \cdots \times H_n$
- f constant on $H \Rightarrow$ well defined on $\mathbb{Z}^n/K$

HSP in $\mathbb{Z}^n$

- $G = \mathbb{Z}^n$, $f : G \to \{0,1\}^s$

  $$f(x) = f(y) \Leftrightarrow x - y \in H$$

- $|G : H| \leq 2^s \Rightarrow H \geq$ rectangular lattice K,

  basis vectors for K have length at most 2^s.

  - $H_1 = \{x \in \mathbb{Z} | (x, 0, \ldots, 0) \in H\}$
    hidden by $f(x, 0, \ldots, 0)$
  - $H_2, \ldots, H_n$ similar
  - Find $H_1, H_2, \ldots, H_n$ by Shor's algorithm/phase estimation

- $H \geq K = H_1 \times H_2 \times \cdots \times H_n$
- f constant on $H \Rightarrow$ well defined on $\mathbb{Z}^n/K$
- hides H/K in finite $\mathbb{Z}^n/K$

Units in number fields

- K number field

Units in number fields

- K number field
  - given by $f(x) \in \mathbb{Q}[x]$ irred., $\deg f = m$, $K \cong \mathbb{Q}[x]/(f(x))$

Units in number fields

- K number field
  - given by $f(x) \in \mathbb{Q}[x]$ irred., $\deg f = m$, $K \cong \mathbb{Q}[x]/(f(x))$
  - standard repr. of elements: polynomials modulo f

Units in number fields

- K number field
  - given by $f(x) \in \mathbb{Q}[x]$ irred., $\deg f = m$, $K \cong \mathbb{Q}[x]/(f(x))$
  - standard repr. of elements: polynomials modulo f
- $\mathcal{O}$ ring of algebraic integers in K

Units in number fields

- K number field
  - given by $f(x) \in \mathbb{Q}[x]$ irred., $\deg f = m$, $K \cong \mathbb{Q}[x]/(f(x))$
  - standard repr. of elements: polynomials modulo f
- $\mathcal{O}$ ring of algebraic integers in K
- **unit group:** $\mathcal{O}^* = \{x \in \mathcal{O} | x^{-1} \in \mathcal{O}\}$

Units in number fields

- K number field
  - given by $f(x) \in \mathbb{Q}[x]$ irred., $\deg f = m$, $K \cong \mathbb{Q}[x]/(f(x))$
  - standard repr. of elements: polynomials modulo f
- $\mathcal{O}$ ring of algebraic integers in K
- **unit group:** $\mathcal{O}^* = \{x \in \mathcal{O} | x^{-1} \in \mathcal{O}\}$
- **Dirichlet's unit theorem:** $\mathcal{O}^* \cong \mathbb{Z}_s \times \mathbb{Z}^r$,

Units in number fields

- K number field
  - given by $f(x) \in \mathbb{Q}[x]$ irred., $\deg f = m$, $K \cong \mathbb{Q}[x]/(f(x))$
  - standard repr. of elements: polynomials modulo f
- $\mathcal{O}$ ring of algebraic integers in K
- **unit group:** $\mathcal{O}^* = \{x \in \mathcal{O} | x^{-1} \in \mathcal{O}\}$
- **Dirichlet's unit theorem:** $\mathcal{O}^* \cong \mathbb{Z}_s \times \mathbb{Z}^r$,
  - s largest s.t. $\sqrt[s]{1} \in K$,

Units in number fields

- K number field
  - given by $f(x) \in \mathbb{Q}[x]$ irred., $\deg f = m$, $K \cong \mathbb{Q}[x]/(f(x))$
  - standard repr. of elements: polynomials modulo f
- $\mathcal{O}$ ring of algebraic integers in K
- **unit group:** $\mathcal{O}^* = \{x \in \mathcal{O} | x^{-1} \in \mathcal{O}\}$
- **Dirichlet's unit theorem:** $\mathcal{O}^* \cong \mathbb{Z}_s \times \mathbb{Z}^r$,
  - s largest s.t. $\sqrt[s]{1} \in K$,
  - $f(x)$ has r_1 real, $2r_2$ imaginary roots

Units in number fields

- K number field
  - given by $f(x) \in \mathbb{Q}[x]$ irred., $\deg f = m$, $K \cong \mathbb{Q}[x]/(f(x))$
  - standard repr. of elements: polynomials modulo f
- $\mathcal{O}$ ring of algebraic integers in K
- **unit group:** $\mathcal{O}^* = \{x \in \mathcal{O} | x^{-1} \in \mathcal{O}\}$
- **Dirichlet's unit theorem:** $\mathcal{O}^* \cong \mathbb{Z}_s \times \mathbb{Z}^r$,
  - s largest s.t. $\sqrt[s]{1} \in K$,
  - $f(x)$ has r_1 real, $2r_2$ imaginary roots
  - $r = r_1 + r_2 - 1$

