Automating Game Reasoning for Decentralized Protocols

Laura Kovács
TU Wien

Informatics 20 YEARS

for(syte) erc

Automating Game Reasoning for Decentralized Protocols

Laura Kovács
TU Wien

Joint work with





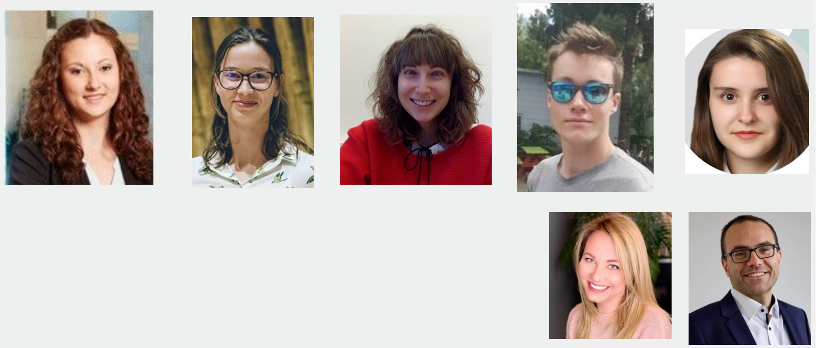
TU WIEN Informatics 20 YEARS

for(syte) erc

Automating Game Reasoning for Decentralized Protocols

Laura Kovács
TU Wien

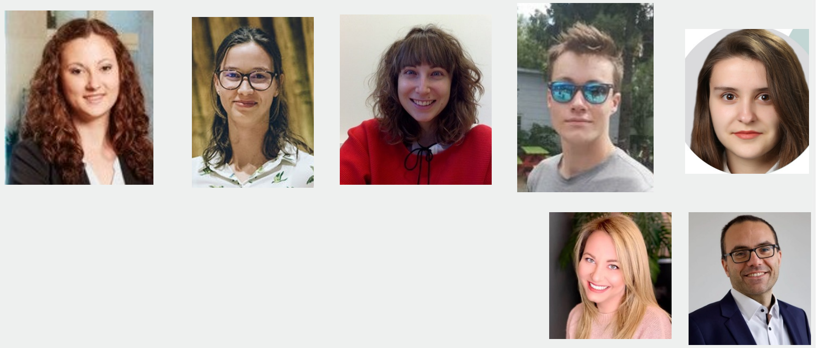
Joint work with



Automating Game Reasoning for Decentralized Protocols

Laura Kovács
TU Wien

Joint work with



Decentralized Protocols – Complementary Views of Security

Can a malicious person steal my data?

Cryptographic Security

Do I want to share my data?

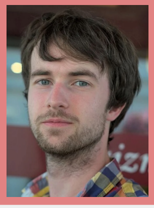
Game-Theoretic Security

Decentralized Protocols – Complementary Views of Security

Can a malicious person steal my data?

Cryptographic Security

CryptoVampire (S&P 2024)



Do I want to share my data?

Game-Theoretic Security

This talk

Decentralized Protocols – Game-Theoretic Security Analysis

Do I want to share my data?

Game-Theoretic Security

Decentralized Protocols – Game-Theoretic Security Analysis

Game-Theoretic Security

Decentralized Protocols – Game-Theoretic Security Analysis

What are my **economic** gains/incentives doing so?

Do I want to share my data?

Acting **honestly** should be the best.

Game-Theoretic Security

Decentralized Protocols – Game-Theoretic Security Analysis

What are my **economic** gains/incentives doing so?

Do I want to share my data?

Deviation from a protocol is not rational.

Acting **honestly** should be the best.

Honest players are never harmed.

Game-Theoretic Security

Decentralized Protocols – Game-Theoretic Security Analysis

Incentive Competability

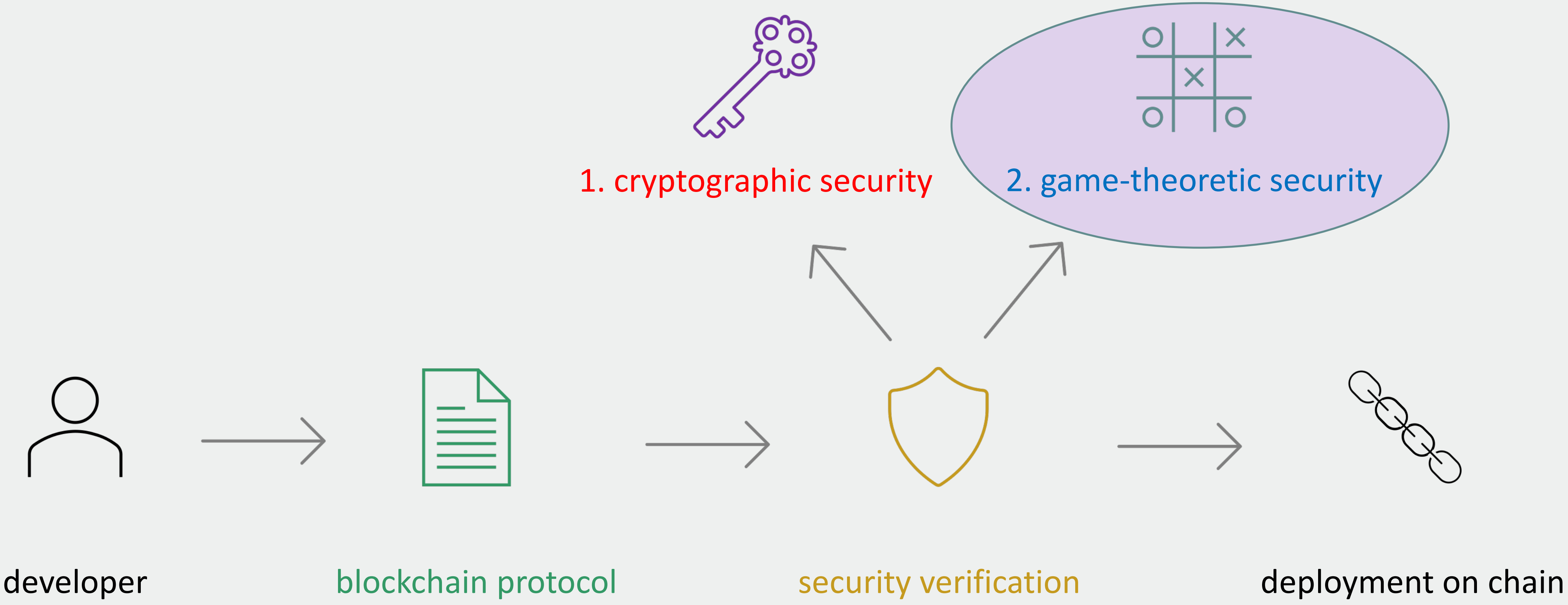
Deviation from
a protocol
is not rational.

Game-Theoretic Security

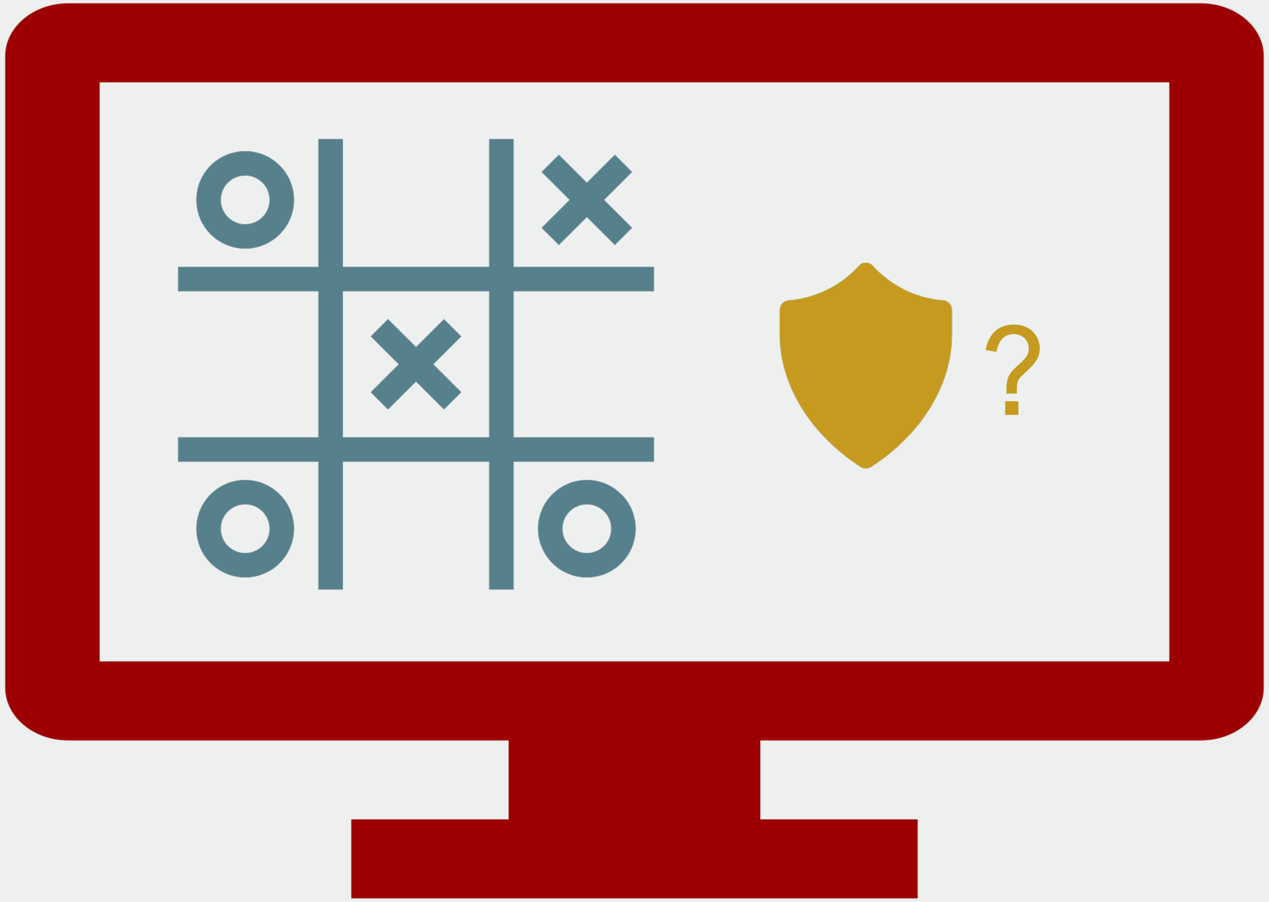
Byzantine Fault Tolerance

Honest players
are never
harmed.

