Non-Cooperative Rational Interactive Proofs

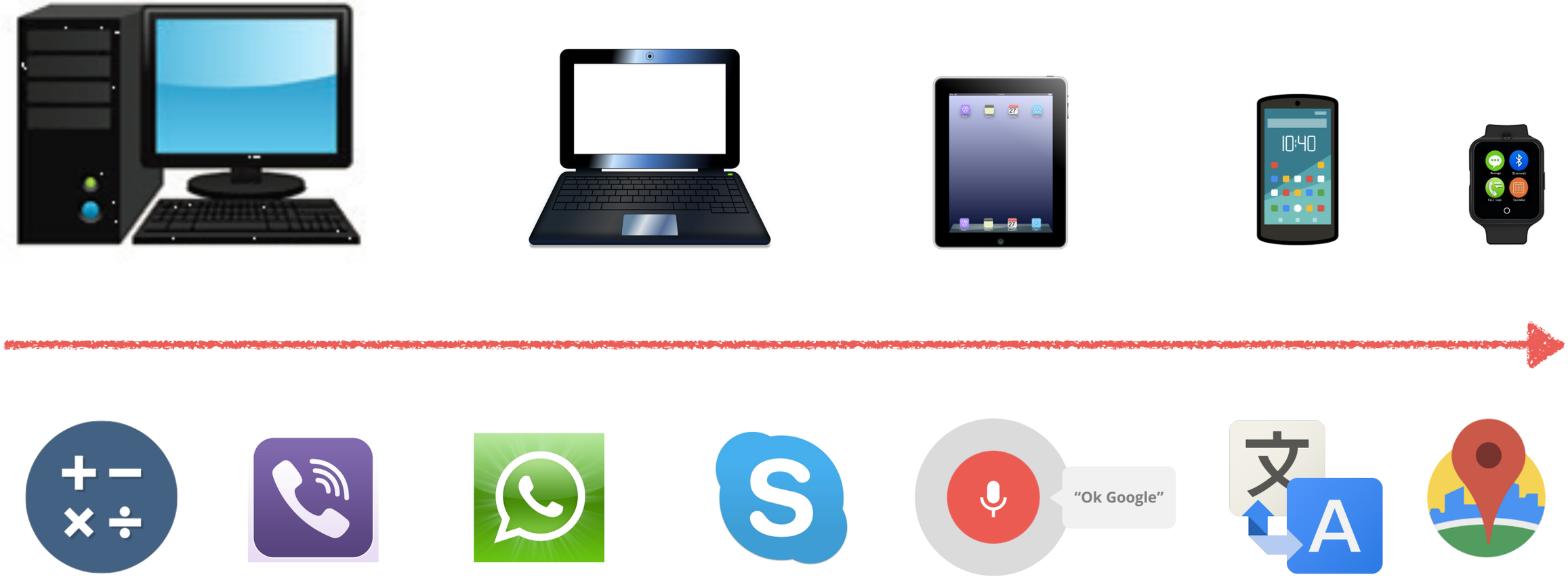
Jing Chen
Stony Brook University

Samuel McCauley
Williams College

Shikha Singh
Williams College

Modern Computing Challenges

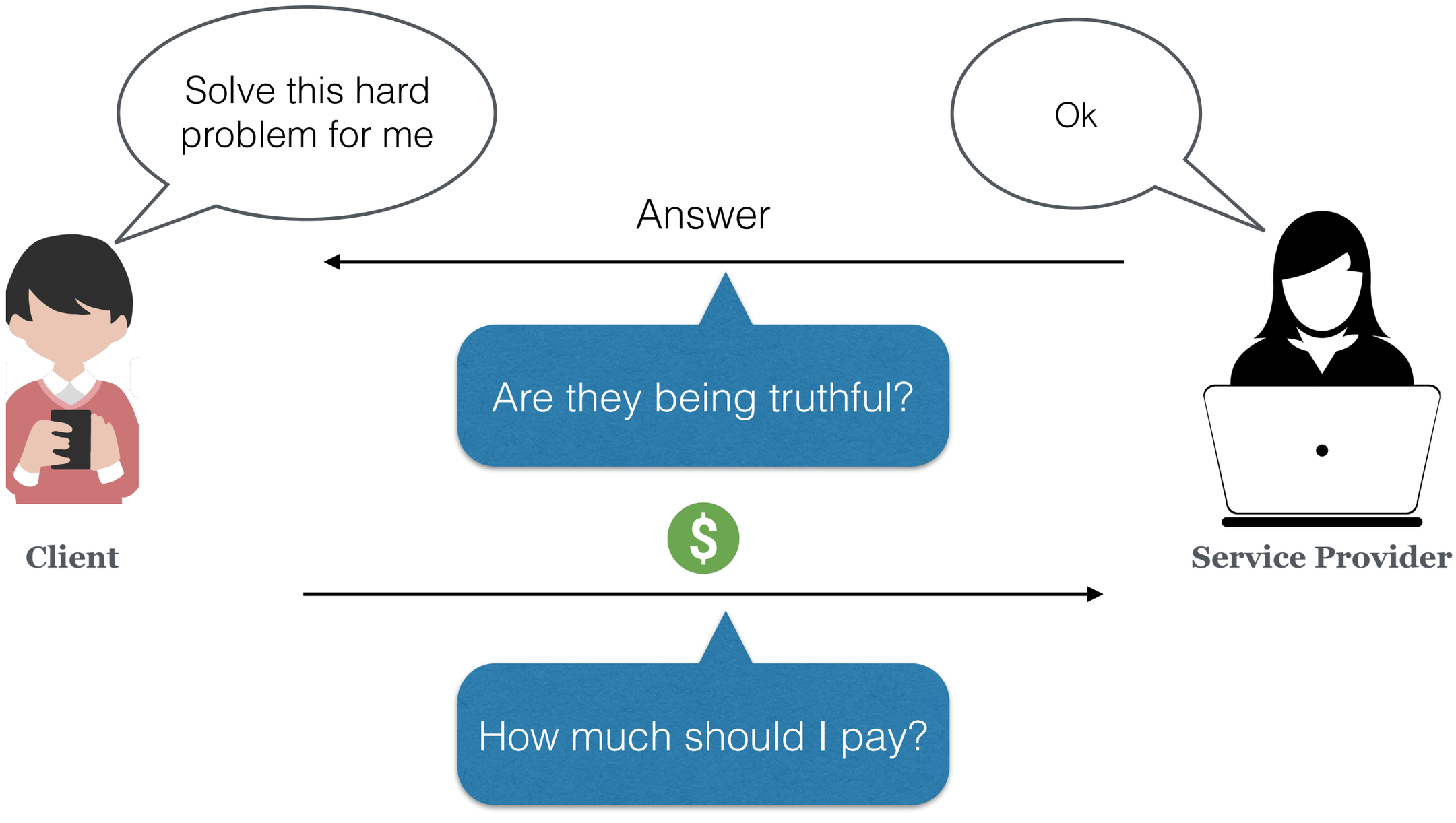
- Computational devices are getting smaller
- The computation they need to perform is getting more complex



Computation: A Commodity

- Today most large-scale computation is outsourced to service providers
- Buy their computational services for money

Outsourcing Computation



Research Question

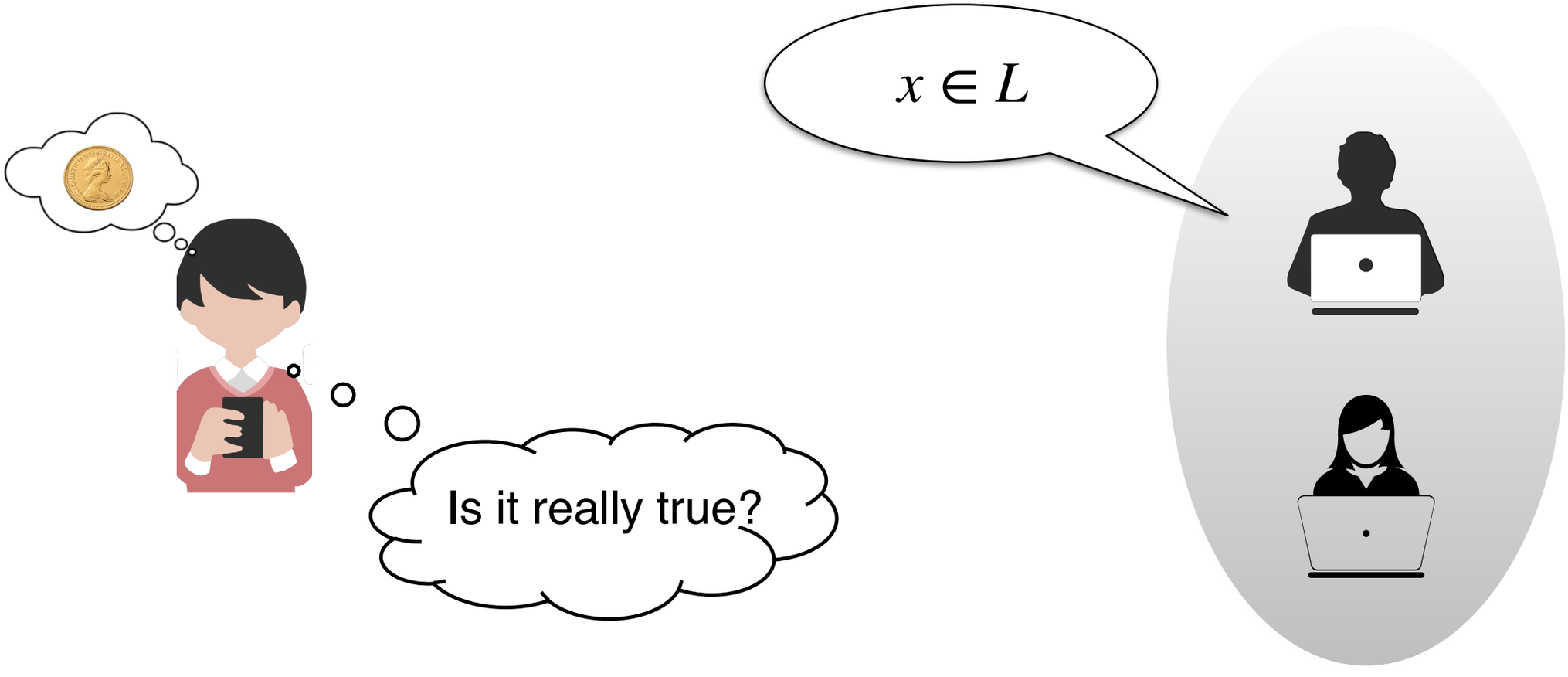
If someone else claims to have solved your problem, how do you know they are telling the truth?

How do we verify correctness of outsourced computation efficiently (without re-executing them)?

(Multi-Prover) Interactive Proofs

[GMR, BM 85, BGWW 88]

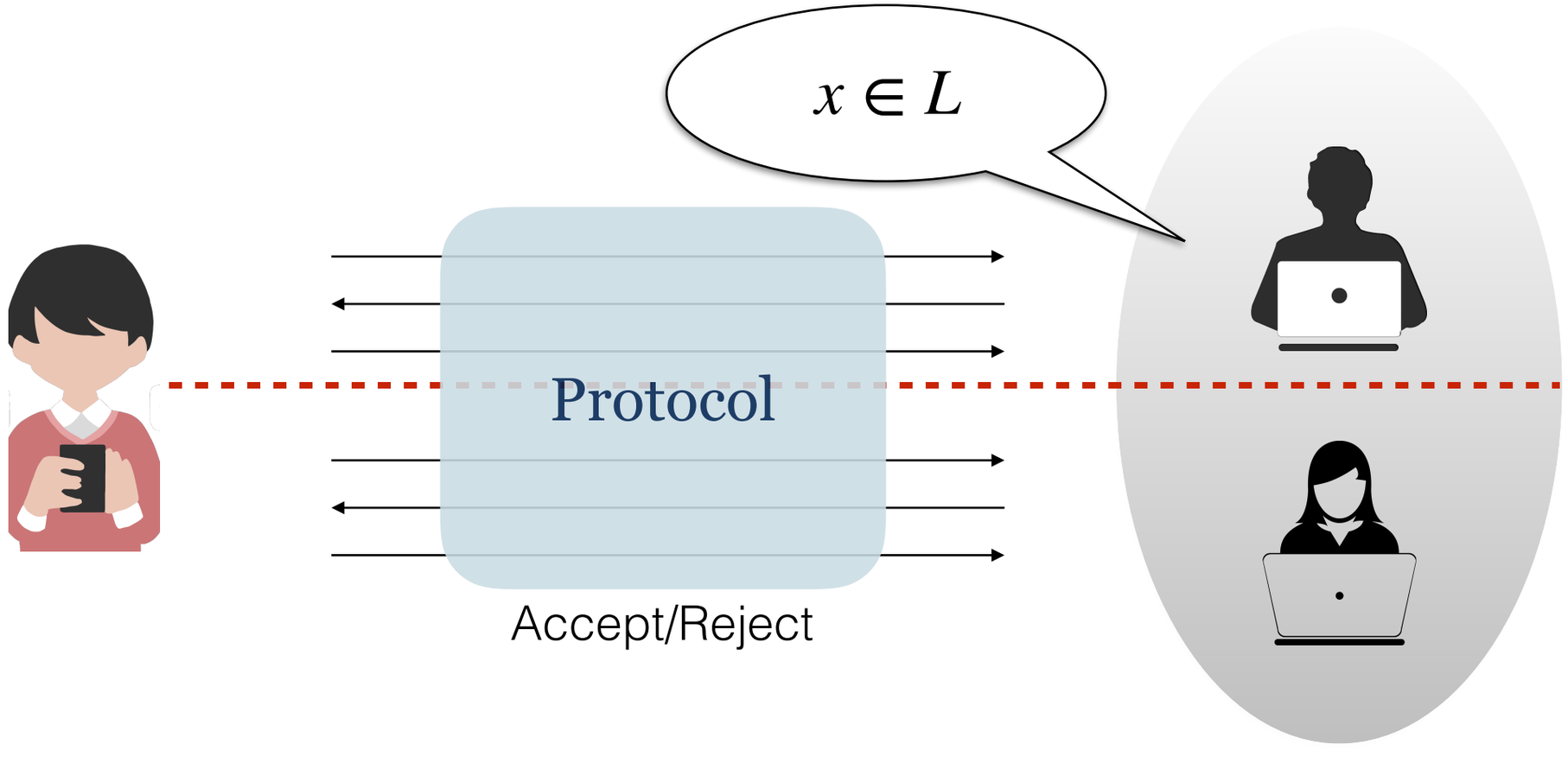
- Formal framework to study verification of outsourced computation
- Verifier is probabilistic polynomial time, provers are unbounded
- Provers goal is to prove that a string x is in language L