Units in number fields

- K number field
  - given by $f(x) \in \mathbb{Q}[x]$ irred., $\deg f = m$, $K \cong \mathbb{Q}[x]/(f(x))$
  - standard repr. of elements: polynomials modulo f
- $\mathcal{O}$ ring of algebraic integers in K
- **unit group:** $\mathcal{O}^* = \{x \in \mathcal{O} | x^{-1} \in \mathcal{O}\}$
- **Dirichlet's unit theorem:** $\mathcal{O}^* \cong \mathbb{Z}_s \times \mathbb{Z}^r$,
  - s largest s.t. $\sqrt[s]{1} \in K$,
  - $f(x)$ has r_1 real, $2r_2$ imaginary roots
  - $r = r_1 + r_2 - 1$
- **Task** find basis for (free part of) $\mathcal{O}^*$

Units in number fields

- K number field
  - given by $f(x) \in \mathbb{Q}[x]$ irred., $\deg f = m$, $K \cong \mathbb{Q}[x]/(f(x))$
  - standard repr. of elements: polynomials modulo f
- $\mathcal{O}$ ring of algebraic integers in K
- **unit group:** $\mathcal{O}^* = \{x \in \mathcal{O} | x^{-1} \in \mathcal{O}\}$
- **Dirichlet's unit theorem:** $\mathcal{O}^* \cong \mathbb{Z}_s \times \mathbb{Z}^r$,
  - s largest s.t. $\sqrt[s]{1} \in K$,
  - $f(x)$ has r_1 real, $2r_2$ imaginary roots
  - $r = r_1 + r_2 - 1$
- **Task** find basis for (free part of) $\mathcal{O}^*$
- finding $\sqrt[s]{1} \in K$ easy (deterministic poly time)

Unit groups - naive HSP approach

- $G = K^*$; $F(x) = x\mathcal{O} = \{xy | y \in \mathcal{O}\}$ principal fractional ideal

Unit groups - naive HSP approach

- $G = K^*$; $F(x) = x\mathcal{O} = \{xy | y \in \mathcal{O}\}$ principal fractional ideal
- hidden subgroup is $\mathcal{O}^*$: $F(x) = F(y) \Leftrightarrow x^{-1}y \in \mathcal{O}^*$.

Unit groups - naive HSP approach

- $G = K^*$; $F(x) = x\mathcal{O} = \{xy | y \in \mathcal{O}\}$ principal fractional ideal
- hidden subgroup is $\mathcal{O}^*$: $F(x) = F(y) \Leftrightarrow x^{-1}y \in \mathcal{O}^*$.
- Difficulties:

Unit groups - naive HSP approach

- $G = K^*$; $F(x) = x\mathcal{O} = \{xy | y \in \mathcal{O}\}$ principal fractional ideal
- hidden subgroup is $\mathcal{O}^*$: $F(x) = F(y) \Leftrightarrow x^{-1}y \in \mathcal{O}^*$.
- Difficulties:
  - encoding of fractional ideals

Unit groups - naive HSP approach

- $G = K^*$; $F(x) = x\mathcal{O} = \{xy | y \in \mathcal{O}\}$ principal fractional ideal
- hidden subgroup is $\mathcal{O}^*$: $F(x) = F(y) \Leftrightarrow x^{-1}y \in \mathcal{O}^*$.
- Difficulties:
  - encoding of fractional ideals
    -factorization into powers of prime ideals

Unit groups - naive HSP approach

- $G = K^*$; $F(x) = x\mathcal{O} = \{xy | y \in \mathcal{O}\}$ principal fractional ideal
- hidden subgroup is $\mathcal{O}^*$: $F(x) = F(y) \Leftrightarrow x^{-1}y \in \mathcal{O}^*$.
- Difficulties:
  - encoding of fractional ideals
    -factorization into powers of prime ideals
    -Hermite normal form (HNF): special basis, can be large

Unit groups - naive HSP approach

- $G = K^*$; $F(x) = x\mathcal{O} = \{xy | y \in \mathcal{O}\}$ principal fractional ideal
- hidden subgroup is $\mathcal{O}^*$: $F(x) = F(y) \Leftrightarrow x^{-1}y \in \mathcal{O}^*$.
- Difficulties:
  - encoding of fractional ideals
    -factorization into powers of prime ideals
    -Hermite normal form (HNF): special basis, can be large
  - K^* infinitely generated

Unit groups - naive HSP approach

- $G = K^*$; $F(x) = x\mathcal{O} = \{xy | y \in \mathcal{O}\}$ principal fractional ideal
- hidden subgroup is $\mathcal{O}^*$: $F(x) = F(y) \Leftrightarrow x^{-1}y \in \mathcal{O}^*$.
- Difficulties:
  - encoding of fractional ideals
    -factorization into powers of prime ideals
    -Hermite normal form (HNF): special basis, can be large
  - K^* infinitely generated
    - not known how to find a "small" piece
    containing the units

