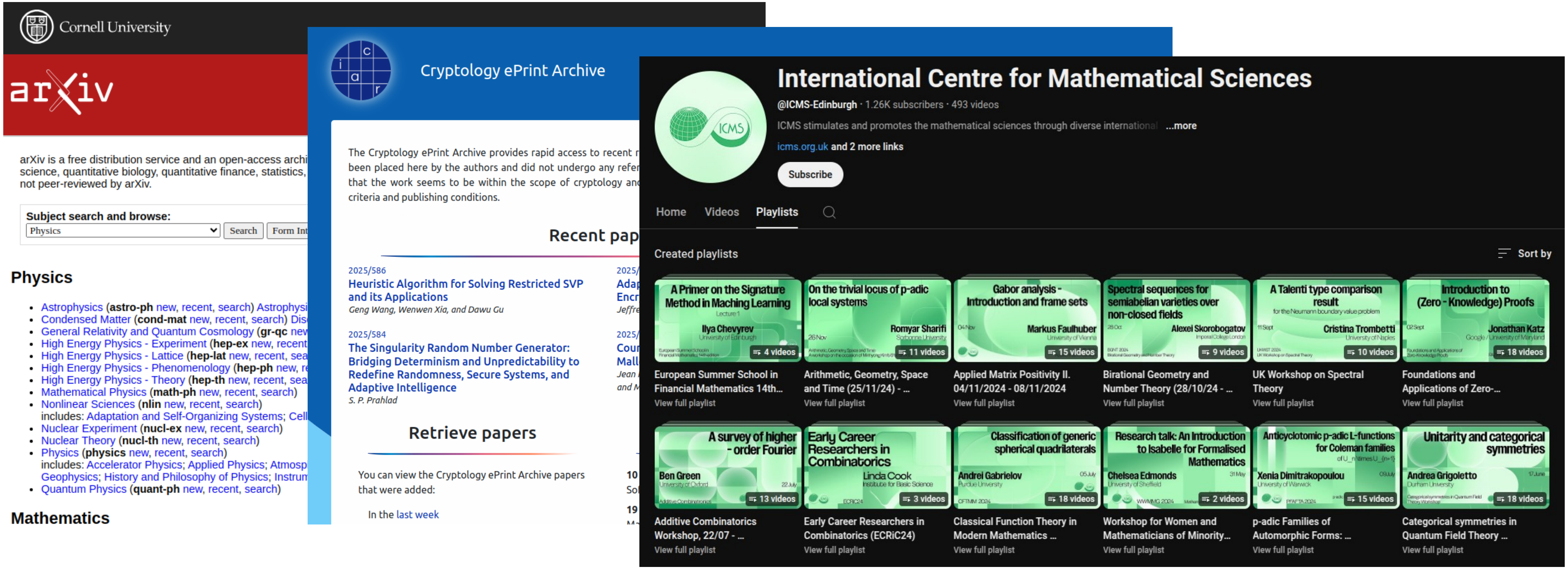
Interactive Proofs and Zero-Knowledge Proofs

Alain Passelègue

April 7, 2025

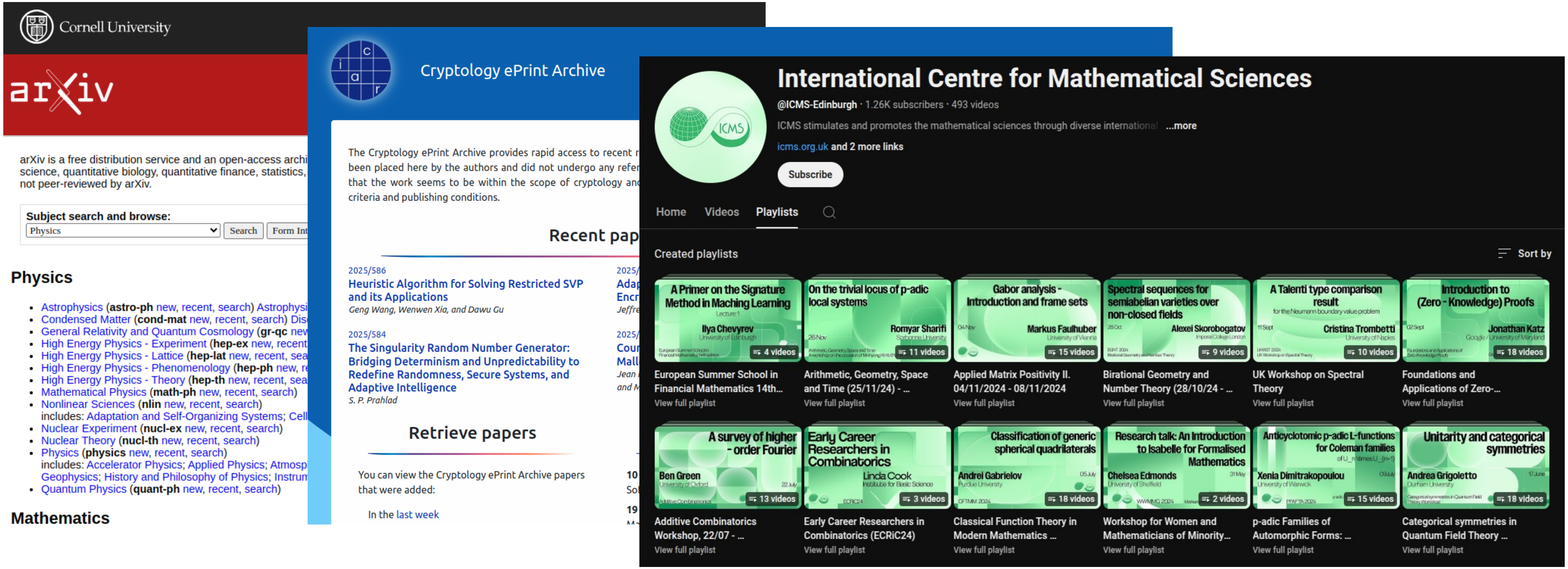
HEAAN CRYPTO LAB

Physical seminars in 2025…?



Why are we still organizing live seminars…?

Physical seminars in 2025...?



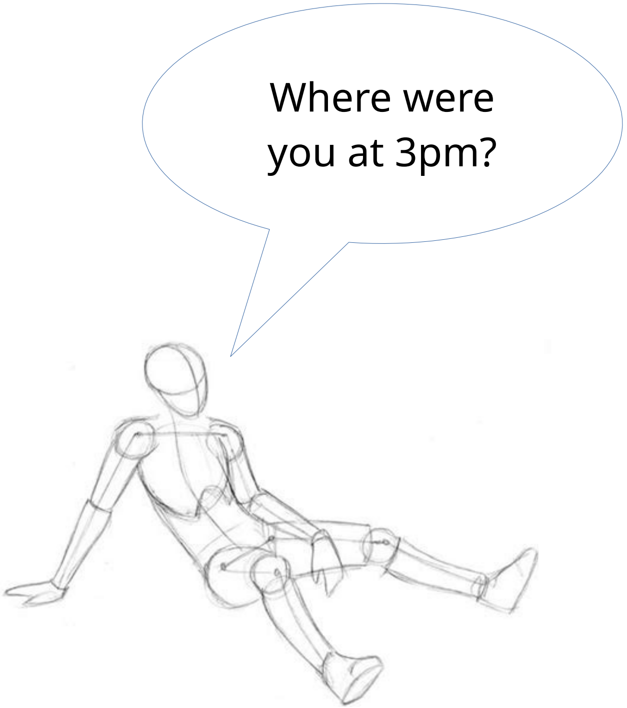
Why are we still organizing live seminars...?

...to ask random questions!

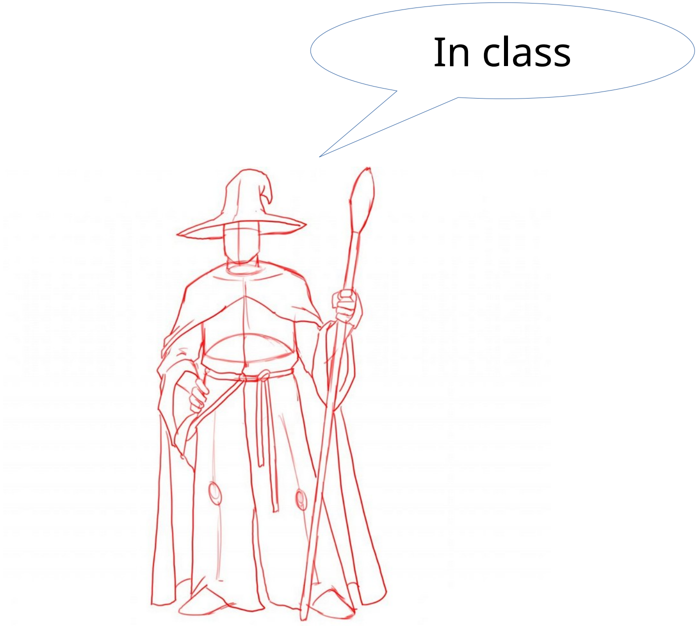
Proofs.

