Quantum Automata for Online Minimization Problems

Kamil Khadiev[1,2], Aliya Khadieva[2]

[1] University of Latvia, Riga, Latvia,
[2] Kazan Federal University, Kazan, Russia

NCMA-2017 - Prague, 2017

Content

- Definitions.

Content

- Definitions.
- Results.

Content

- Definitions.
- Results.
- Main Ideas.

Online Minimization Problem

Online Minimization Problem

Online Minimization Problem

- $\mathcal{I}$ is set of possible inputs; $I \in \mathcal{I}$ is input; $I = (x_1, \ldots, x_n)$;

Online Minimization Problem

Online Minimization Problem

- $\mathcal{I}$ is set of possible inputs; $I \in \mathcal{I}$ is input; $I = (x_1, \ldots, x_n)$;
- $O = (y_1, \ldots, y_n)$ is feasible output for I;

Online Minimization Problem

Online Minimization Problem

- $\mathcal{I}$ is set of possible inputs; $I \in \mathcal{I}$ is input; $I = (x_1, \ldots, x_n)$;
- $O = (y_1, \ldots, y_n)$ is feasible output for I;
- $cost(I, O)$ is cost of output O for input I;

Online Minimization Problem

Online Minimization Problem

- $\mathcal{I}$ is set of possible inputs; $I \in \mathcal{I}$ is input; $I = (x_1, \ldots, x_n)$;
- $O = (y_1, \ldots, y_n)$ is feasible output for I;
- $cost(I, O)$ is cost of output O for input I;
- O' is optimal, if $cost(I, O')$ is smallest.

Online Minimization Problem

Online Minimization Problem

- $\mathcal{I}$ is set of possible inputs; $I \in \mathcal{I}$ is input; $I = (x_1, \ldots, x_n)$;
- $O = (y_1, \ldots, y_n)$ is feasible output for I;
- $cost(I, O)$ is cost of output O for input I;
- O' is optimal, if $cost(I, O')$ is smallest.

Deterministic Online Algorithm

- A computes $A(I) = (y_1, \ldots, y_n)$;

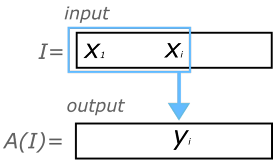
Online Minimization Problem

Online Minimization Problem

- $\mathcal{I}$ is set of possible inputs; $I \in \mathcal{I}$ is input; $I = (x_1, \ldots, x_n)$;
- $O = (y_1, \ldots, y_n)$ is feasible output for I;
- $cost(I, O)$ is cost of output O for input I;
- O' is optimal, if $cost(I, O')$ is smallest.

Deterministic Online Algorithm

- A computes $A(I) = (y_1, \ldots, y_n)$;
- y_i depends on $x_1, \ldots, x_i$.

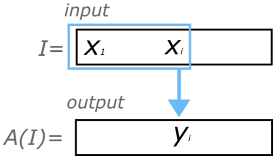


Online Minimization Problem

- $\mathcal{I}$ is set of possible inputs; $I \in \mathcal{I}$ is input; $I = (x_1, \ldots, x_n)$;
- $O = (y_1, \ldots, y_n)$ is feasible output for I;
- $cost(I, O)$ is cost of output O for input I;
- O' is optimal, if $cost(I, O')$ is smallest.

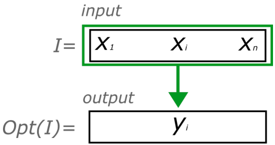
Deterministic Online Algorithm

- A computes $A(I) = (y_1, \ldots, y_n)$;
- y_i depends on $x_1, \ldots, x_i$.



Competitive Ratio

- $Opt(I)$ is optimal offline solution;



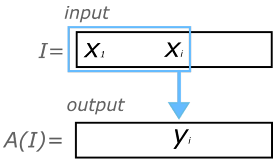
Online Minimization Problem

Online Minimization Problem

- $\mathcal{I}$ is set of possible inputs; $I \in \mathcal{I}$ is input; $I = (x_1, \ldots, x_n)$;
- $O = (y_1, \ldots, y_n)$ is feasible output for I;
- $cost(I, O)$ is cost of output O for input I;
- O' is optimal, if $cost(I, O')$ is smallest.

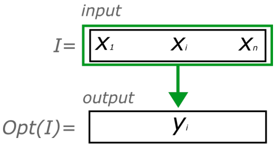
Deterministic Online Algorithm

- A computes $A(I) = (y_1, \ldots, y_n)$;
- y_i depends on $x_1, \ldots, x_i$.



Competitive Ratio

- $Opt(I)$ is optimal offline solution;
- A is c-competitive, if there is constant $\alpha > 0$ such that for any $I \in \mathcal{I}$: $cost(I, A(I)) \leq c \cdot cost(I, Opt(I)) + \alpha$.