Unit groups - naive HSP approach

- $G = K^*$; $F(x) = x\mathcal{O} = \{xy | y \in \mathcal{O}\}$ principal fractional ideal
- hidden subgroup is $\mathcal{O}^*$: $F(x) = F(y) \Leftrightarrow x^{-1}y \in \mathcal{O}^*$.
- Difficulties:
  - encoding of fractional ideals
    -factorization into powers of prime ideals
    -Hermite normal form (HNF): special basis, can be large
  - K^* infinitely generated
    - not known how to find a "small" piece
    containing the units
  - generators may have exponential size

Unit groups - naive HSP approach

- $G = K^*$; $F(x) = x\mathcal{O} = \{xy | y \in \mathcal{O}\}$ principal fractional ideal
- hidden subgroup is $\mathcal{O}^*$: $F(x) = F(y) \Leftrightarrow x^{-1}y \in \mathcal{O}^*$.
- Difficulties:
    - encoding of fractional ideals
      -factorization into powers of prime ideals
      -Hermite normal form (HNF): special basis, can be large
    - K^* infinitely generated
      - not known how to find a "small" piece
      containing the units
    - generators may have exponential size
      using usual representations

Unit groups - naive HSP approach

- $G = K^*$; $F(x) = x\mathcal{O} = \{xy | y \in \mathcal{O}\}$ principal fractional ideal
- hidden subgroup is $\mathcal{O}^*$: $F(x) = F(y) \Leftrightarrow x^{-1}y \in \mathcal{O}^*$.
- Difficulties:
  - encoding of fractional ideals
    -factorization into powers of prime ideals
    -Hermite normal form (HNF): special basis, can be large
  - K^* infinitely generated
    - not known how to find a "small" piece
    containing the units
  - generators may have exponential size
    using usual representations
  - Solution: *"compact representations"* (Thiel 95)

Unit groups - naive HSP approach

- $G = K^*$; $F(x) = x\mathcal{O} = \{xy | y \in \mathcal{O}\}$ principal fractional ideal
- hidden subgroup is $\mathcal{O}^*$: $F(x) = F(y) \Leftrightarrow x^{-1}y \in \mathcal{O}^*$.
- Difficulties:
  - encoding of fractional ideals
    -factorization into powers of prime ideals
    -Hermite normal form (HNF): special basis, can be large
  - K^* infinitely generated
    - not known how to find a "small" piece
    containing the units
  - generators may have exponential size
    using usual representations
  - Solution: *"compact representations"* (Thiel 95)
    special straight-line programs

Unit groups - the Log map

- $K = \mathbb{Q}[\alpha] = \mathbb{Q}[x]/(f(x))$

Unit groups - the Log map

- $K = \mathbb{Q}[\alpha] = \mathbb{Q}[x]/(f(x))$
- Roots of $f(x)$ $\alpha_1, \ldots, \alpha_{r_1}$ (real)

Unit groups - the Log map

- $K = \mathbb{Q}[\alpha] = \mathbb{Q}[x]/(f(x))$
- Roots of $f(x)$ $\alpha_1, \ldots, \alpha_{r_1}$ (real)

  $\alpha_{r_1+1}, \overline{\alpha_{r_1+1}}, \ldots, \alpha_{r_1+r_2}, \overline{\alpha_{r_1+r_2}}$ (imaginary)

Unit groups - the Log map

- $K = \mathbb{Q}[\alpha] = \mathbb{Q}[x]/(f(x))$
- Roots of $f(x)$ $\alpha_1, \ldots, \alpha_{r_1}$ (real)

  $\alpha_{r_1+1}, \overline{\alpha_{r_1+1}}, \ldots, \alpha_{r_1+r_2}, \overline{\alpha_{r_1+r_2}}$ (imaginary)
- **absolute values** $y = g(\alpha) \in K$, $|y|_i = |g(\alpha_i)|$

Unit groups - the Log map

- $K = \mathbb{Q}[\alpha] = \mathbb{Q}[x]/(f(x))$
- Roots of $f(x)$ $\alpha_1, \ldots, \alpha_{r_1}$ (real)

  $\alpha_{r_1+1}, \overline{\alpha_{r_1+1}}, \ldots, \alpha_{r_1+r_2}, \overline{\alpha_{r_1+r_2}}$ (imaginary)
- **absolute values** $y = g(\alpha) \in K$, $|y|_i = |g(\alpha_i)|$

  $r_1 + r_2$ achimedean absolute values

Unit groups - the Log map

- $K = \mathbb{Q}[\alpha] = \mathbb{Q}[x]/(f(x))$
- Roots of $f(x)$ $\alpha_1, \ldots, \alpha_{r_1}$ (real)

  $\alpha_{r_1+1}, \overline{\alpha_{r_1+1}}, \ldots, \alpha_{r_1+r_2}, \overline{\alpha_{r_1+r_2}}$ (imaginary)
- **absolute values** $y = g(\alpha) \in K$, $|y|_i = |g(\alpha_i)|$

  $r_1 + r_2$ achimedean absolute values

  Log : $K^* \to \mathbb{R}^r$ $\quad y \mapsto \log|y|_1, \ldots, \log|y|_r$ $(r = r_1 + r_2 - 1)$

