Title: Phase transitions in N-SAT

Date: Feb 10, 2006 04:00 PM

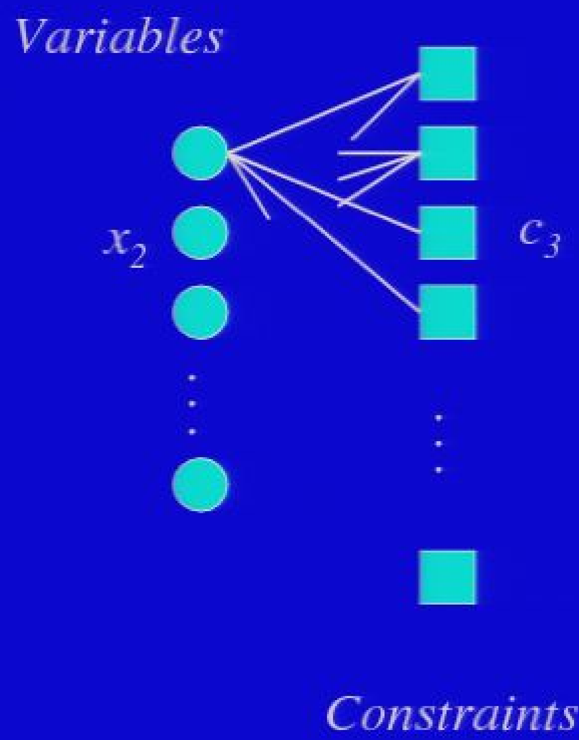
URL: http://pirsa.org/06020021

Abstract:

The Setting: Random CSPs

- n variables with small, discrete domains
- m competing constraints

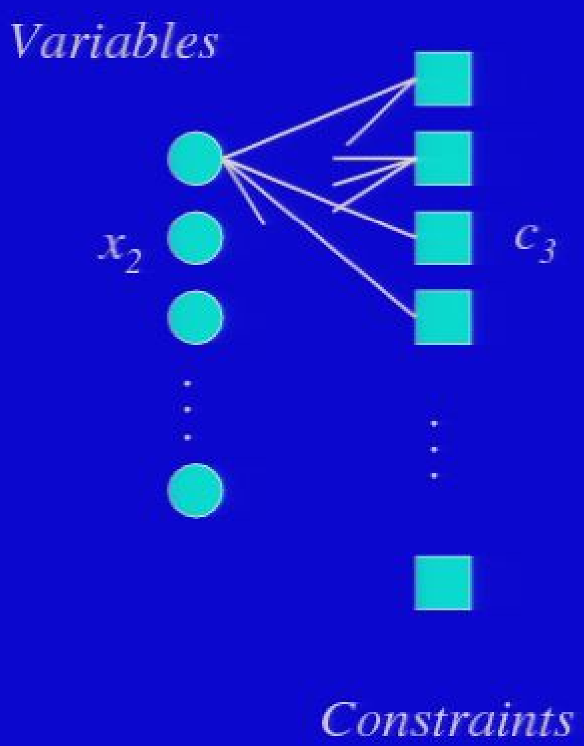
- Random bipartite graph:
- Sparse graph, i.e. m=Θ(n)



The Setting: Random CSPs

- n variables with small, discrete domains
- m competing constraints

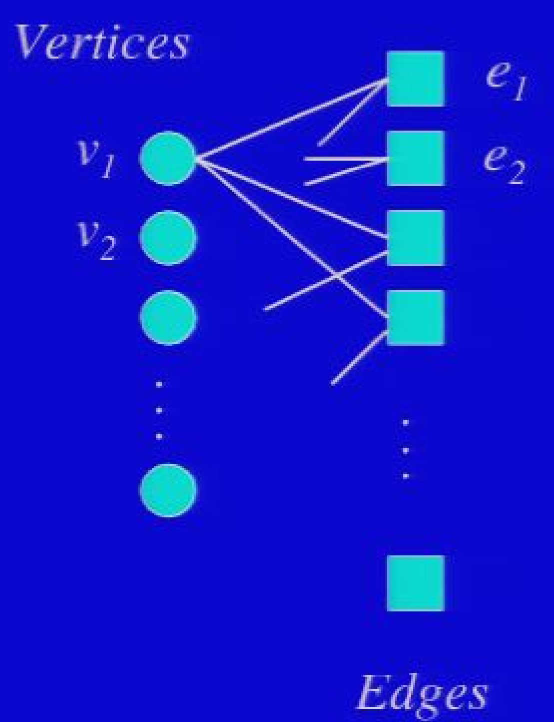
- Random bipartite graph:
- Sparse graph, i.e. m=Θ(n)

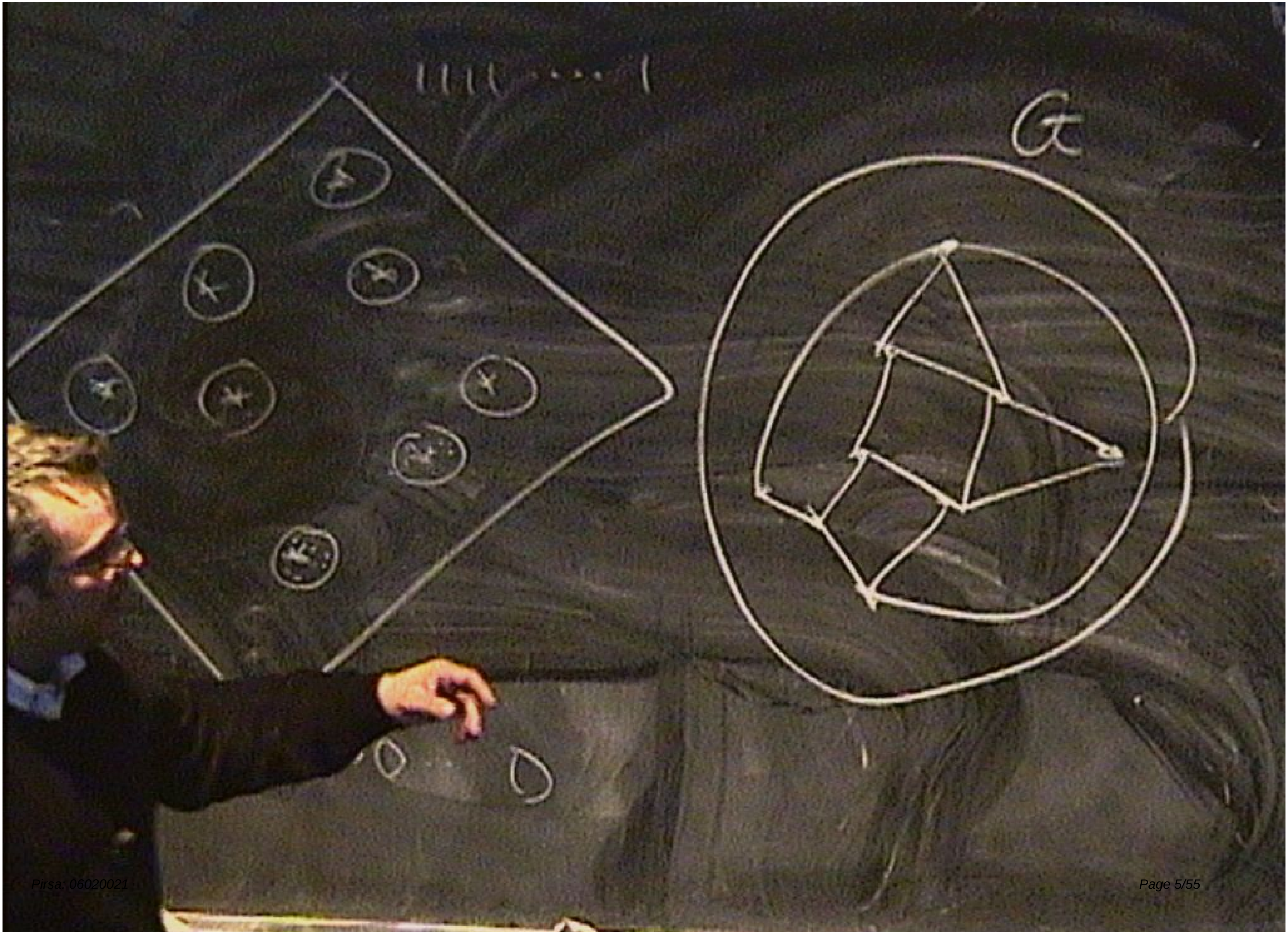


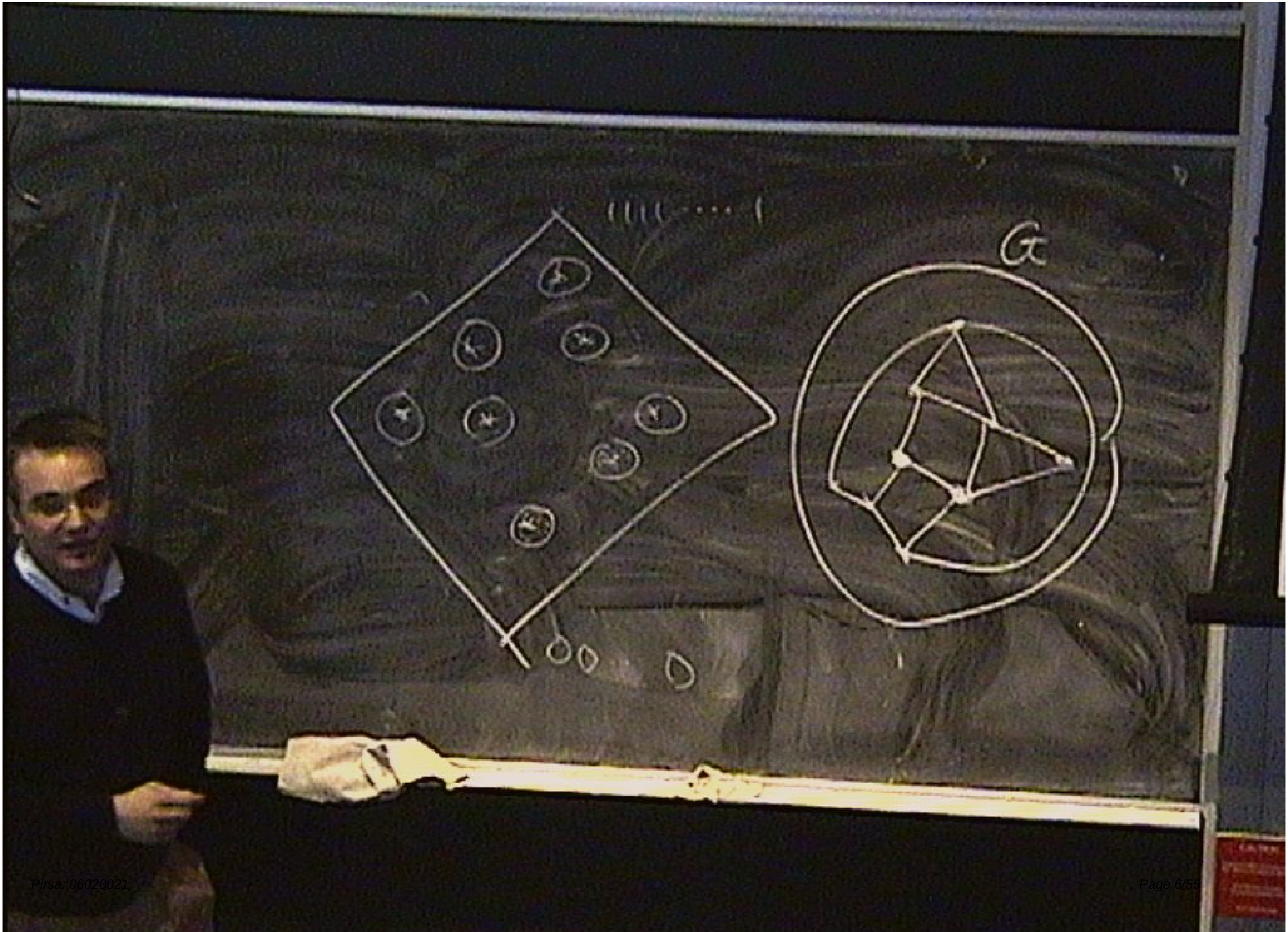
Random Graph k-coloring

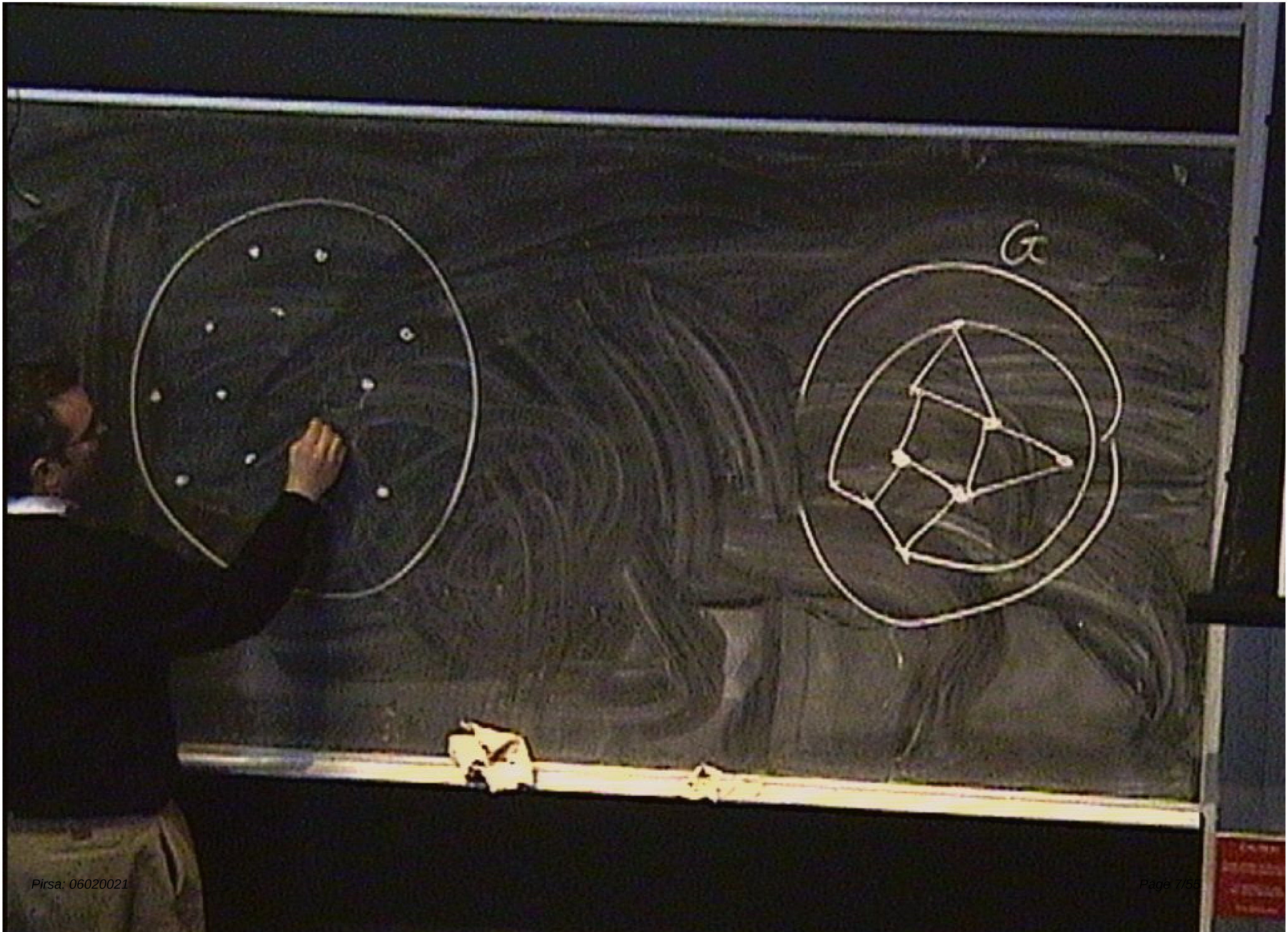
- Each vertex is a variable with domain {1,2,...,k}
- Each edge is a "not-equal" constraint on two variables

