Symmetry Effects in Computation

Andrew Chi-Chih Yao

Tsinghua University

China

1 . ITCS

Institute for Theoretical Computer Science

- To nurture and develop the talents of students and researchers
- Aims to become one of the leading research centers on theoretical computer science in the world

ITCS

- **Director: Prof. Andrew Yao**
- **Permanent academic Staff: 3**
- **Foreign Postdocs: 3**
- **Visiting Professors: 2**
- **Chair Professors: 21**
- **Visitors: 94 (2004-2007)**
- **PhD Students: 26**
- **Partners:**
  - **Chinese U of HK**
- **Space: 1500 square meters**

ITCS

2. Teaching

Tsinghua－Microsoft Special Pilot CS Class

--- Bill Gates announced this new joint program on April 19,2007

--- 30 New Students per year

Tsinghua－Microsoft Special Pilot CS Class

Tsinghua—Microsoft Special Pilot CS Class

3 . Research

- **Five Projects**
  - **973: 1**
  - **863: 1**
  - **NSFC: 3**

Research (Cont.)

1. A Study of Some Key Problems in the Theory for Secure Computation (The National Basic Research Program of China (973) (No. 2007 CB 807900)

2. Study on Agricultural Bio-Environmental Information Acquisition and Wireless Sensors Network Technologies (Hi-Tech research & Development Program of China (863 Project)(No.2006AA10Z216)

Projects (Cont.)

3. Communication Complexity and Quantum Computation (National Natural Science Foundation of China Grant ,No. 60553001)
4. Research on Reasoning and Verification of Provenancable Equipment Grid Service Chain Models (National Natural Science Foundation of China Grant, No. 60604033)
5. Study on quantum and classical decision tree complexity (National Natural Science Foundation of China Grant ,No. 60603005)

Publications

	Total	Top Conferences (FOCS, STOC,etc)
• 2007:	28	18
• 2006:	17	7
• 2005:	5	1
• 2004:	6	3

Best Paper in FOCS in 2006

Best Paper in ICALP 2007

Research Topics

(1) Algorithms

(2) Complexity Theory & Cryptography

(3) Quantum Information Processing

(4) Wireless Sensor Network

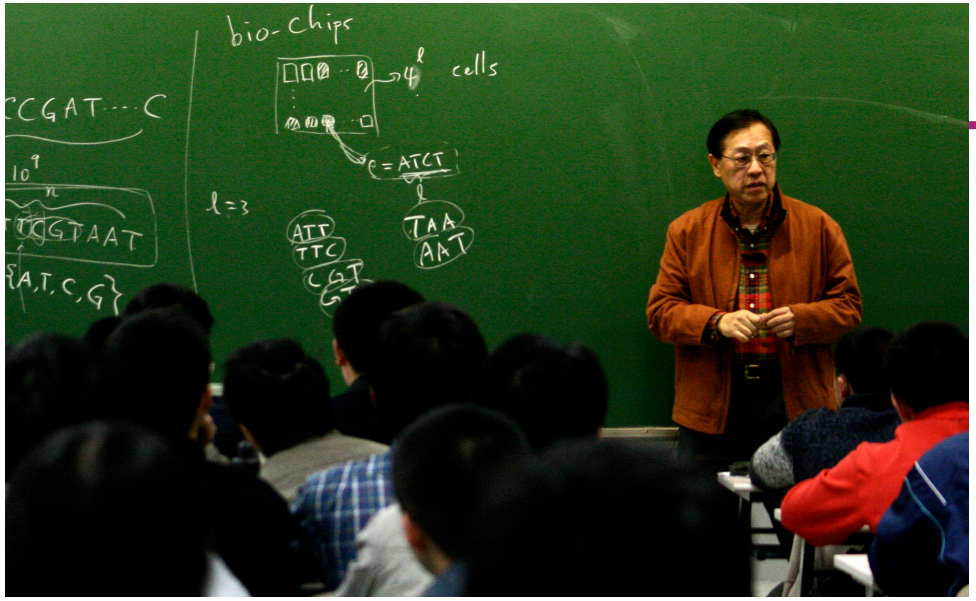
4 . *Activities*

- **PKC07**
- **China Theory Day**
- **China Theory Week**
- **Tsinghua-CUHK Theory Workshop 2007**

PKC07 (April 16-20,2007)