Unit groups - the Log map

- $K = \mathbb{Q}[\alpha] = \mathbb{Q}[x]/(f(x))$
- Roots of $f(x)$ $\alpha_1, \ldots, \alpha_{r_1}$ (real)

  $\alpha_{r_1+1}, \overline{\alpha_{r_1+1}}, \ldots, \alpha_{r_1+r_2}, \overline{\alpha_{r_1+r_2}}$ (imaginary)
- **absolute values** $y = g(\alpha) \in K$, $|y|_i = |g(\alpha_i)|$

  $r_1 + r_2$ achimedean absolute values

  Log : $K^* \to \mathbb{R}^r$ $\quad y \mapsto \log|y|_1, \ldots, \log|y|_r$ $(r = r_1 + r_2 - 1)$
- **Dirichlet:** Log$(\mathcal{O}^*)$ **full lattice** in $\mathbb{R}^r$.

Unit groups - the Log map

- $K = \mathbb{Q}[\alpha] = \mathbb{Q}[x]/(f(x))$
- Roots of $f(x)$ $\alpha_1, \ldots, \alpha_{r_1}$ (real)

  $\alpha_{r_1+1}, \overline{\alpha_{r_1+1}}, \ldots, \alpha_{r_1+r_2}, \overline{\alpha_{r_1+r_2}}$ (imaginary)
- **absolute values** $y = g(\alpha) \in K$, $|y|_i = |g(\alpha_i)|$

  $r_1 + r_2$ achimedean absolute values

  $\mathrm{Log} : K^* \to \mathbb{R}^r \quad y \mapsto \log|y|_1, \ldots, \log|y|_r \ (r = r_1 + r_2 - 1)$
- **Dirichlet:** $\mathrm{Log}(\mathcal{O}^*)$ **full lattice** in $\mathbb{R}^r$.
- Remarks: for $y \in \mathcal{O}^*$:

$$\prod_{i=1}^{r_1} |y|_i \prod_{j=r_1+1}^{r_1+r_2} |y|_j^2 = \mathrm{Norm(y)} = 1.$$

$\mathcal{O} \cap \ker \mathrm{Log} = \langle \sqrt[s]{1} \rangle$ (Kronecker)

Unit groups - minima

- partial order on K^*/some subgroup $a \leq b$ iff $|a|_i \leq |b_i|$ $(i = 1, \ldots, r_1 + r_2)$

Unit groups - minima

- partial order on K^*/some subgroup $a \leq b$ iff $|a|_i \leq |b_i|$ $(i = 1, \ldots, r_1 + r_2)$
- **Minimal element** of a fractional ideal I:

Unit groups - minima

- partial order on $K^*/$some subgroup $a \leq b$ iff $|a|_i \leq |b_i|$ $(i = 1, \ldots, r_1 + r_2)$
- **Minimal element** of a fractional ideal I:
  - minimal element of I w.r.t $\leq$ above

Unit groups - minima

- partial order on K^*/some subgroup $a \leq b$ iff $|a|_i \leq |b_i|$ $(i = 1, \ldots, r_1 + r_2)$
- **Minimal element** of a fractional ideal I:
  - minimal element of I w.r.t $\leq$ above
  - $a \in I \setminus \{0\}$, s.t. $\nexists a' \in I \setminus \{0\}$:
    $|a'|_i \leq |a|_i$ for $i = 1, \ldots, r+1$ and $\exists i$: $|a'|_i < |a|_i$

Unit groups - minima

- partial order on K^*/some subgroup $a \leq b$ iff $|a|_i \leq |b_i|$ $(i = 1, \ldots, r_1 + r_2)$
- **Minimal element** of a fractional ideal I:
  - minimal element of I w.r.t $\leq$ above
  - $a \in I \setminus \{0\}$, s.t. $\nexists a' \in I \setminus \{0\}$:
    $|a'|_i \leq |a|_i$ for $i = 1, \ldots, r+1$ and $\exists i$: $|a'|_i < |a|_i$
- Examples in $\mathcal{O}$: units + usually $\exists$ others

Unit groups - minima

- partial order on K^*/some subgroup $a \leq b$ iff $|a|_i \leq |b_i|$ $(i = 1, \ldots, r_1 + r_2)$
- **Minimal element** of a fractional ideal I:
  - minimal element of I w.r.t $\leq$ above
  - $a \in I \setminus \{0\}$, s.t. $\nexists a' \in I \setminus \{0\}$:
    $|a'|_i \leq |a|_i$ for $i = 1, \ldots, r+1$ and $\exists i$: $|a'|_i < |a|_i$
- Examples in $\mathcal{O}$: units + usually $\exists$ others
- Minkowksi's convex body thm $\Rightarrow$

