Title: Phase transitions in NP complete problems

Date: Feb 10, 2006 11:00 AM

URL: http://pirsa.org/06020020

Abstract:

Satisfiability

Given a Boolean formula (CNF), decide if a satisfying truth assignment exists.

$$(\overline{x}_{12} \vee x_5) \wedge (x_{34} \vee \overline{x}_{21} \vee x_5 \vee \overline{x}_{27}) \wedge \cdots \wedge (x_{12}) \wedge (x_{21} \vee x_9 \vee \overline{x}_{13})$$

Cook's Theorem: Satisfiability is NP-complete.

---

k-SAT: Each clause has exactly k literals.

$k = 2$: Pick any variable and set it arbitrarily. (1 choice)

Satisfy any implications (repeatedly).

Either get a subformula or a contradiction.

$k \geq 3$: NP-complete

Satisfiability

Given a Boolean formula (CNF), decide if a satisfying truth assignment exists.

$$(\overline{x}_{12} \vee x_5) \wedge (x_{34} \vee \overline{x}_{21} \vee x_5 \vee \overline{x}_{27}) \wedge \cdots \wedge (x_{12}) \wedge (x_{21} \vee x_9 \vee \overline{x}_{13})$$

Cook's Theorem: Satisfiability is NP-complete.

---

k-SAT: Each clause has exactly k literals.

$k = 2$: Pick any variable and set it arbitrarily. (1 choice)

Satisfy any implications (repeatedly).

Either get a subformula or a contradiction.

$k \geq 3$: NP-complete

Satisfiability

Given a Boolean formula (CNF), decide if a satisfying truth assignment exists.

$$(\overline{x}_{12} \vee x_5) \wedge (x_{34} \vee \overline{x}_{21} \vee x_5 \vee \overline{x}_{27}) \wedge \cdots \wedge (x_{12}) \wedge (x_{21} \vee x_9 \vee \overline{x}_{13})$$

Cook's Theorem: Satisfiability is NP-complete.

---

k-SAT: Each clause has exactly k literals.

$k = 2$: Pick any variable and set it arbitrarily. (1 choice)

Satisfy any implications (repeatedly).

Either get a subformula or a contradiction.

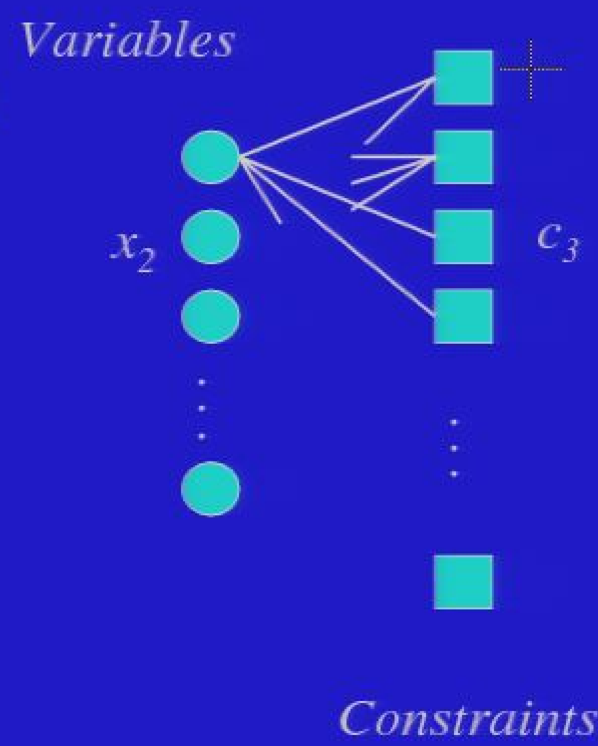
$k \geq 3$: NP-complete

tructure.ps - GSview
Edit Options View Orientation Media Help

The Setting: Random CSPs

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

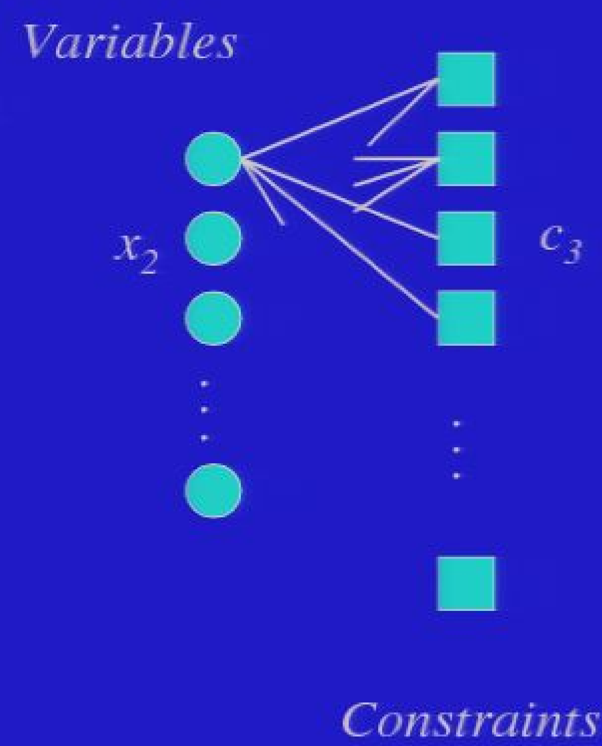
- Random bipartite graph:
- Sparse graph, i.e. m=Θ(n)



The Setting: Random CSPs

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

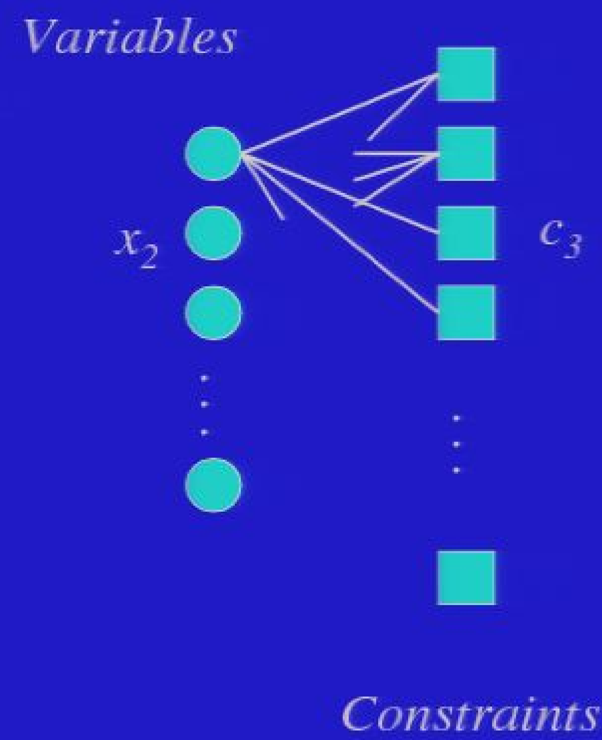
- Random bipartite graph:
- Sparse graph, i.e. m=Θ(n)



The Setting: Random CSPs

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

- Random bipartite graph:
- Sparse graph, i.e. $m=\Theta(n)$



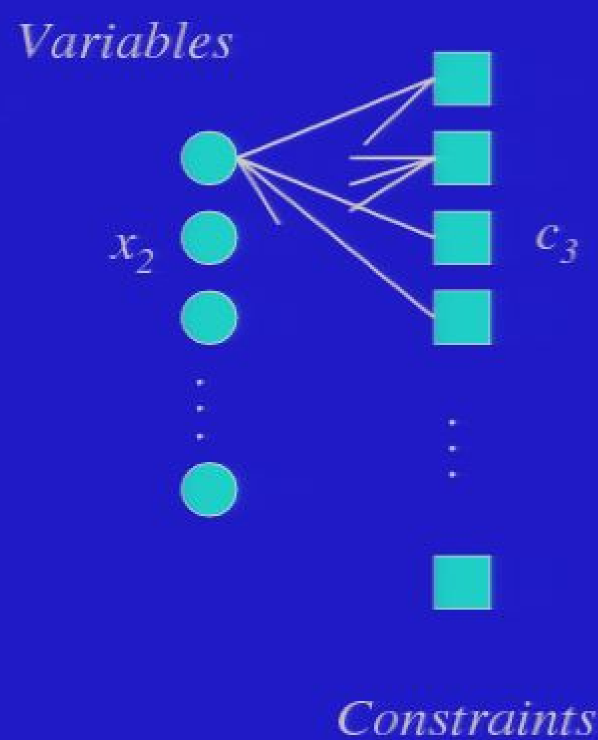
tructure.ps - GSview

Edit Options View Orientation Media Help

The Setting: Random CSPs

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

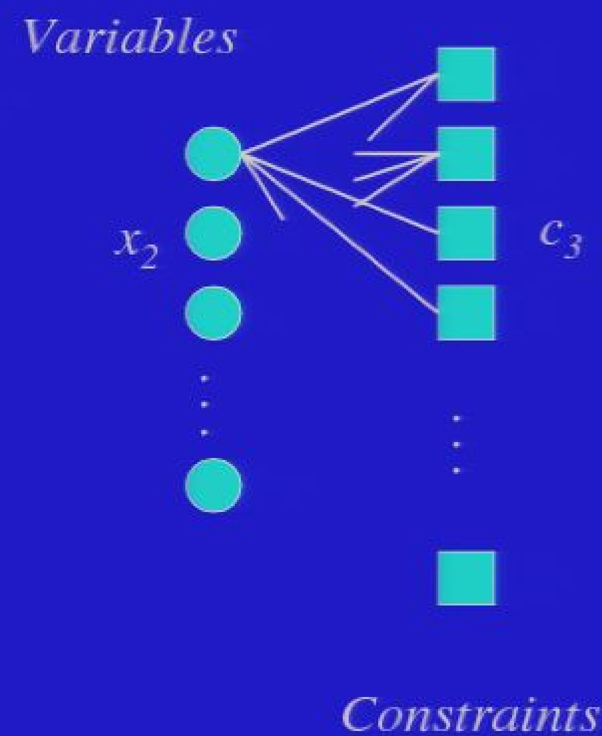
- Random bipartite graph:
- Sparse graph, i.e. m=Θ(n)



The Setting: Random CSPs

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

- Random bipartite graph:
- Sparse graph, i.e. $m=\Theta(n)$



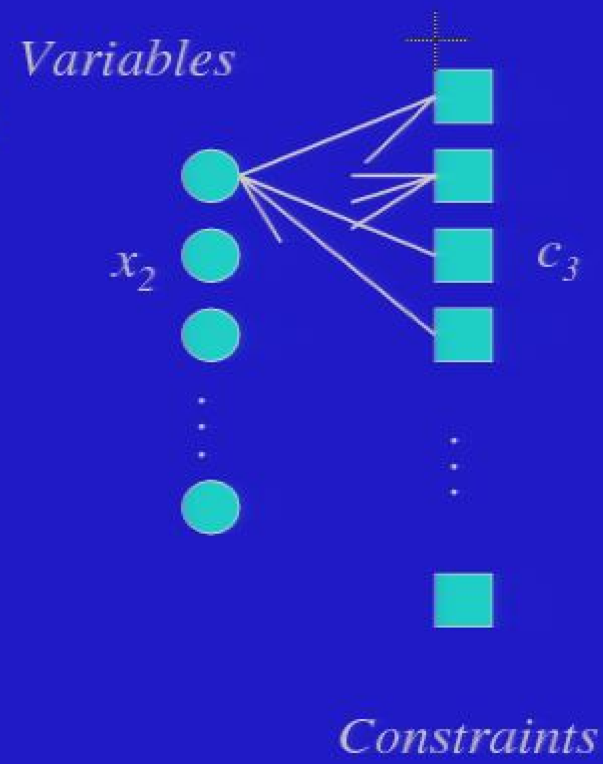
tructure.ps - GSview

Edit Options View Orientation Media Help

The Setting: Random CSPs

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

- Random bipartite graph:
- Sparse graph, i.e. $m=\Theta(n)$



dobe Acrobat Standard - [bas.pdf]
File Edit View Document Tools Advanced Window Help
Open Save Print Email Search Create PDF Review & Comment Secure Sign
Select Text 73% How To...?

Satisfiability

Given a Boolean formula (CNF), decide if a satisfying truth assignment exists.

$$(\overline{x}_{12} \vee x_5) \wedge (x_{34} \vee \overline{x}_{21} \vee x_5 \vee \overline{x}_{27}) \wedge \cdots \wedge (x_{12}) \wedge (x_{21} \vee x_9 \vee \overline{x}_{13})$$

Cook's Theorem: Satisfiability is NP-complete.

---

k-SAT: Each clause has exactly k literals.

$k = 2$: Pick any variable and set it arbitrarily. (1 choice)

Satisfy any implications (repeatedly).

Either get a subformula or a contradiction.

$k \geq 3$: NP-complete

11 x 8.5 in

dobe Acrobat Standard - [bas.pdf]

File Edit View Document Tools Advanced Window Help

Open Save Print Em Review & Comment Secure Sign

Select Text How To...?

Cascade Shift+Ctrl+J
Tile
Close All Shift+Ctrl+W
Split
Clipboard Viewer
Full Screen View Ctrl+L
1 basic.pdf
2 bas.pdf

tisfiability

Given a Boolean formula (CNF), decide if a satisfying truth assignment exists.

$$(\overline{x}_{12} \vee x_5) \wedge (x_{34} \vee \overline{x}_{21} \vee x_5 \vee \overline{x}_{27}) \wedge \cdots \wedge (x_{12}) \wedge (x_{21} \vee x_9 \vee \overline{x}_{13})$$

Cook's Theorem: Satisfiability is NP-complete.

---

k-SAT: Each clause has exactly k literals.

$k = 2$: Pick any variable and set it arbitrarily. (1 choice)

Satisfy any implications (repeatedly).

Either get a subformula or a contradiction.

$k \geq 3$: NP-complete

11 x 8.5 in

Satisfiability

Given a Boolean formula (CNF), decide if a satisfying truth assignment exists.

$$(\overline{x}_{12} \vee x_5) \wedge (x_{34} \vee \overline{x}_{21} \vee x_5 \vee \overline{x}_{27}) \wedge \cdots \wedge (x_{12}) \wedge (x_{21} \vee x_9 \vee \overline{x}_{13})$$

Cook's Theorem: Satisfiability is NP-complete.

---

k-SAT: Each clause has exactly k literals.

$k = 2$: Pick any variable and set it arbitrarily. (1 choice)

Satisfy any implications (repeatedly).

Either get a subformula or a contradiction.

$k \geq 3$: NP-complete

Random k-SAT

- Since the mid-70s a number of models have been proposed for Random SATisfiability.

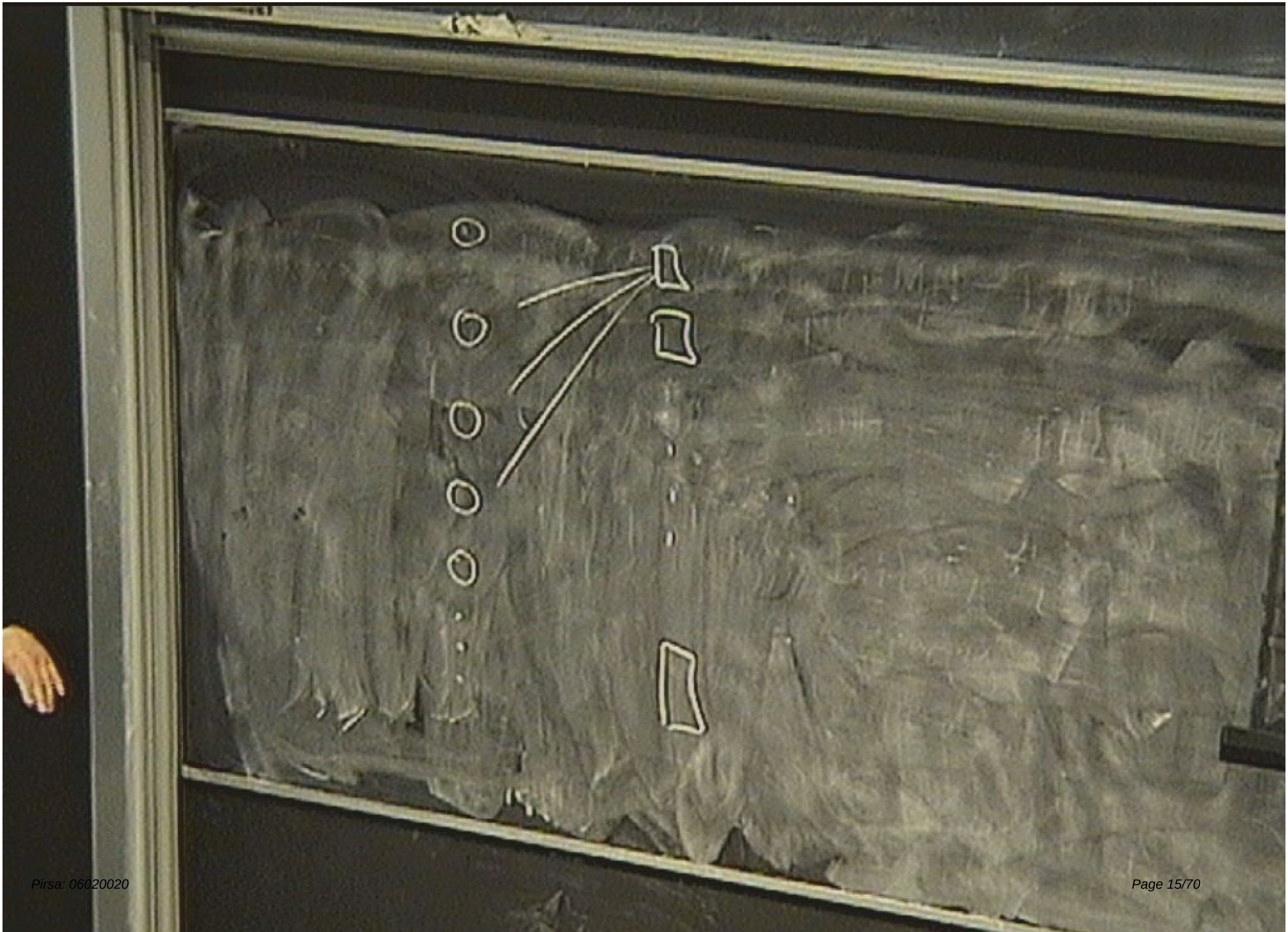
Most models generate formulas that are too easy.

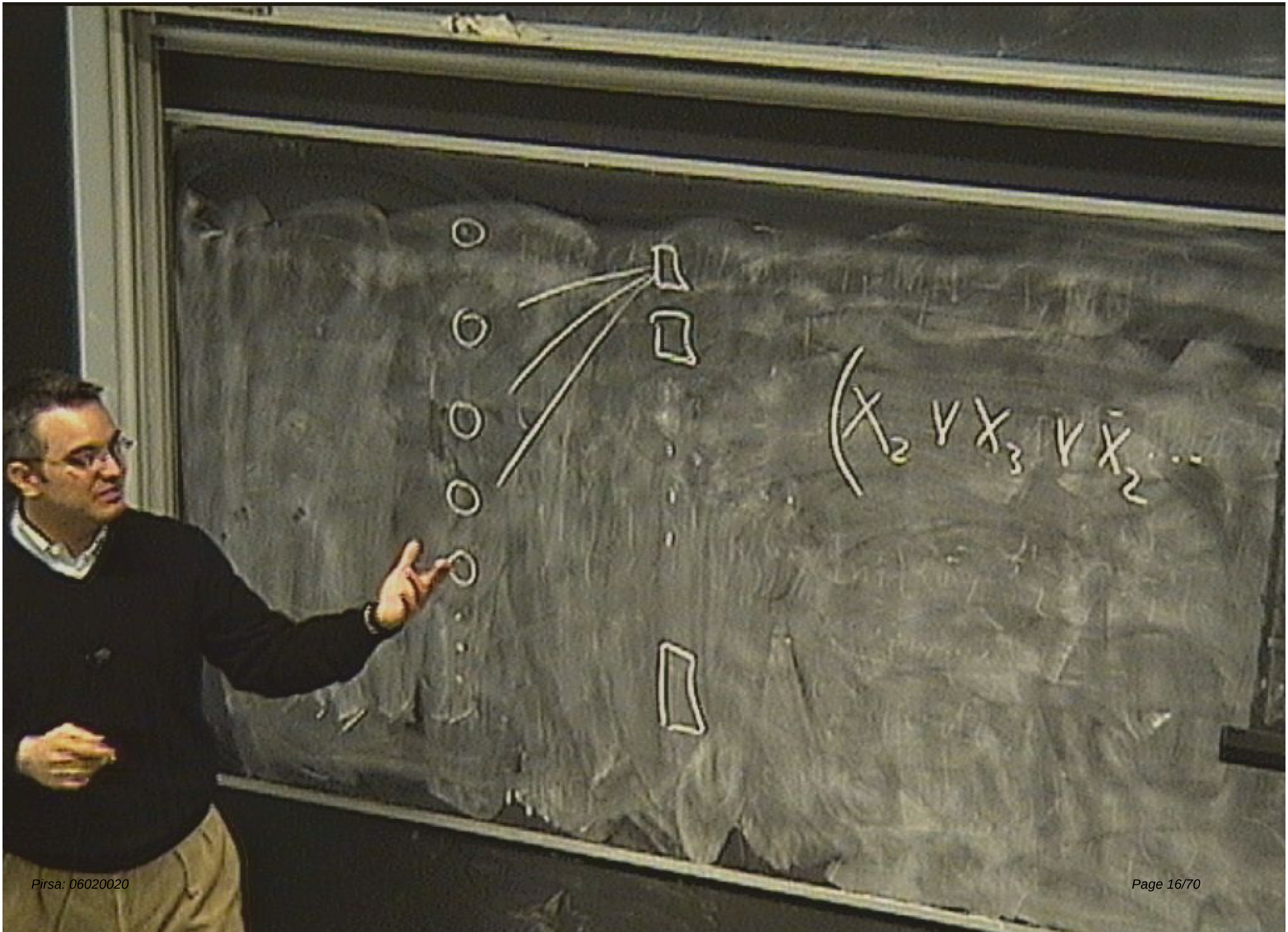
Let $A_{k,n}$ be the set of all $2^k \binom{n}{k}$ k-clauses on n variables. [with distinct, non-complementary literals]

$\mathcal{F}_k(n, m)$: a random k-SAT formula with m clauses over n variables, formed by selecting uniformly at random m clauses from $A_{k,n}$ [with replacement]

For all $k \geq 3$ and $r > 2^k$, there exists $\rho(k, r) > 0$ such that almost surely: $\mathcal{F}_k(n, rn)$ is unsatisfiable but every resolution proof of its unsatisfiability has at least $2^{\rho n}$ clauses.

