University of Alberta
Computer Poker Research Group

Solving Large Extensive-Form Games Quicker

Richard Gibson

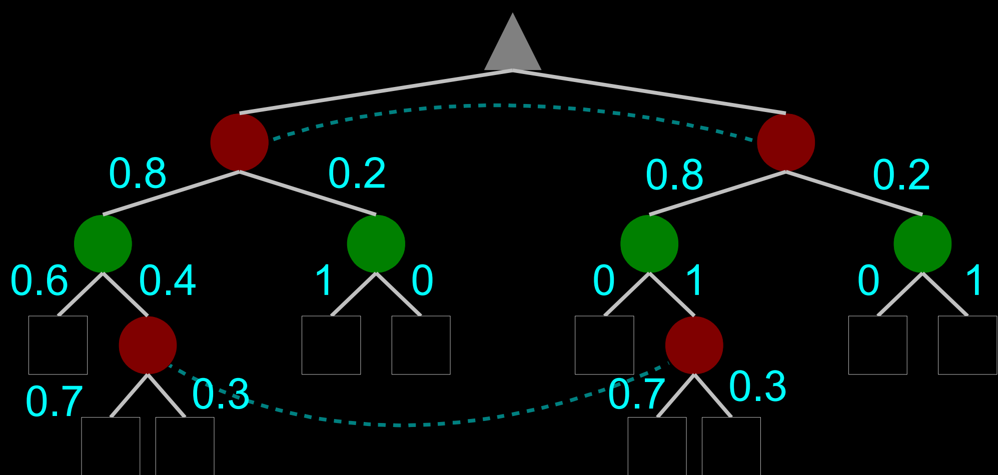
Joint work with:

Marc Lanctot
Neil Burch
Michael Johanson
Nolan Bard
Duane Szafron
Michael Bowling

Alberta Innovates
NSERC CRSNG
compute • calcul CANADA
WestGrid

What is this talk about?

We are interesting in “solving” large sequential decision making problems with imperfect information.



- Best known algorithm: CFR (Counterfactual Regret Minimization).
- Monte Carlo sampling variants of CFR have been shown to reduce computation time in many games.
  - Still takes many days of off-line computation.

This talk asks the question:

Can we solve games **faster**?