Randomized Online Algorithms

Randomized Online Algorithm

Randomized Online Algorithms

Randomized Online Algorithm

- $R^\psi := R^\psi(I) = (y_1, \cdots, y_n)$;

Randomized Online Algorithms

Randomized Online Algorithm

- $R^{\psi} := R^{\psi}(I) = (y_1, \cdots, y_n)$;
- y_i depends on $\psi, x_1, \ldots, x_i$;

Randomized Online Algorithms

Randomized Online Algorithm

- $R^{\psi} := R^{\psi}(I) = (y_1, \cdots, y_n)$;
- y_i depends on $\psi, x_1, \ldots, x_i$;
- ψ is the content of a random tape;

Randomized Online Algorithms

Randomized Online Algorithm

- $R^\psi := R^\psi(I) = (y_1, \cdots, y_n)$;
- y_i depends on $\psi, x_1, \ldots, x_i$;
- ψ is the content of a random tape;
- R is c-competitive, if $\exists \alpha > 0 : \forall I \in \mathcal{I}$:
  $\mathbb{E}[cost(I, R^\psi(I))] \leq c \cdot cost(I, Opt(I)) + \alpha$.

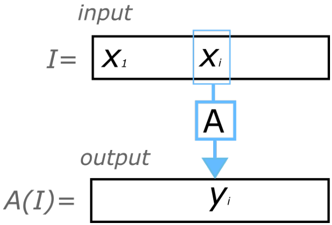
Randomized Online Algorithms

Randomized Online Algorithm

- $R^{\psi} := R^{\psi}(I) = (y_1, \cdots, y_n)$;
- y_i depends on $\psi, x_1, \ldots, x_i$;
- ψ is the content of a random tape;
- R is c-competitive, if $\exists \alpha > 0 : \forall I \in \mathcal{I}$:
  $\mathbb{E}[cost(I, R^{\psi}(I))] \leq c \cdot cost(I, Opt(I)) + \alpha$.

We use automata as an algorithm

- We use automata as an algorithm.



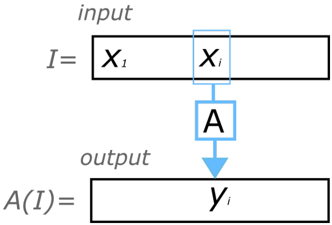
Randomized Online Algorithms

Randomized Online Algorithm

- $R^\psi := R^\psi(I) = (y_1, \cdots, y_n)$;
- y_i depends on $\psi, x_1, \ldots, x_i$;
- ψ is the content of a random tape;
- R is c-competitive, if $\exists \alpha > 0 : \forall I \in \mathcal{I}$:
  $\mathbb{E}[cost(I, R^\psi(I))] \leq c \cdot cost(I, Opt(I)) + \alpha$.

We use automata as an algorithm

- We use automata as an algorithm.
- $|S| = o(n)$;
- $|S| = n^{O(1)}$.



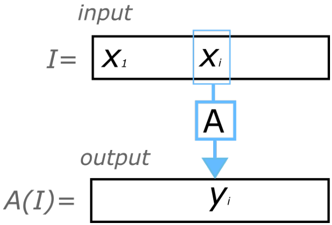
Randomized Online Algorithms

Randomized Online Algorithm

- $R^{\psi} := R^{\psi}(I) = (y_1, \cdots, y_n)$;
- y_i depends on $\psi, x_1, \ldots, x_i$;
- ψ is the content of a random tape;
- R is c-competitive, if $\exists \alpha > 0 : \forall I \in \mathcal{I}$:
  $\mathbb{E}[cost(I, R^{\psi}(I))] \leq c \cdot cost(I, Opt(I)) + \alpha$.

We use automata as an algorithm

- We use automata as an algorithm.
- $|S| = o(n)$;
- $|S| = n^{O(1)}$.

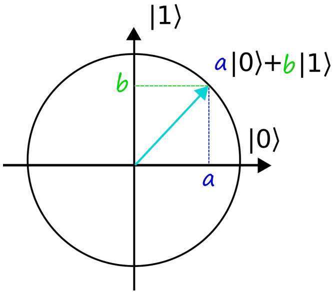


- J. Boyar, S. Irani, and K. Larsen, 2009;
- Y. Giannakopoulos and E. Koutsoupias, 2015;
- J. Boyar, K. Larsen, and A. Maiti, 2015.

Quantum Bit

Quantum bit

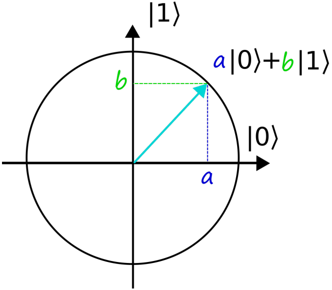
- State: $a|0\rangle + b|1\rangle$;



Quantum Bit

Quantum bit

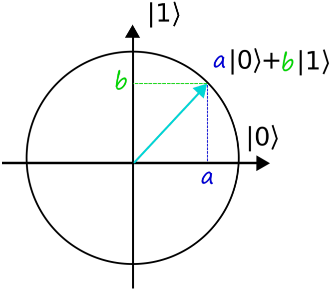
- State: $a|0\rangle + b|1\rangle$;
- a, b are complex numbers; $|a|^2 + |b|^2 = 1$;