[Chvátal, Szemerédi 88]



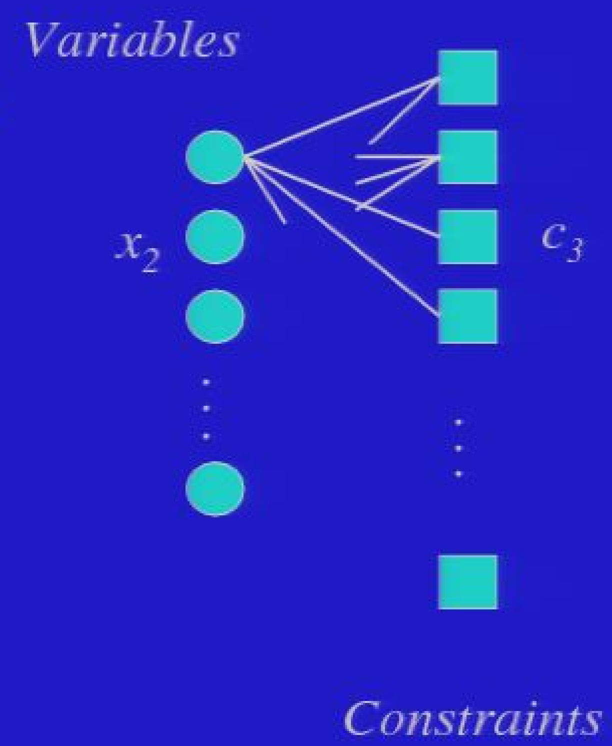

(X2 V X3 V X̄2 ...

tructure.ps - GSview
Edit Options View Orientation Media Help

The Setting: Random CSPs

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

- Random bipartite graph:
- Sparse graph, i.e. $m=\Theta(n)$



Adobe Acrobat Standard - [bas.pdf]

File Edit View Document Tools Advanced Window Help

Open Save Print Email Search Create PDF Review & Comment Secure Sign

Select Text 73% How To...?

Random k-SAT

- Since the mid-70s a number of models have been proposed for Random SATisfiability.

Most models generate formulas that are too easy.

Let $A_{k,n}$ be the set of all $2^k \binom{n}{k}$ k-clauses on n variables. [with distinct, non-complementary literals]

$\mathcal{F}_k(n, m)$: a random k-SAT formula with m clauses over n variables, formed by selecting uniformly at random m clauses from $A_{k,n}$ [with replacement]

For all $k \geq 3$ and $r > 2^k$, there exists $\rho(k, r) > 0$ such that almost surely: $\mathcal{F}_k(n, rn)$ is unsatisfiable but every resolution proof of its unsatisfiability has at least $2^{\rho n}$ clauses.

[Chvátal, Szemerédi 88]

11 x 8.5 in

Random k-SAT

- Since the mid-70s a number of models have been proposed for Random SATisfiability.

Most models generate formulas that are too easy.

Let $A_{k,n}$ be the set of all $2^k \binom{n}{k}$ k-clauses on n variables. [with distinct, non-complementary literals]

$\mathcal{F}_k(n, m)$: a random k-SAT formula with m clauses over n variables, formed by selecting uniformly at random m clauses from $A_{k,n}$ [with replacement]

For all $k \geq 3$ and $r > 2^k$, there exists $\rho(k, r) > 0$ such that almost surely: $\mathcal{F}_k(n, rn)$ is unsatisfiable but every resolution proof of its unsatisfiability has at least $2^{\rho n}$ clauses.

[Chvátal, Szemerédi 88]

Random k-SAT

- Since the mid-70s a number of models have been proposed for Random SATisfiability.

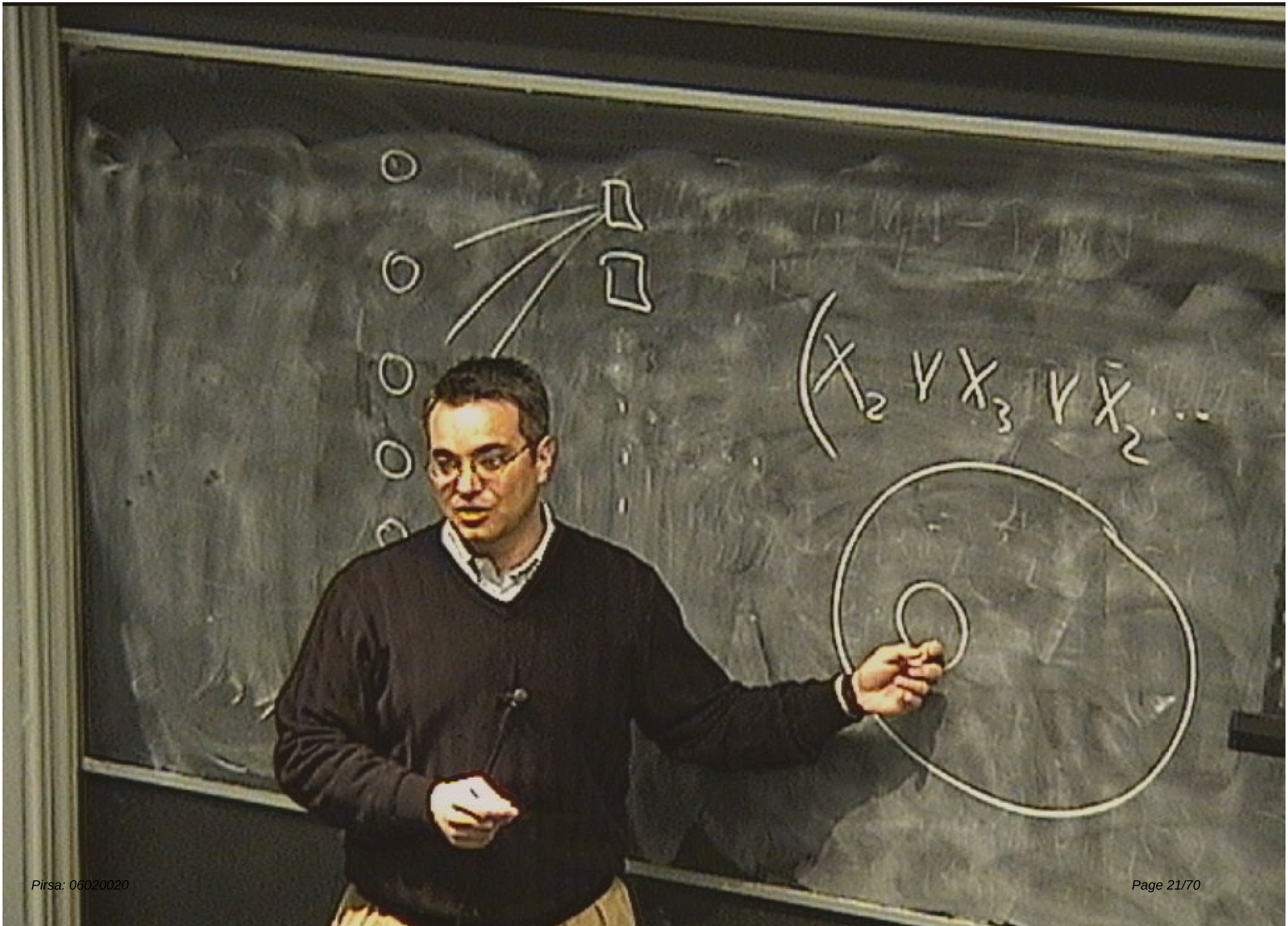
Most models generate formulas that are too easy.

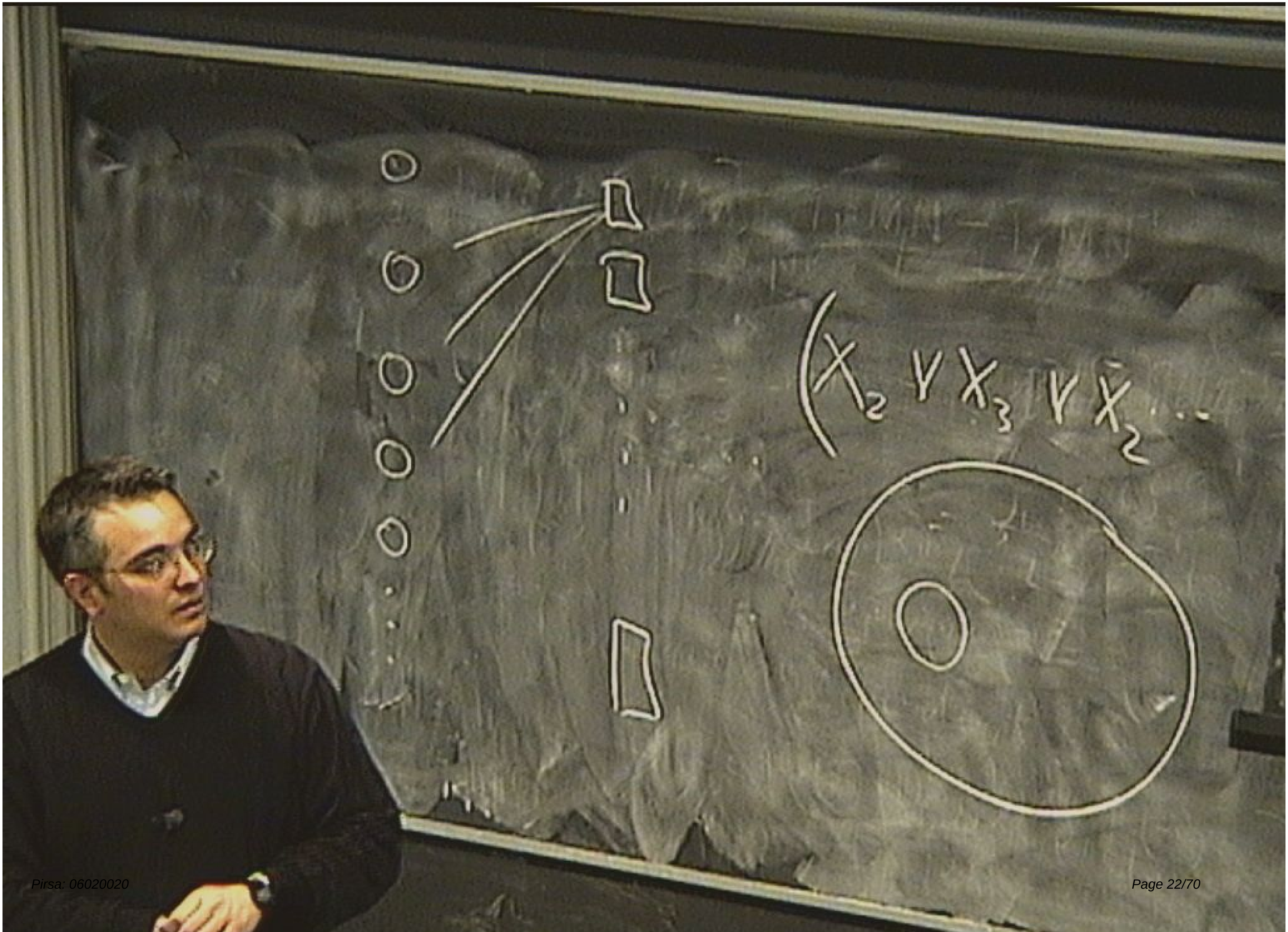
Let $A_{k,n}$ be the set of all $2^k \binom{n}{k}$ k-clauses on n variables. [with distinct, non-complementary literals]

$\mathcal{F}_k(n, m)$: a random k-SAT formula with m clauses over n variables, formed by selecting uniformly at random m clauses from $A_{k,n}$ [with replacement]

For all $k \geq 3$ and $r > 2^k$, there exists $\rho(k, r) > 0$ such that almost surely: $\mathcal{F}_k(n, rn)$ is unsatisfiable but every resolution proof of its unsatisfiability has at least $2^{\rho n}$ clauses.

[Chvátal, Szemerédi 88]


(X2 V X3 V X̄2 ...


$(X_2 \vee X_3 \vee \bar{X}_2 \ldots$

Random k-SAT

- Since the mid-70s a number of models have been proposed for Random SATisfiability.

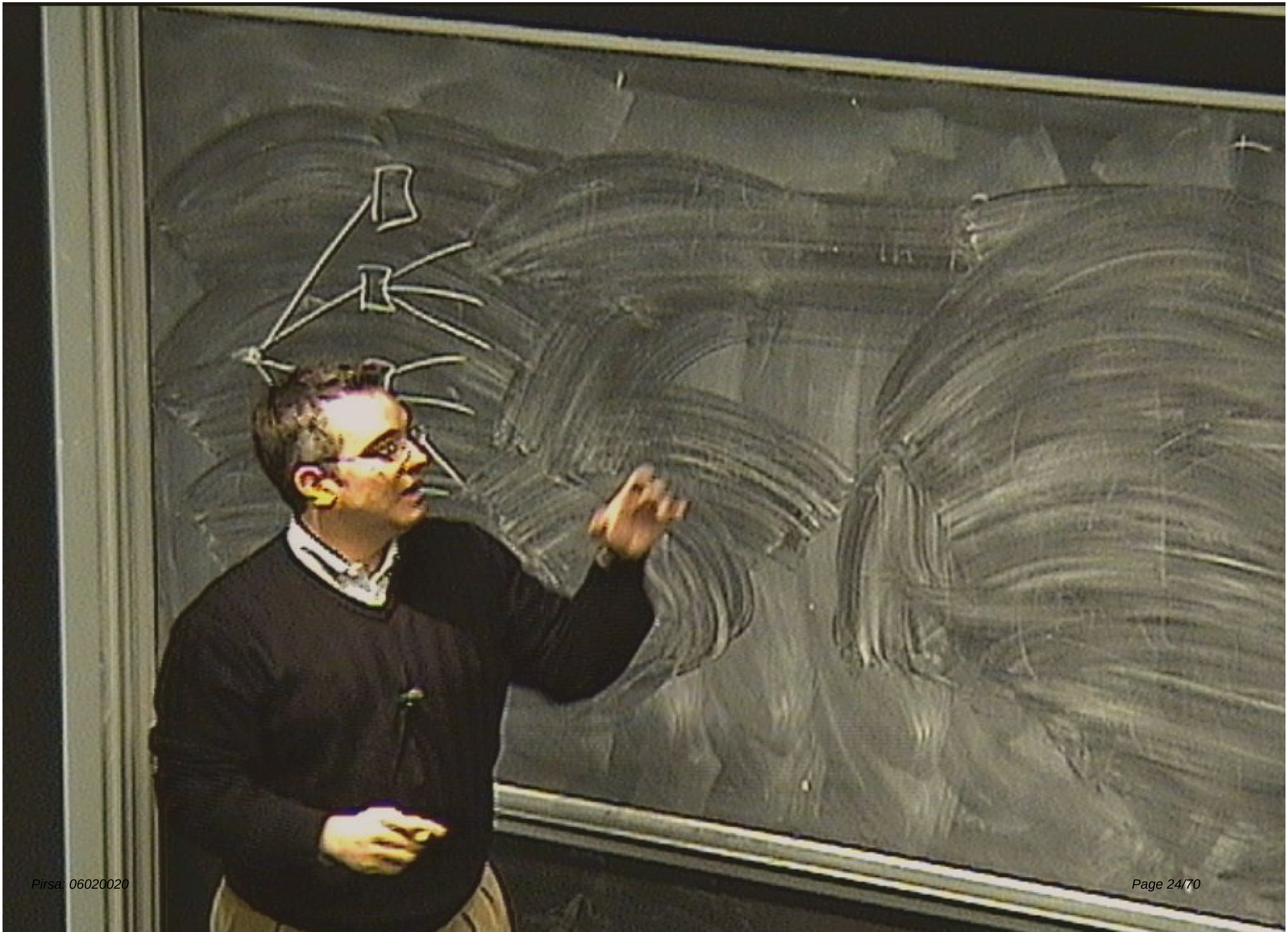
Most models generate formulas that are too easy.

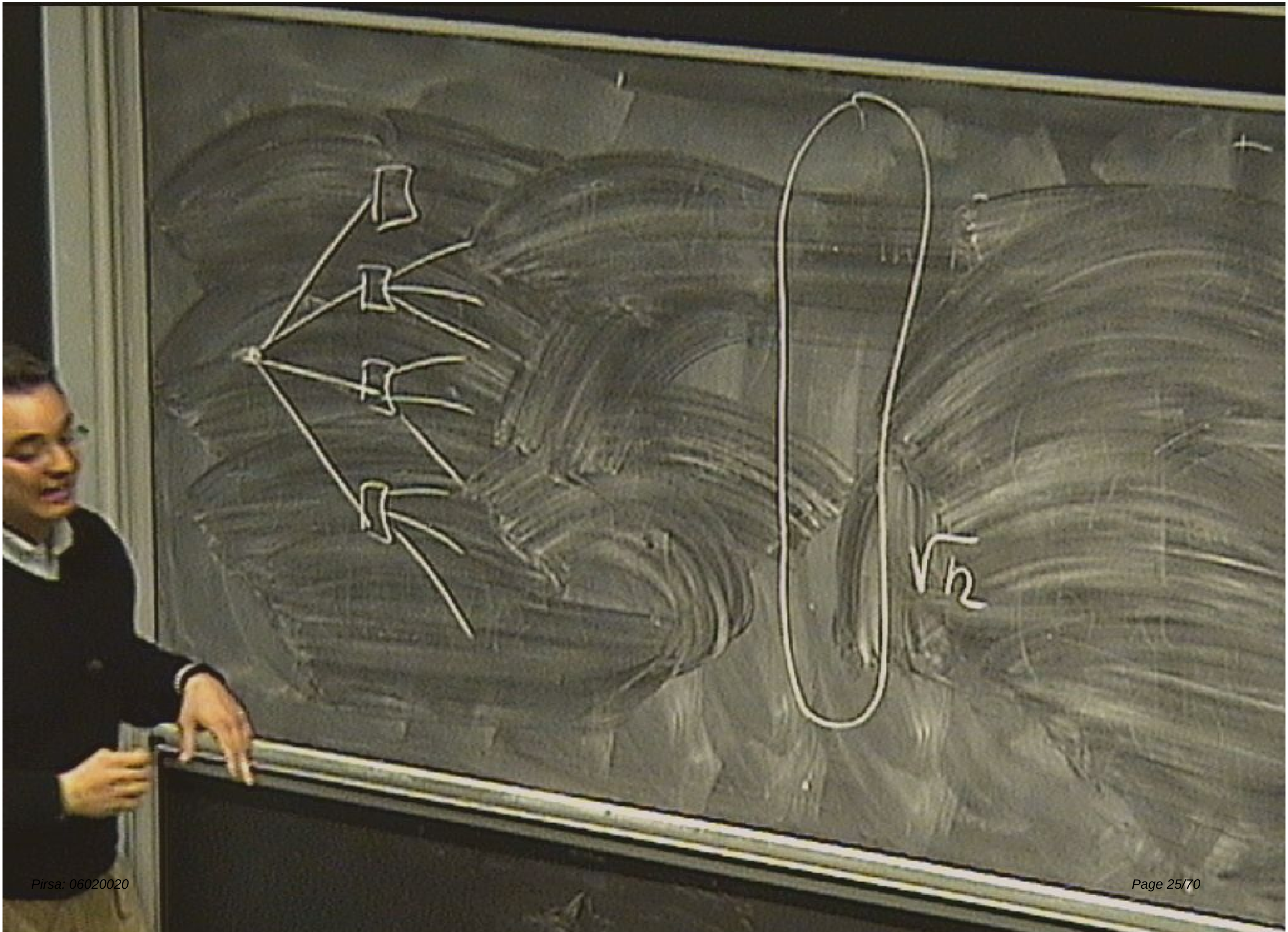
Let $A_{k,n}$ be the set of all $2^k\binom{n}{k}$ k-clauses on n variables. [with distinct, non-complementary literals]