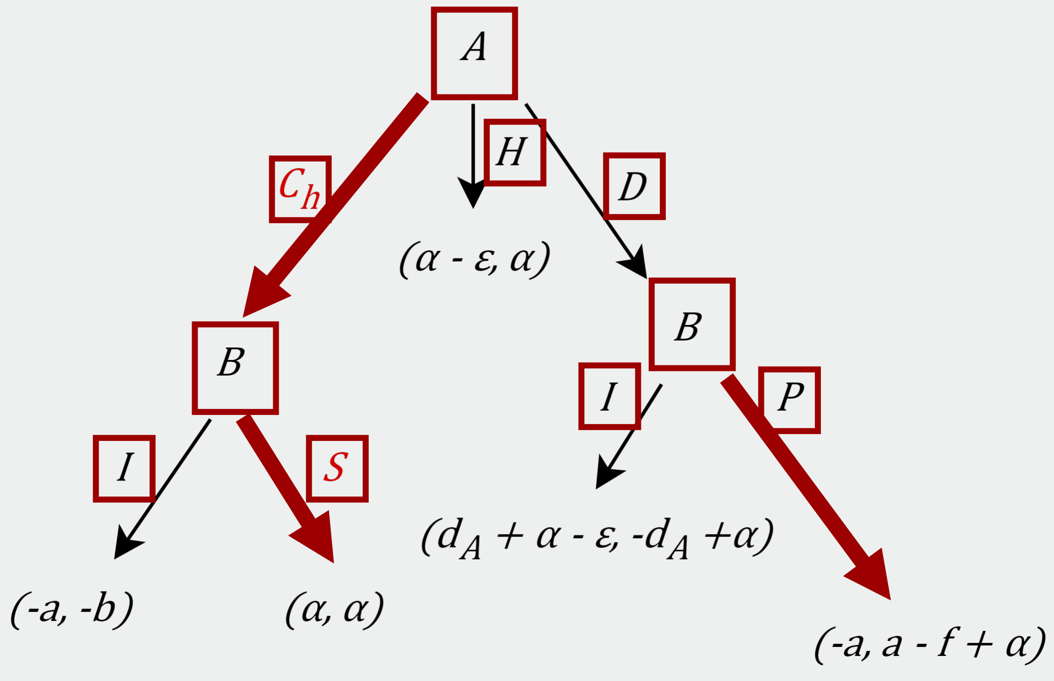
Our Vision: Automated Game-Theoretic Security Reasoning



Automated Game-Theoretic Security Reasoning



Game-Theoretic Models (CSF 2023)

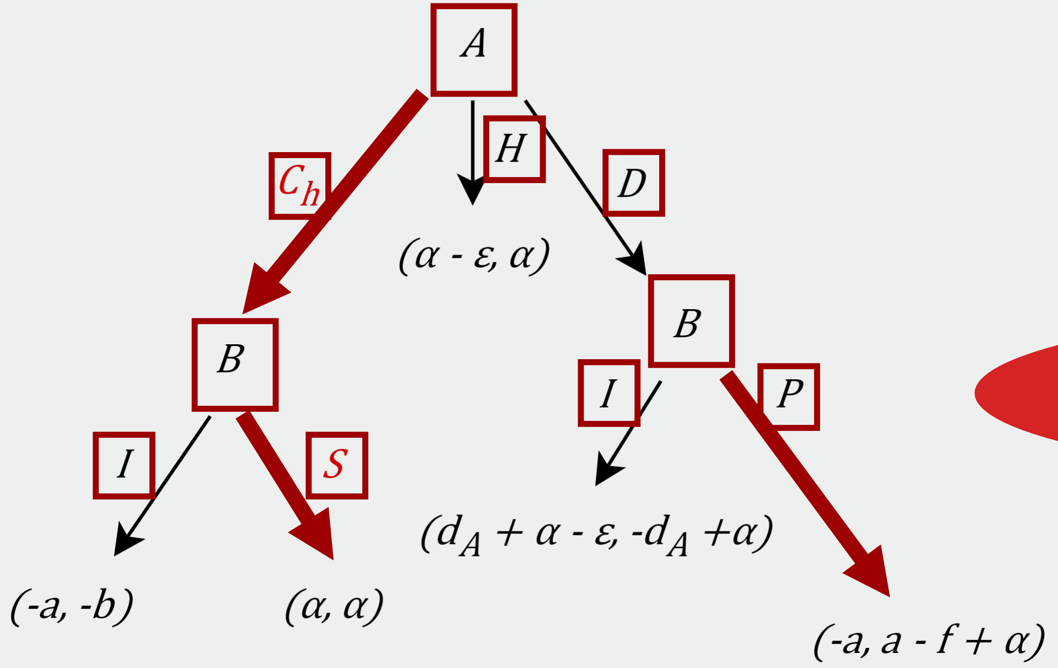


Simplified Closing Game (Bitcoin)

players: A and B

actions: C_h, H, D, I, S and P

Game-Theoretic Models (CSF 2023)



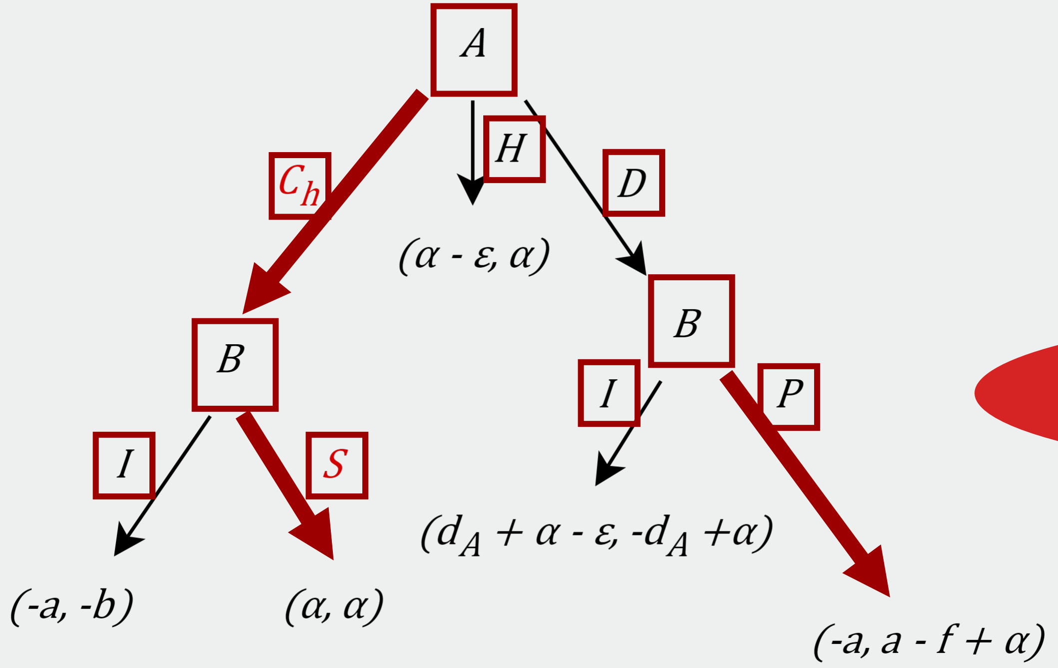
Simplified Closing Game (Bitcoin)

players: A and B

actions: C_h, H, D, I, S and P

Close collaboratively and honestly: yielding fair split

Game-Theoretic Models (CSF 2023)



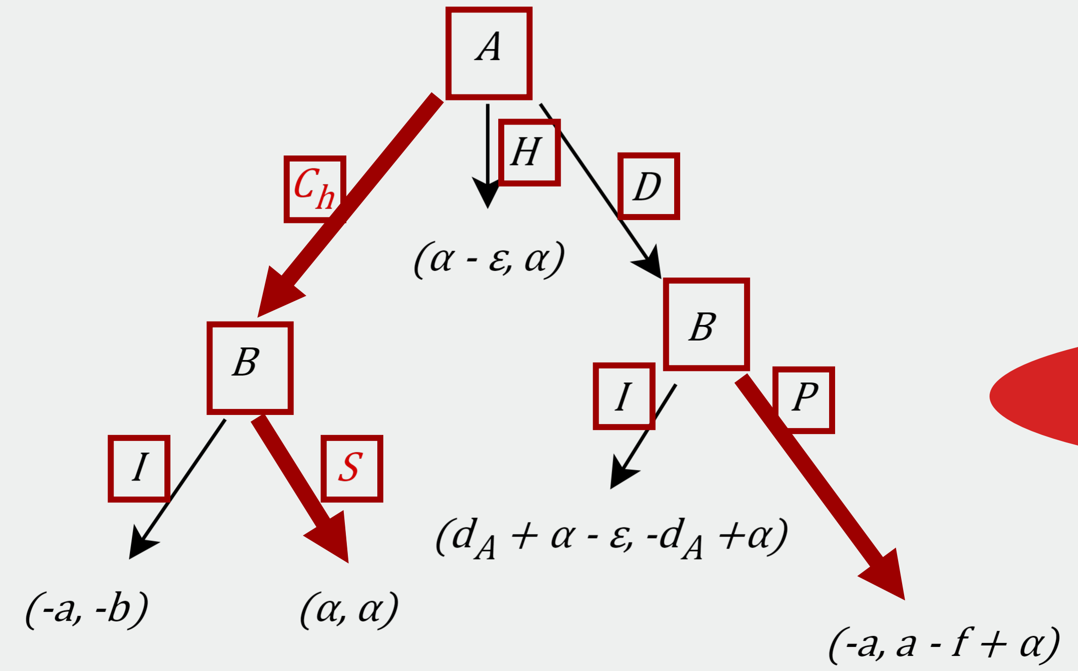
Simplified Closing Game (Bitcoin)

players: A and B

actions: C_h, H, D, I, S and P

Close unilaterally and honestly,
do not consider previous moves.

Game-Theoretic Models (CSF 2023)



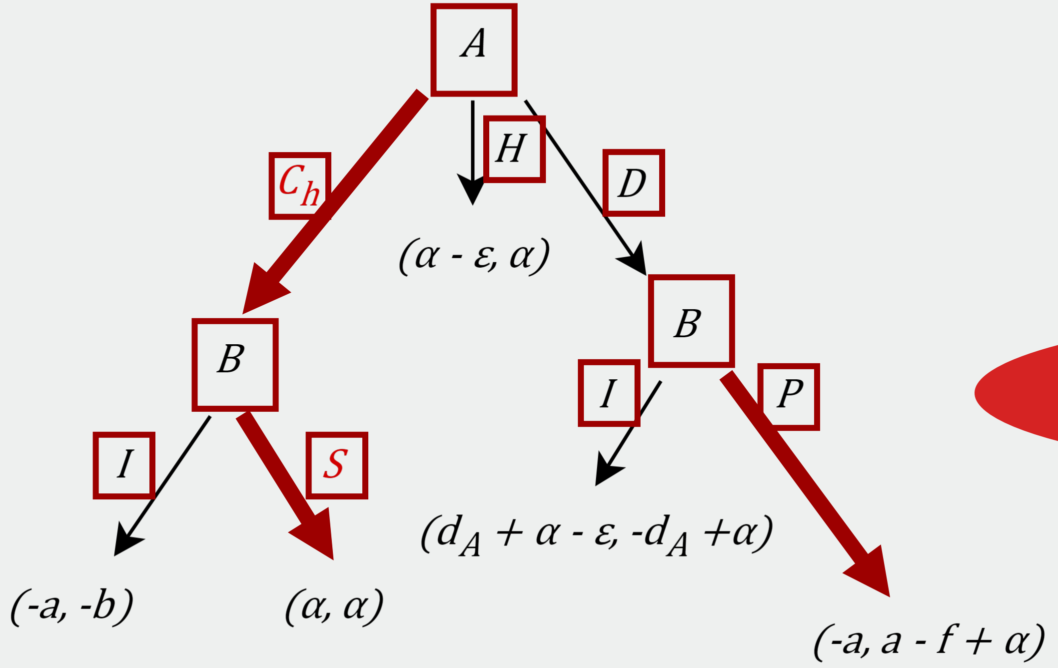
Simplified Closing Game (Bitcoin)

players: A and B

actions: C_h, H, D, I, S and P

Close unilaterally and dishonestly, with profits d_A for A and d_B for B

Game-Theoretic Models (CSF 2023)



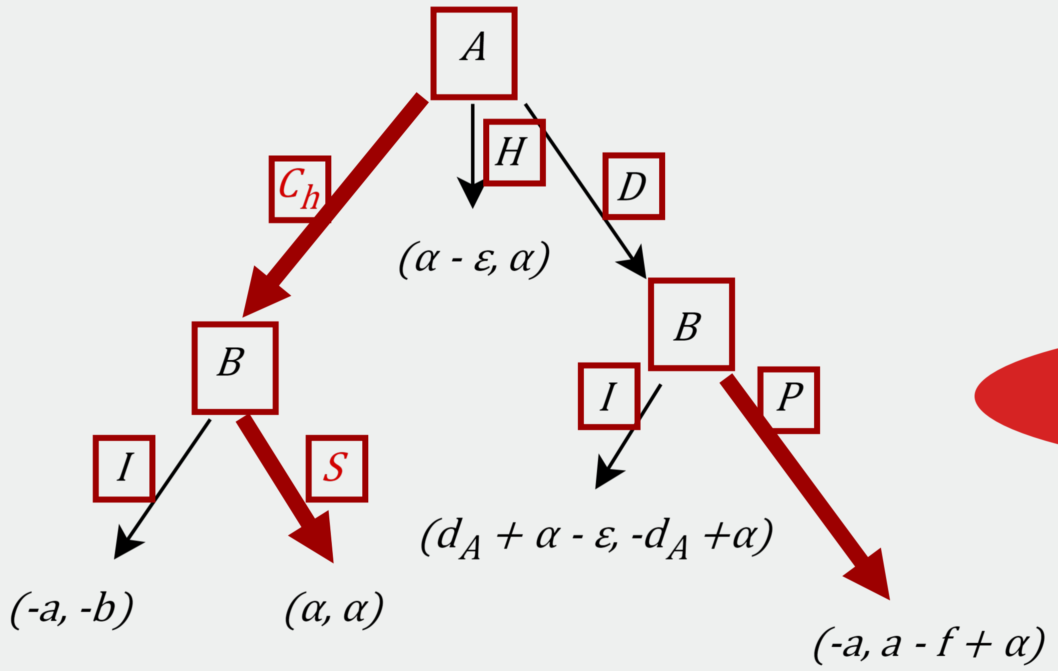
Simplified Closing Game (Bitcoin)

players: A and B

actions: C_h, H, D, I, S and P

Ignore previous action and do nothing

Game-Theoretic Models (CSF 2023)



