Microsoft Research

Each year Microsoft Research hosts hundreds of influential speakers from around the world including leading scientists, renowned experts in technology, book authors, and leading academics, and makes videos of these lectures freely available.

Super-Human AI for Strategic Reasoning

Tuomas Sandholm

Angel Jordan Professor of Computer Science, Carnegie Mellon University

Founder and Director, Electronic Marketplaces Laboratory

Co-Director, CMU AI

Founder, President, and CEO, Strategic Machine, Inc.

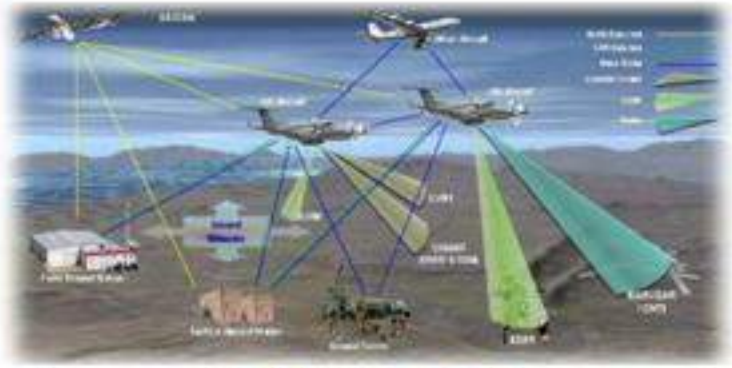
Founder, President, and CEO, Strategy Robot, Inc.

Founder, President, and CEO, Optimized Markets, Inc.

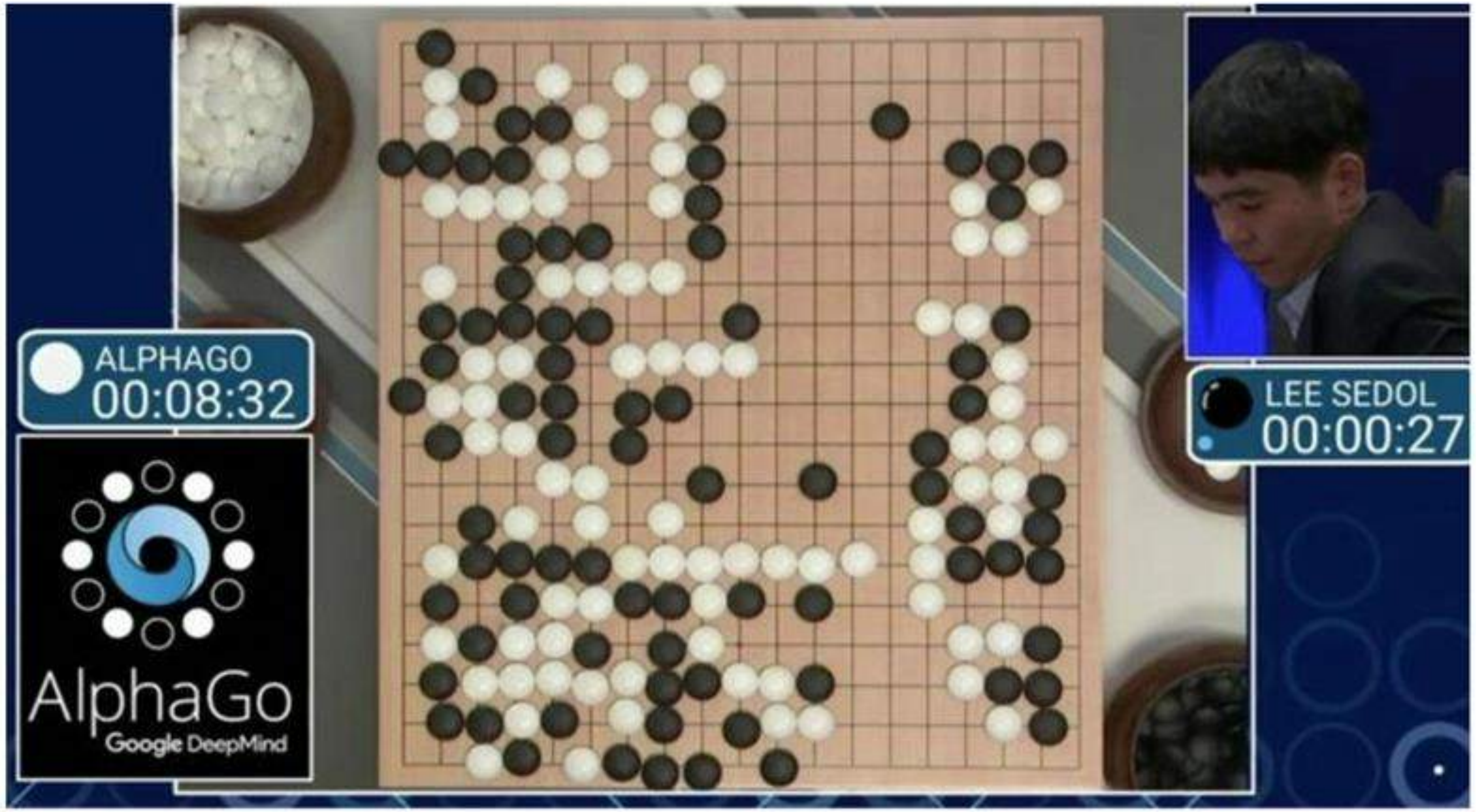
Paper in *Science* (Dec. 2017), joint work with my PhD student **Noam Brown**

Imperfect-information games




AlphaGo & AlphaZero



Those techniques extend to other **perfect-information** games

Perfect-information games

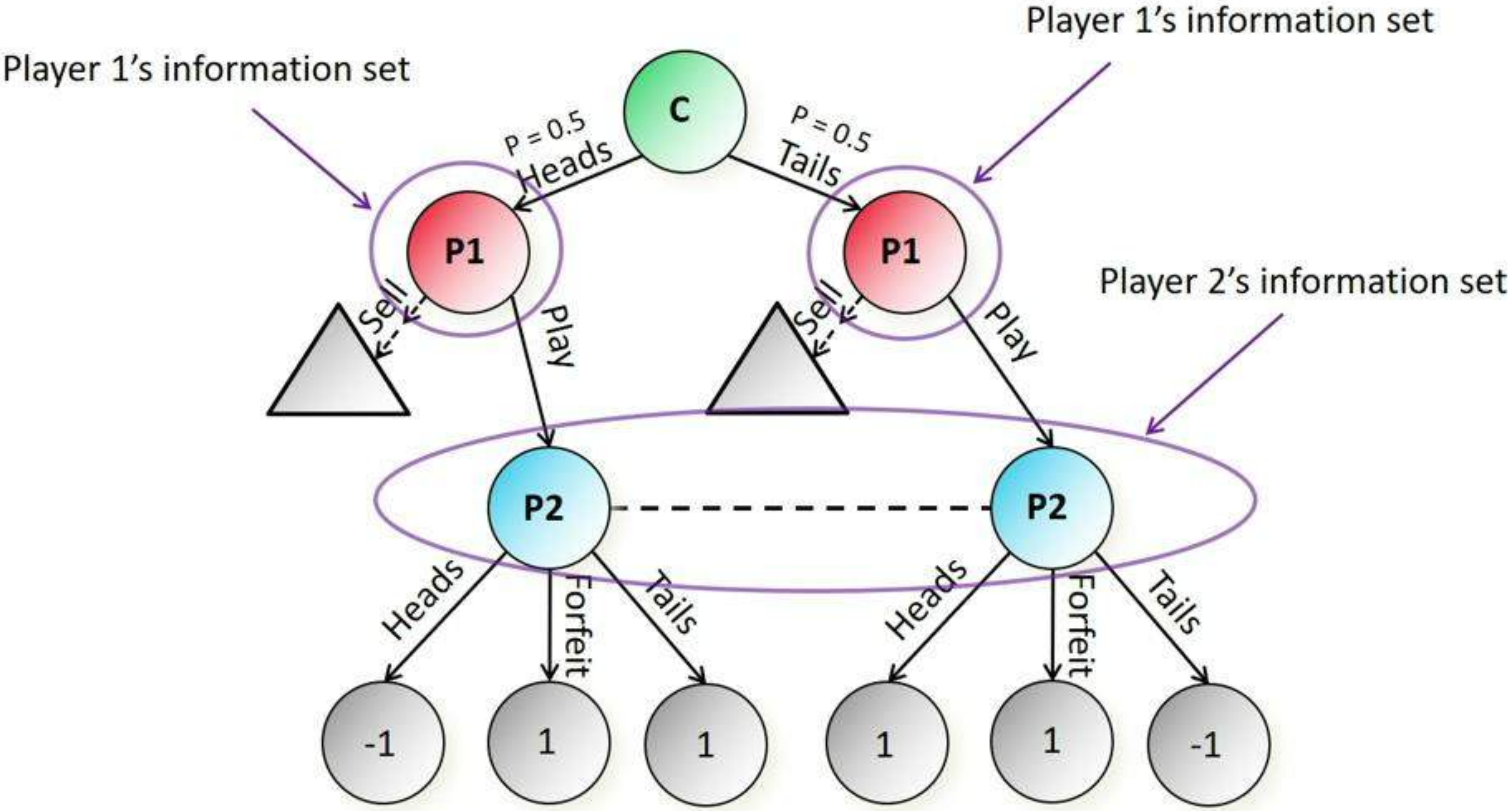
Sicilian Defense

Queen's Gambit

- Subgames can be solved with information from the subgame only
- This is **not true** in imperfect-information games

Imperfect-information games

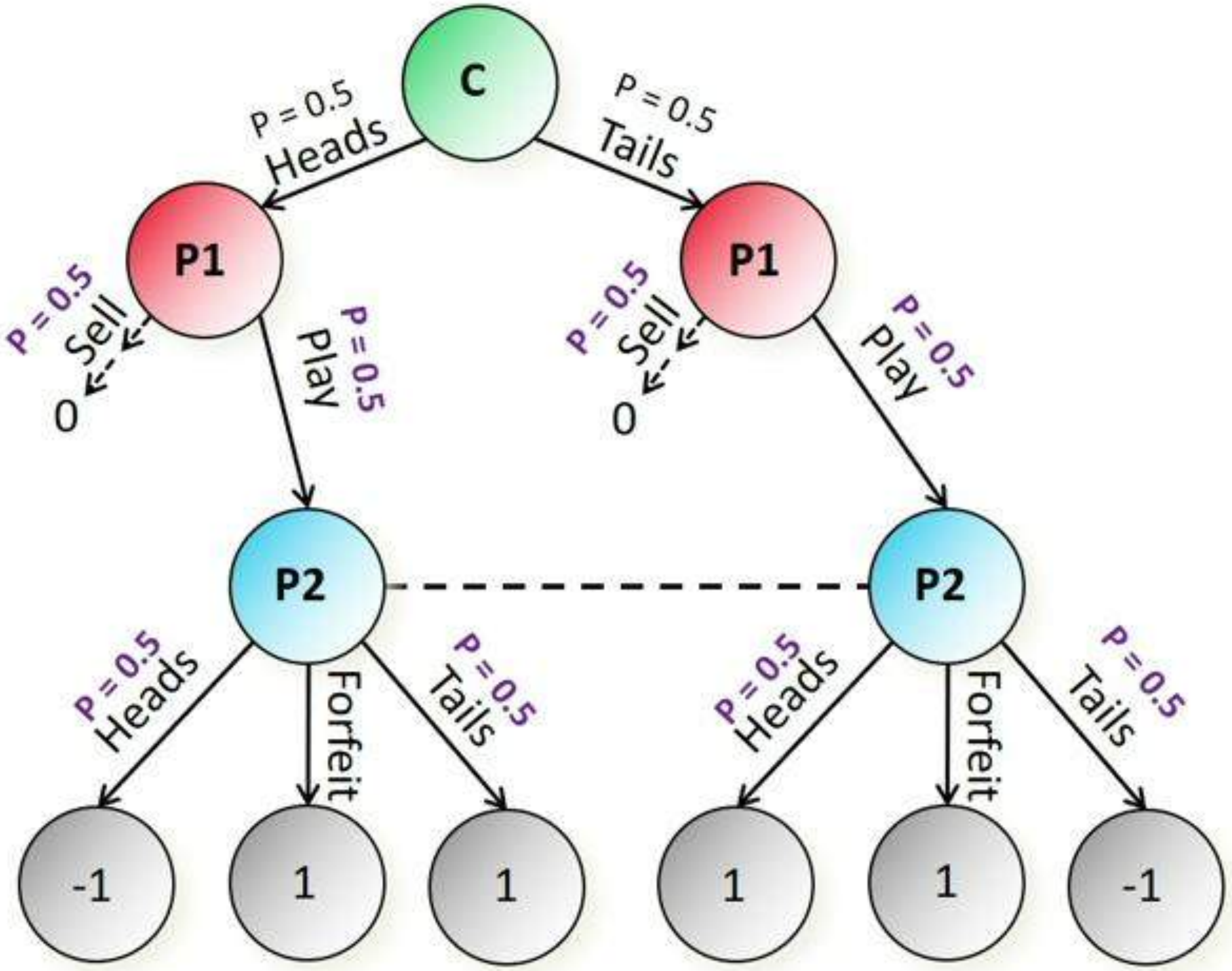
Example game: "Coin toss"

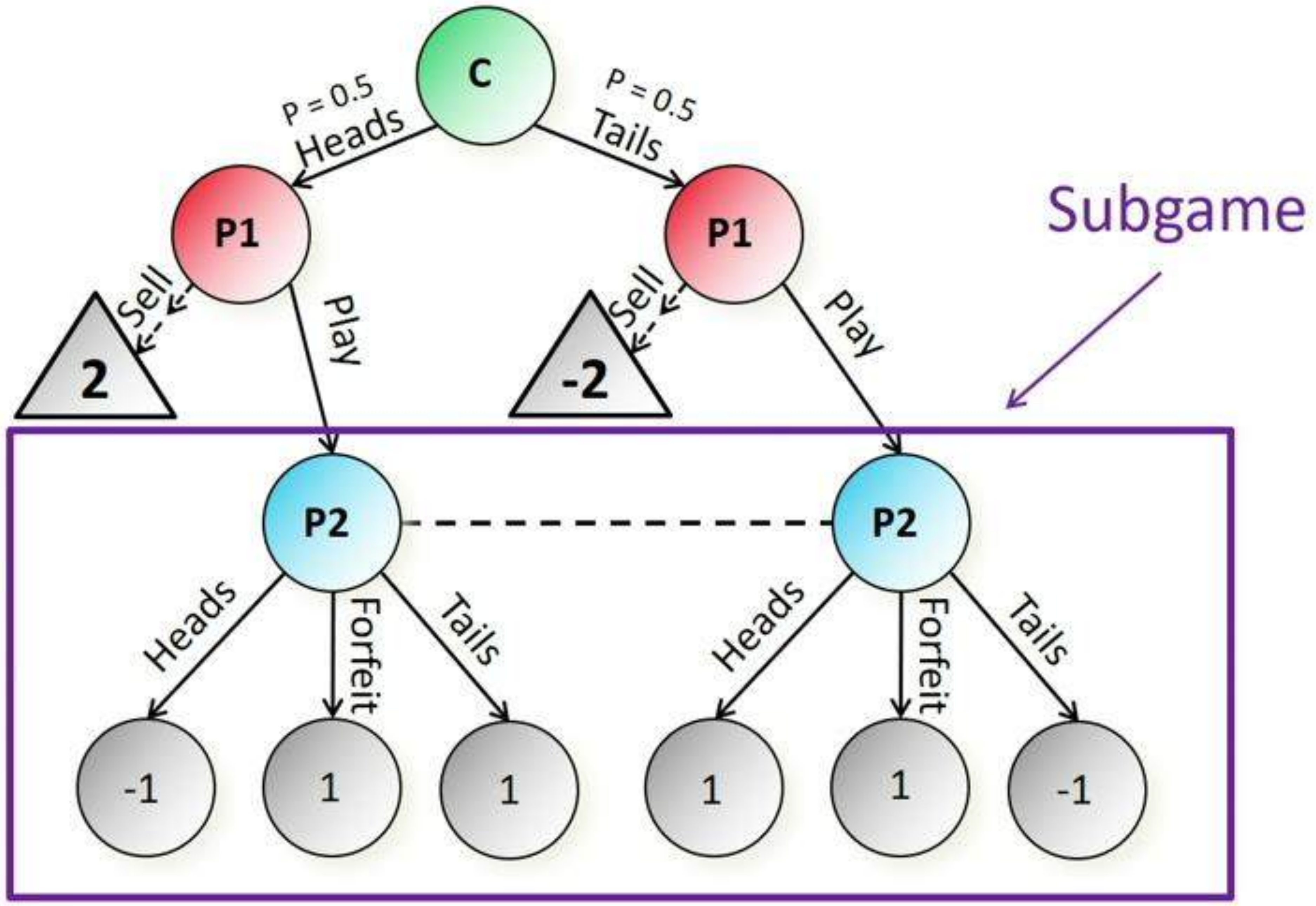


What is rational play?