China Theory Days

China Theory Days

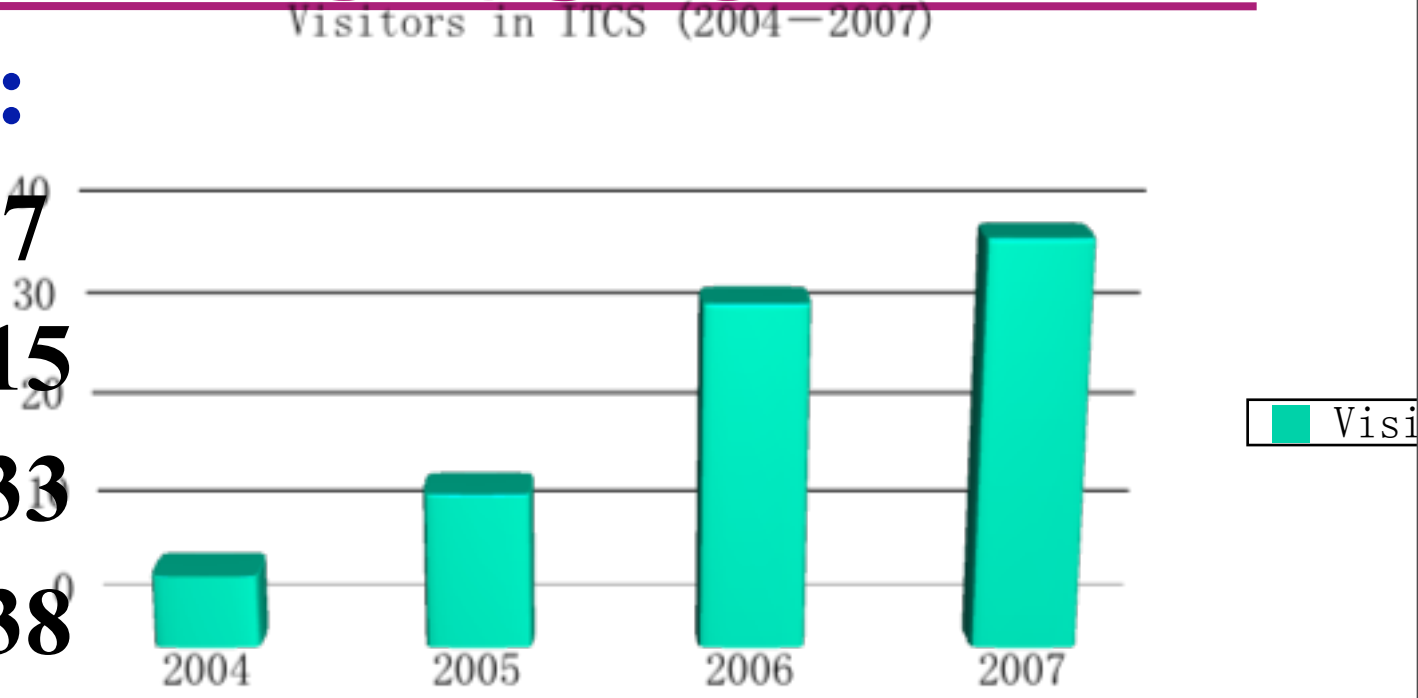


China Theory Week 2007

China Theory Week 2007

Visitors

- **Visitors:**
  - **2004: 7**
  - **2005: 15**
  - **2006: 33**
  - **2007: 38**



Symmetry Effects in

I. Introduction

- **Symmetry -- Group Theory**
  - ***The set of transformations leaving an object invariant.***

    e.g. the 5 rotations leaving a regular pentagon invariant

- **Computational Complexity**
  - ***How long it takes to compute a specific function.***

    e.g. What's the complexity of computing x^n

    (by the fastest algorithm)?

Computer x^n, where $n = 2^k$

An algorithm:

$$y_1 \leftarrow x * x$$

$$y_2 \leftarrow y_1 * y_1$$

$$y_3 \leftarrow y_2 * y_2$$

$$\ldots.$$

$$y_k \leftarrow y_{k-1} * y_{k-1} (= x^{2^k})$$

Question: Is there a faster algorithm?

Complexity Model

- **An algorithm is a sequence** $y_1, y_2, \ldots, y_m$

  $y_j \neg u \circ v$ where u, v

  are constants, x or y_i $(i < j)$

  and $\circ \in \{+, -, *, \div\}$

- **The cost of the algorithm is *m***
- **The complexity of the problem is minimum cost of any algorithm**

Complexity Model (Cont.)

Theorem: **The complexity of computing** x^{2^k} **is equal to** ***k***

Take any algorithm $y_1, y_2, \cdots, y_m$

Then y_j **is a rational function of degree** 2^j

(Proof by induction on ***j*** **)**

Hence $2^m \geq 2^k \quad \Rightarrow \quad m \geq k$

Central Difficulty in Computational Complexity:

How to prove lower bounds ?