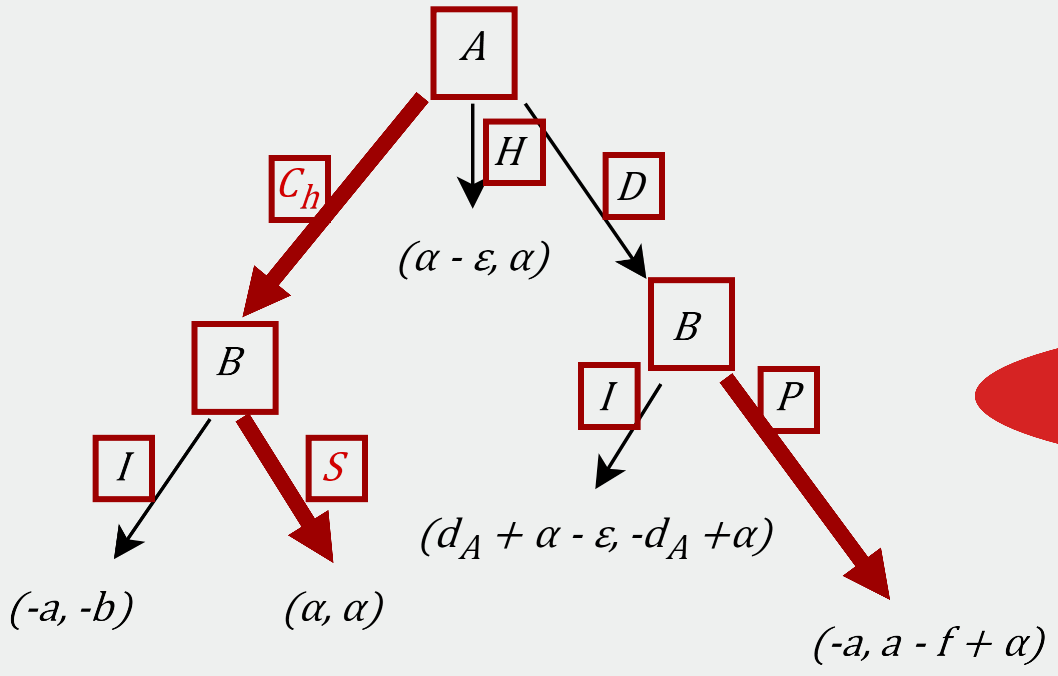
Simplified Closing Game (Bitcoin)

players: A and B

actions: C_h, H, D, I, S and P

Sign collaborative closing of the other player

Game-Theoretic Models (CSF 2023)



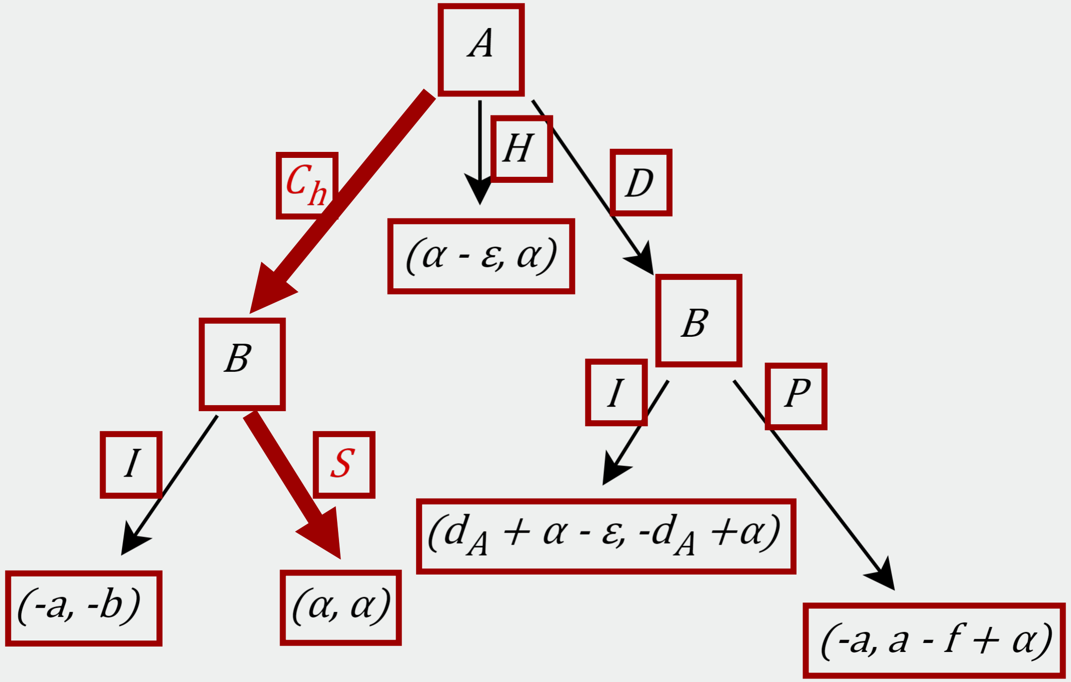
Simplified Closing Game (Bitcoin)

players: A and B

actions: C_h, H, D, I, S and P

Prove other player tried to close dishonestly

Game-Theoretic Models (CSF 2023)



Simplified Closing Game (Bitcoin)

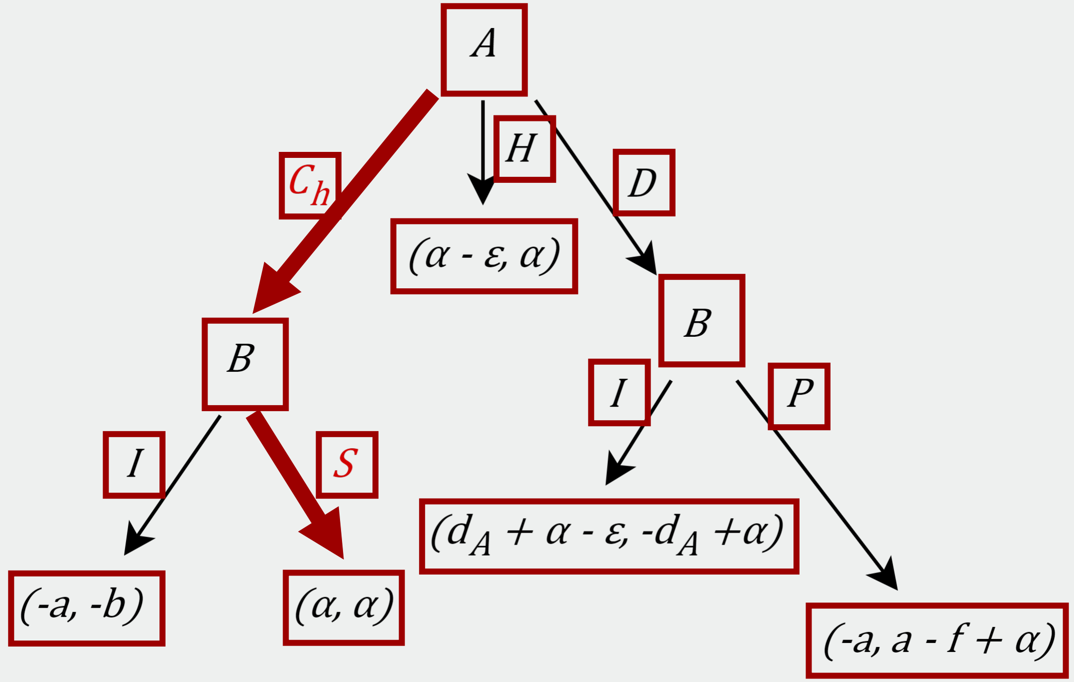
players: A and B

actions: C_h, H, D, I, S and P

utilities: (u_A, u_B), terms of reals

- benefit of closing a channel: $\boldsymbol{\alpha}$>0
- opportunity cost: $\boldsymbol{\varepsilon}$>0 (cost of closing)
- transaction fee: **f**>0

Game-Theoretic Models (CSF 2023)



Simplified Closing Game (Bitcoin)

players: A and B

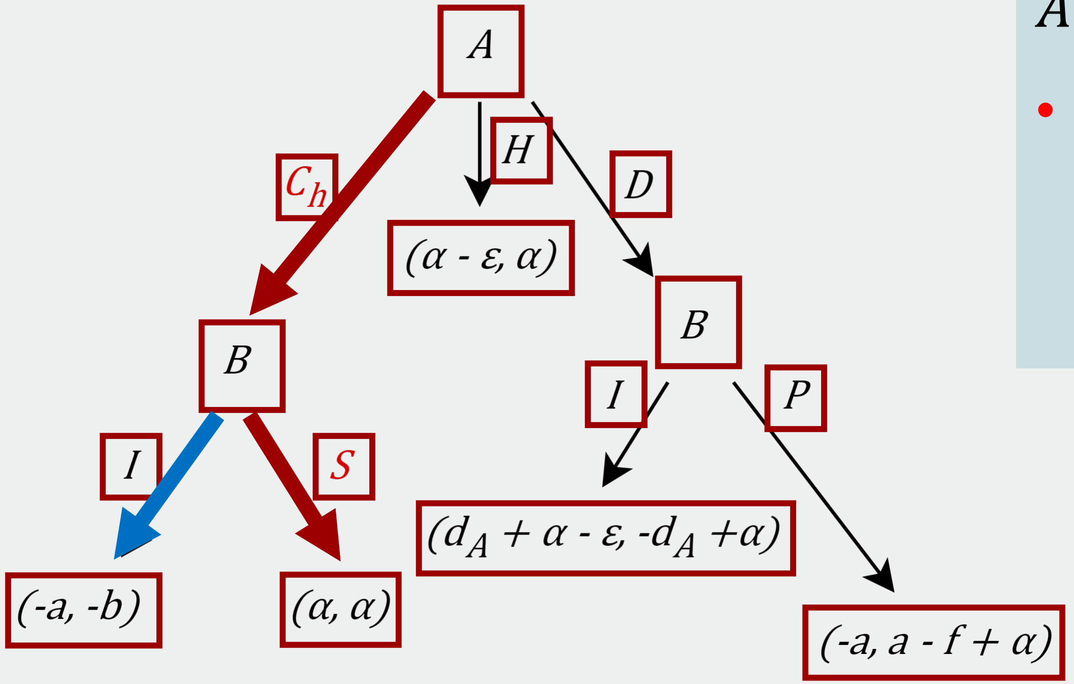
actions: C_h, H, D, I, S and P

utilities: (u_A, u_B), terms of reals

joint strategy: one action per node

honest behavior: intended scenario

Game-Theoretic Models (CSF 2023)

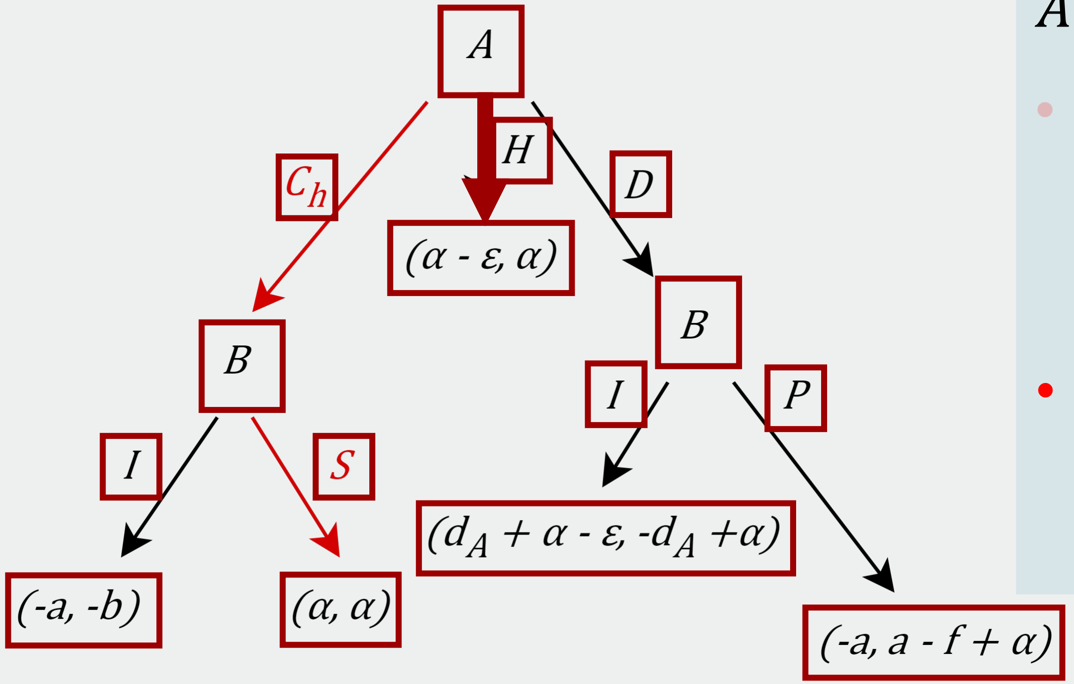


Simplified Closing Game (Bitcoin)

A chooses

- C_h: if *B* ignores (I), then funds are locked;
  if B signs (S), then both players get the closing benefit $\boldsymbol{\alpha}$.