Unit groups - minima

- partial order on K^*/some subgroup $a \leq b$ iff $|a|_i \leq |b_i|$ $(i = 1, \ldots, r_1 + r_2)$
- **Minimal element** of a fractional ideal I:
  - minimal element of I w.r.t $\leq$ above
  - $a \in I \setminus \{0\}$, s.t. $\nexists a' \in I \setminus \{0\}$:
    $|a'|_i \leq |a|_i$ for $i = 1, \ldots, r+1$ and $\exists i$: $|a'|_i < |a|_i$
- Examples in $\mathcal{O}$: units + usually $\exists$ others
- Minkowksi's convex body thm $\Rightarrow$

  Bound on log of norms of minima: poly in

Unit groups - minima

- partial order on K^*/some subgroup $a \leq b$ iff $|a|_i \leq |b_i|$ $(i = 1, \ldots, r_1 + r_2)$
- **Minimal element** of a fractional ideal I:
  - minimal element of I w.r.t $\leq$ above
  - $a \in I \setminus \{0\}$, s.t. $\nexists a' \in I \setminus \{0\}$:
    $|a'|_i \leq |a|_i$ for $i = 1, \ldots, r+1$ and $\exists i$: $|a'|_i < |a|_i$
- Examples in $\mathcal{O}$: units + usually $\exists$ others
- Minkowksi's convex body thm $\Rightarrow$

  Bound on log of norms of minima: poly in
  - #bits of input $(f(x))$

Unit groups - minima

- partial order on K^*/some subgroup $a \leq b$ iff $|a|_i \leq |b_i|$ $(i = 1, \ldots, r_1 + r_2)$
- **Minimal element** of a fractional ideal I:
  - minimal element of I w.r.t $\leq$ above
  - $a \in I \setminus \{0\}$, s.t. $\nexists a' \in I \setminus \{0\}$: $|a'|_i \leq |a|_i$ for $i = 1, \ldots, r+1$ and $\exists i$: $|a'|_i < |a|_i$
- Examples in $\mathcal{O}$: units + usually $\exists$ others
- Minkowksi's convex body thm $\Rightarrow$

  Bound on log of norms of minima: poly in
  - #bits of input $(f(x))$
  - log of norm of I

Unit groups - minima

- partial order on K^*/some subgroup $a \leq b$ iff $|a|_i \leq |b_i|$ $(i = 1, \ldots, r_1 + r_2)$
- **Minimal element** of a fractional ideal I:
  - minimal element of I w.r.t $\leq$ above
  - $a \in I \setminus \{0\}$, s.t. $\nexists a' \in I \setminus \{0\}$:
    $|a'|_i \leq |a|_i$ for $i = 1, \ldots, r+1$ and $\exists i$: $|a'|_i < |a|_i$
- Examples in $\mathcal{O}$: units + usually $\exists$ others
- Minkowksi's convex body thm $\Rightarrow$

  Bound on log of norms of minima: poly in
  - #bits of input $(f(x))$
  - log of norm of I

  $\Downarrow$

Unit groups - minima

- partial order on $K^*/$some subgroup $a \leq b$ iff $|a|_i \leq |b_i|$ $(i = 1, \ldots, r_1 + r_2)$
- **Minimal element** of a fractional ideal I:
  - minimal element of I w.r.t $\leq$ above
  - $a \in I \setminus \{0\}$, s.t. $\nexists a' \in I \setminus \{0\}$:
    $|a'|_i \leq |a|_i$ for $i = 1, \ldots, r + 1$ and $\exists i$: $|a'|_i < |a|_i$
- Examples in $\mathcal{O}$: units + usually $\exists$ others
- Minkowksi's convex body thm $\Rightarrow$

  Bound on log of norms of minima: poly in
  - #bits of input $(f(x))$
  - log of norm of I

  $\Downarrow$
- finitely ($\leq$ exponentially) many minima in I

Unit groups - reduced ideals

- I **reduced** if 1 is a minimum of I

Unit groups - reduced ideals

- I **reduced** if 1 is a minimum of I
- If I reduced fractional ideal then

Unit groups - reduced ideals

- I **reduced** if 1 is a minimum of I
- If I reduced fractional ideal then
  - $I^{-1} = \{x \in K^* | xI \in \mathcal{O}\}$ ideal of $\mathcal{O}$

Unit groups - reduced ideals

- I **reduced** if 1 is a minimum of I
- If I reduced fractional ideal then
  - $I^{-1} = \{x \in K^* | xI \in \mathcal{O}\}$ ideal of $\mathcal{O}$
  - I has poly size HNF

Unit groups - reduced ideals

- I **reduced** if 1 is a minimum of I
- If I reduced fractional ideal then
  - $I^{-1} = \{x \in K^* | xI \in \mathcal{O}\}$ ideal of $\mathcal{O}$
  - I has poly size HNF

(by Minkowski's convex body thm)

Unit groups - reduced ideals

- I **reduced** if 1 is a minimum of I
- If I reduced fractional ideal then
  - $I^{-1} = \{x \in K^* | xI \in \mathcal{O}\}$ ideal of $\mathcal{O}$
  - I has poly size HNF

(by Minkowski's convex body thm)