(Multi-Prover) Interactive Proofs

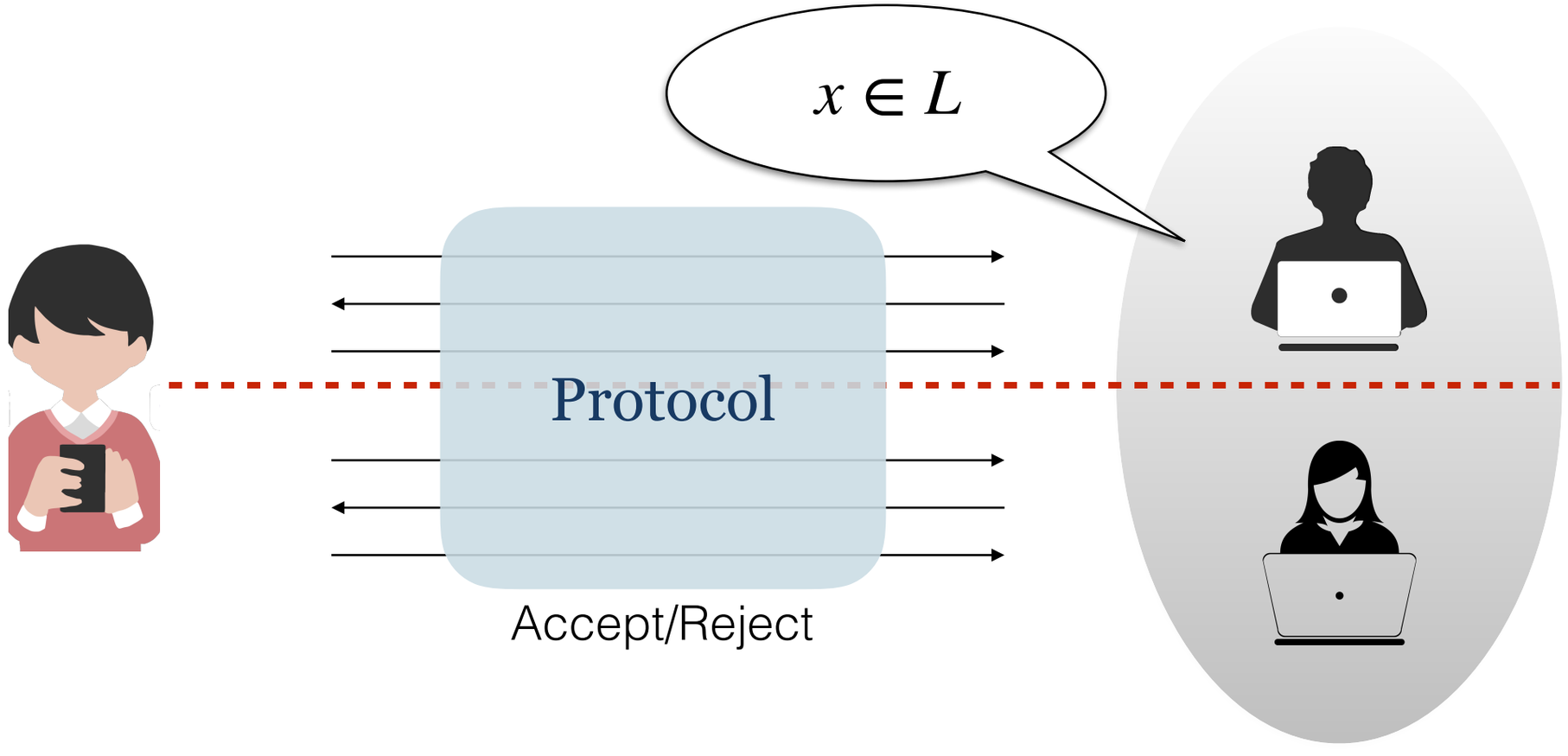
[GMR, BM 85, BGWW 88]

- Verifier interacts with each prover **separately**
    - Asking them questions to check if they are being truthful
- Finally, if Verifier is convinced, he **accepts**. Otherwise, he **rejects**



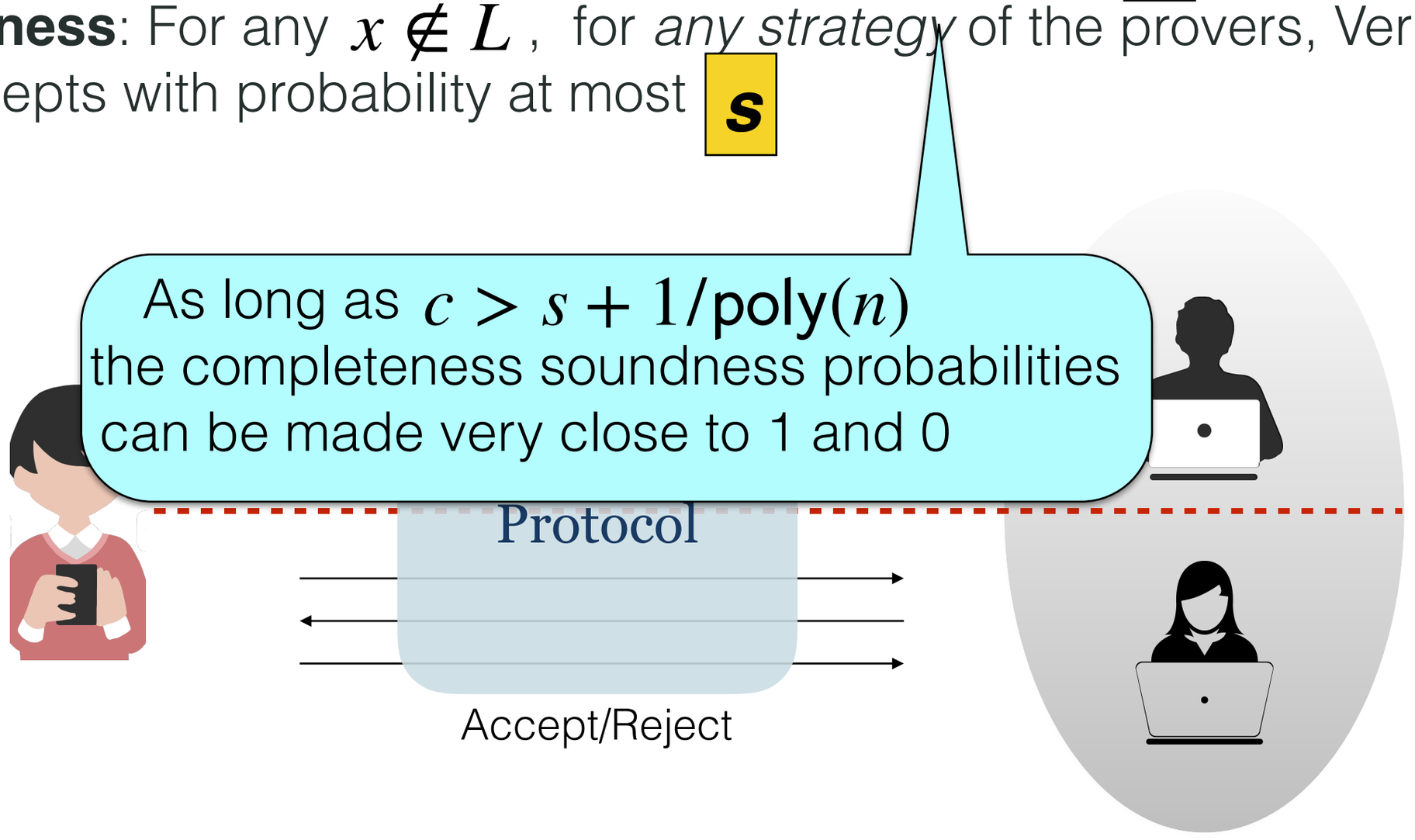
Interactive Proof Guarantees

- **Completeness**: For any $x \in L$, there exists a strategy of the provers such that Verifier accepts with probability at least 2/3
- **Soundness**: For any $x \notin L$, for *any strategy* of the provers, Verifier accepts with probability at most 1/3



Interactive Proof Guarantees

- **Completeness**: For any $x \in L$, there exists a strategy of the provers such that Verifier accepts with probability at least **c**
- **Soundness**: For any $x \notin L$, for *any strategy* of the provers, Verifier accepts with probability at most **s**



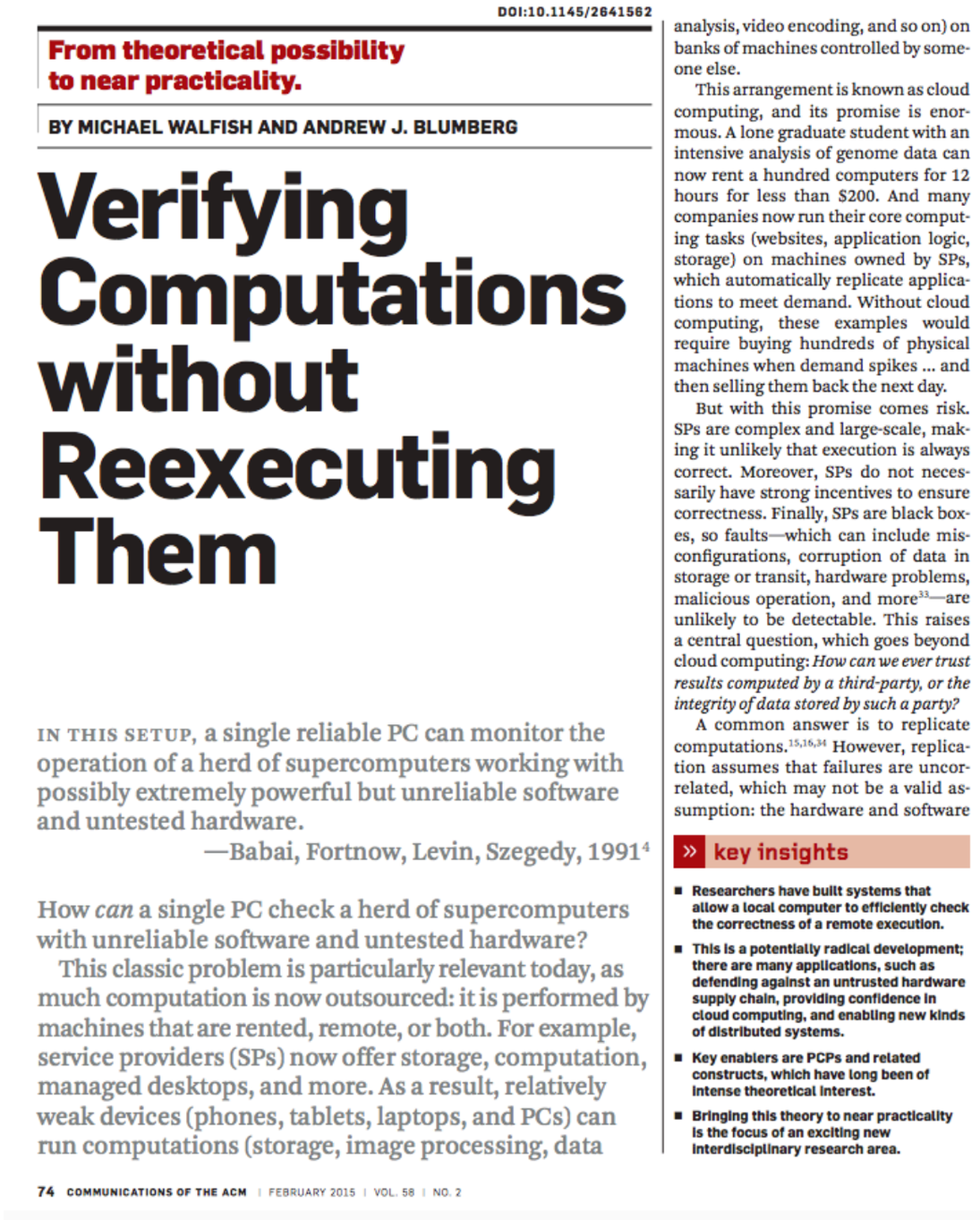
Rich History of IPs

- Widely-studied area with deep results in complexity theory
  - IP = **PSPACE** [S90]
  - MIP = **NEXP** [BFL91, BFLS91]
    - Probabilistically checkable proofs for **NP**

      [AS92, FGLSS91, AS92, ALMSS92]
- Game-theoretic characterization of complexity classes
  - Using refereed-games model [CS96, FL92, FA92, FK9, etc.]
  - **EXP** characterized as two-player zero-sum game [FK97]

Formal Framework to Study Computation Outsourcing

- IP for muggles [GKR08]
- Proofs of proximity [RVW13, GR15, KR15]
- Survey of recent developments [W15]
- **Rational proofs** and arguments
  [AM12&13, GHRV14&16, CMS16, etc.]
- Refereed-games based delegation
  [AM12, AM13, GHRV14, GHRV16, etc.]