Game-Theoretic Models (CSF 2023)

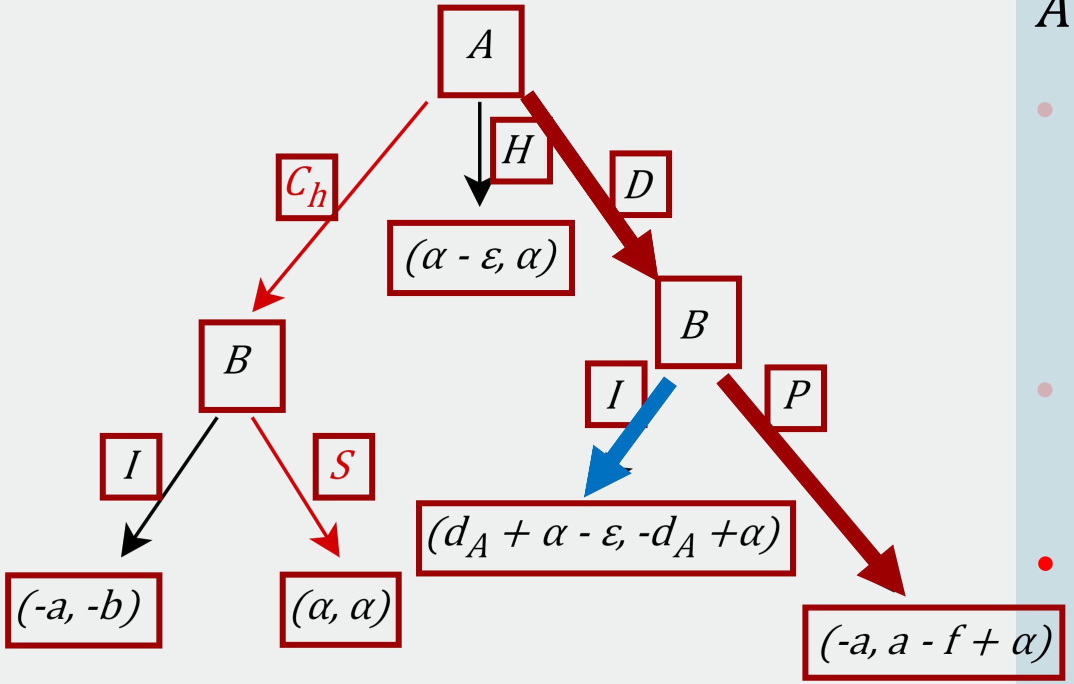


Simplified Closing Game (Bitcoin)

A chooses

- C_h: if B ignores (I), then funds are locked;
  if B signs (S), then both players get the closing benefit α.
- H: both players get benefits, but A waits for closing timeout;

Game-Theoretic Models (CSF 2023)

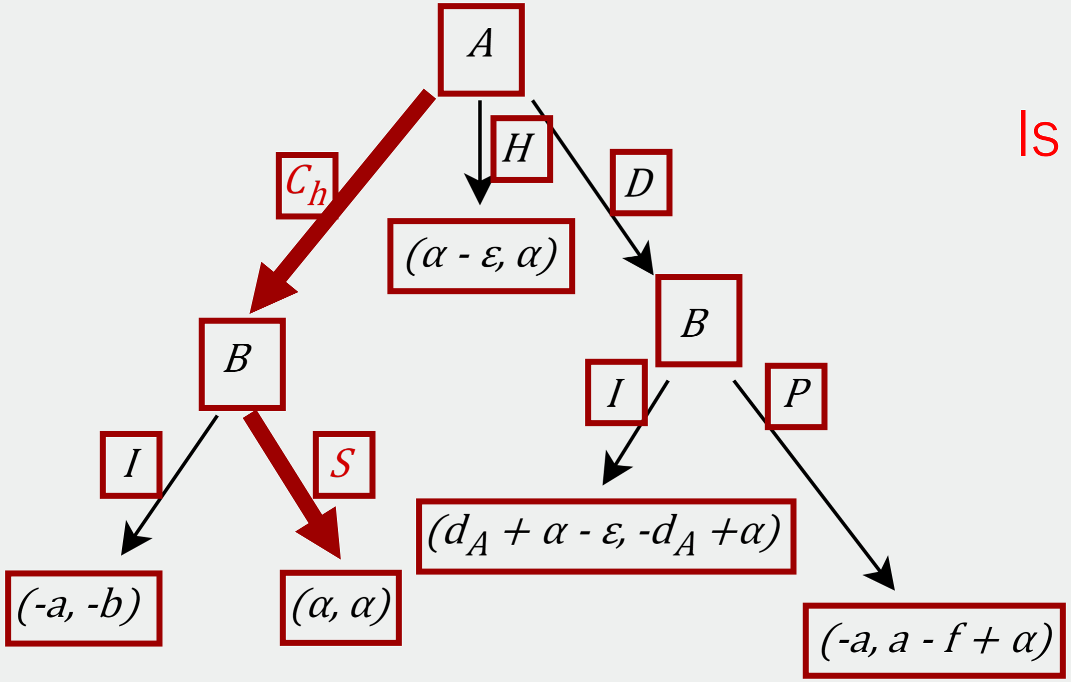


Simplified Closing Game (Bitcoin)

A chooses

- C_h: if B ignores (I), then funds are locked;
  if B signs (S), then both players get the closing benefit $\boldsymbol{\alpha}$.
- H: both players get benefits, but A waits for closing timeout;
- D: if B ignores (I), the funds of B are lost;
  if B proves (P) dishonest A, then funds A are given to B, with transaction fee f paid.

Game-Theoretic Models (CSF 2023)



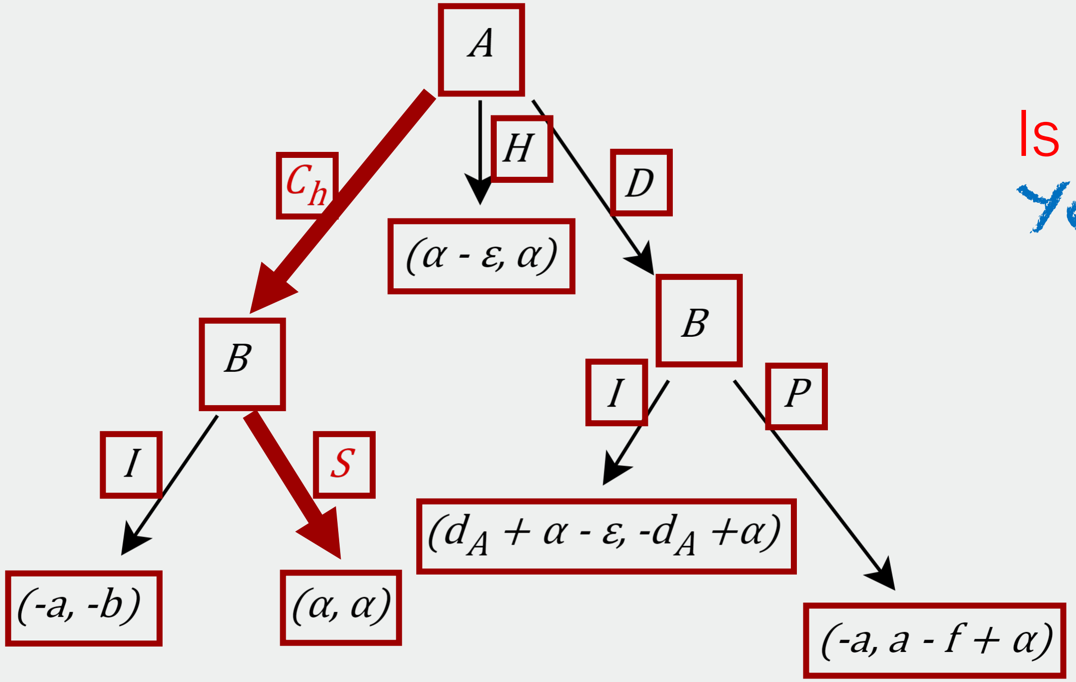
Simplified Closing Game (Bitcoin)

Is there a way to financially harm A?

Is deviating rational?

Honest behavior (C_h, S)

Game-Theoretic Models (CSF 2023)



Simplified Closing Game (Bitcoin)

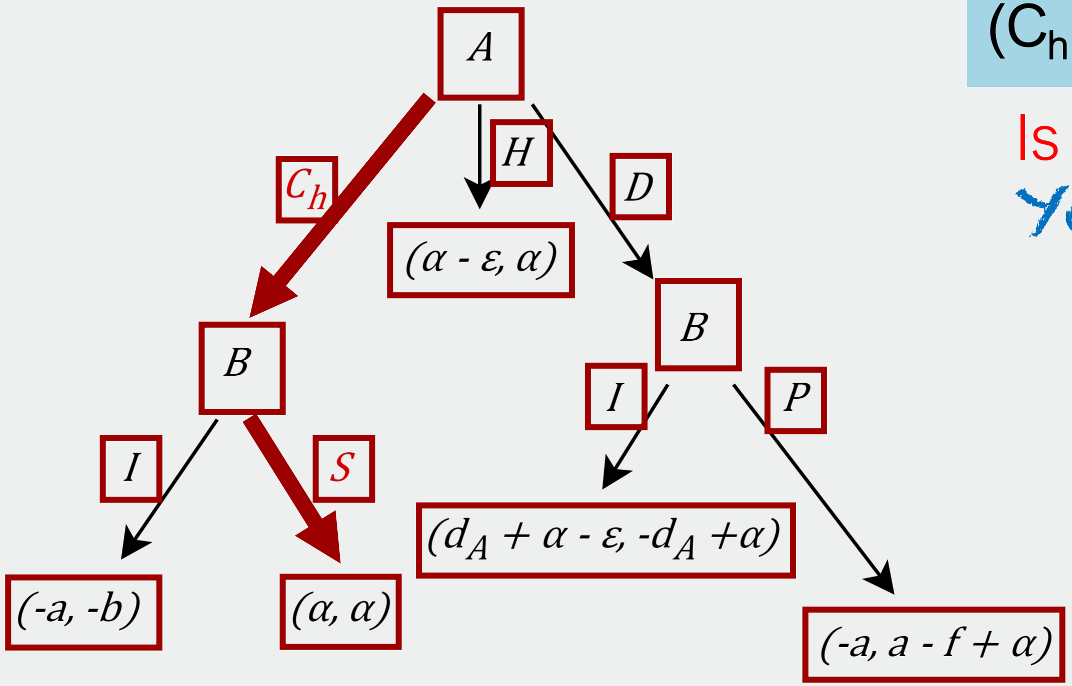
Honest behavior (C_h, S)

Is there a way to financially harm A?