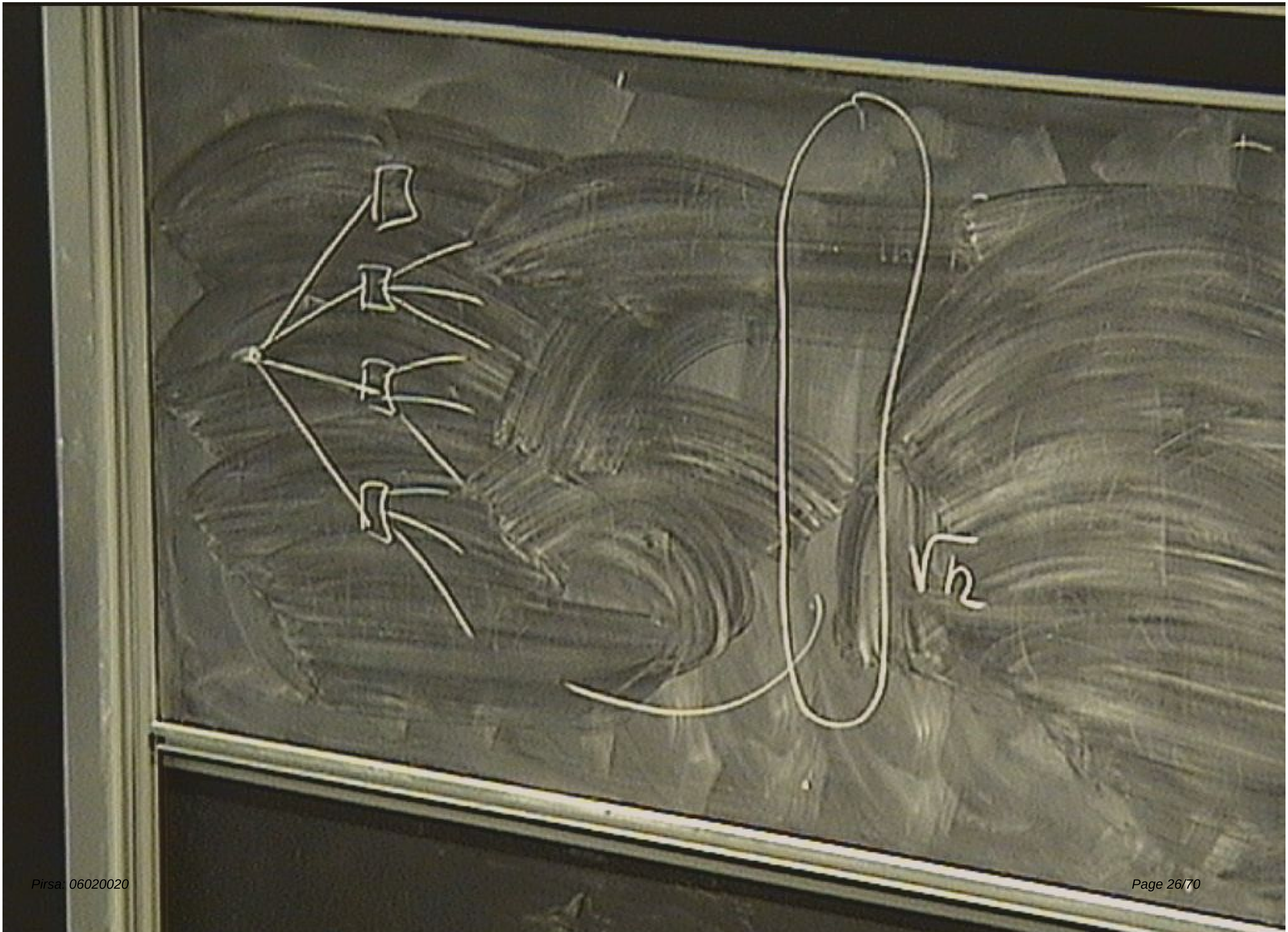
$\mathcal{F}_k(n, m)$: a random k-SAT formula with m clauses over n variables, formed by selecting uniformly at random m clauses from $A_{k,n}$ [with replacement]

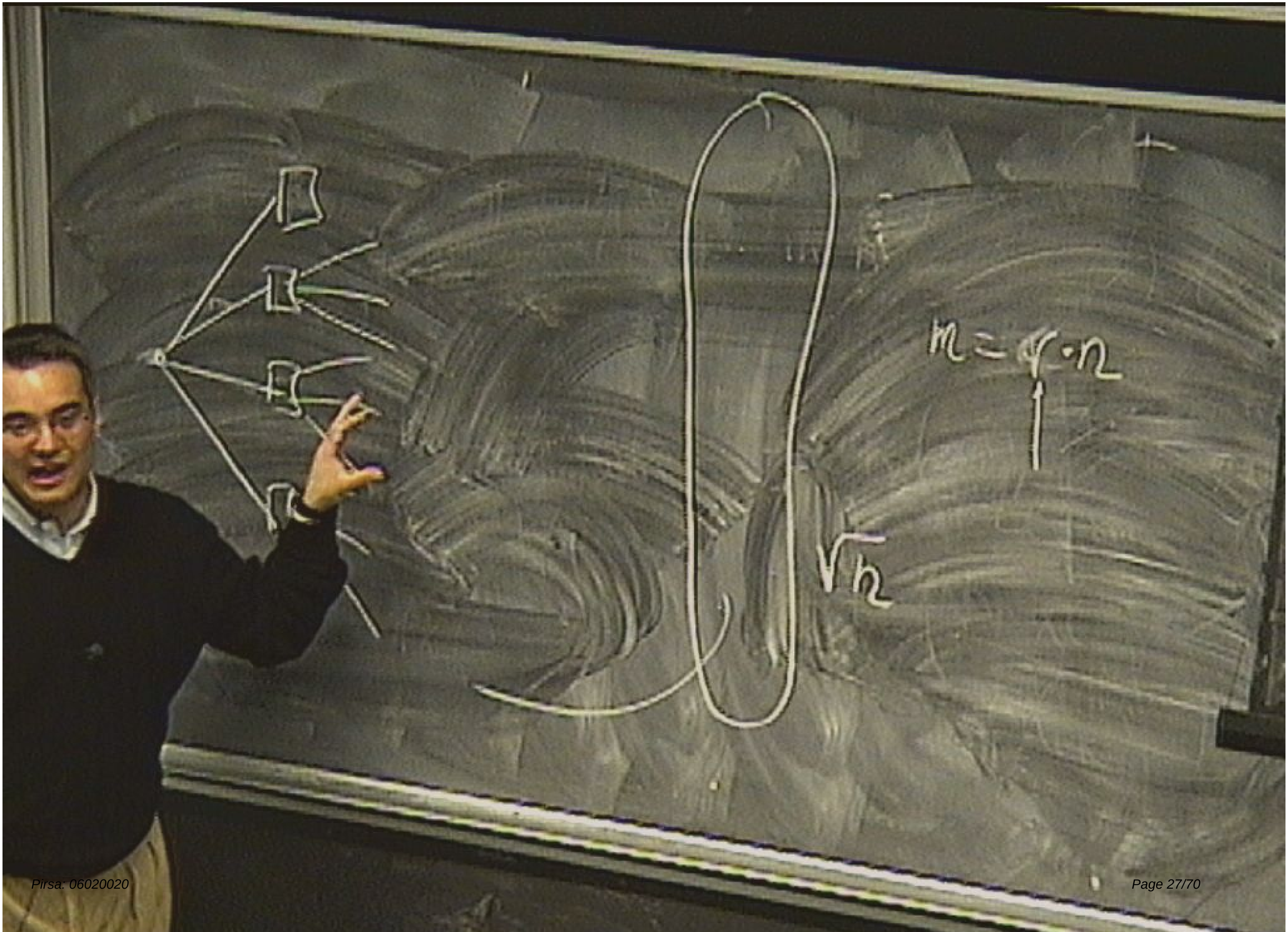
For all $k \geq 3$ and $r > 2^k$, there exists $\rho(k, r) > 0$ such that almost surely: $\mathcal{F}_k(n, rn)$ is unsatisfiable but every resolution proof of its unsatisfiability has at least $2^{\rho n}$ clauses.

[Chvátal, Szemerédi 88]




√n


√h


m = q · n
√n

Random k-SAT

- Since the mid-70s a number of models have been proposed for Random SATisfiability.

Most models generate formulas that are too easy.

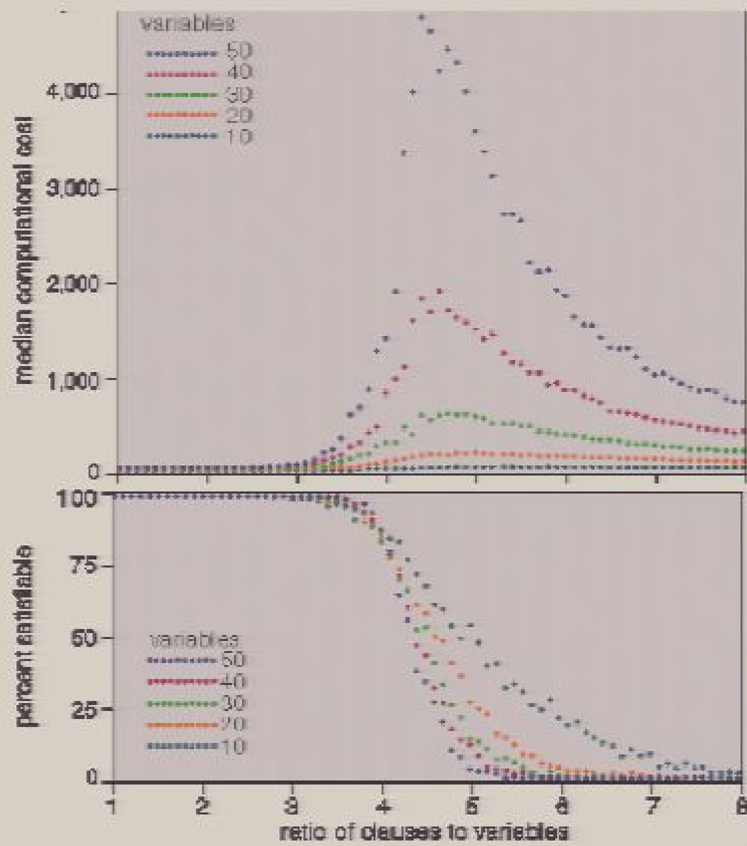
Let $A_{k,n}$ be the set of all $2^k\binom{n}{k}$ k-clauses on n variables. [with distinct, non-complementary literals]

$\mathcal{F}_k(n,m)$: a random k-SAT formula with m clauses over n variables, formed by selecting uniformly at random m clauses from $A_{k,n}$ [with replacement]

For all $k \geq 3$ and $r > 2^k$, there exists $\rho(k,r) > 0$ such that almost surely: $\mathcal{F}_k(n, rn)$ is unsatisfiable but every resolution proof of its unsatisfiability has at least $2^{\rho n}$ clauses.

[Chvátal, Szemerédi 88]

Satisfiability Threshold Conjecture



[Mitchel, Selman, Levesque 92]

Conjecture: For each k, there exists a constant r_k such that for any $\epsilon > 0$,

$$\lim_{n\to\infty} \Pr[\mathcal{F}_k(n,m) \text{is satisfiable}] = \begin{cases} 1 & \text{if } m = (r_k - \epsilon)n \\ 0 & \text{if } m = (r_k + \epsilon)n \end{cases}$$

Known Results

- $k = 2$: Yes, $r_2 = 1$. [Chvátal,Reed 92], [Goerdt 92], [Fernandez de la Vega 92]

Idea: Look at the "forced choices" branching process.

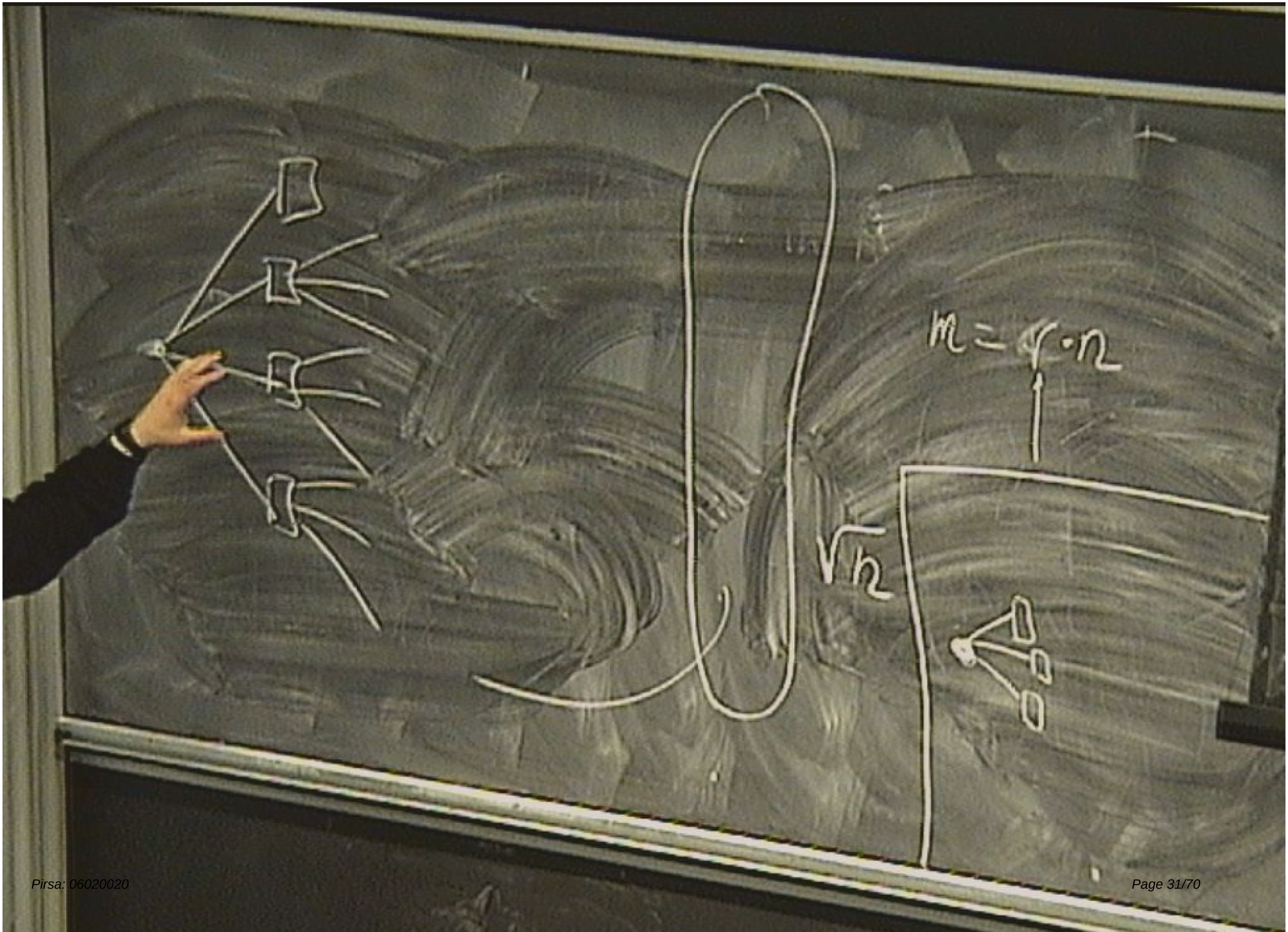
- $k \geq 3$: We don't know if r_k exists.

- Easy bounds: $$\frac{2^k}{k} < r_k < 2^k.$$

- [Friedgut 97]: For each $k \geq 2$ there exists a function $r_k(n)$ such that

$$\lim_{n\to\infty} \Pr[\mathcal{F}_k(n,m) \text{ is satisfiable}] = \begin{cases} 1 & \text{if } m = (r_k(n) - \epsilon)n \\ 0 & \text{if } m = (r_k(n) + \epsilon)n \end{cases}$$

Idea: All small subformulas are innocuous.


m = r·n
Vh

Known Results

- $k = 2$: Yes, $r_2 = 1$. [Chvátal,Reed 92], [Goerdt 92], [Fernandez de la Vega 92]

Idea: Look at the "forced choices" branching process.

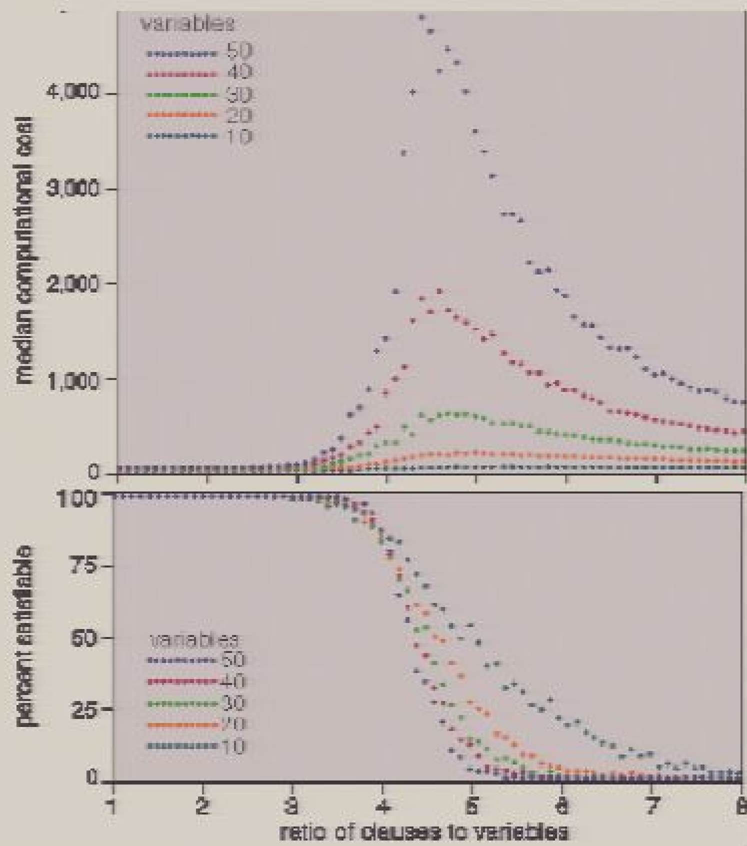
- $k \geq 3$: We don't know if r_k exists.
- Easy bounds: $$\frac{2^k}{k} < r_k < 2^k.$$

- [Friedgut 97]: For each $k \geq 2$ there exists a function $r_k(n)$ such that

$$\lim_{n \to \infty} \Pr[\mathcal{F}_k(n, m) \text{ is satisfiable}] = \begin{cases} 1 & \text{if } m = (r_k(n) - \epsilon)n \\ 0 & \text{if } m = (r_k(n) + \epsilon)n \end{cases}$$

Idea: All small subformulas are innocuous.

Satisfiability Threshold Conjecture



[Mitchel, Selman, Levesque 92]

Conjecture: For each k, there exists a constant r_k such that for any $\epsilon > 0$,

$$\lim_{n\to\infty} \Pr[\mathcal{F}_k(n,m) \text{is satisfiable}] = \begin{cases} 1 & \text{if } m = (r_k - \epsilon)n \\ 0 & \text{if } m = (r_k + \epsilon)n \end{cases}$$

Known Results

- $k = 2$: Yes, $r_2 = 1$. [Chvátal,Reed 92], [Goerdt 92], [Fernandez de la Vega 92]

Idea: Look at the "forced choices" branching process.