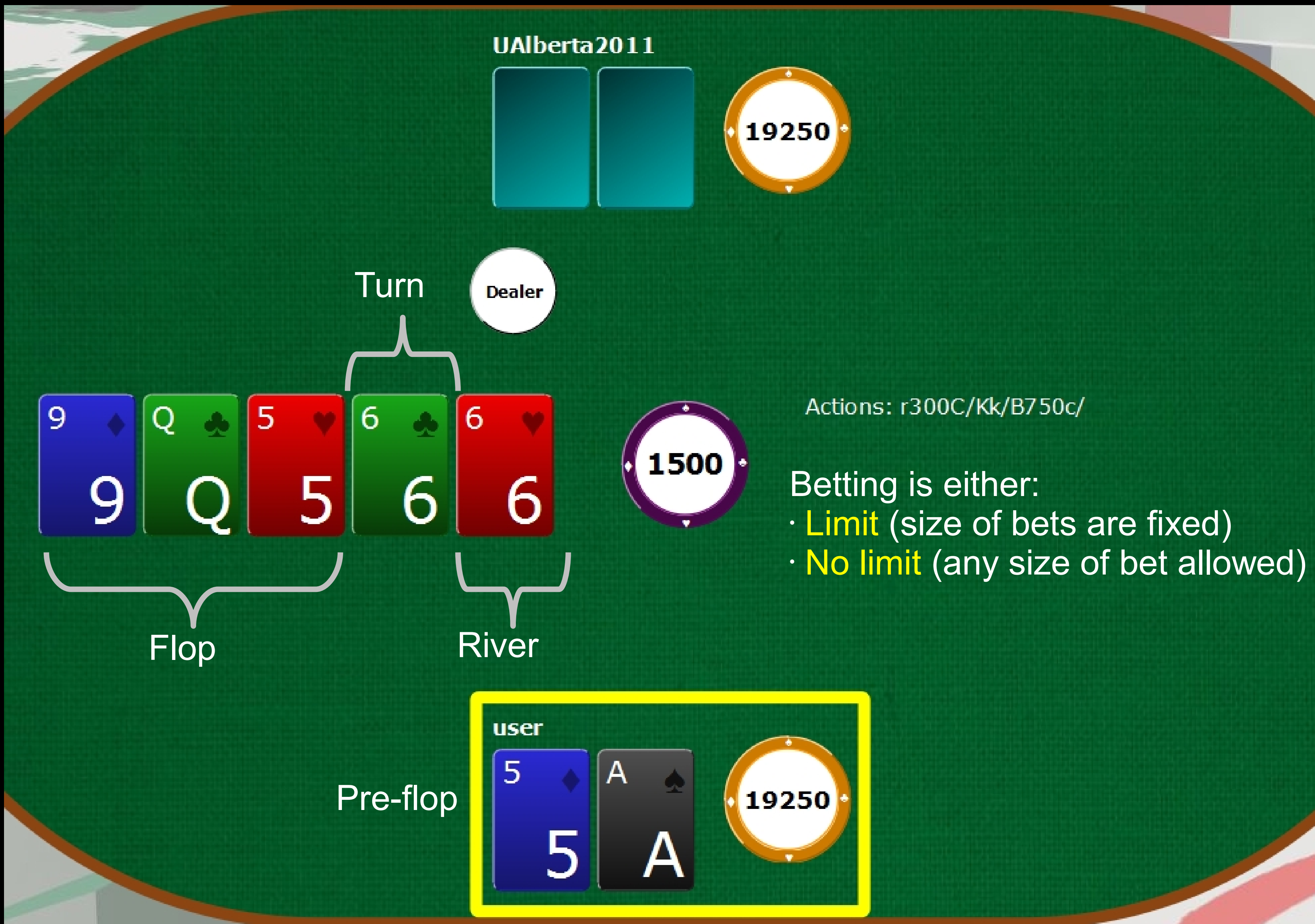
Outline of Talk

- Extensive-form Games
  - Examples
  - Terminology
  - Solution concepts
- Counterfactual Regret Minimization (CFR)
  - Base algorithm for solving extensive-form games
  - Older variants
- New, Faster CFR Variants
  - Probing
  - Public Chance Sampling
  - Average Strategy Sampling
- Conclusions and Future Work

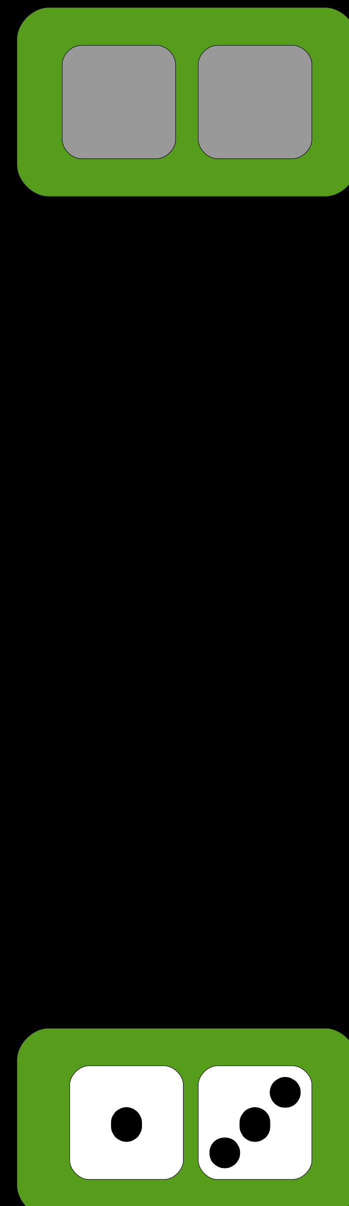
Outline of Talk

- Extensive-form Games
  - Examples
  - Terminology
  - Solution concepts
- Counterfactual Regret Minimization (CFR)
  - Base algorithm for solving extensive-form games
  - Older variants
- New, Faster CFR Variants
  - Probing
  - Public Chance Sampling
  - Average Strategy Sampling
- Conclusions and Future Work