$\Downarrow$

Unit groups - reduced ideals

- I **reduced** if 1 is a minimum of I
- If I reduced fractional ideal then
  - $I^{-1} = \{x \in K^* | xI \in \mathcal{O}\}$ ideal of $\mathcal{O}$
  - I has poly size HNF

    (by Minkowski's convex body thm)

    $\Downarrow$
  - finitely (at most exponentially) many reduced ideals

Unit groups - reduced ideals

- I **reduced** if 1 is a minimum of I
- If I reduced fractional ideal then
  - $I^{-1} = \{x \in K^* | xI \in \mathcal{O}\}$ ideal of $\mathcal{O}$
  - I has poly size HNF

    (by Minkowski's convex body thm)

    $\Downarrow$
  - finitely (at most exponentially) many reduced ideals
- reduced principal ideals: $\frac{1}{a}\mathcal{O}$, where a minimum of $\mathcal{O}$.

Unit groups - neighboring minima

- Buchmann's algorithm

Unit groups - neighboring minima

- Buchmann's algorithm
  - Given $x \in \mathbb{R}^r$,

Unit groups - neighboring minima

- Buchmann's algorithm
  - Given $x \in \mathbb{R}^r$,
  - List:

Unit groups - neighboring minima

- Buchmann's algorithm
  - Given $x \in \mathbb{R}^r$,
  - List:

(compact representations for)

Unit groups - neighboring minima

- Buchmann's algorithm
  - Given $x \in \mathbb{R}^r$,
  - List:

(compact representations for)

minima y s.t.

Unit groups - neighboring minima

- Buchmann's algorithm
  - Given $x \in \mathbb{R}^r$,
  - List:

(compact representations for)

minima y s.t.

$\mathrm{Log}(y)$ close to x.

Unit groups - neighboring minima

- Buchmann's algorithm
  - Given $x \in \mathbb{R}^r$,
  - List:

    (compact representations for)

    minima y s.t.

    $\mathrm{Log}(y)$ close to x.
  - in polynomial time for constant degree

Unit groups - neighboring minima

- Buchmann's algorithm
  - Given $x \in \mathbb{R}^r$,
  - List:

    (compact representations for)

    minima y s.t.

    $\mathrm{Log}(y)$ close to x.
  - in polynomial time for constant degree

    (polynomially many such y)

Unit groups - neighboring minima

- Buchmann's algorithm
  - Given $x \in \mathbb{R}^r$,
  - List:

    (compact representations for)

    minima y s.t.

    $\mathrm{Log}(y)$ close to x.
  - in polynomial time for constant degree

    (polynomially many such y)
- Hiding function $F : \mathbb{R}^r \ni x \mapsto (I_x, \delta_x)$ **hides** $\mathrm{Log}(\mathcal{O}^*)$, where

Unit groups - neighboring minima

- Buchmann's algorithm
  - Given $x \in \mathbb{R}^r$,
  - List:

    (compact representations for)

    minima y s.t.

    $\mathrm{Log}(y)$ close to x.
  - in polynomial time for constant degree

    (polynomially many such y)
- Hiding function $F : \mathbb{R}^r \ni x \mapsto (I_x, \delta_x)$ **hides** $\mathrm{Log}(\mathcal{O}^*)$, where
  - y minimum with $\mathrm{Log}(y)$ "downwards closest" to x

Unit groups - neighboring minima

- Buchmann's algorithm
  - Given $x \in \mathbb{R}^r$,
  - List:

    (compact representations for)

    minima y s.t.

    $\mathrm{Log}(y)$ close to x.
  - in polynomial time for constant degree

    (polynomially many such y)
- Hiding function $F : \mathbb{R}^r \ni x \mapsto (I_x, \delta_x)$ **hides** $\mathrm{Log}(\mathcal{O}^*)$, where
  - y minimum with $\mathrm{Log}(y)$ "downwards closest" to x

    (appropriate choice if more)

Unit groups - neighboring minima

- Buchmann's algorithm
  - Given $x \in \mathbb{R}^r$,
  - List:

    (compact representations for)

    minima y s.t.

    $\mathrm{Log}(y)$ close to x.
  - in polynomial time for constant degree

    (polynomially many such y)
- Hiding function $F : \mathbb{R}^r \ni x \mapsto (I_x, \delta_x)$ **hides** $\mathrm{Log}(\mathcal{O}^*)$, where
  - y minimum with $\mathrm{Log}(y)$ "downwards closest" to x

    (appropriate choice if more)
  - $I_x = y^{-1}\mathcal{O}$ reduced ideal (by HNF)

Unit groups - neighboring minima

- Buchmann's algorithm
  - Given $x \in \mathbb{R}^r$,
  - List:

    (compact representations for)

    minima y s.t.