Quantum Bit

Quantum bit

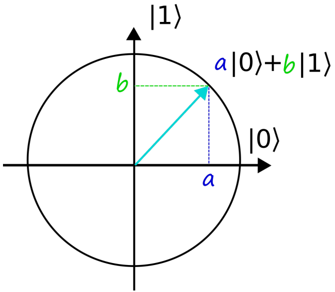
- State: $a|0\rangle + b|1\rangle$;
- a, b are complex numbers; $|a|^2 + |b|^2 = 1$;
- unitary transition: U is 2×2 unitary matrix;



Quantum Bit

Quantum bit

- State: $a|0\rangle + b|1\rangle$;
- a, b are complex numbers; $|a|^2 + |b|^2 = 1$;
- unitary transition: U is 2×2 unitary matrix;
- measurement: $Pr\{|0\rangle\} = |a|^2$; $Pr\{|1\rangle\} = |b|^2$.



Quantum Automata as an Online Algorithm

Quantum Automata

- t qubits, $|S| = d = 2^t$ states;

Quantum Automata as an Online Algorithm

Quantum Automata

- t qubits, $|S| = d = 2^t$ states;
- $|\psi\rangle$ is d-dimensional complex-value vector of amplitudes;

Quantum Automata as an Online Algorithm

Quantum Automata

- t qubits, $|S| = d = 2^t$ states;
- $|\psi\rangle$ is d-dimensional complex-value vector of amplitudes;
- $\sum_{i=1}^{d} \left| |\psi\rangle[i] \right|^2 = 1$;

Quantum Automata as an Online Algorithm

Quantum Automata

- t qubits, $|S| = d = 2^t$ states;
- $|\psi\rangle$ is d-dimensional complex-value vector of amplitudes;
- $\sum_{i=1}^{d} \left||\psi\rangle[i]\right|^2 = 1$;
- unitary transition: U^0 or U^1;

Quantum Automata as an Online Algorithm

Quantum Automata

- t qubits, $|S| = d = 2^t$ states;
- $|\psi\rangle$ is d-dimensional complex-value vector of amplitudes;
- $\sum_{i=1}^{d} \left| |\psi\rangle[i] \right|^2 = 1$;
- unitary transition: U^0 or U^1;
- we can measure some of qubits.

Quantum Automata as an Online Algorithm

Quantum Automata

- t qubits, $|S| = d = 2^t$ states;
- $|\psi\rangle$ is d-dimensional complex-value vector of amplitudes;
- $\sum_{i=1}^{d} \left| |\psi\rangle[i] \right|^2 = 1$;
- unitary transition: U^0 or U^1;
- we can measure some of qubits.

Quantum Online Algorithm

- $Q := Q(I) = (y_1, \cdots, y_n)$;

Quantum Automata as an Online Algorithm

Quantum Automata

- t qubits, $|S| = d = 2^t$ states;
- $|\psi\rangle$ is d-dimensional complex-value vector of amplitudes;
- $\sum_{i=1}^{d} \left||\psi\rangle[i]\right|^2 = 1$;
- unitary transition: U^0 or U^1;
- we can measure some of qubits.

Quantum Online Algorithm

- $Q := Q(I) = (y_1, \cdots, y_n)$;
- y_i depends on $x_1, \ldots, x_i, y_1, \ldots, y_{i-1}$;

Quantum Automata as an Online Algorithm

Quantum Automata

- t qubits, $|S| = d = 2^t$ states;
- $|\psi\rangle$ is d-dimensional complex-value vector of amplitudes;
- $\sum_{i=1}^{d} \left||\psi\rangle[i]\right|^2 = 1$;
- unitary transition: U^0 or U^1;
- we can measure some of qubits.

Quantum Online Algorithm

- $Q := Q(I) = (y_1, \cdots, y_n)$;
- y_i depends on $x_1, \dots, x_i, y_1, \dots, y_{i-1}$;
- Q can measure qubits several times during computation;

Quantum Automata as an Online Algorithm

Quantum Automata

- t qubits, $|S| = d = 2^t$ states;
- $|\psi\rangle$ is d-dimensional complex-value vector of amplitudes;
- $\sum_{i=1}^{d} \left||\psi\rangle[i]\right|^2 = 1$;
- unitary transition: U^0 or U^1;
- we can measure some of qubits.

Quantum Online Algorithm

- $Q := Q(I) = (y_1, \cdots, y_n)$;
- y_i depends on $x_1, \ldots, x_i, y_1, \ldots, y_{i-1}$;
- Q can measure qubits several times during computation;
- Q is c-competitive, if $\exists \alpha > 0 : \forall I \in \mathcal{I}$:
  $\mathbb{E}[cost(I, Q(I))] \leq c \cdot cost(I, Opt(I)) + \alpha$.

Results on Sublinear Size

Theorems 3.1,3.4,3.5

For special online minimization problem (n, k, w, r)-PNH, for $w > r$ we have:

Results on Sublinear Size

Theorems 3.1,3.4,3.5

For special online minimization problem (n, k, w, r)-PNH, for $w > r$ we have:

- There is $\frac{r+w}{2r}$-competitive in expectation quantum automaton Q with 2 quantum states.