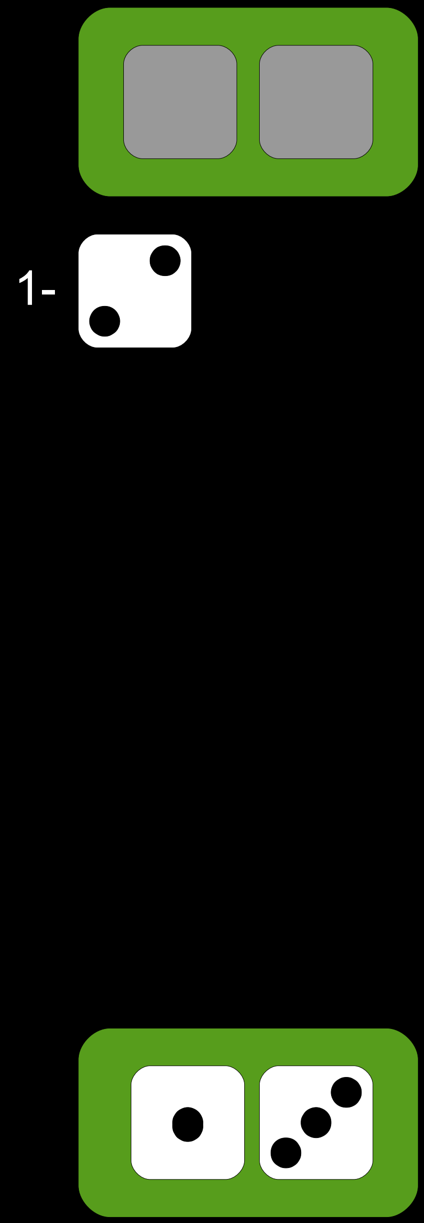
Texas Hold'em Poker



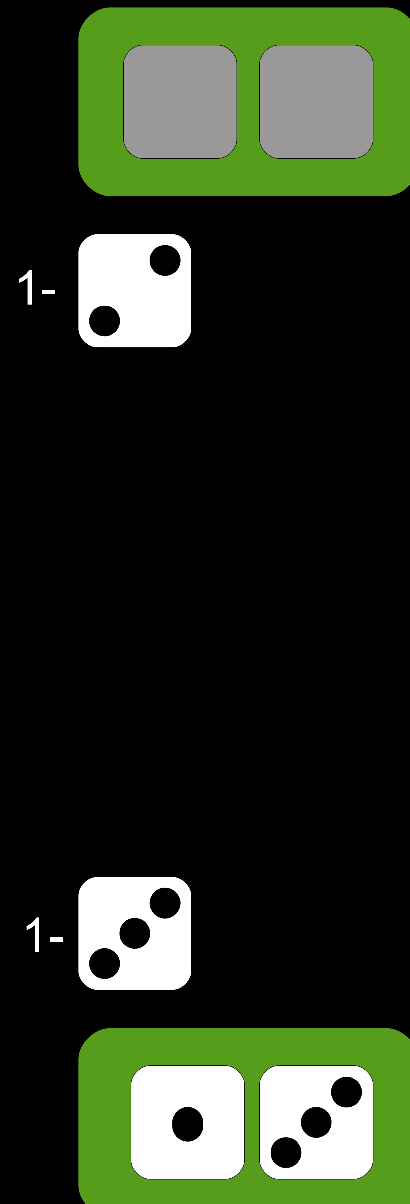
Bluff



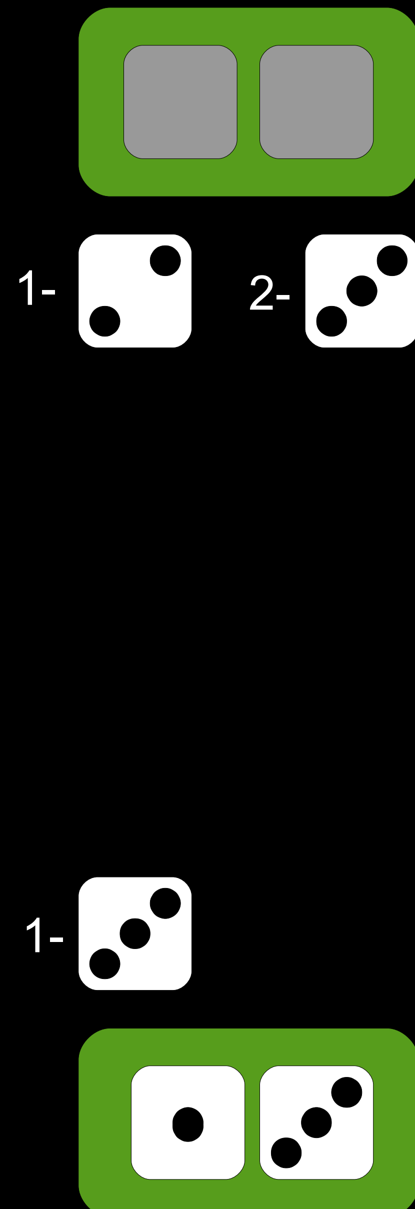
Bluff



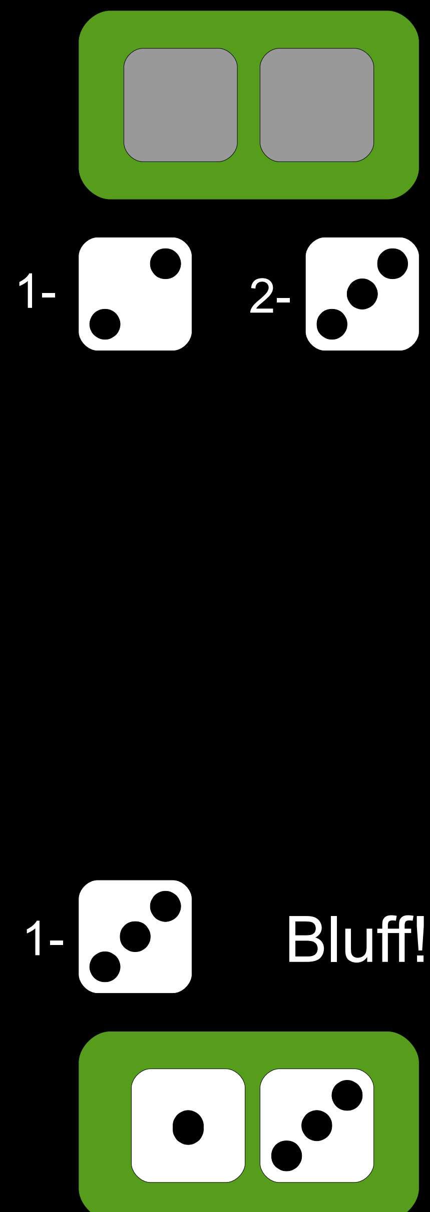
Bluff



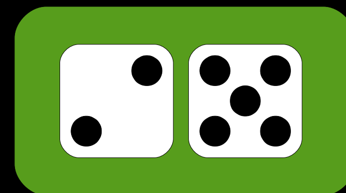
Bluff

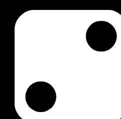
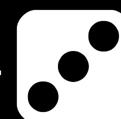