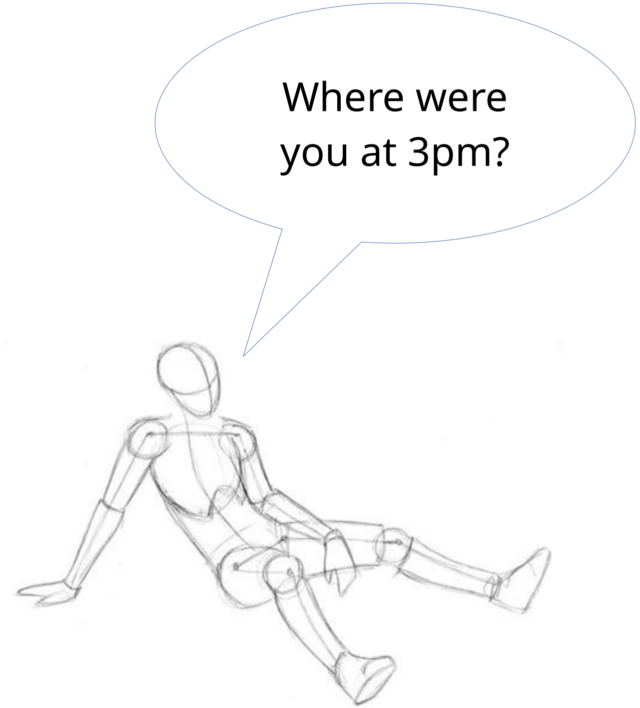
Proofs in real life

Proofs in real life



Proofs in real life





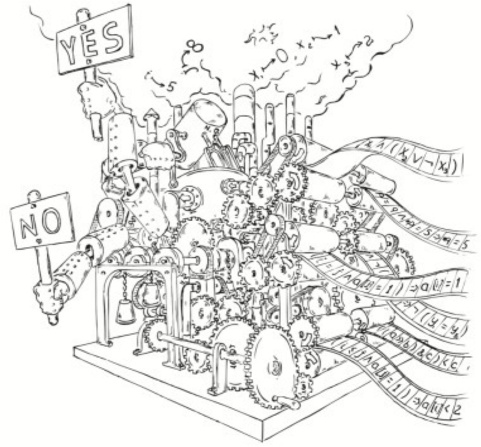
Proofs in real life

In class

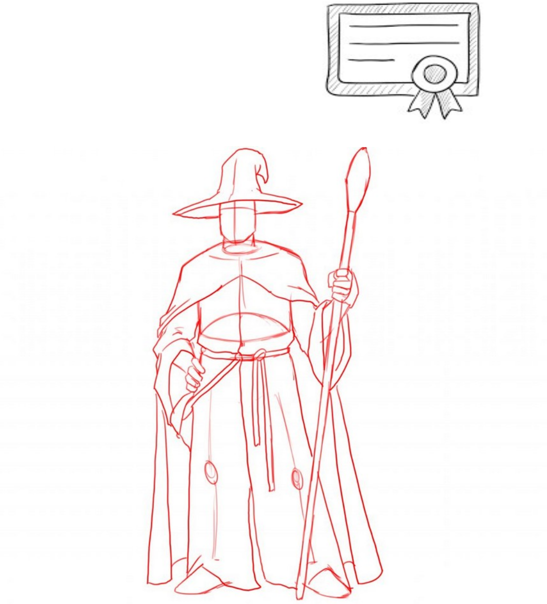
I can bear witness

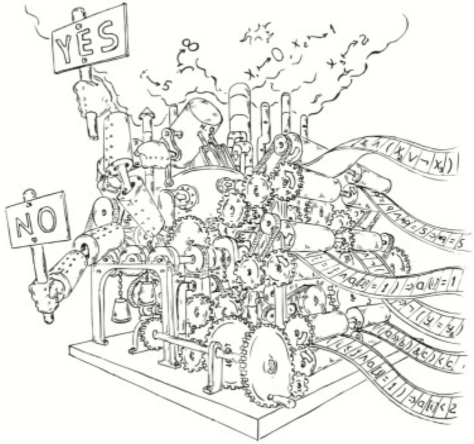
Where were you at 3pm?

Proofs in mathematics

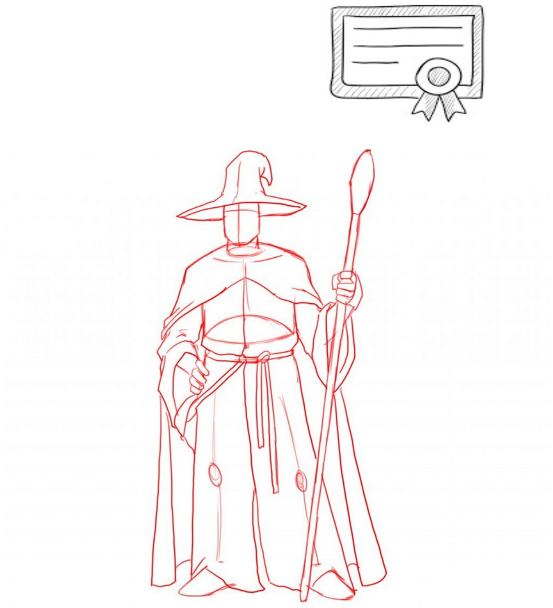


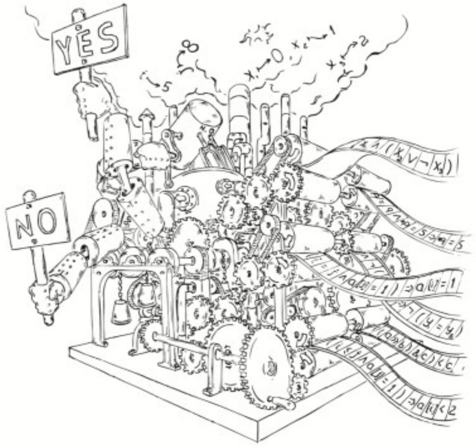
Proofs in mathematics



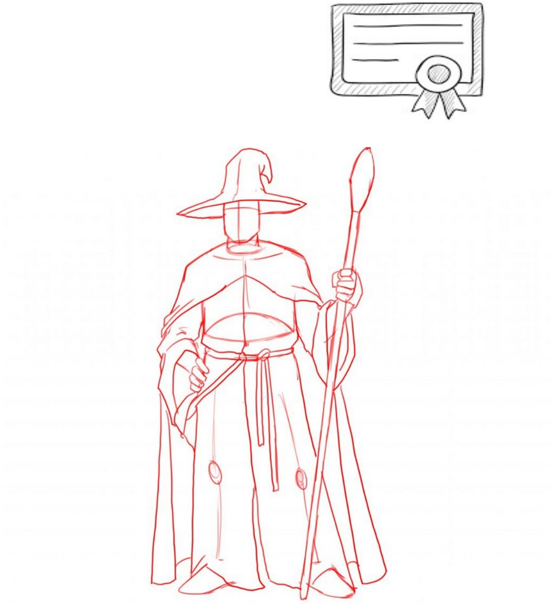


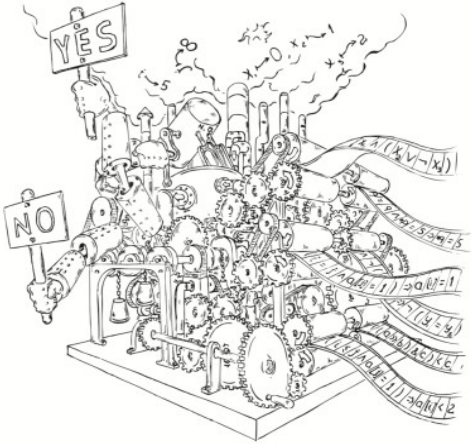
Proofs in mathematics





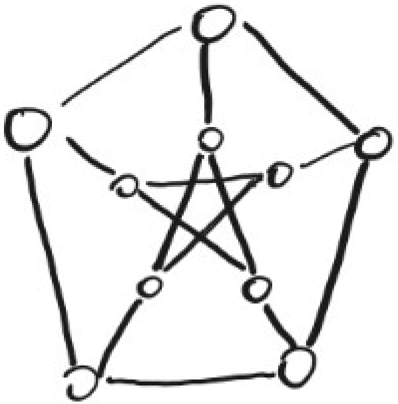
Proofs in mathematics

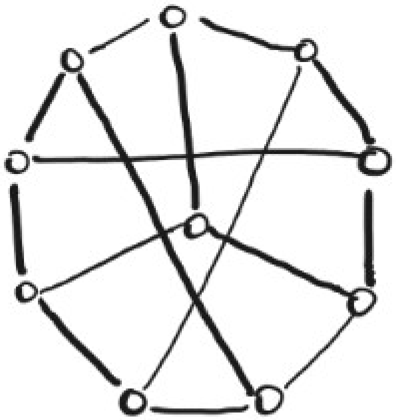




Proof π

Example 1: Graph Isomorphism

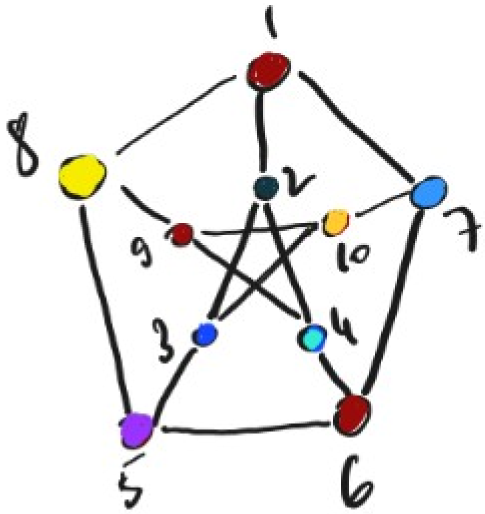


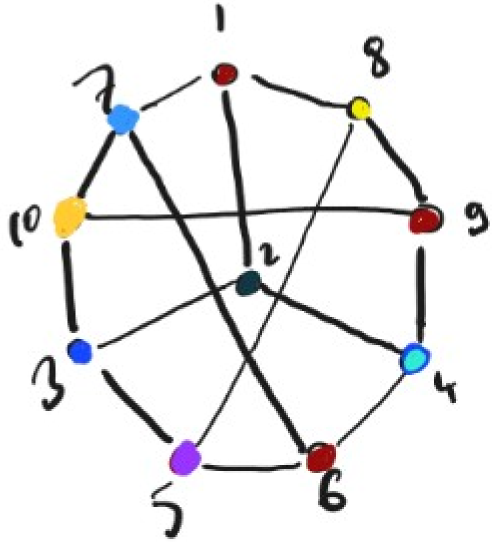


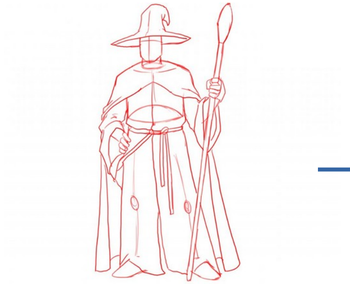
$\pi \in \mathfrak{S}_N$ s.t. $\pi(G_0) = G_1$