A General Approach:

Characterize the functions which are computable in a small number of steps. Then show that our target function does not fit that characterization.

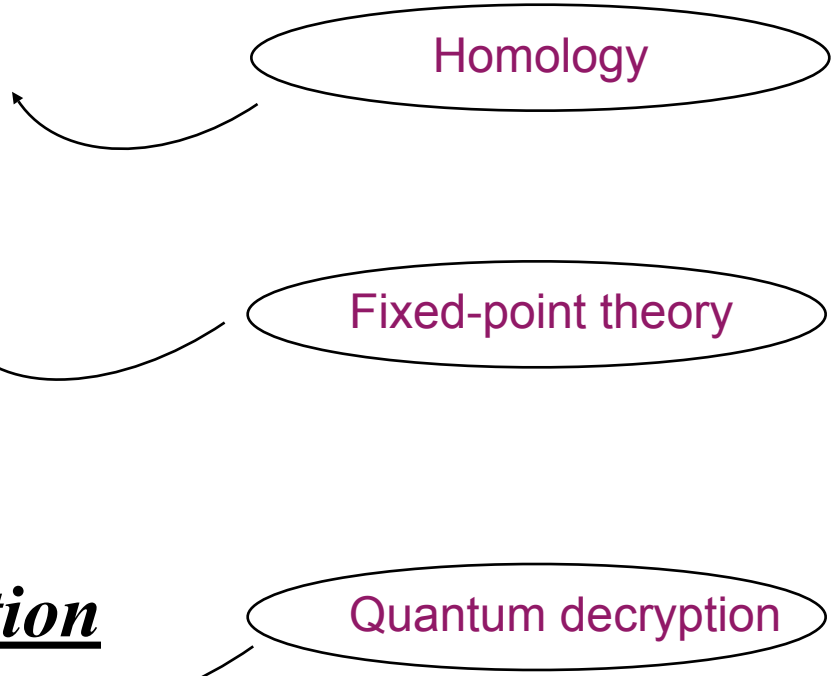
- **e.g.** ***Within j*** **steps, only polynomials of degree less than** 2^j **can be computed.**
- ***Symmetry argument*** **enters in more delicate cases.**

Fundamental Algorithms in Computer Science

- **Data Organization**
  - *search, sorting, ranking*
  - *similarity, redundancy*
- **Networks**
  - *routing, connectivity*
  - *content delivery*
- **Information Security**
  - *data encryption/decryption*
  - *protocols for secure transactions*

Fundamental Algorithms in Computer Science

- **Data Organization**
  - *search, sorting, ranking*
  - *similarity, redundancy*
- **Networks**
  - *routing, connectivity*
  - *content delivery*
- **Information Security**
  - *data encryption/decryption*
  - *protocols for secure transactions*



II. Data Organization

Membership Problem for $S \subseteq R^n$ **:**

Given input $\vec{x} \in R^n$ **, decide whether** $\vec{x} \in S$

Many computational problems can be phrased as membership problems

Example 1. Element Distinctness

Given ***n*** **numbers** $x_1, x_2, ..., x_n$ **, decide whether all of them are distinct**

$$S = \left\{ \vec{x} \;\middle|\; x_i \neq x_j \text{ for all } i \neq j \right\}$$

II. Data Organization

Example 2. k-Element Distinctness

Given *n* numbers $x_1, x_2, \ldots, x_n$, decide whether there are *k* of them with the same value

$\overline{S} \subseteq R^n$ is the set of all $\vec{x} = (x_1, x_2, \ldots, x_n)$ such that there exist $i_1 < i_2 < \cdots < i_k$ with $x_{i_1} = x_{i2} = \cdots = x_{i_k}$

For *k=2* , this reduces to *Element Distinctness*

II. Data Organization

- **An algorithm is an adaptive sequence** $y_1, y_2, \cdots$

  either $y_j \neg u \circ v$ where u, v

  are constants, x_k or y_i $(i < j)$

  and $\circ \in \{+, -, *, \div\}$

  or y_j is a branching operation

  $u : 0$ where u can be x_k or y_i $(i < j)$

- **The cost of the algorithm is the max # of operations**
- **The complexity *C(S)* for the problem is minimum cost of any algorithm**

II. Data Organization

Any relation between *C(S)* and classical properties of the set *S* ?