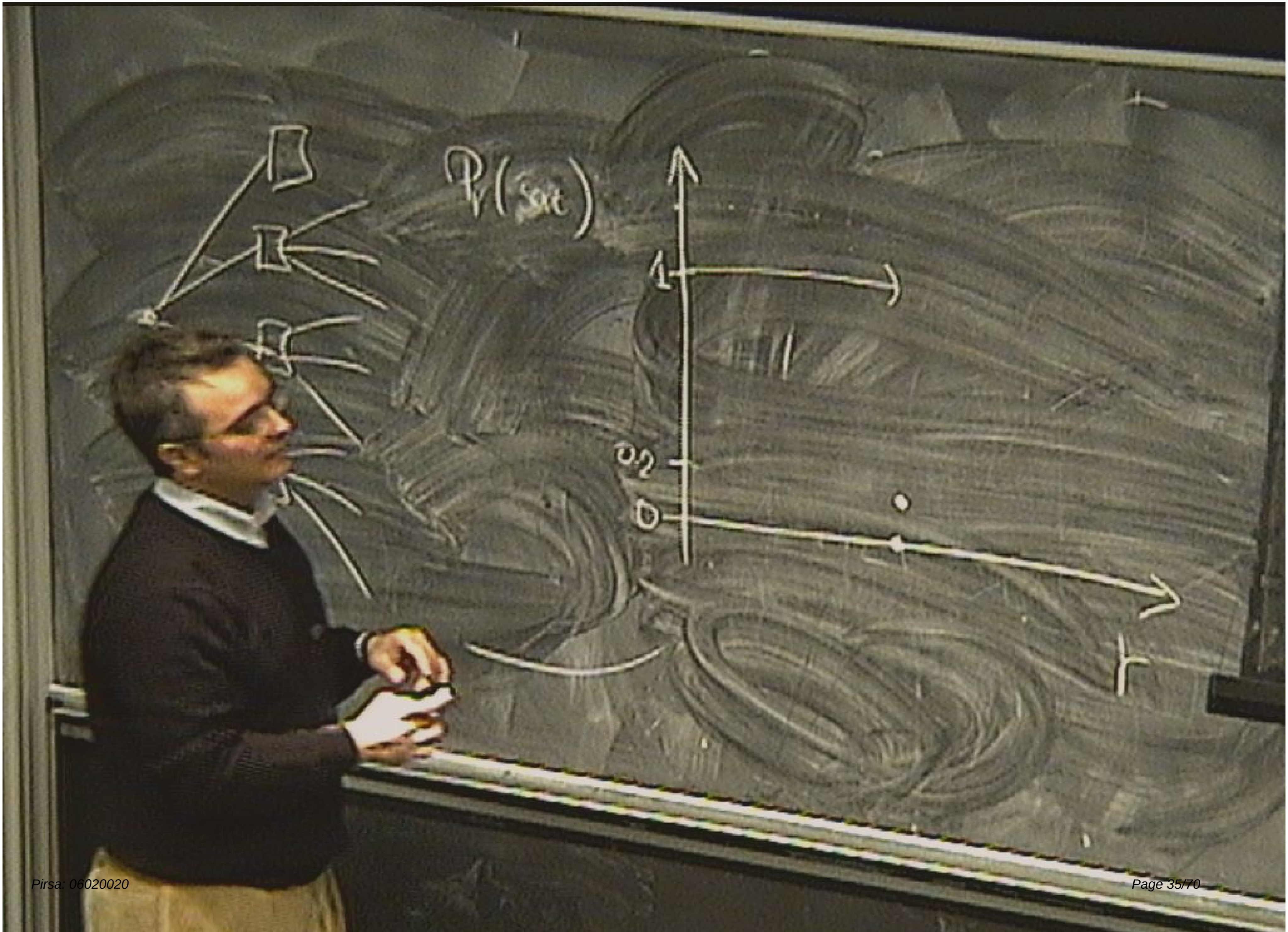
- $k \geq 3$: We don't know if r_k exists.

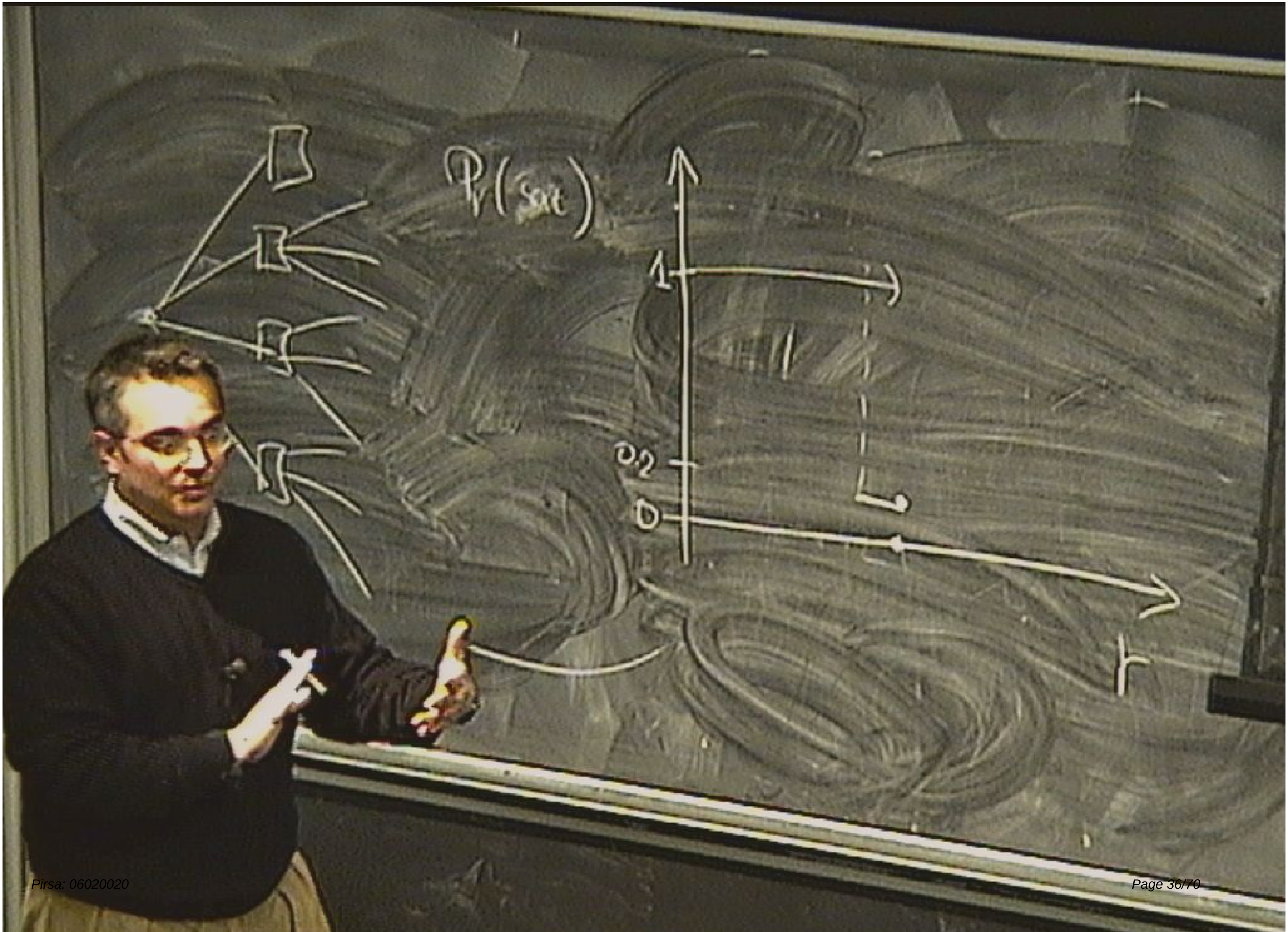
- Easy bounds:
$$\frac{2^k}{k} < r_k < 2^k.$$

- [Friedgut 97]: For each $k \geq 2$ there exists a function $r_k(n)$ such that

$$\lim_{n \to \infty} \Pr[\mathcal{F}_k(n,m) \text{ is satisfiable}] = \begin{cases} 1 & \text{if } m = (r_k(n) - \epsilon)n \\ 0 & \text{if } m = (r_k(n) + \epsilon)n \end{cases}$$

Idea: All small subformulas are innocuous.


Pr(Sat)
1
0.2
0


Pr(sat)
1
0.2
0
r

Known Results

- $k = 2$: Yes, $r_2 = 1$. [Chvátal,Reed 92], [Goerdt 92], [Fernandez de la Vega 92]

Idea: Look at the "forced choices" branching process.

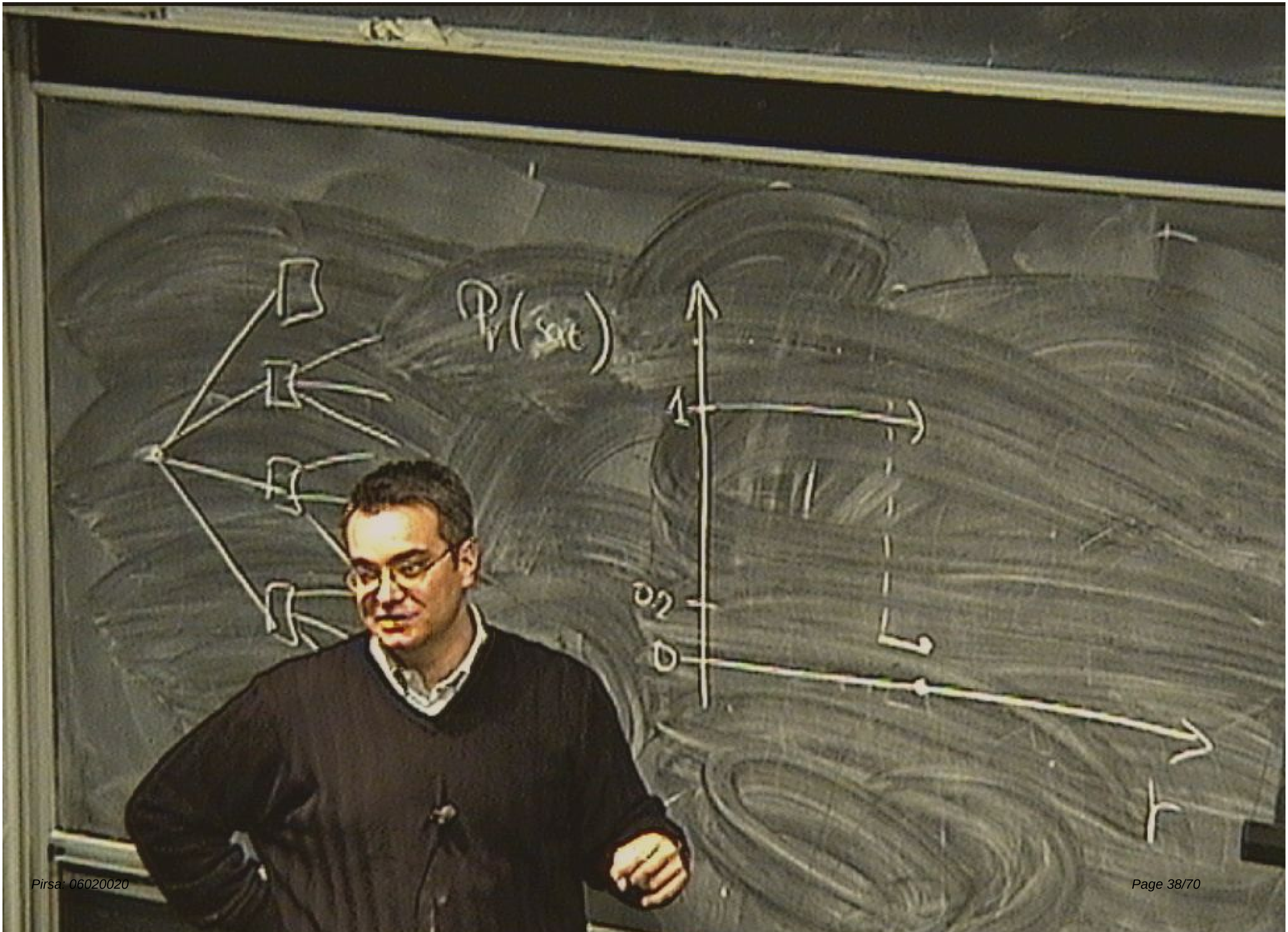
- $k \geq 3$: We don't know if r_k exists.

- Easy bounds: $$\frac{2^k}{k} < r_k < 2^k.$$

- [Friedgut 97]: For each $k \geq 2$ there exists a function $r_k(n)$ such that

$$\lim_{n \to \infty} \Pr[\mathcal{F}_k(n,m) \text{ is satisfiable}] = \begin{cases} 1 & \text{if } m = (r_k(n) - \epsilon)n \\ 0 & \text{if } m = (r_k(n) + \epsilon)n \end{cases}$$

Idea: All small subformulas are innocuous.



Pr(succ)
1
1/2
0
r

Known Results

- $k = 2$: Yes, $r_2 = 1$. [Chvátal,Reed 92], [Goerdt 92], [Fernandez de la Vega 92]

Idea: Look at the "forced choices" branching process.

- $k \geq 3$: We don't know if r_k exists.

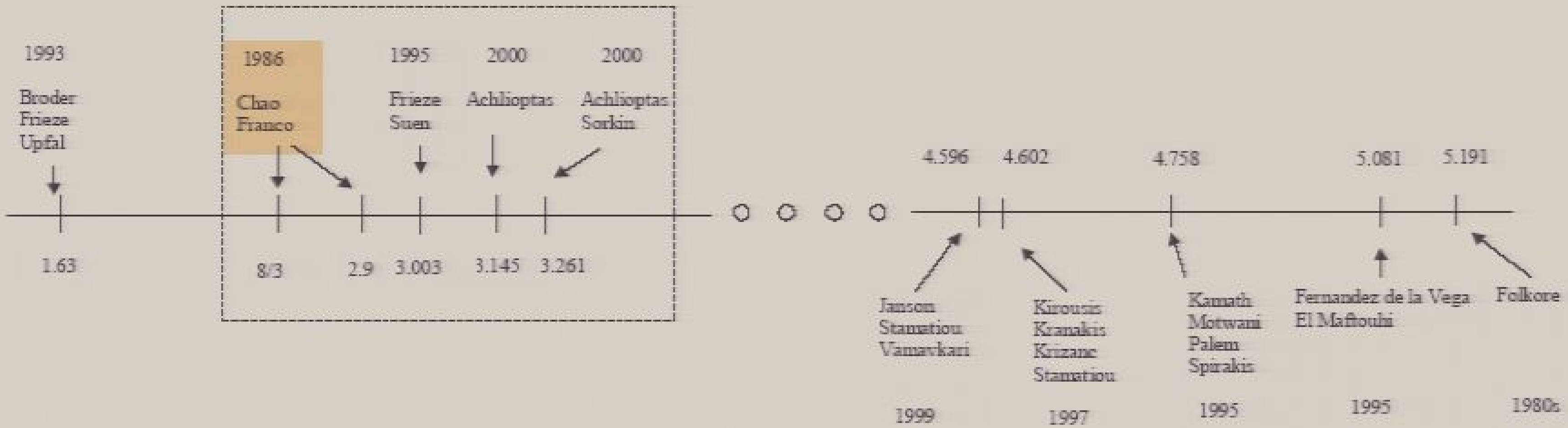
- Easy bounds: $$\frac{2^k}{k} < r_k < 2^k.$$

- [Friedgut 97]: For each $k \geq 2$ there exists a function $r_k(n)$ such that

$$\lim_{n \to \infty} \Pr[\mathcal{F}_k(n,m) \text{ is satisfiable}] = \begin{cases} 1 & \text{if } m = (r_k(n) - \epsilon)n \\ 0 & \text{if } m = (r_k(n) + \epsilon)n \end{cases}$$

Idea: All small subformulas are innocuous.

Random 3-SAT



- Upper bounds come from probabilistic counting arguments.
- Pure literal heuristic: satisfy only literals whose complement does not appear in the formula. Exact analysis gives $\gamma_3 = 1.637...$

Unit-Clause Propagation (and Extensions)

If there exist 1-clauses (unit clauses)
then
pick a 1-clause u.a.r. and satisfy it
else
select a literal ℓ and satisfy it

- Value assignments are permanent (no backtracking)
- Failure occurs iff a 0-clause is ever generated
- The algorithm goes on to set all the variables even if a 0-clause is generated

select

UC: Pick a variable x u.a.r.; select $\ell \in \{x, \overline{x}\}$ u.a.r. $8/3$

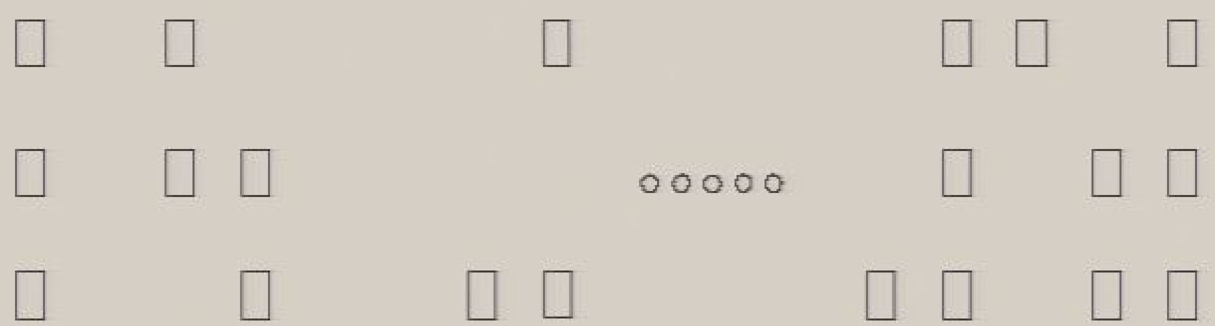
UCwm: Pick a variable x u.a.r.; select $\ell \in \{x, \overline{x}\}$ that appears among more 3-clauses. 2.9

GUC: Pick a shortest clause $c = (\ell_1 \vee \cdots \ell_q)$ u.a.r.; select $\ell \in \{\ell_1, \ldots, \ell_q\}$ u.a.r. 3.003

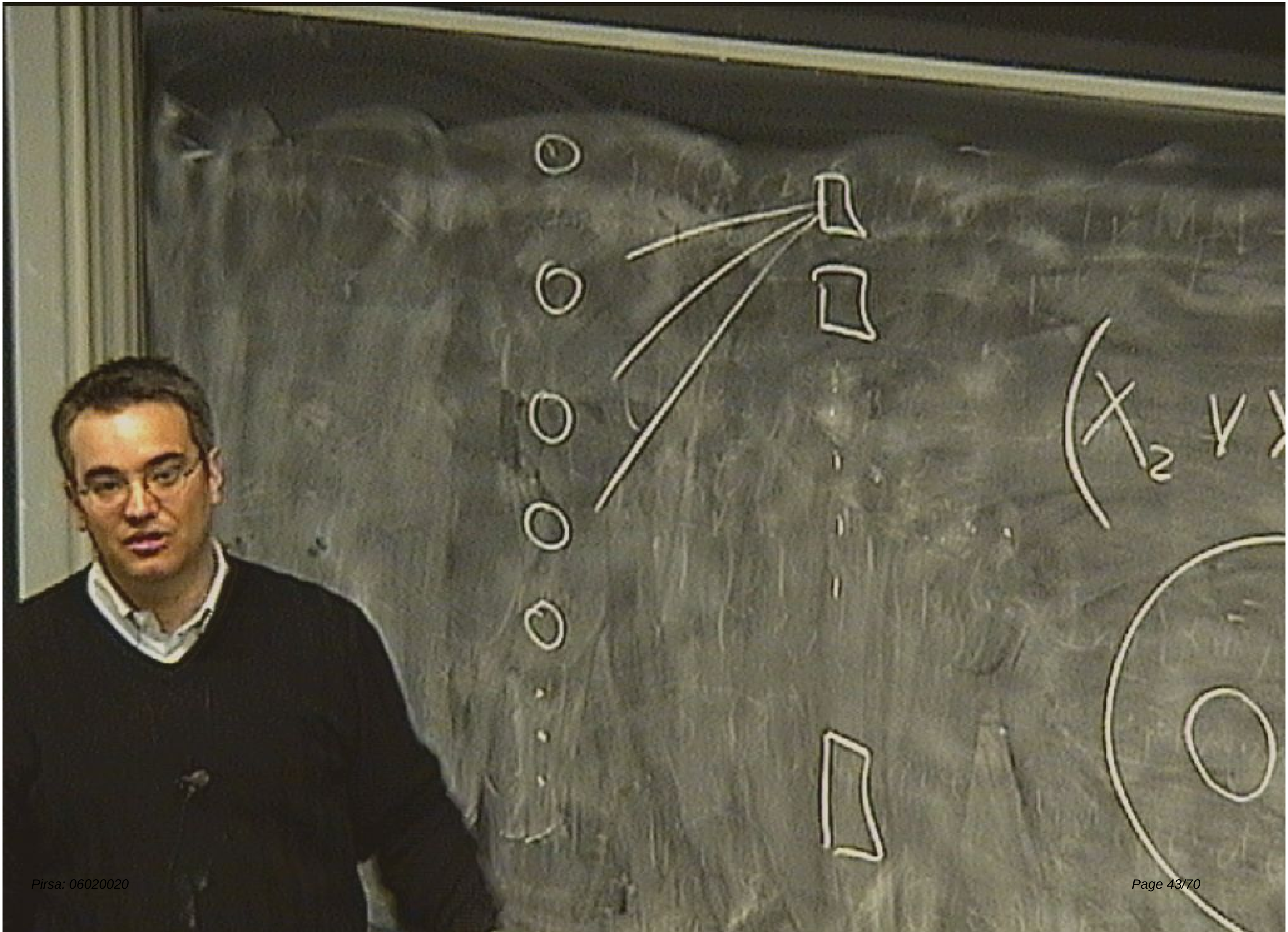
Uniform Randomness

For all $0 \leq i \leq 3$ and all $0 \leq t \leq n$:

The set of i-clauses remaining after t steps is uniformly random conditional on its size.