DOI:10.1145/2641562

From theoretical possibility to near practicality.

BY MICHAEL WALFISH AND ANDREW J. BLUMBERG

Verifying Computations without Reexecuting Them

IN THIS SETUP, a single reliable PC can monitor the operation of a herd of supercomputers working with possibly extremely powerful but unreliable software and untested hardware.

—Babai, Fortnow, Levin, Szegedy, 1991[4]

How *can* a single PC check a herd of supercomputers with unreliable software and untested hardware?

This classic problem is particularly relevant today, as much computation is now outsourced: it is performed by machines that are rented, remote, or both. For example, service providers (SPs) now offer storage, computation, managed desktops, and more. As a result, relatively weak devices (phones, tablets, laptops, and PCs) can run computations (storage, image processing, data analysis, video encoding, and so on) on banks of machines controlled by someone else.

This arrangement is known as cloud computing, and its promise is enormous. A lone graduate student with an intensive analysis of genome data can now rent a hundred computers for 12 hours for less than $200. And many companies now run their core computing tasks (websites, application logic, storage) on machines owned by SPs, which automatically replicate applications to meet demand. Without cloud computing, these examples would require buying hundreds of physical machines when demand spikes ... and then selling them back the next day.

But with this promise comes risk. SPs are complex and large-scale, making it unlikely that execution is always correct. Moreover, SPs do not necessarily have strong incentives to ensure correctness. Finally, SPs are black boxes, so faults—which can include misconfigurations, corruption of data in storage or transit, hardware problems, malicious operation, and more[33]—are unlikely to be detectable. This raises a central question, which goes beyond cloud computing: *How can we ever trust results computed by a third-party, or the integrity of data stored by such a party?*

A common answer is to replicate computations.[15,16,34] However, replication assumes that failures are uncorrelated, which may not be a valid assumption: the hardware and software

» key insights

- Researchers have built systems that allow a local computer to efficiently check the correctness of a remote execution.
- This is a potentially radical development; there are many applications, such as defending against an untrusted hardware supply chain, providing confidence in cloud computing, and enabling new kinds of distributed systems.
- Key enablers are PCPs and related constructs, which have long been of intense theoretical interest.
- Bringing this theory to near practicality is the focus of an exciting new interdisciplinary research area.

The Two Extremes Of Existing MIP Models

[GMR, BM 85, BGWW 88]

Classic IPs

Refereed Games

Cooperative ——— Competitive

Rational IPs

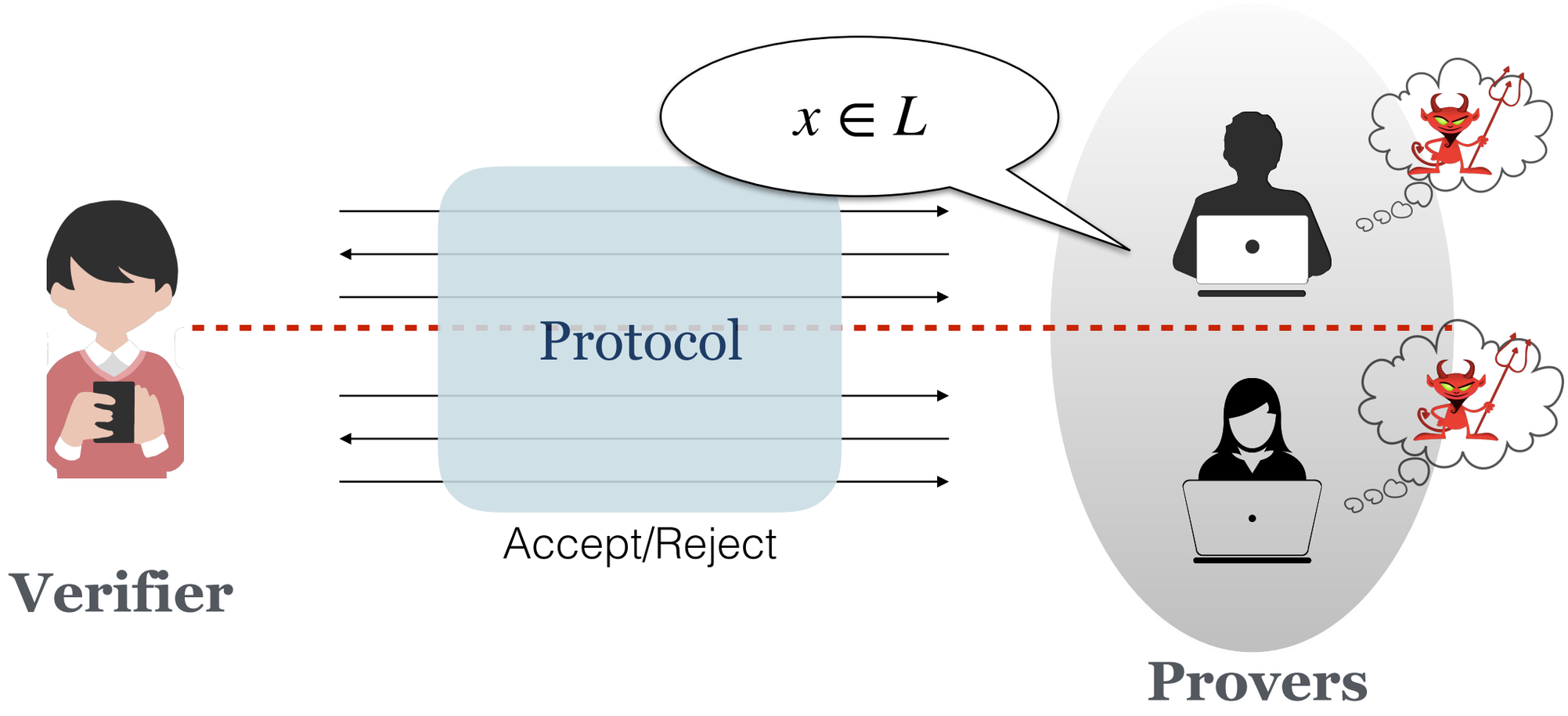
[AM12, CMS16]

[CS76, FK97, FKS95, etc.]

Classic IPs

Cooperative & Adversarial

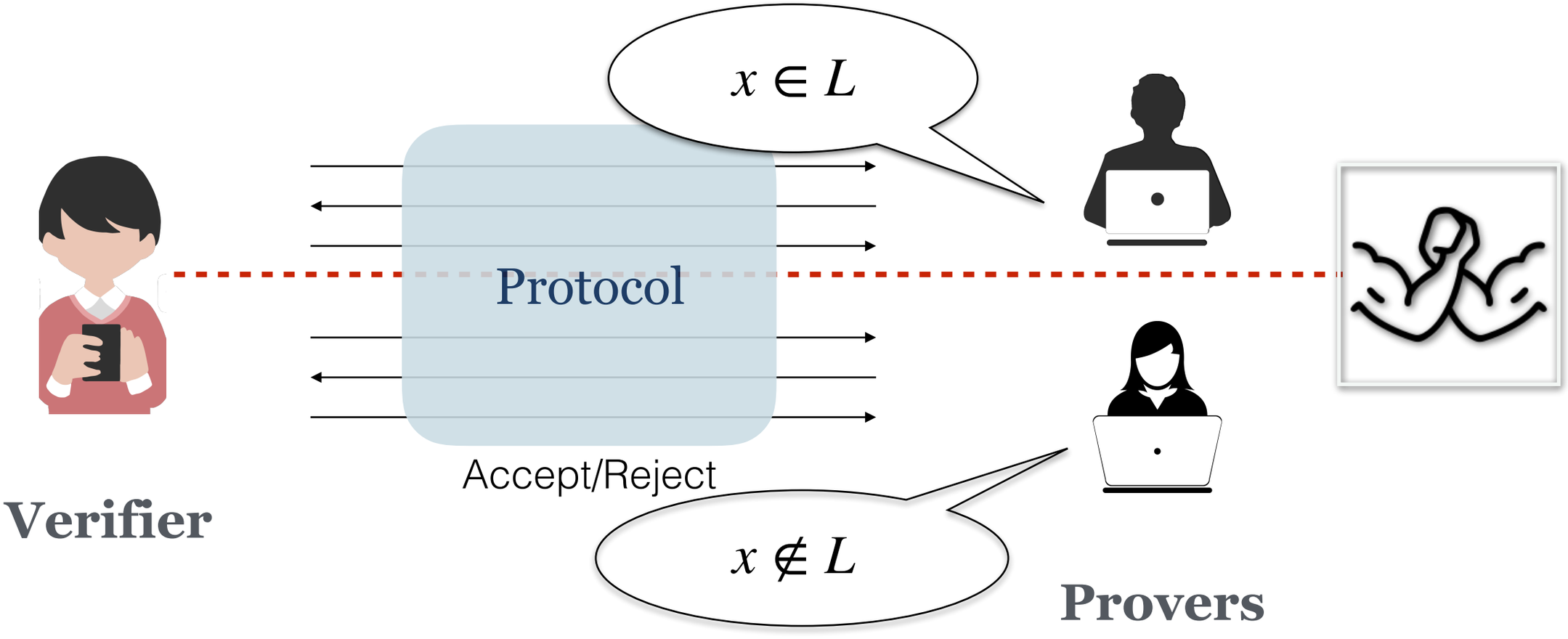
- Provers work together as a team to mislead the verifier
  - Convince the verifier that $x \in L$ (no matter what the truth is)
  - Joint utility $=$ Prob[V accepts claim $x \in L$]



Refereed Games

Competitive & Opinionated

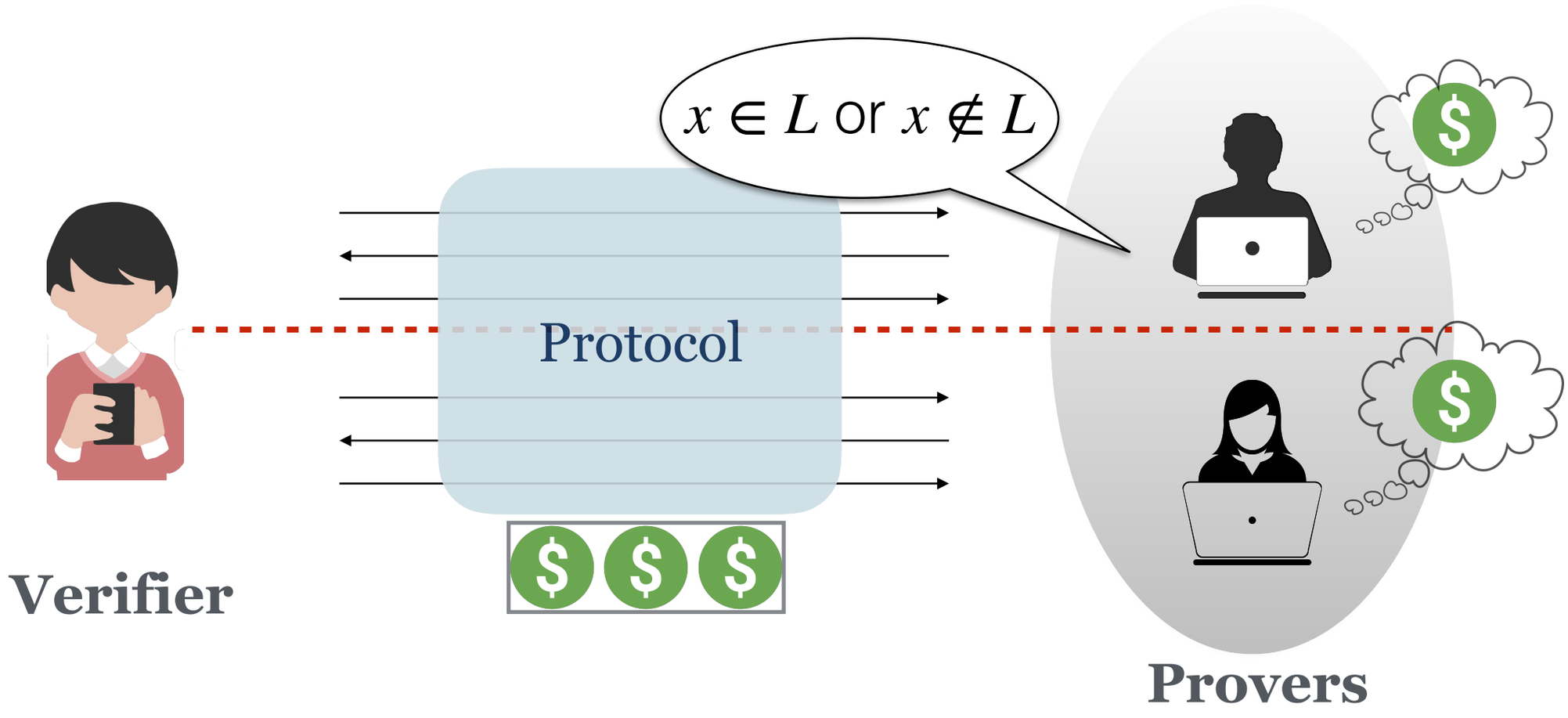
- Provers compete with each other in a zero-sum game
  - P_1 proves $x \in L$, P_2 proves $x \notin L$ (at least one honest prover)
  - $u_1 = \text{Prob}(V \text{ accepts } x \in L)$, $u_2 = \text{Prob}(V \text{ accepts } x \notin L)$



Rational IPs

Cooperative & Rational

- Provers work together as a team to **maximize total payment**
  - Prove whichever claim $x \in L$ or $x \notin L$ maximizes payment
  - Joint utility $u = \mathbb{E}(R)$ (expected reward given by verifier)



In Between the Extremes: Generalized Incentives

[GMR, BM 85, BGWW 88]

Classic IPs

Non-Cooperative RIPs

Refereed Games

Cooperative

Competitive

Neither Cooperative or Competitive

[This Work]

Rational IPs

[AM12, CMS16]

[CS76, FK97, FKS95, etc.]