$\beta_0(S)$ **: rank of the homology group** $H_0(S)$

(# of path-connected components of *S*)

For Element Distinctness of n elements $\beta_0(S) = n!$

Theorem [Ben-Or 1983] $C(S) \geq \Omega(\log(\beta_0(S)))$

Corollary: The complexity for Element Distinctness is

$$\Omega(n \log n)$$

II. Data Organization

$\beta_i(S)$: **rank of the homology group** $H_i(S)$

(i-th Betti number)

Theorem [Yao 1994][Bjorner,Lovasz 1994]

$$C(S) \geq \Omega(\log(\sum_i \beta_i(S)))$$

For *k*-Element Distinctness of *n* elements

$$\log(\sum_i \beta_i(S)) = \Theta(n \log \frac{n}{k})$$

This leads to $C(S) = \Theta(n \log \frac{n}{k})$

III. Network Connectivity

Is a Network on *n* nodes *connected* ?

***2-connected* ?**

- **Graph *G* on *n* nodes can be represented by an *n* x *n* *0-1* matrix .**
- **An algorithm probes entries of the matrix until *f(G) = yes/no* can be determined.**
- **The complexity *C(f)* is the # of probes used by the smartest algorithm for deciding property *f***

III. Network Connectivity

Theorem **(Kahn, Saks, Sturtevant 1984)**

C(f) **=n(n-1)/2 for connectivity (and in fact any nontrivial monotone graph property) on *n* vertices, if *n* is a prime power.**

An approach based on Topology/Group Theory

A *simplicial complex* $\Delta \subseteq R^m$ **is a set of points decomposable into a disjoint union of simplices.**

Call Δ ***contractible*** **if it can be shrunk into a point.**

III. Network Connectivity

Proof Strategy: let $m = n(n-1)/2$

(A) Associate with f **a simplicial complex** $\Delta_f \subseteq R^{m-1}$

(B) Lemma: $C(f) < m \Rightarrow \Delta_f$ is contractible

(C) Show that Δ_f **is not contractible, using a fixed-point theorem of Oliver (1975)**

(B)+(C) $\Rightarrow \quad C(f) = n(n-1)/2$

III. Network Connectivity

Still Open:

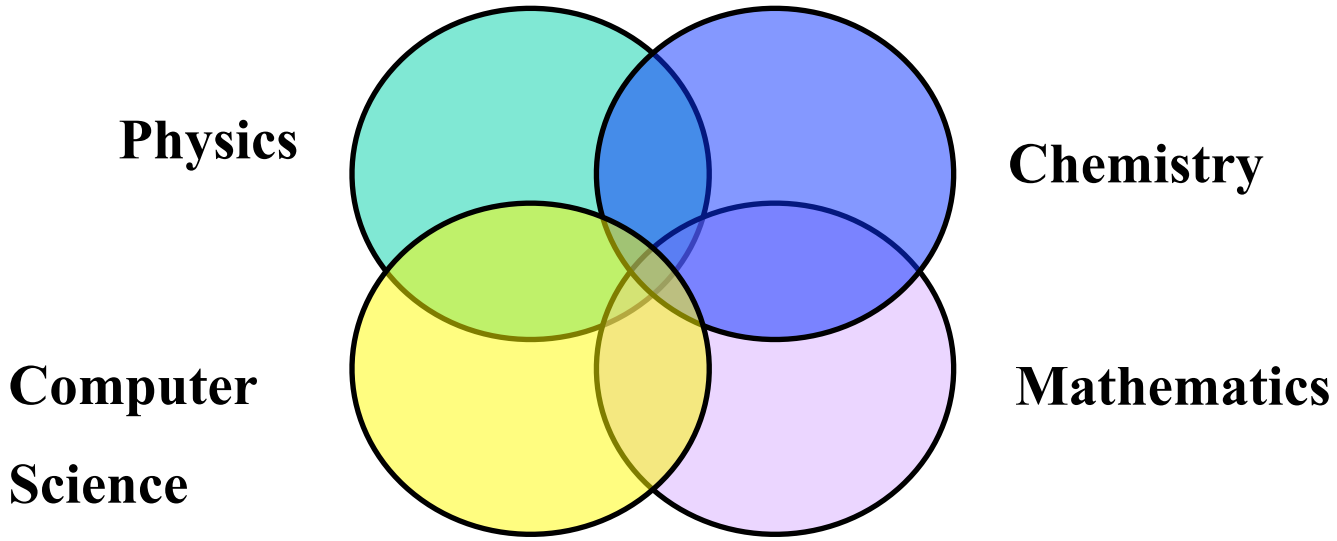
Is C(f)=n(n-1)/2 for every nontrivial monotone graph property f for every n?