Yes: (C_h, I), when $a>0$.

Is deviating rational?

Game-Theoretic Models (CSF 2023)



Simplified Closing Game (Bitcoin)

Honest behavior (C_h, S)

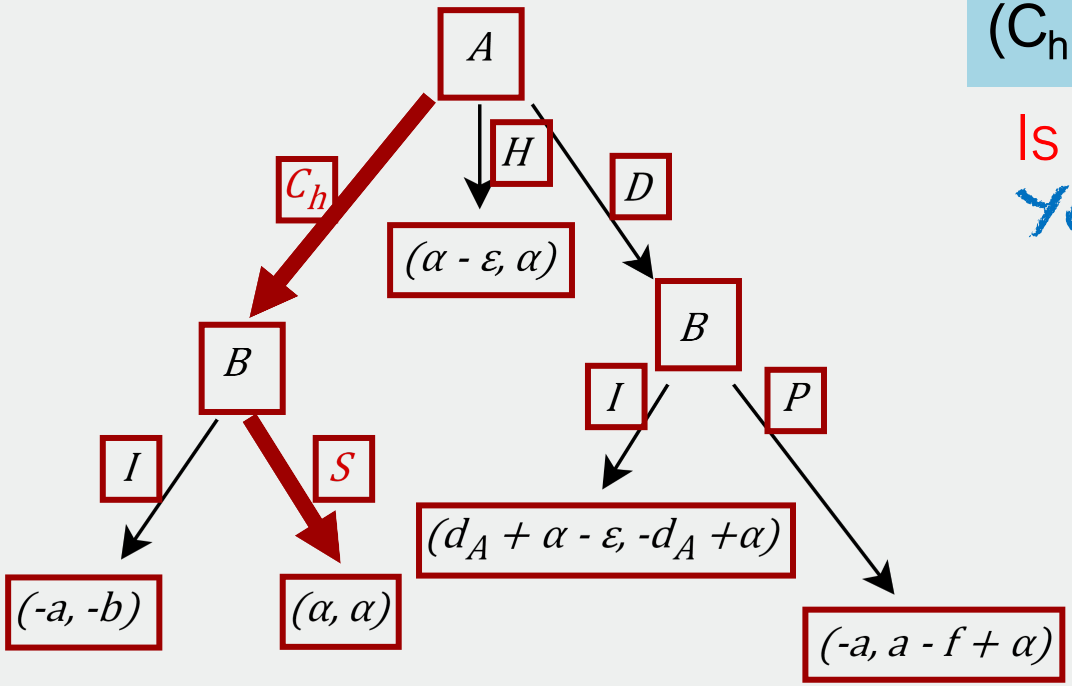
(C_h, S) is not Byzantine Fault Tolerant

Is there a way to financially harm A?

Yes: (C_h, I), when a>0.

Is deviating rational?

Game-Theoretic Models (CSF 2023)



Simplified Closing Game (Bitcoin)

Honest behavior (C_h, S)

(C_h, S) is not Byzantine Fault Tolerant ⊗

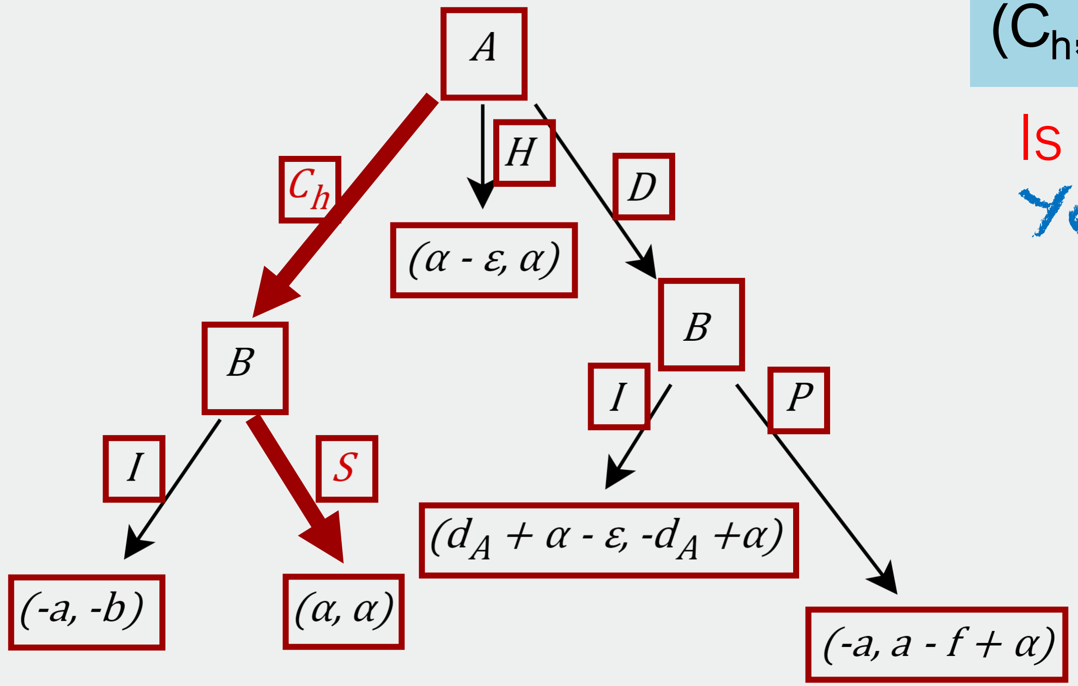
Is there a way to financially harm A?

Yes: (C_h, I), when a>0.

Is deviating from (C_h, S) rational?

No: (C_h, S) yields fair splits.

Game-Theoretic Models (CSF 2023)



Simplified Closing Game (Bitcoin)

Honest behavior (C_h, S)

(C_h, S) is not Byzantine Fault Tolerant

Is there a way to financially harm A?

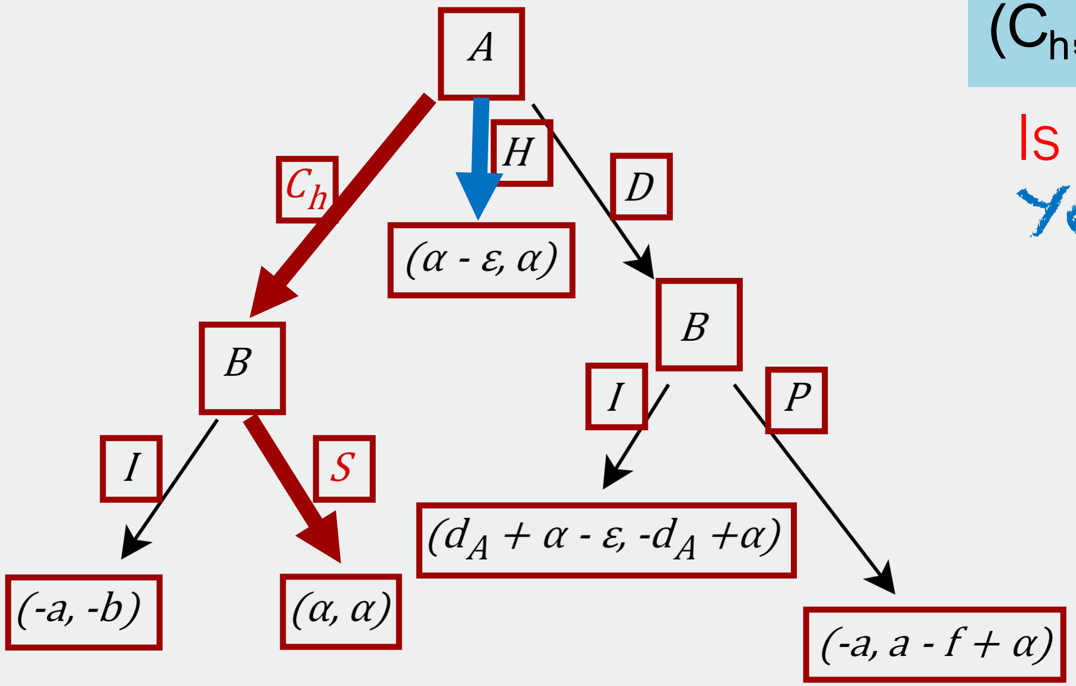
Yes: (C_h, I), when a>0.

(C_h, S) is Incentive Compatible ✓

Is deviating from (C_h, S) rational?

No: (C_h, S) yields fair splits.

Game-Theoretic Models (CSF 2023)



Simplified Closing Game (Bitcoin)

(C_h, S) is not Byzantine Fault Tolerant ⊗

Is there a way to financially harm A?

Yes: (C_h, I), when a>0.

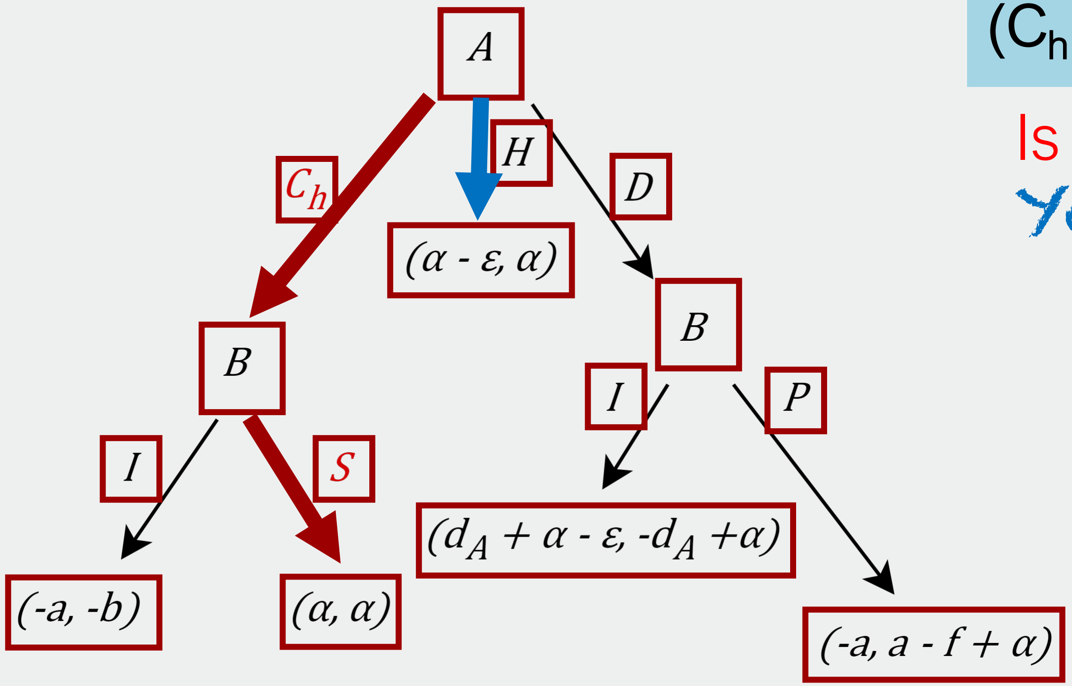
(C_h, S) is Incentive Compatible ✓

Is deviating from (C_h, S) rational?

No: (C_h, S) yields fair splits.

Is deviating from (H) rational?

Game-Theoretic Models (CSF 2023)



Simplified Closing Game (Bitcoin)

(C_h, S) is not Byzantine Fault Tolerant

Is there a way to financially harm A?

Yes: (C_h, I), when a>0.