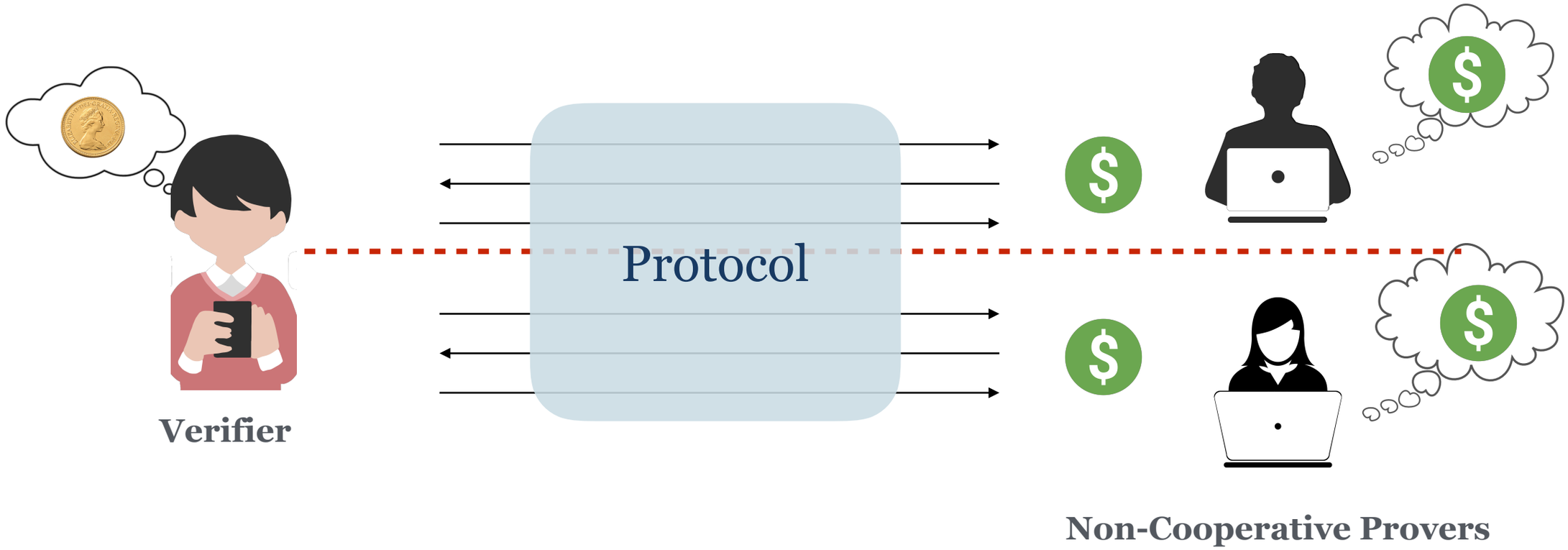
Verification of Outsourced Computation:

A Mechanism-Design Problem

How can we design payment-based protocols that incentivize rational and non-cooperative provers to give us correct answers?

Non-Cooperative Rational Interactive Proofs (ncRIP)

- Each prover gets paid separately
- Want to maximize their own expected payment, given others' strategies



Mechanism Design Considerations

- As all the provers are selfish and act on their own
    - Need a meaningful equilibrium concept
    - Where no prover can unilaterally deviate to improve payment
- Need to design the rules of the game (protocol and payment) s.t.
    - Verifier learns the correct answer at such an equilibrium

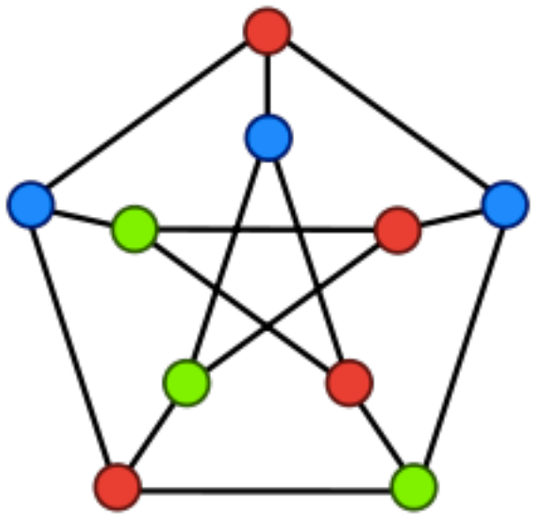
Equilibrium Intuition

- We consider a simple protocol and reason using backward induction
    - Reason backwards in time to decide how provers will act

- This captures the spirit of ncRIP, without the messy details
    - This is not the actual equilibrium but it’ll do for now

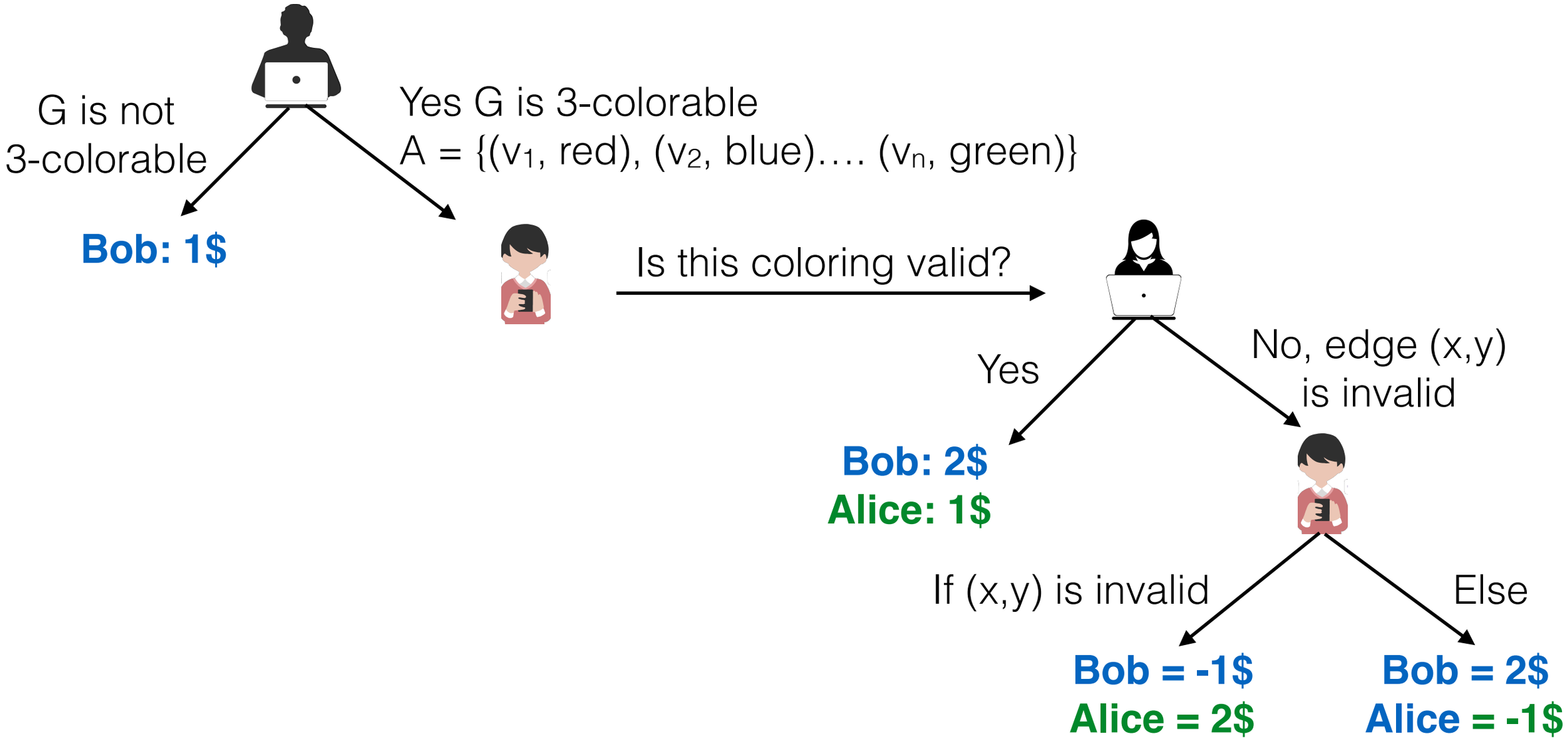
Example Protocol for NP

- Consider the NP-complete of Graph Coloring (GC):
    - Is a given graph 3-colorable?
- Warm up: O(log n)-time rational proof for GC
- Similar to PCPs, Verifier has “oracle access” to purported proof
- Simplified model (brushing some details under the rug)



A 3-colorable graph

ncRIP for NP


G is not
3-colorable
Yes G is 3-colorable
A = {(v1, red), (v2, blue)…. (vn, green)}
Bob: 1$
Is this coloring valid?
Yes
No, edge (x,y)
is invalid
Bob: 2$
Alice: 1$
If (x,y) is invalid
Else
Bob = -1$
Alice = 2$
Bob = 2$
Alice = -1$

ncRIP for NP

If A is an invalid coloring

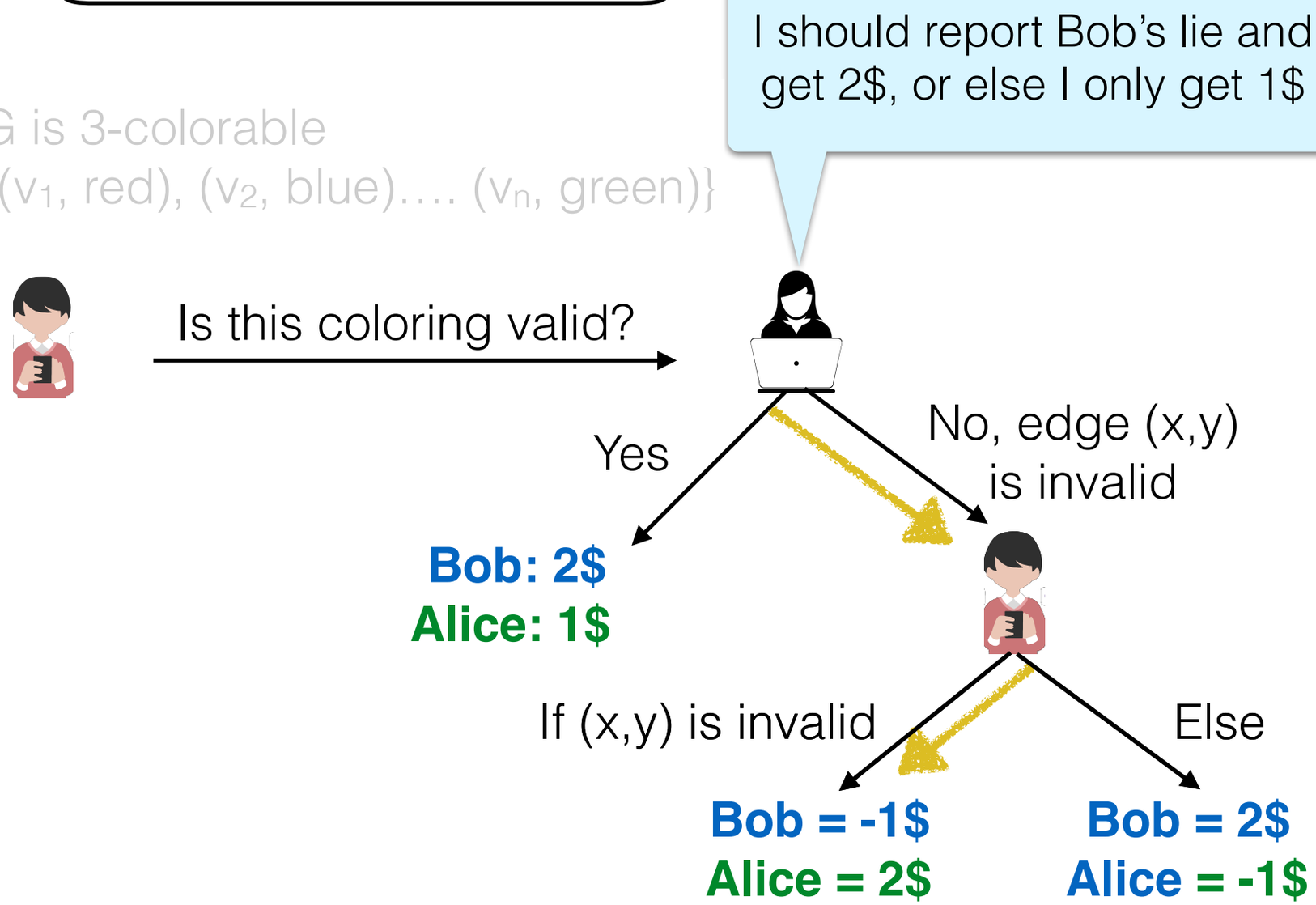
I should report Bob's lie and get 2$, or else I only get 1$