Nash Equilibrium: a profile of strategies in which no player can improve by deviating (beliefs derived from strategies using Bayes rule). **Robust**

ϵ-Nash Equilibrium: No player can improve by more than ϵ




C
P = 0.5
Heads
P = 0.5
Tails
P1
P1
Sell
Play
Sell
Play
2
-2
Subgame
P2
P2
Heads
Forfeit
Tails
Heads
Forfeit
Tails
-1
1
1
1
1
-1

Tackling imperfect-info games

- Techniques for perfect-info games don't apply
- Application-independent techniques that algorithmically create the strategy
- Challenges
  – Uncertainty about what others and chance will do
  – Hidden state => need to interpret signals
  => use game theory

Poker

- Recognized challenge problem in game theory, OR, and AI
  - [Nash 1950]
  - [Kuhn 1950]
  - [Newman 1959]
  - [Waterman 1970]
  - [Zadeh 1977]
  - [Caro 1984]
  - [Pfeffer & Koller 1995]
  - [Billings *et al.* 1998]
  - [Schaeffer *et al.* 1999]
  - [Shi & Littman 2001]
  - [Billings *et al.* 2003]
- Tremendous progress in the last 14 years
  - Rhode Island Hold'em solved (10^9 nodes) [Gilpin & Sandholm 2005]
  - Annual Computer Poker Competition started in 2006
  - Limit Texas Hold'em near-optimally solved (10^{13} decisions) [Bowling *et al.* 2015]

Heads-up no-limit Texas hold'em

- Has become the main ***benchmark and challenge problem*** in AI for imperfect-information games
- 10^{161} situations
- Mostly played on the Internet
  - Also in World Series of Poker, NBC Heads-Up Championship, etc.
  - Featured in *Casino Royale* and *Rounders*
- "Purest form of poker"
- No prior AI has beaten top humans

Texas hold'em

- Chance deals 2 cards to each player
- Round of betting
- Chance deals 3 shared cards
- Round of betting
- Chance deals 1 shared card
- Round of betting
- Chance deals 1 shared card
- Round of betting

Brains vs AI Rematch

- *Libratus* (= our AI) against four of the ***best*** heads-up no-limit Texas Hold'em specialist pros



- 120,000 hands over 20 days in January 2017
- $200,000 divided among the pros based on performance
- Conservative experiment design


POKER NIGHT
WE'LL BE
BACK SHORTLY

Final result [Brown & S. *Science*]

Libratus beat the top humans in this game by a lot

- Statistical significance 99.98%, i.e., 0.0002
- 147 mbb/hand
- Each human lost to Libratus
- "Science breakthrough of the year 2017", 2017 Supercomputing Award, …

Lengpudashi vs humans event

- 36,000 hands against 6 Chinese poker players
  - WSOP bracelet winner
  - Expertise in computer science & ML
  - Studied Libratus's hand histories in advance

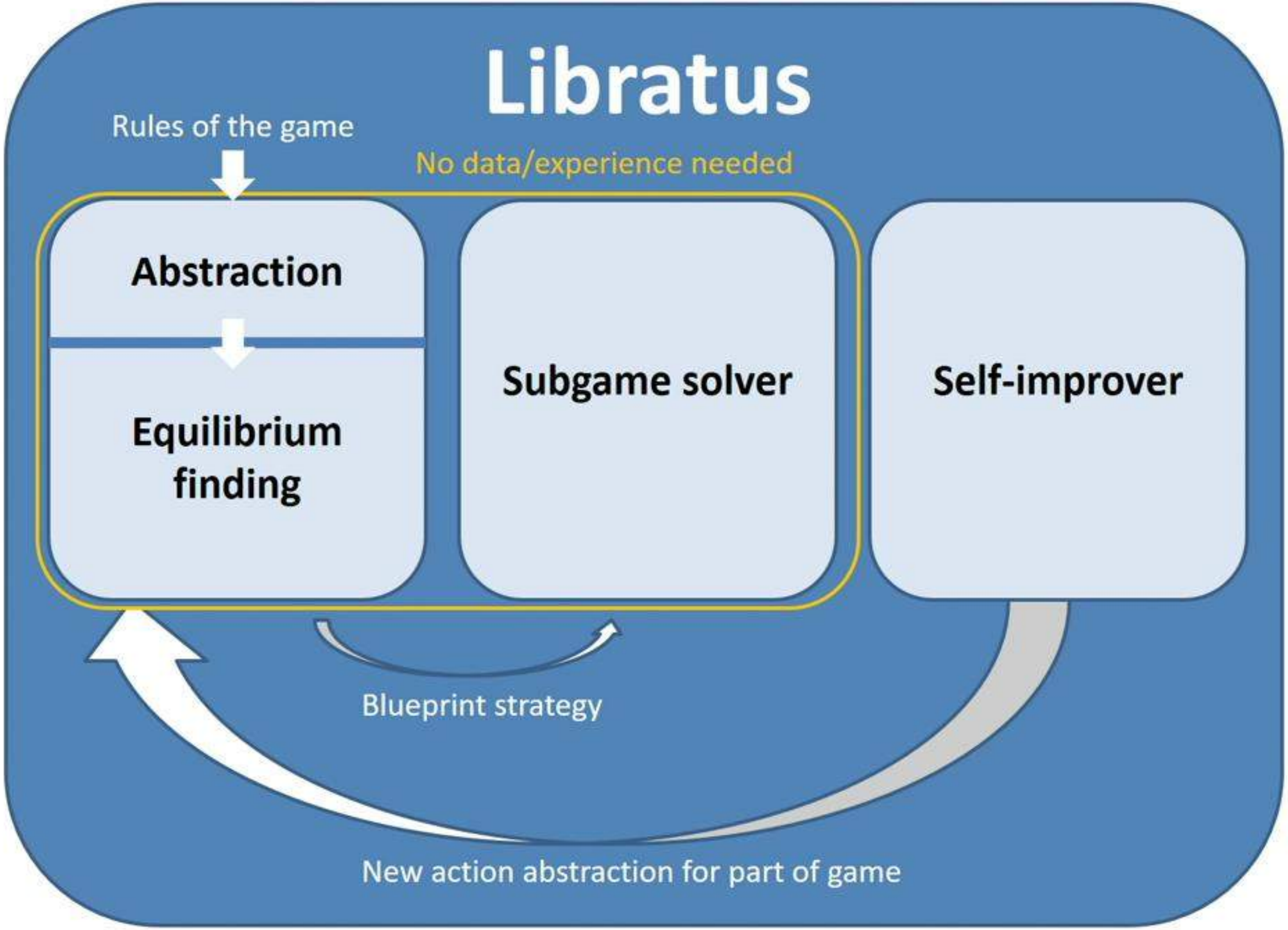
Lengpudashi vs humans event

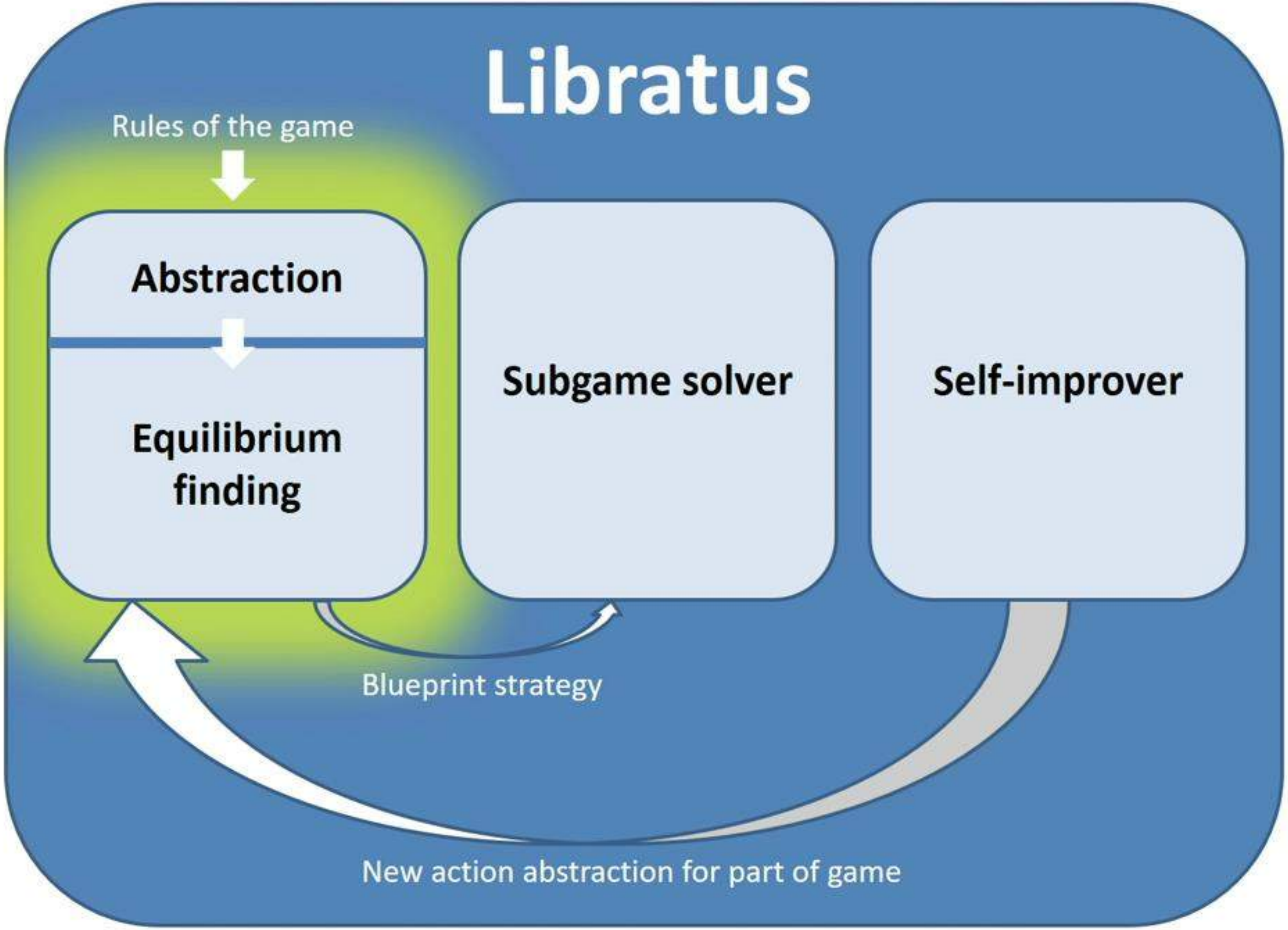
- 36,000 hands against 6 Chinese poker players
  - WSOP bracelet winner
  - Expertise in computer science & ML
  - Studied Libratus's hand histories in advance
- Lengudashi won by 220 mbb/hand
  - Won each of the 9 sessions
  - Also beat each human individually
- Demonstrated that this approach is not frail
  - Unlike what has been found with ML approaches (e.g., for Go and DOTA2)

How does *Libratus* work?

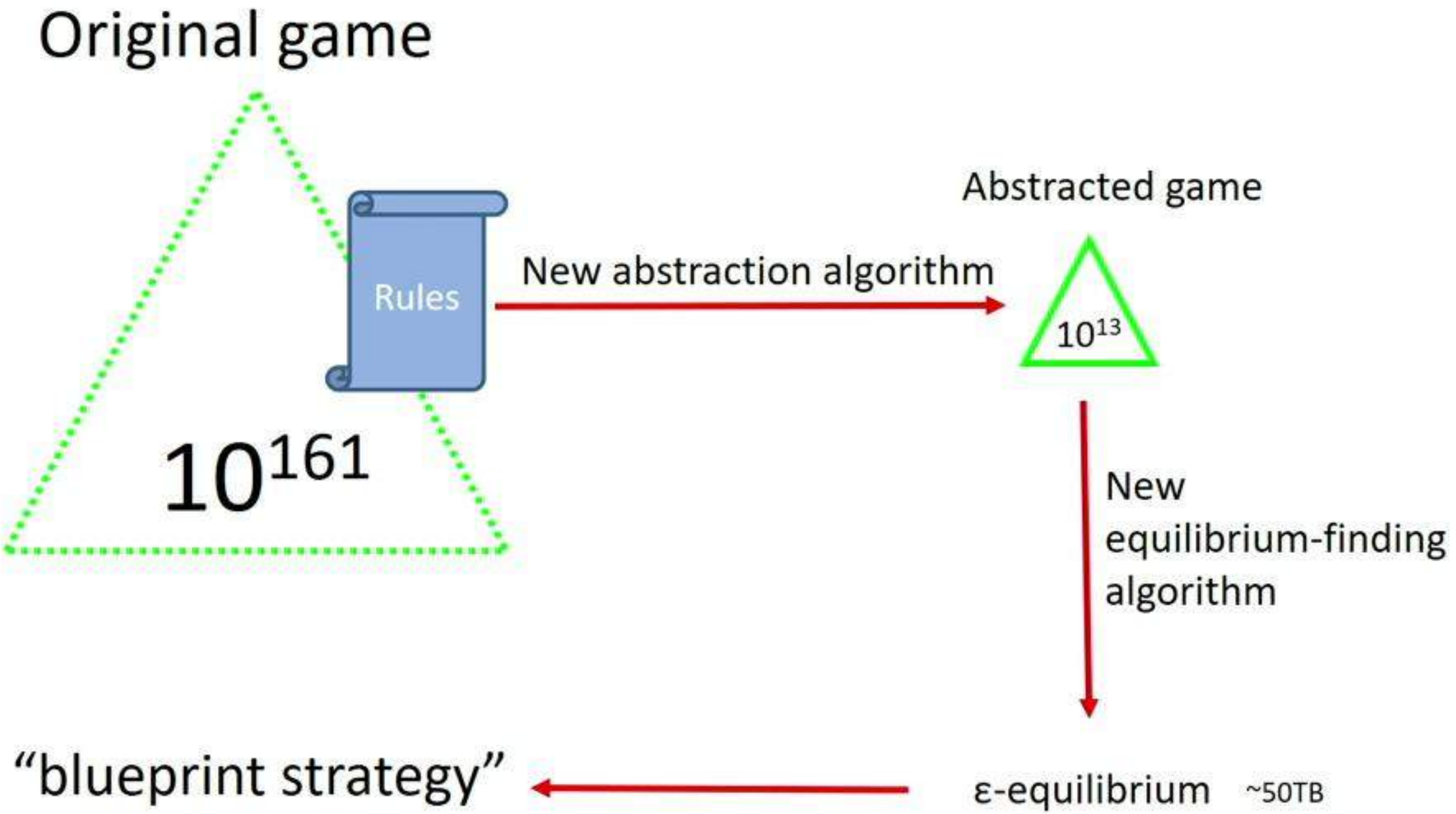
Provably correct techniques


Bridges supercomputer


Libratus
Rules of the game
No data/experience needed
Abstraction
Equilibrium finding
Subgame solver
Self-improver
Blueprint strategy
New action abstraction for part of game


Libratus
Rules of the game
Abstraction
Equilibrium finding
Subgame solver
Self-improver
Blueprint strategy
New action abstraction for part of game

Abstraction [Gilpin & Sandholm EC-06, *J. of the ACM* 2007…]