Checks $\pi(G_0) = G_1$

Example 1: Graph Isomorphism





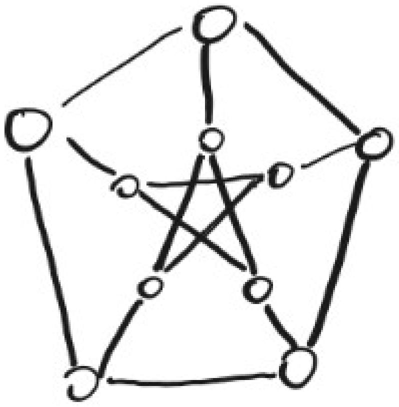


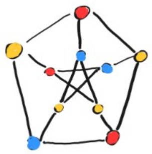
π

$\pi \in \mathfrak{S}_N$ s.t. $\pi(G_0) = G_1$

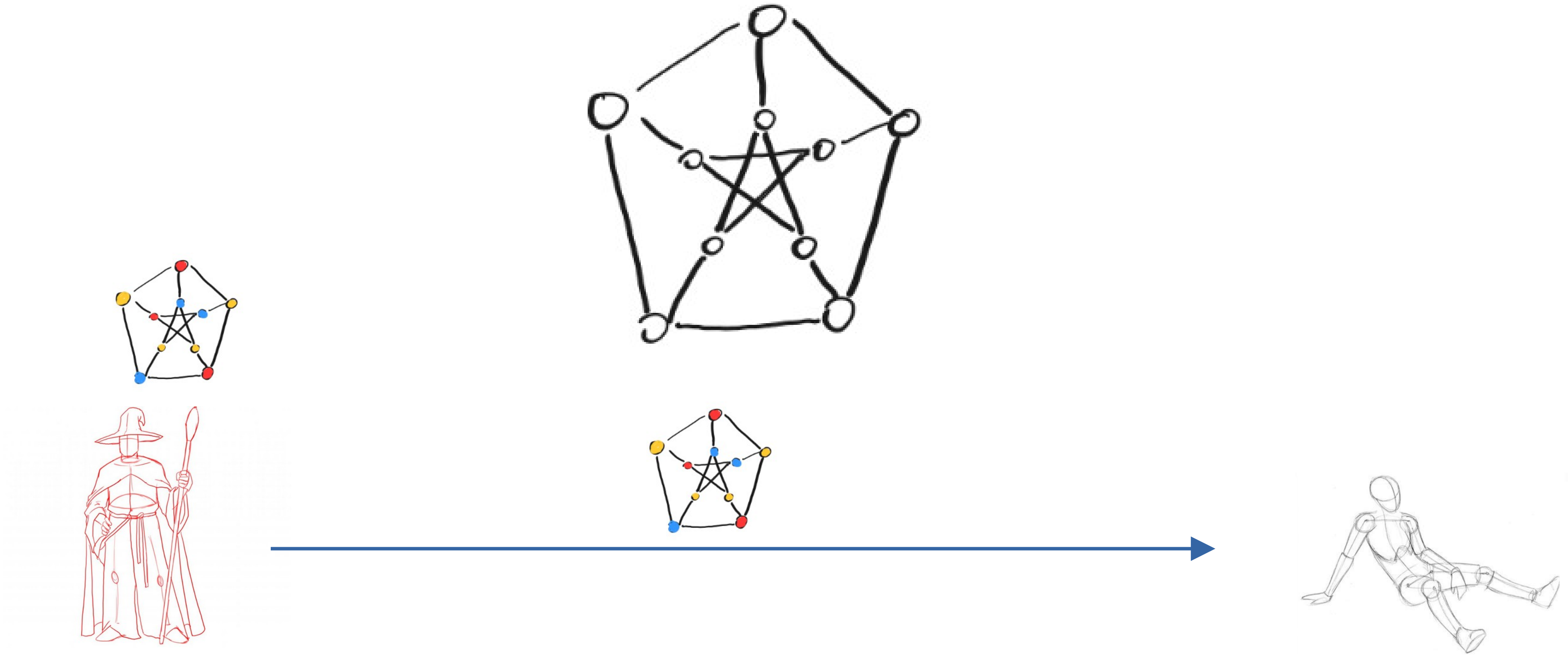
Checks $\pi(G_0) = G_1$

Example 2: 3-coloring





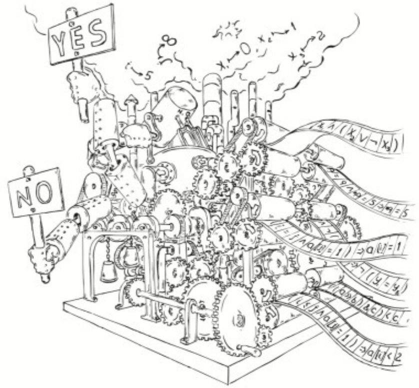
Example 2: 3-coloring



checks the 3-coloring is valid

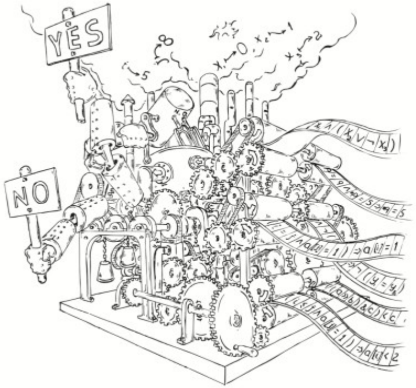
Classical proofs

Language x



Classical proofs

Language x



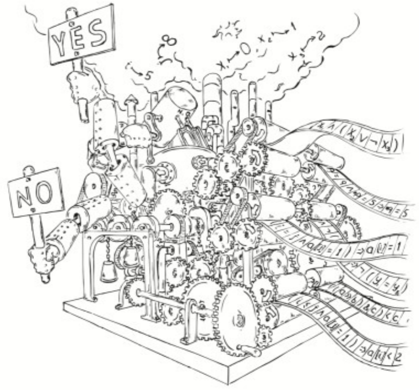
Statement x

Classical proofs

(unbounded)
Prover

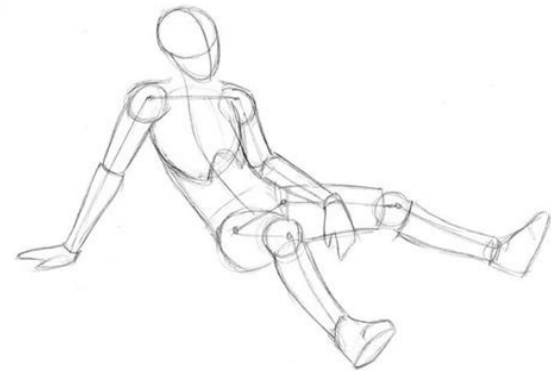
knows a witness w

Language x



Statement x

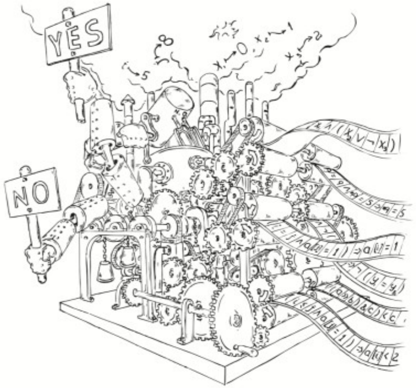
(computationally bounded)
Verifier



Classical proofs

Language x

(unbounded)
Prover



(computationally bounded)
Verifier

Statement x

knows a witness w

Proof π
(short)

able to verify if the proof is valid or not

Formalizing proofs

A language $\mathcal{L} \subseteq \{0,1\}^*$ is efficiently verifiable if there exists a poly-time verifier V such that:

- **Completeness:**
  If $x \in \mathcal{L}$, there exists a witness $w \in \{0,1\}^*$ with $|w| = poly(|x|)$ such that

  $$V(x,w) = 1$$

Formalizing proofs

A language $\mathcal{L} \subseteq \{0,1\}^*$ is efficiently verifiable if there exists a poly-time verifier V such that:

- **Completeness:**
  If $x \in \mathcal{L}$, there exists a witness $w \in \{0,1\}^*$ with $|w| = poly(|x|)$ such that

  $$V(x,w) = 1$$

- **Soundness:**
  If $x \notin \mathcal{L}$, then for all $poly(|x|)$-size witnesses $w \in \{0,1\}^*$, we have:

  $$V(x,w) = 0$$

An alternative definition of NP

A language $\mathcal{L} \subseteq \{0,1\}^*$ is in NP if there exists a poly-time verifier V such that:

- **Completeness:**
  If $x \in \mathcal{L}$, there exists a witness $w \in \{0,1\}^*$ with $|w| = poly(|x|)$ such that

  $$V(x,w) = 1$$

- **Soundness:**
  If $x \notin \mathcal{L}$, then for all $poly(|x|)$-size witnesses $w \in \{0,1\}^*$, we have:

  $$V(x,w) = 0$$

An alternative definition of NP

A language $\mathcal{L} \subseteq \{0,1\}^*$ is in NP if there exists a poly-time verifier V such that:

- **Completeness:**
  If $x \in \mathcal{L}$, there exists a witness $w \in \{0,1\}^*$ with $|w| = poly(|x|)$ such that

$$V(x,w) = 1$$

- **Soundness:**

V = the poly-time NDTM
w = choices such that V(x) = 1

Are we stuck with NP?