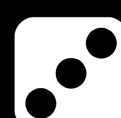
Bluff

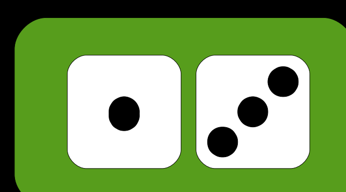


Bluff

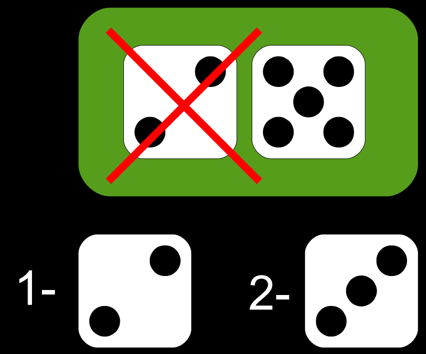


1-  2- 

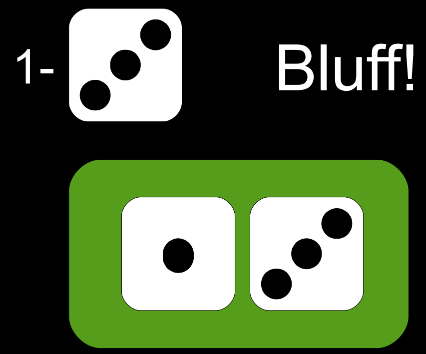
1-  Bluff!



Bluff



- Play continues until one player has no dice left and loses.
- Winner gets +1 utility, loser gets -1 utility.