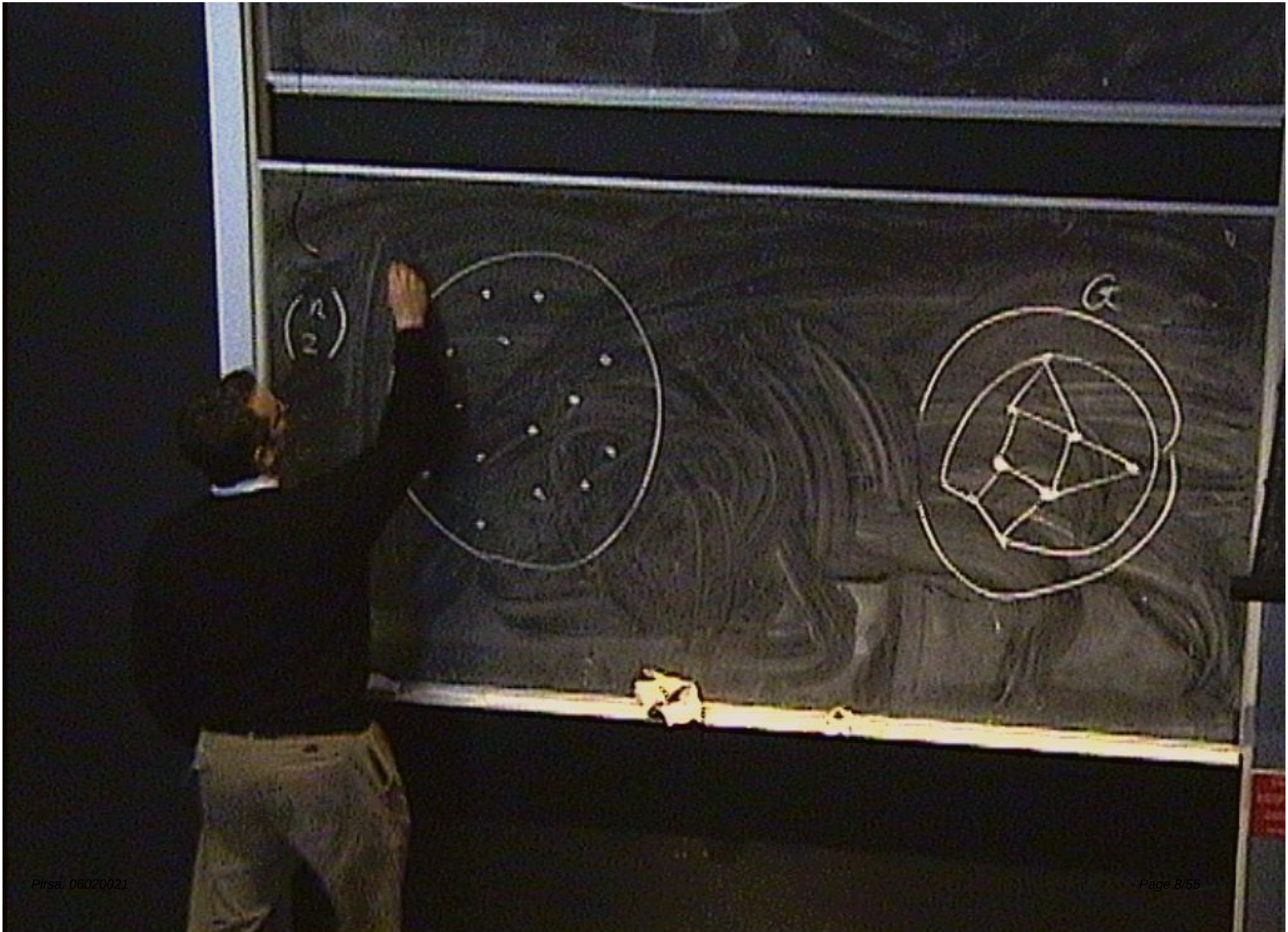
- G(n,m) random graph: the two variables are chosen randomly
- Random r-regular: each variable is chosen r times