Theorem [Yao 1988] **Every nontrivial monotone bipartite graph property f satisfies** $C(f)=n^2$

IV. Quantum Decryption

Quantum Computing:

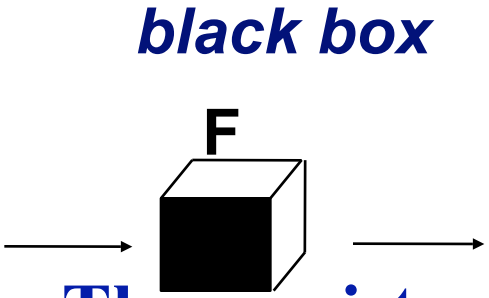
Confluence of ideas



Also, Material Science, Engineering . . .

IV. Quantum Decryption

Simon's Problem:



- ***2 to 1 mapping:*** **There exists a *secret* *s* such that**

$$F(x+s) = F(x)$$

- ***Problem:*** **Determine *s***
- ***Note:*** **classical algorithms must make exponentially many queries *F(x) = ?***

IV. Quantum Decryption

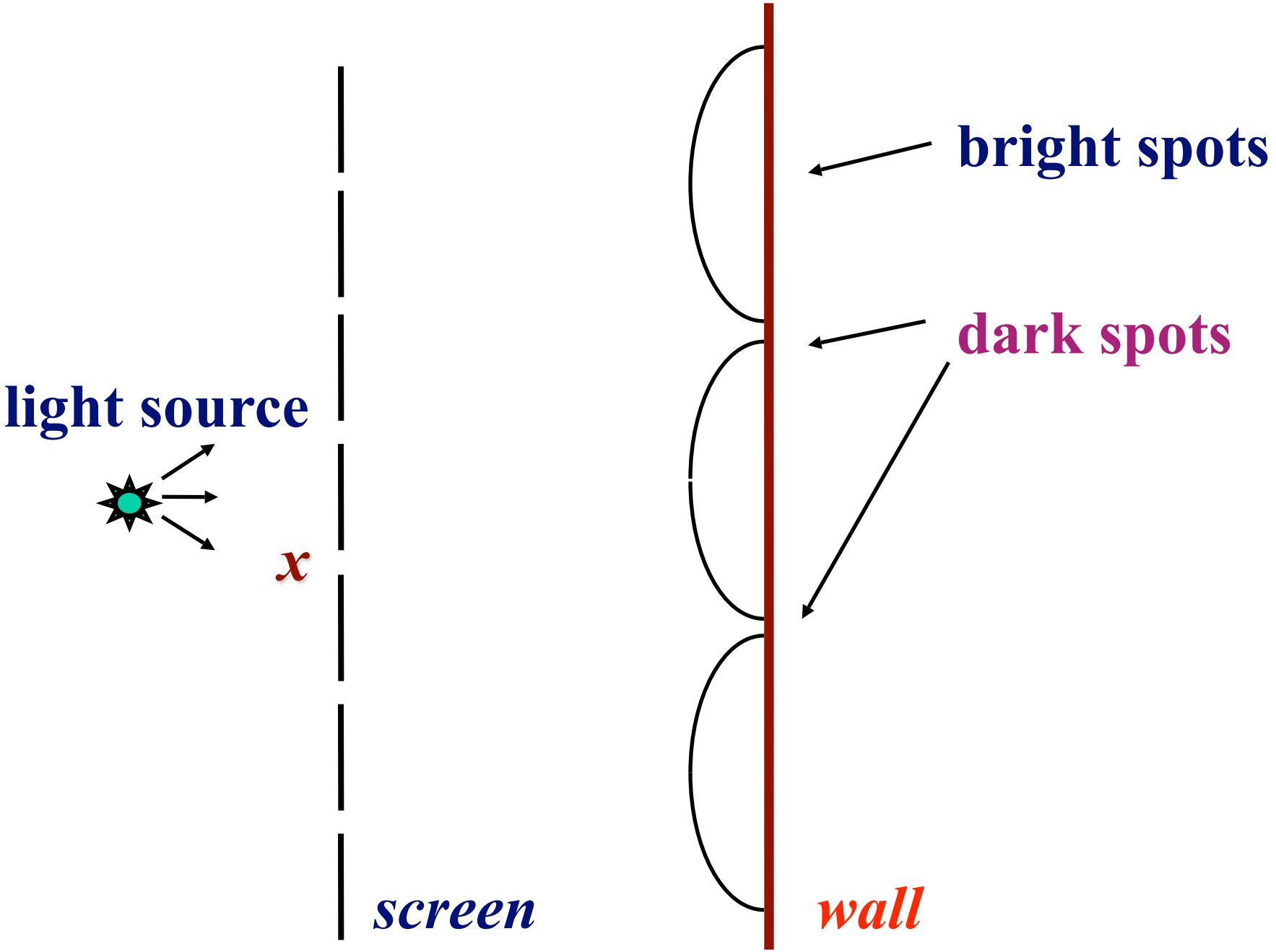
A Problem about Symmetry:

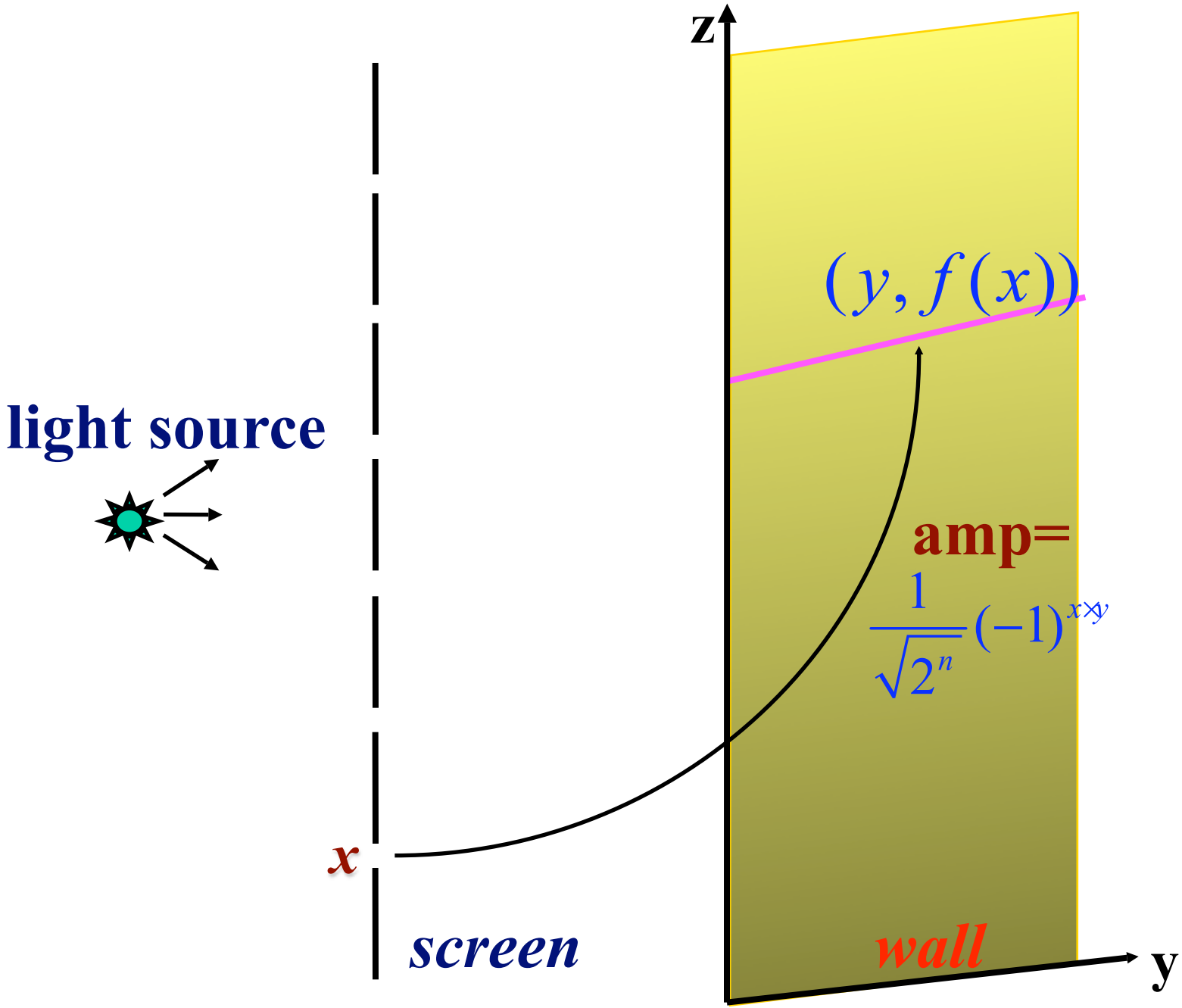
Hypercube $\{0,1\}^n$ **is an abelian group**