(C_h, S) is Incentive Compatible

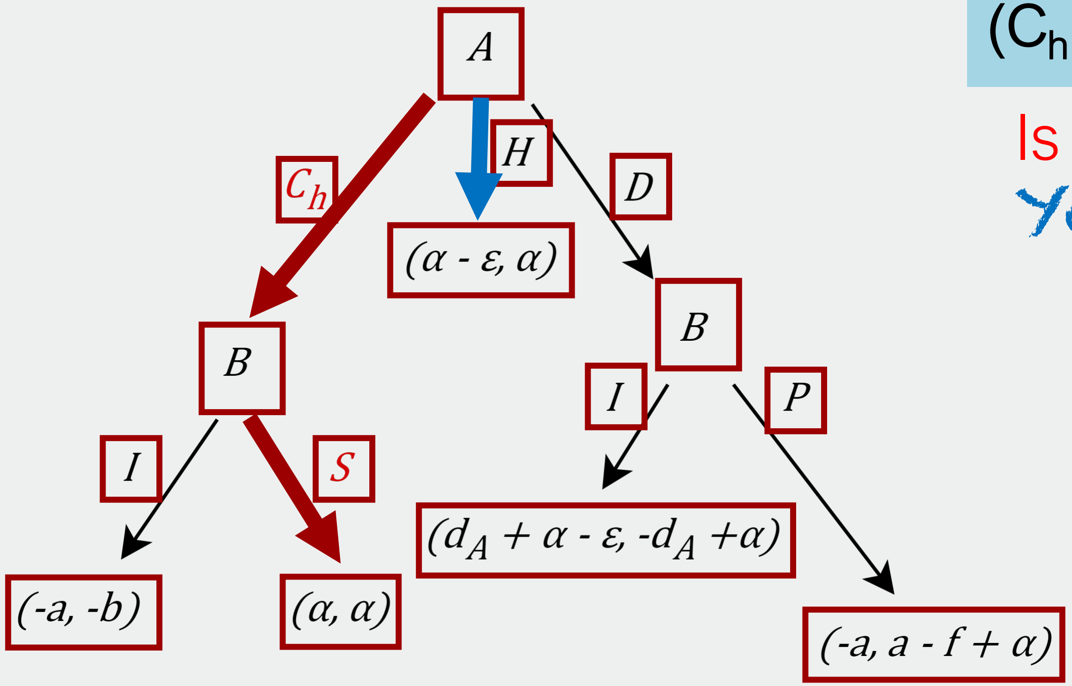
Is deviating from (C_h, S) rational?

No: (C_h, S) yields fair splits.

Is deviating from (H) rational?

Yes: (C_h, S) yields better results for A.

Game-Theoretic Models (CSF 2023)



Simplified Closing Game (Bitcoin)

(C_h, S) is not Byzantine Fault Tolerant ⊗

Is there a way to financially harm A?

Yes: (C_h, I), when a>0.

(C_h, S) is Incentive Compatible ✓

Is deviating from (C_h, S) rational?

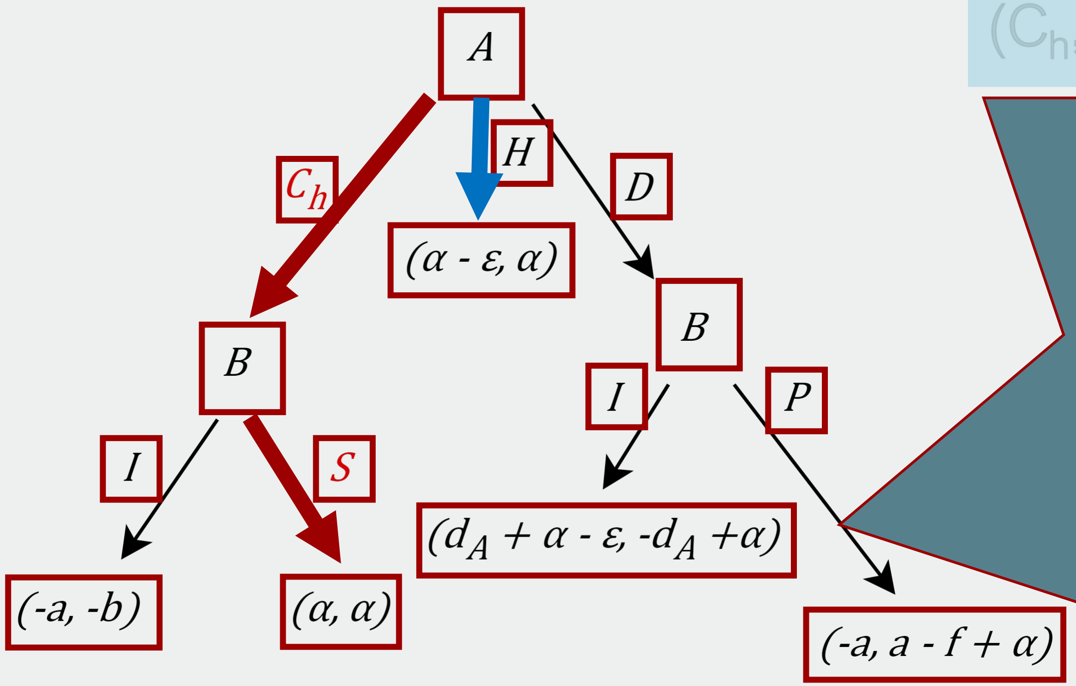
No: (C_h, S) yields fair splits.

(H) is not Incentive Compatible ⊗

Is deviating from (H) rational?

Yes: (C_h, S) yields better results for A.

Game-Theoretic Models (CSF 2023)



Simplified Closing Game (Bitcoin)

Symbolic execution/reasoning on games!

Yes: (C_h, S) yields better results for A.

Blockchain Protocols as Games (CSF 2023)

Closing Game (Bitcoin)

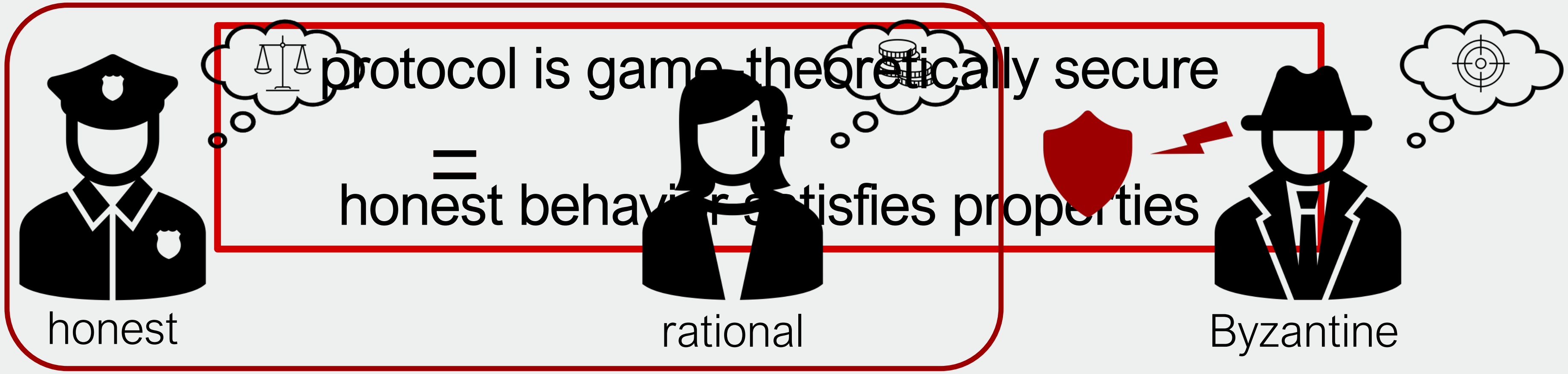
Verifying Game-Theoretic Models (CSF 2023)

1. Incentive Compatibility (IC)

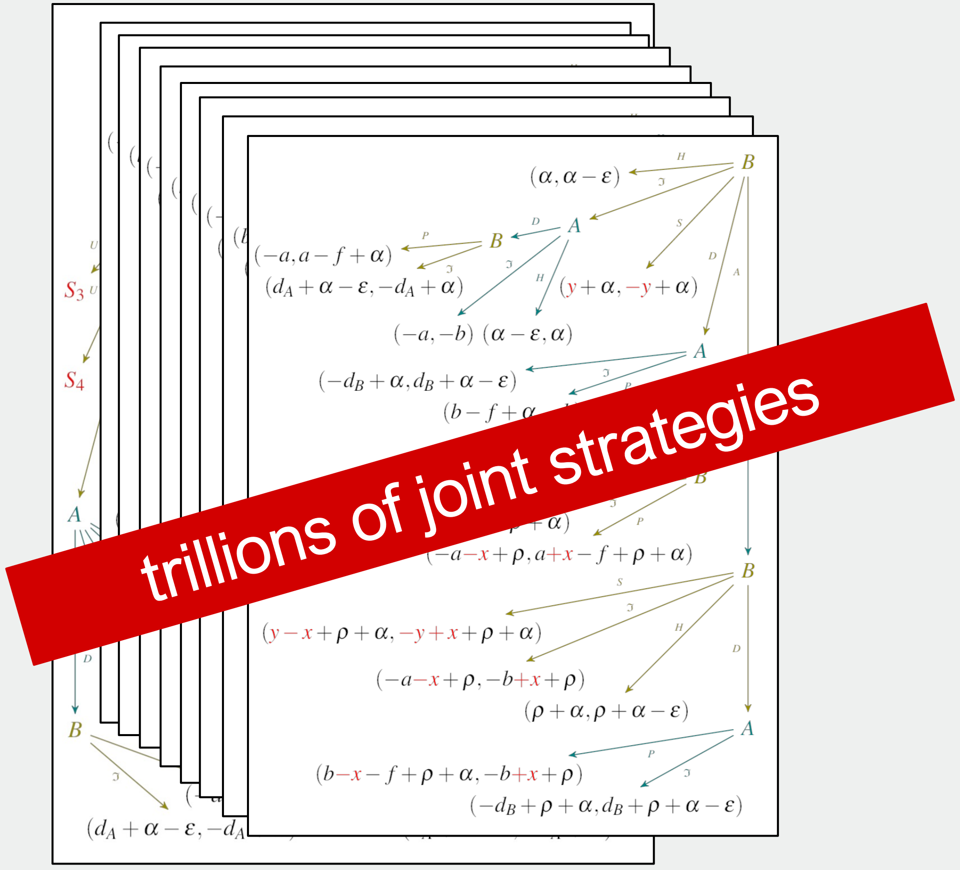
   honest behavior always rational

2. Byzantine Fault Tolerance (BFT)

   honest players never harmed



Security of Closing Game (CSF 2023)

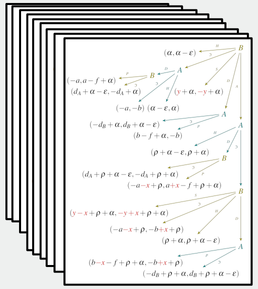


Closing Game (Bitcoin)

Is there a joint strategy that is …

- incentive compatible?
- Byzantine fault tolerant?

Automated Reasoning via SMT in Real Arithmetic (CCS 2023)



game

$\alpha > \epsilon, \epsilon > \rho, \ldots$

initial constraints

(C_h, S)

honest behavior

BFT
security property

$$
\begin{gathered}
\left(v_H^{\emptyset} \vee v_{C_h}^{\emptyset} \vee v_D^{\emptyset} \vee v_{C_c}^{\emptyset}\right) \wedge \\
\left(\neg v_H^{\emptyset} \vee \neg v_{C_h}^{\emptyset}\right) \wedge \\
\left(\neg v_H^{\emptyset} \vee \neg v_D^{\emptyset}\right) \wedge \\
\left(\neg v_H^{\emptyset} \vee \neg v_{C_c}^{\emptyset}\right) \wedge \\
\left(\neg v_{C_h}^{\emptyset} \vee \neg v_D^{\emptyset}\right) \wedge \\
\left(\neg v_{C_h}^{\emptyset} \vee \neg v_{C_c}^{\emptyset}\right) \wedge \\
\left(\neg v_D^{\emptyset} \vee \neg v_{C_c}^{\emptyset}\right) \wedge \\
\ldots
\end{gathered}
$$