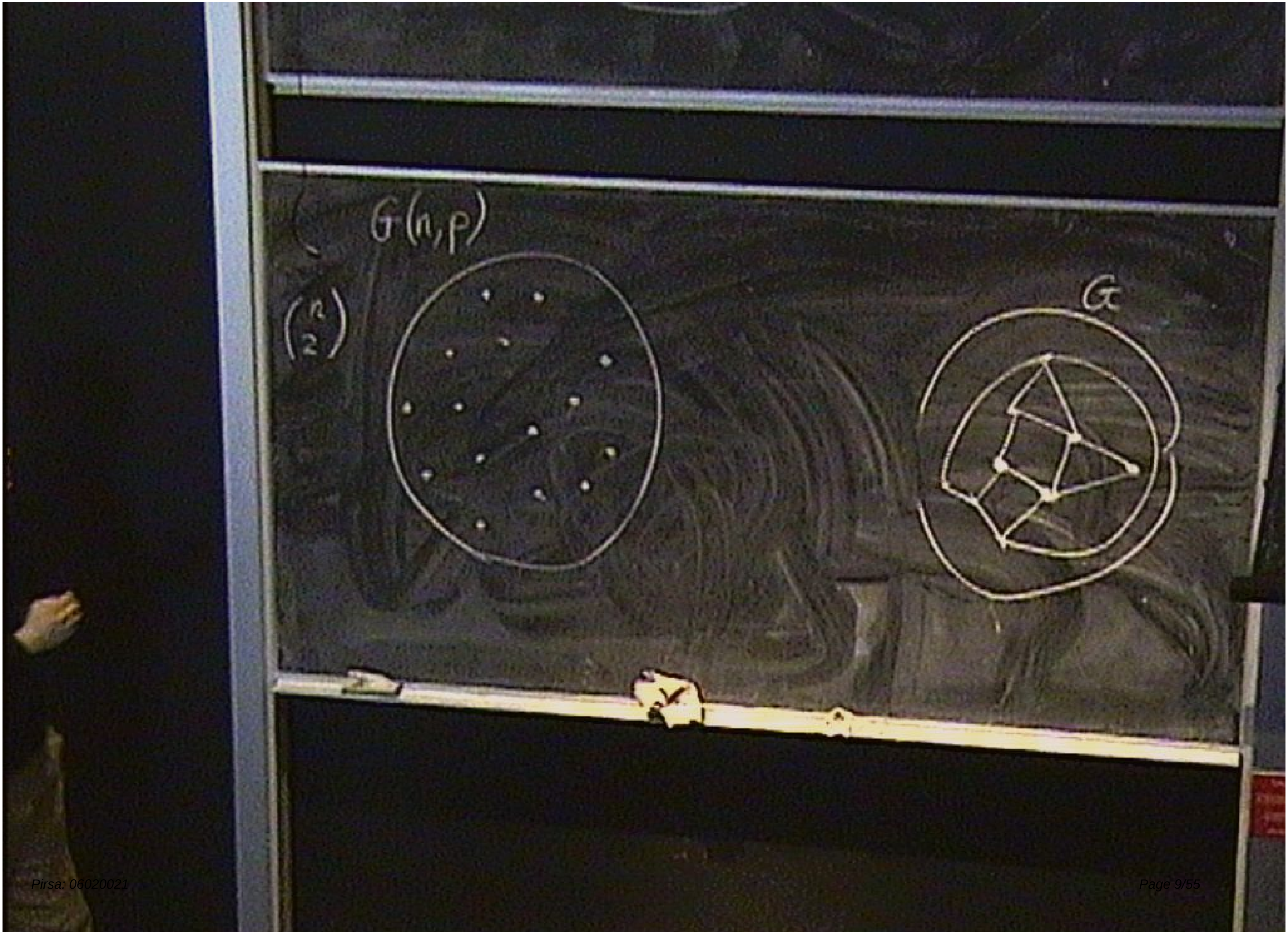


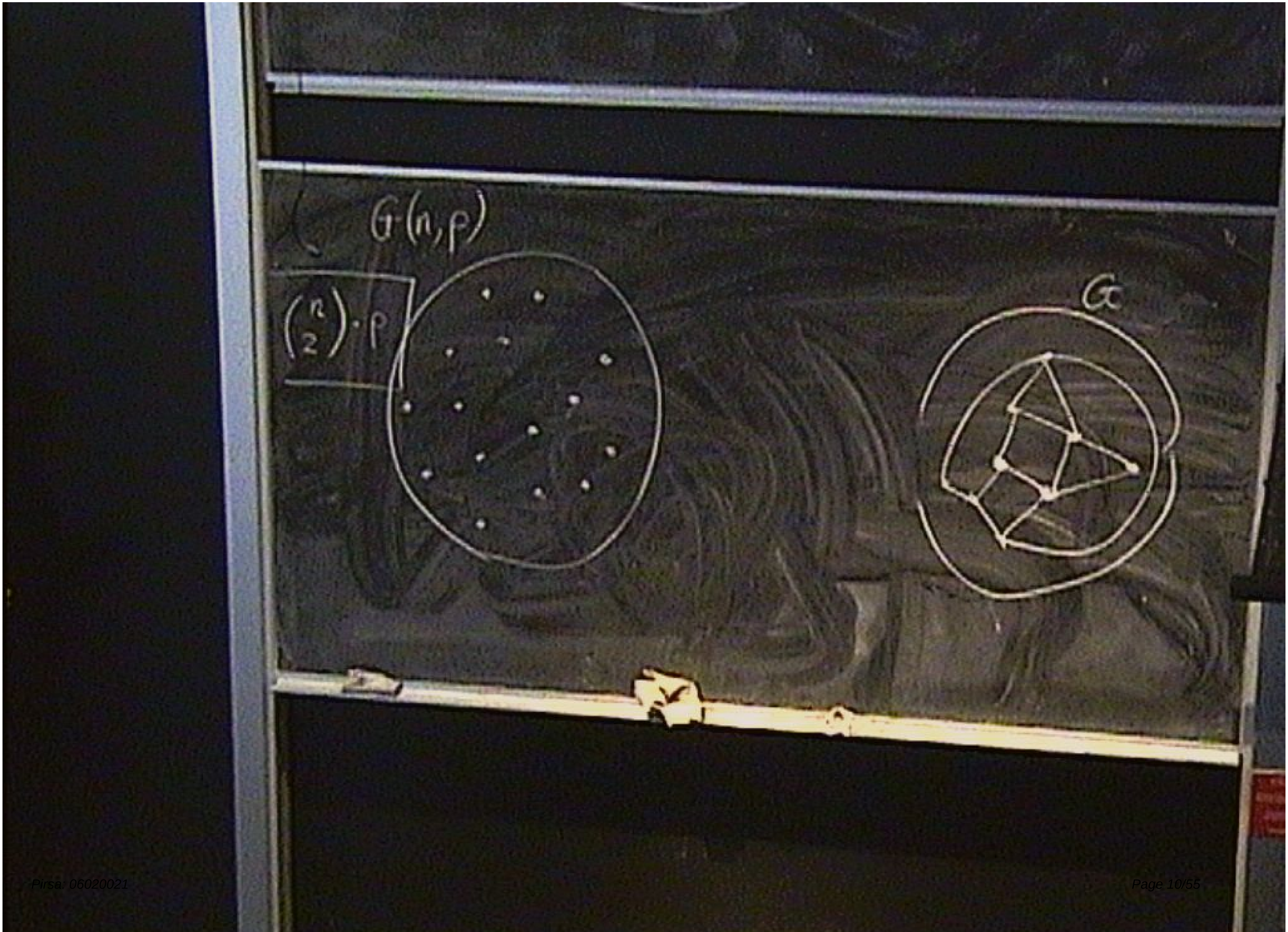


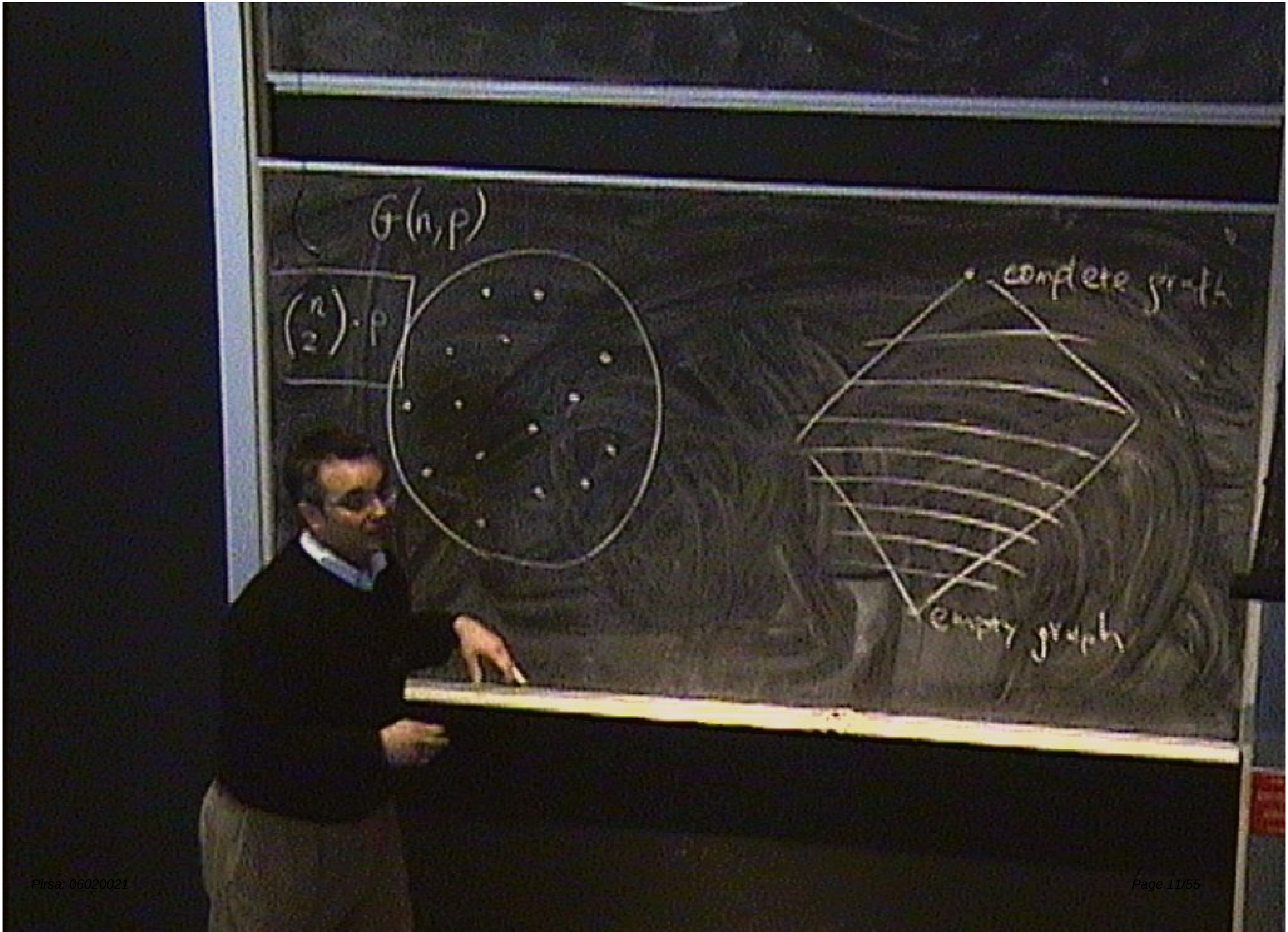


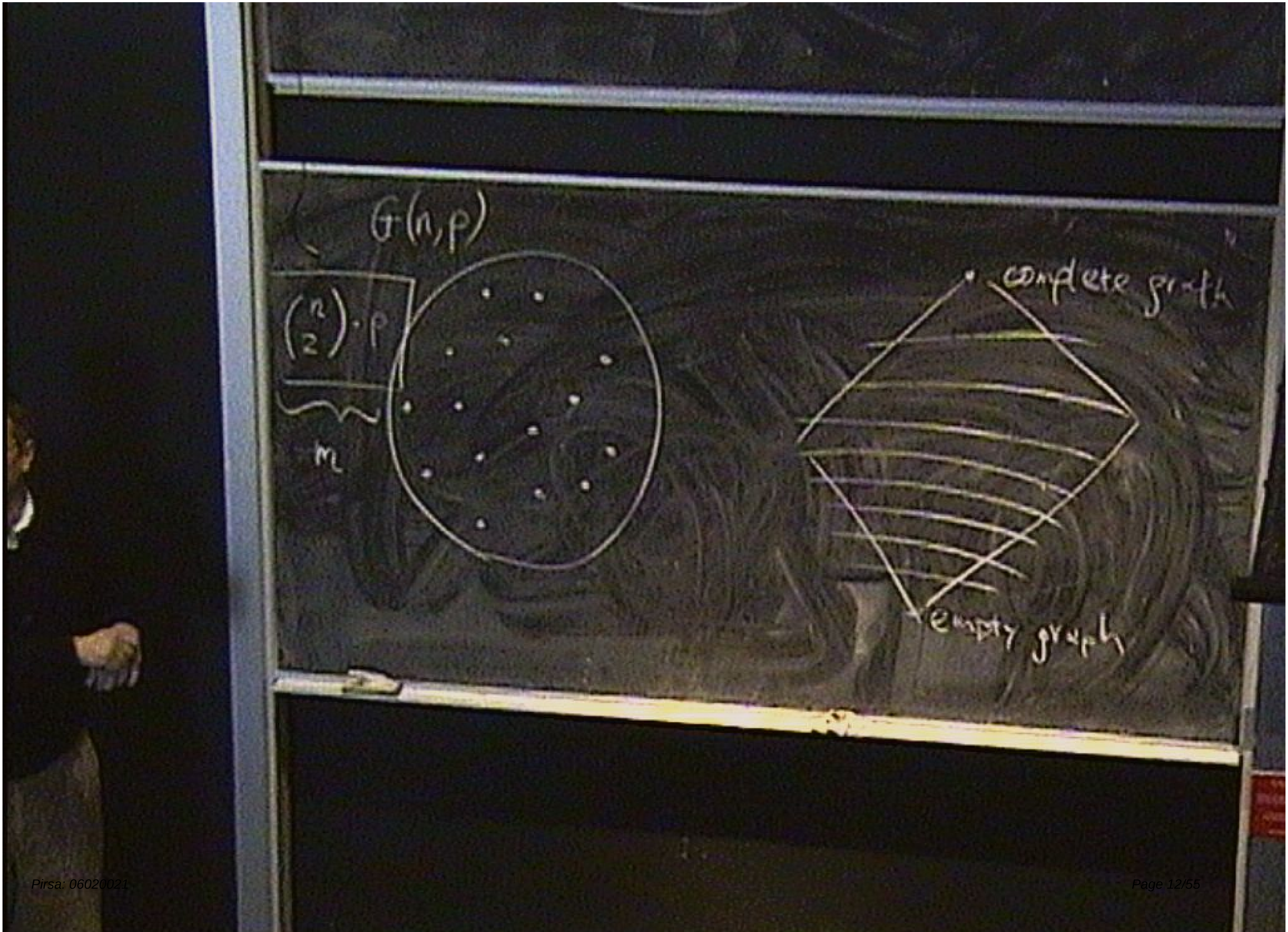

G

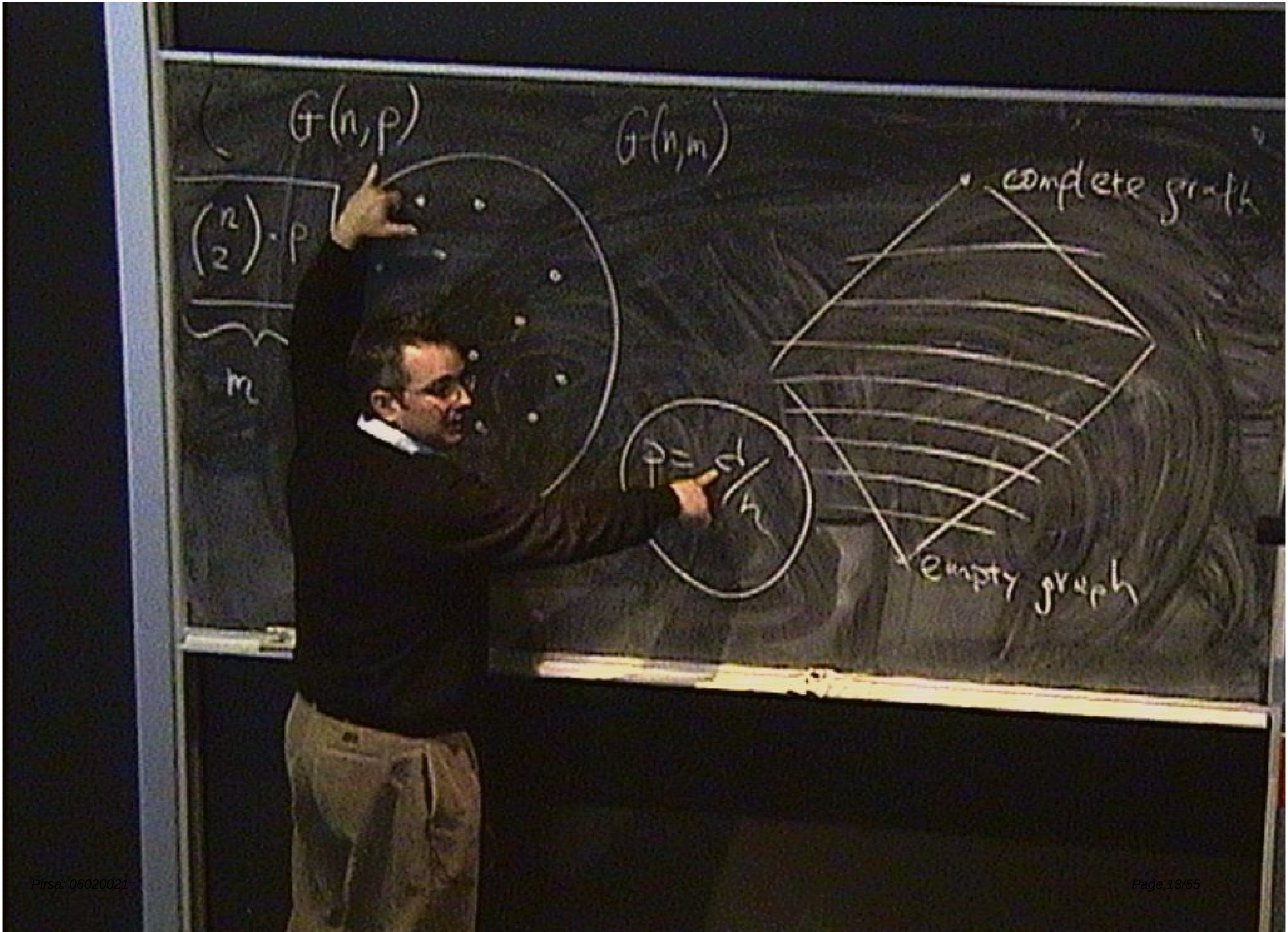

G

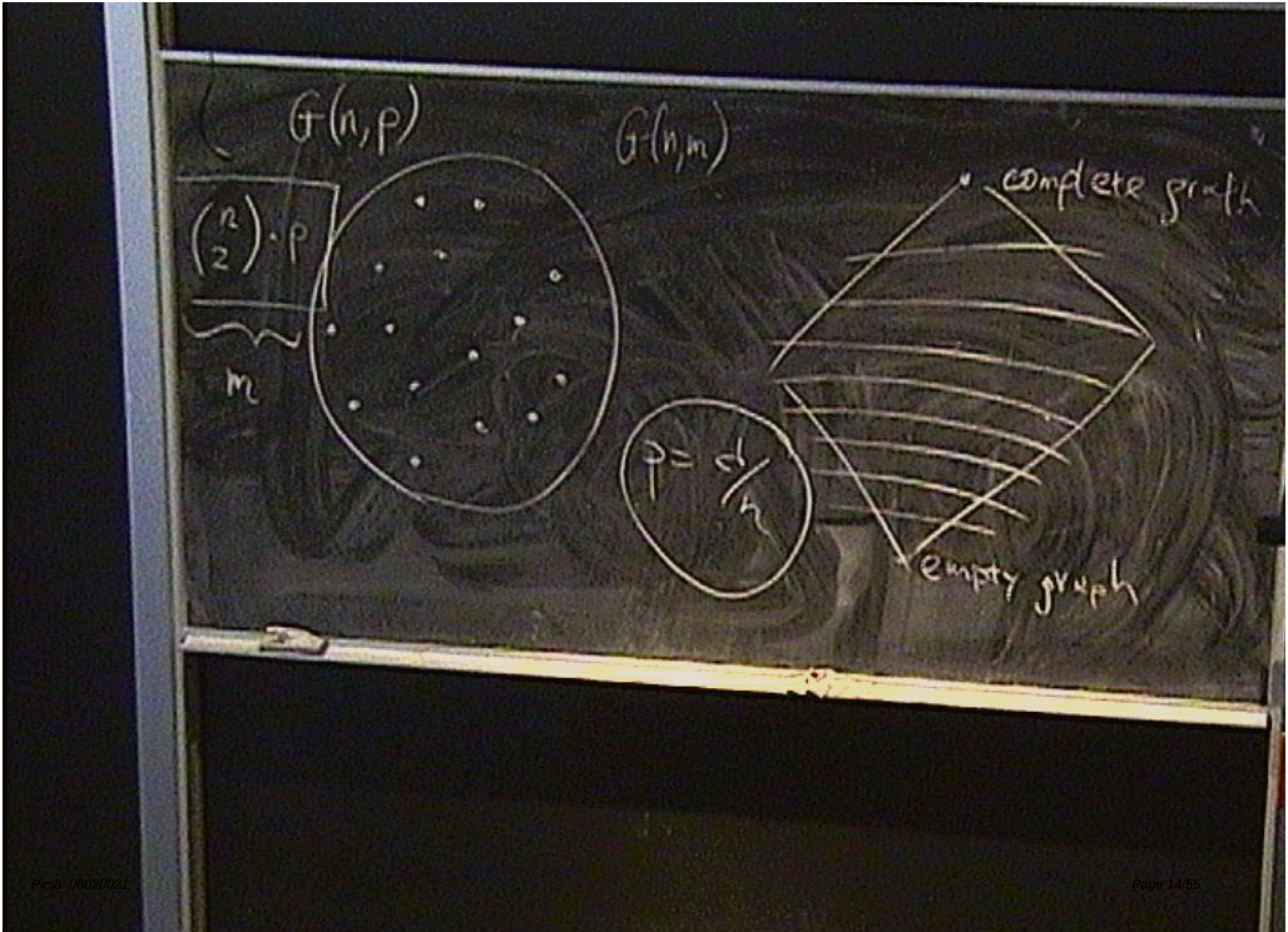

G(n,p)
$\binom{n}{2}$
G

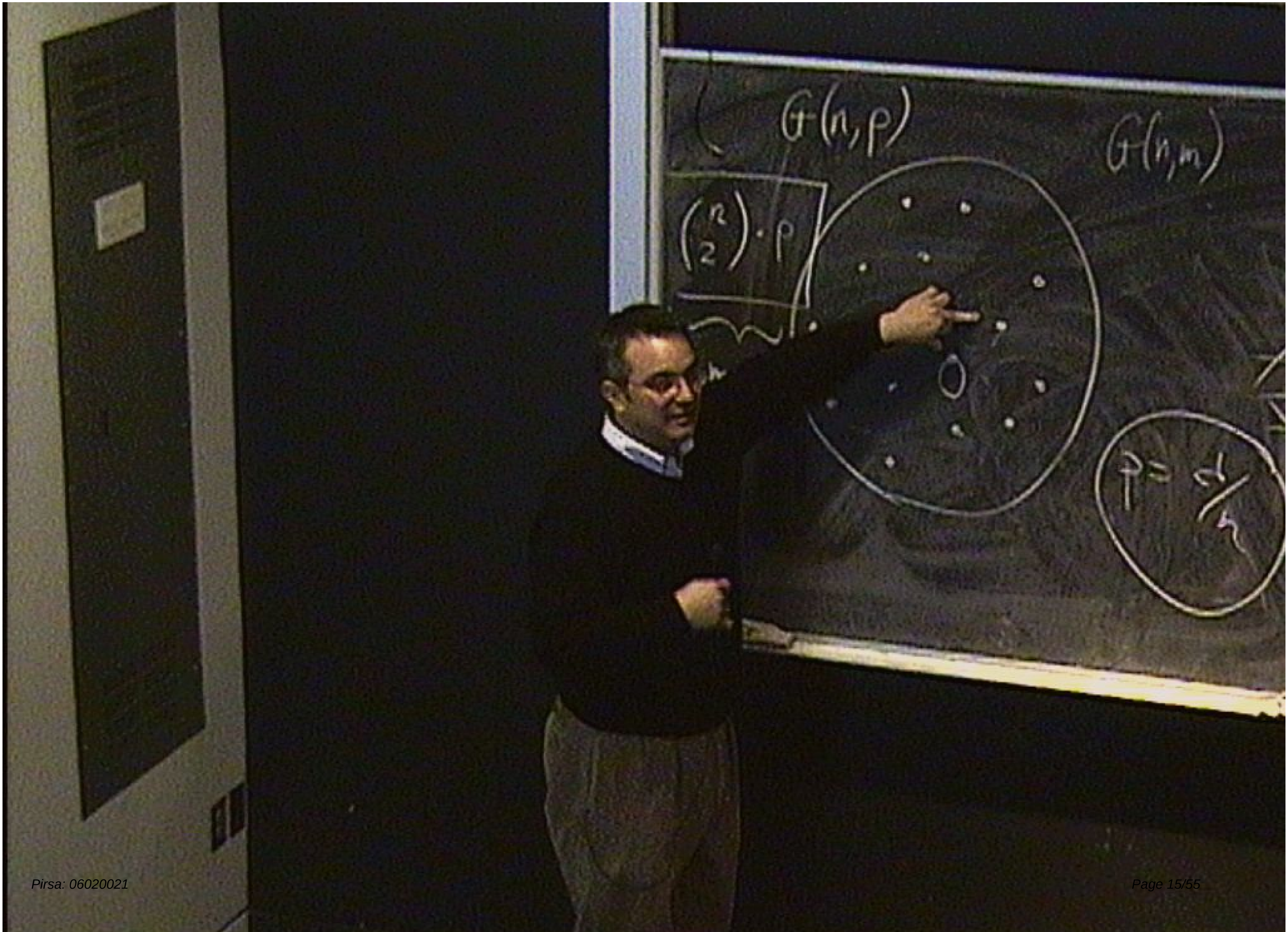

G(n,p)
(n 2)·p
G

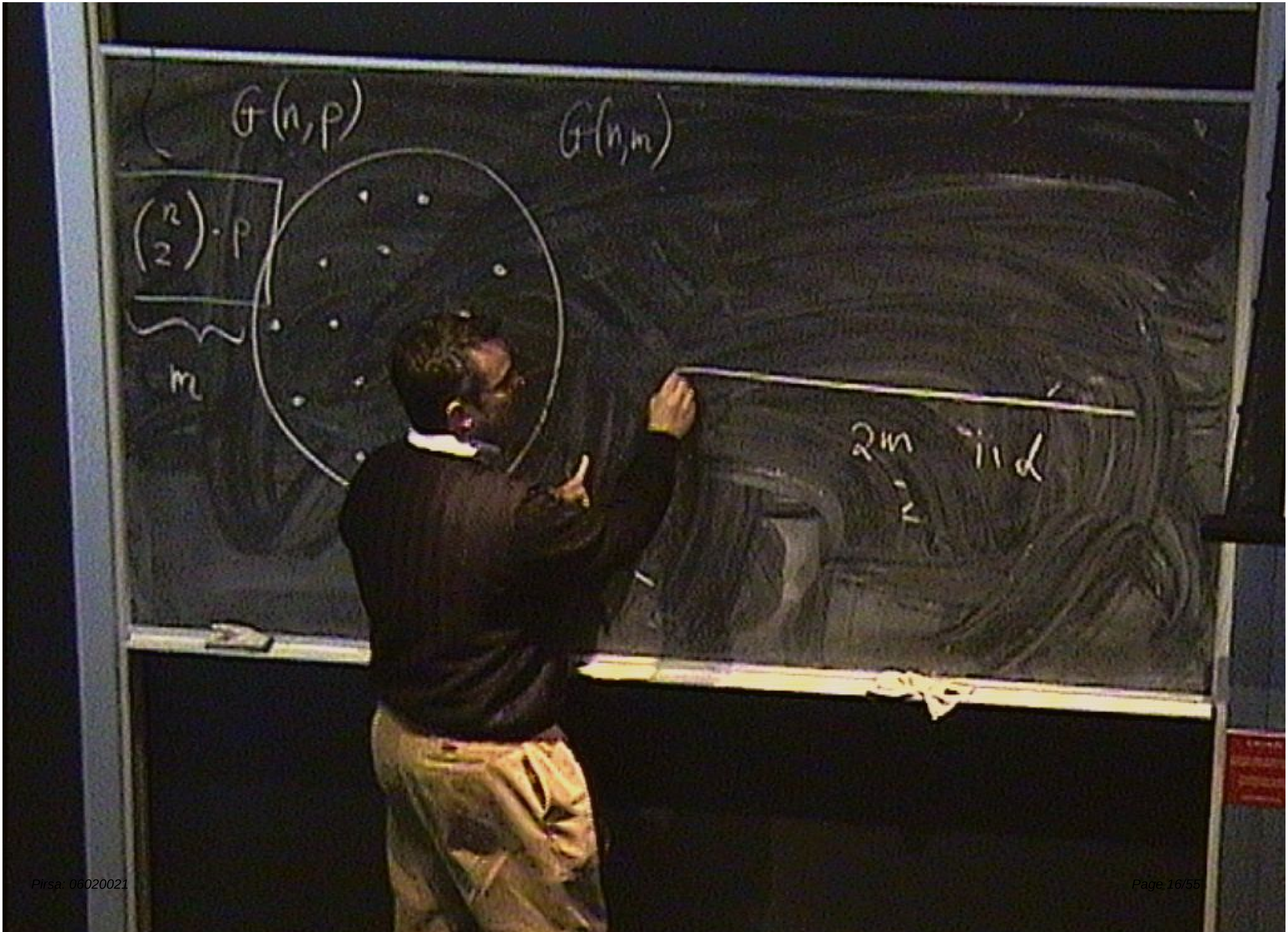

G(n,p)
$\binom{n}{2} \cdot p$
complete graph
empty graph

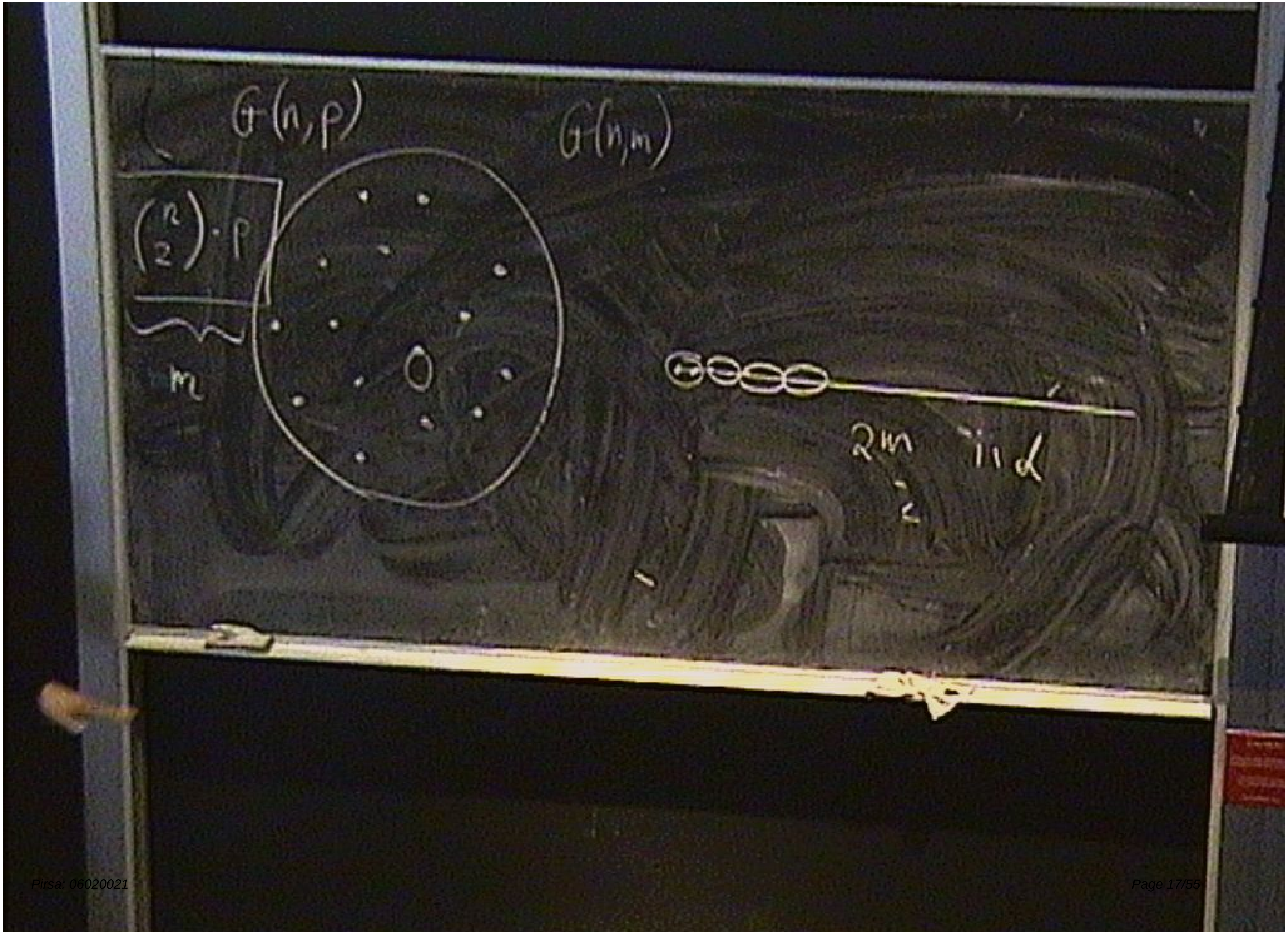

G(n, p)
$\binom{n}{2} \cdot p$
m
complete graph
empty graph


G(n,p)
G(n,m)
$\binom{n}{2} \cdot p$
m
complete graph
p = d/n
empty graph


G(n,p)
G(n,m)
$\binom{n}{2} \cdot p$
m
complete graph
p = d/n
empty graph


G(n,p)
G(n,m)

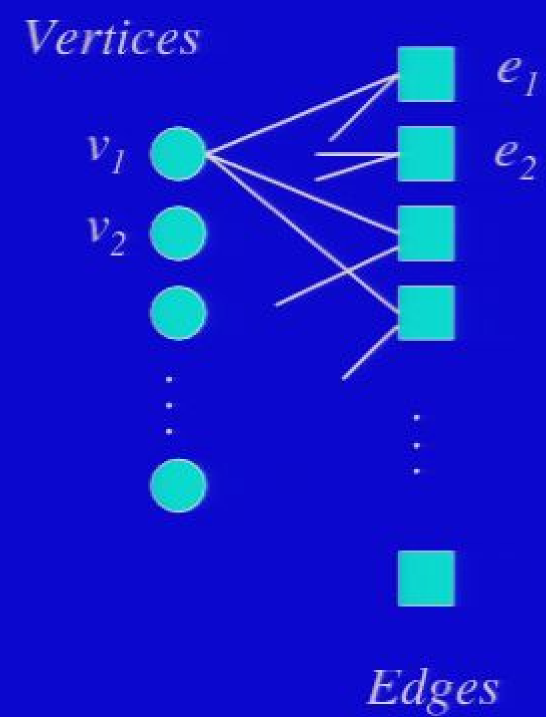