    $\mathrm{Log}(y)$ close to x.
  - in polynomial time for constant degree

    (polynomially many such y)
- Hiding function $F : \mathbb{R}^r \ni x \mapsto (I_x, \delta_x)$ **hides** $\mathrm{Log}(\mathcal{O}^*)$, where
  - y minimum with $\mathrm{Log}(y)$ "downwards closest" to x

    (appropriate choice if more)
  - $I_x = y^{-1}\mathcal{O}$ reduced ideal (by HNF)
  - $\delta_x = x - \mathrm{Log}(y)$

Hidden lattice in $\mathbb{R}^r$

- function $f : \mathbb{R}^r \longrightarrow *$

Hidden lattice in $\mathbb{R}^r$

- function $f : \mathbb{R}^r \longrightarrow *$
- hides full lattice L:

Hidden lattice in $\mathbb{R}^r$

- function $f : \mathbb{R}^r \longrightarrow *$
- hides full lattice L:

$$f(x) = f(y) \Leftrightarrow x - y \in L$$

Hidden lattice in $\mathbb{R}^r$

- function $f : \mathbb{R}^r \longrightarrow *$
- hides full lattice L:

$$f(x) = f(y) \Leftrightarrow x - y \in L$$

need good discretized version on $\frac{1}{N}\mathbb{Z}$

Hidden lattice in $\mathbb{R}^r$

- function $f : \mathbb{R}^r \longrightarrow *$
- hides full lattice L:

$$f(x) = f(y) \Leftrightarrow x - y \in L$$

need good discretized version on $\frac{1}{N}\mathbb{Z}$

- Task: Find approx. basis of L

Hidden lattice in $\mathbb{R}^r$

- function $f : \mathbb{R}^r \longrightarrow *$
- hides full lattice L:

$$f(x) = f(y) \Leftrightarrow x - y \in L$$

need good discretized version on $\frac{1}{N}\mathbb{Z}$

- Task: Find approx. basis of L
- Fourier sampling finds (dual) basis in in poly time if

Hidden lattice in $\mathbb{R}^r$

- function $f : \mathbb{R}^r \longrightarrow *$
- hides full lattice L:

$$f(x) = f(y) \Leftrightarrow x - y \in L$$

need good discretized version on $\frac{1}{N}\mathbb{Z}$

- Task: Find approx. basis of L
- Fourier sampling finds (dual) basis in in poly time if
  - r is constant

Hidden lattice in $\mathbb{R}^r$

- function $f : \mathbb{R}^r \longrightarrow *$
- hides full lattice L:

$$f(x) = f(y) \Leftrightarrow x - y \in L$$

need good discretized version on $\frac{1}{N}\mathbb{Z}$

- Task: Find approx. basis of L
- Fourier sampling finds (dual) basis in in poly time if
  - r is constant
  - L is not very "ill-positioned"

Hidden lattice in $\mathbb{R}^r$

- function $f : \mathbb{R}^r \longrightarrow *$
- hides full lattice L:

$$f(x) = f(y) \Leftrightarrow x - y \in L$$

need good discretized version on $\frac{1}{N}\mathbb{Z}$

- Task: Find approx. basis of L
- Fourier sampling finds (dual) basis in in poly time if
  - r is constant
  - L is not very "ill-positioned"
- *Details: Hallgren, Vollmer–Schmidt, STOC 2005.*

Hidden lattice in $\mathbb{R}^r$

- function $f : \mathbb{R}^r \longrightarrow *$
- hides full lattice L:

$$f(x) = f(y) \Leftrightarrow x - y \in L$$

need good discretized version on $\frac{1}{N}\mathbb{Z}$

- Task: Find approx. basis of L
- Fourier sampling finds (dual) basis in in poly time if
  - r is constant
  - L is not very "ill-positioned"
- *Details: Hallgren, Vollmer–Schmidt, STOC 2005.*

  Real quadratic case: Jozsa 2003.

Hidden lattice in $\mathbb{R}^r$

- function $f : \mathbb{R}^r \to *$
- hides full lattice L:

$$f(x) = f(y) \Leftrightarrow x - y \in L$$

need good discretized version on $\frac{1}{N}\mathbb{Z}$

- Task: Find approx. basis of L
- Fourier sampling finds (dual) basis in in poly time if
  - r is constant
  - L is not very "ill-positioned"
- *Details: Hallgren, Vollmer–Schmidt, STOC 2005.*

  Real quadratic case: Jozsa 2003.
- approx. basis of $\mathrm{Log}(\mathcal{O}^*) \to$

Hidden lattice in $\mathbb{R}^r$

- function $f : \mathbb{R}^r \longrightarrow *$
- hides full lattice L:

$$f(x) = f(y) \Leftrightarrow x - y \in L$$

need good discretized version on $\frac{1}{N}\mathbb{Z}$

- Task: Find approx. basis of L
- Fourier sampling finds (dual) basis in in poly time if
  - r is constant
  - L is not very "ill-positioned"
- *Details: Hallgren, Vollmer–Schmidt, STOC 2005.*

  Real quadratic case: Jozsa 2003.
- approx. basis of $\mathrm{Log}(\mathcal{O}^*) \longrightarrow$

  compact repr. of generators for $\mathcal{O}^*$ (Thiel).