Convince me of something I cannot check

What made it possible?

- Interaction:

  the verifier and the prover interacts in a series of questions/responses

- Randomness:

  questions cannot be predicted by the prover:

  - for $x \in \mathcal{L}$, it can always find the good answer
  - for $x \notin \mathcal{L}$, it fails with some probability

What made it possible?

- Interaction:

  the verifier and the prover interacts in a series of questions/responses

- Randomness:

  questions cannot be predicted by the prover:

  → for $x \in \mathcal{L}$, it can always find the good answer

  → for $x \notin \mathcal{L}$, it fails with some probability

Both are required!

What made it possible?

- Interaction:
  the verifier and the prover interacts in a series of questions/responses

- Randomness:
  questions cannot be predicted by the prover:
  - for $x \in \mathcal{L}$, it can always find the good answer
  - for $x \notin \mathcal{L}$, it fails with some probability

$\Rightarrow$ The verifier can only be convinced up to some (possibly very large) probability

Interactive proofs.

Interactive proofs

[Goldwasser-Micali-Rackhoff'86]

(unbounded)
Prover

knows a witness w

Statement x

a_1 →

← q_1

a_2 →

← q_2

$\vdots$

a_L →

← q_L

Proof π

(computationally bounded)
Verifier

↓

accept/reject

A formal definition of IP

A language $\mathcal{L} \subseteq \{0,1\}^*$ admits an interactive proof system if there exists an unbounded prover P and a probabilistic poly-time verifier V such that:

- **<u>Completeness:</u>**
  If $x \in \mathcal{L}$, then
  $$Pr[\langle P, V\rangle(x) = 1] \geq 2/3$$
- **<u>Soundness:</u>**
  If $x \notin \mathcal{L}$, then
  $$Pr[\langle P, V\rangle(x) = 1] \leq 1/3$$

A formal definition of IP

A language $\mathcal{L} \subseteq \{0,1\}^*$ admits an interactive proof system if there exists an unbounded prover P and a probabilistic poly-time verifier V such that:

- **Completeness:**
  If $x \in \mathcal{L}$, then
  $$Pr[\langle P, V \rangle(x) = 1] \geq 2/3$$
- **Soundness:**
  If ...

One can amplify the bounds by iterating the process... This exponentially converges

A formal definition of IP

A language $\mathcal{L} \subseteq \{0,1\}^*$ admits an interactive proof system if there exists an unbounded prover P and a probabilistic poly-time verifier V such that:

- **<u>Completeness:</u>**
  If $x \in \mathcal{L}$, then
  $$Pr[\langle P, V \rangle(x) = 1] \geq 1 - 2^{-n}$$
- **<u>Soundness:</u>**
  If $x \notin \mathcal{L}$, then
  $$Pr[\langle P, V \rangle(x) = 1] \leq 2^{-n}$$

A formal definition of IP

A language $\mathcal{L} \subseteq \{0,1\}^*$ admits an interactive proof system if there exists an unbounded prover P and a probabilistic poly-time verifier V such that:

- **Completeness:**
  If $x \in \mathcal{L}$, then
  $$Pr[\langle P, V \rangle(x) = 1] \geq 1 - 2^{-n}$$
- **Soundness:**
  If $x \notin \mathcal{L}$, then
  $$Pr[\langle P, V \rangle(x) = 1] \leq 2^{-n}$$

A formal definition of IP

A language $\mathcal{L} \subseteq \{0,1\}^*$ admits an interactive proof system if there exists an unbounded prover P and a probabilistic poly-time verifier V such that:

- **Completeness:**
  If $x \in \mathcal{L}$, then

$$Pr[\langle P, V\rangle(x) = 1] \geq 1 - 2^{-n}$$

- **Soundness:**
  If $x \notin \mathcal{L}$, then

IP = languages that admit an interative proof system

A formal definition of IP

A language $\mathcal{L} \subseteq \{0,1\}^*$ admits an interactive proof system if there exists an unbounded prover P and a probabilistic poly-time verifier V such that:

If we want perfect soundness, we are stuck with classical (NP) proofs

- **Soundness:**
  If $x \notin \mathcal{L}$, then

$$Pr[\langle P, V \rangle(x) = 1] \leq 2^{-n}$$

Actual definition of IP

A language $\mathcal{L} \subseteq \{0,1\}^*$ admits an interactive proof system if there exists an unbounded prover P and a probabilistic poly-time verifier V such that:

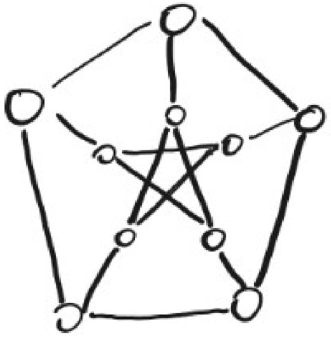
- **Completeness:**
  If $x \in \mathcal{L}$, then
  $$Pr[\langle P, V\rangle(x) = 1] \geq 1 - 2^{-n}$$
- **Soundness:**
  If $x \notin \mathcal{L}$, **then for any unbounded prover P^***
  $$Pr[\langle P^*, V\rangle(x) = 1] \leq 2^{-n}$$

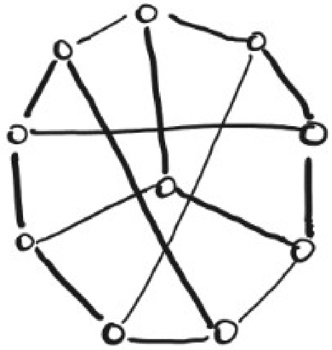
Benefits of interactive proofs

Interactive proofs can offer:

- **Simpler verification**
- Proofs for languages **beyond NP**
- Additional properties, such as **zero-knowledge**

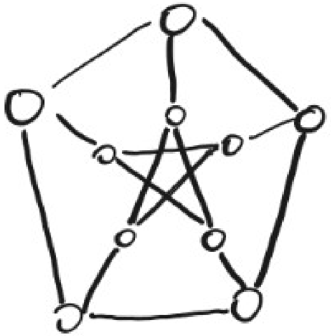
Back to example 1: Graph Isomorphism

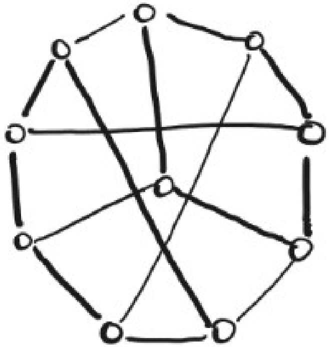


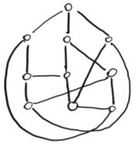


$$\pi \in \mathfrak{S}_N \text{ s.t. } \pi(G_0) = G_1$$

Back to example 1: Graph Isomorphism



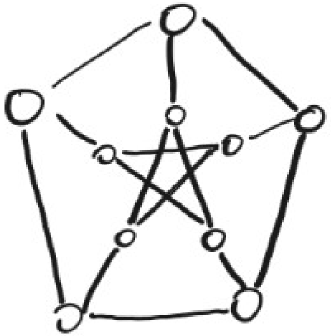


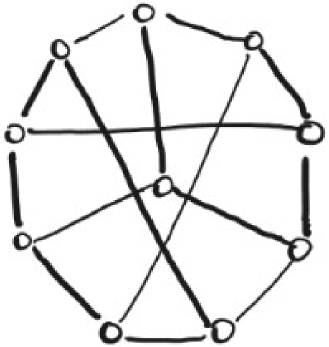


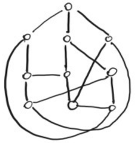
Pick a secret $\sigma \in \mathfrak{S}_N$, reveal $H = \sigma(G_0)$

$\pi \in \mathfrak{S}_N$ s.t. $\pi(G_0) = G_1$

Back to example 1: Graph Isomorphism





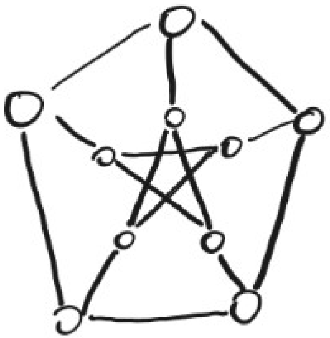


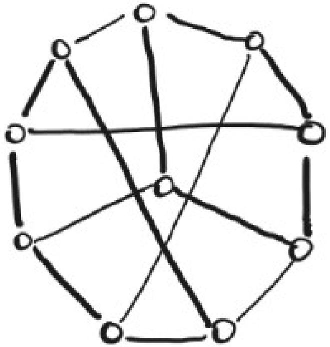
Pick a secret $\sigma \in \mathfrak{S}_N$, reveal $H = \sigma(G_0)$