G(n,p)
G(n,m)
m

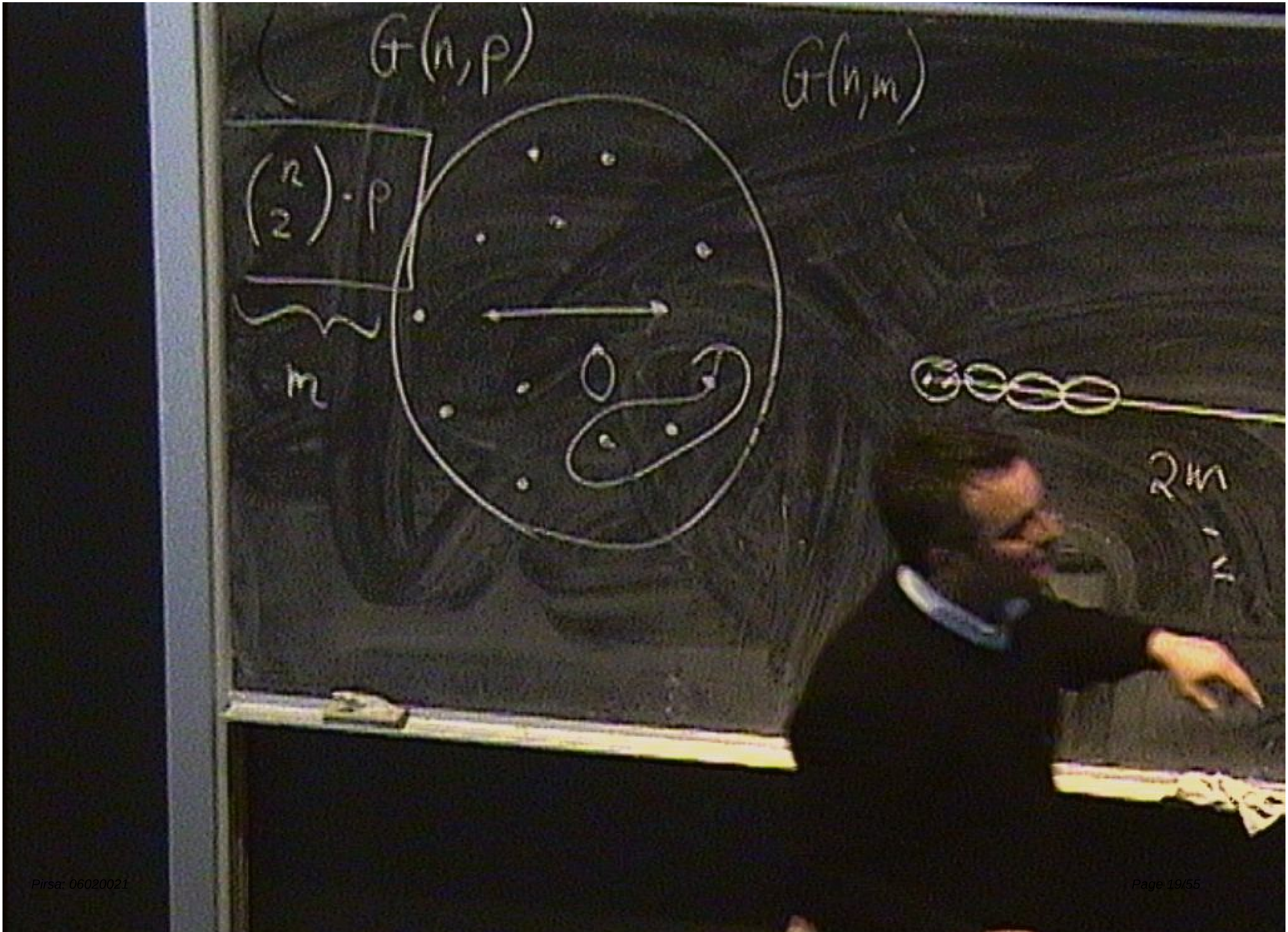

G(n,p)
G(n,m)
$\binom{n}{2}\cdot p$
m
2m

Random Graph k-coloring

- Each vertex is a variable with domain {1,2,…,k}
- Each edge is a "not-equal" constraint on two variables

- G(n,m) random graph: the two variables are chosen randomly
- Random r-regular: each variable is chosen r times

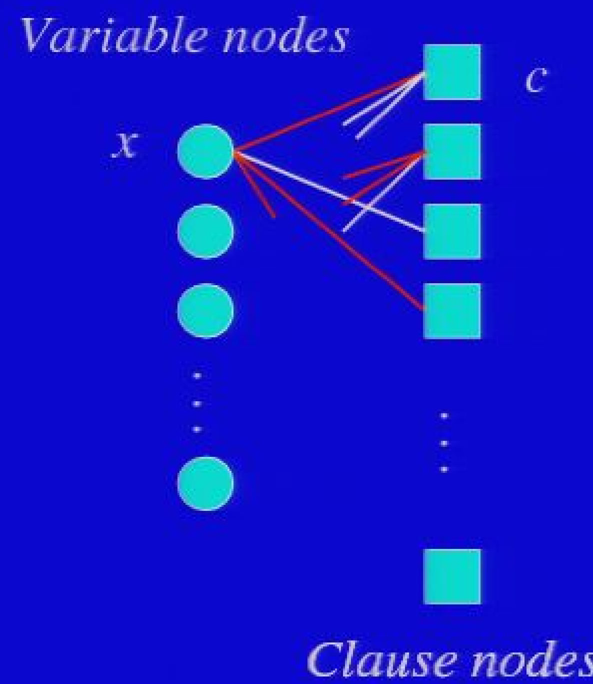



G(n,p)
G(n,m)
$\binom{n}{2} \cdot p$
m
2m

Random k-SAT

- Variables are binary.
- Every constraint (k-clause) binds k variables.
- Forbids exactly one of the 2^k possible joint values.

- Random k-SAT = each clause picks k random literals.

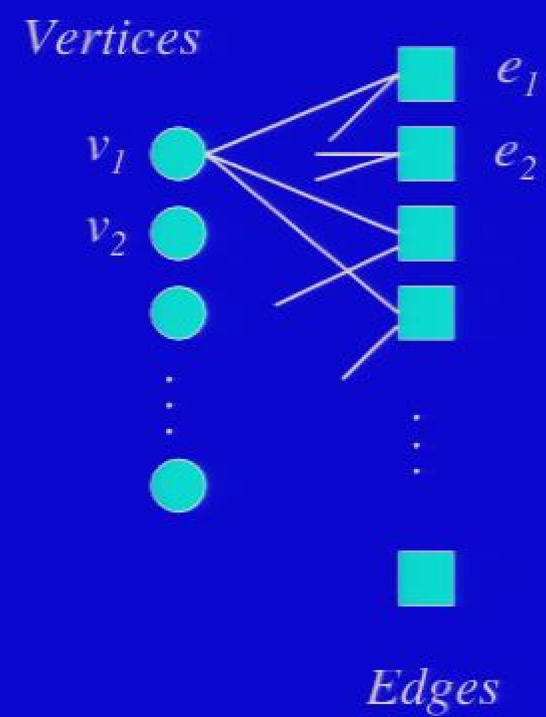


Similarly: NAE k-SAT, hypergraph 2-coloring, XOR-SAT...