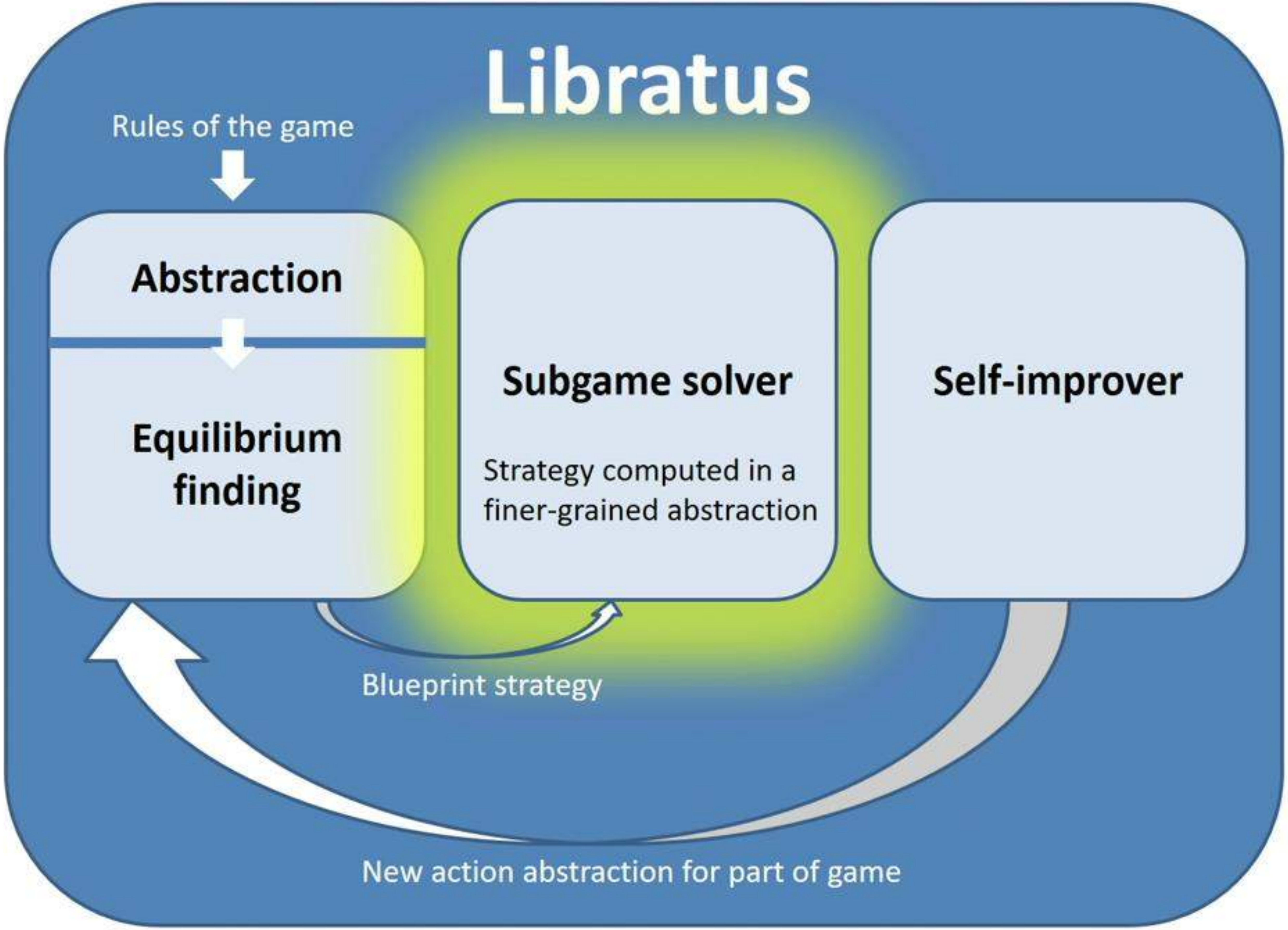
Foreshadowed by Shi & Littman 01, Billings et al. IJCAI-03

Abstraction in Libratus

- Abstracting chance's action (cards in poker)
  - Same algorithm that we used in *Tartanian8* [Brown, Ganzfried & Sandholm AAMAS-15]
  - But much finer abstraction
    - 1st and 2nd betting round: no abstraction
    - 3rd betting round: 55M card histories -> 2.5M buckets
    - 4th betting round: 2.4B card histories -> 1.25M buckets
- Abstracting player's actions (bet sizes in poker)
  - Largely based on what top humans and AIs do
  - Added radical bet sizes
  - Optimized some of the bet sizes in the early parts of the tree [Brown & Sandholm AAAI-14]

Our equilibrium-finding algorithm

- Improvement on Monte-Carlo Counterfactual Regret Minimization [Lanctot *et al.* *NIPS*-09]
- Starts visiting less often paths where our own actions don't look promising (similar to Brown & Sandholm NIPS-15 paper and AAAI-17 workshop paper)
  => Speedup => can solve larger abstractions
- Also, the imperfect-recall abstraction, in effect, becomes finer grained
  => Better solution quality
- Distributed across 1 + 195 compute nodes
  - Distribution along game tree, not "embarrassingly parallel"


Libratus
Rules of the game
Abstraction
Equilibrium finding
Subgame solver
Strategy computed in a finer-grained abstraction
Self-improver
Blueprint strategy
New action abstraction for part of game

New ideas in subgame solver

NIPS-17 best paper award

- Provably *safe* subgame solving taking into account opponent's mistakes in the hand so far
- Off-tree actions & nested subgame solving
- Subgame solving starts much earlier
- No card abstraction in the subgame
- Changed our action abstraction between hands

New ideas in subgame solver

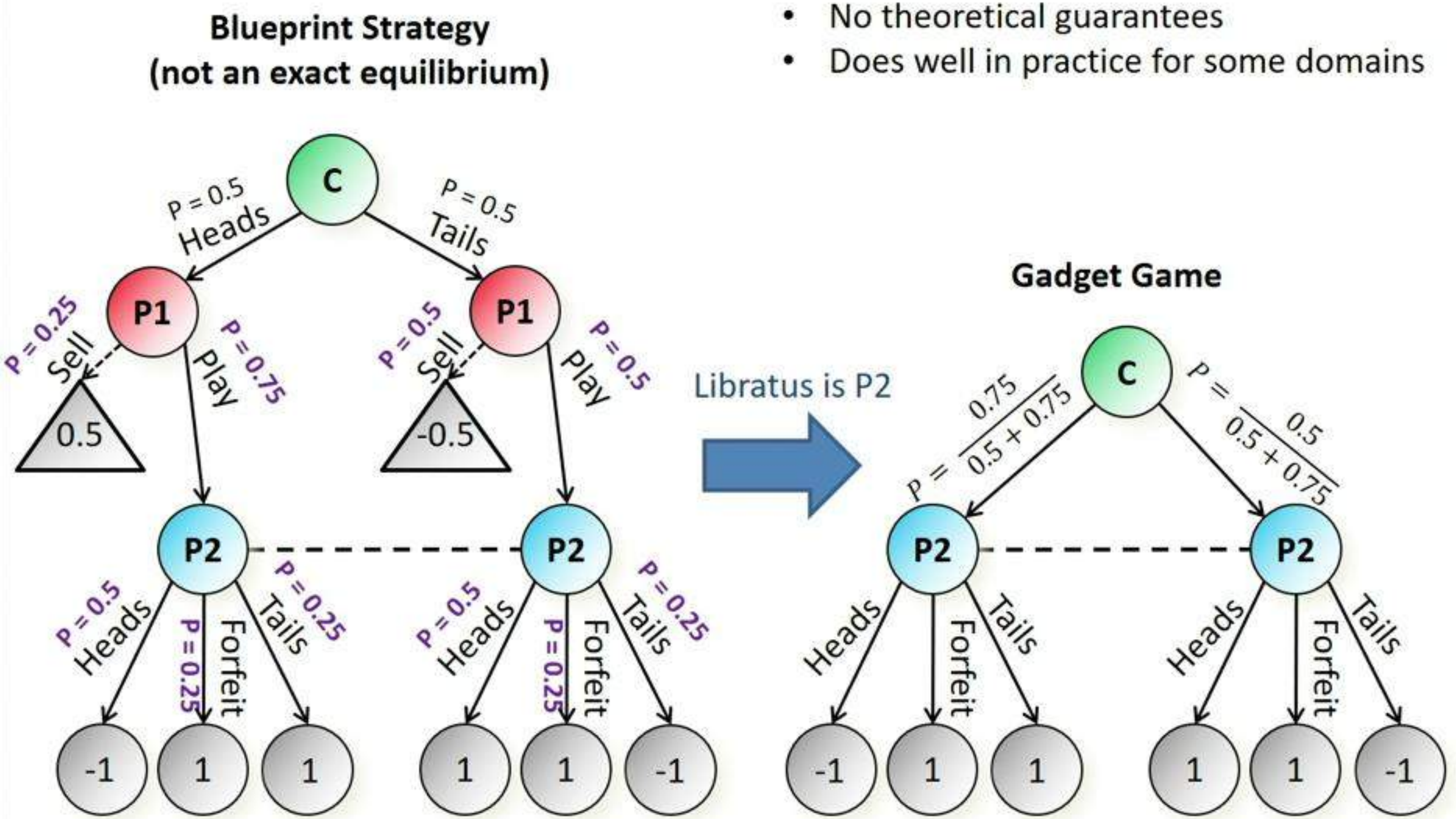
NIPS-17 best paper award

- Provably *safe* subgame solving taking into account opponent's mistakes in the hand so far
- Off-tree actions & nested subgame solving
- Subgame solving starts much earlier
- No card abstraction in the subgame
- Changed our action abstraction between hands

Unsafe subgame solving

[Ganzfried & Sandholm AAAMAS 2015]

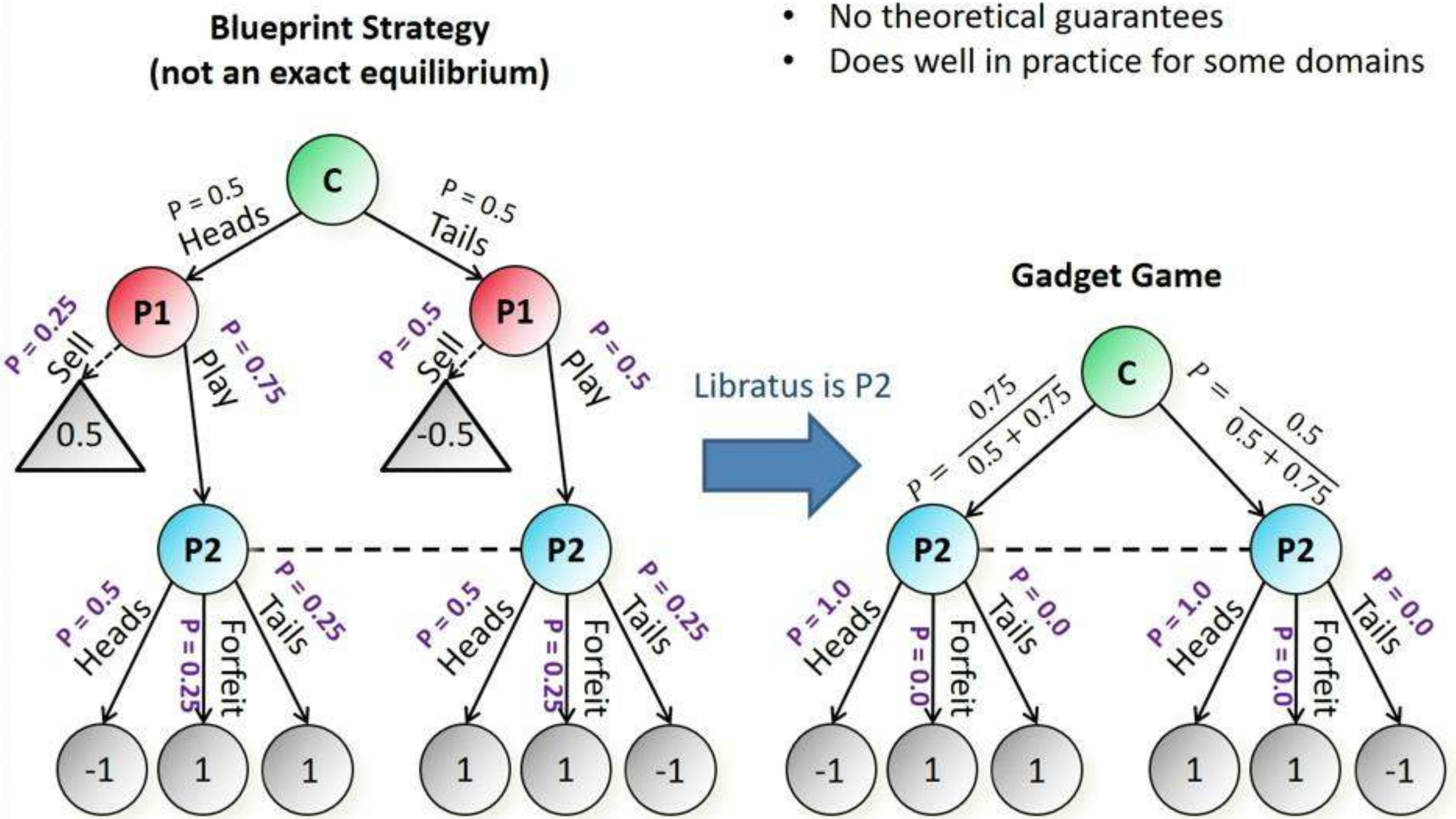
- No theoretical guarantees
- Does well in practice for some domains



Unsafe subgame solving

[Ganzfried & Sandholm AAAMAS 2015]

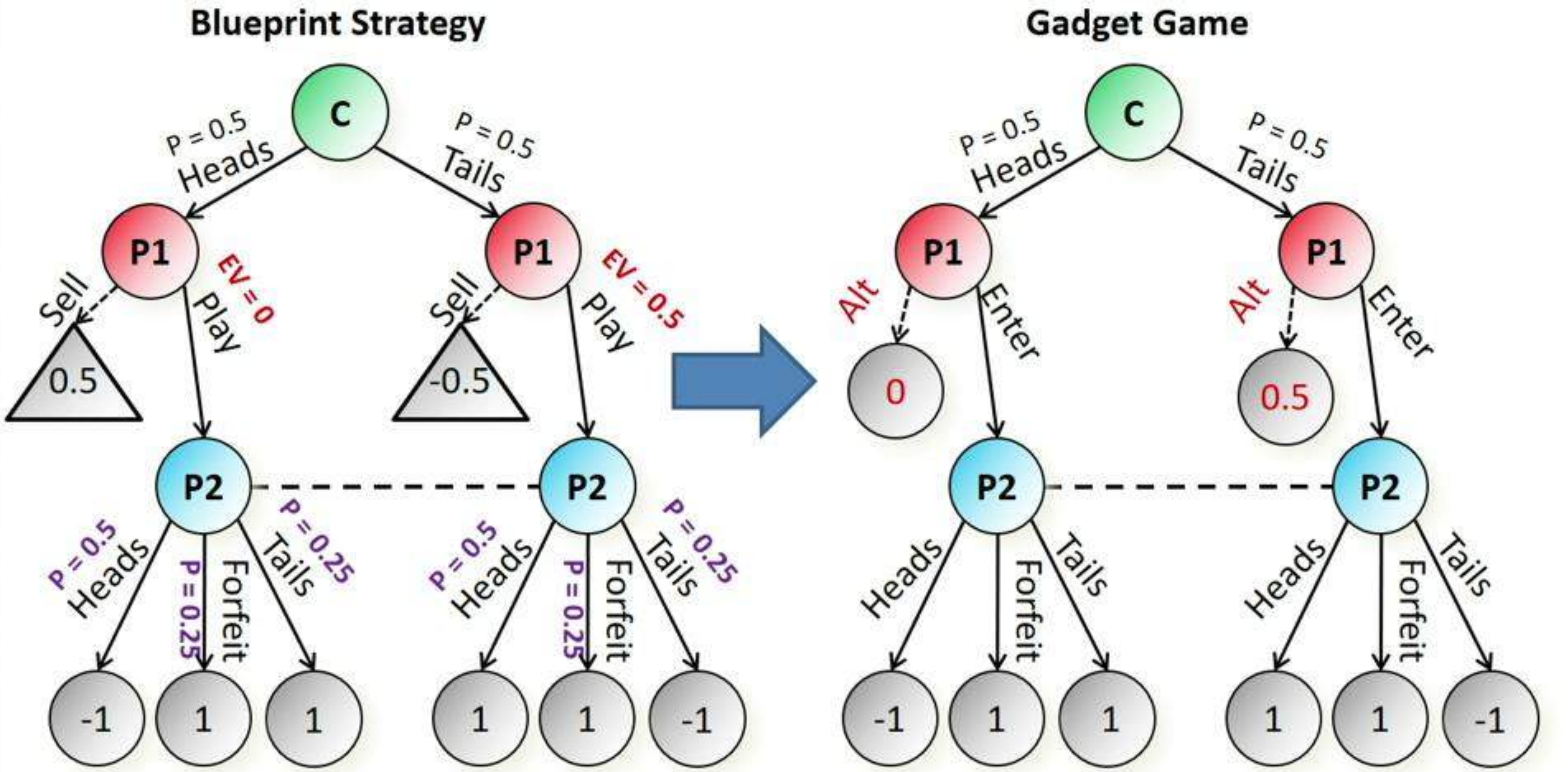
- No theoretical guarantees
- Does well in practice for some domains