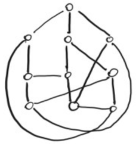
Pick $b \leftarrow U(\{0,1\})$, request mapping from G_b to H

$\pi \in \mathfrak{S}_N$ s.t. $\pi(G_0) = G_1$

Back to example 1: Graph Isomorphism







Pick a secret $\sigma \in \mathfrak{S}_N$, reveal $H = \sigma(G_0)$

Pick $b \leftarrow U(\{0,1\})$, request mapping from G_b to H

Reveal ψ, where $\psi = \sigma$ if $b = 0$, else $\sigma \circ \pi^{-1}$

$\pi \in \mathfrak{S}_N$ s.t. $\pi(G_0) = G_1$

Checks $\psi(G_b) = H$

Back to example 1: Graph Isomorphism

- **Completeness:**
  If $x \in \mathcal{L}$, then

$$Pr[\langle P, V\rangle(x) = 1] = 1$$

Pick a secret $\sigma \in \mathfrak{S}_N$, reveal $H = \sigma(G_0)$ →

← Pick $b \leftarrow U(\{0,1\})$, request mapping from G_b to H

Reveal ψ, where $\psi = \sigma$ if $b = 0$, else $\sigma \circ \pi^{-1}$ →

$\pi \in \mathfrak{S}_N$ s.t. $\pi(G_0) = G_1$

Checks $\psi(G_b) = H$

Back to example 1: Graph Isomorphism

- **Soundness:**
  If $x \notin \mathcal{L}$, whatever a cheating prover does to sample H, it fails to answer the challenge with probability at least $1/2$

Back to example 1: Graph Isomorphism

- **Soundness:**
  If $x \notin \mathcal{L}$, whatever a cheating prover does to sample H, it fails to answer the challenge with probability

H is isomorphic to a most one of the G_b's

reveal a graph H

Checks $\psi(G_b) = H$

Back to example 1: Graph Isomorphism

- **Soundness:**
  If $x \notin \mathcal{L}$, whatever a cheating prover does to sample H, it fails to answer the challenge with probability

H is isomorphic to a most one of the G_b's

$\text{reveal a graph } H$ →

$\text{Pick } b \leftarrow U(\{0,1\}), \text{request mapping from } G_b \text{ to } H$ ←

$\text{Reveal } \psi$ →

$\text{Checks } \psi(G_b) = H$

Back to example 1: Graph Isomorphism

- **Soundness:**
  If $x \notin \mathcal{L}$, whatever a cheating prover does to sample H, it fails to answer the challenge with probability at least $1/2$

$$Pr[\langle P^*, V \rangle(x) = 1] \leq 1/2$$

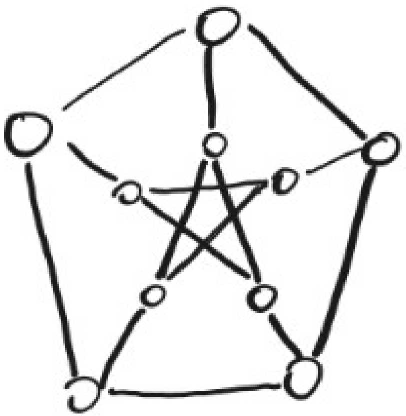
reveal a graph H →

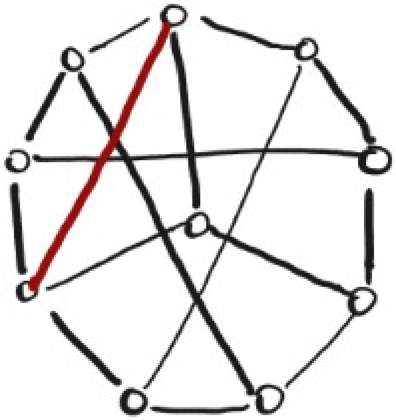
← Pick $b \leftarrow U(\{0,1\})$, request mapping from G_b to H

Reveal ψ →

Checks $\psi(G_b) = H$

Example 2: Graph Non-Isomorphism



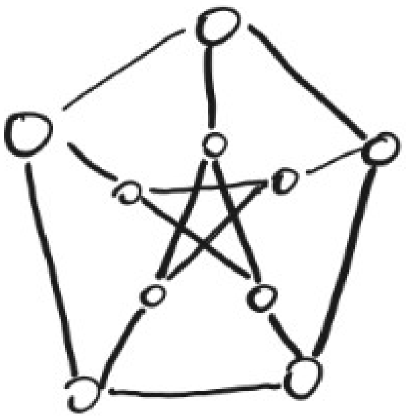


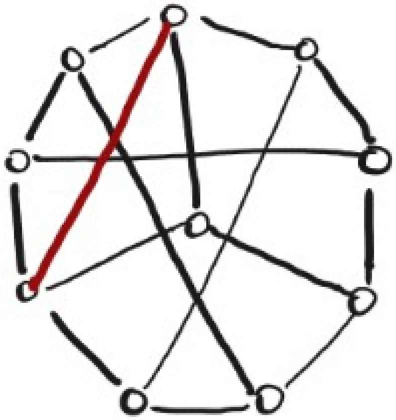
Only exponential-size classical (= non-interactive) proofs known

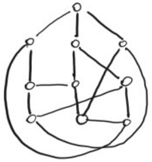
GNI is in co-NP, but it is conjectured that GNI is not in NP:

- polynomial hierarchy would collapse at level 2 [Schöning'88]
- GNI is in QP [Babai'16]

Example 2: Graph Non-Isomorphism







Pick $b \leftarrow U(\{0,1\}), \sigma \in \mathfrak{S}_N,$ reveal $\sigma(G_b)$

Return $b' \in \{0,1\}$

Checks $b' = b$

Example 2: Graph Non-Isomorphism

- **Completeness:**
  If $x \in \mathcal{L}$, then

$$Pr[\langle P, V \rangle(x) = 1] = 1$$

Pick $b \leftarrow U(\{0,1\}), \sigma \in \mathfrak{S}_N,$ reveal $\sigma(G_b)$

Return $b' \in \{0,1\}$

Checks $b' = b$

Example 2: Graph Non-Isomorphism

- **Soundness:**
  If $x \notin \mathcal{L}$, then $G_0 \equiv G_1$ and the distribution of the verifier's message is independent of b. The prover fails to guess b with probability $1/2$

$$Pr[\langle P, V\rangle(x) = 1] \leq 1/2$$

$\leftarrow$ Pick $b \leftarrow U(\{0,1\}), \sigma \in \mathfrak{S}_N,$ reveal $\sigma(G_b)$

$\rightarrow$ Return $b' \in \{0,1\}$

Checks $b' = b$

So, what can we prove with IP?

	Classical proofs	Interactive proofs
NP ∃ solution	✔	
co-NP ∀	?	
#P 178 solutions	?	
PSPACE ∃∀∃...∀	?	

So, what can we prove with IP?

	Classical proofs	Interactive proofs
NP ∃ solution	✔	✔
co-NP ∀	?	✔
#P 178 solutions	?	✔
PSPACE ∃∀∃…∀	?	✔

So, what can we prove with IP?

	Classical proofs	Interactive proofs
NP ∃ solution	✓	✓
co-NP ∀		
#P 178 solutions	?	✓
PSPACE ∃∀∃...∀	?	✓

Thm: [Fortnow-Karloff-Lund-Nissan'89, Shamir'89]

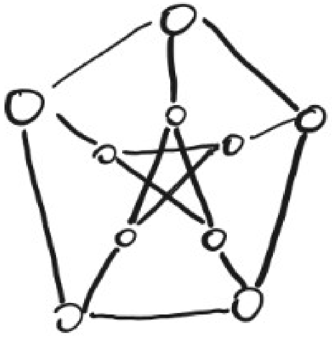
IP = PSPACE

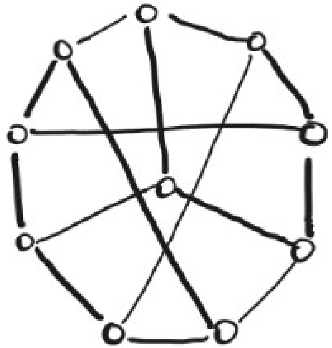
More about interactive proofs

- Our GNI proof requires private coins for the verifier
- What about public-coin protocols? (Arthur-Merlin classes, AM)
- AM = IP [Goldwasser-Sipser'86]
- Proof relies on the “Set lower bound” AM protocol

Zero-knowledge proofs.

Back to example 1 again: Graph Isomorphism





$\text{Pick a secret } \sigma \in \mathfrak{S}_N, \text{ reveal } H = \sigma(G_0)$ →

$\text{Pick } b \leftarrow U(\{0,1\}), \text{request mapping from } G_b \text{ to } H$ ←

$\text{Reveal } \psi, \text{ where } \psi = \sigma \text{ if } b = 0, \text{ else } \sigma \circ \pi^{-1}$ →

$\pi \in \mathfrak{S}_N \text{ s.t. } \pi(G_0) = G_1$

$\text{Checks } \psi(G_b) = H$

Back to example 1 again: Graph Isomorphism

What does the verifier learn about the witness?