Random Graph k-coloring

- Each **vertex** is a variable with domain {1,2,...,k}
- Each **edge** is a “not-equal” constraint on two variables

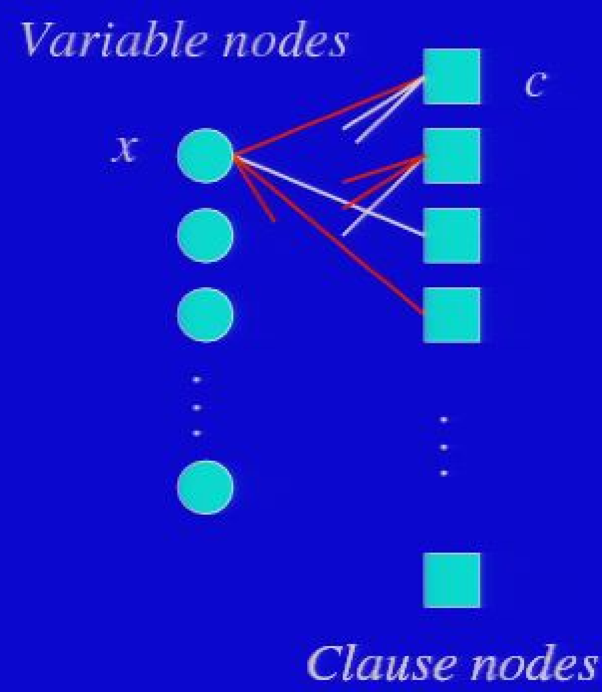
- **G(n,m)** random graph: the two variables are chosen randomly
- Random **r-regular**: each variable is chosen r times



Random k-SAT

- Variables are binary.
- Every constraint (k-clause) binds k variables.
- Forbids exactly one of the 2^k possible joint values.

- Random k-SAT = each clause picks k random literals.



Similarly: NAE k-SAT, hypergraph 2-coloring, XOR-SAT...

A simple k-coloring algorithm

- Repeat
  - Pick a random uncolored vertex
  - Assign it the lowest allowed number (color)

Works when $d \leq k \log k$

[Bollobás, Thomasson 84]
[McDiarmid 84]

- There are no k-colorings for $d \geq 2k \log k$

Twice as good is possible

As d grows, $G(n, d/n)$ is k-colorable for

$$d \sim 2k \log k$$

[Shamir, Spencer 87], [Bollobás 89], [Frieze 90], [Łuczak 91]

Twice as good is possible

As d grows, $G(n, d/n)$ is k-colorable for

$$d \sim 2k \log k$$

[Shamir, Spencer 87], [Bollobás 89], [Frieze 90], [Łuczak 91]

Average degree	10^{60}	10^{80}	10^{100}	10^{130}	10^{1000}
$\times k \log k$	1.01	1.12	1.19	1.31	1.75

Only two possible values

Theorem. For every $d > 0$, there exists an integer $k = k(d)$ such that w.h.p. the chromatic number of $G(n, p = d/n)$

is either k or $k + 1$

[Łuczak 91]

Only two possible values

Theorem. For every $d > 0$, there exists an integer $k = k(d)$ such that w.h.p. the chromatic number of $G(n, p = d/n)$

is either k or $k + 1$

[Łuczak 91]

The Values

Theorem. For every $d > 0$, ~~there exists an integer $k = k(d)$ such that~~ w.h.p. the chromatic number of $G(n, p = d/n)$

$$\text{is either } k \text{ or } k+1$$

where k is the smallest integer s.t. $d < 2k \log k$.

[A., Naor '04]

Examples

- If $d = 7$, w.h.p. the chromatic number is 4 or 5 .

Examples

- If $d = 7$, w.h.p. the chromatic number is 4 or 5 .

- If $d = 10^{60}$, w.h.p. the chromatic number is

$$377145549067226075809014239493833600551612641764765068157\,5$$

or

$$377145549067226075809014239493833600551612641764765068157\,6$$

Random regular graphs

Theorem. For every integer $d > 0$, w.h.p. the chromatic number of a random d-regular graph

is either k, $k+1$, or $k+2$

where k is the smallest integer s.t. $d < 2k \log k$.

[A., Moore '04]

Bounds for the k-SAT threshold

[A., Peres '04]

For all $k \geq 3$:

$$2^k \ln 2 - k < r_k < 2^k \ln 2$$

k	3	4	5	7	10	20	21
Upper bound	4.51	10.23	21.33	87.88	708.94	726, 817	1, 453, 635
Lower bound	3.52	7.91	18.79	84.82	704.94	726, 809	1, 453, 626
Best algorithm	3.52	5.54	9.63	33.23	172.65	95, 263	181, 453

Bounds for the k-SAT threshold

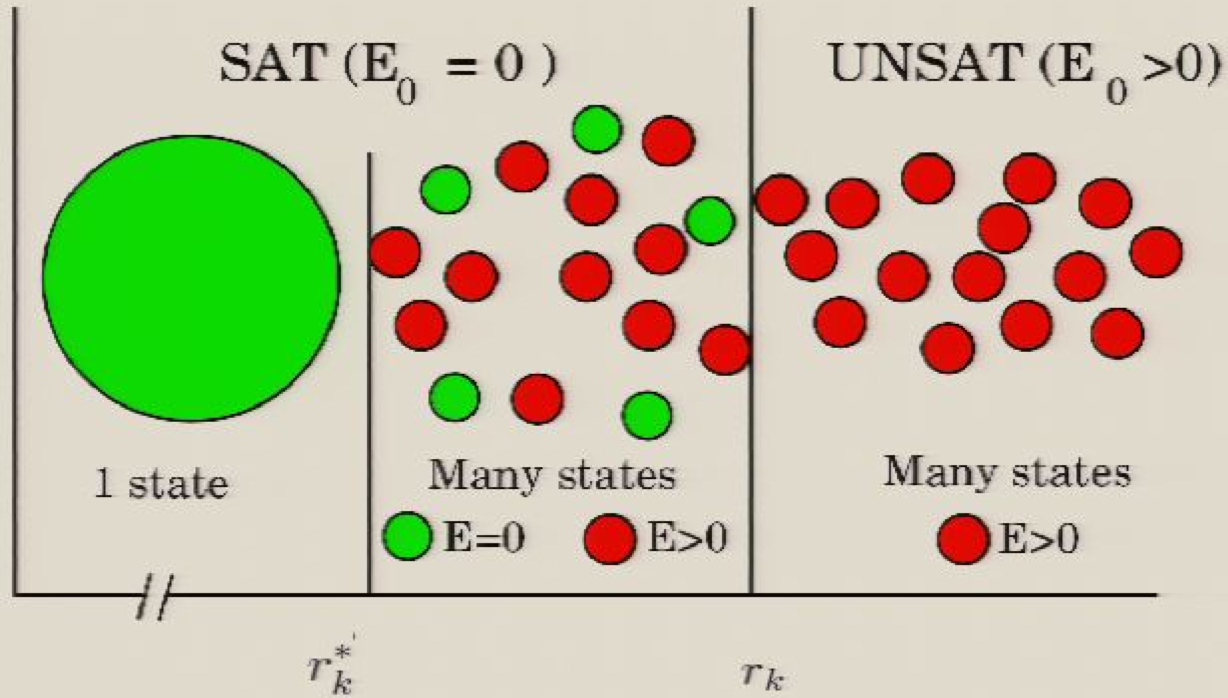
[A., Peres '04]

For all $k \geq 3$:

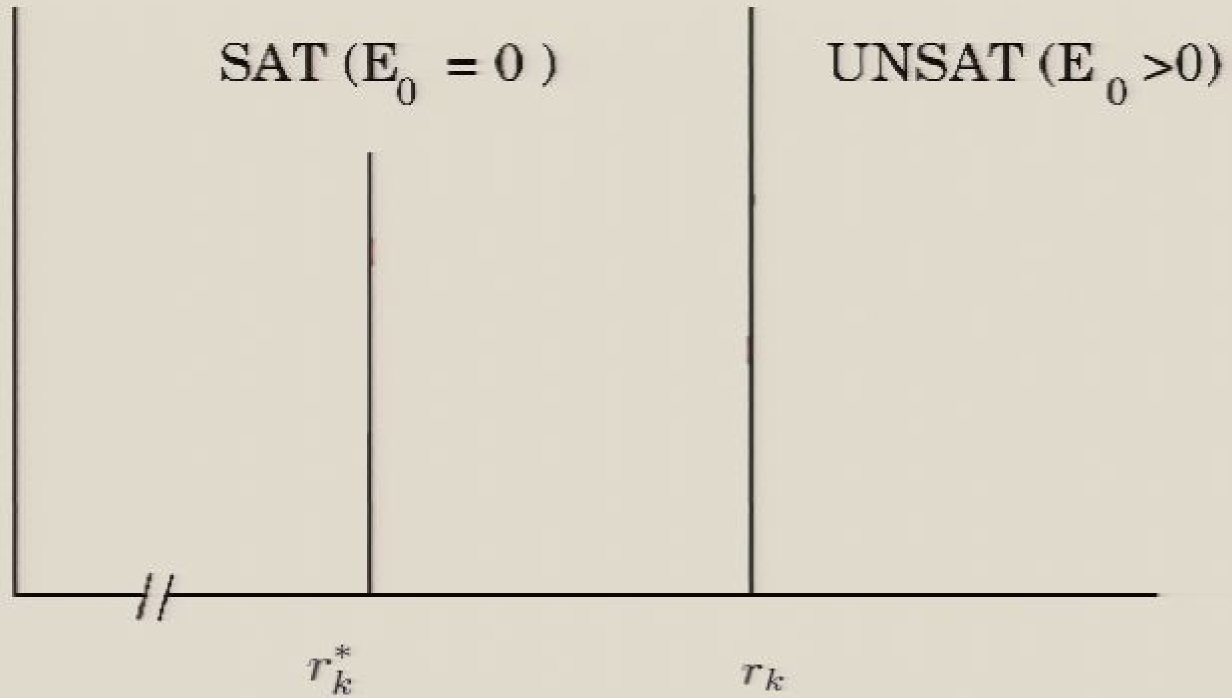
$$2^k \ln 2 - k < r_k < 2^k \ln 2$$

k	3	4	5	7	10	20	21
Upper bound	4.51	10.23	21.33	87.88	708.94	726,817	1,453,635
Lower bound	3.52	7.91	18.79	84.82	704.94	726,809	1,453,626
Best algorithm	3.52	5.54	9.63	33.23	172.65	95,263	181,453

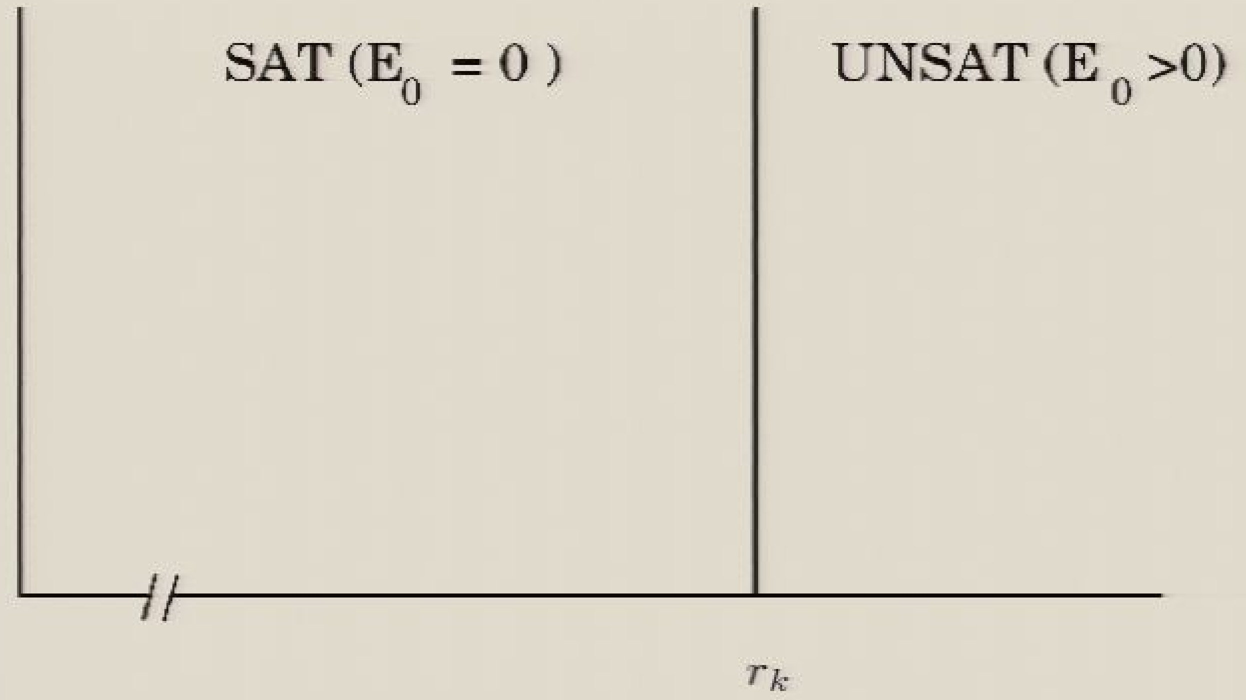
They say....


SAT (E_0 = 0)
UNSAT (E_0 >0)
1 state
Many states
E=0
E>0
Many states
E>0
r_k^*
r_k

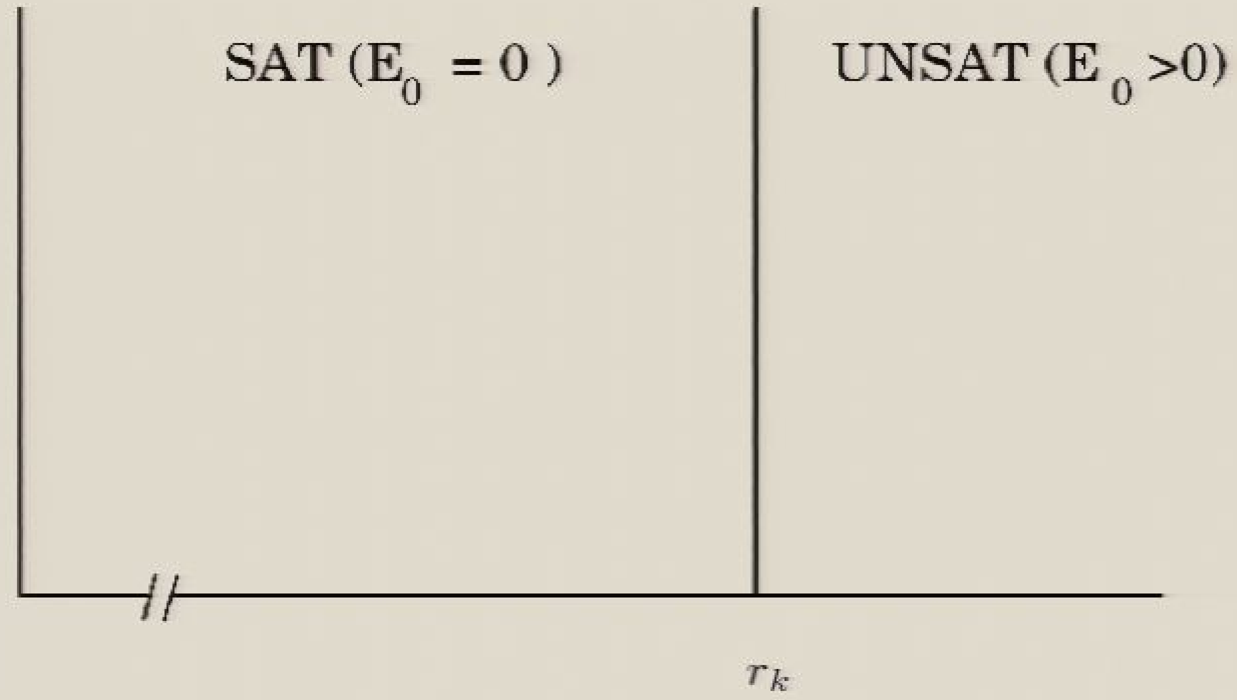
They say....