G is not 3-colorable

$R_1 = 1$

Yes G is 3-colorable

A = {(v_1, red), (v_2, blue)…. (v_n, green)}



ncRIP for NP

If A is a valid coloring

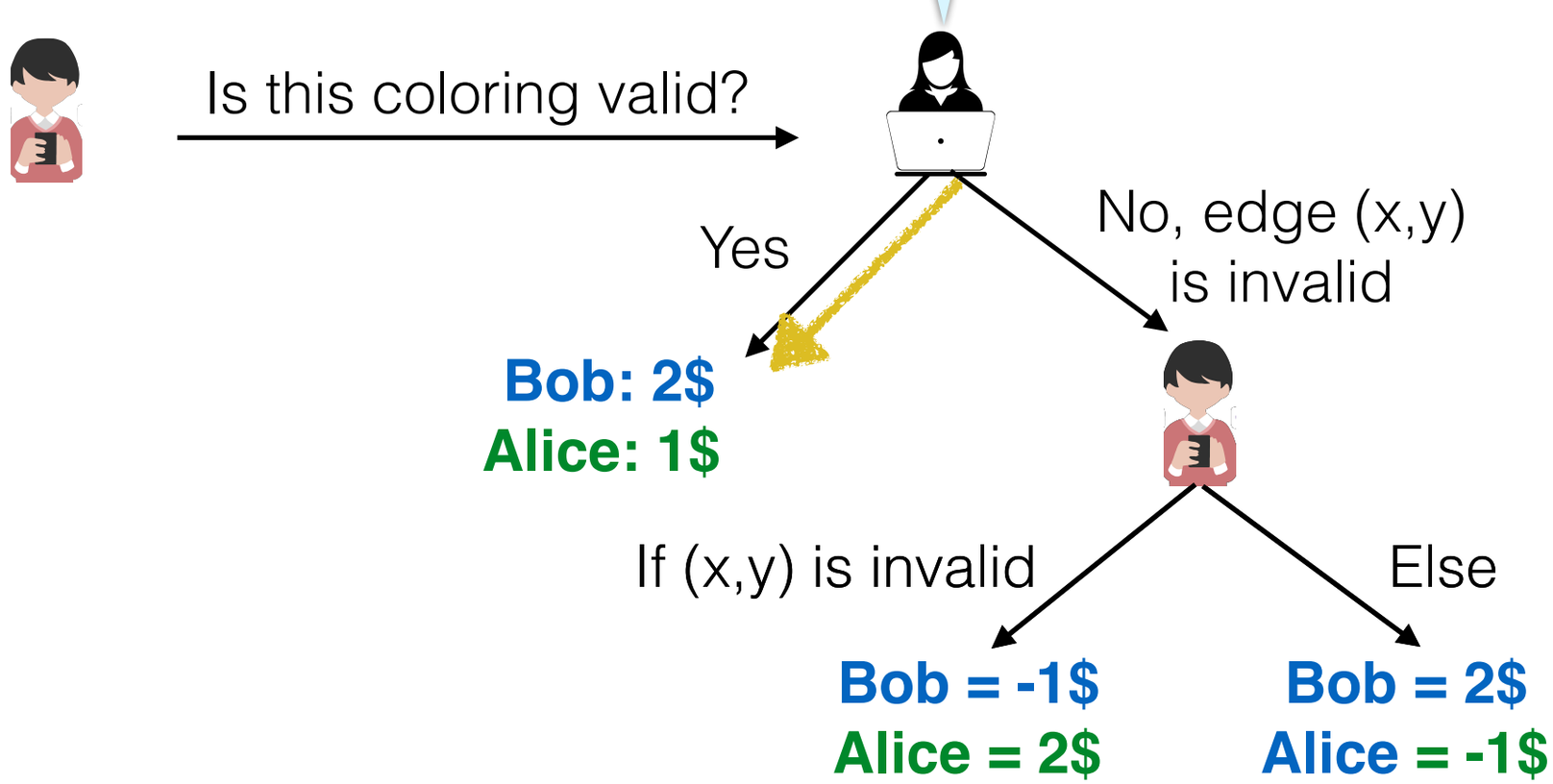
I should agree with Bob and get 1$ or else I will get caught and get -1$

G is not 3-colorable

$R_1 = 1$

Yes G is 3-colorable

A = {(v_1, red), (v_2, blue)…. (v_n, green)}



ncRIP for NP

If G is not 3-colorable

G is not 3-colorable

Bob: 1$

If I lie and send an invalid coloring, Alice will report me and I will get -1$. I should be truthful and get 1$.

Yes G is 3-colorable
A = {(v_1, red), (v_2, blue)…. (v_n, green)}

Is this coloring valid?