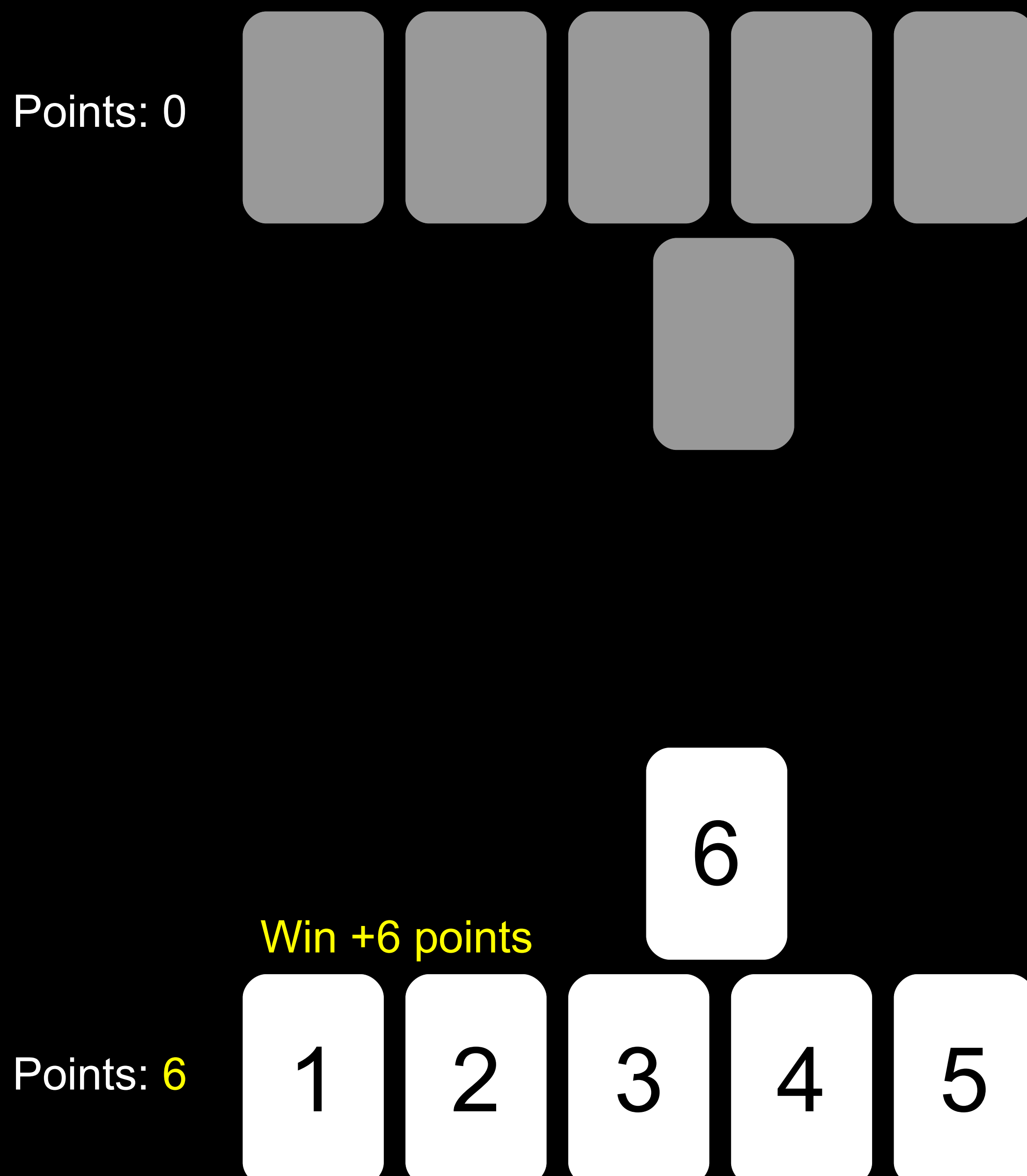
Goofspiel

Points: 0

Points: 0

1 2 3 4 5 6

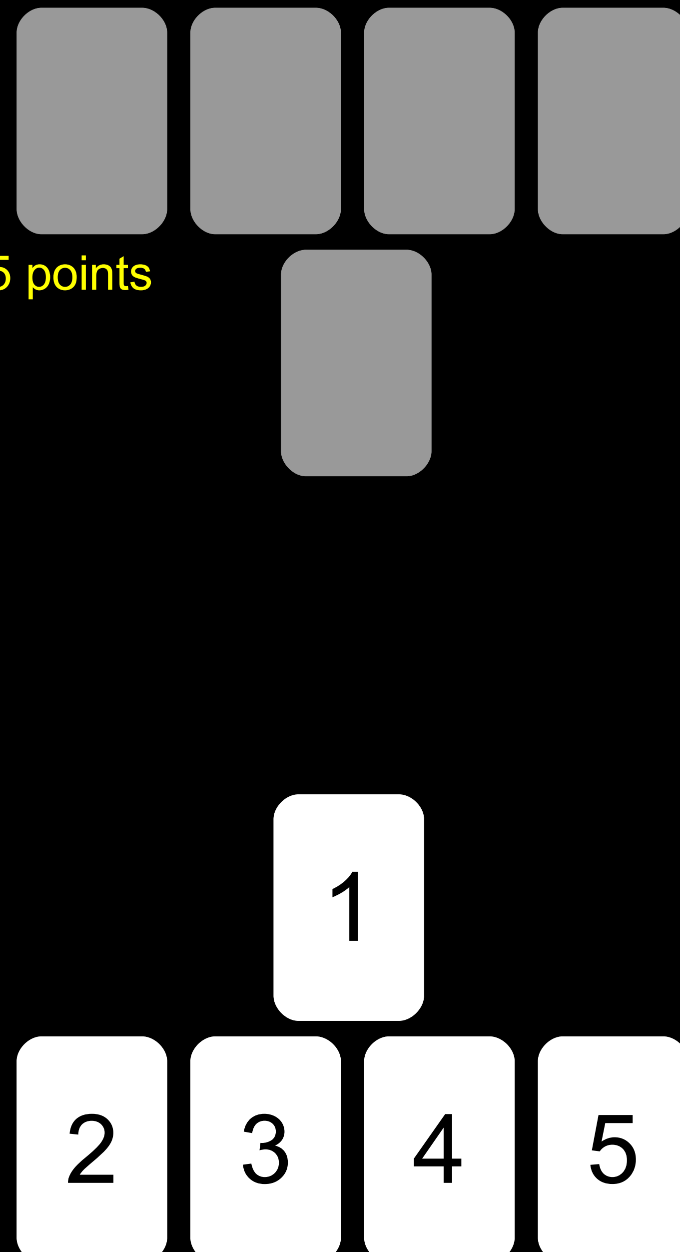
Goofspiel



Goofspiel

Points: 5

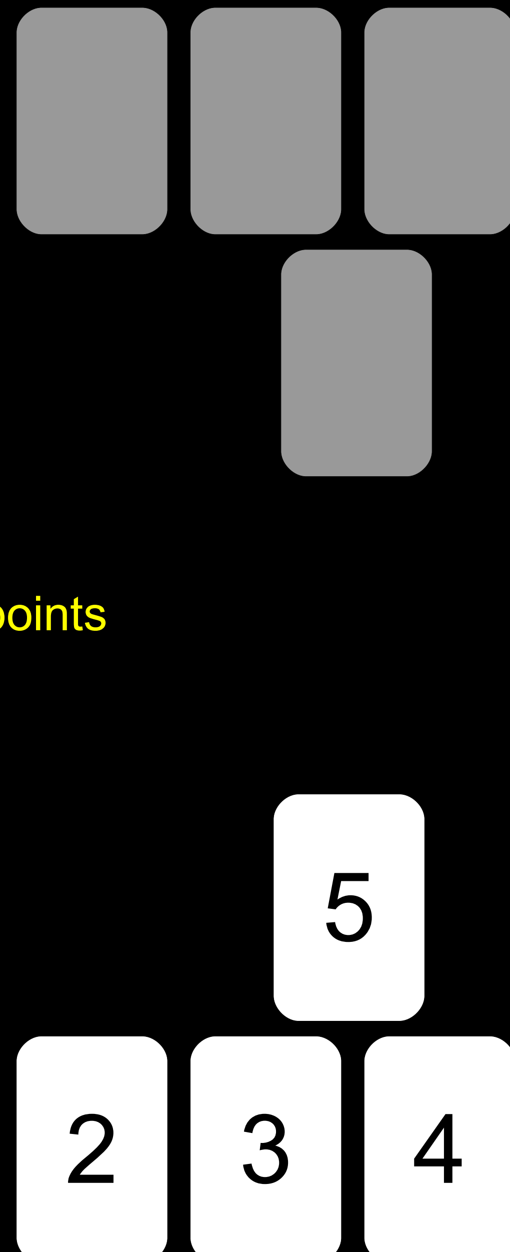
Win +5 points



Points: 6

Goofspiel

Points: 5



Tie 0 points

Points: 6

Goofspiel

Points: 5

4

Win +3 points

Points: 9

2 3

Goofspiel

Points: 5

Tie 0 points

2

3

Points: 9

Goofspiel

Points: 6

Win +1 point

3

Points: 9

Goofspiel

Points: 6

- The player with the most points at the end of the game wins.
- The winner receives +1 utility, loser receives -1 utility.