Principal ideal

- I principal, if $I = a\mathcal{O}$ for some $a \in K^*$
  a must be a minimum of I
- Task: given I (by HNF), find a s.t. $I = a\mathcal{O}$
  (or "not principal")
- Discrete log-like hiding function:
  - $F_I : \mathbb{Z} \times \mathbb{R}^r$
  - assume $I = a\mathcal{O}$, $I = I_\zeta$ with $\zeta = \text{Log}(a)$
  - want: $F_I(k, x) = F(I_{k\zeta - x}) = (I_{k\zeta - x}, \delta_{k\zeta - x}$
  - $k\zeta - x = \text{Log}(\text{minimum of } I^k)$ "downwards closest" to $-x$
  - this computes F_I without knowing ζ
- Hidden subgroup $\ni (1, \zeta')$ $\zeta' = \text{Log}(a')$, $I = a\mathcal{O}$.

Class group under GRH

- Thiel (94): "small" prime ideals $P_1, \ldots, P_\ell$ generate class group
- $G = \mathbb{Z}^\ell$, (quantum-valued) hiding function:

$$(k_1, \ldots, k_\ell) \mapsto |R(J)\rangle = \sum_{I \in R(J)} |I\rangle, \text{ where}$$

- $J = P_1^{k_1} \cdots P_\ell^{k_\ell}$,
- $R(J) = \{\text{reduced ideals} \sim J\}$
- computing $|J\rangle|0\rangle \mapsto |J\rangle|R(J)\rangle$:
  - $M(J) := \{\text{minima of } J\}$
  - "easy": $|J\rangle \sum_{\mu \in M(J)} |\mu\rangle|\mu^{-1}J\rangle$
  - $|J\rangle \sum_{I \in R(J)} (\sum_{\mu \in M(J): I = \mu^{-1}J} |\mu\rangle|J\rangle)|I\rangle$
  - term in middle term computable from I and J (principal ideal algorithm)

Open problems

- Sketched algorithms: Unit group, etc. constant degree

Open problems

- Sketched algorithms: Unit group, etc. constant degree
- Unit group, etc for non-constant degree

Open problems

- Sketched algorithms: Unit group, etc. constant degree
- Unit group, etc for non-constant degree
- Other approaches?

Open problems

- Sketched algorithms: Unit group, etc. constant degree
- Unit group, etc for non-constant degree
- Other approaches?
- Dispensing of GRH?

Open problems

- Sketched algorithms: Unit group, etc. constant degree
- Unit group, etc for non-constant degree
- Other approaches?
- Dispensing of GRH?
- Solving norm equations from L^* to K^*, where $Q \leq K < L$

Open problems

- Sketched algorithms: Unit group, etc. constant degree
- Unit group, etc for non-constant degree
- Other approaches?
- Dispensing of GRH?
- Solving norm equations from L^* to K^*, where $Q \leq K < L$
  - Hasse's local-global principle

Open problems

- Sketched algorithms: Unit group, etc. constant degree
- Unit group, etc for non-constant degree
- Other approaches?
- Dispensing of GRH?
- Solving norm equations from L^* to K^*, where $Q \leq K < L$
  - Hasse's local-global principle
    - First glance: gives "probably" identity test modulo $N(L^*)$

Open problems

- Sketched algorithms: Unit group, etc. constant degree
- Unit group, etc for non-constant degree
- Other approaches?
- Dispensing of GRH?
- Solving norm equations from L^* to K^*, where $Q \leq K < L$
  - Hasse's local-global principle
    - First glance: gives "probably" identity test modulo $N(L^*)$
    - $\exists$ "correct" non-constructive identity test

Open problems

- Sketched algorithms: Unit group, etc. constant degree
- Unit group, etc for non-constant degree
- Other approaches?
- Dispensing of GRH?
- Solving norm equations from L^* to K^*, where $Q \leq K < L$
  - Hasse's local-global principle
    - First glance: gives "probably" identity test modulo $N(L^*)$
    - $\exists$ "correct" non-constructive identity test
  - good "small" subgroup of L^* (S-units)?

Open problems

- Sketched algorithms: Unit group, etc. constant degree
- Unit group, etc for non-constant degree
- Other approaches?
- Dispensing of GRH?
- Solving norm equations from L^* to K^*, where $Q \leq K < L$
  - Hasse's local-global principle
    - First glance: gives "probably" identity test modulo $N(L^*)$
    - $\exists$ "correct" non-constructive identity test
  - good "small" subgroup of L^* (S-units)?
- Further HSP-like problems in algebraic number theory?

Open problems

- Sketched algorithms: Unit group, etc. constant degree
- Unit group, etc for non-constant degree
- Other approaches?
- Dispensing of GRH?
- Solving norm equations from L^* to K^*, where $Q \leq K < L$
  - Hasse's local-global principle
    - First glance: gives "probably" identity test modulo $N(L^*)$
    - $\exists$ "correct" non-constructive identity test
  - good "small" subgroup of L^* (S-units)?
- Further HSP-like problems in algebraic number theory?
- Your favorite number theory problem?