Re-solve refinement

[Burch *et al*. AAAI 2014]

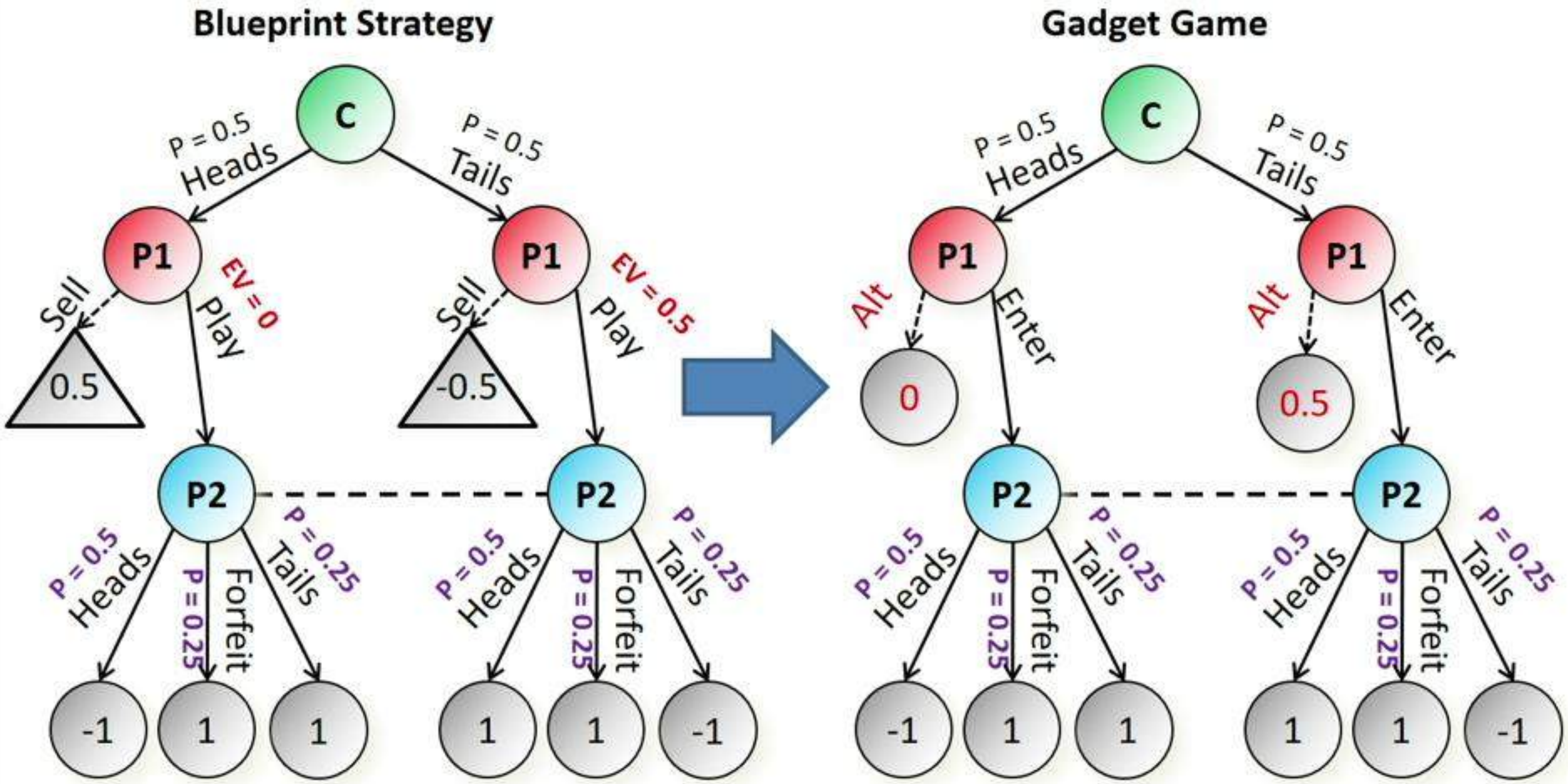
- P1 can choose between entering the subgame or taking the EV (according to the blueprint) of the subgame
- Makes sure opponent's EV for entering the subgame is no higher than in the blueprint strategy
  => Strategy provably no worse than blueprint strategy
- But may miss obvious opportunities for improvement (e.g., not forfeiting)



Re-solve refinement

[Burch *et al*. AAAI 2014]

- P1 can choose between entering the subgame or taking the EV (according to the blueprint) of the subgame
- Makes sure opponent's EV for entering the subgame is no higher than in the blueprint strategy
  => Strategy provably no worse than blueprint strategy
- But may miss obvious opportunities for improvement (e.g., not forfeiting)

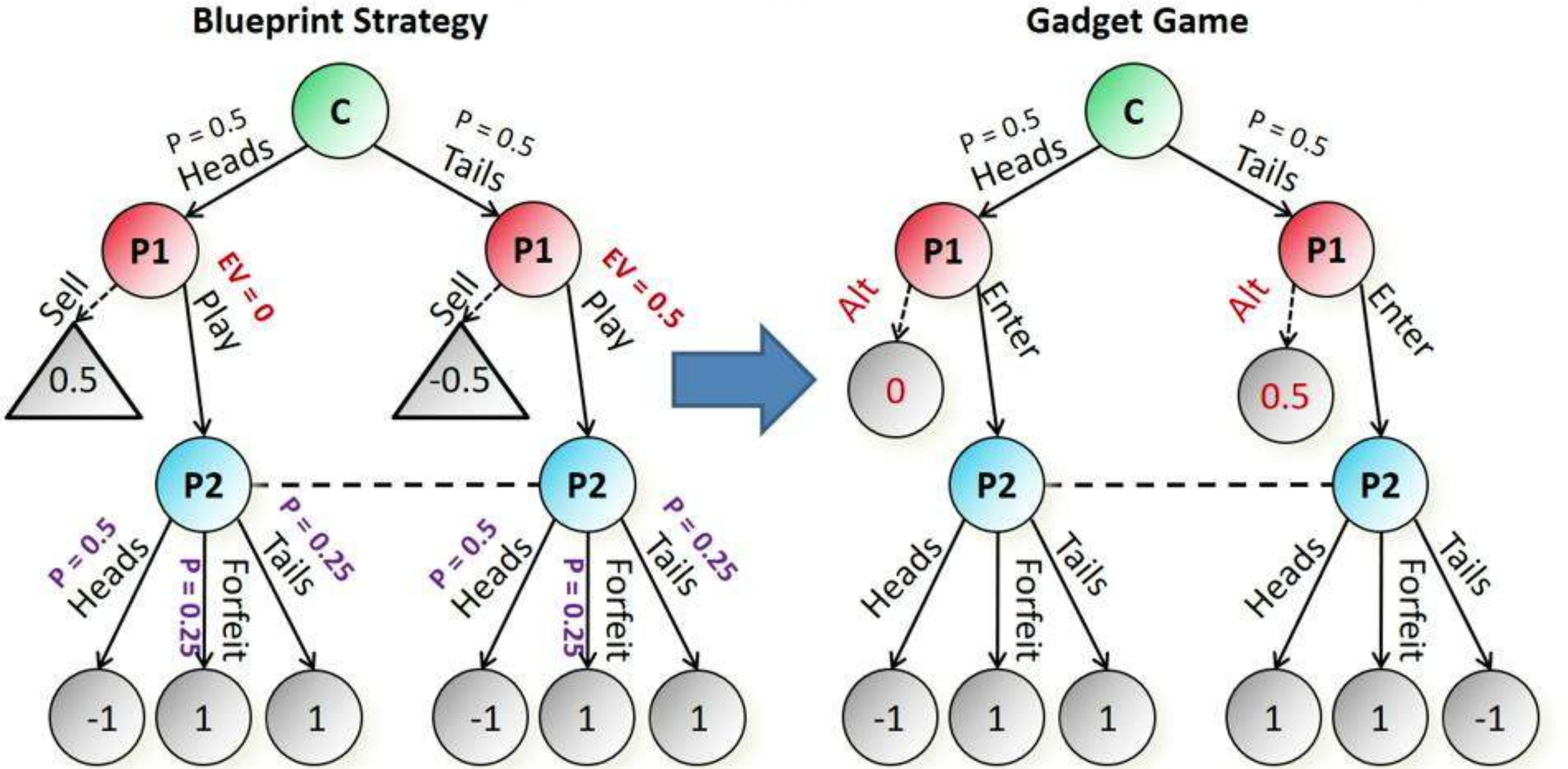


Maxmargin refinement [Moravcik et al. AAAI 2016]

Similar to Re-solve, but punishes P1 as much as possible for choosing Enter rather than Alt

$Margin_{Heads} = EV[Alt_{Heads}] - EV[Enter_{Heads}]$

Maximizes the minimum margin (Re-solve simply attempts to make all margins nonnegative)

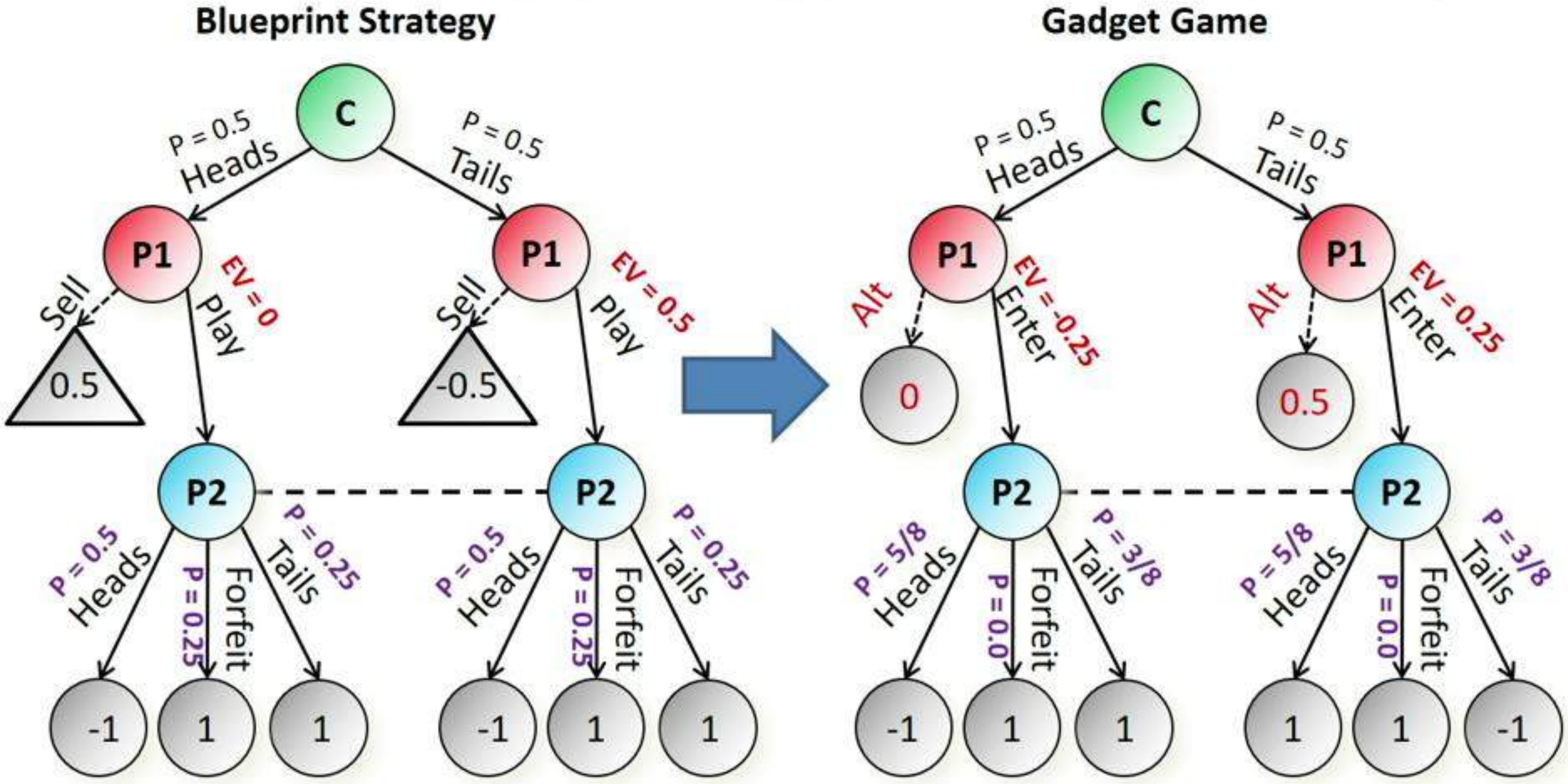


Maxmargin refinement [Moravcik et al. AAAI 2016]

Similar to Re-solve, but punishes P1 as much as possible for choosing Enter rather than Alt

$Margin_{Heads} = EV[Alt_{Heads}] - EV[Enter_{Heads}]$

Maximizes the minimum margin (Re-solve simply attempts to make all margins nonnegative)

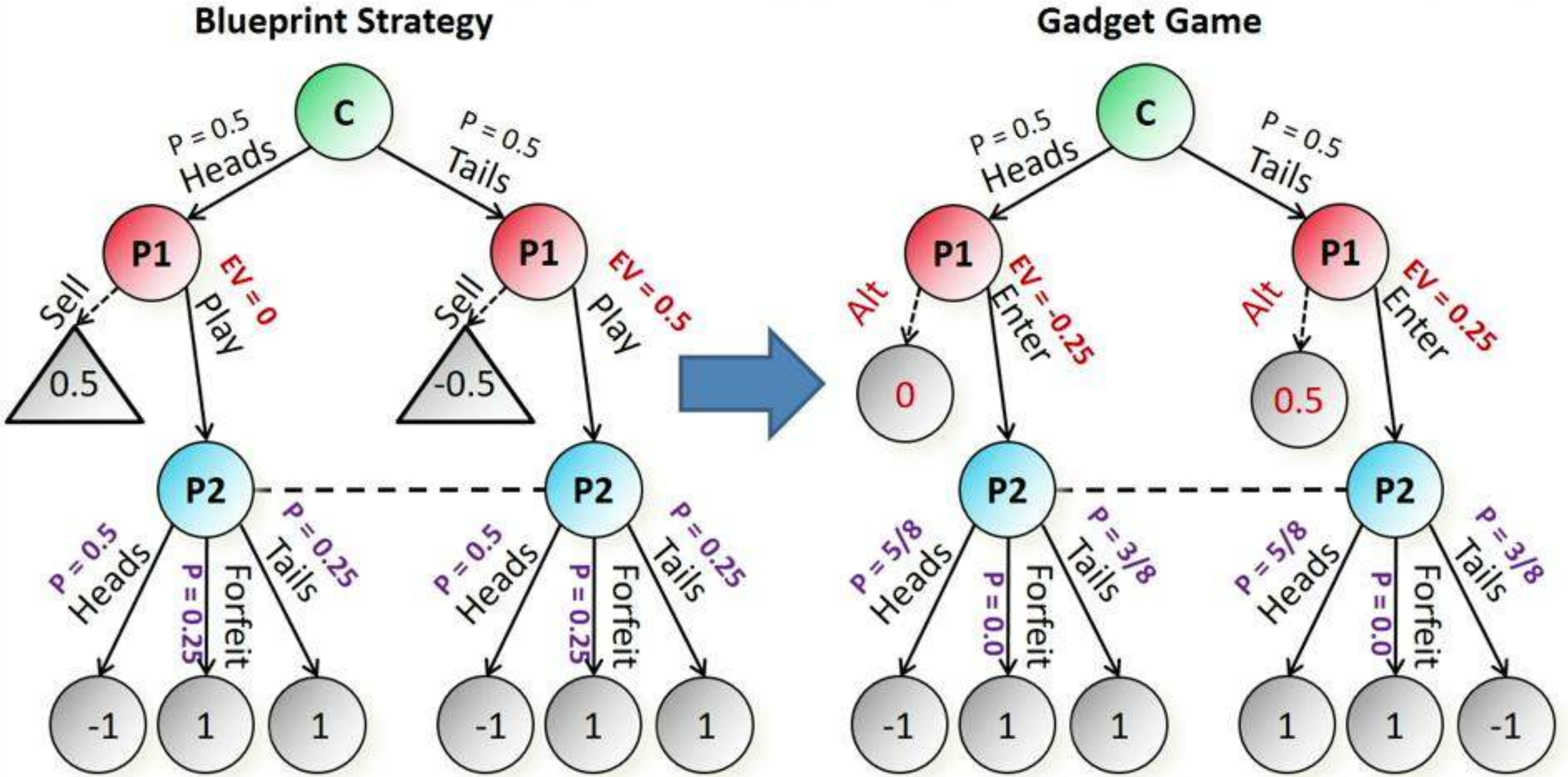


Maxmargin refinement [Moravcik et al. AAAI 2016]