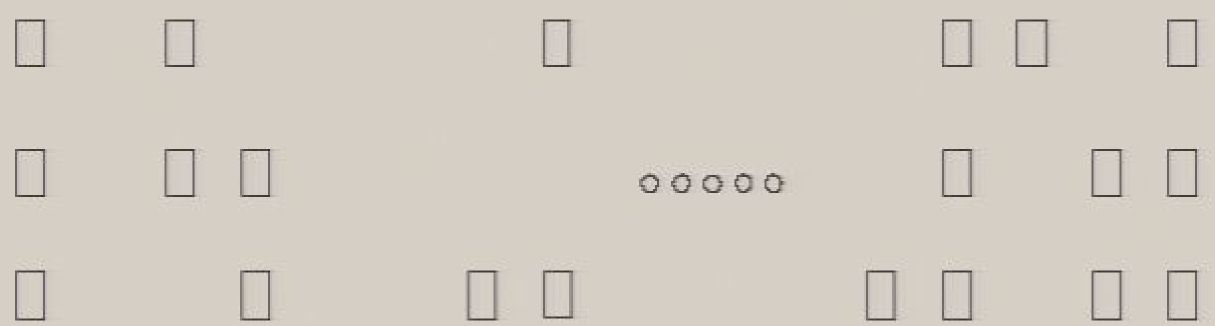
- Initially, all cards are "face down"; 3 cards per clause.
- We can $\begin{cases} \text{Name a variable} \\ \quad\text{or} \\ \text{Point to a card} \end{cases}$
- As a result, all cards with the named/underlying variable turn "face up".
- After we set the variable: all cards corresponding to satisfied clauses get removed;
  all cards corresponding to the unsatisfied literal get removed.



Uniform Randomness

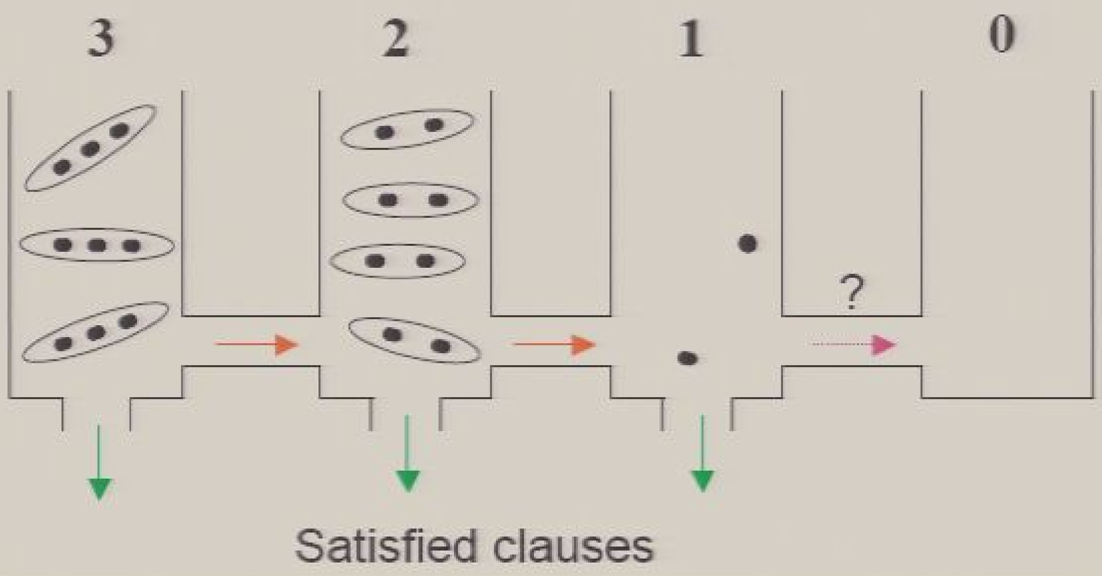
For all $0 \leq i \leq 3$ and all $0 \leq t \leq n$:

The set of i-clauses remaining after t steps is uniformly random conditional on its size.



- Initially, all cards are "face down"; 3 cards per clause.
- We can { Name a variable / or / Point to a card
- As a result, all cards with the named/underlying variable turn "face up".
- After we set the variable: all cards corresponding to satisfied clauses get removed;
  all cards corresponding to the unsatisfied literal get removed.

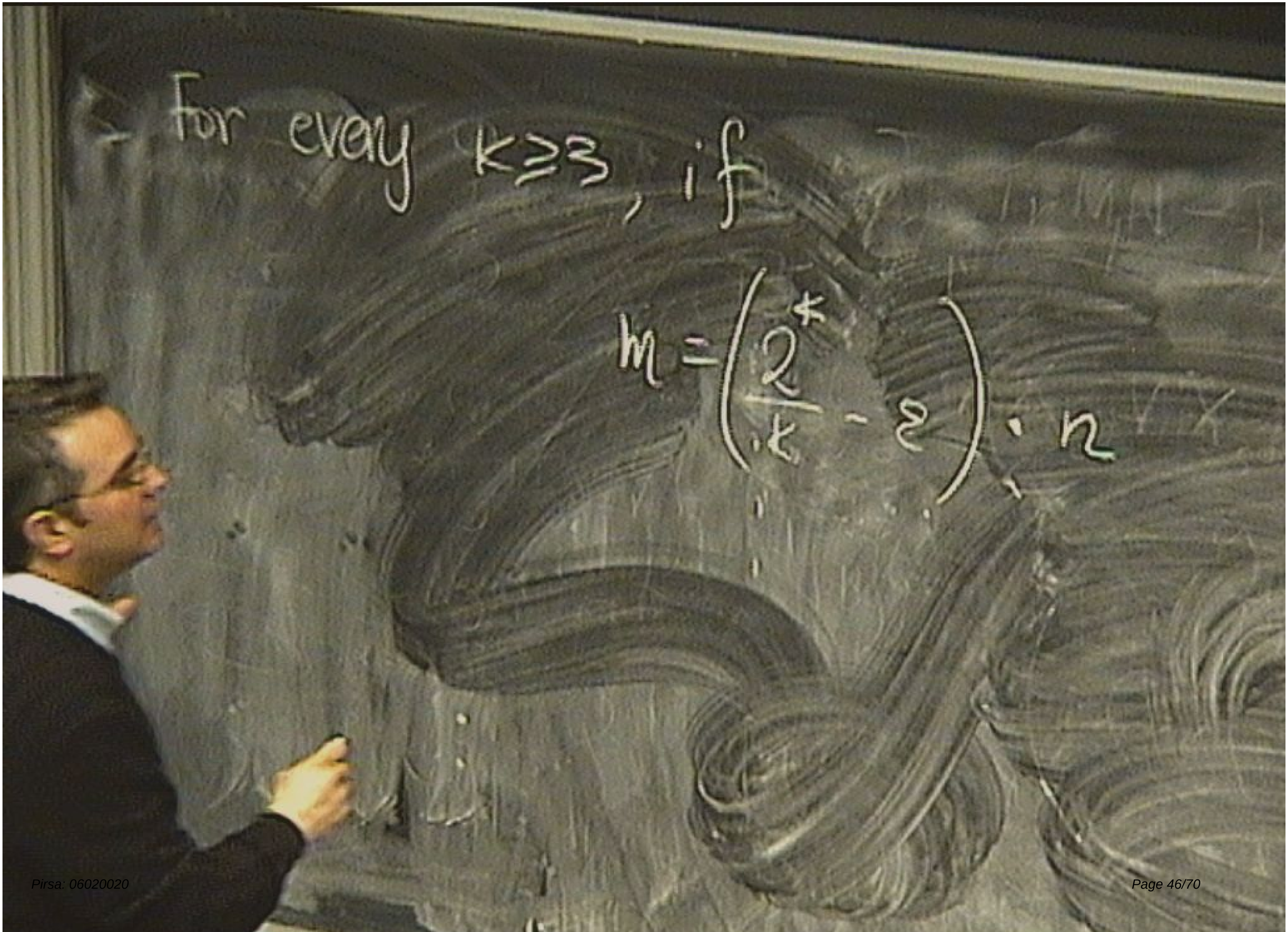
Flows and Buckets

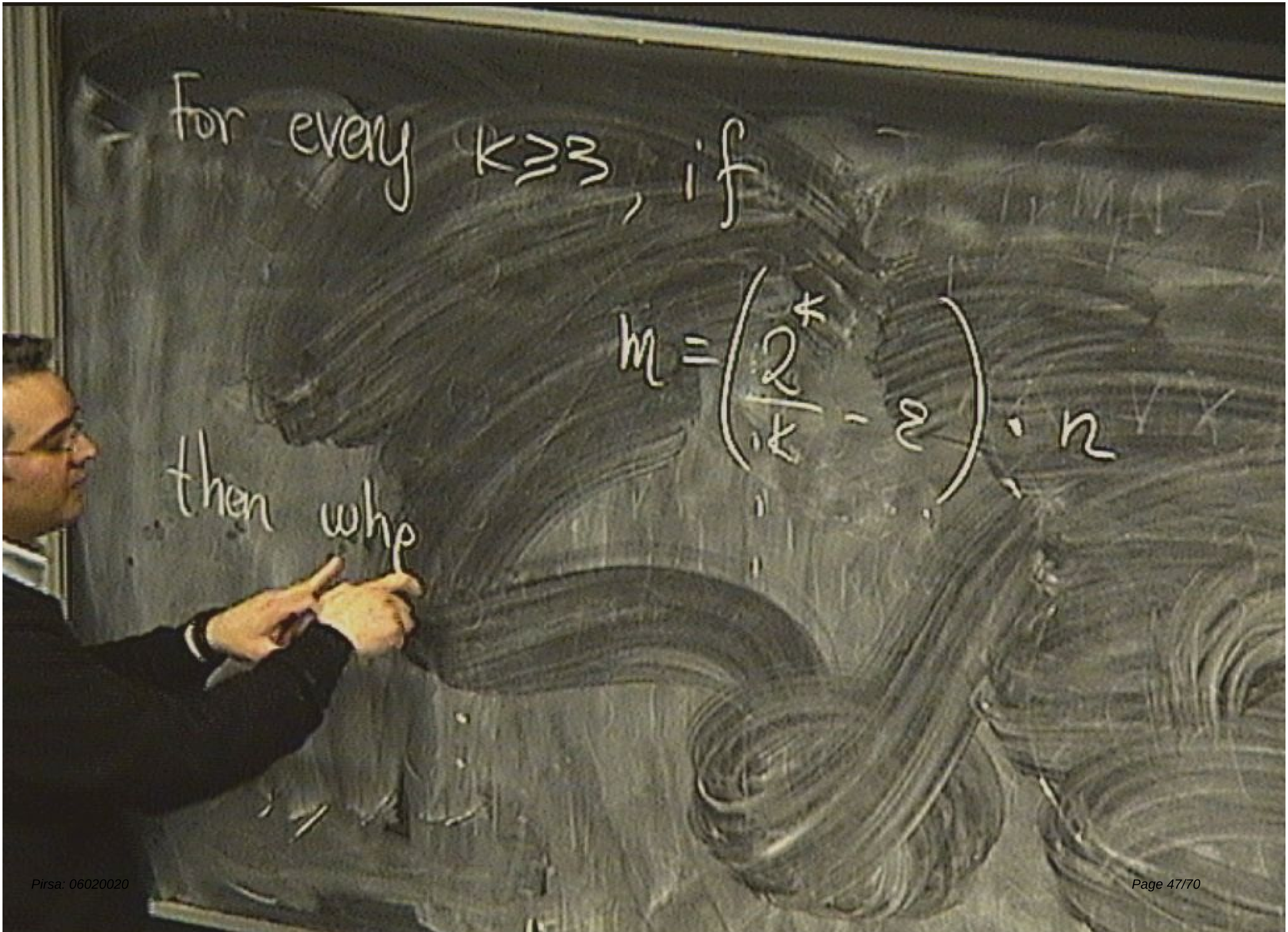


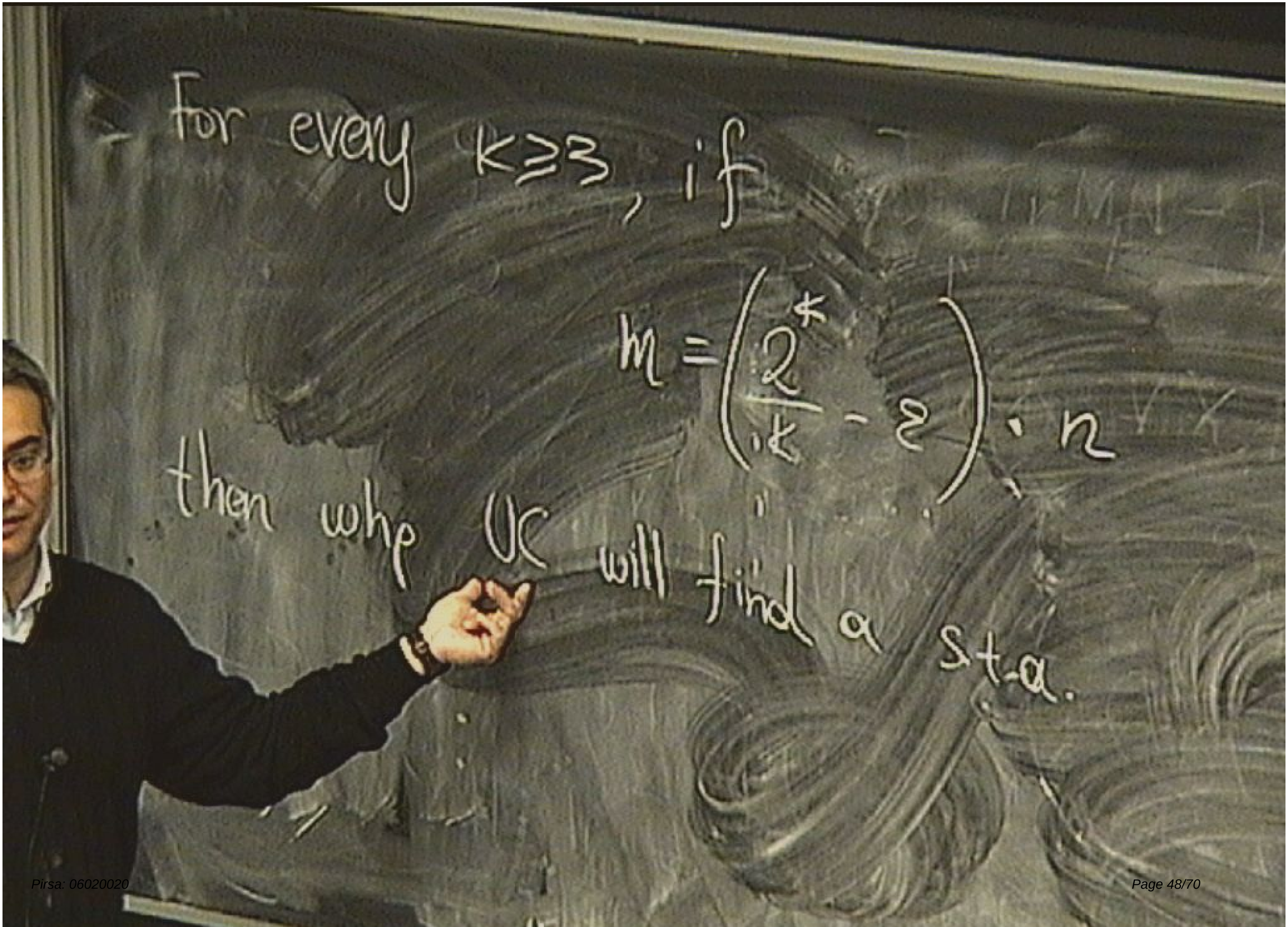
$C_i(t)$ is the number of i-clauses remaining after t variables are set.

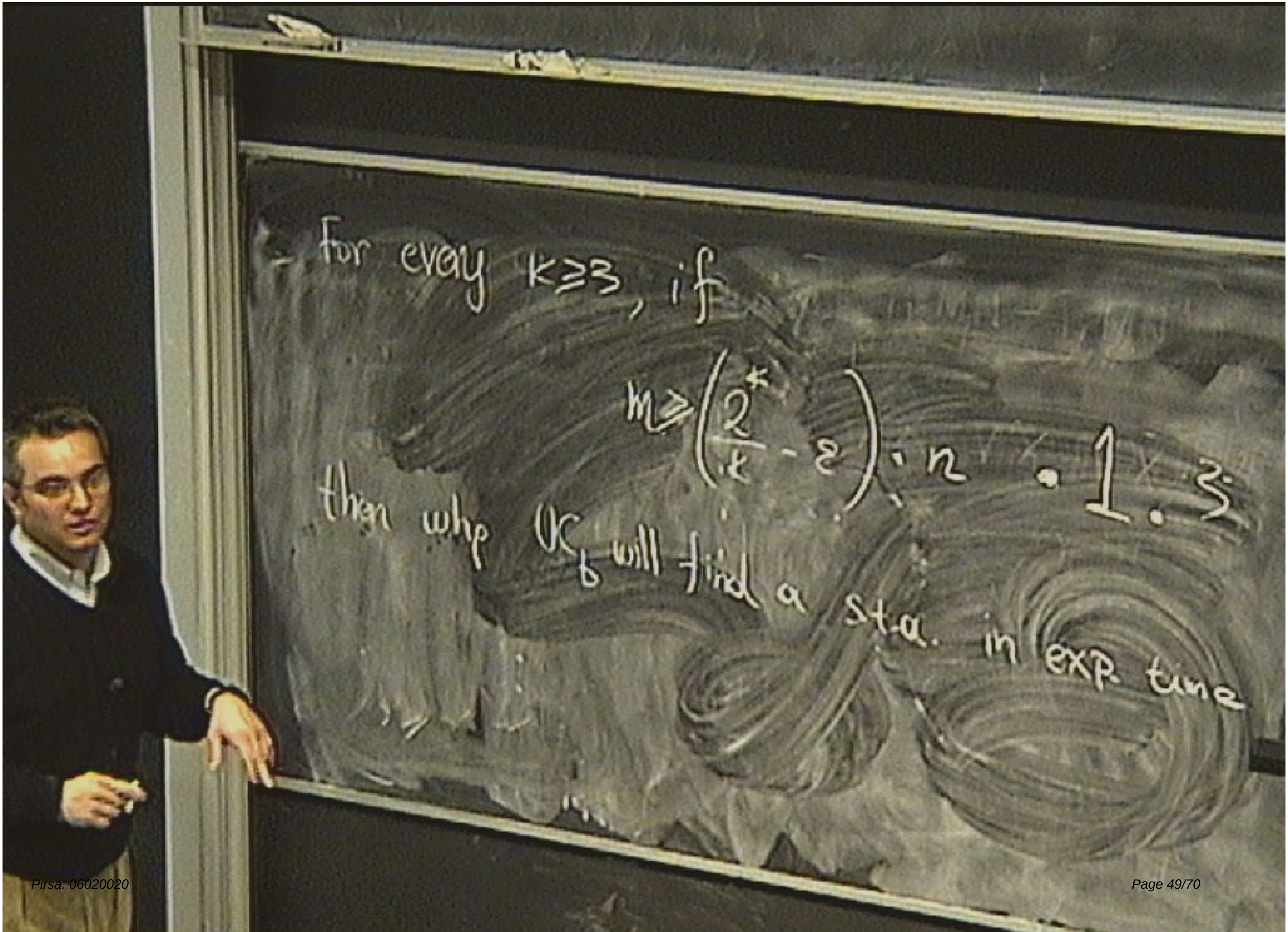
- If for some t, $\dfrac{C_2(t)}{n-t} > (1+\delta)$ the algorithm will a.s. fail.
- The expected number of 1-clauses generated in round t is $\dfrac{C_2(t)}{n-t} + o(1)$.
- If for all t, $\dfrac{C_2(t)}{n-t} < (1-\delta)$ the algorithm succeeds with probability at least $\psi = \psi(\delta) > 0$.