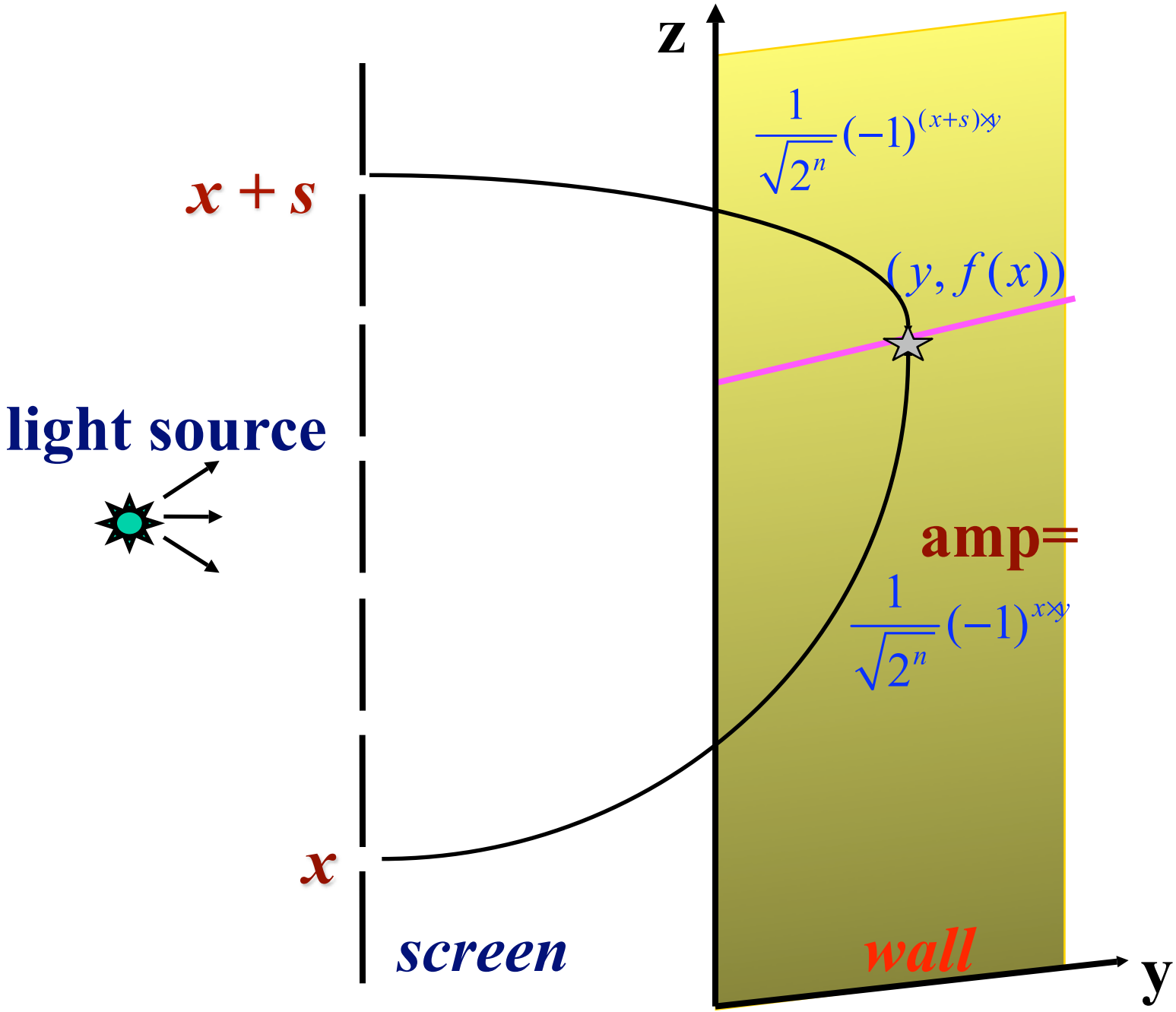
under $+$ **;** ***hidden subgroup*** $G_s = \{0, s\}$