Similar to Re-solve, but punishes P1 as much as possible for choosing Enter rather than Alt

$Margin_{Heads} = EV[Alt_{Heads}] - EV[Enter_{Heads}]$

Maximizes the minimum margin (Re-solve simply attempts to make all margins nonnegative)

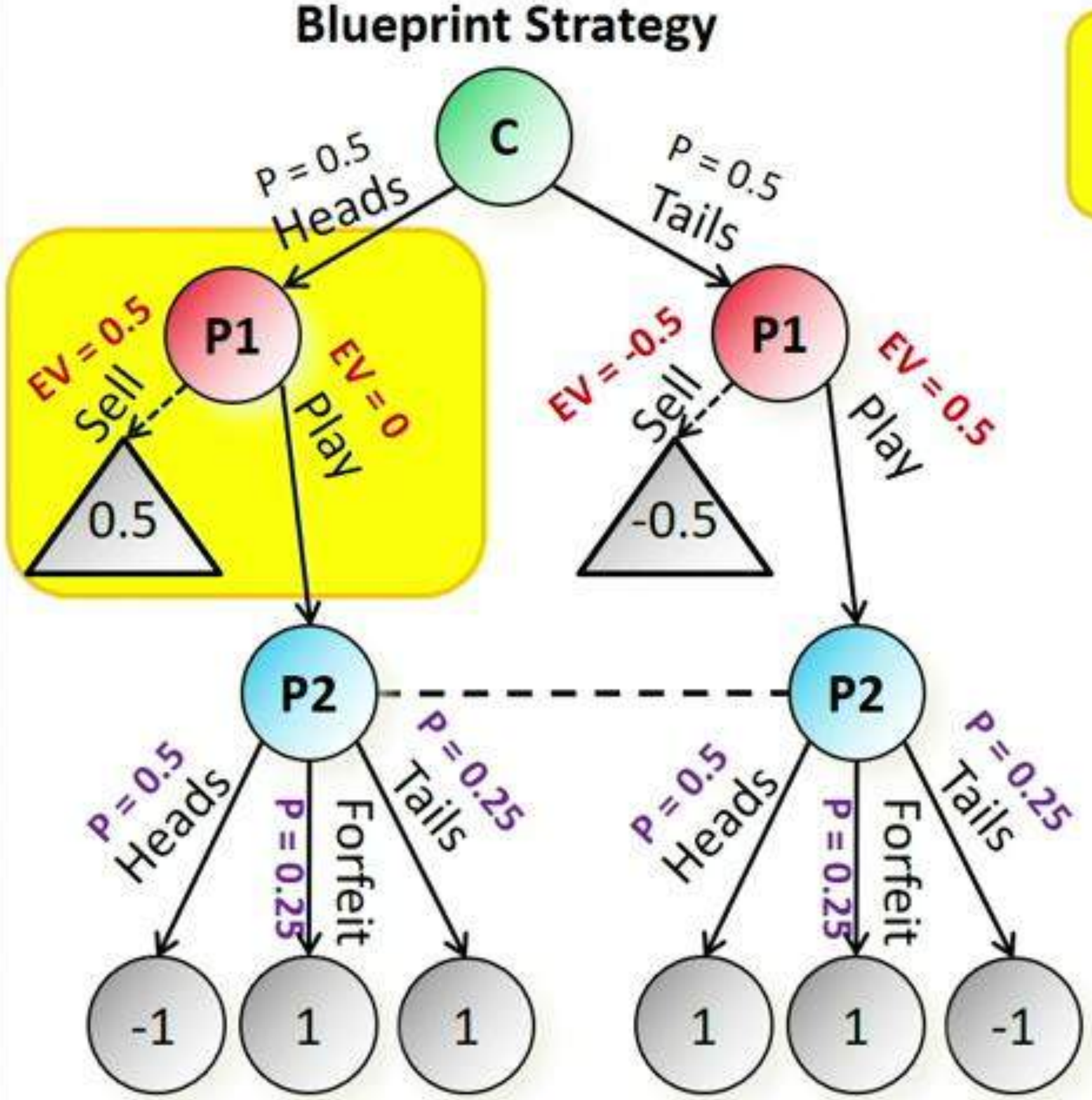


Problem: While we focus on reducing P1's EV for Heads in the subgame to -0.25, P1 can just Sell for 0.5 in Heads

Reach-maxmargin refinement:

(solving a single subgame here)

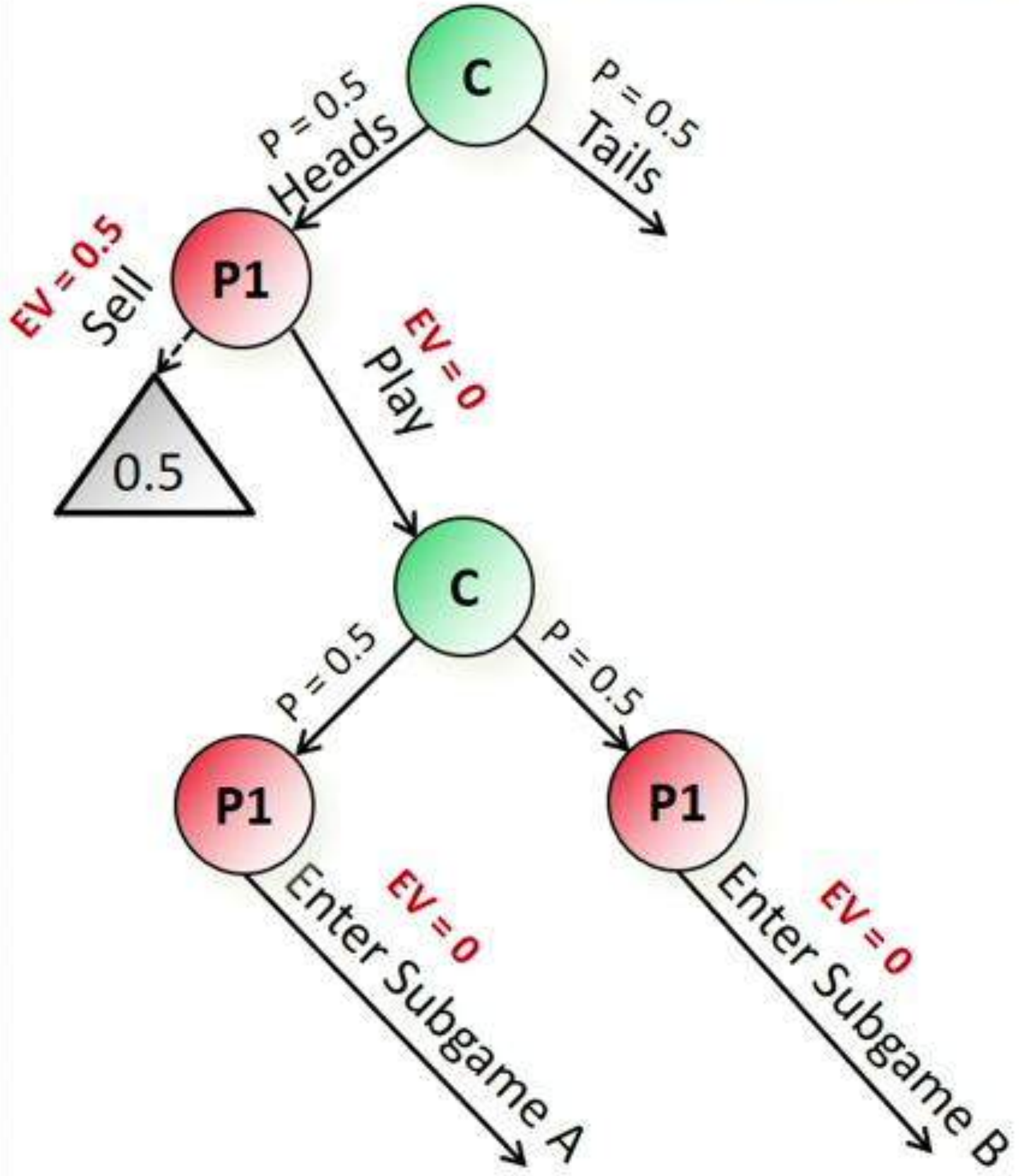
[Brown & Sandholm AAAI-17 workshop, NIPS-17, Science-17; related to Jackson AAAI-15 workshop]



- If P1 chooses Play following Heads, P1 is ***gifting*** us 0.5
- So, in Gadget Game we can increase the alternative payoff following Heads by 0.5, because choosing Play would still be a mistake for P1 there
- Thus the Gadget Game solver focuses on reducing P1's EV for other types she may have

Reach-maxmargin refinement: multiple subgames

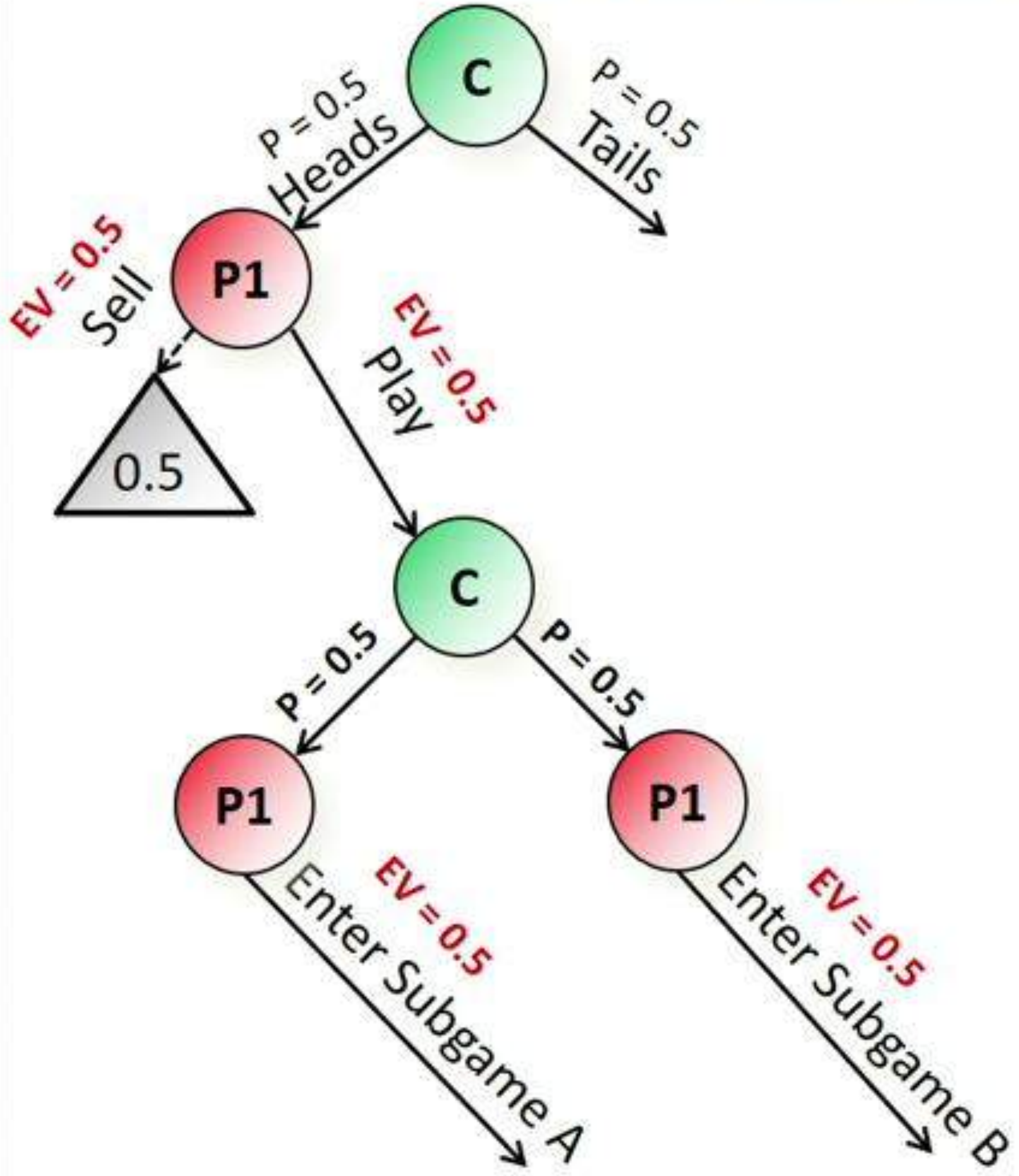
[Brown & Sandholm AAAI-17 workshop, NIPS-17, Science-17]



- If multiple subgames are refined, off-path EVs might not remain constant
- Solution: split gifts among subgames by probability subgame is reached

Reach-maxmargin refinement: multiple subgames

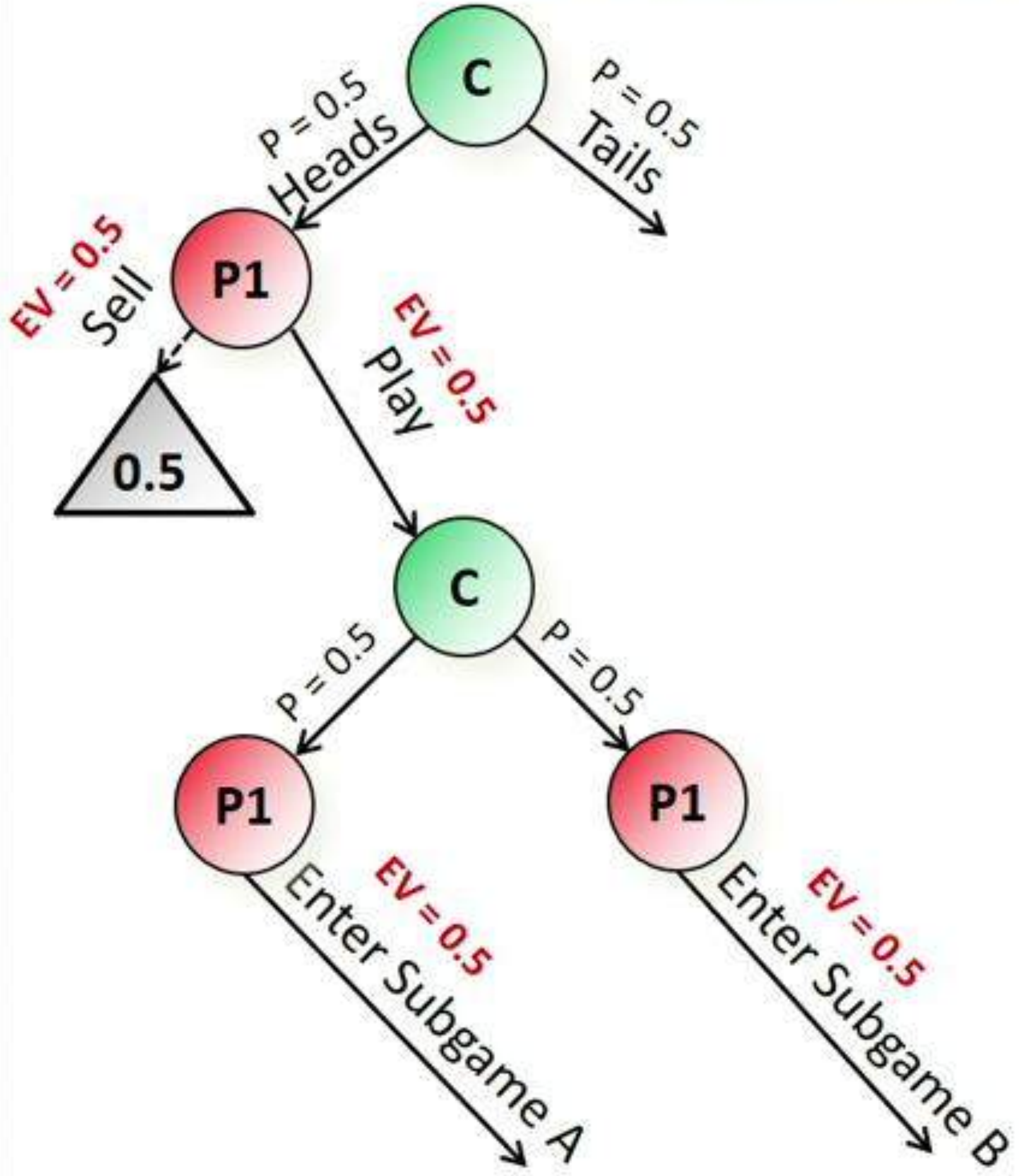
[Brown & Sandholm AAAI-17 workshop, NIPS-17, Science-17]