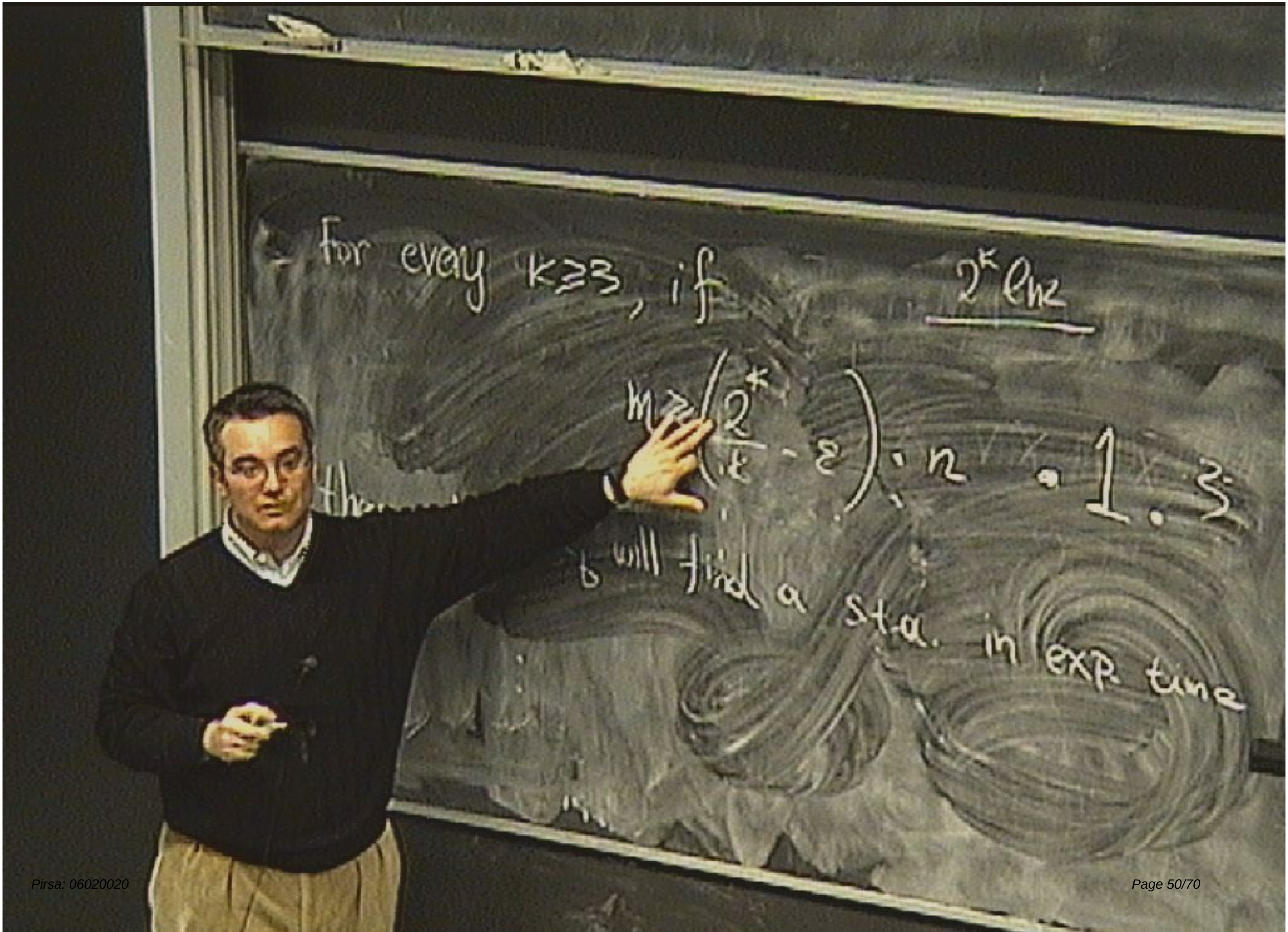
If for some r^* we can show that a.s. $\dfrac{C_2(t)}{n-t} < (1-\delta)$ for all t then $r_3 \geq r^*$.

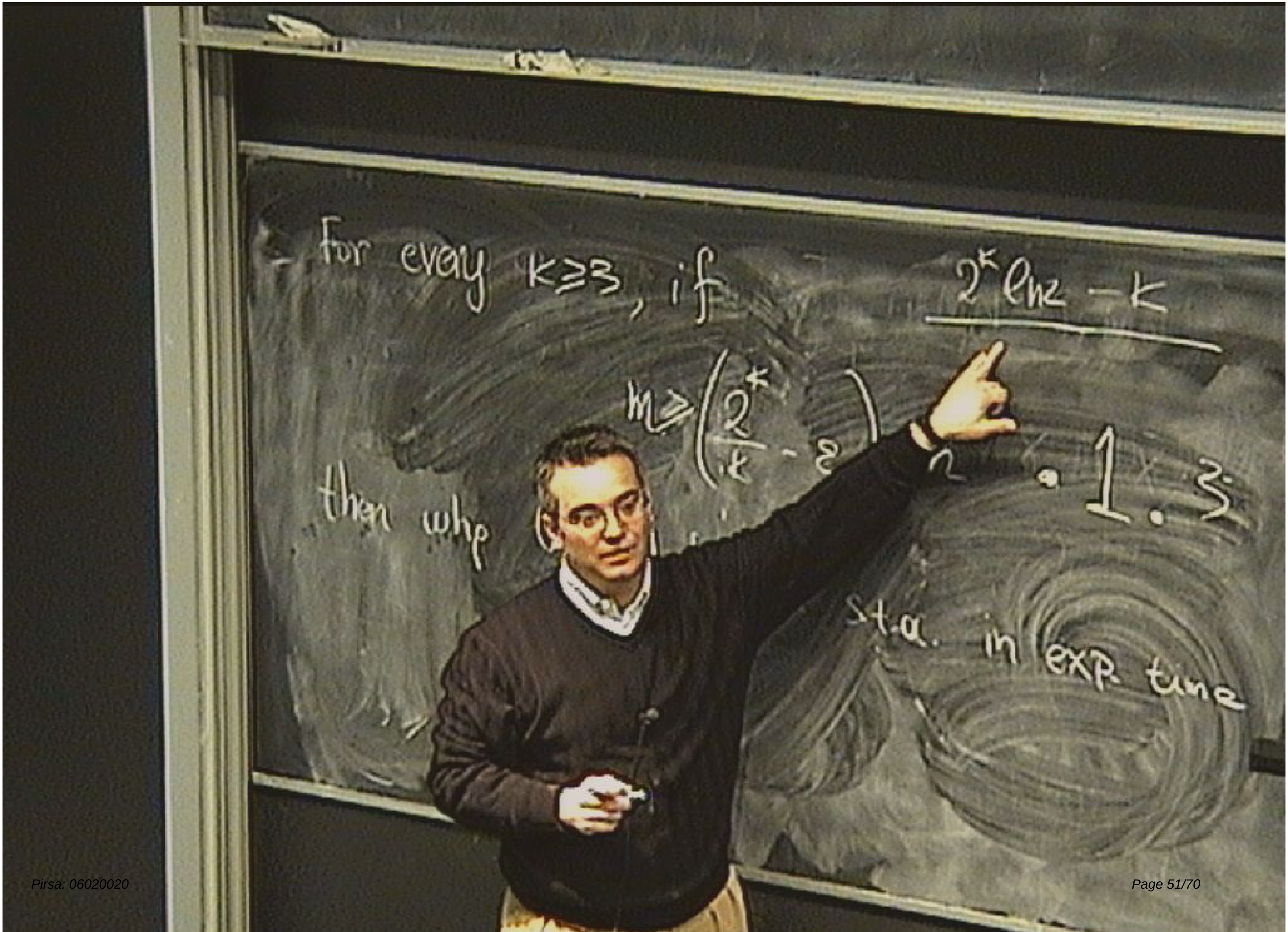

For every k≥3, if
$m = \left(\frac{2^k}{k} - \varepsilon\right) \cdot n$

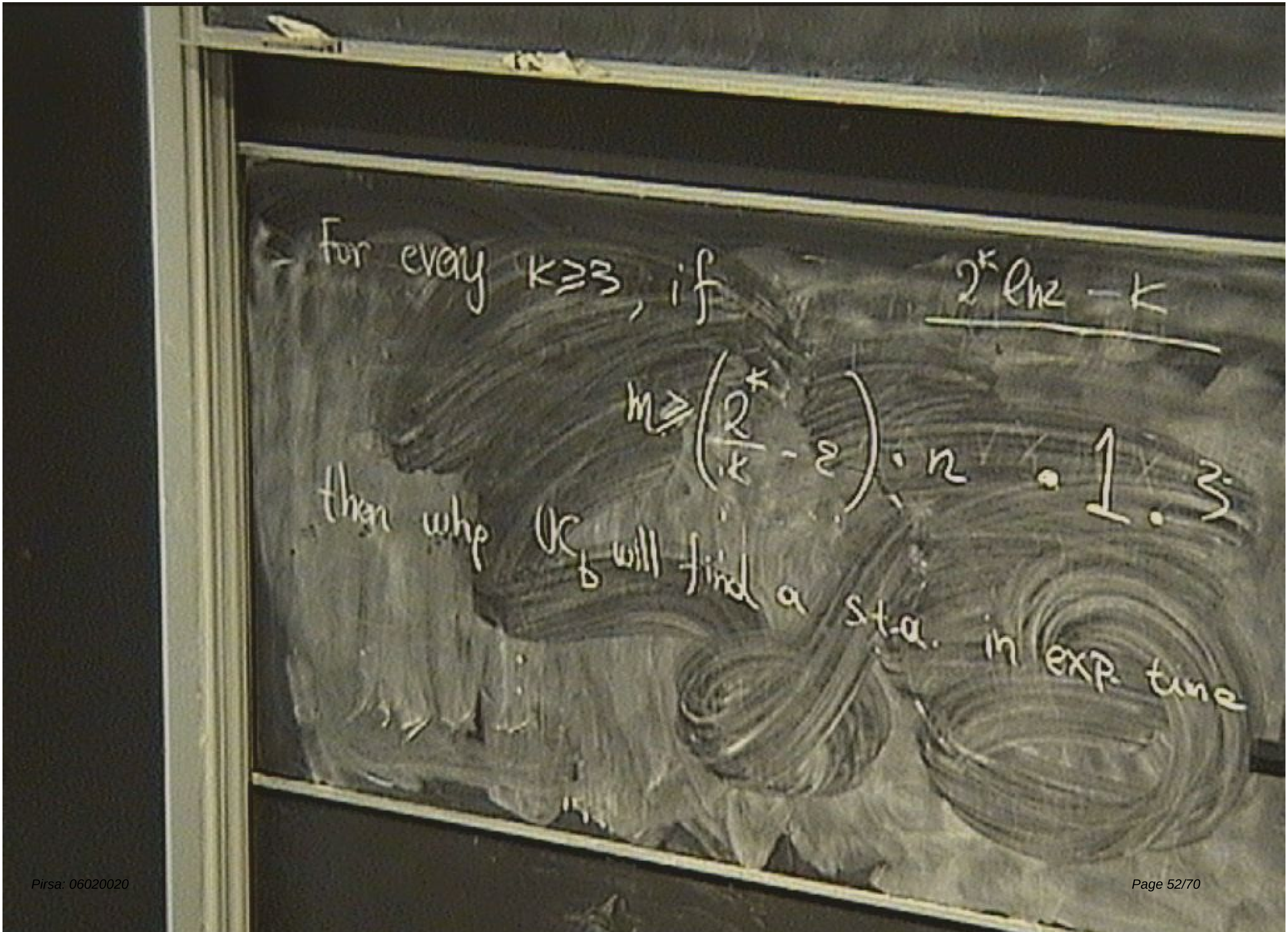

for every k≥3, if
m = (2^k/k − ε) · n
then whe

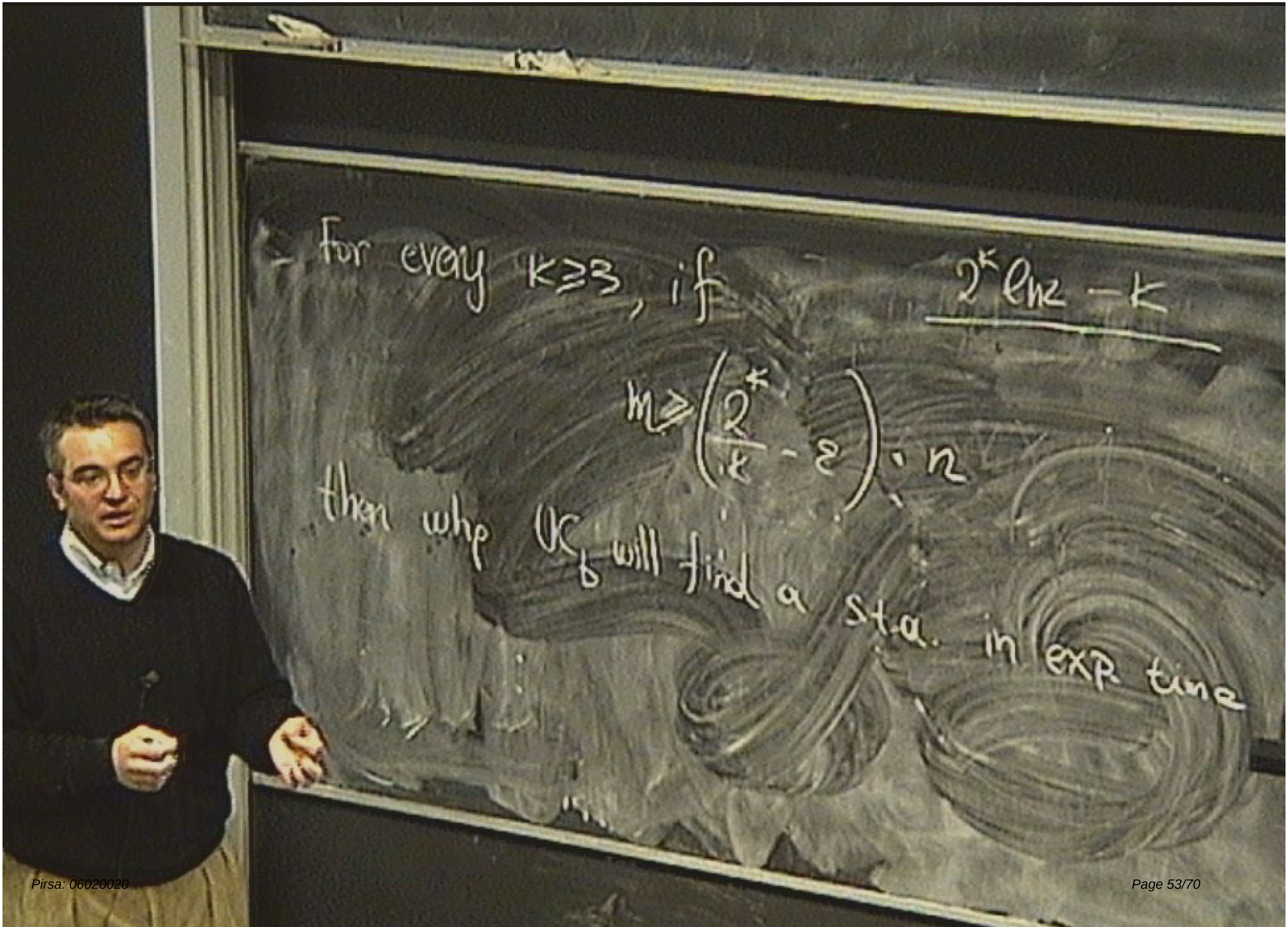

For every k≥3, if
m = (2^k/k - ε) · n
then whp UC will find a s.t.a.


For every k≥3, if
m ≥ (2^k / k − ε) · n · 1.3
then whp OC_b will find a s.t.a. in exp. time

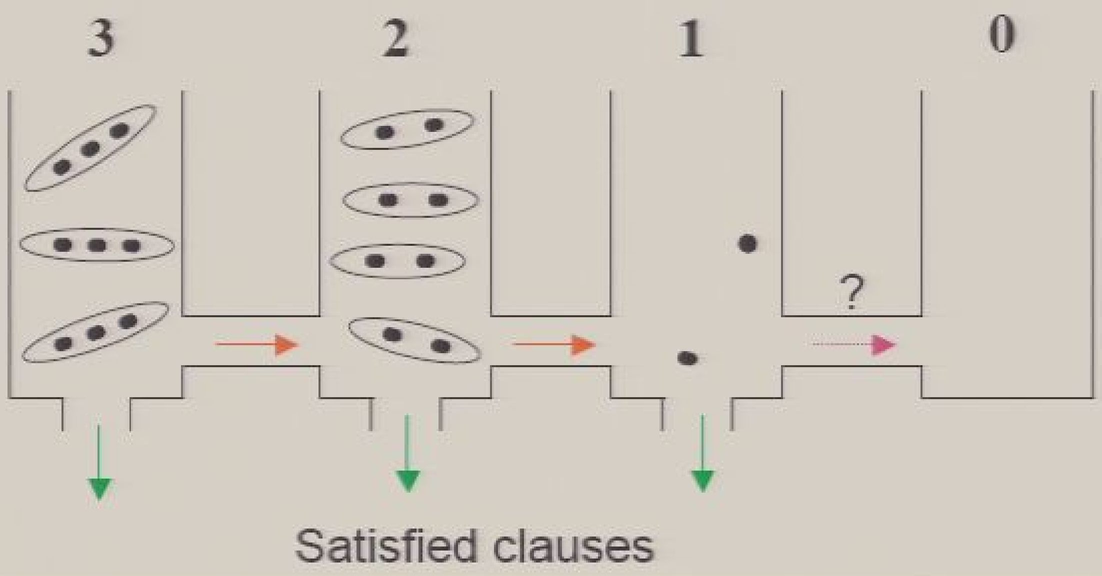

For every k≥3, if
will find a s.t.a. in EXP time


for every k≥3, if
2^k ln2 − k
then whp
S.t.a. in exp time


For every k≥3, if
2^k ln2 − k
m ≥ (2^k/k − ε) · n
then whp OC_b will find a s.t.a. in exp time


For every k≥3, if
2^k ln2 − k
m≥(2^k/k − ε)·n
then whp WC_b will find a s.t.a. in EXP time

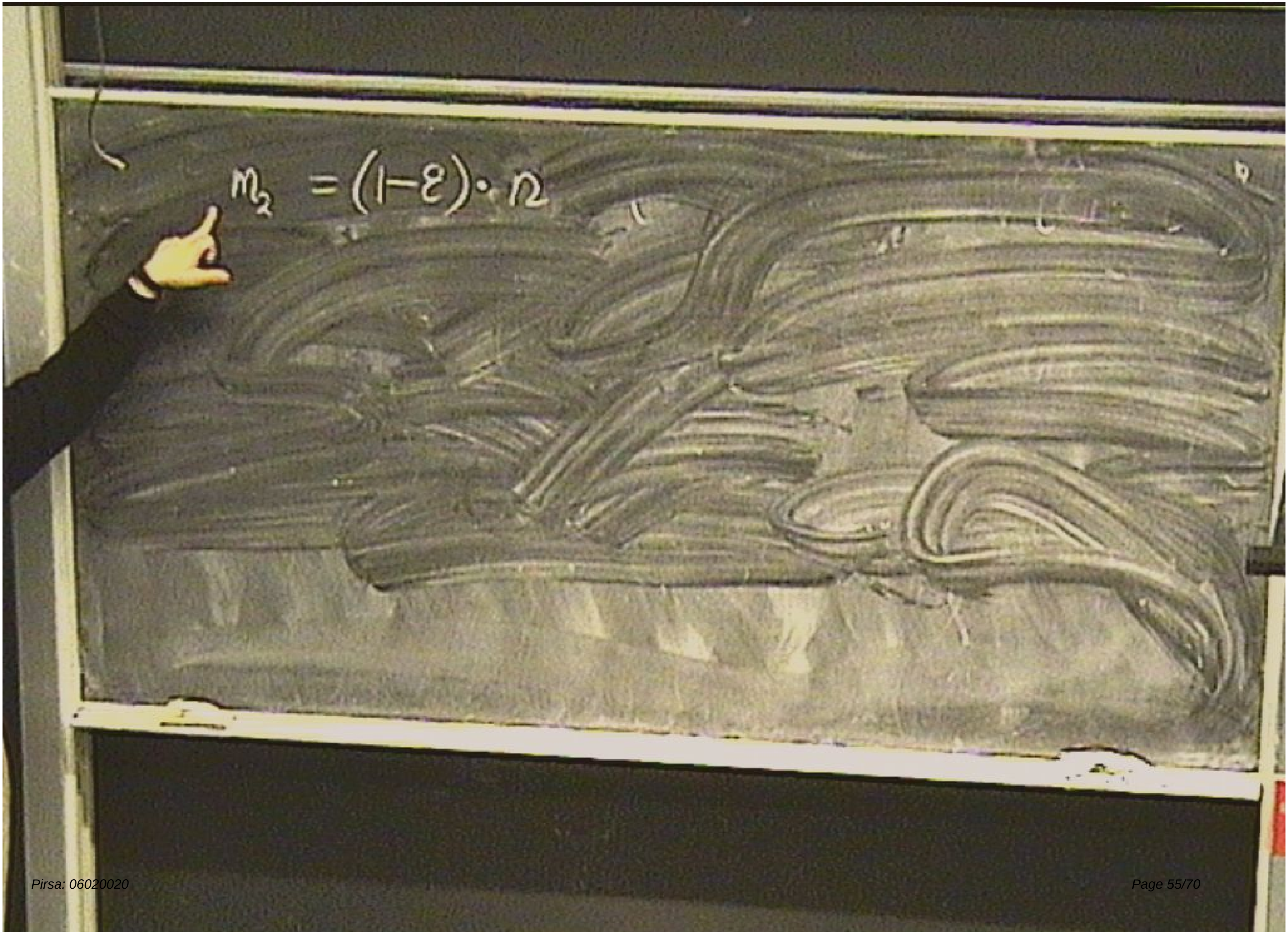
Flows and Buckets



$C_i(t)$ is the number of i-clauses remaining after t variables are set.