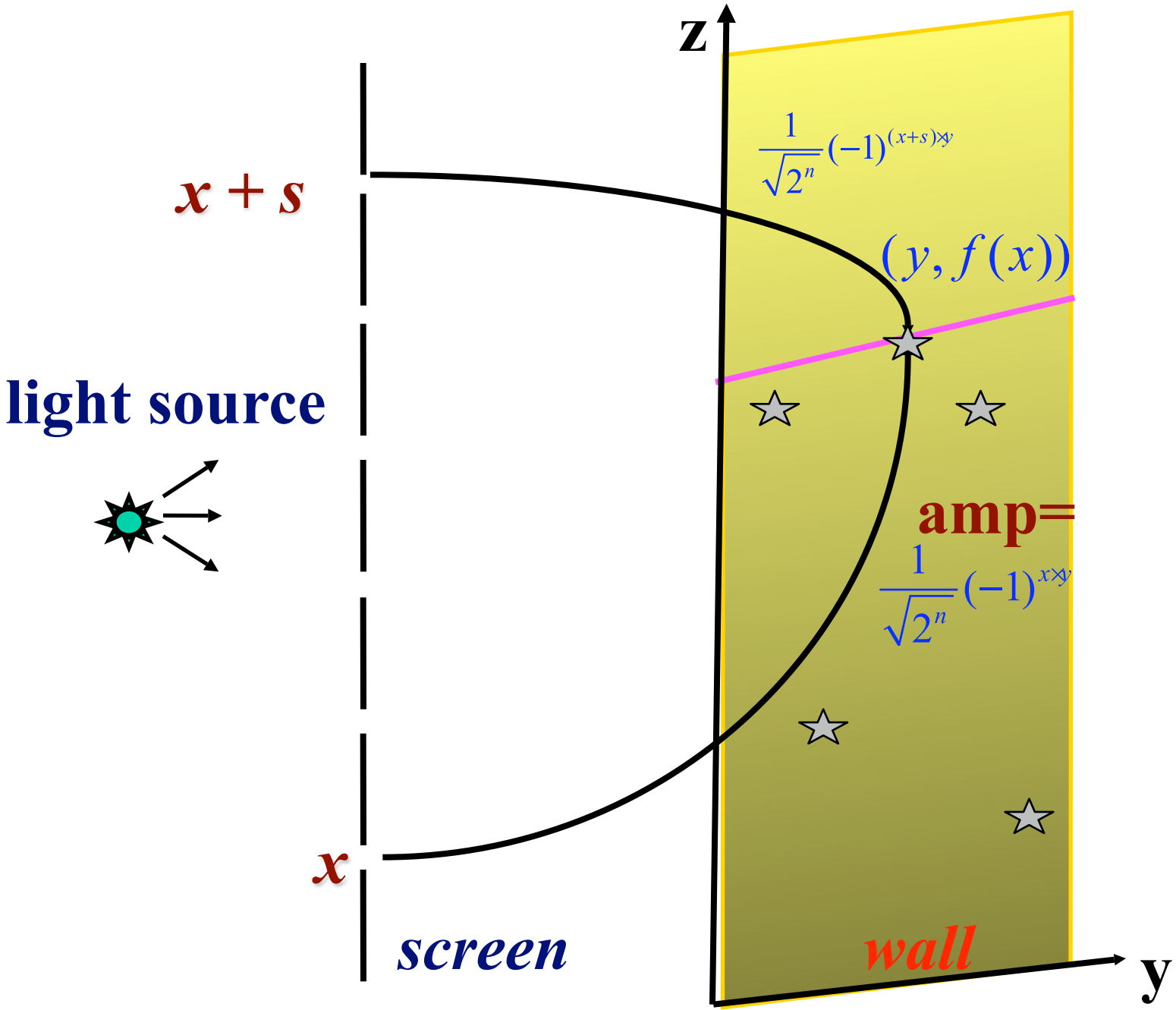
Function F **is constant on any coset** $\{x, x+s\}$

Goal**: To identify the** ***hidden subgroup*** G_s


light source
x
screen
bright spots
dark spots
wall


z
$(y, f(x))$
light source
amp=
$\frac{1}{\sqrt{2^n}}(-1)^{x \times y}$
x
screen
wall
y


z
$\frac{1}{\sqrt{2^n}}(-1)^{(x+s)\times y}$
$x+s$
$(y, f(x))$
light source
amp=
$\frac{1}{\sqrt{2^n}}(-1)^{x\times y}$
x
screen
wall
y


z
$\frac{1}{\sqrt{2^n}}(-1)^{(x+s)\times y}$
$x+s$
$(y, f(x))$
light source
amp=
$\frac{1}{\sqrt{2^n}}(-1)^{x\times y}$
x
screen
wall
y

IV. Quantum Decryption

- ***Light patterns*** **on the wall determines s**
- ***Quantum computing:***

  -- don't need 2^N holes on screen

  or 2^N x 2^N dots on the wall

  -- can be implemented with N x 2N bits

  -- each bright spot location 

  1 bit of information on s

IV. Quantum Decryption

A very important result:

Shor (1994) developed an efficient quantum algorithm for factoring large integers; His method uses an approach similar to Simon's algorithm.

$$N = p * q$$

secret

Conclusions

- ***Symmetry imposes limitation on how fast a computation can be done.***
- ***Symmetry sometimes leads to fast algorithms (i.e. quantum decryption).***
- ***Computer science is not just about hardware & programming.***
- ***How to design efficient algorithms is a deep subject related to math and physics.***

Thank You!

http://itcs.tsinghua.edu.cn