- If multiple subgames are refined, off-path EVs might not remain constant
- Solution: split gifts among subgames by probability subgame is reached

Reach-maxmargin refinement: multiple subgames

[Brown & Sandholm AAAI-17 workshop, NIPS-17, Science-17]

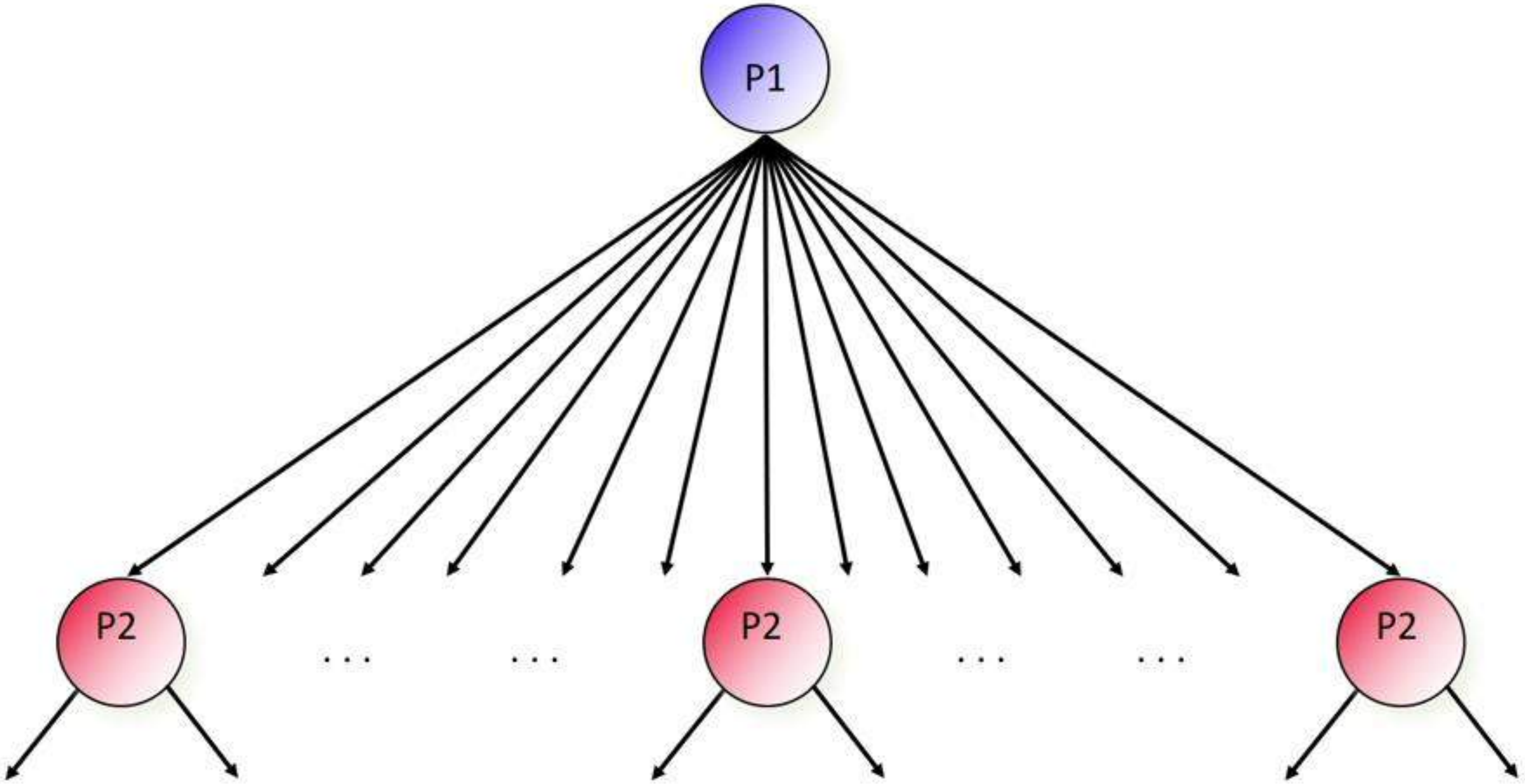


- Gifts might not be as large as we thought, because the subgames they come from will be improved
- Solution: substitute a lower bound on the gift
  - In practice, this can be an estimate

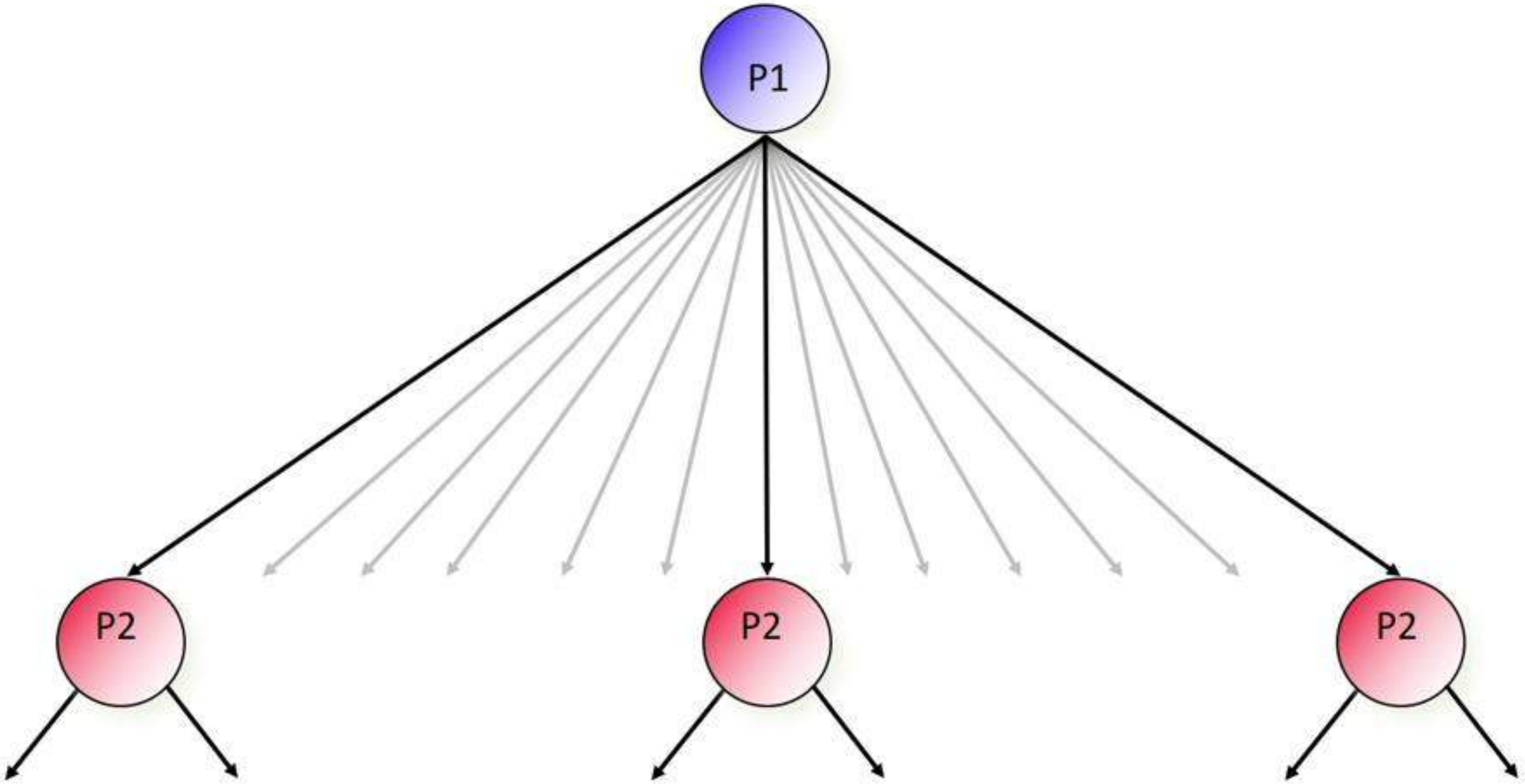
New ideas in subgame solver

- Provably *safe* subgame solving taking into account opponent's mistakes in the hand so far
- Off-tree actions & nested subgame solving
- Subgame solving starts much earlier
- No card abstraction in the subgame
- Changed our action abstraction between hands

Action abstraction

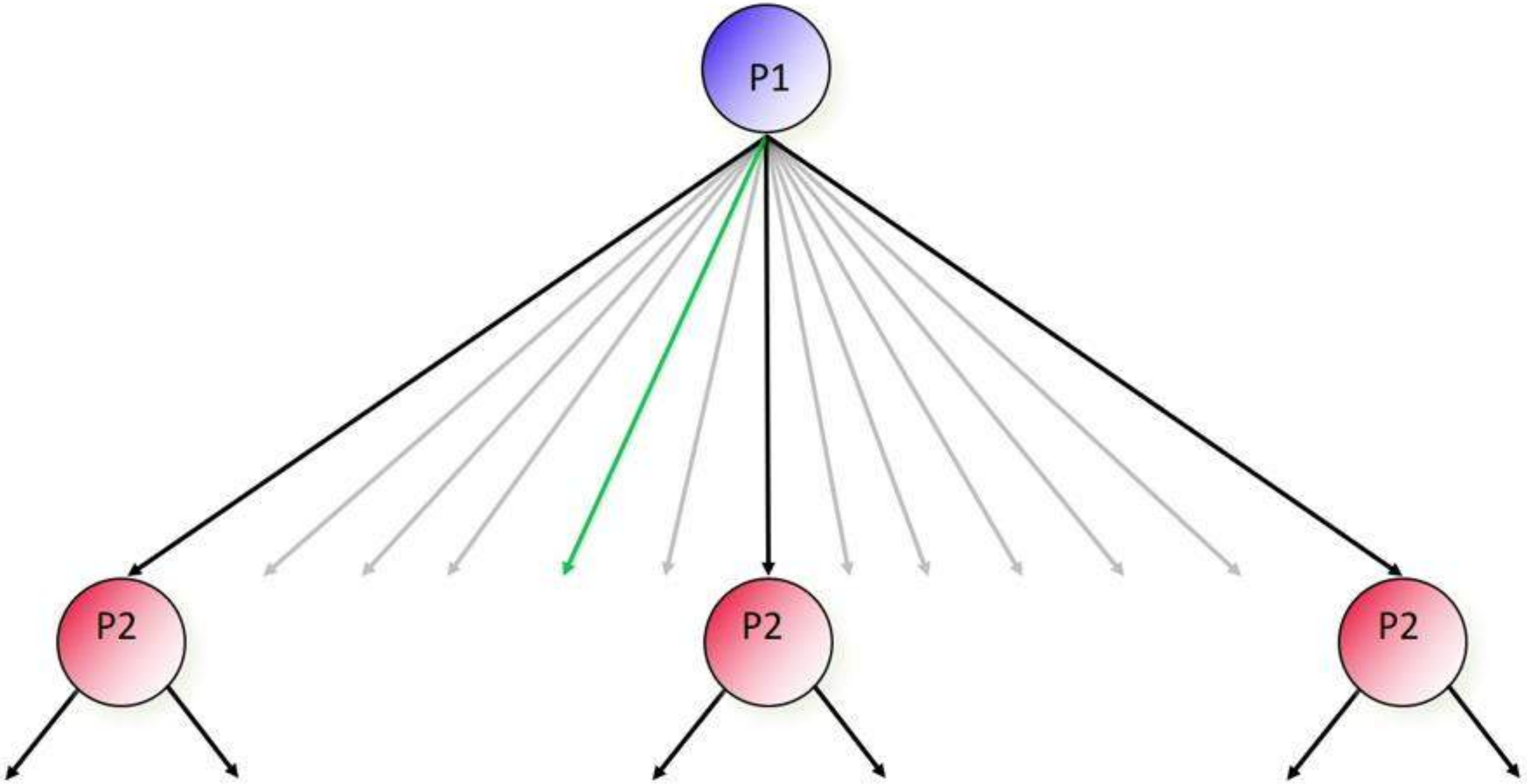


Action abstraction



[Gilpin et al. AAMAS-08], [Hawkin et al. AAAI-11, AAAI-12], [Brown & Sandholm AAAI-14]

Action translation

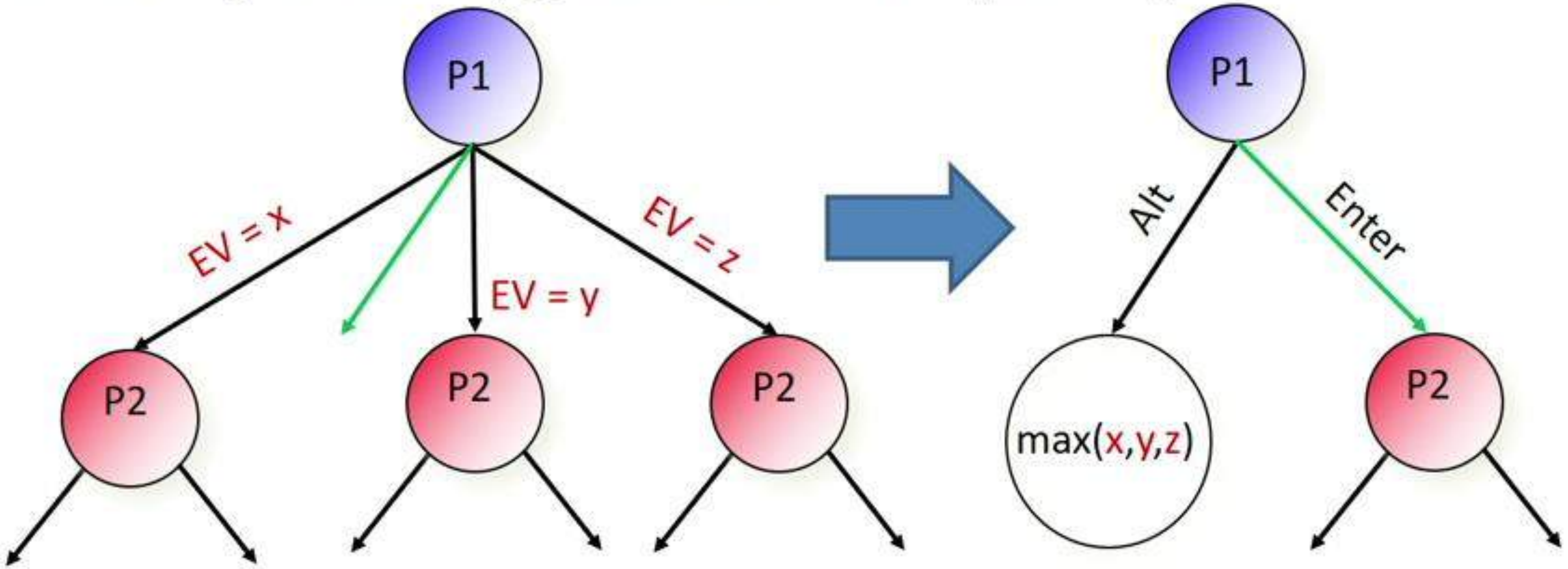


[Gilpin et al. AAMAS-08], [Schnizlein et al. IJCAI-09], [Ganzfried & Sandholm IJCAI-13]

Subgame solving even with "off-tree" actions

[Brown & Sandholm AAAI-17 workshop, arXiv, NIPS-17]

- Idea: Solve a subgame in real time (e.g., for the off-tree action that opponent took)



- **Theorem 1.** (even with hidden actions) Let Δ be such that there exists a Nash equilibrium where each alt payoff for Player 1 is within Δ of the Nash equilibrium EV at that point for Player 1. Then our Reach-Maxmargin subgame-solving technique yields a 2Δ-minmax strategy for Player 2 for the full game.
- **Theorem 2.** (even with hidden actions) If Player 1's alt payoffs are computed from a best response (in the full game) to Player 2's blueprint strategy, then the strategy formed by our Reach-Maxmargin subgame-solving technique has no greater regret for Player 2 in the full game than Player 2's blueprint strategy.
- ***"Nested subgame solving":*** can be reapplied for every subsequent opponent (e.g., off-tree) action
  - **Theorem 1'.** Player 2's strategy is a 2Δ * (number of reapplications within one play-through of the game)-minmax strategy.
  - **Theorem 2'.** Exactly like Theorem 2.