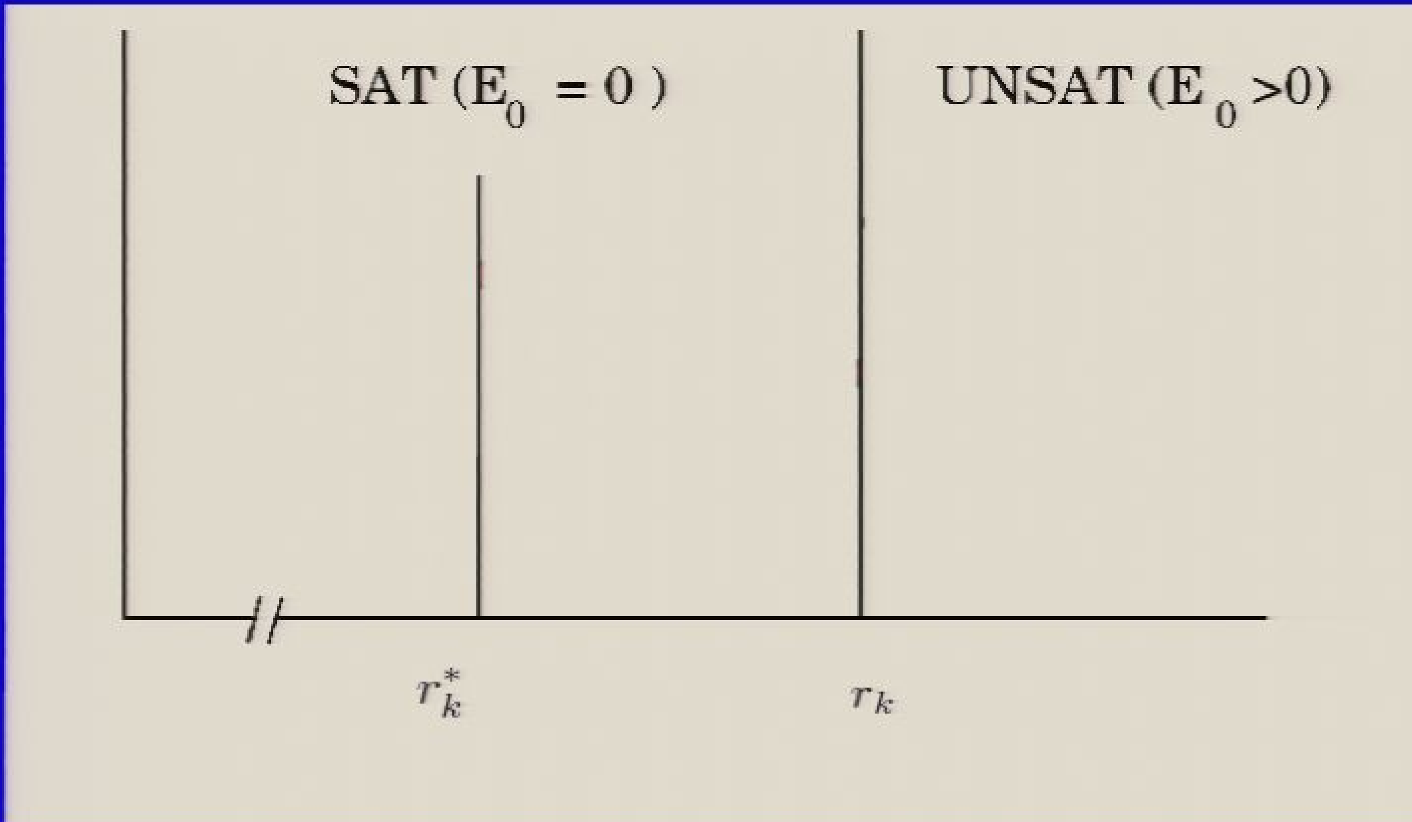
They say....



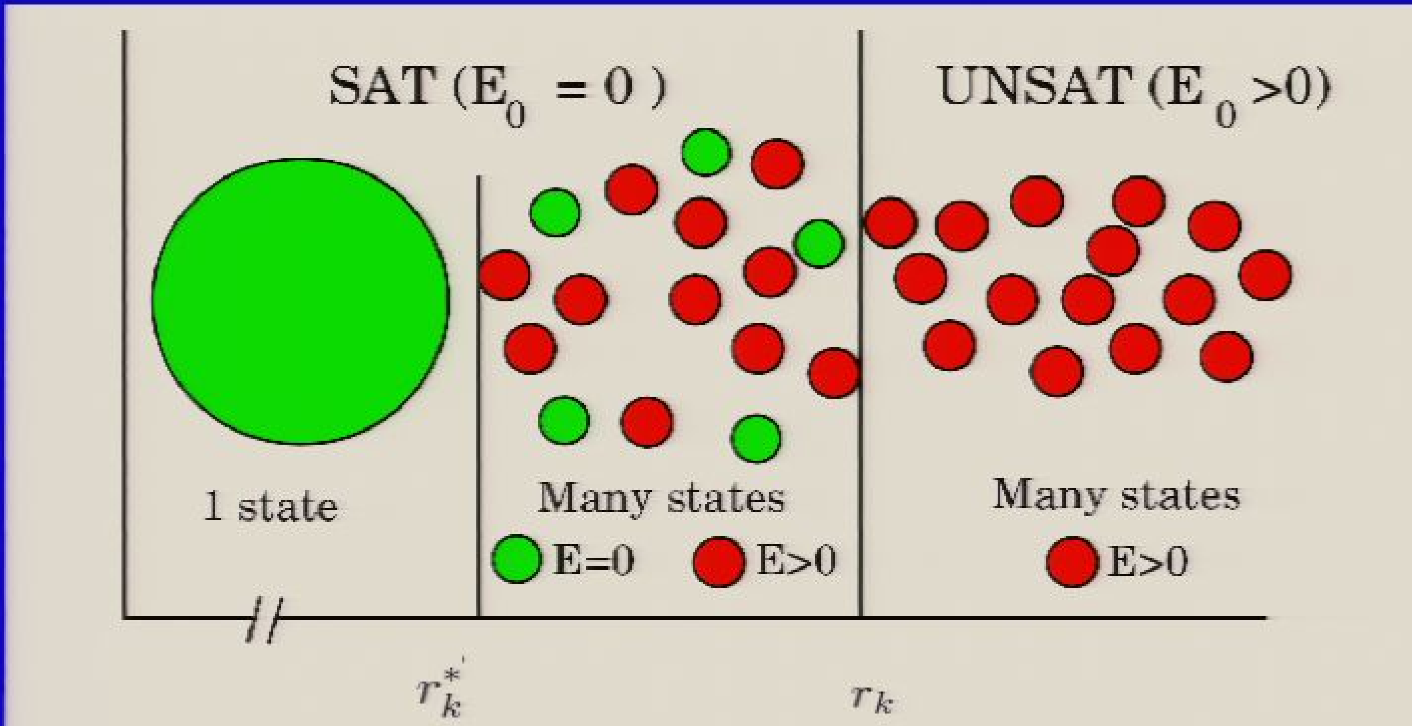
They say....

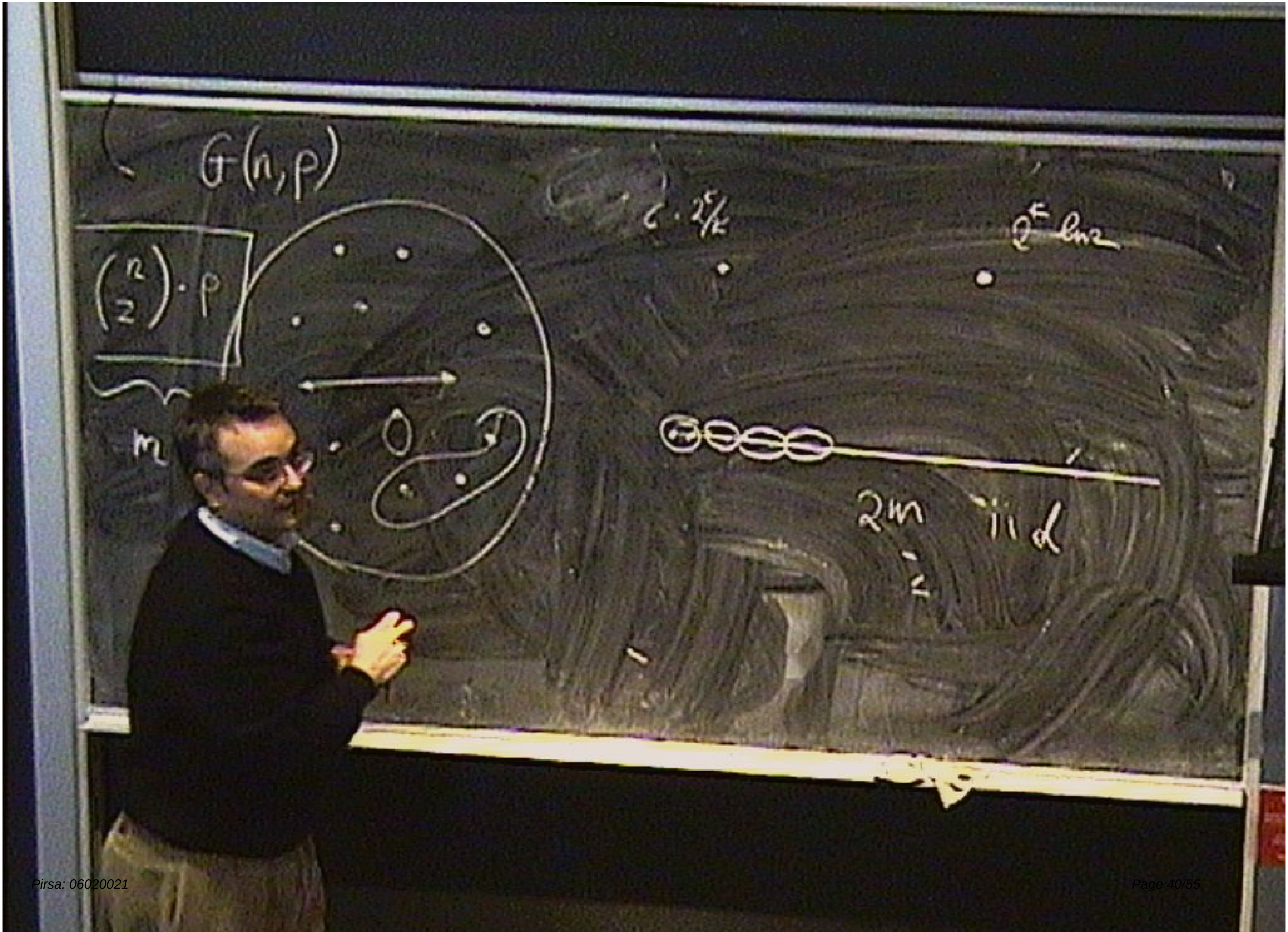


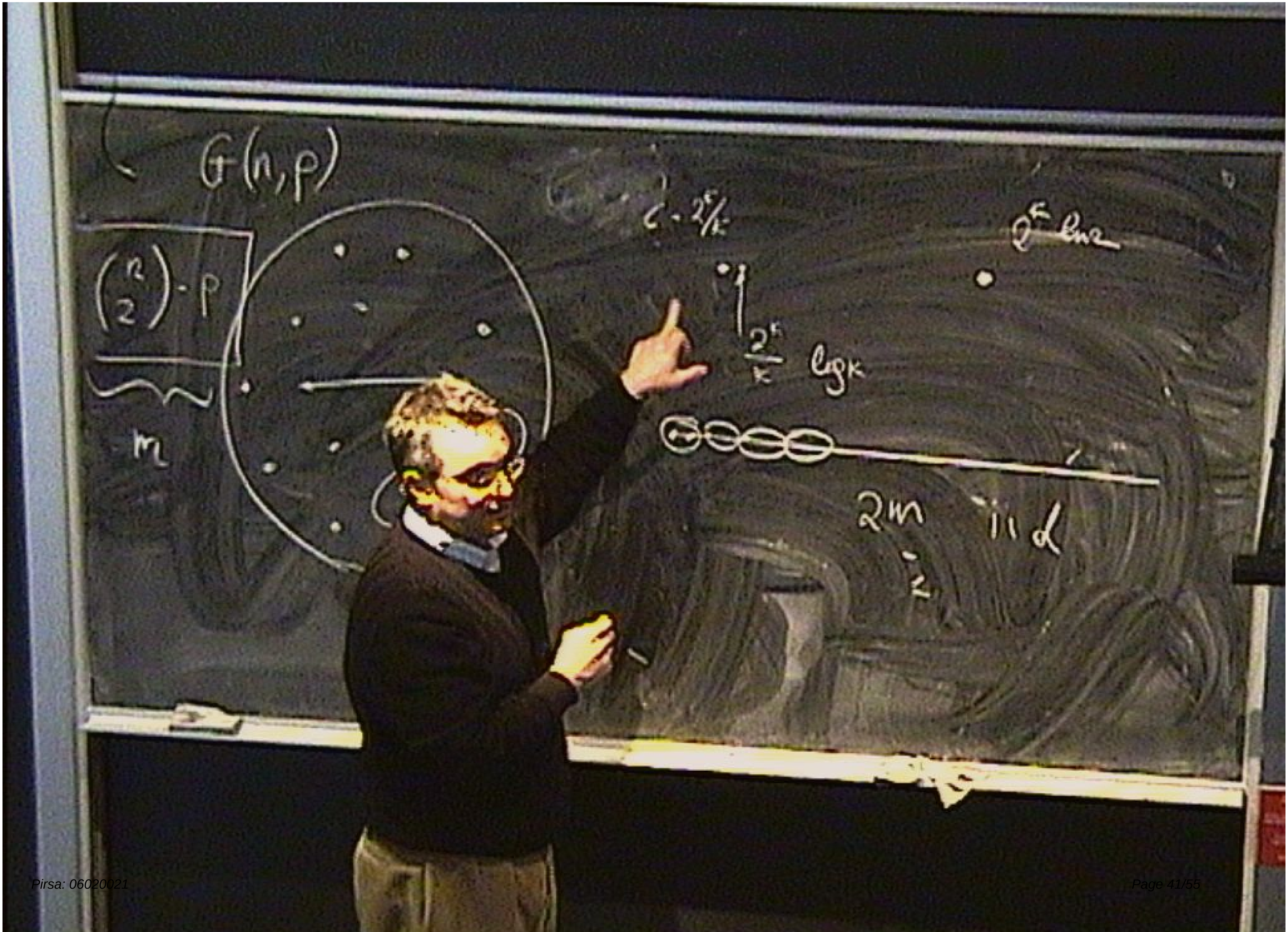
They say....

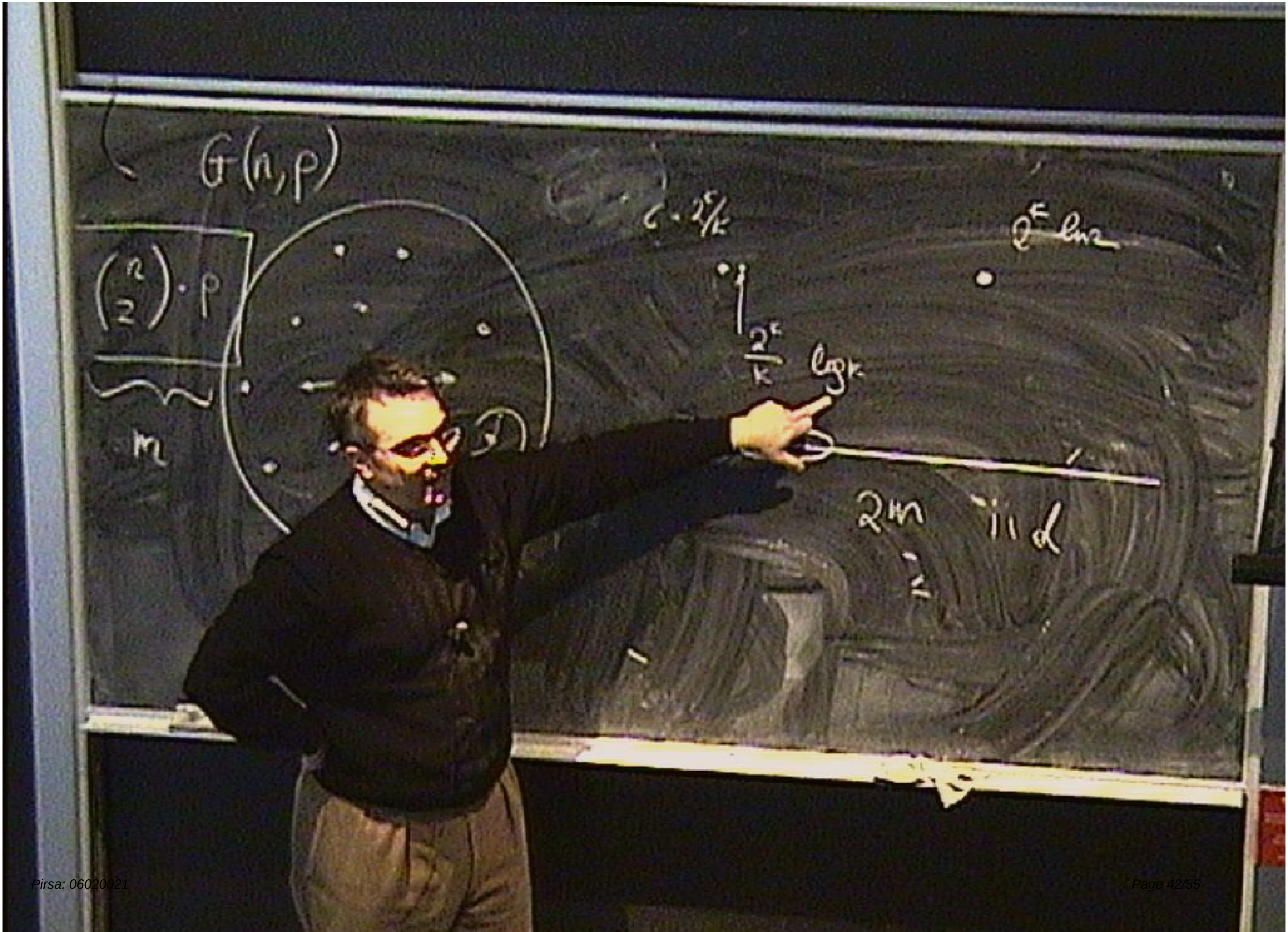


They say....