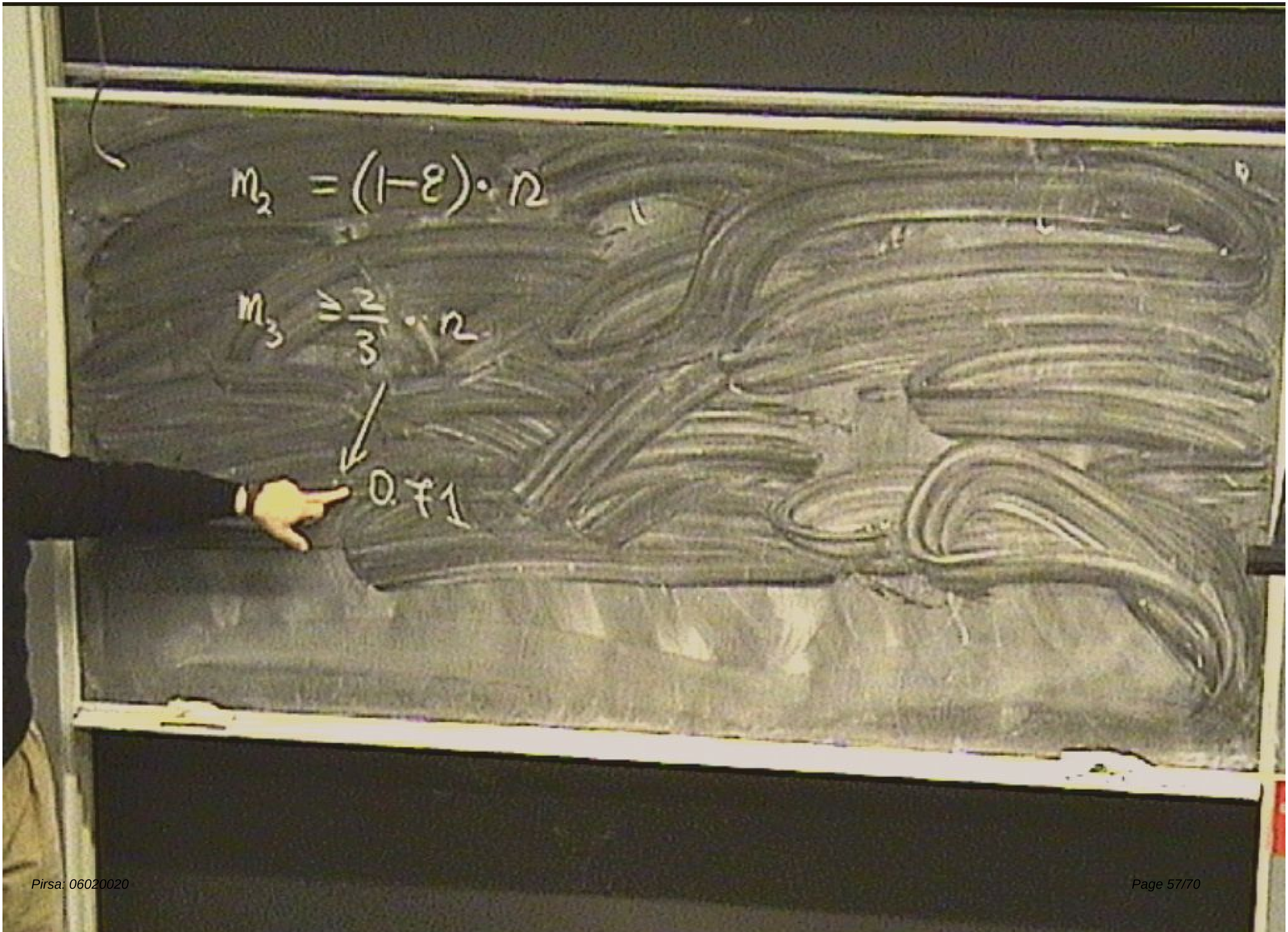
- If for some t, $\dfrac{C_2(t)}{n-t} > (1+\delta)$ the algorithm will a.s. fail.
- The expected number of 1-clauses generated in round t is $\dfrac{C_2(t)}{n-t} + o(1)$.
- If for all t, $\dfrac{C_2(t)}{n-t} < (1-\delta)$ the algorithm succeeds with probability at least $\psi = \psi(\delta) > 0$.

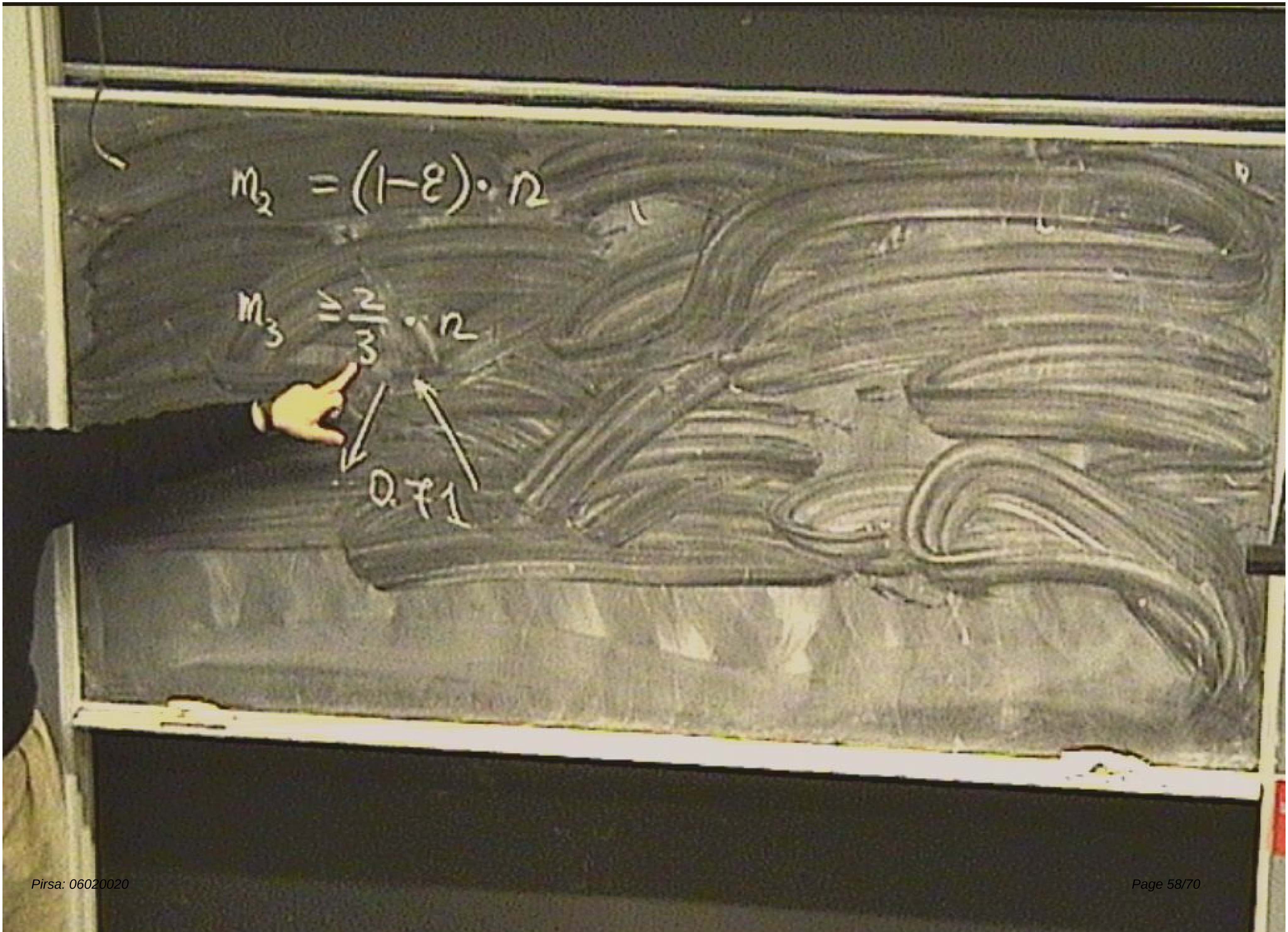
If for some r^* we can show that $\boxed{\text{a.s. } \dfrac{C_2(t)}{n-t} < (1-\delta) \text{ for all } t}$ then $r_3 \geq r^*$.