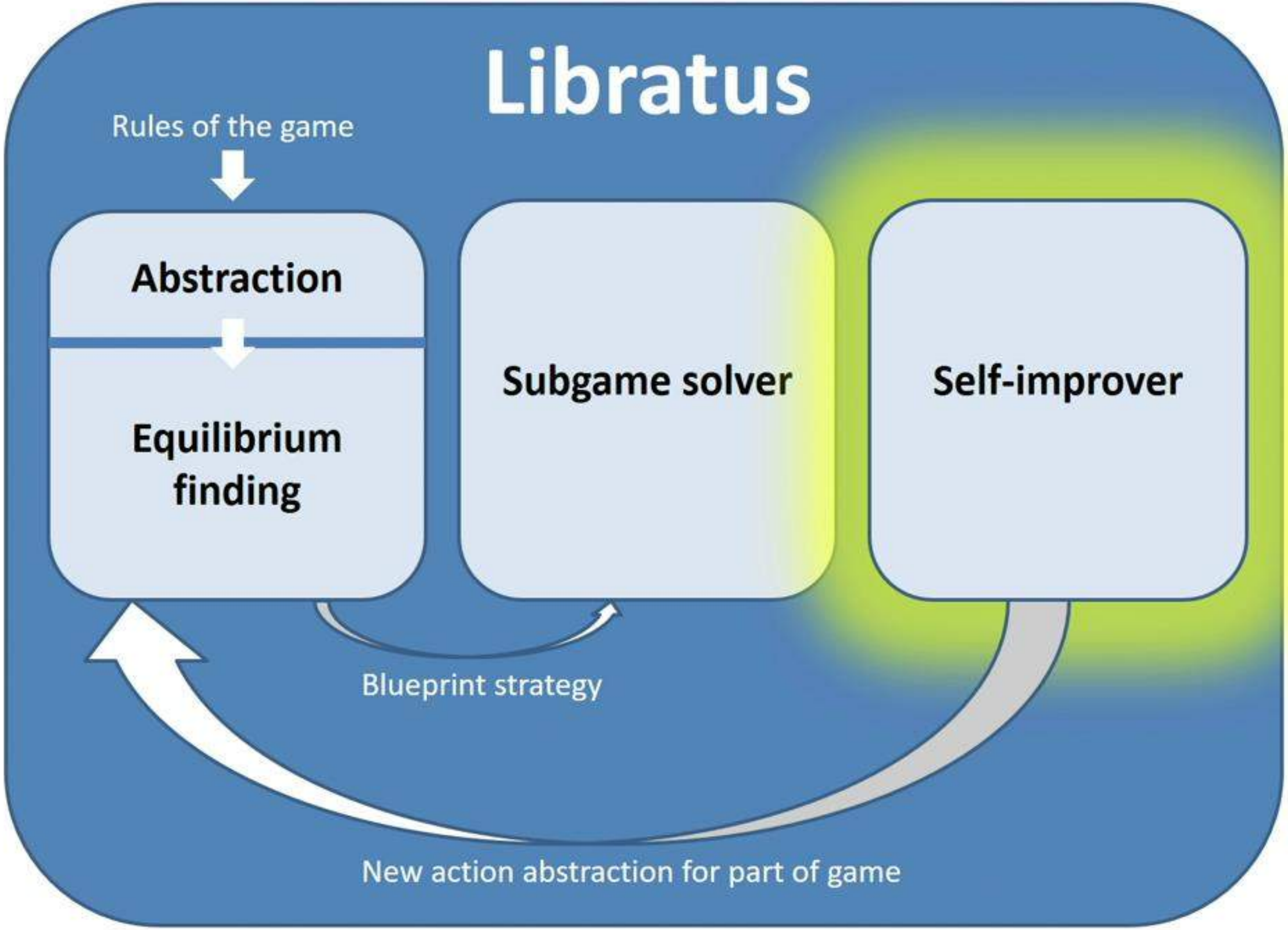
Experiments on medium-sized games

- Our best **reach subgame solving** technique has **3x** less exploitability than the best prior safe subgame-solving technique
- **Nested reach subgame solving** is **12x** less exploitable than best action-mapping technique

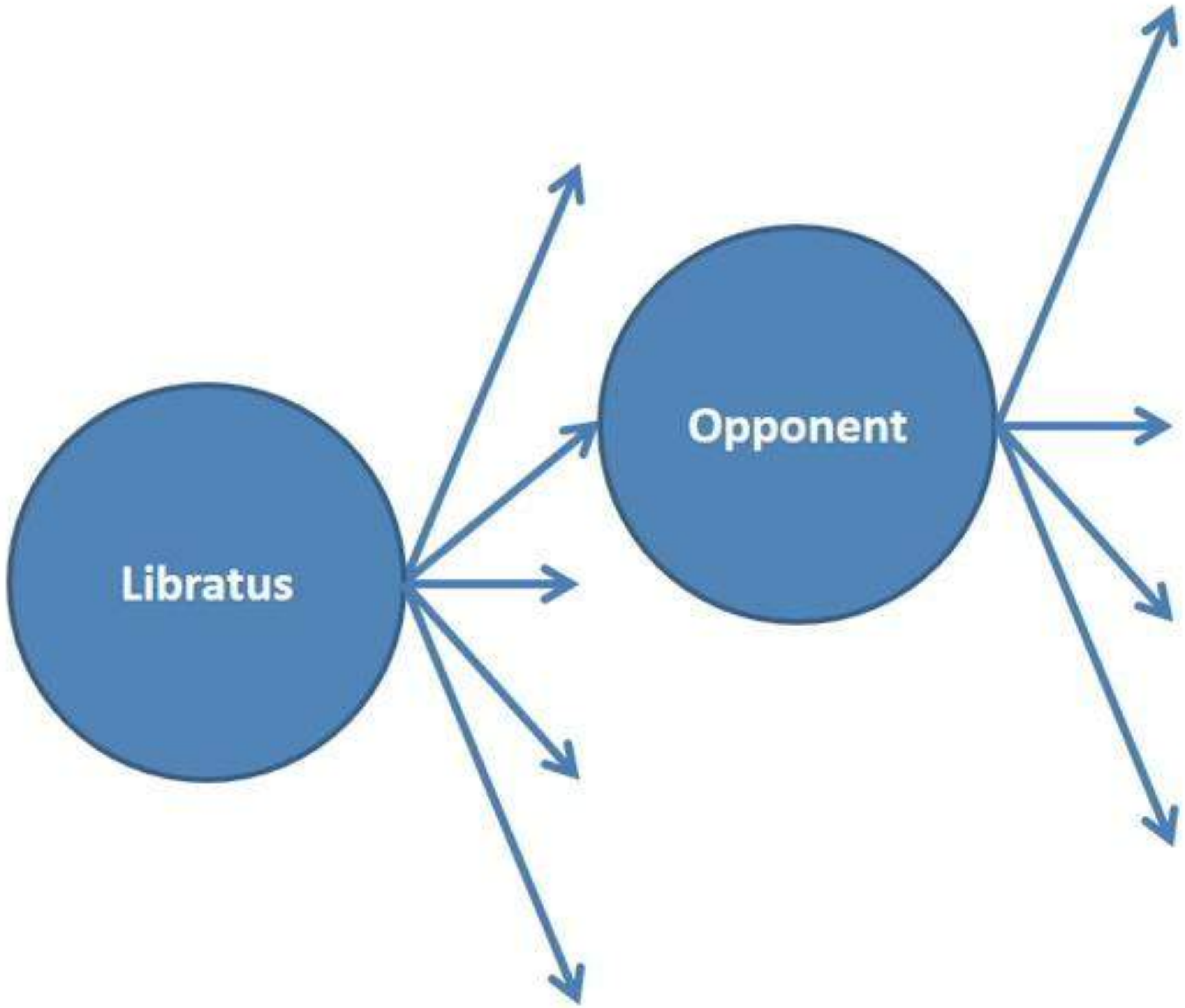
New ideas in subgame solver

- Provably *safe* subgame solving taking into account opponent's mistakes in the hand so far
- Off-tree actions & nested subgame solving
- Subgame solving starts much earlier
- No card abstraction in the subgame
- Changed our action abstraction between hands

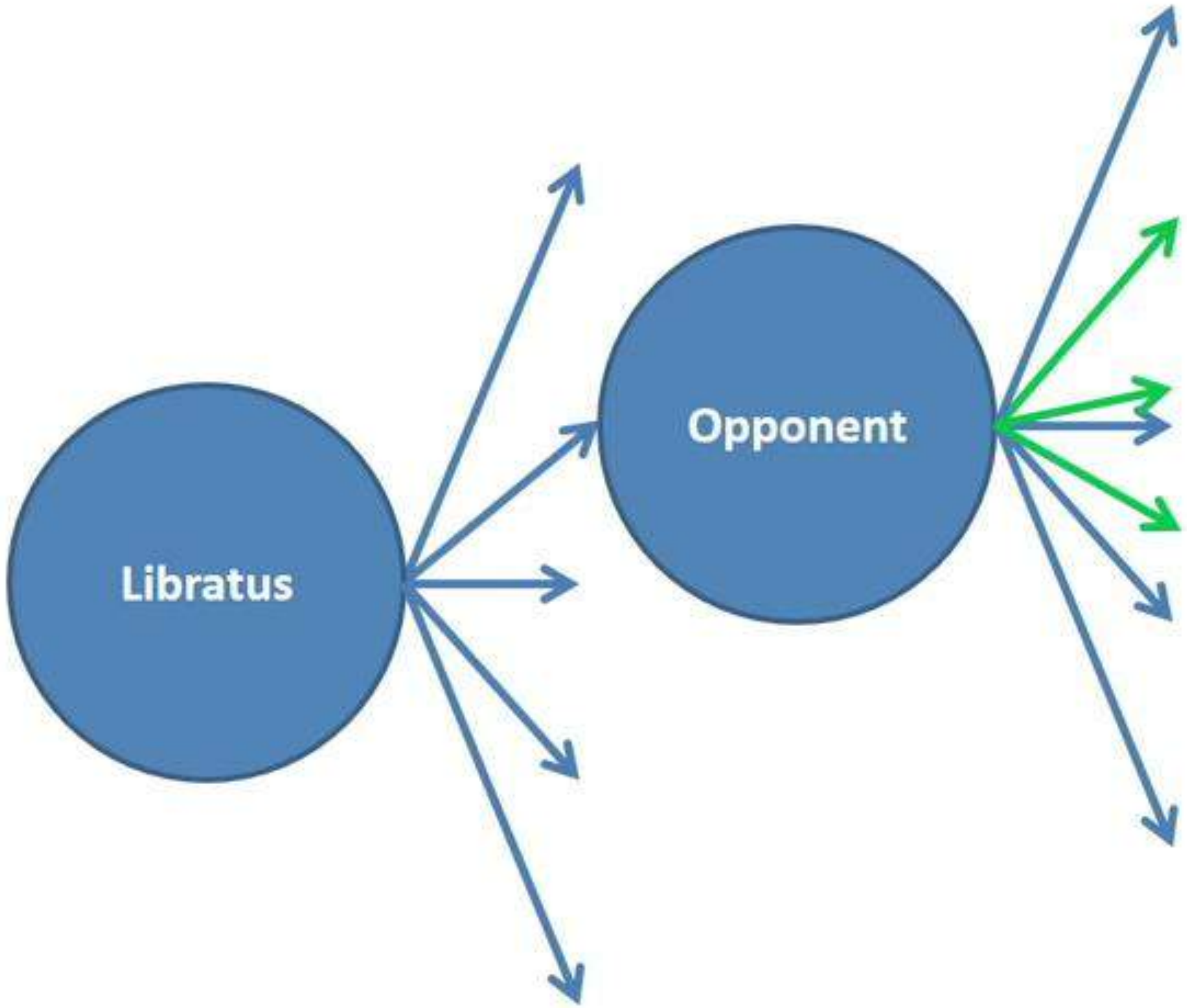
Libratus's "balance" and use of "blockers"


Libratus
Rules of the game
Abstraction
Equilibrium finding
Subgame solver
Self-improver
Blueprint strategy
New action abstraction for part of game