Yes

Bob: 2$
Alice: 1$

No, edge (x,y) is invalid

If (x,y) is invalid

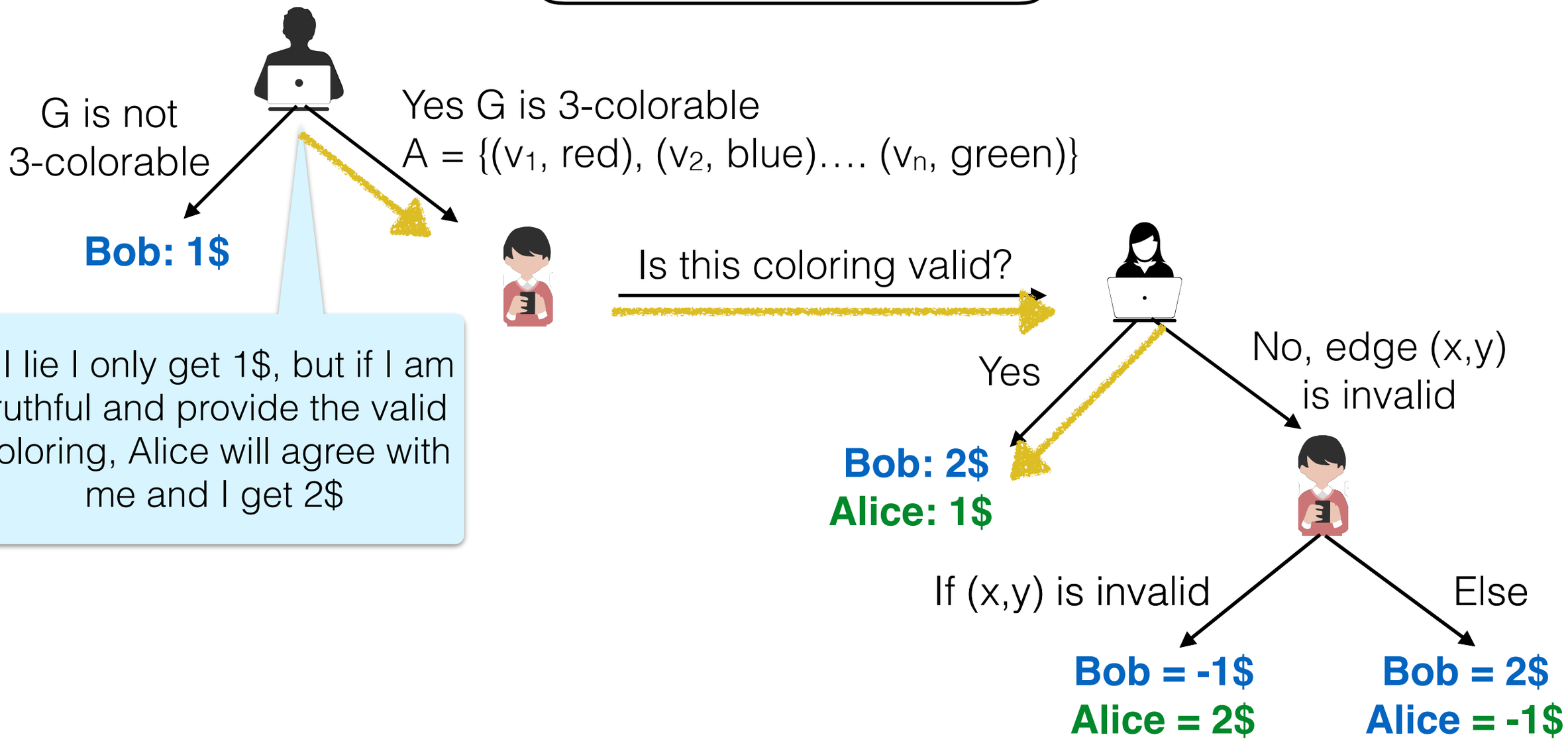
Bob = -1$
Alice = 2$

Else

Bob = 2$
Alice = -1$

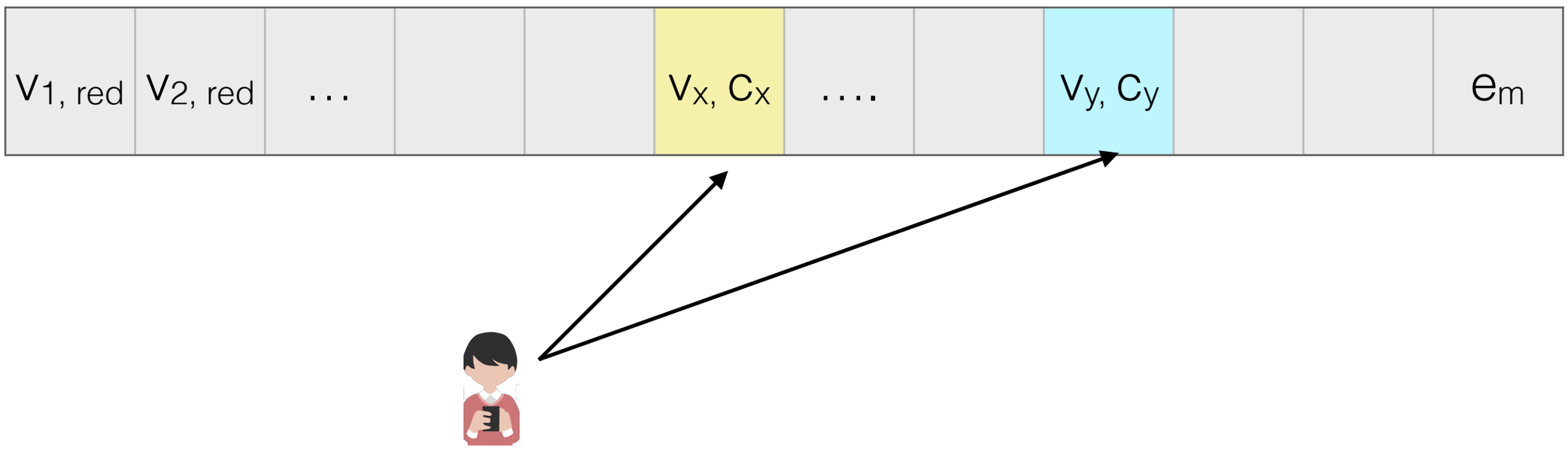
ncRIP for NP

If G is 3-colorable



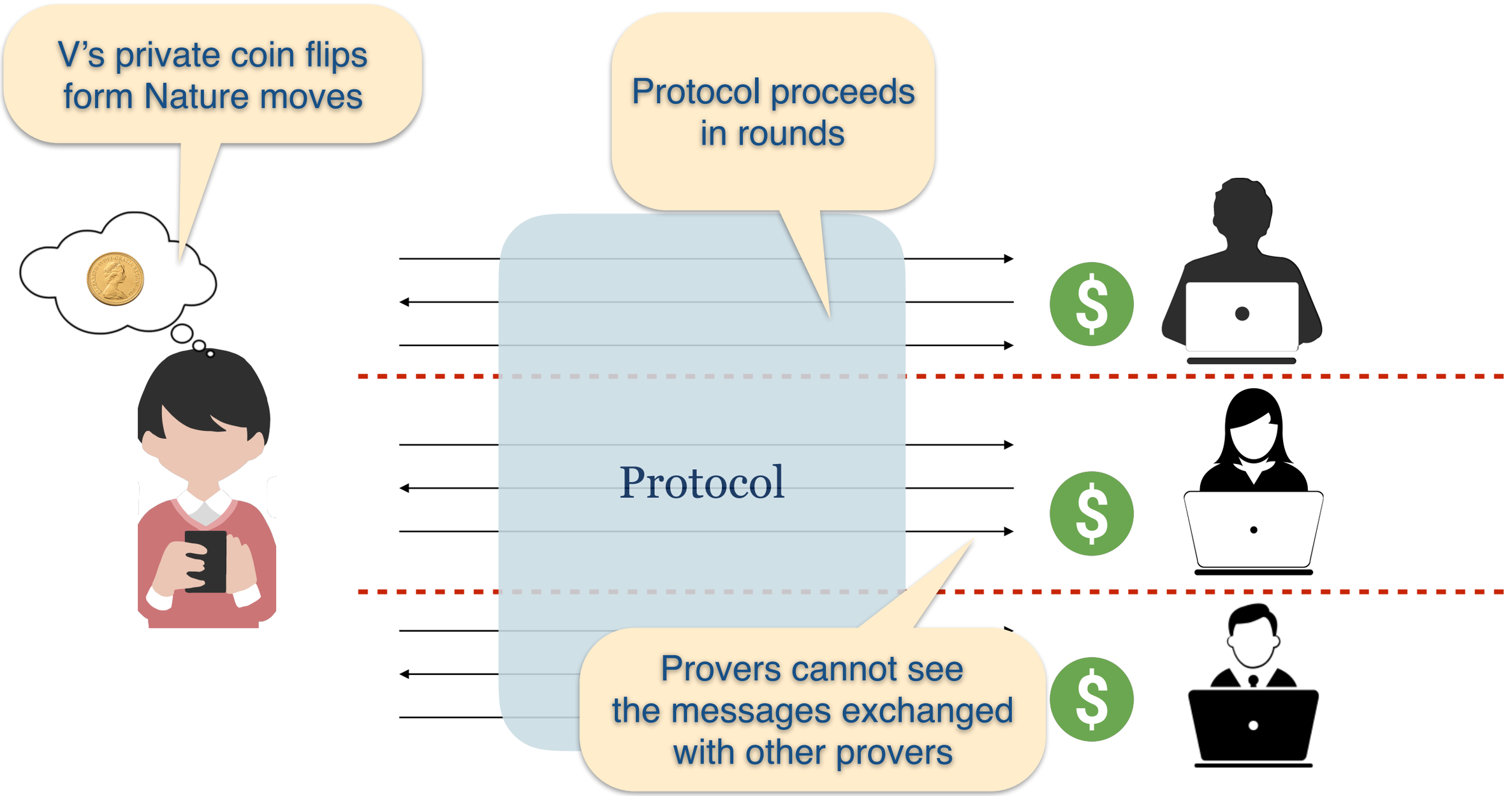
Takeaways: ncRIP for NP

- Natural and intuitive protocol
- Provers are cooperative sometimes, conflicting at other times
- Super-efficient: Verifier just has to check a single edge!
- **Constant utility gap** (~payment lost when lying)

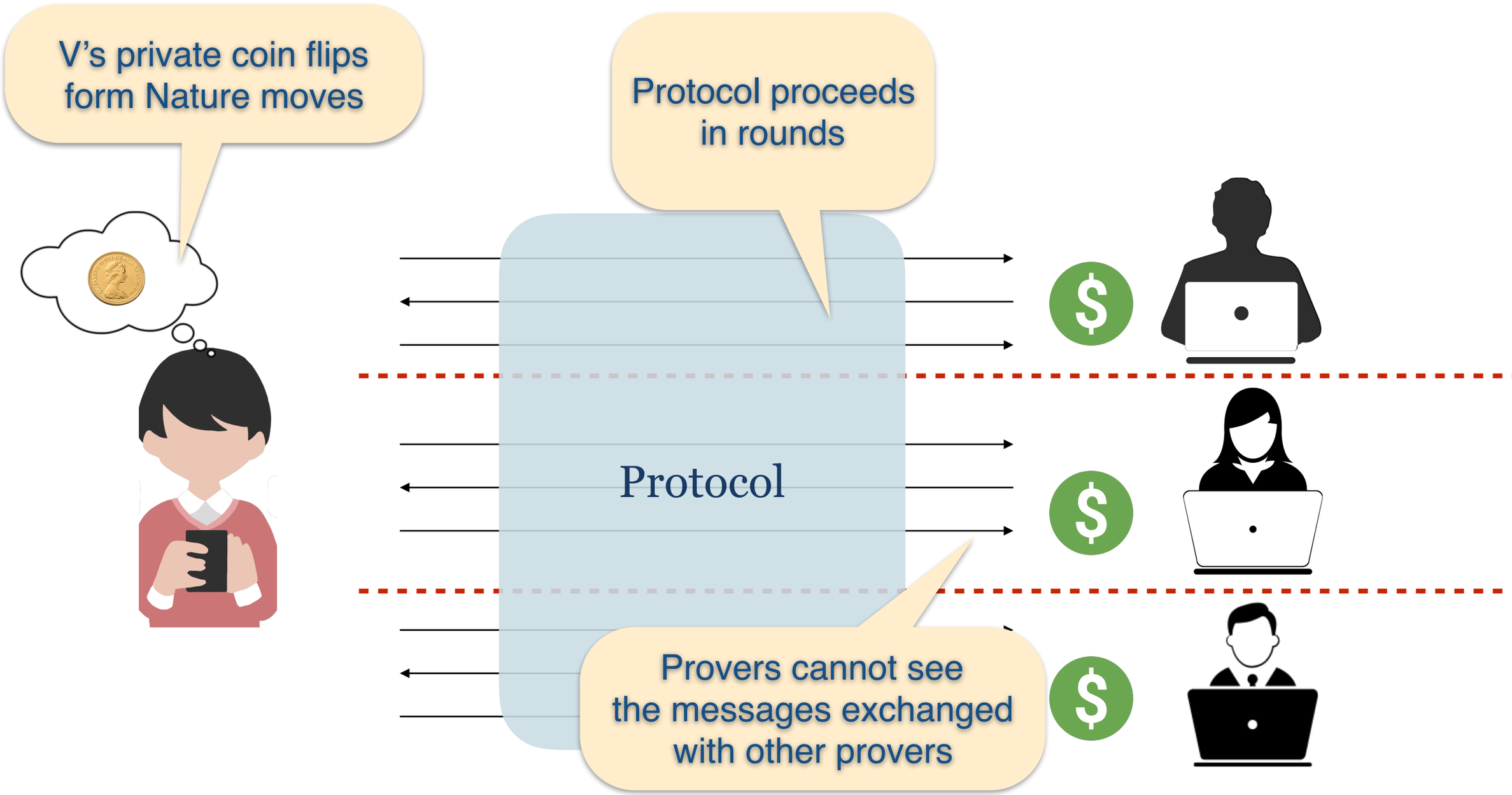


Towards the Solution Concept for ncRIP

Structure of the Game?

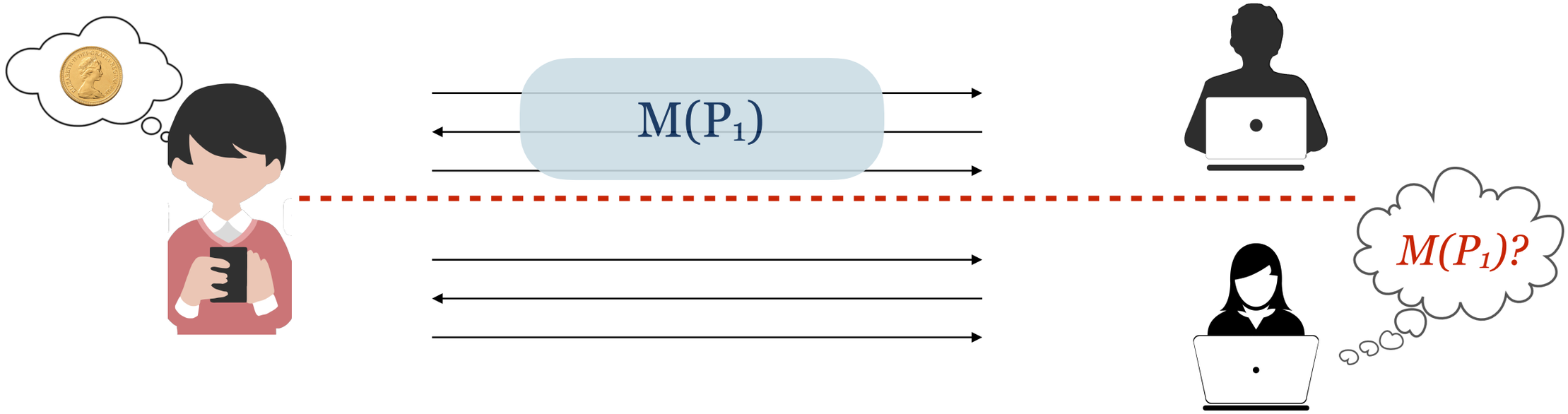


Extensive-form Game with Imperfect Information



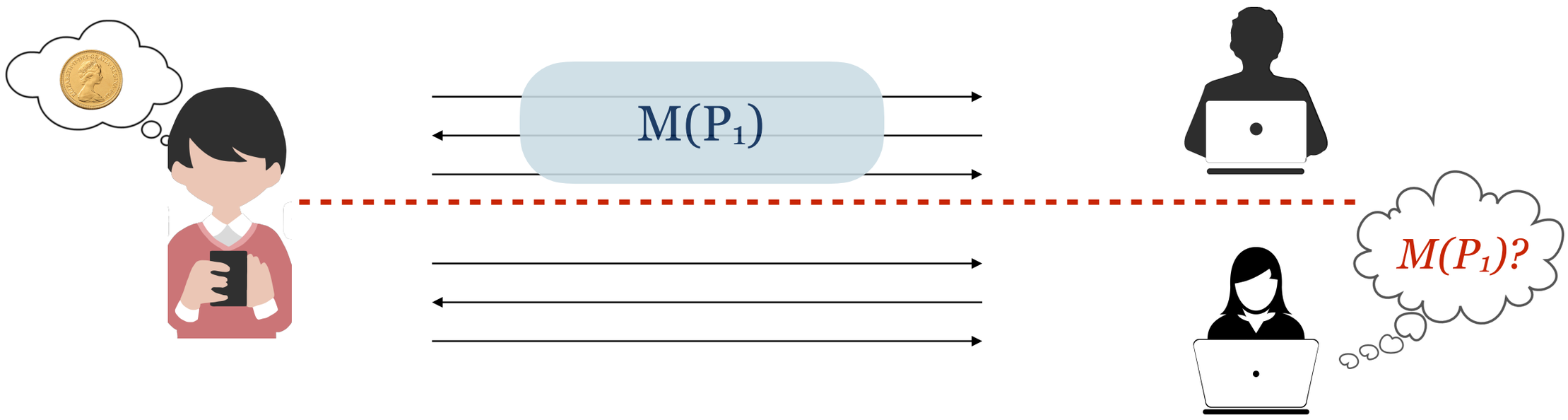
Dealing with Imperfect Information

- Players have probabilistic beliefs about history of game so far
- Along the equilibrium path, beliefs derived naturally using Bayes rule
- Off the equilibrium path ("unreachable information sets")
    - Different solution concepts in game theory treat it differently



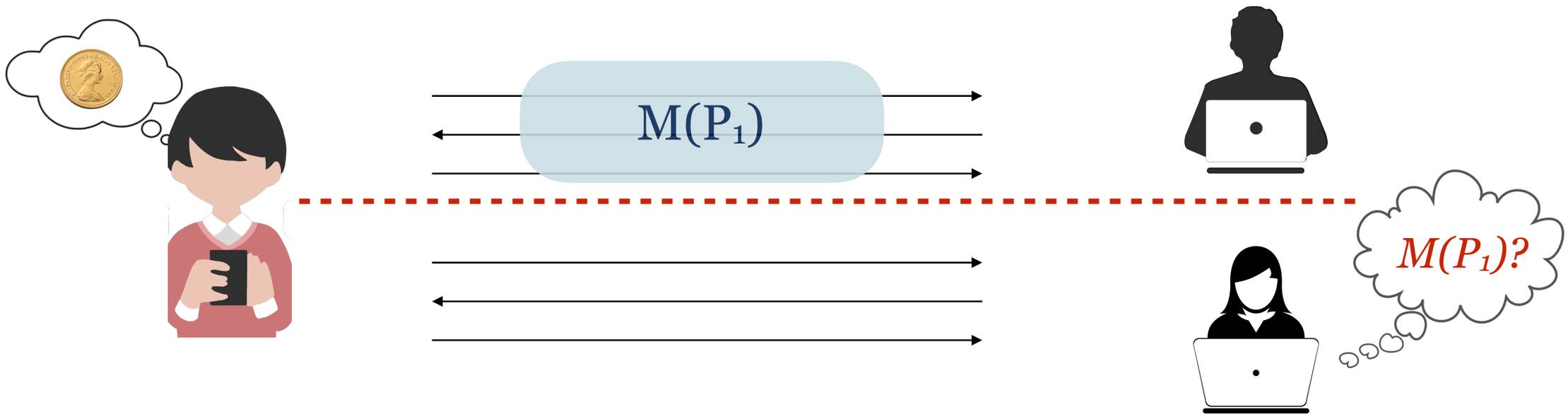
Dealing with Imperfect Information

- Players have probabilistic beliefs about history of game so far
- Along the equilibrium path, beliefs derived naturally using Bayes rule
- Off the equilibrium path ("unreachable information sets")
    - Different solution concepts in game theory treat it differently
    - **Sequential equilibrium:** external "belief system" that is consistent
    - **Trembling-hand:** ε probability to all unreachable information sets