Pick a secret $\sigma \in \mathfrak{S}_N$, reveal $H = \sigma(G_0)$ →

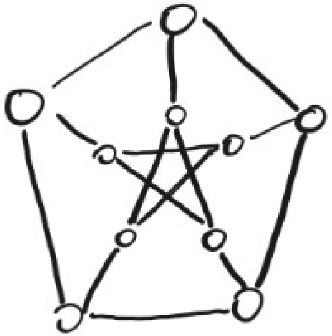
← Pick $b \leftarrow U(\{0,1\})$, request mapping from G_b to H

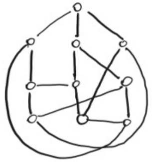
Reveal ψ, where $\psi = \sigma$ if $b = 0$, else $\sigma \circ \pi^{-1}$ →

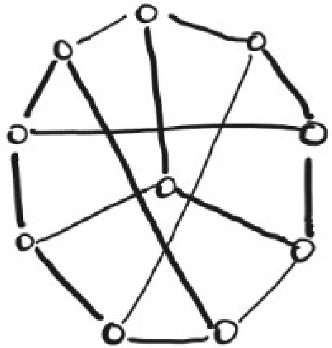
$\pi \in \mathfrak{S}_N$ s.t. $\pi(G_0) = G_1$

Checks $\psi(G_b) = H$

The view of the verifier







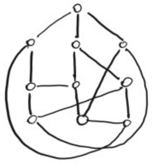
$$H = \sigma(G_0)$$

$$b \leftarrow U(\{0,1\})$$

$$\sigma \text{ if } b = 0, \text{ else } \sigma \circ \pi^{-1}$$

$$\pi \in \mathfrak{S}_N \text{ s.t. } \pi(G_0) = G_1$$

The view of the verifier



$$H = \sigma(G_0) = \sigma \circ \pi^{-1}(G_1)$$

$$b \leftarrow U(\{0,1\})$$

$$\sigma \text{ if } b = 0, \text{ else } \sigma \circ \pi^{-1}$$

The view of the verifier

$$b \leftarrow U(\{0,1\})$$

The view of the verifier

$$H = \sigma(G_b) \text{ for } \sigma \leftarrow \mathfrak{S}_N$$

$$b \leftarrow U(\{0,1\})$$

The view of the verifier

$$H = \sigma(G_b) \text{ for } \sigma \leftarrow \mathfrak{S}_N$$

→

$$b \leftarrow U(\{0,1\})$$

←

$$\sigma$$

→

Zero-knowledge interactive proofs

An interactive proof system (P, V) is:

- **Honest-verifier zero-knowledge:**
  if for $x \in \mathcal{L}$, there exists a probabilistic, poly-time simulator Sim_V

  such that we have:

$$\{\langle P, V \rangle(x)\} \approx \{\mathsf{Sim}_V(x)\}$$

Zero-knowledge interactive proofs

An interactive proof system (P, V) is:

- **<u>Honest-verifier zero-knowledge:</u>**
  if for $x \in \mathcal{L}$, there exists a probabilistic, poly-time simulator Sim_V

  such that we have:

$$\{\langle P, V\rangle(x)\} \approx \{\mathsf{Sim}_V(x)\}$$

The (honest) verifier learns nothing more than what it could get from the statement itself

Zero-knowledge interactive proofs

An interactive proof system (P,V) is:

- **Zero-knowledge:**
  If for $x \in \mathcal{L}$, for any (possibly malicious) verifier V^*, there exists a probabilistic, poly-time simulator Sim_{V^*} such that we have:

$$\{\langle P, V^* \rangle(x)\} \approx \{\mathsf{Sim}_{V^*}(x)\}$$

Zero-knowledge interactive proofs

An interactive proof system (P, V) is:

- **Zero-knowledge:**
  If for $x \in \mathcal{L}$, for any (possibly malicious) verifier V^*, there exists a probabilistic, poly-time simulator Sim_{V^*} such that we have:

$$\{\langle P, V^* \rangle(x)\} \approx \{\mathsf{Sim}_{V^*}(x)\}$$

Whatever it does, a verifier learns nothing more than what it could get from the statement itself

Different flavours of zero-knowledge

$$\{\langle P, V^* \rangle(x)\} \approx \{\mathsf{Sim}_{V^*}(x)\}$$

- Computational zero-knowledge
  simulated transcripts are hard to distinguish from real ones by PPT adversaries
- Statistical zero-knowledge
  an unbounded adversary learns nothing except with negligible probability
- Perfect zero-knowledge
  simulated transcripts and real transcripts are identically distributed

Different flavours of zero-knowledge

$$\{\langle P, V^* \rangle(x)\} \approx \{\mathsf{Sim}_{V^*}(x)\}$$

- Computational zero-knowledge = **CZK**

  simulated transcripts are hard to distinguish from real ones by PPT adversaries

- Statistical zero-knowledge = **SZK**

  an unbounded adversary learns nothing except with negligible probability

- Perfect zero-knowledge = **PZK**

  simulated transcripts and real transcripts are identically distributed

Different flavours of zero-knowledge

$$\{\langle P, V^* \rangle(x)\} \approx \{\mathsf{Sim}_{V^*}(x)\}$$

- Computational zero-knowledge = **CZK**

  simulated transcripts are hard to distinguish from real ones by PPT adversaries

- Statistical zero-knowledge = **SZK**

  an unbounded adversary learns nothing except with negligible probability

- Perfect zero-knowledge = **PZK**

  simulated transcripts and real transcripts are identically distributed

$$\text{BPP} \subseteq \text{PZK} \subseteq \text{SZK} \subseteq \text{CZK} \subseteq \text{IP}$$

NP $\subseteq$ CZK

Commitment scheme

$\mathsf{Com}(x; r)$ →

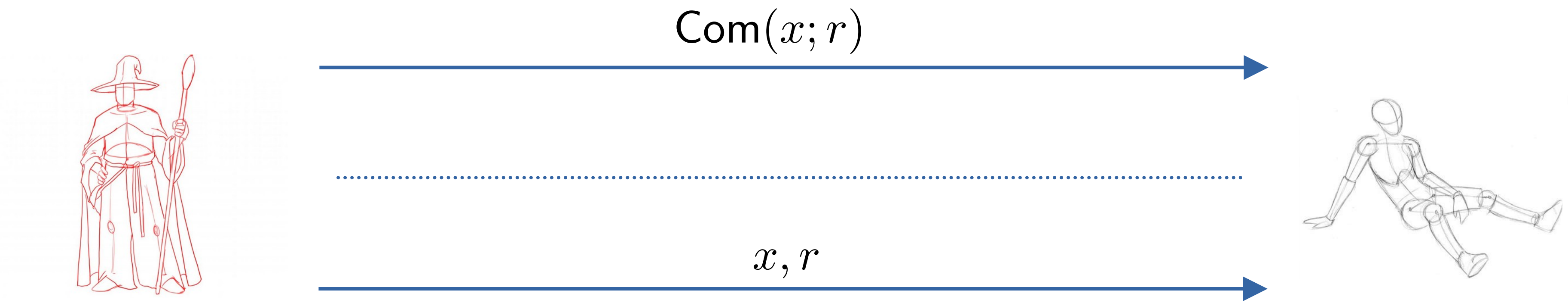
Commitment scheme

$\mathsf{Com}(x; r)$ →

······························

x, r →

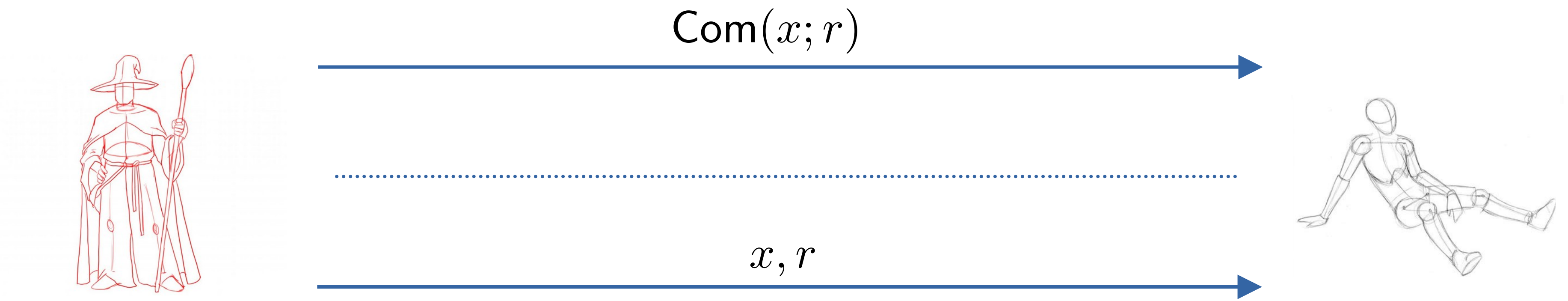
Commitment scheme



- **Hiding:**

  The receiver cannot learn anything about the committed value x before it is open

Commitment scheme



- **Hiding:**

  The receiver cannot learn anything about the committed value x before it is open

- **Binding:**

  The sender cannot open the commitment to any other value $x' \neq x$

Commitment scheme

$\mathsf{Com}(x; r)$ →

x, r →

- **Hiding:**

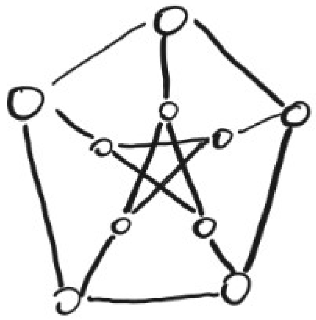
  The receiver cannot learn anything about the committed value x before it is open