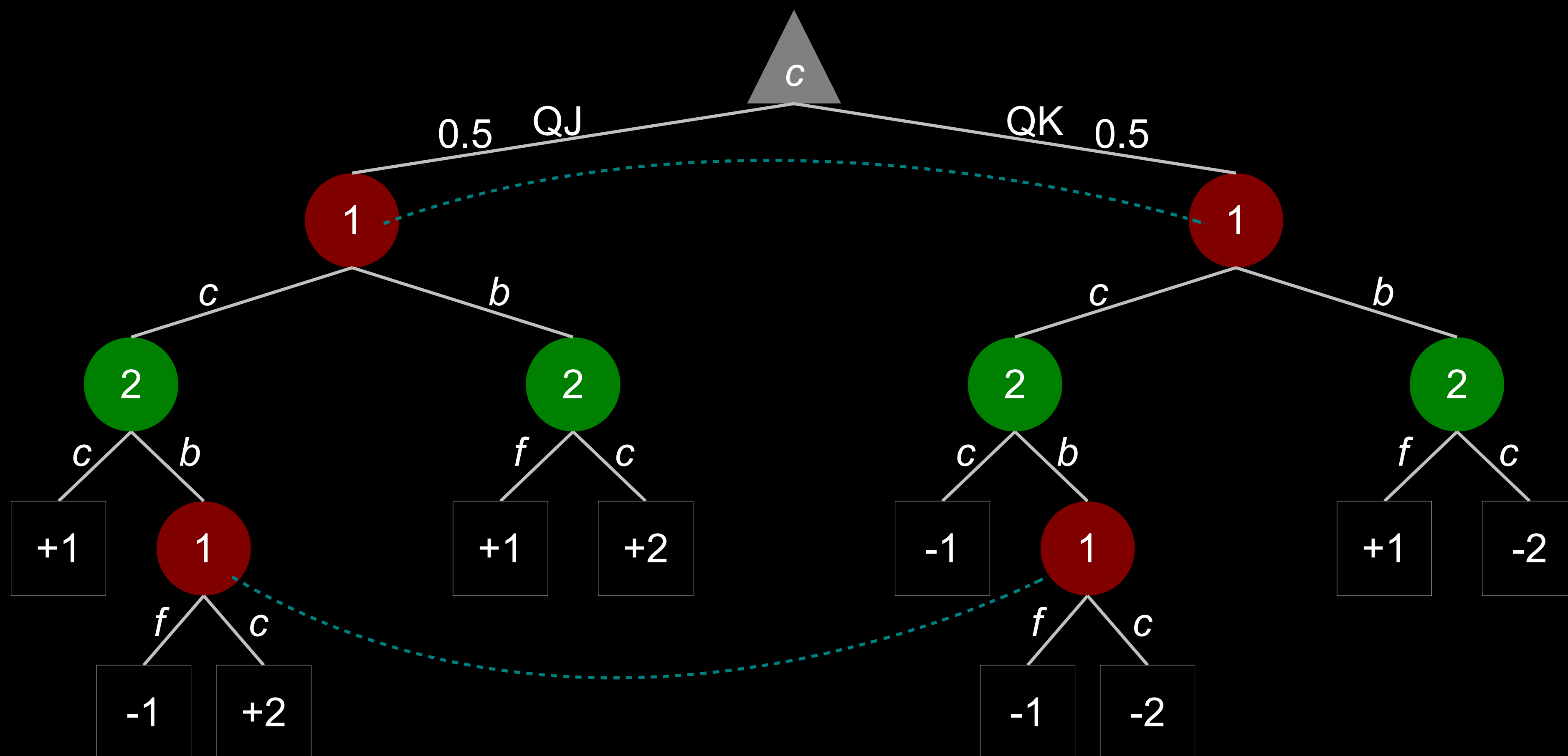
Points: 9

Winner!