Imperfect Information: Our Perspective

- We're using the provers to solve computational problems
- Mechanism-design goal:
    - **Giving correct answer is best response** (regardless of beliefs)
- Want beliefs at unreachable information sets to be irrelevant
- Beliefs caused by Nature moves still captured in the standard way



Strong Sequential Equilibrium (SSE)

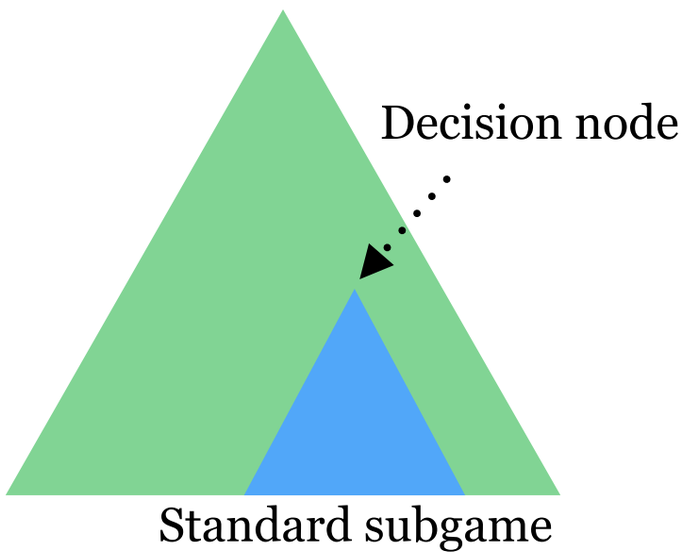
- Refinement of sequential equilibrium
- Same as sequential equilibrium at reachable information sets
- At unreachable information sets:
    - Players have a single best response (to any beliefs)
    - In ncRIP, this is usually the “correct answer”

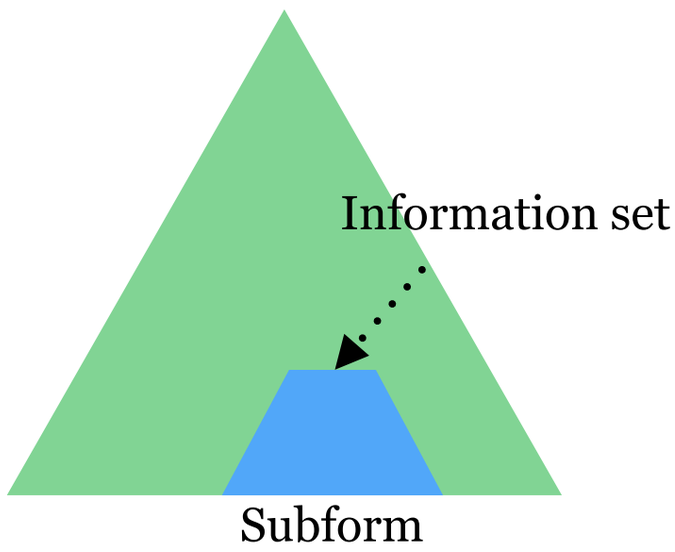
Of course not every extensive-form game will have an SSE! But we want to design mechanisms that can enforce the strong requirements

Solution Concept for ncRIP: Dominant SSE

- Apply further refinements to SSE: **dominant SSE**
    - Take max-version to eliminate dominated equilibria
    - Similar to subgame-perfection, also eliminate equilibria that are weakly dominated within “subgames”

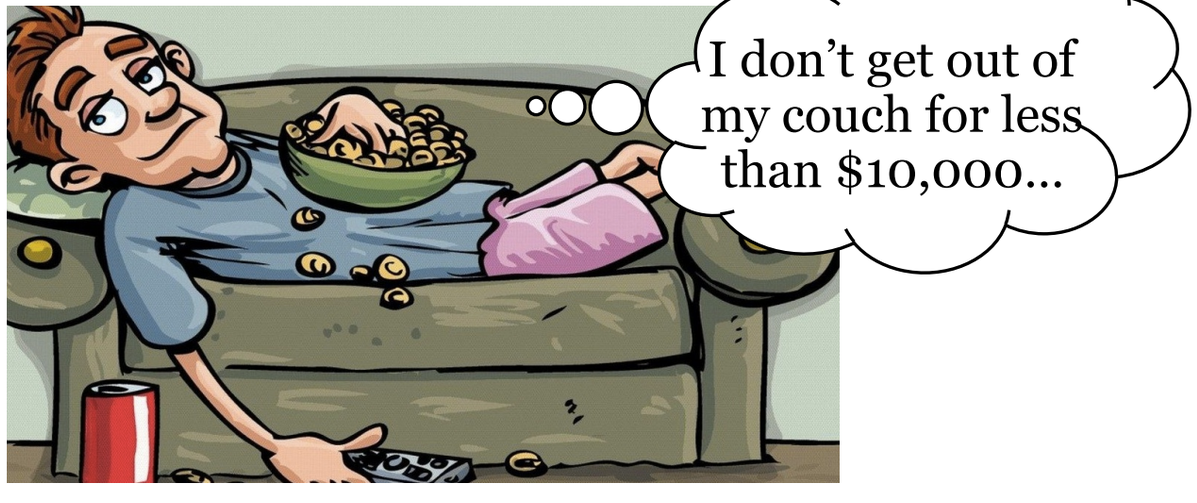
- A protocol is a ncRIP if there exists a dominant SSE and under all such equilibria V gets the correct answer to the decision problem





“Soundness Guarantee”: Utility Gap

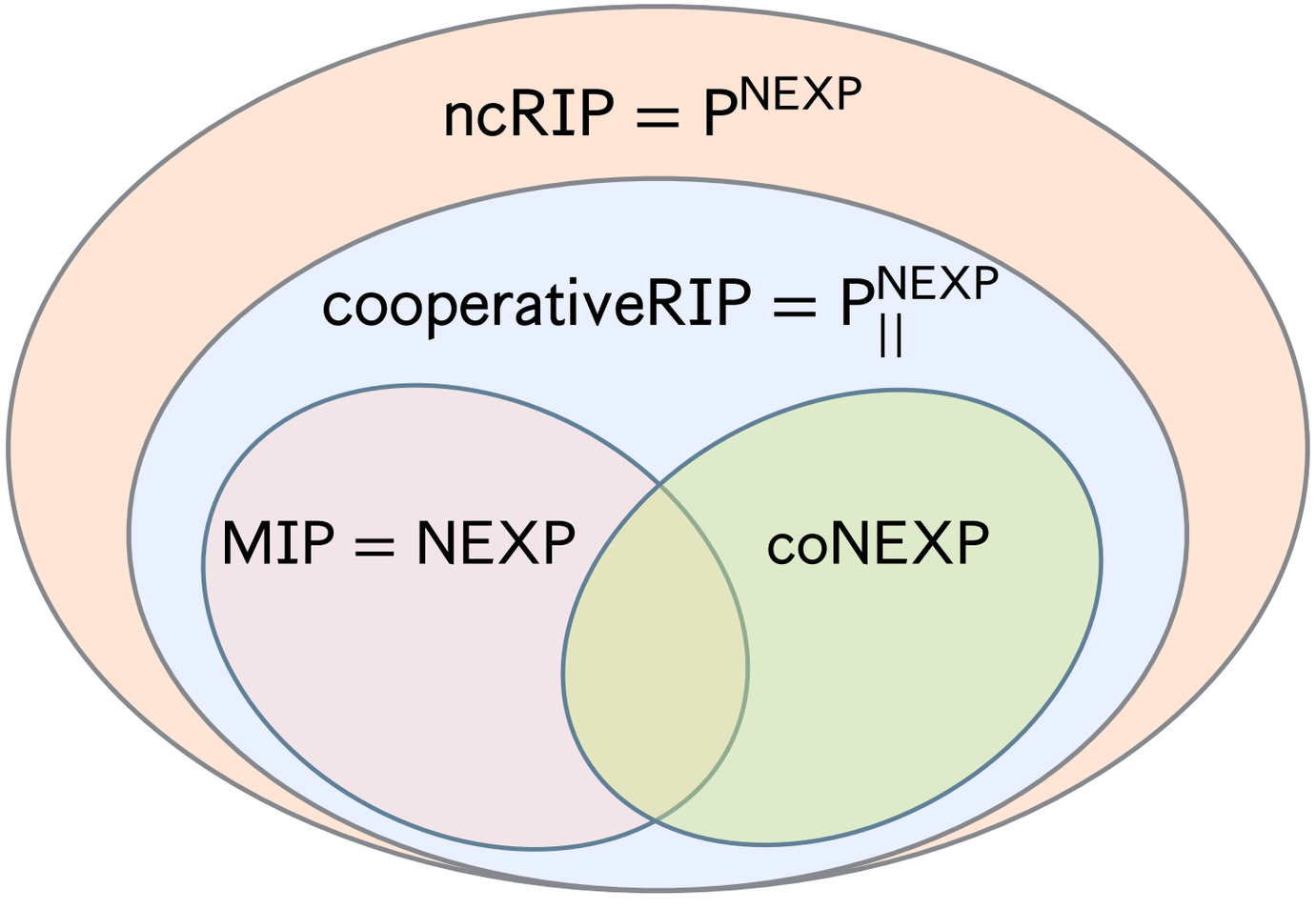
- Way to model “bounded rationality” of provers
- **Utility gap** of $g(n)$ means provers lose at least $1/g(n)$ on lying
- If s^* is optimal (truthful) and s' is a deviation then,
    - $u(x; s^*) > u(x; s') + 1/g(n)$
- Smaller the gap $g(n)$, better the guarantee of the protocol!
    - When $g(n) = O(\text{poly}(n))$, can be amplified by repetition



1. Analogous to the gap between completeness and soundness in IPs
2. poly(n) sufficient to amplify gap

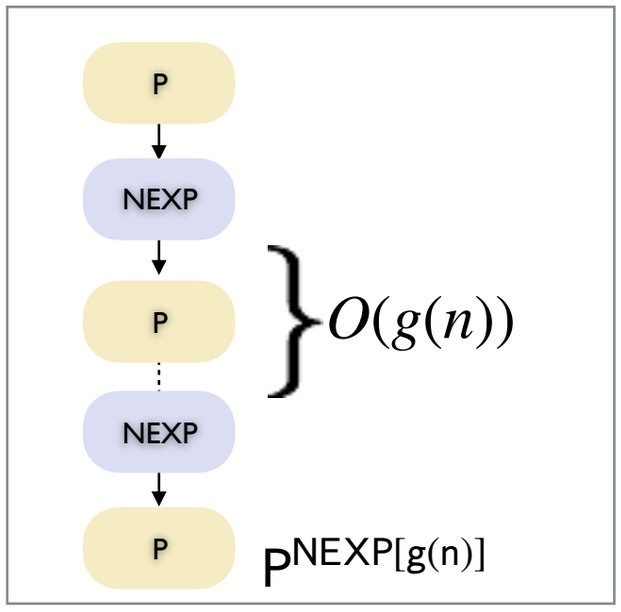
Our Main Result

The power of non-cooperative rational proofs (with **polynomial utility gap**) is the same as a polynomial-time Turing machine with **adaptive queries** to an NEXP oracle.