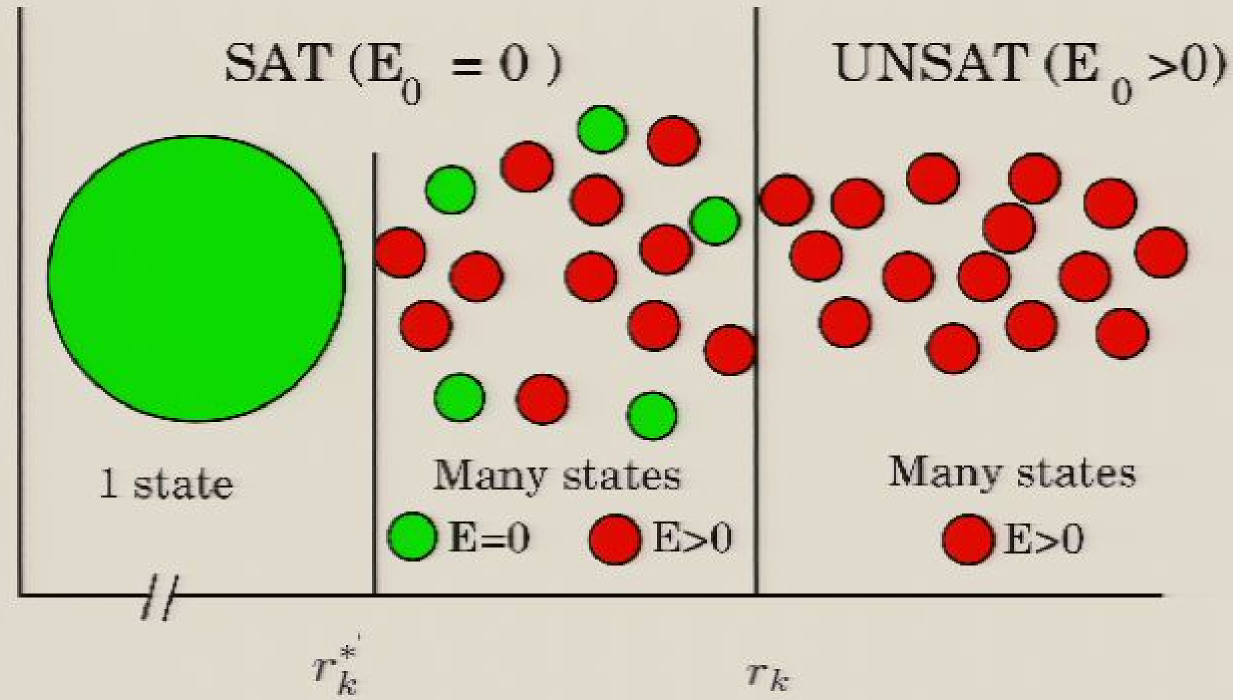

SAT (E_0 = 0)
UNSAT (E_0 >0)
1 state
Many states
E=0
E>0
Many states
E>0
r_k^{*}
r_k


G(n,p)
2m

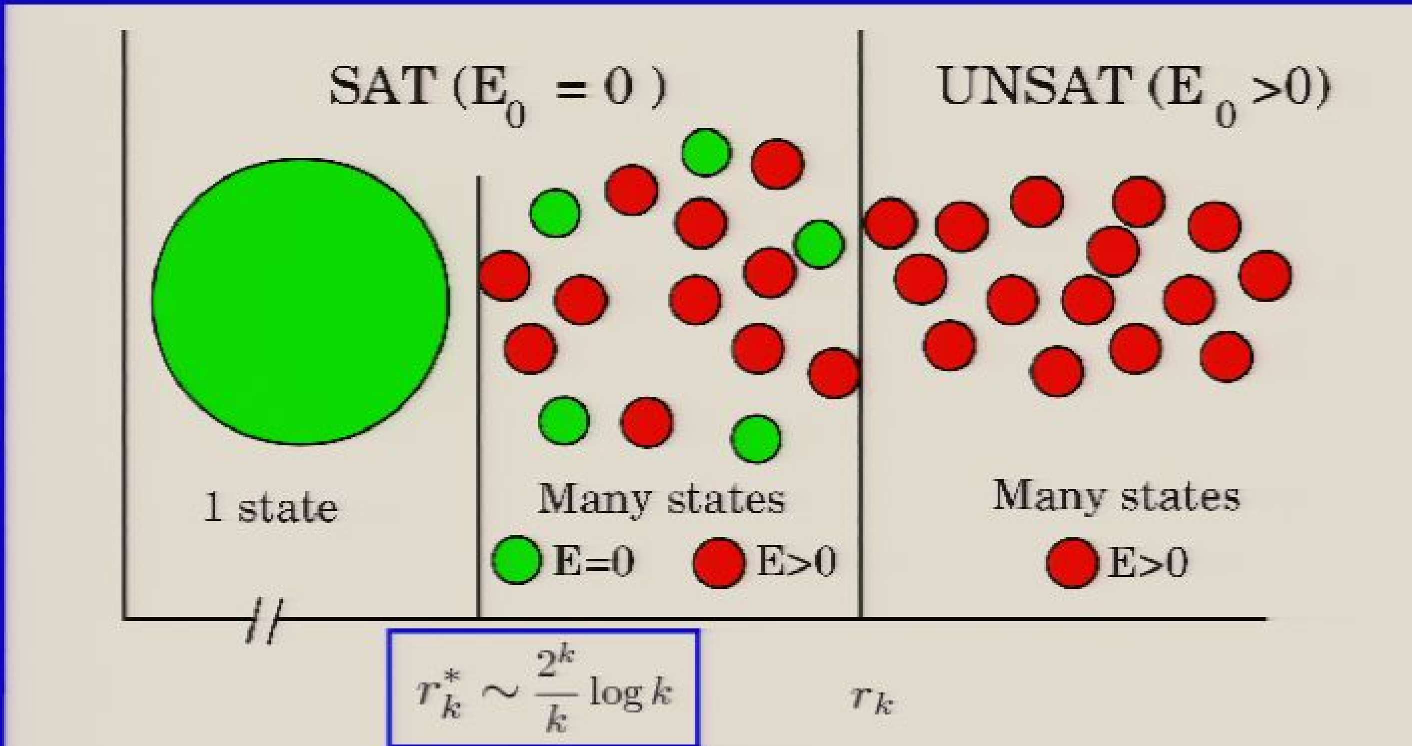

G(n,p)
(n 2) · p
m
2m
n d
ln 2


G(n,p)
m
2m

They say....


SAT (E_0 = 0)
UNSAT (E_0 >0)
1 state
Many states
E=0
E>0
Many states
E>0
r_k^*
r_k

They say....


SAT (E_0 = 0)
UNSAT (E_0 >0)
1 state
Many states
E=0
E>0
Many states
E>0
$r_k^* \sim \frac{2^k}{k} \log k$
r_k

They are right!

For $r > 2^{k-1} \ln 2$ it is easy to prove:

- Small diameter
- Far apart from one another
- Exponentially many

[Mora, Mezárd, Zecchina '05] [A., Ricci-Tersenghi '05]

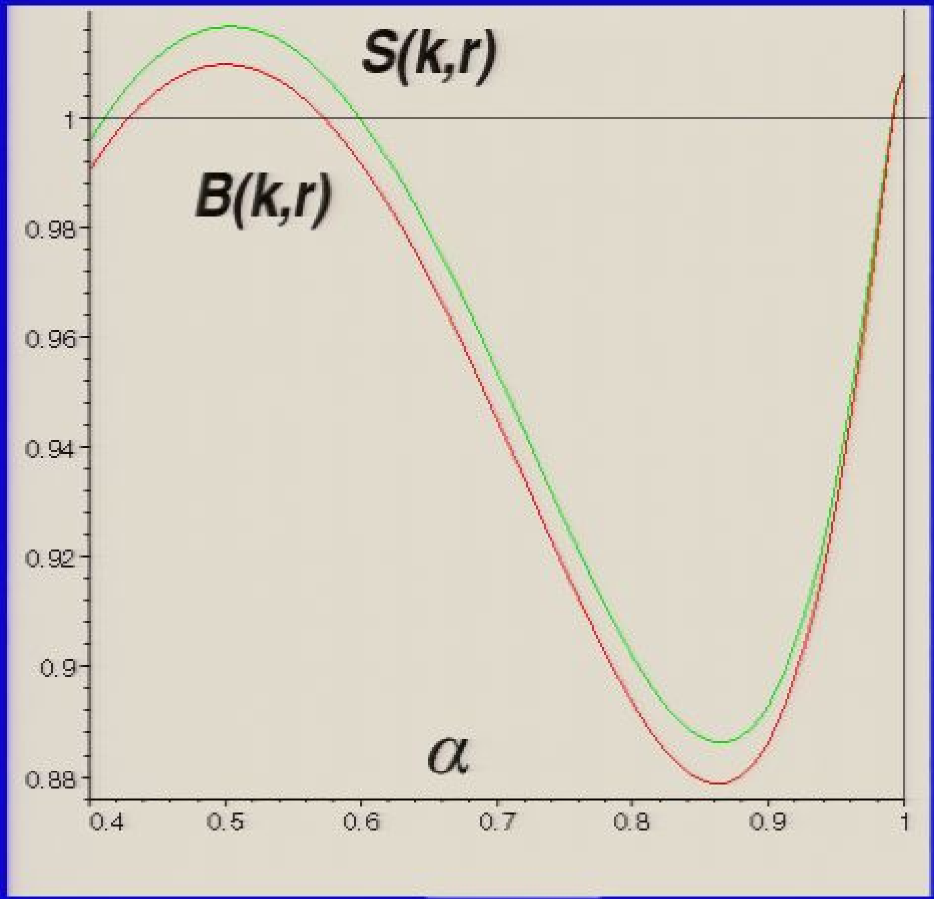
Random 10-SAT, r=700

The expected number of pairs of satisfying assignments having overlap αn is

$S(k,r)^n \times poly(n)$

The expected number of pairs of balanced sat. assignments having overlap αn is

$B(k,r)^n \times poly(n)$



Long-range dependencies

between the marginals over truth assignments

- Approximate the fraction p_i of satisfying truth assignments in which variable x_i takes value 1.
- Set x_i to 1 with probability p and simplify.

Long-range dependencies

between the marginals over truth assignments

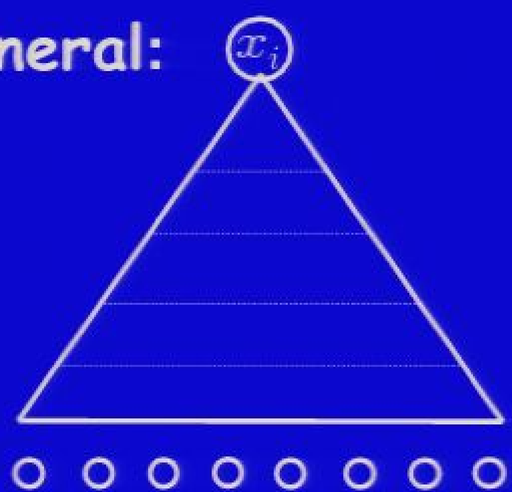
- Approximate the fraction p_i of satisfying truth assignments in which variable x_i takes value 1.
- Set x_i to 1 with probability p and simplify.

Long-range dependencies

between the marginals over truth assignments

- Approximate the fraction p_i of satisfying truth assignments in which variable x_i takes value 1.
- Set x_i to 1 with probability p and simplify.

In general:

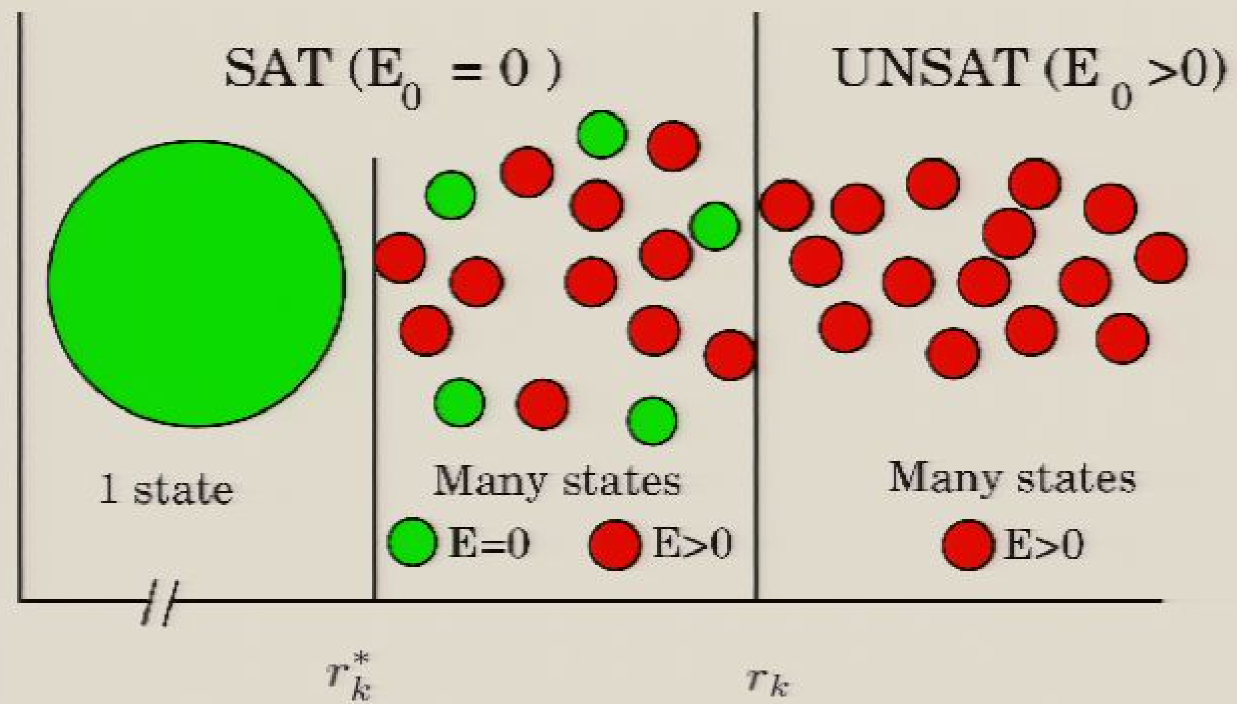


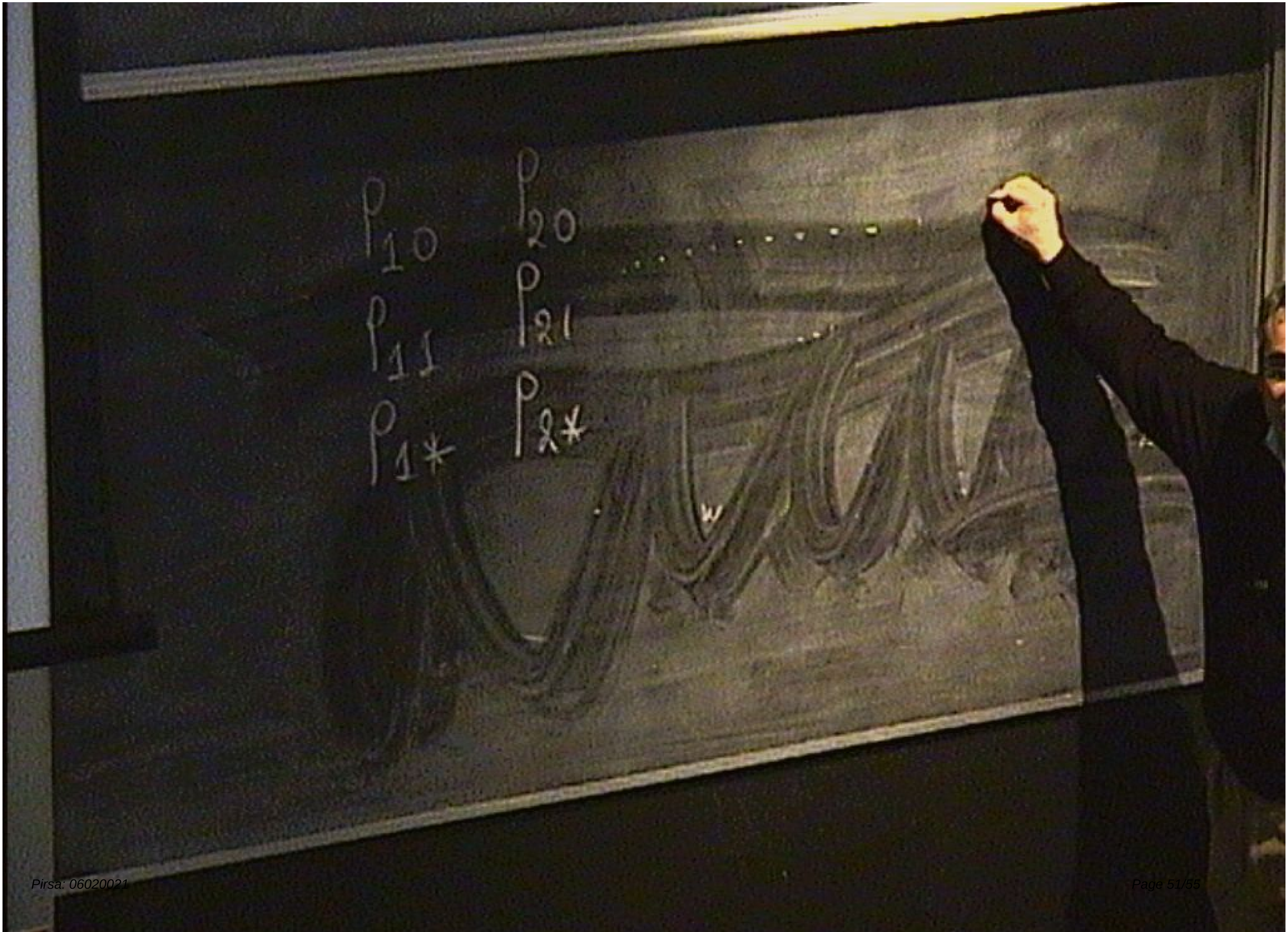
Given boundary Λ:

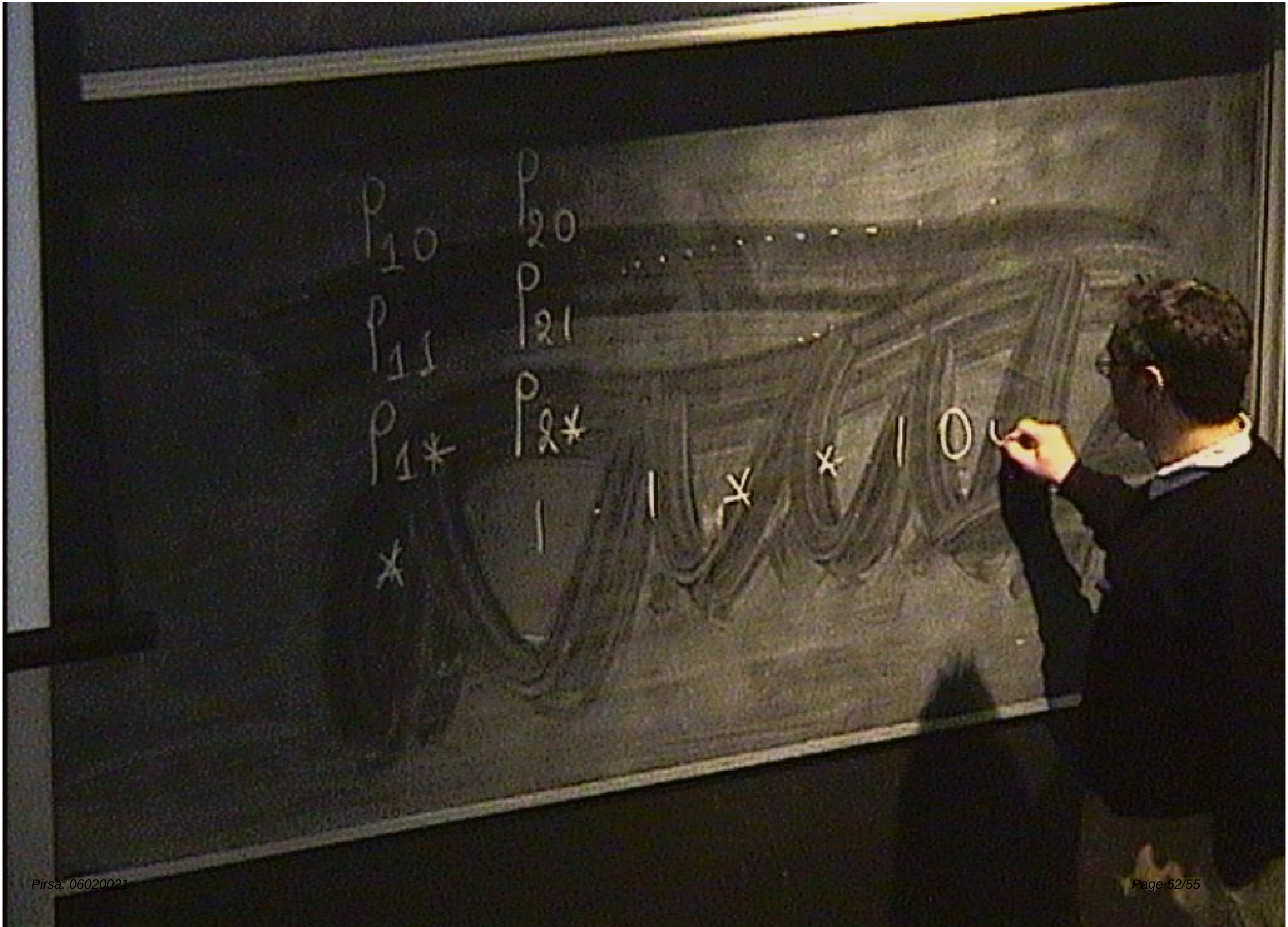
compute p_Λ

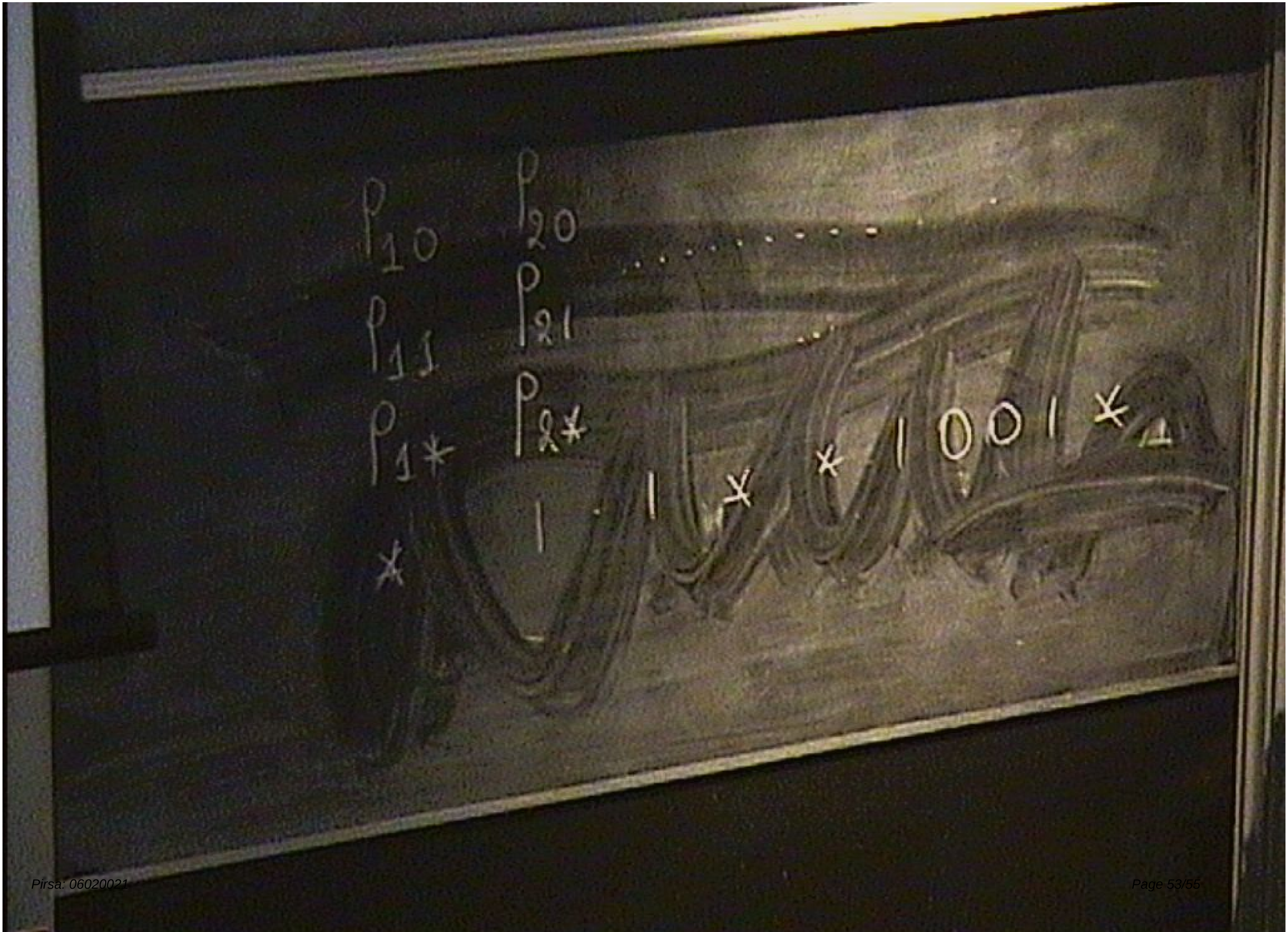
$$p_i = \sum_{\Lambda} p_\Lambda \times \mathrm{Ext}(\Lambda)$$

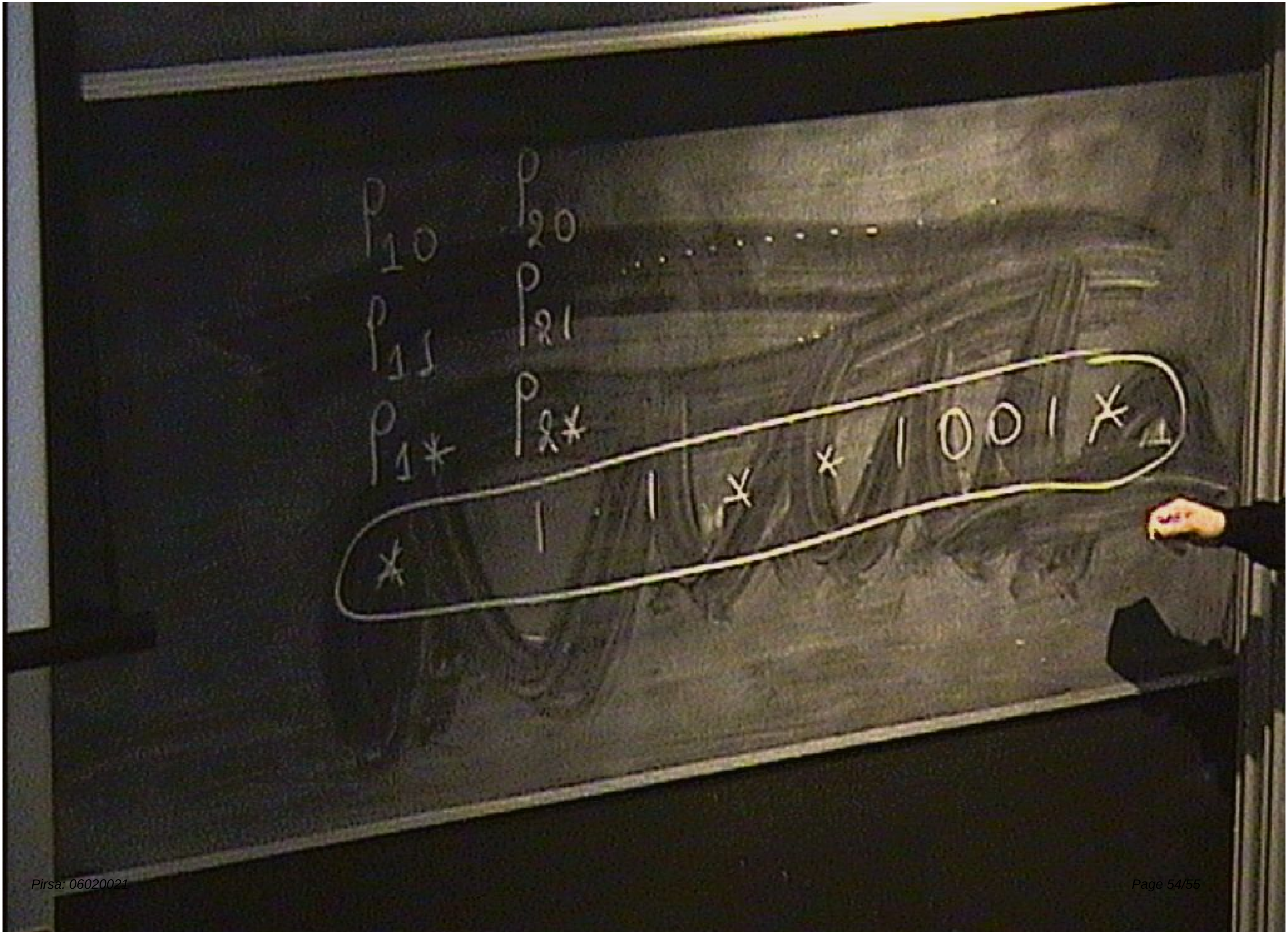
How do we overcome this?




ρ_{10} ρ_{20}
ρ_{11} ρ_{21}
ρ_{1*} ρ_{2*}


ρ_{10} ρ_{20}
ρ_{11} ρ_{21}
ρ_{1*} ρ_{2*}
* 1 1 * * 1 0


ρ10 ρ20
ρ11 ρ21
ρ1* ρ2*
* 1
1 * * 1 0 0 1 *


ρ10 ρ20
ρ11 ρ21
ρ1* ρ2*
* 1 1 * * 1 0 0 1 *

How do we overcome this?