Other Examples of Extensive Games

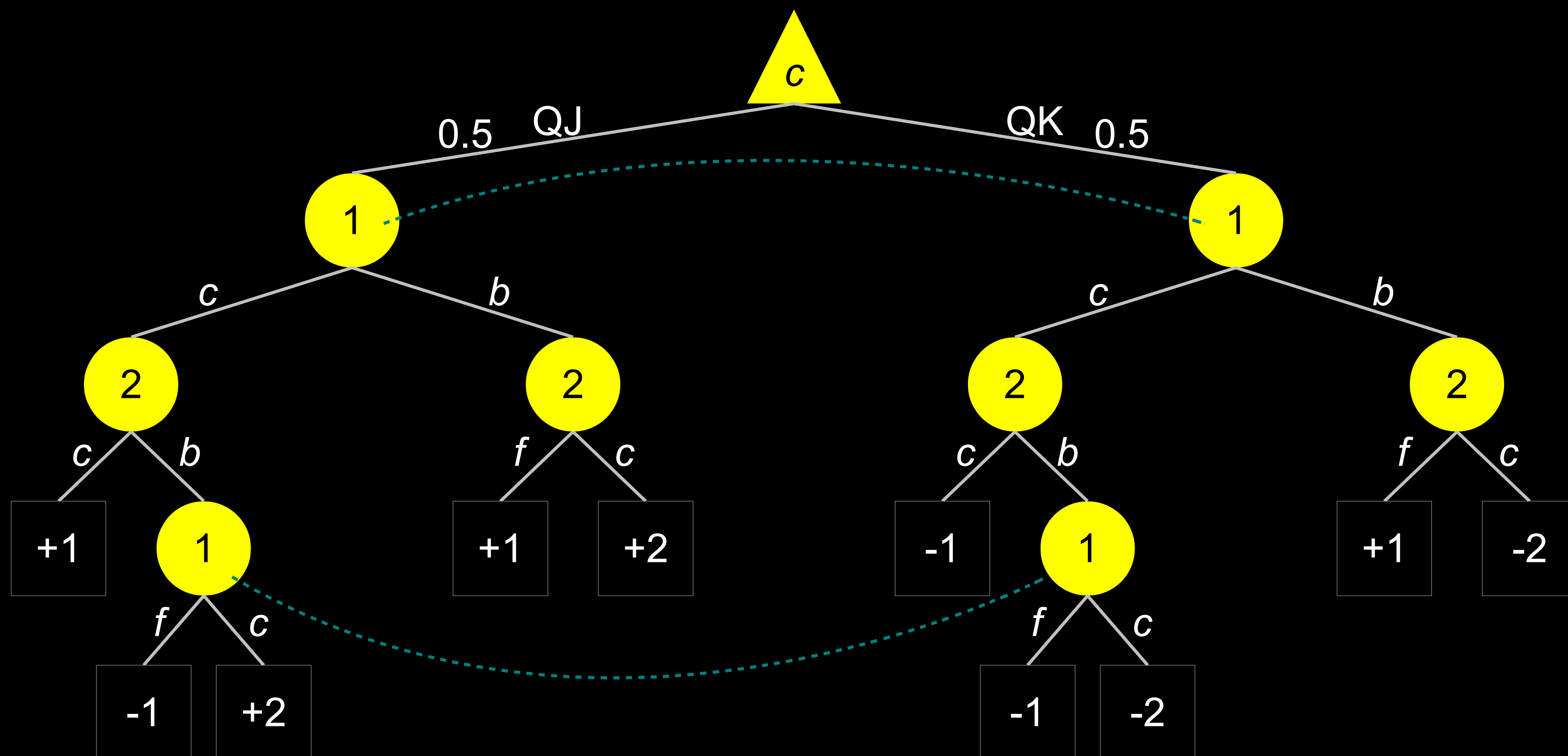
- Airport security
- Football
- Driving a car
- Etc. etc.