Results on Sublinear Size

Theorems 3.1,3.4,3.5

For special online minimization problem (n,k,w,r)-PNH, for $w>r$ we have:

- There is $\frac{r+w}{2r}$-competitive in expectation quantum automaton Q with 2 quantum states.
- Any deterministic automaton A with $o(n)$ states is at least $\frac{w}{r}$-competitive.

Results on Sublinear Size

Theorems 3.1,3.4,3.5

For special online minimization problem (n, k, w, r)-PNH, for $w > r$ we have:

- There is $\frac{r+w}{2r}$-competitive in expectation quantum automaton Q with 2 quantum states.
- Any deterministic automaton A with $o(n)$ states is at least $\frac{w}{r}$-competitive.
- Any probabilistic automaton R with $o(n)$ states is at least $\frac{r+7w}{8r}$-competitive in expectation.

Results on Polynomial Size

Theorems 3.6,3.8

For special online minimization problem (n, w, r)-PNEH, for $w > r$ we have:

Results on Polynomial Size

Theorems 3.6,3.8

For special online minimization problem (n, w, r)-PNEH, for $w > r$ we have:

- There is $(r(1-\varepsilon)^2/2 + w(\frac{1-\varepsilon^2}{2} + \varepsilon))/r$-competitive in expectation quantum automaton Q with $O(n)$ quantum states.

Results on Polynomial Size

Theorems 3.6,3.8

For special online minimization problem (n, w, r)-PNEH, for $w > r$ we have:

- There is $(r(1-\varepsilon)^2/2+w(\frac{1-\varepsilon^2}{2}+\varepsilon))/r$-competitive in expectation quantum automaton Q with $O(n)$ quantum states.
- Any deterministic automaton A with $2^{o(n)}$ states is $\frac{w}{r}$-competitive.

Definitions of (n, k, w, r)-PNH and (n, w, r)-PNEH

$I=$

2	X^1	2	X^2	2	X^3

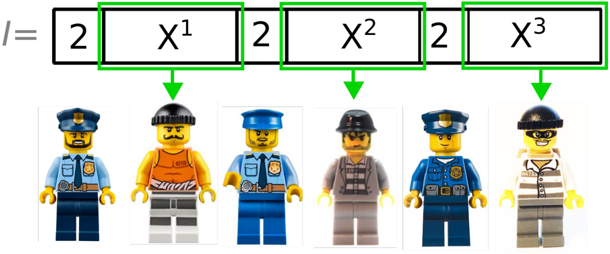
Definitions of (n, k, w, r)-PNH and (n, w, r)-PNEH