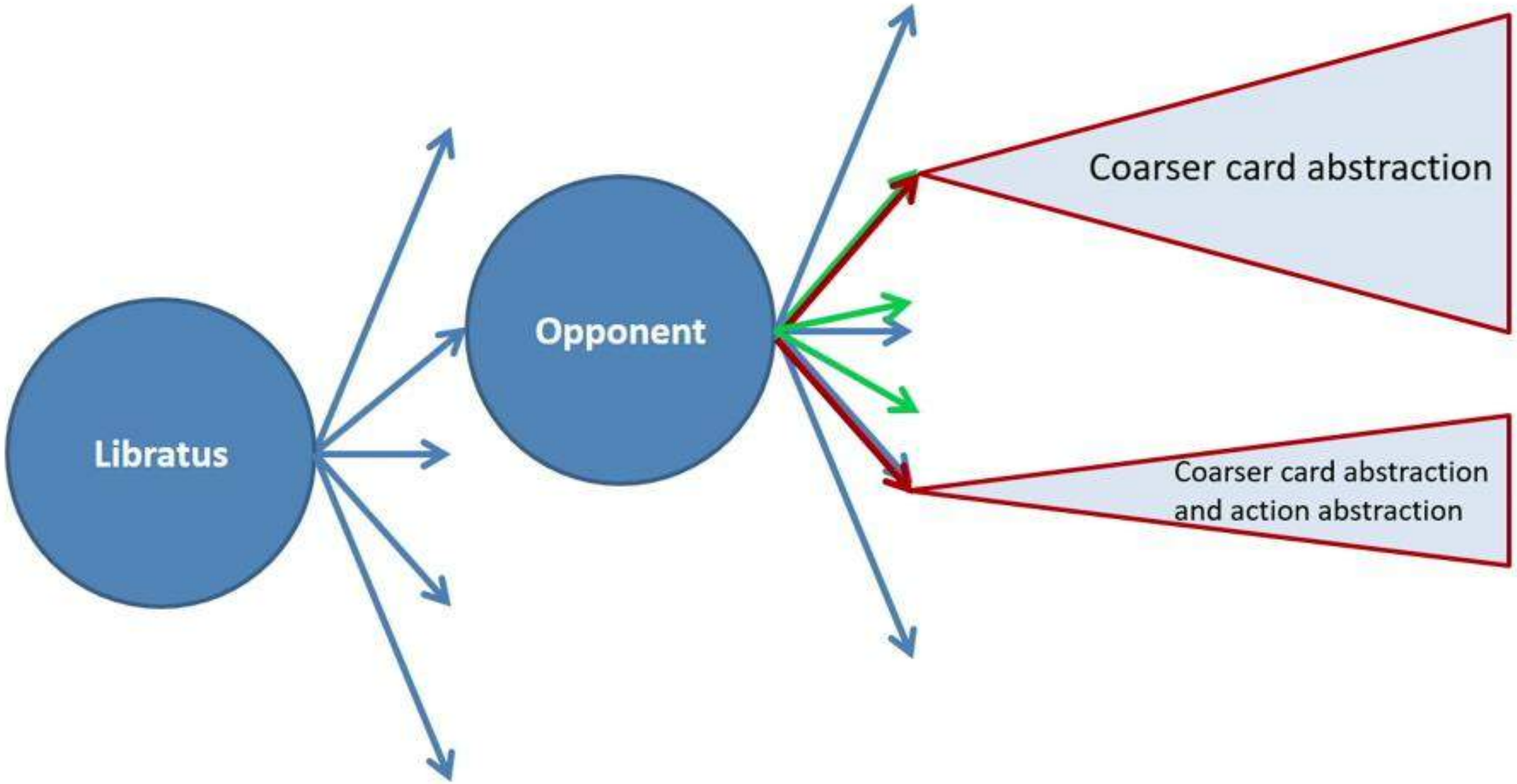
Filling holes in the action tree



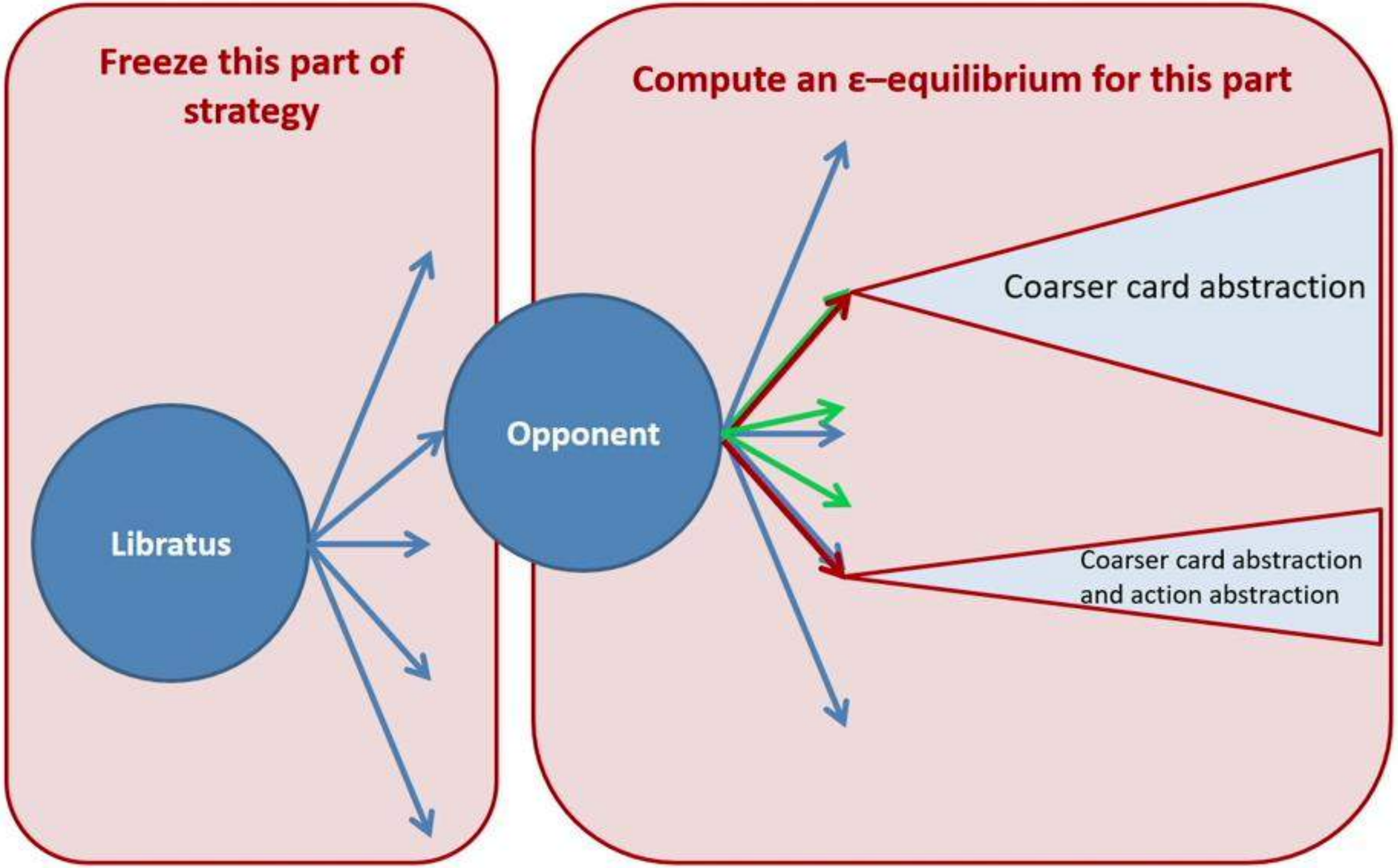
Filling holes in the action tree



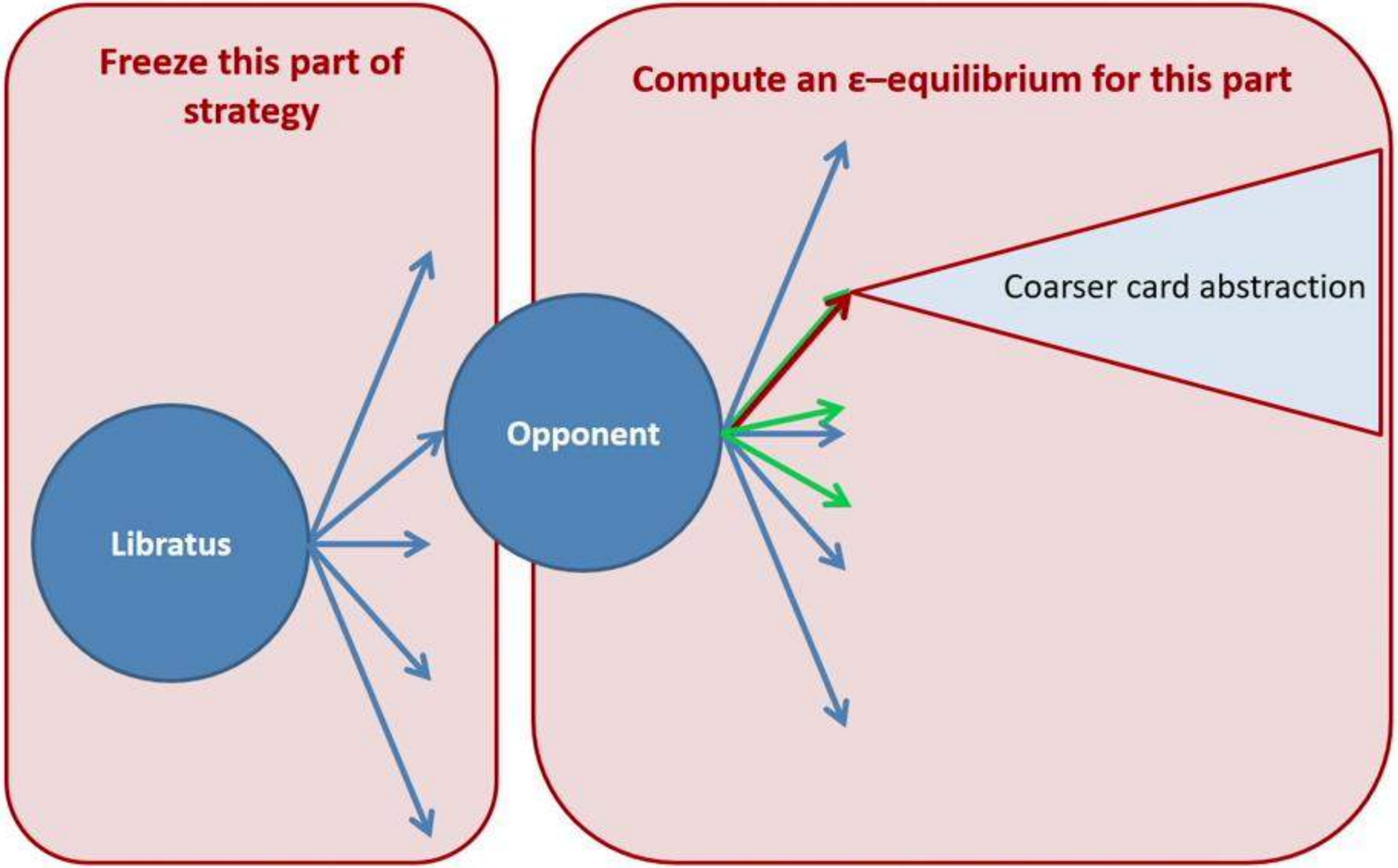
Filling holes in the action tree



Filling holes in the action tree



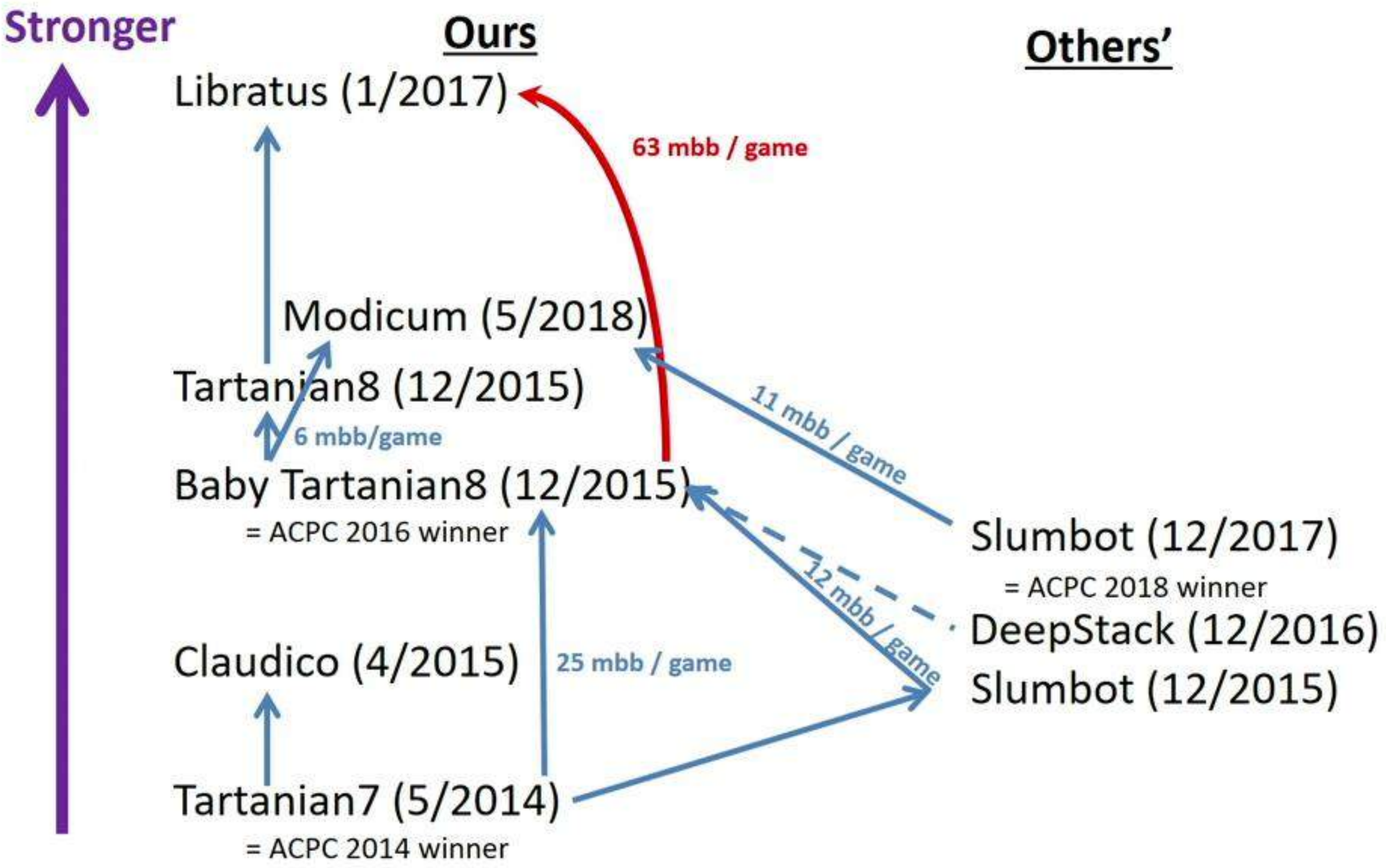
Filling holes in the action tree



We do this for top *k* holes

Libratus fixing its own weaknesses

Head-to-head strength of top AIs



Observations about Libratus's play

- Strengths:
  - Small bets & huge bets & huge all-ins
  - Multiple bet sizes in any one situation
  - "Limping", "donk betting"
  - "Perfect balance"
  - Mixed strategy
  - Probability distributions over players' hands; not just "range-based"
  - Near-perfect subgame play; "great use of blockers"
  - Different bet sizings used in subgames
- Weaknesses?
  - No opponent exploitation

Is safe (equilibrium) play timid/boring?

Some current research on this topic in my lab

- Lossy abstraction algorithms with bounds [Kroer & Sandholm EC-14, EC-16, AI^3 2018]
  - They also apply to modeling
  - To do: Unify with our best practical abstraction algorithms
- Simultaneous abstraction & equilibrium finding [Brown & Sandholm AAAI-15]
- General warm starting for CFR algorithms [Brown & Sandholm AAAI-16]
- New, fastest algorithm in CFR family [Brown & Sandholm manuscript]
- Regret decomposition framework [Farina, Kroer, Sandholm manuscript]
- Equilibrium-finding algorithms by developing new theory of gradient-based techniques [Kroer *et al.* EC-15, EC-17, 2 manuscripts]
- Algorithms for equilibrium refinements [Farina, Kroer & Sandholm ICML-17], [Kroer, Farina & Sandholm IJCAI-17], [Farina, Gatti & Sandholm manuscript]
- Equilibrium finding for >2 agents [Berg & Sandholm AAAI-17]
- Exploration vs. exploitation vs. exploitability [Ganzfried & Sandholm TEAC-15]
- Depth-limited solving in imperfect-info games [Brown & Sandholm arXiv-18]

Applying this new technology capability

Most real-world settings are imperfect-information games

STRATEGIC MACHINE, INC.

Commercial applications:

- Strategic pricing & strategic product portfolio optimization
- Gaming (incl. video games), training, streams, bot detection, …
- Automated negotiation, negotiation support, other strategic augmentation
- Electricity markets
- Finance: Strategic execution of trades, trading large blocks, illiquid markets, OTC, strategic portfolio construction, …
- Investment banking
- Political campaigns (e.g., media spending)
- Business strategy
- Bidding & auction design
- Strategic market segmentation
- Interaction between vehicles & semi-autonomous fleets
- Acquisition strategy (e.g., movie rights: portfolio construction and negotiation strategy)
- Sports
- Steering evolution and biological adaptation [Sandholm 2012, AAAI-15 SMT Blue Skies]
  - E.g., for medical treatment planning [Kroer & Sandholm IJCAI-16] and synthetic biology

Government applications

STRATEGY ROBOT, INC.

For the first time, doesn't assume a naïve "Red".
Unlike in simulation and current war gaming, Red's strategy is not input but rather output!

- Strategic, operative, and tactical level planning in adversarial military settings
- Force planning & acquisition
- War gaming
- Training
- Intelligence
- National security, trade, and diplomatic strategies
- Next-generation cybersecurity
  - Wireless jamming [DeBruhl et al.]
  - Zero-day vulnerability discovery & play-out
  - Operating systems, sequential screening, honeypotting, …
- Physical security
- Automated negotiation & negotiation support
- Other strategic augmentation