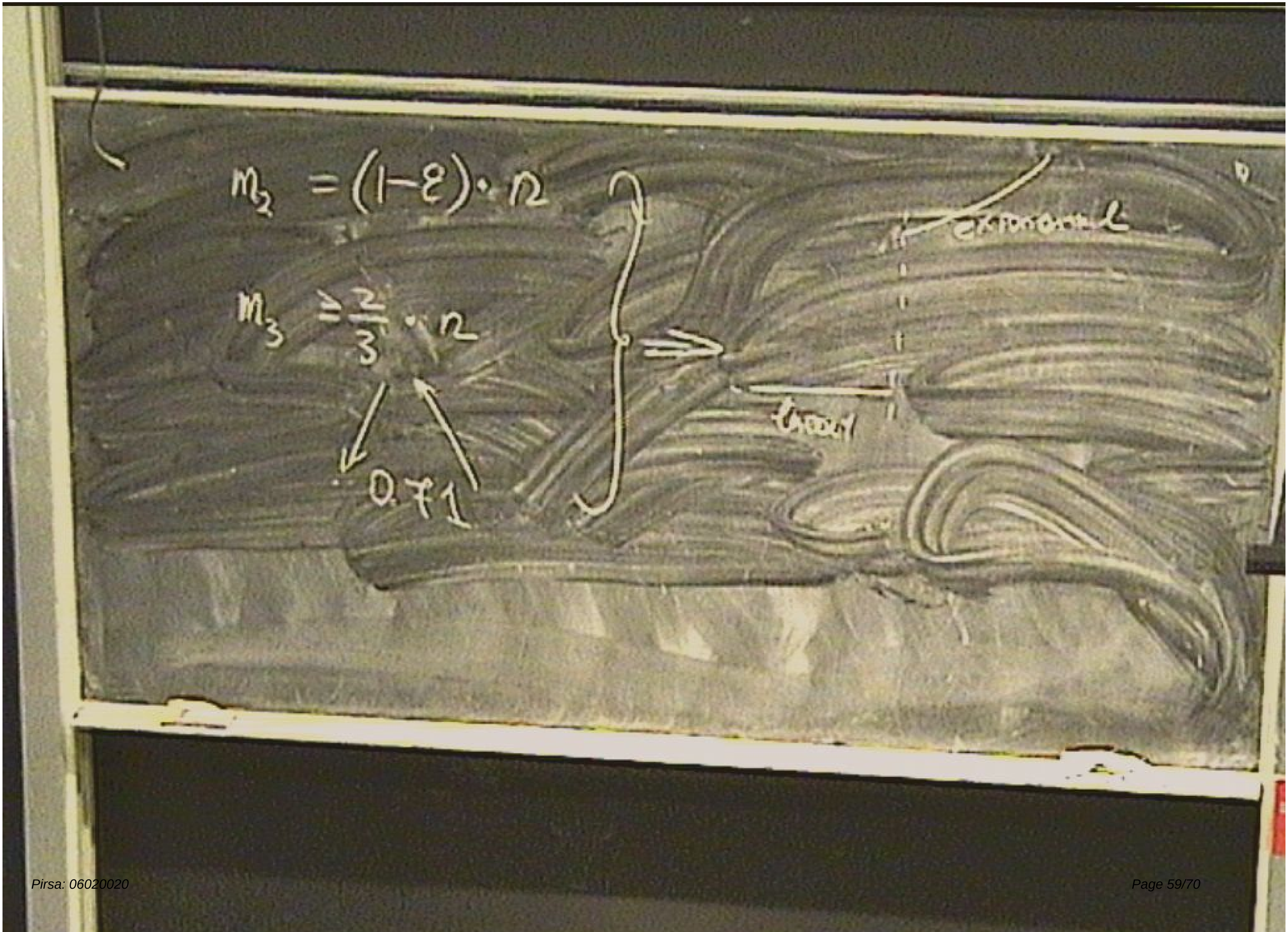

$m_2 = (1-\varepsilon)\cdot n$

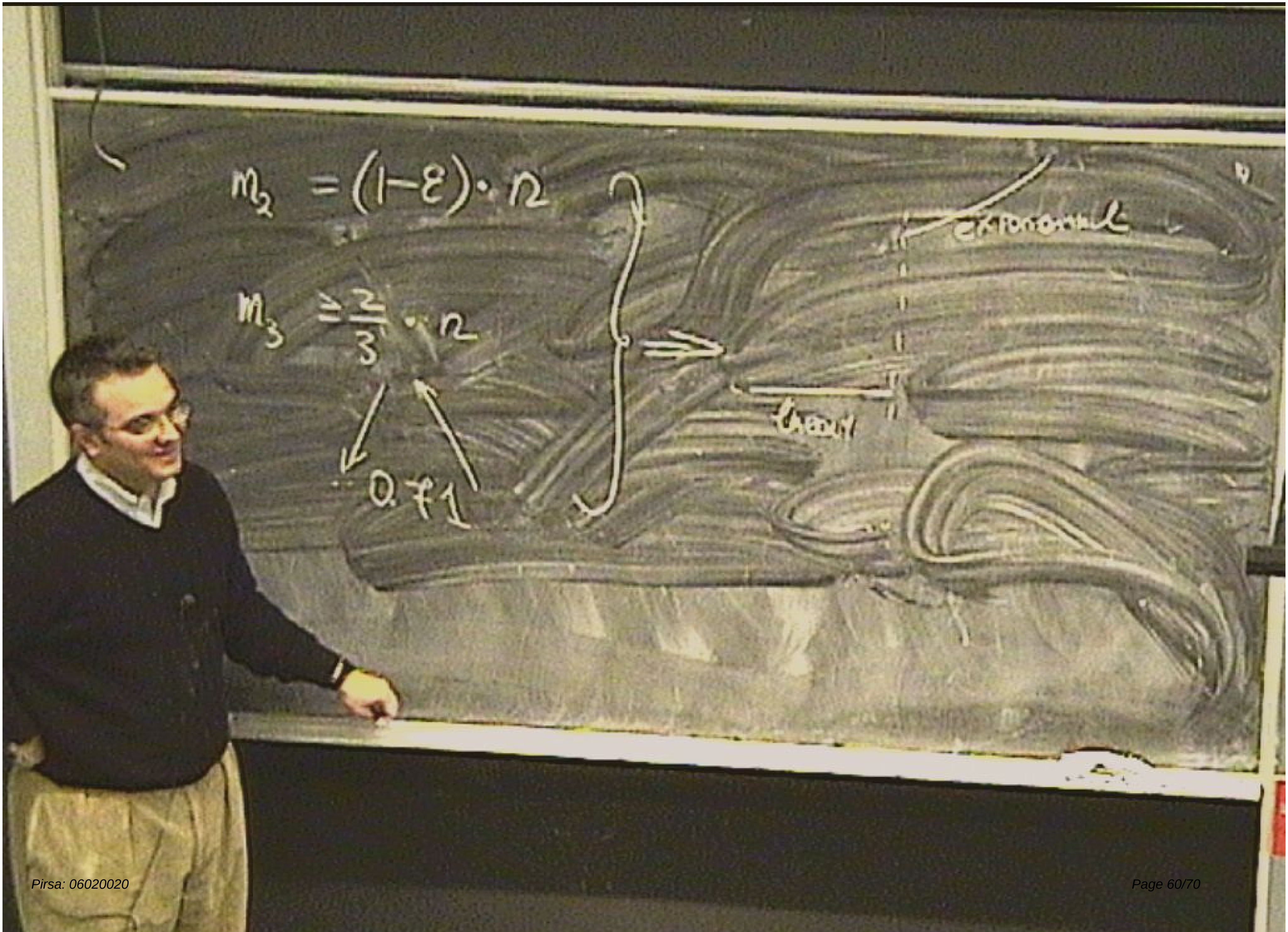
$$m_2 = (1-\varepsilon)\cdot n$$

$$m_3 = \delta \cdot n$$

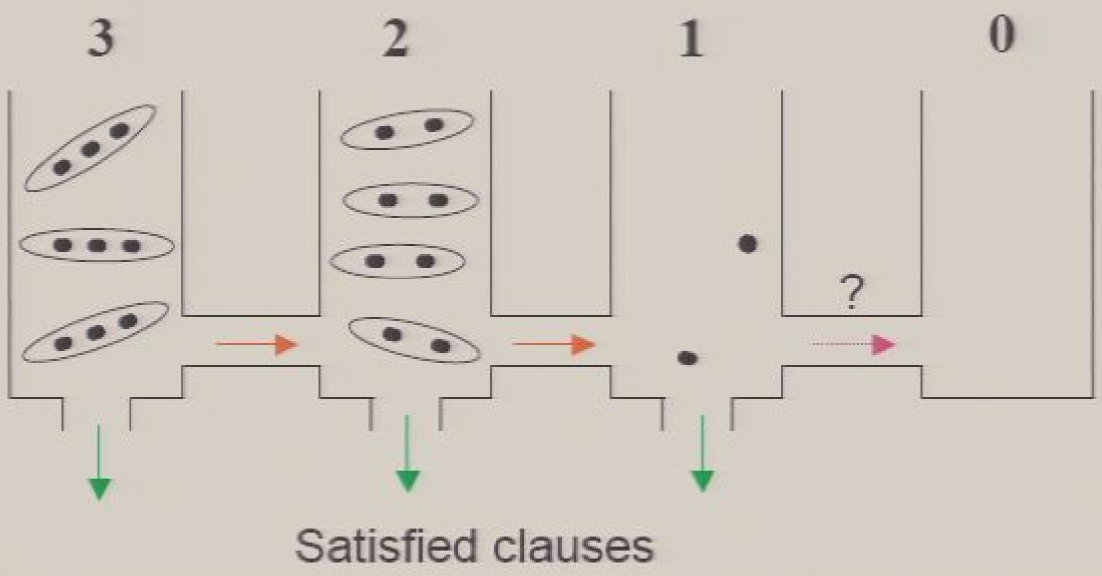

$m_2 = (1-\varepsilon)\cdot n$
$m_3 \geq \frac{2}{3}\cdot n$
0.71


m2 = (1−ε)·n
m3 ≥ 2/3 · n
0.71


$m_2 = (1-\varepsilon) \cdot n$
$m_3 \geq \frac{2}{3} \cdot n$
0.71


$m_2 = (1-\varepsilon)\cdot n$
$m_3 \geq \frac{2}{3}\cdot n$
0.71
exponential

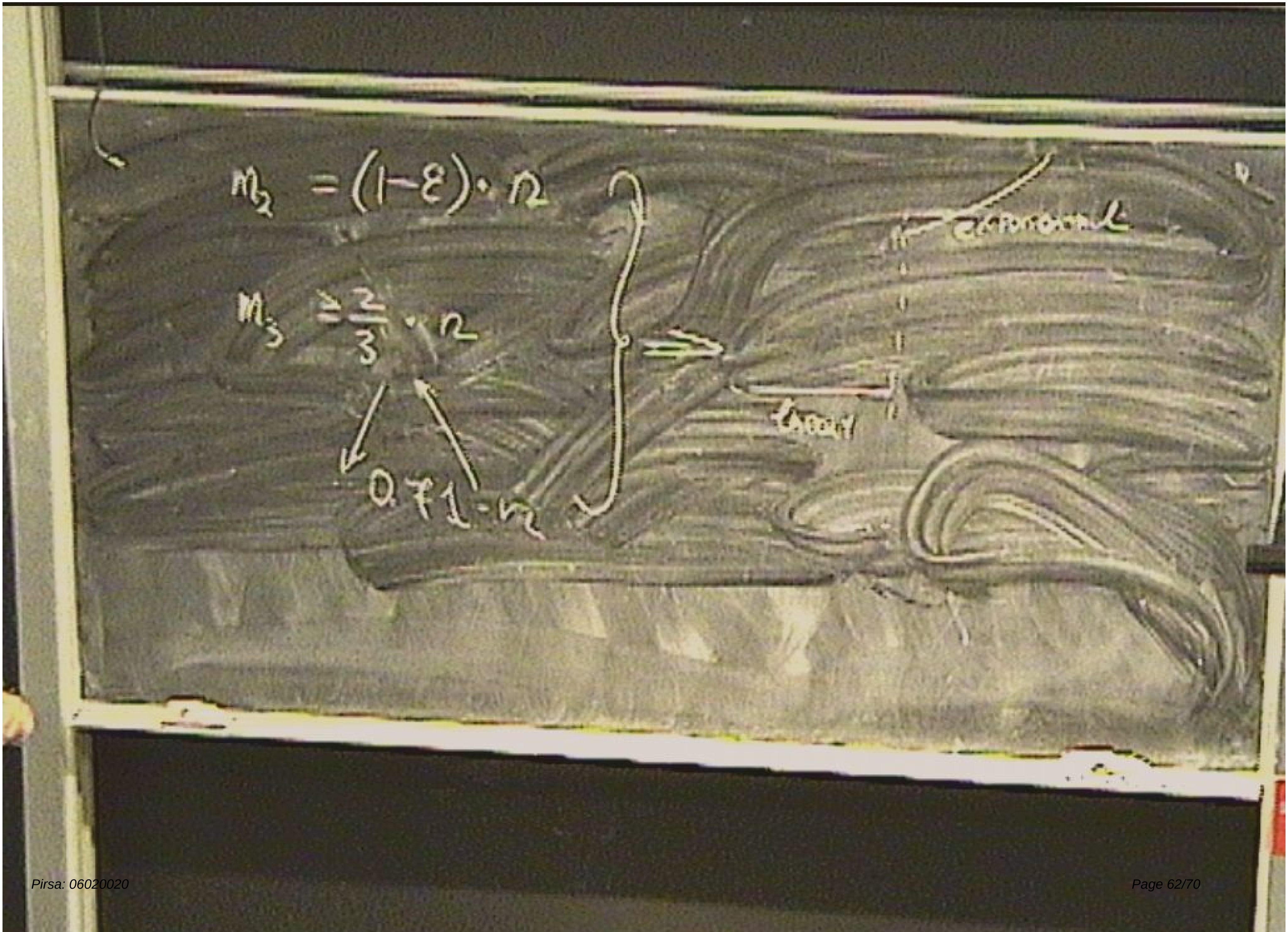
Flows and Buckets



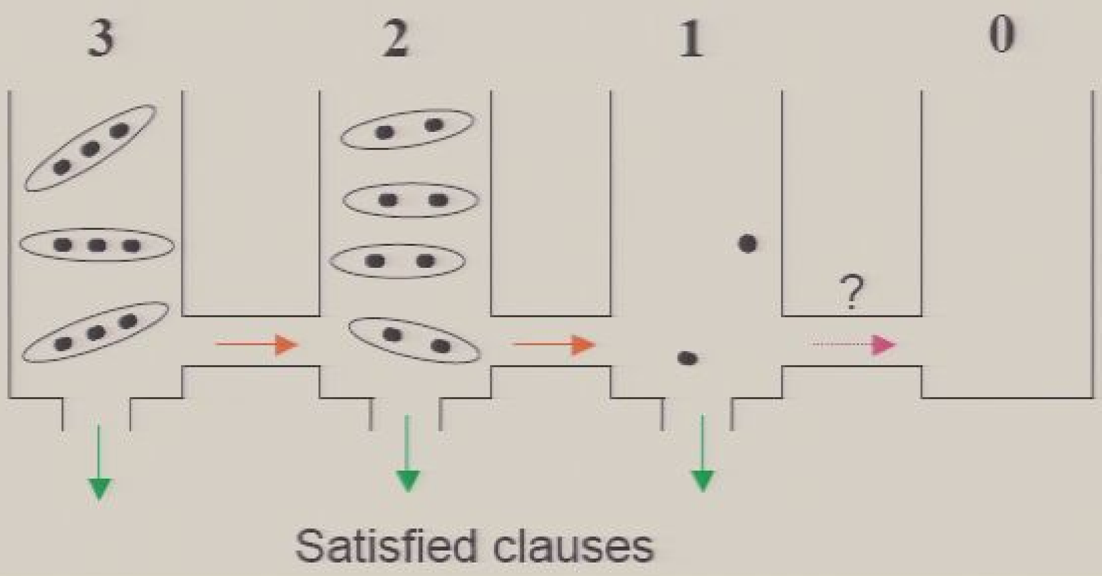
$C_i(t)$ is the number of i-clauses remaining after t variables are set.

- If for some t, $\frac{C_2(t)}{n-t} > (1+\delta)$ the algorithm will a.s. fail.
- The expected number of 1-clauses generated in round t is $\frac{C_2(t)}{n-t} + o(1)$.
- If for all t, $\frac{C_2(t)}{n-t} < (1-\delta)$ the algorithm succeeds with probability at least $\psi = \psi(\delta) > 0$.

If for some r^* we can show that a.s. $\frac{C_2(t)}{n-t} < (1-\delta)$ for all t then $r_3 \geq r^*$.


$m_2 = (1-\varepsilon) \cdot n$
$m_3 \geq \frac{2}{3} \cdot n$
$0.71 \cdot n$

Flows and Buckets



$C_i(t)$ is the number of i-clauses remaining after t variables are set.

- If for some t, $\dfrac{C_2(t)}{n-t} > (1+\delta)$ the algorithm will a.s. fail.
- The expected number of 1-clauses generated in round t is $\dfrac{C_2(t)}{n-t} + o(1)$.
- If for all t, $\dfrac{C_2(t)}{n-t} < (1-\delta)$ the algorithm succeeds with probability at least $\psi = \psi(\delta) > 0$.

If for some r^* we can show that $\boxed{\text{a.s. } \dfrac{C_2(t)}{n-t} < (1-\delta) \text{ for all } t}$ then $r_3 \geq r^*$.

Differential Equations

[Kurtz 78, Karp Sipser 81, Wormald 95]

If we have random variables $Y_1, Y_2, \ldots, Y_k$ evolving jointly such that:

- At each step t,

$$\mathbf{E}\left[\Delta Y_i \mid \mathcal{H}\right] = f_i\left(Y_1/n, \ldots, Y_k/n, t/n\right) + o(1)$$

where the f_i are all Lipschitz continuous.

- The r.v. ΔY_i have reasonable tail behavior.

Then w.h.p. $Y_i(t) = y_i(t) \cdot n + o(n)$ where $y_i(t)$ is the solution of $\dfrac{\mathrm{d}y_i}{\mathrm{d}t} = f_i$.

The evolution is stable under small perturbations of the state.

Differential Equations in action

UC

$$\mathbf{E}(\Delta C_3(t)) = -\frac{3C_3(t)}{n-t} \qquad s_3'(x) = -\frac{3s_3(x)\cdot n}{(1-x)\cdot n} \qquad [x \equiv t/n]$$

$$C_3(0) = rn \qquad s_3(0) = r$$

$$\mathbf{E}(\Delta C_2(t)) = \frac{1}{2} \times \frac{3C_3(t)}{n-t} - \frac{2C_2(t)}{n-t} \qquad s_2'(x) = \frac{3s_3(x)}{2(1-x)} - \frac{2s_2(x)}{1-x}$$

$$C_2(0) = 0 \qquad s_2(0) = 0$$

GUC

$$\mathbf{E}(\Delta C_2(t)) = \frac{3C_3(t)}{2(n-t)} - \frac{2C_2(t)}{n-t} - \left(1 - \frac{C_2(t)}{n-t}\right) \qquad s_2'(x) = \frac{3s_3(x)}{2(1-x)} - \frac{s_2(x)}{(1-x)} - 1$$

$$C_2(0) = 0 \qquad s_2(0) = 0$$

$$\frac{C_2(t)}{n-t} < 1 \iff \frac{s_2(x)\cdot n}{(1-x)\cdot n} < 1 \iff \begin{cases} \frac{3}{2}rx(1-x) < 1 \iff r < 8/3 & \text{UC} \\ \frac{3}{2}rx(1-x/2) + \ln(1-x) < 1 \iff r < 3.003\ldots & \text{GUC} \end{cases}$$

Differential Equations

[Kurtz 78, Karp Sipser 81, Wormald 95]

If we have random variables $Y_1, Y_2, \ldots, Y_k$ evolving jointly such that:

- At each step t,

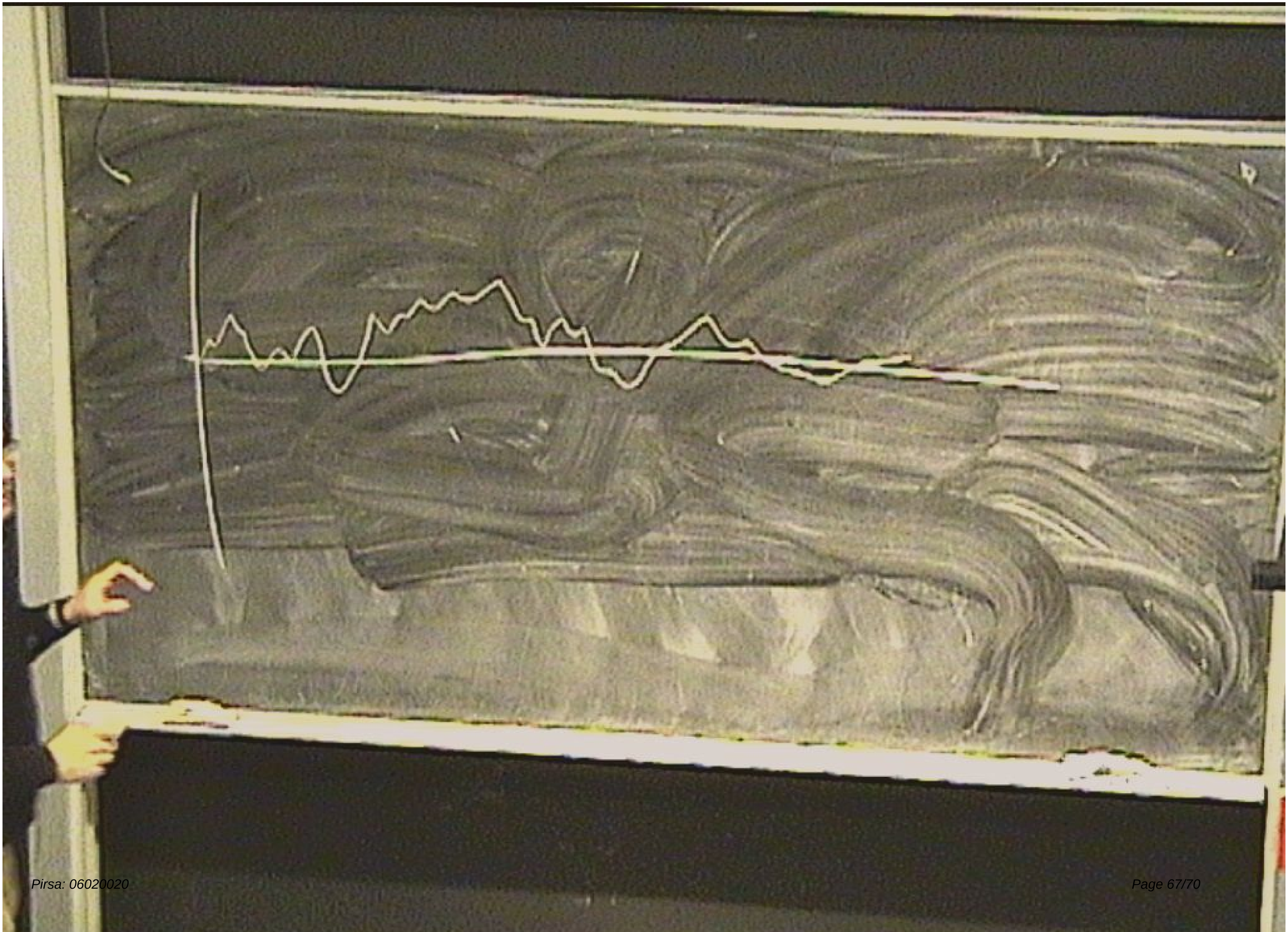
$$\mathbf{E}\left[\Delta Y_i \mid \mathcal{H}\right] = f_i\left(Y_1/n, \ldots, Y_k/n, t/n\right) + o(1)$$

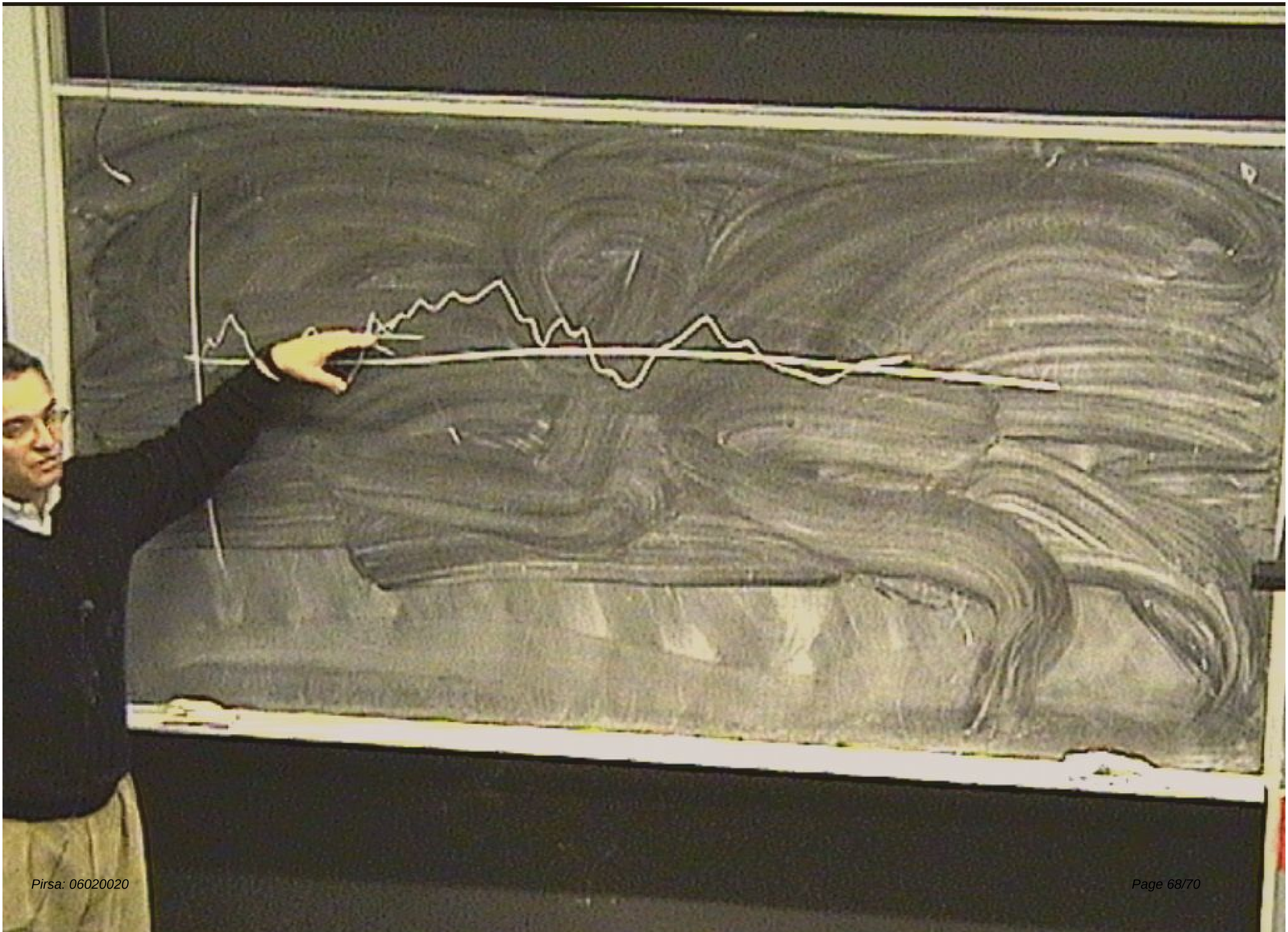
where the f_i are all Lipschitz continuous.

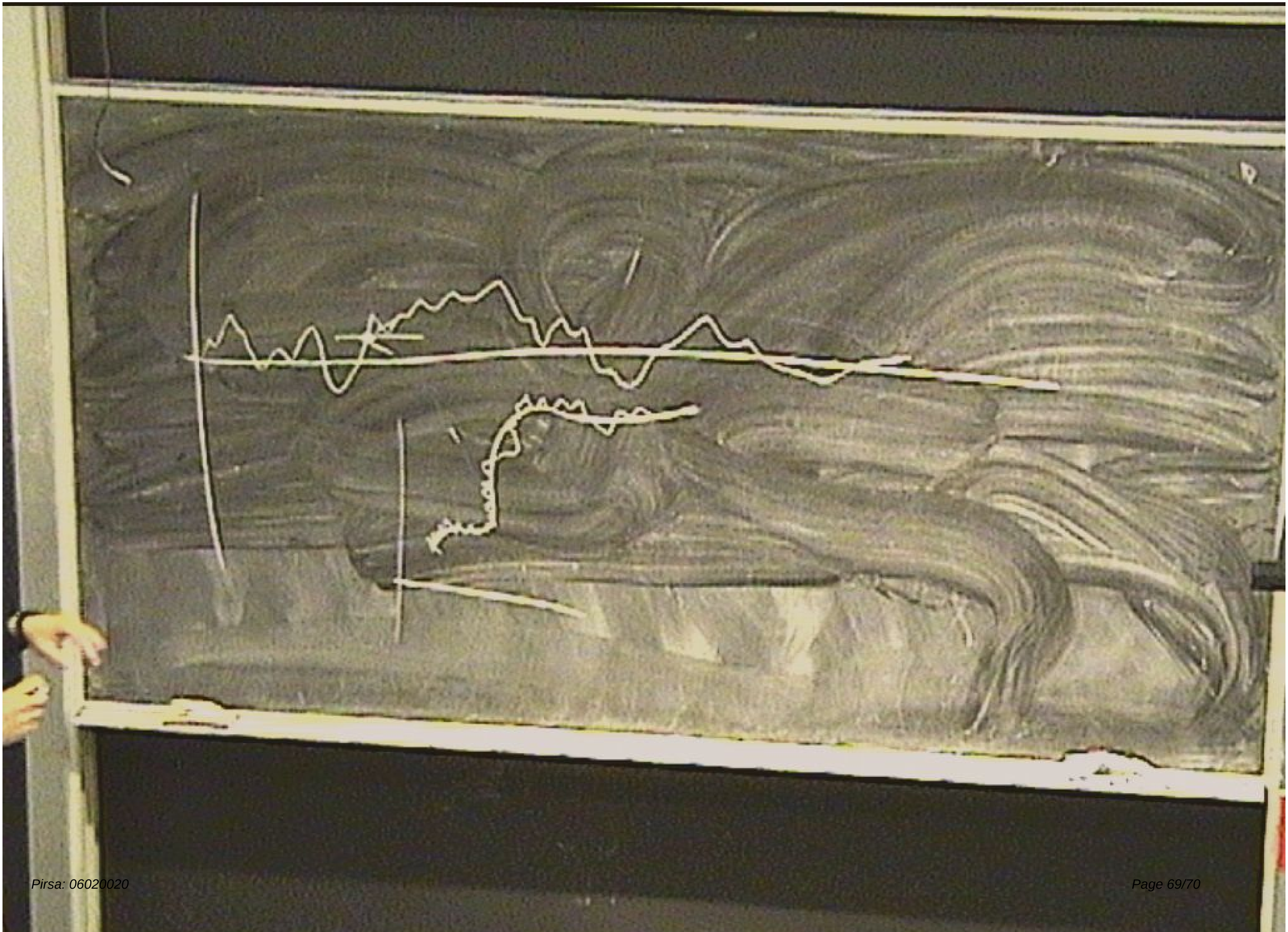
- The r.v. ΔY_i have reasonable tail behavior.

Then w.h.p. $Y_i(t) = y_i(t) \cdot n + o(n)$ where $y_i(t)$ is the solution of $\frac{\mathrm{d}y_i}{\mathrm{d}t} = f_i$.

The evolution is stable under small perturbations of the state.







Getting better algorithms

- Use a model for the analysis that allows explicit access to degree information: formulas are now uniformly random, conditional on their entire degree sequence.

- Dispense with "uniform-randomness" for the 2-clauses. Since 2-SAT is tractable, we can afford a less naive approach for 2-clauses.

General k

UC:	$\frac{2^k}{k}$	[Chao, Franco 85]
SC:	$1.12 \cdot \frac{2^k}{k}$	[Chvátal, Reed 92]
GUC:	$1.87 \cdot \frac{2^k}{k}$	[Frieze, Suen 95]