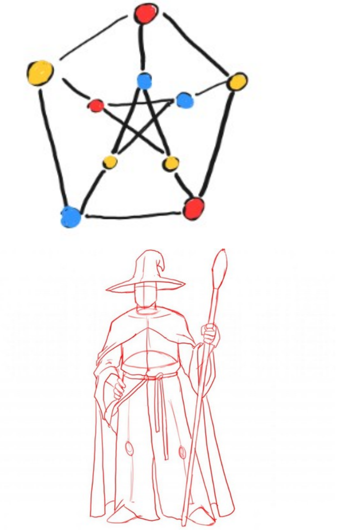
- **Binding:**

  The sender cannot open the commitment to any other value $x' \neq x$

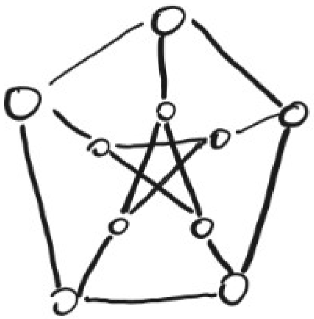
Commitment schemes with stat./comp. hiding and comp./stat. binding can be constructed assuming one-way functions exist

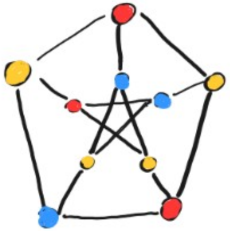
A zero-knowledge proof for 3-coloring





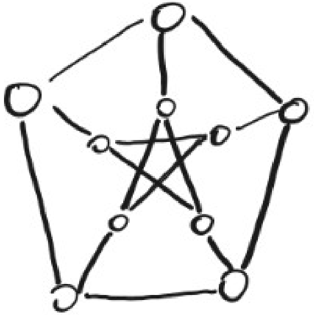
A zero-knowledge proof for 3-coloring

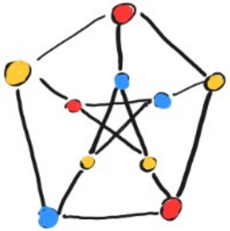




$\mathsf{Com}(c_k; r_k)_{k \in [N]}$

A zero-knowledge proof for 3-coloring

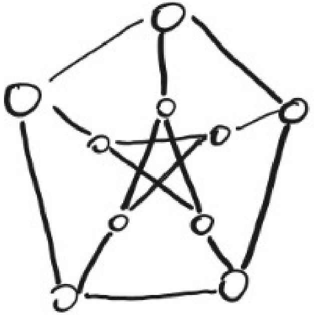


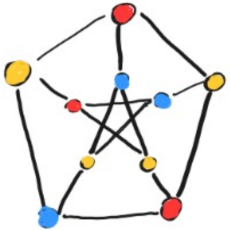


$\mathsf{Com}(c_k; r_k)_{k \in [N]}$ →

← $(i, j) \leftarrow U(E)$

A zero-knowledge proof for 3-coloring





$\mathsf{Com}(c_k; r_k)_{k \in [N]}$ →

← $(i, j) \leftarrow U(E)$

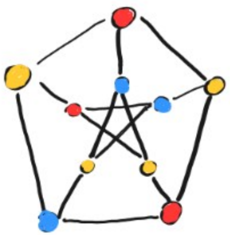
c_i, r_i, c_j, r_j →

Accept if $c_i \neq c_j$ and valid openings

A zero-knowledge proof for 3-coloring

- **Completeness:**
  If $x \in \mathcal{L}$, then

$$Pr[\langle P, V \rangle(x) = 1] = 1$$



$\mathsf{Com}(c_k; r_k)_{k \in [N]}$ →

← $(i, j) \leftarrow U(E)$

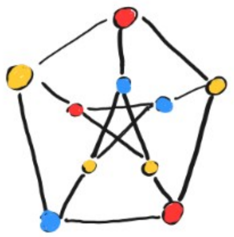
c_i, r_i, c_j, r_j →

Accept if $c_i \neq c_j$ and valid openings

A zero-knowledge proof for 3-coloring

- **Soundness:**
  If $x \notin \mathcal{L}$, then there must be an edge with the same color at both ends

$$Pr[\langle P^*, V\rangle(x) = 1] \leq 1 - \frac{1}{E}$$



$\mathsf{Com}(c_k; r_k)_{k\in[N]}$ →

← $(i,j) \leftarrow U(E)$

c_i, r_i, c_j, r_j →

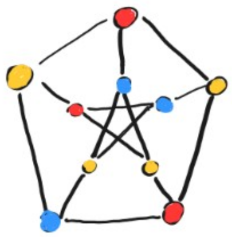
Accept if $c_i \neq c_j$ and valid openings

A zero-knowledge proof for 3-coloring

Binding: the prover has to open the two colors it committed

- **Soundness:**
  If $x \notin \mathcal{L}$, then there must be an edge with the same color at both ends

$$Pr[\langle P^*, V\rangle(x) = 1] \leq 1 - \frac{1}{E}$$



$\mathsf{Com}(c_k; r_k)_{k \in [N]}$ →

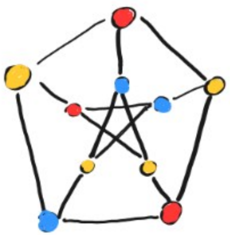
← $(i, j) \leftarrow U(E)$

c_i, r_i, c_j, r_j →

Accept if $c_i \neq c_j$ and valid openings

A zero-knowledge proof for 3-coloring

- **Honest-verifier zero-knowledge:**
  Actually, the verifier learns the color of 2 vertices at each iteration...
  There is an easy fix!



$$\mathsf{Com}(c_k; r_k)_{k \in [N]} \longrightarrow$$

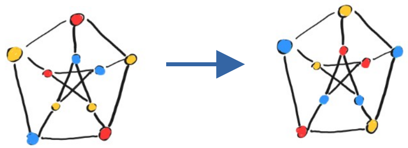
$$\longleftarrow (i, j) \leftarrow U(E)$$

$$c_i, r_i, c_j, r_j \longrightarrow$$

Accept if $c_i \neq c_j$ and valid openings

A zero-knowledge proof for 3-coloring

- **Honest-verifier zero-knowledge:**



randomly permutes the 3 colors, then commit

$\mathsf{Com}(c_k; r_k)_{k \in [N]}$ →

← $(i, j) \leftarrow U(E)$

c_i, r_i, c_j, r_j →

Accept if $c_i \neq c_j$ and valid openings

A zero-knowledge proof for 3-coloring

- **Honest-verifier zero-knowledge:**
  If $x \in \mathcal{L}$, then, we construct a simulator as:

randomly permutes the 3 colors, then commit

$\mathsf{Com}(c_k; r_k)_{k \in [N]}$ →

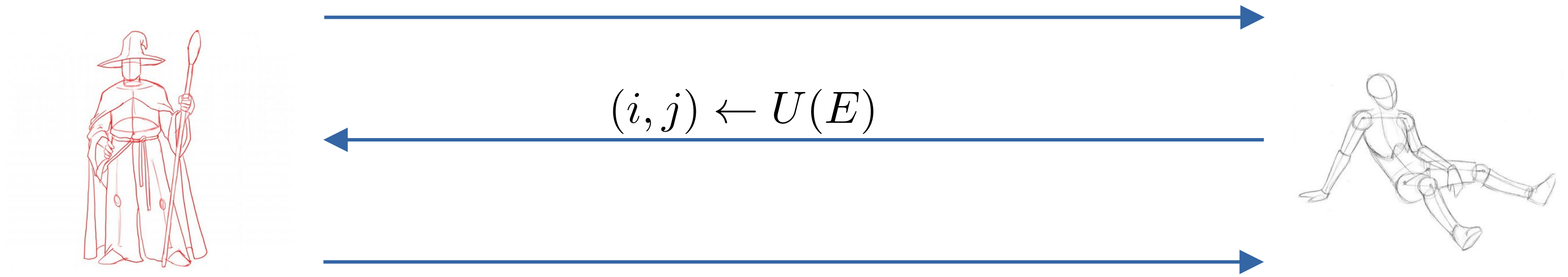
← $(i, j) \leftarrow U(E)$

c_i, r_i, c_j, r_j →

Accept if $c_i \neq c_j$ and valid openings

A zero-knowledge proof for 3-coloring

- **<u>Honest-verifier zero-knowledge:</u>**
  If $x \in \mathcal{L}$, then, we construct a simulator as:



A zero-knowledge proof for 3-coloring

- **Honest-verifier zero-knowledge:**
  If $x \in \mathcal{L}$, then, we construct a simulator as:

$$\text{For } k \in [N] \setminus \{i, j\}, \mathsf{Com}(0; r_k)_k$$

$$c_i \leftarrow U(\{1,2,3\}), c_j \leftarrow U(\{1,2,3\} \setminus \{c_i\})$$

$$\mathsf{Com}(c_i; r_i), \mathsf{Com}(c_j; r_j)$$

$$(i, j) \leftarrow U(E)$$

A zero-knowledge proof for 3-coloring

- **Honest-verifier zero-knowledge:**
  If $x \in \mathcal{L}$, then, we construct a simula

Hiding: from the verifier's perspective, non-open values look like commitments of 0

$$\text{For } k \in [N] \setminus \{i,j\}, \mathsf{Com}(0; r_k)_k$$

$$c_i \leftarrow U(\{1,2,3\}), c_j \leftarrow U(\{1,2,3\} \setminus \{c_i\})$$

$$\mathsf{Com}(c_i; r_i), \mathsf{Com}(c_j; r_j)$$

$$(i,j) \leftarrow U(E)$$

A zero-knowledge proof for 3-coloring

- **Honest-verifier zero-knowledge:**
  If $x \in \mathcal{L}$, then, we construct a simulator as:

$$\text{For } k \in [N] \setminus \{i, j\}, \mathsf{Com}(0; r_k)_k$$

$$c_i \leftarrow U(\{1,2,3\}), c_j \leftarrow U(\{1,2,3\} \setminus \{c_i\})$$

$$\mathsf{Com}(c_i; r_i), \mathsf{Com}(c_j; r_j)$$

$$(i, j) \leftarrow U(E)$$

$$c_i, r_i, c_j, r_j$$

Computational ZK

- One can actually prove that this protocol achieves computational zero-knowledge, but it is a bit more complicated $\Rightarrow$ *even a malicious verifier really learns nothing about the valid coloring*