SMT constraints

SMT solver

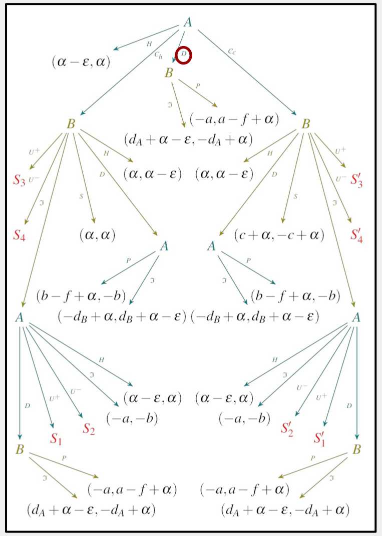
satisfiable

secure

unsatisfiable

not secure

Automated Reasoning via SMT in Real Arithmetic (CCS 2023)



game

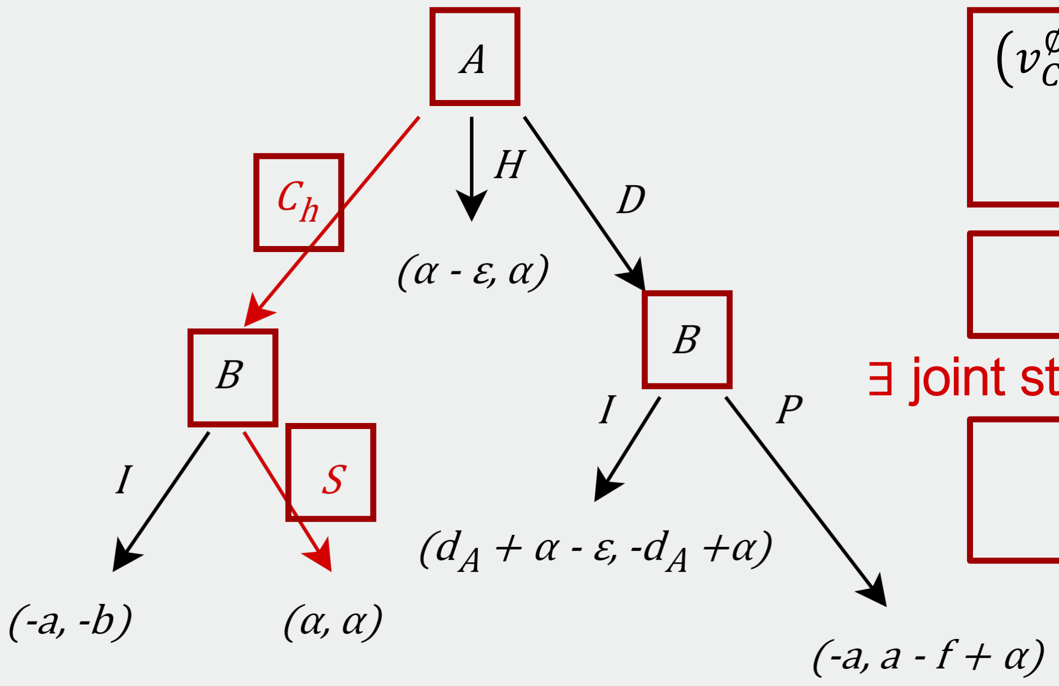
action
≙
Boolean variable

$$\begin{aligned}
&(v_H^\emptyset \lor v_{C_h}^\emptyset \lor v_D^\emptyset \lor v_{C_c}^\emptyset) \land\\
&(\neg v_H^\emptyset \lor \neg v_{C_h}^\emptyset) \land\\
&(\neg v_H^\emptyset \lor \neg v_D^\emptyset) \land\\
&(\neg v_H^\emptyset \lor \neg v_{C_c}^\emptyset) \land\\
&(\neg v_{C_h}^\emptyset \lor \neg v_D^\emptyset) \land\\
&(\neg v_{C_h}^\emptyset \lor \neg v_{C_c}^\emptyset) \land\\
&(\neg v_D^\emptyset \lor \neg v_{C_c}^\emptyset) \land\\
&\ldots
\end{aligned}$$

SMT constraints

Is there a **honest joint strategy**? $\Longleftrightarrow$ Is there a **model**?

Example: SMT Encoding (CCS 2023)



Simplified Closing Game (Bitcoin)

$$\left(v^{\emptyset}_{C_h} \vee v^{\emptyset}_{H} \vee v^{\emptyset}_{D}\right) \wedge \left(\neg v^{\emptyset}_{C_h} \vee \neg v^{\emptyset}_{H}\right) \wedge \left(\neg v^{\emptyset}_{C_h} \vee \neg v^{\emptyset}_{D}\right) \wedge \left(\neg v^{\emptyset}_{H} \vee \neg v^{\emptyset}_{D}\right),$$

$$\ldots,$$

joint strategy constraint

$$v^{\emptyset}_{C_h} \wedge v_S^{(C_h)},$$

∃ joint strategy $\forall \alpha, \epsilon, \ldots : BFT(\alpha, \epsilon, \ldots)$

honest behavior constraint

$$\forall \alpha, \epsilon, \ldots : \mathrm{BFT}\left(\alpha, \epsilon, \ldots, v^{\emptyset}_{C_h}, v^{\emptyset}_{H}, \ldots\right)$$

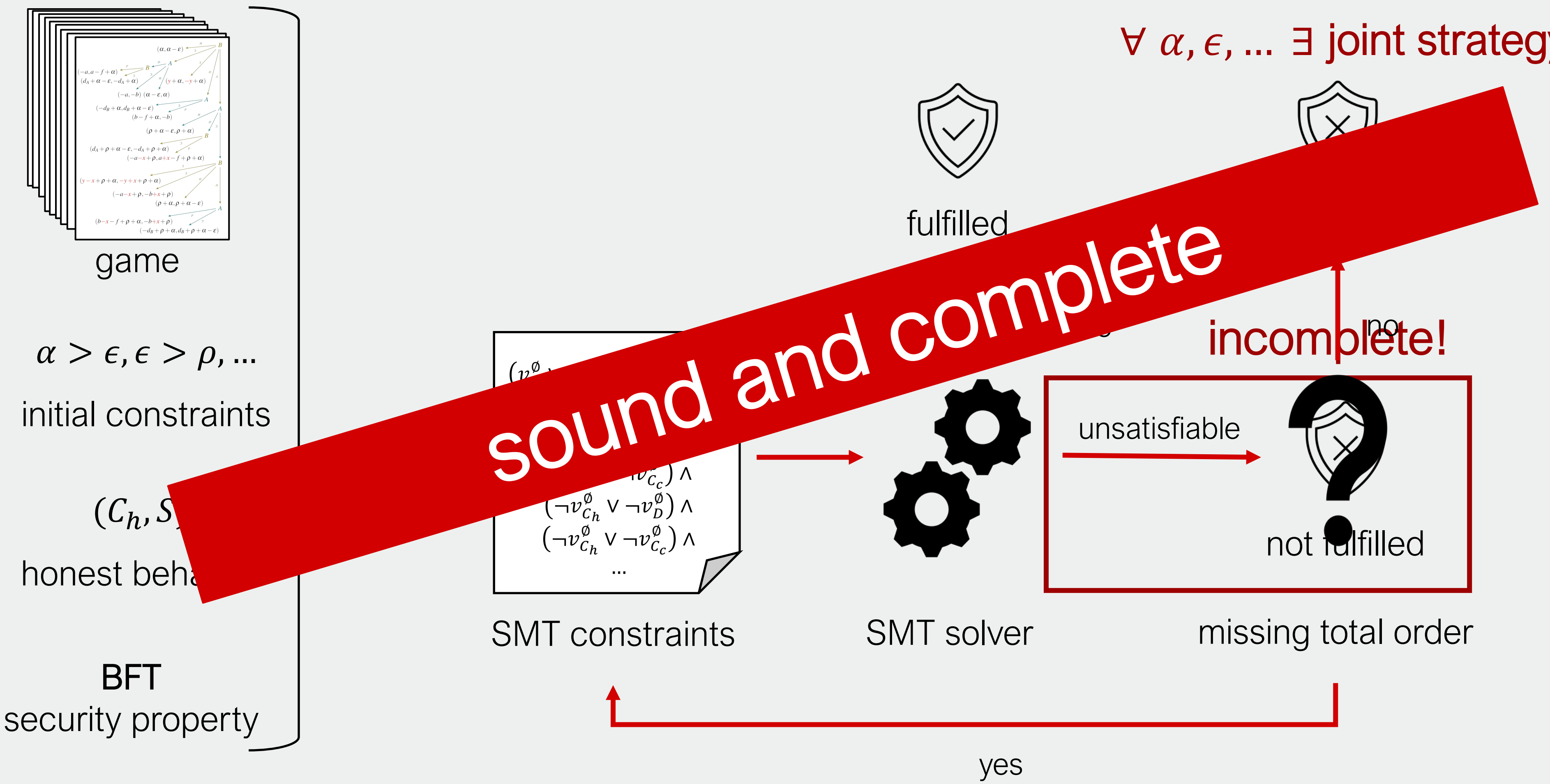
security property constraint

model ≙ honest joint strategy

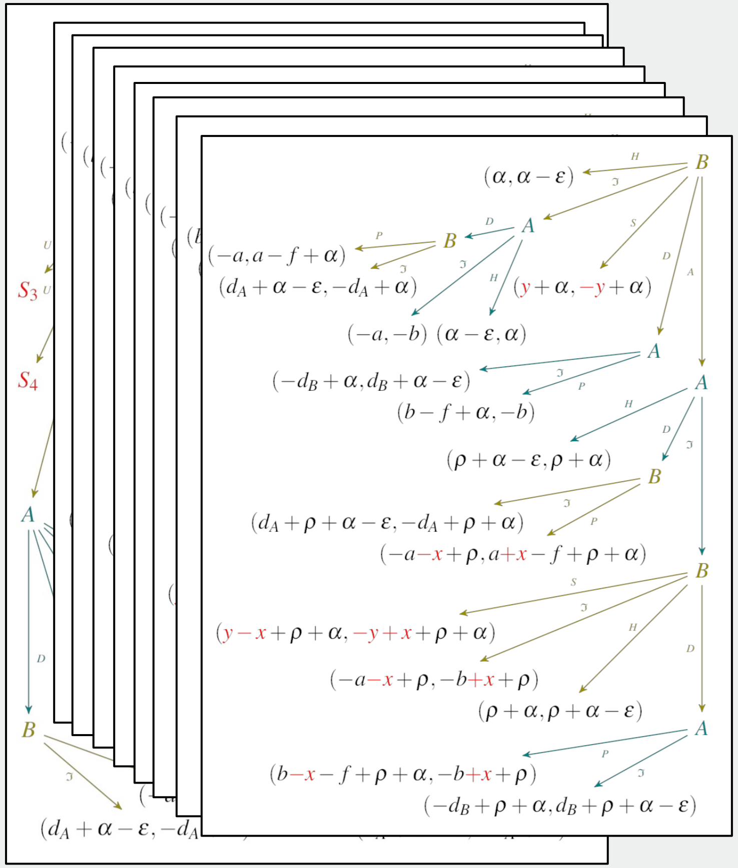
CheckMate (CCS 2023)

∀ total orders of $\alpha, \epsilon, \ldots$ ∃ joint strategy ~~∃ joint strategy ∀ total order $\alpha, \epsilon, \ldots$: $BFT(\alpha, \epsilon, \ldots)$~~

∀ $\alpha, \epsilon, \ldots$ ∃ joint strategy : $BFT(\alpha, \epsilon, \ldots)$



Security of Closing Game – Revisited (CSF 2023, CCS 2023, LPAR 2024)



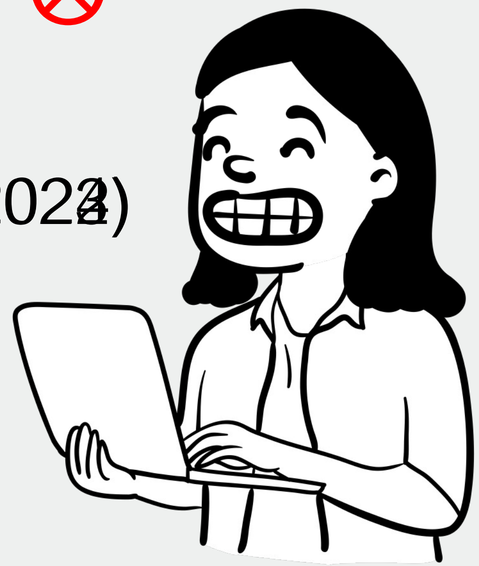
Closing Game (Bitcoin)