$I=$

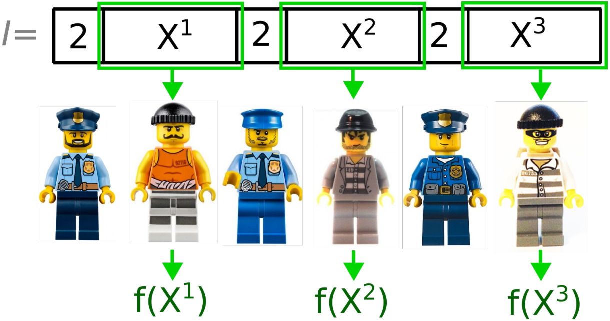
2	X^1	2	X^2	2	X^3

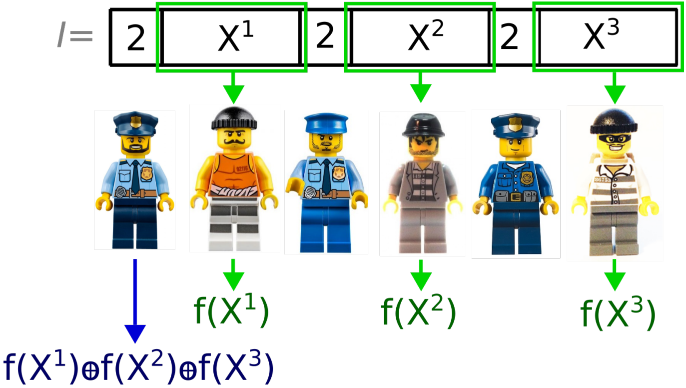


Definitions of (n,k,w,r)-PNH and (n,w,r)-PNEH

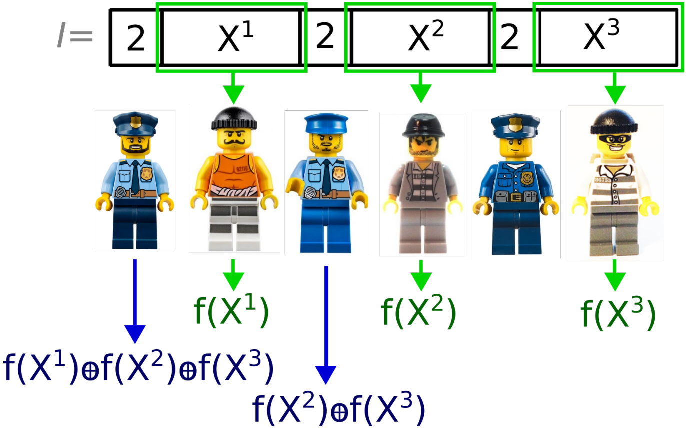


Definitions of (n, k, w, r)-PNH and (n, w, r)-PNEH

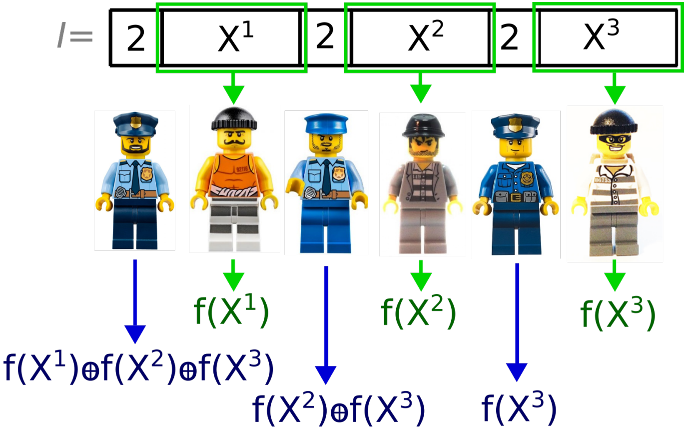



I=
2
X1
2
X2
2
X3
f(X1)
f(X2)
f(X3)
f(X1)⊕f(X2)⊕f(X3)

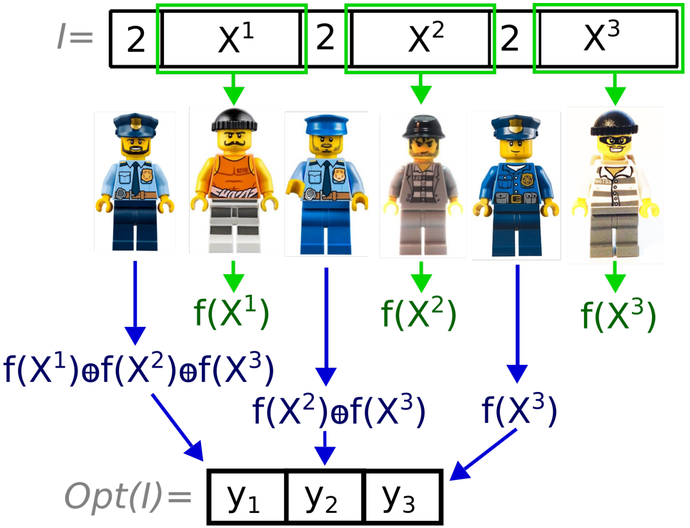
Definitions of (n, k, w, r)-PNH and (n, w, r)-PNEH



Definitions of (n, k, w, r)-PNH and (n, w, r)-PNEH

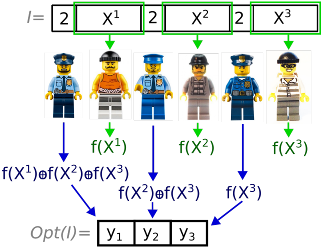


Definitions of (n, k, w, r)-PNH and (n, w, r)-PNEH



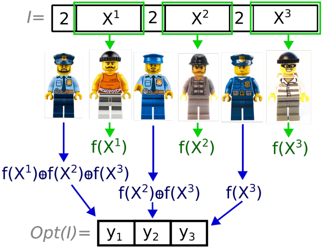
Definitions of (n, k, w, r)-PNH and (n, w, r)-PNEH

- $r < w$;



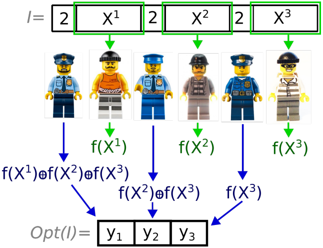
Definitions of (n, k, w, r)-PNH and (n, w, r)-PNEH

- $r < w$;
- $cost(I, O) = r$, if $y_1 = f(X_1) \oplus f(X_2) \oplus f(X_3)$, $y_2 = f(X_2) \oplus f(X_3)$ and $y_3 = f(X_3)$;



Definitions of (n,k,w,r)-PNH and (n,w,r)-PNEH

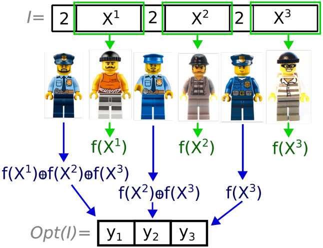
- $r < w$;
- $cost(I,O) = r$, if $y_1 = f(X_1) \oplus f(X_2) \oplus f(X_3)$, $y_2 = f(X_2) \oplus f(X_3)$ and $y_3 = f(X_3)$;
- $cost(I,O) = w$, otherwise.