- It is actually a **ZK proof of knowledge**: if a prover convinces a verifier, then the prover **has to know** a valid 3-coloring $\Rightarrow$ *the proof reveals nothing but it would be possible to extract a valid 3-coloring from interaction with the prover*

- Since 3-coloring is NP-complete, we obtain ZK-proofs for any statement in NP (assuming commitment schemes exist)...

$$\Rightarrow \text{NP} \subseteq \text{CZK}$$

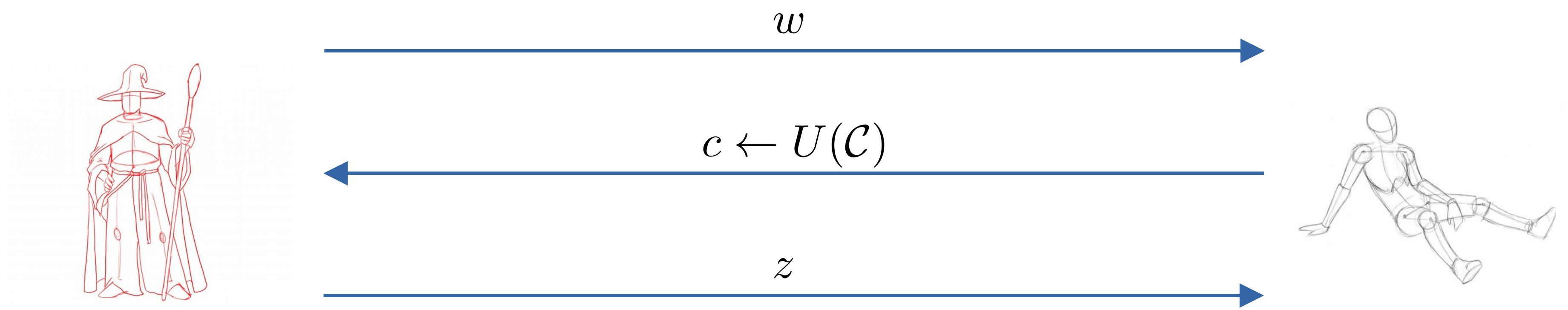
Concluding remarks.

Succinct ZK Proofs (ZK-SNARKs, ...)

- Combining ZK proofs with PCP lead to succinct zero-knowledge proofs (ZK-SNARKs)
- They allow to prove statements with extremely fast verification
- This is particularly useful for proving a complicated computation was honestly performed... Verification can be **much simpler** than the actual computation!

Non-Interactive Zero-Knowledge Proofs

- A lot of ZK proofs can be made non-interactive by relying on cryptographic hash functions using the Fiat-Shamir transform [Fiat-Shamir'86]



Non-Interactive Zero-Knowledge Proofs

- A lot of ZK proofs can be made non-interactive by relying on cryptographic hash functions using the Fiat-Shamir transform [Fiat-Shamir'86]

$$w$$

$$c \leftarrow H(w)$$

$$z$$

Conclusion

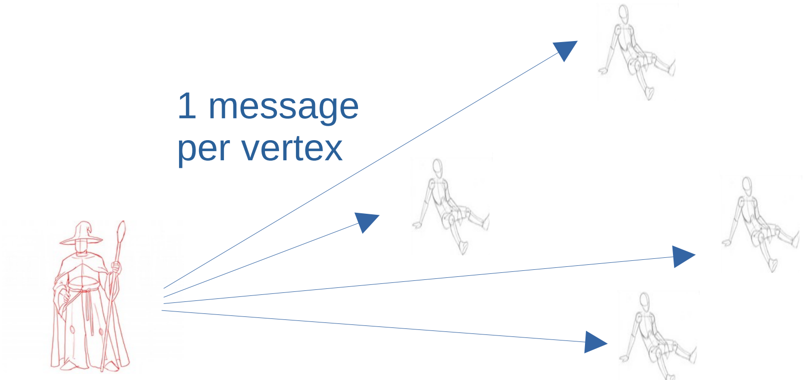
- ZK proofs are massively used in practice (they are at the core of modern digital signatures such as Schnorr or Dilithium)
- ZK proofs can be used to force honest behaviour in arbitrary scenarios
- We can prove statements about private data with ZK proofs (e.g., on encrypted data)
- There is high interest in succinct proofs for cloud computing, ML, cryptocurrencies... as they allow to certify the result of a computation at minimal cost

Some material and open problems

- To learn more:
  - zkproof.org
  - YouTube: Berkeley RDI Center - Zero-Knowledge Proofs MOOC
  - YouTube: ICMS - Foundations and Applications of Zero-Knowledge Proofs

Some material and open problems

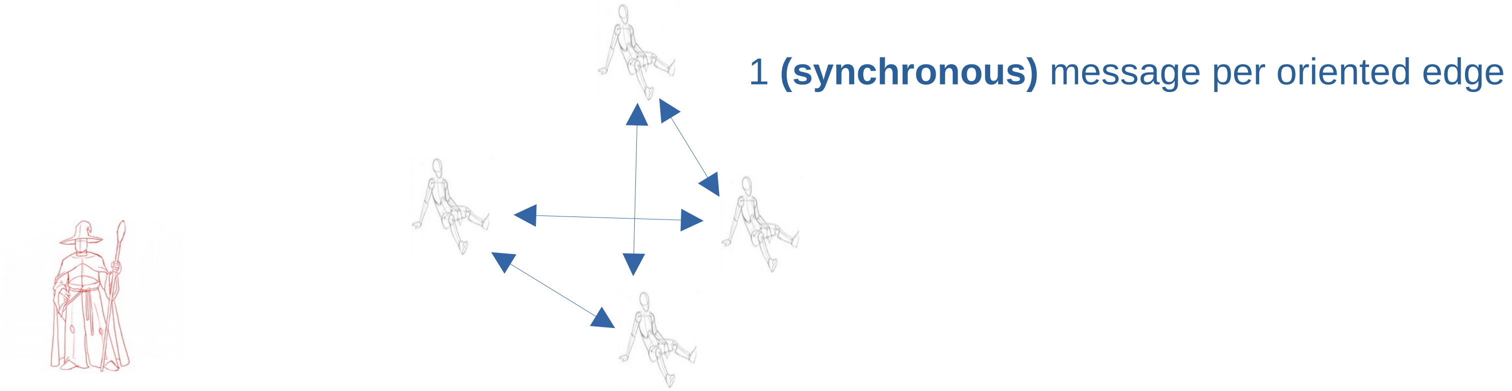
- To learn more:
  - zkproof.org
  - YouTube: Berkeley RDI Center - Zero-Knowledge Proofs MOOC
  - YouTube: ICMS - Foundations and Applications of Zero-Knowledge Proofs
- Some interesting open problems in *https://eprint.iacr.org/2025/202.pdf* - Distributed Non-Interactive ZK Proofs



Some material and open problems

- To learn more:
  - zkproof.org
  - YouTube: Berkeley RDI Center - Zero-Knowledge Proofs MOOC
  - YouTube: ICMS - Foundations and Applications of Zero-Knowledge Proofs

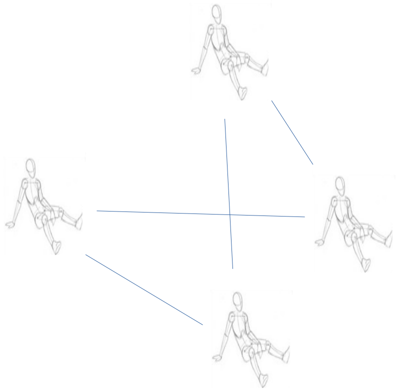
- Some interesting open problems in *https://eprint.iacr.org/2025/202.pdf* - Distributed Non-Interactive ZK Proofs



Some material and open problems

- To learn more:
  - zkproof.org
  - YouTube: Berkeley RDI Center - Zero-Knowledge Proofs MOOC
  - YouTube: ICMS - Foundations and Applications of Zero-Knowledge Proofs

- Some interesting open problems in *https://eprint.iacr.org/2025/202.pdf* - Distributed Non-Interactive ZK Proofs



Goal: convince the network of some property (e.g. triangle-freeness) in ZK, possibly in presence of coalitions of malicious nodes

Some material and open problems

- To learn more:
  - zkproof.org
  - YouTube: Berkeley RDI Center - Zero-Knowledge Proofs MOOC
  - YouTube: ICMS - Foundations and Applications of Zero-Knowledge Proofs

- Some interesting open problems in *https://eprint.iacr.org/2025/202.pdf* - Distributed Non-Interactive ZK Proofs

- Non NP-complete graph problems in SZK?

Some material and open problems

- To learn more:
  - zkproof.org
  - YouTube: Berkeley RDI Center - Zero-Knowledge Proofs MOOC
  - YouTube: ICMS - Foundations and Applications of Zero-Knowledge Proofs

- Some interesting open problems in *https://eprint.iacr.org/2025/202.pdf* - Distributed Non-Interactive ZK Proofs

- Non NP-complete graph problems in SZK?

Thanks!