Is the **honest behavior** …

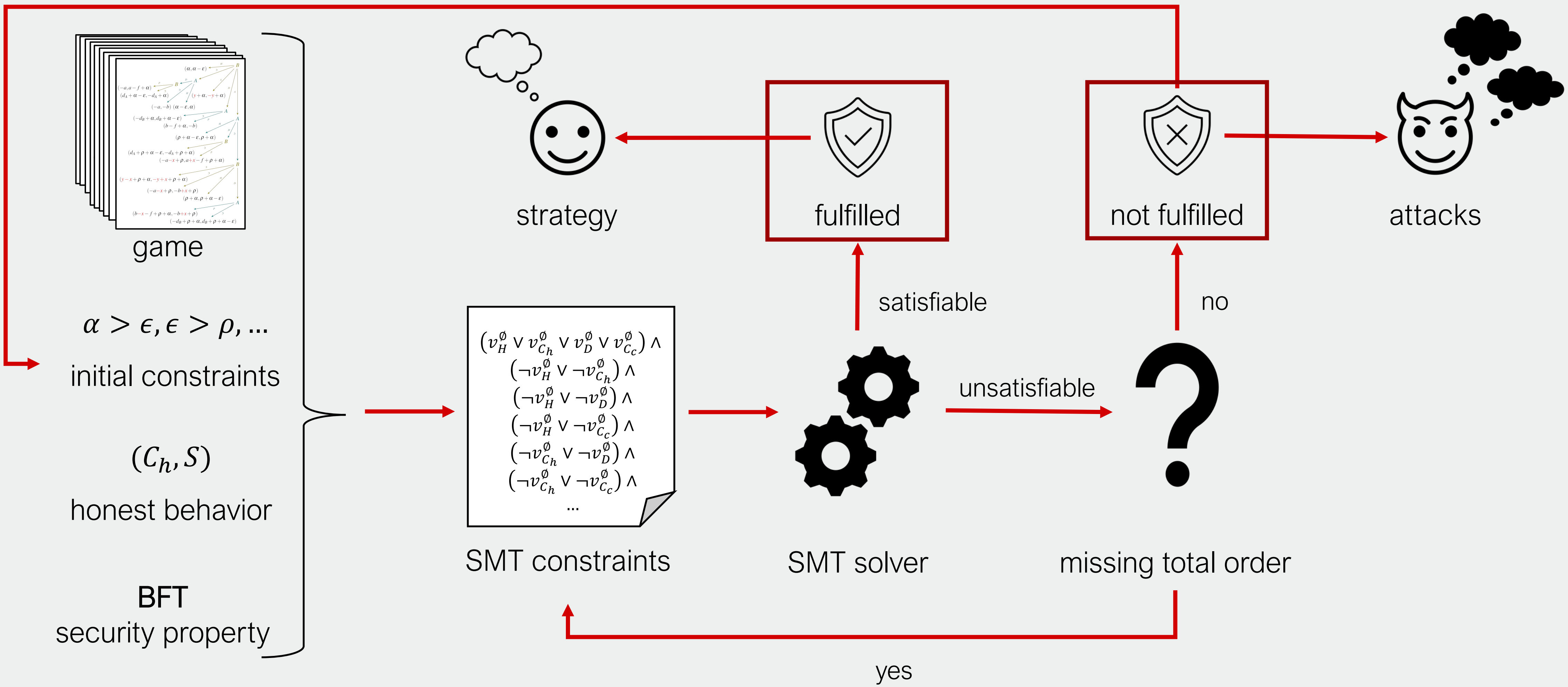
- incentive compatible? ✓
- Byzantine fault tolerant? ⊗

→ 3 minutes execution time (2023)

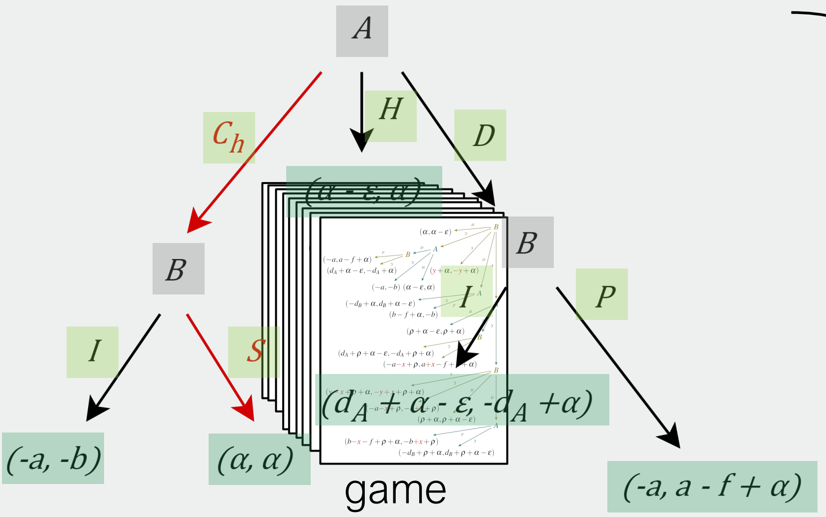
0.5 seconds execution time (2024)



CheckMate (CCS 2023, LPAR 2024)



CheckMate - Input Structure (LPAR 2024)



$\alpha > \epsilon, \epsilon > \rho, \ldots$

initial constraints

(C_h, S)

honest behavior

BFT

security property

```
{ "players": ["A","B"],
  "actions": ["H","D","C_h","S","I","P"],
  "infinitesimals": ["alpha","epsilon"],
  "constants": ["a","b","c","f","d_A"],
  "initial_constraints": ["alpha > epsilon","epsilon > 0",...],
  "property_constraints": {
        "weak_immunity": ["a >= f"],
        "weaker_immunity": ["a >= f"],
        "collusion_resilience": [],
        "practicality": ["a >= f"]},
  "honest_histories": [["C_h","S"]],
  "tree": {
    "player": "A",
    "children": [
      { "action": "H",
        "child": {"utility": [{"player": "A", "value": "alpha - epsilon"},
                              {"player": "B", "value": "alpha"}]}},
      { "action": "C_h",
        "child": {"player": "B",
                  "children": [{ "action": "I",
                                 "child": {"utility": [{"player": "A", "value": "-a"},
                                                       {"player": "B", "value": "-b"}]}},
                               { "action": "S",
                                 "child": {"utility": [{"player": "A", "value": "alpha"},
                                                       {"player": "B", "value": "alpha"}]}}]}},
      { "action": "D",
        "child": {"player": "B",
                  "children": [...]}}]}}
```

BFT

IC

CheckMate - Output Structure (LPAR 2024)

Calling CheckMate on Simplified Closing Game:

- add honest behavior (H)
- remove initial constraint $a \geq f$

property

honest behavior

case splits

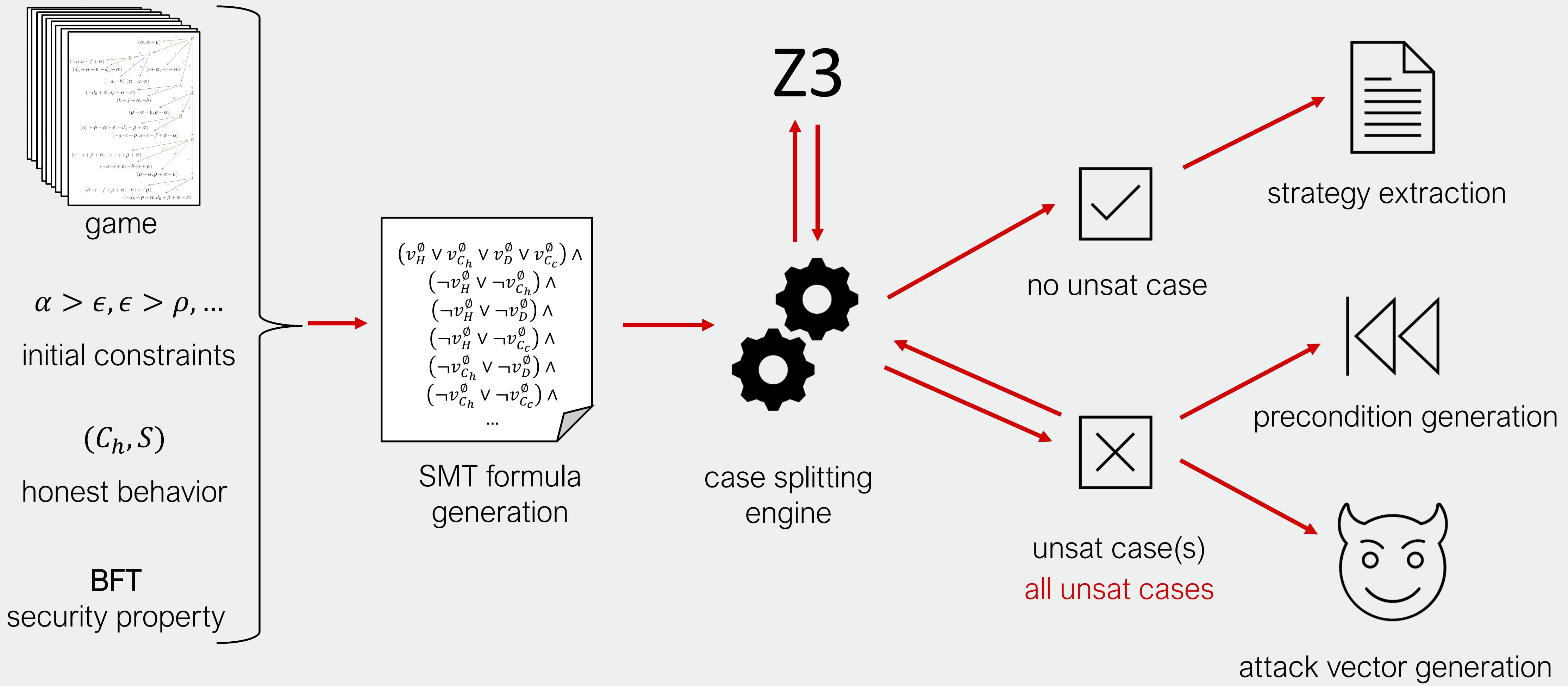
result

```
WEAK IMMUNITY

Is history [H] weak immune?
        Require case split on (> (- a f) 0.0)
        Case [(> (- a f) 0.0)] satisfies property.
        Require case split on (= (- a f) 0.0)
        Case [(<= (- a f) 0.0), (= (- a f) 0.0)] satisfies property.
        Case [(<= (- a f) 0.0), (distinct (- a f) 0.0)] violates property.
NO, it is not weak immune.


Is history [C_H,S] weak immune?
        Case [] violates property.
NO, it is not weak immune.
```

CheckMate Features (LPAR 2024)



Experimental Evaluation (LPAR 2024)

current version

initial prototype

game-theoretic

blockchain protocol

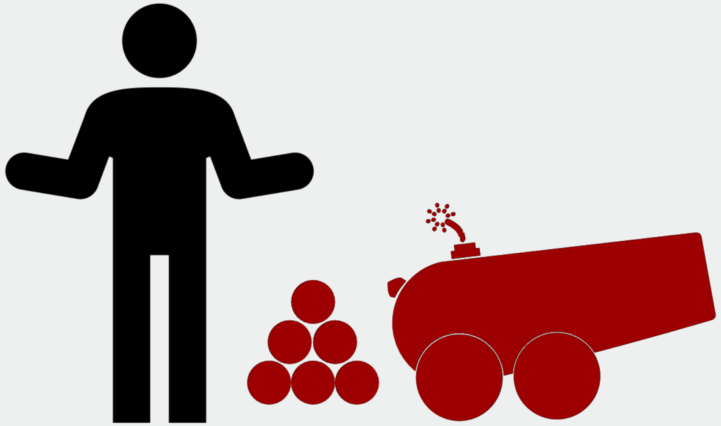
Game	Nodes	Players	Histories	Time (v1)	Time (v0)
Splits_{wi}	5	2	3	0.03	0.35
Splits_{cr}	5	2	3	0.03	0.35
Market Entry	5	2	3	0.02	0.28
Simplified Closing	8	2	2	0.02	0.26
Simplified Routing	17	5	1	0.02	0.31
Pirate	52	4	40	1.07	27.08
Closing	221	2	2	0.34	9.60
3-Player Routing	21,688	3	1	6.83	242.54
G (Figure 2)	5	2	1	0.02	0.18
Centipede	19	3	1	0.07	0.48
EBOS	31	4	1	0.02	0.53
Auction	92	4	1	0.11	1.72
Unlocking Routing	36,113	5	1	10.85	478.58
Tic Tac Toe Concise	58,748	2	1	107.84	254.87
Tic Tac Toe	549,946	2	1	TO	TO

Summary – Game-Theoretic Security

incentive compatibility,

Byzantine fault tolerance

CheckMate



automated

game analysis

game strategies,
symbolic utilities,
…

security proof