Definitions of (n, k, w, r)-PNH and (n, w, r)-PNEH

- $r < w$;
- $cost(I, O) = r$, if $y_1 = f(X_1) \oplus f(X_2) \oplus f(X_3)$, $y_2 = f(X_2) \oplus f(X_3)$ and $y_3 = f(X_3)$;
- $cost(I, O) = w$, otherwise.

(n, k, w, r)-PNH: $f(X) = PartialMOD_k(X)$

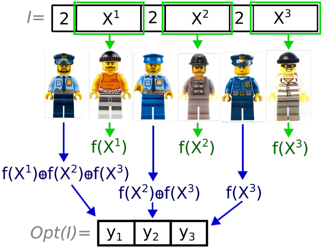


Definitions of (n,k,w,r)-PNH and (n,w,r)-PNEH

- $r < w$;
- $cost(I,O) = r$, if $y_1 = f(X_1) \oplus f(X_2) \oplus f(X_3)$, $y_2 = f(X_2) \oplus f(X_3)$ and $y_3 = f(X_3)$;
- $cost(I,O) = w$, otherwise.

(n,k,w,r)-PNH: $f(X) = PartialMOD_k(X)$

- Let $\#_1(X) = v \cdot 2^k$;

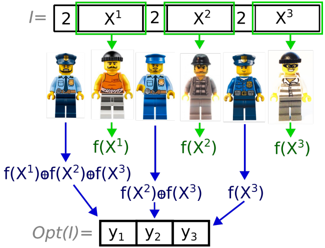


Definitions of (n,k,w,r)-PNH and (n,w,r)-PNEH

- $r < w$;
- $cost(I,O) = r$, if $y_1 = f(X_1) \oplus f(X_2) \oplus f(X_3)$, $y_2 = f(X_2) \oplus f(X_3)$ and $y_3 = f(X_3)$;
- $cost(I,O) = w$, otherwise.

(n,k,w,r)-PNH: $f(X) = PartialMOD_k(X)$

- Let $\#_1(X) = v \cdot 2^k$;
- $PartialMOD_k(X) = v \ mod \ 2$.



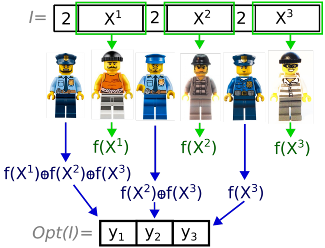
Definitions of (n,k,w,r)-PNH and (n,w,r)-PNEH

- $r < w$;
- $cost(I,O) = r$, if $y_1 = f(X_1) \oplus f(X_2) \oplus f(X_3)$, $y_2 = f(X_2) \oplus f(X_3)$ and $y_3 = f(X_3)$;
- $cost(I,O) = w$, otherwise.

(n,k,w,r)-PNH: $f(X) = PartialMOD_k(X)$

- Let $\#_1(X) = v \cdot 2^k$;
- $PartialMOD_k(X) = v \mod 2$.

(n,w,r)-PNEH: $f(X) = EQ(X)$



Definitions of (n, k, w, r)-PNH and (n, w, r)-PNEH

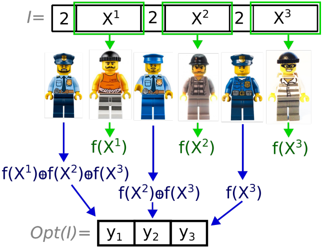
- $r < w$;
- $cost(I, O) = r$, if $y_1 = f(X_1) \oplus f(X_2) \oplus f(X_3)$, $y_2 = f(X_2) \oplus f(X_3)$ and $y_3 = f(X_3)$;
- $cost(I, O) = w$, otherwise.

(n, k, w, r)-PNH: $f(X) = PartialMOD_k(X)$

- Let $\#_1(X) = v \cdot 2^k$;
- $PartialMOD_k(X) = v \ mod \ 2$.

(n, w, r)-PNEH: $f(X) = EQ(X)$

- let $X = \sigma_1, \dots, \sigma_m, 3, \gamma_1, \dots, \gamma_t$;



Definitions of (n, k, w, r)-PNH and (n, w, r)-PNEH

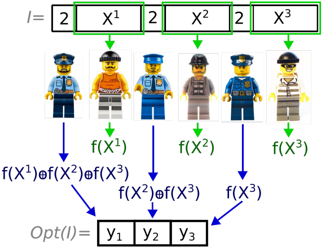
- $r < w$;
- $cost(I, O) = r$, if $y_1 = f(X_1) \oplus f(X_2) \oplus f(X_3)$, $y_2 = f(X_2) \oplus f(X_3)$ and $y_3 = f(X_3)$;
- $cost(I, O) = w$, otherwise.

(n, k, w, r)-PNH: $f(X) = PartialMOD_k(X)$

- Let $\#_1(X) = v \cdot 2^k$;
- $PartialMOD_k(X) = v \ mod \ 2$.