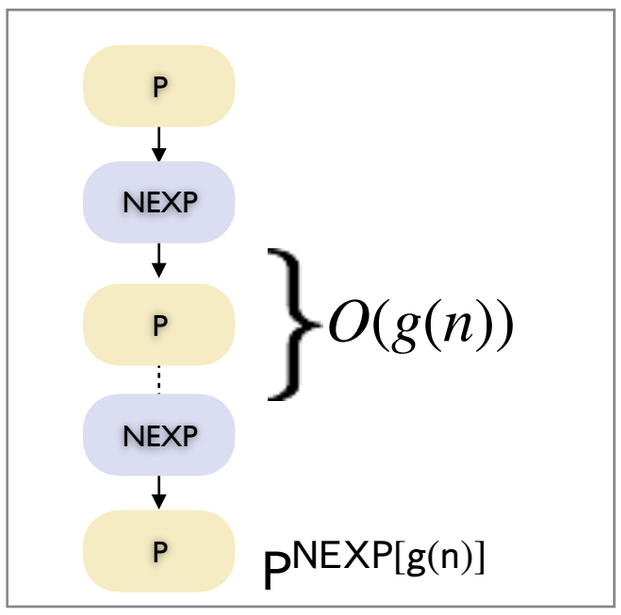
Other Results and Implications

- For any polynomially bounded utility-gap function $g(n)$
  - $g(n)$-gap-**ncRIP** $=$ $\mathsf{P}^{\mathsf{NEXP}[g(n)]}$
  - $g(n)$-gap-**coRIP** $=$ $\mathsf{P}^{\mathsf{NEXP}[g(n)]}_{||}$ (previous work)



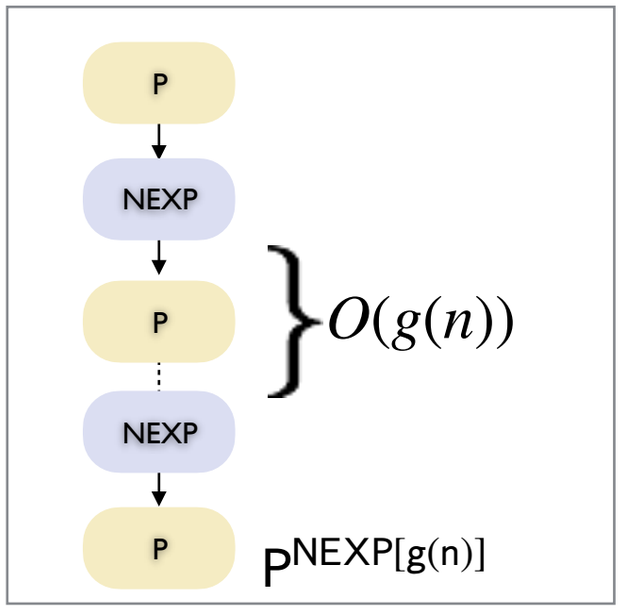
Other Results and Implications

- For any polynomially bounded utility-gap function $g(n)$
  - $g(n)$-gap-**ncRIP** $= \mathsf{P}^{\mathsf{NEXP[g(n)]}}$
  - $g(n)$-gap-**coRIP** $= \mathsf{P}^{\mathsf{NEXP[g(n)]}}_{||}$ (previous work)
- Take away: Non-cooperative provers can be used to handle "adaptive queries", while cooperative provers cannot



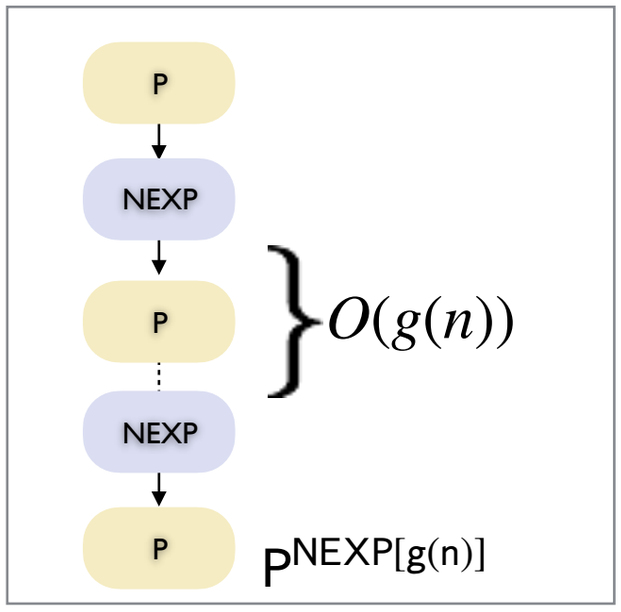
Other Results and Implications

- For any polynomially bounded utility-gap function $g(n)$
  - $g(n)$-gap-**ncRIP** $= \mathsf{P}^{\mathsf{NEXP[g(n)]}}$
  - $g(n)$-gap-**coRIP** $= \mathsf{P}^{\mathsf{NEXP[g(n)]}}_{||}$ (previous work)
- Take away: Non-cooperative provers can be used to handle "adaptive queries", while cooperative provers cannot
- Our understanding of **adaptive-oracle complexity classes** is limited
  - This game-theoretic characterization can help!



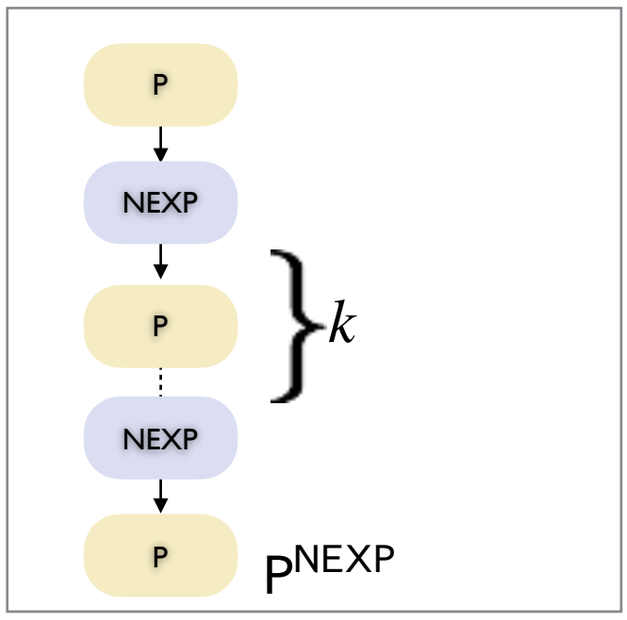
Summary and Takeaways

- For any polynomially bounded utility-gap function $g(n)$
  - $g(n)$-gap-**ncRIP** = $\mathsf{P}^{\mathsf{NEXP}[g(n)]}$
  - $g(n)$-gap-**coRIP** = $\mathsf{P}^{\mathsf{NEXP}[g(n)]}_{||}$ (previous work)
- Take away: Non-cooperative provers can be used to handle "adaptive queries", while cooperative provers cannot
- Our understanding of **adaptive-oracle complexity classes** is limited
  - This game-theoretic characterization can help!
- Beyond Nash for **extensive-form mechanisms**
  - SSE tailored for "verifiable" mechanisms



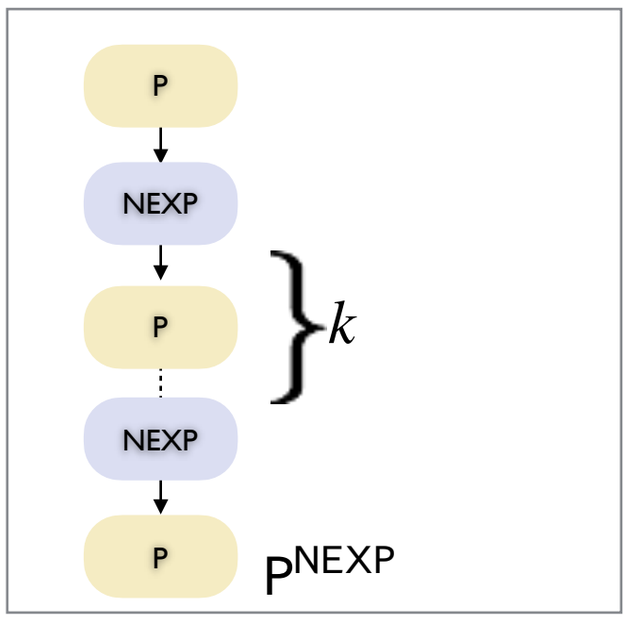
Lower Bound: $\mathsf{P}^{\mathsf{NEXP}} \subseteq \mathsf{ncRIP}$

- Uses a ncRIP protocol for **NEXP**
- Let $L \in \mathsf{P}^{\mathsf{NEXP}}$ and M^O be the oracle TM for L
- V asks P_1 for answer bit c (indicating whether $x \in L$ or not) and answer $o_1, o_2, \ldots, o_k$ to all oracle queries



Lower Bound: $\mathsf{P}^{\mathsf{NEXP}} \subseteq$ ncRIP

- Uses a ncRIP protocol for **NEXP**
- Let $L \in \mathsf{P}^{\mathsf{NEXP}}$ and M^O be the oracle TM for L
- V asks P_1 for answer bit c (indicating whether $x \in L$ or not) and answer $o_1, o_2, \ldots, o_k$ to all oracle queries
- V uses $x, o_1, \ldots, o_k$ to simulate machine M^O in poly-time
  - V uses P_2, P_3 to check one of the **NEXP** oracle queries
  - P_1 gets \$1 if their answers match, else \$0
- Any dominant SSE strategy leads to correct answer



Upper Bound: ncRIP $\subseteq$ P^{NEXP}

- Challenging: exponential search space for poly-time machine
    - **(Exponential game tree)** Polynomial-time V imposes exponential-size probability distribution of Nature moves
    - **(Exponential strategy space)** Need to search through exponentially many strategies to one that is a dominant SSEs

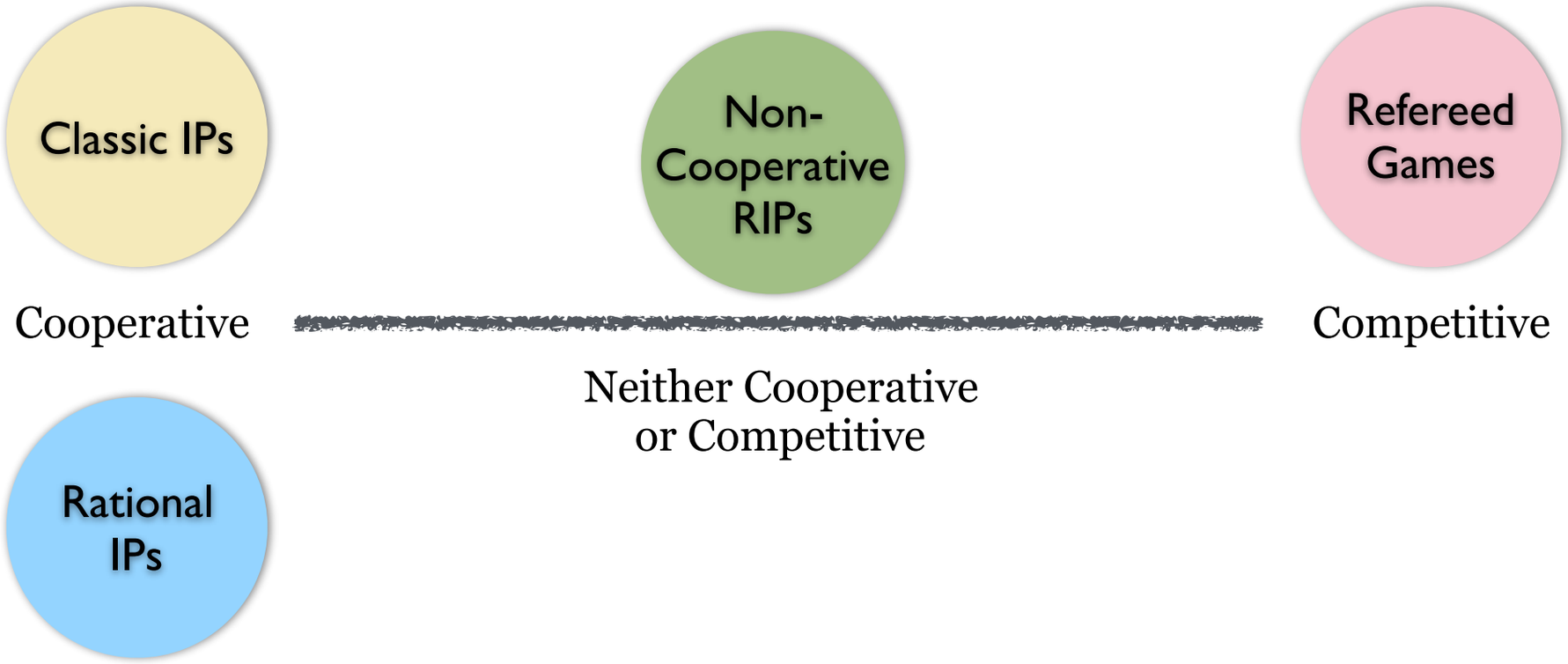
Upper Bound: ncRIP $\subseteq$ P^{NEXP}

- Challenging: exponential search space for poly-time machine
    - **(Exponential game tree)** Polynomial-time V imposes exponential-size probability distribution of Nature moves
    - **(Exponential strategy space)** Need to search through exponentially many strategies to one that is a dominant SSEs
- **(Oracle can only help so much)** An NEXP oracle cannot directly verify dominant SSEs

Upper Bound: ncRIP $\subseteq$ P^{NEXP}

- Challenging: exponential search space for poly-time machine
    - **(Exponential game tree)** Polynomial-time V imposes exponential-size probability distribution of Nature moves
    - **(Exponential strategy space)** Need to search through exponentially many strategies to one that is a dominant SSEs
- **(Oracle can only help so much)** An NEXP oracle cannot directly verify dominant SSEs
- Proof Idea. **(Careful pruning)**
    - Prune the Nature moves of V , while preserving all other properties (dominant SSE, gap, etc)
    - Prune strategy-search space based on utility gap and properties of dominant SSEs

Conclusion



- Leveraging the space between cooperative and competitive incentives
- Non-cooperative provers lead to simple & efficient protocols
- Opens up many new directions: scaled-down proofs and arguments
- SSE: independent interest as a solution concept for mechanism design