(n, w, r)-PNEH: $f(X) = EQ(X)$

- let $X = \sigma_1, \dots, \sigma_m, 3, \gamma_1, \dots, \gamma_t$;
- let $bin(\sigma_1, \dots, \sigma_m) = \sum_{i=1}^{m} 2^{i-1}\sigma_i$;



Definitions of (n,k,w,r)-PNH and (n,w,r)-PNEH

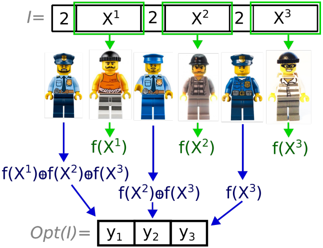
- $r < w$;
- $cost(I,O) = r$, if $y_1 = f(X_1) \oplus f(X_2) \oplus f(X_3)$, $y_2 = f(X_2) \oplus f(X_3)$ and $y_3 = f(X_3)$;
- $cost(I,O) = w$, otherwise.

(n,k,w,r)-PNH: $f(X) = PartialMOD_k(X)$

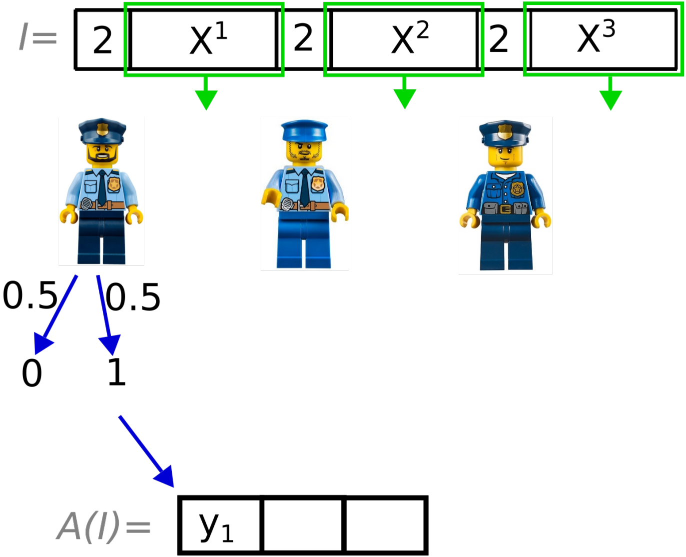
- Let $\#_1(X) = v \cdot 2^k$;
- $PartialMOD_k(X) = v \mod 2$.

(n,w,r)-PNEH: $f(X) = EQ(X)$

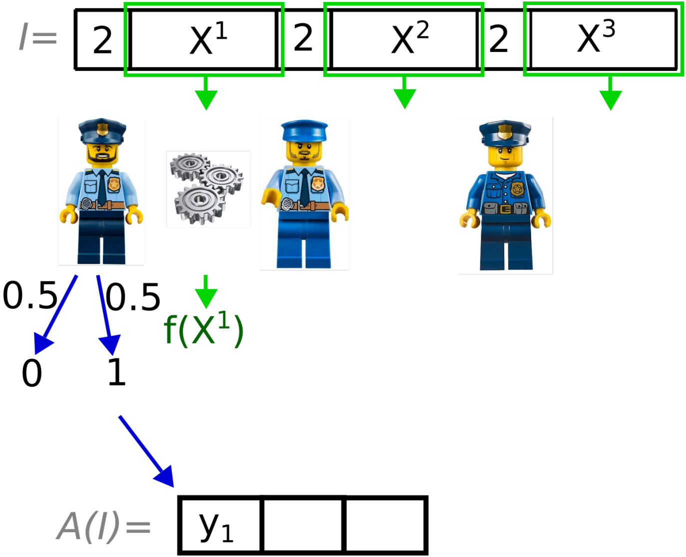
- let $X = \sigma_1,\dots,\sigma_m,3,\gamma_1,\dots,\gamma_t$;
- let $bin(\sigma_1,\dots,\sigma_m) = \sum_{i=1}^{m} 2^{i-1}\sigma_i$;
- $EQ(X) = 1$, iff $bin(\sigma_1,\dots,\sigma_m) = bin(\gamma_1,\dots,\gamma_t)$.



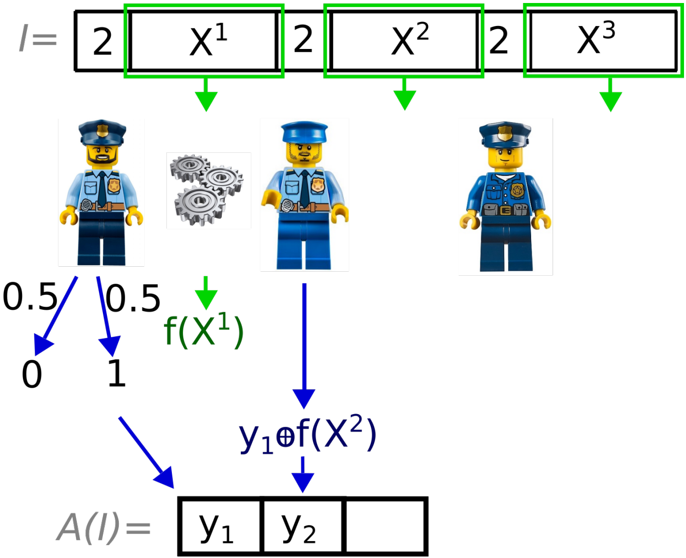
Quantum Automata for (n, k, w, r)-PNH and (n, w, r)-PNEH



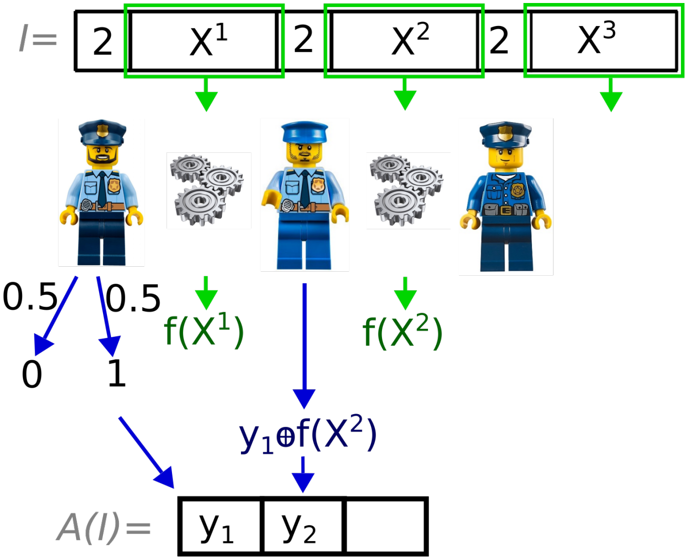
Quantum Automata for (n,k,w,r)-PNH and (n,w,r)-PNEH



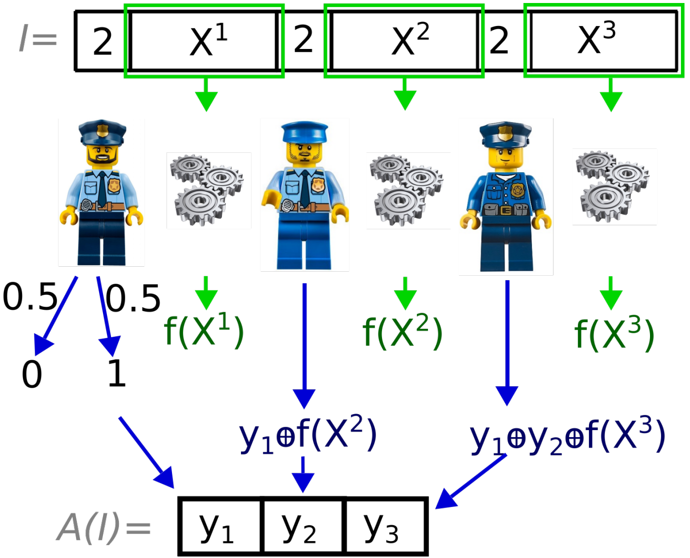
Quantum Automata for (n, k, w, r)-PNH and (n, w, r)-PNEH



Quantum Automata for (n, k, w, r)-PNH and (n, w, r)-PNEH



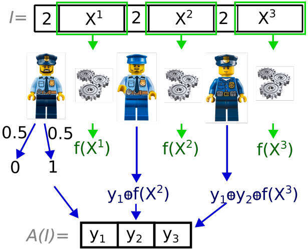
Quantum Automata for (n, k, w, r)-PNH and (n, w, r)-PNEH



Quantum Automata for (n, k, w, r)-PNH and (n, w, r)-PNEH

A. Ambainis and A. Yakaryılmaz, 2012: *PartialMOD$_k$*

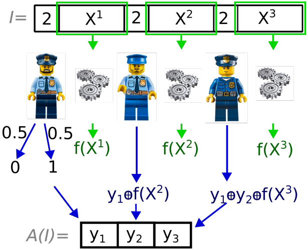
- There is 1 qubit exact quantum automaton.



Quantum Automata for (n, k, w, r)-PNH and (n, w, r)-PNEH

A. Ambainis and A. Yakaryılmaz, 2012: *PartialMOD$_k$*

- There is 1 qubit exact quantum automaton.
- No deterministic or probabilistic automaton with less than 2^k states.



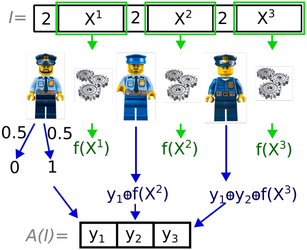
Quantum Automata for (n, k, w, r)-PNH and (n, w, r)-PNEH

A. Ambainis and A. Yakaryılmaz, 2012: *PartialMOD$_k$*

- There is 1 qubit exact quantum automaton.
- No deterministic or probabilistic automaton with less than 2^k states.

A. Ambainis and R. Freivalds, 1998; F. Ablayev and A. Vasilyev, 2009: *EQ*

- There is $n^{O(1)}$ states quantum automaton with one side error ε.
- No deterministic automaton with $2^{o(n)}$ states.



Thank you for your attention!

Děkuji!