Extensive-Form Game Terminology



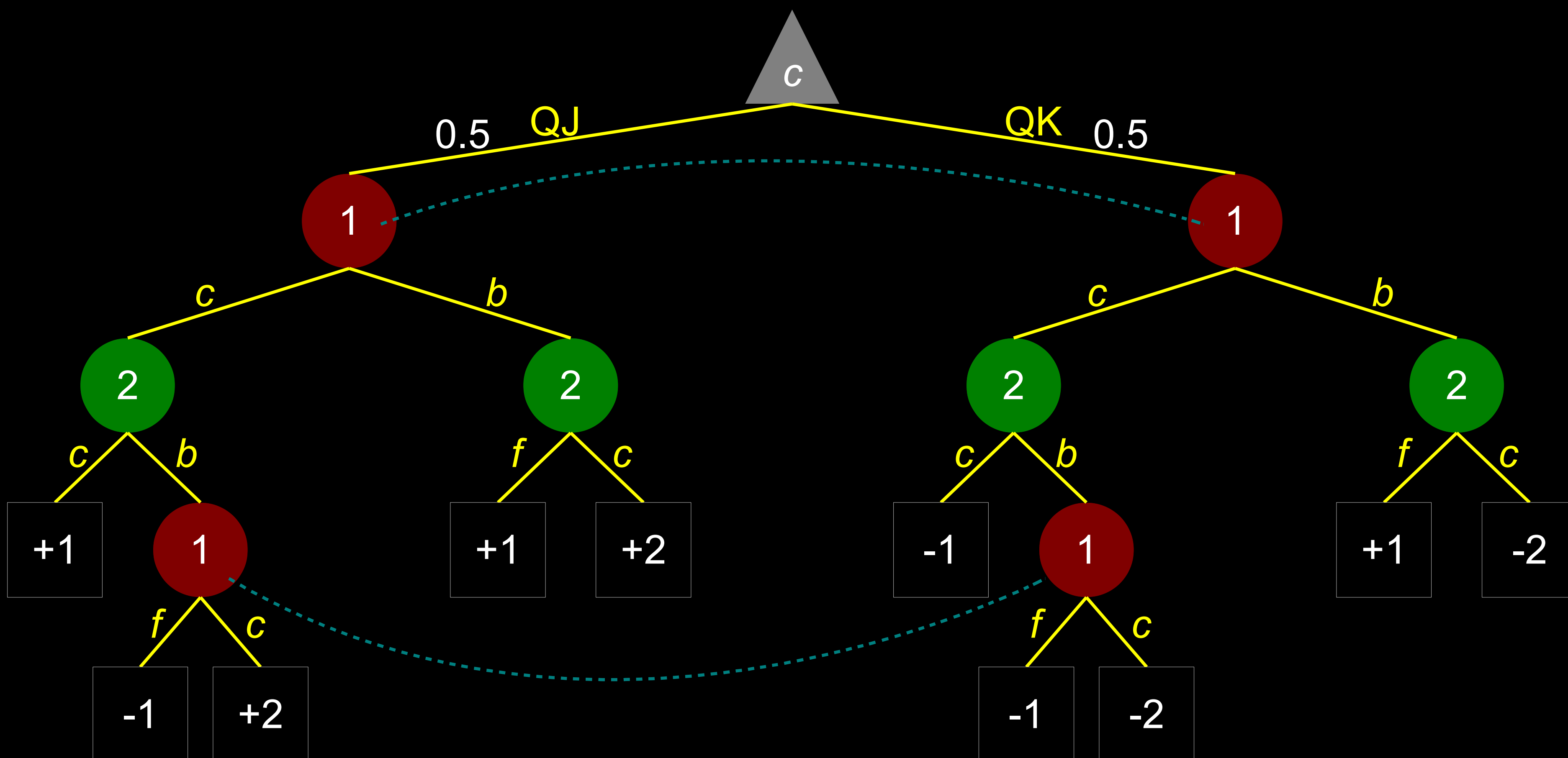
Extensive-Form Game Terminology

Nodes are histories (game states)



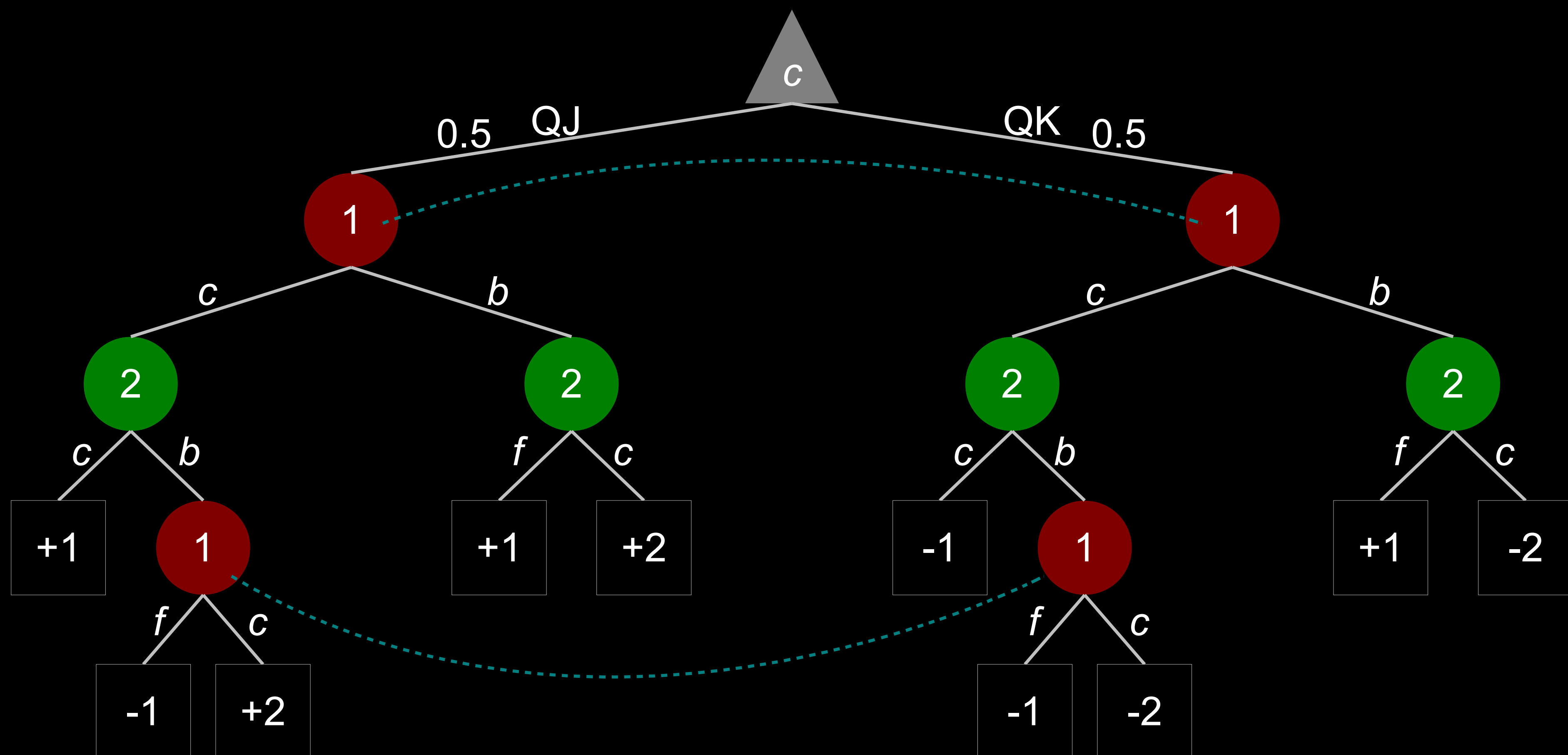
Extensive-Form Game Terminology

Edges are actions



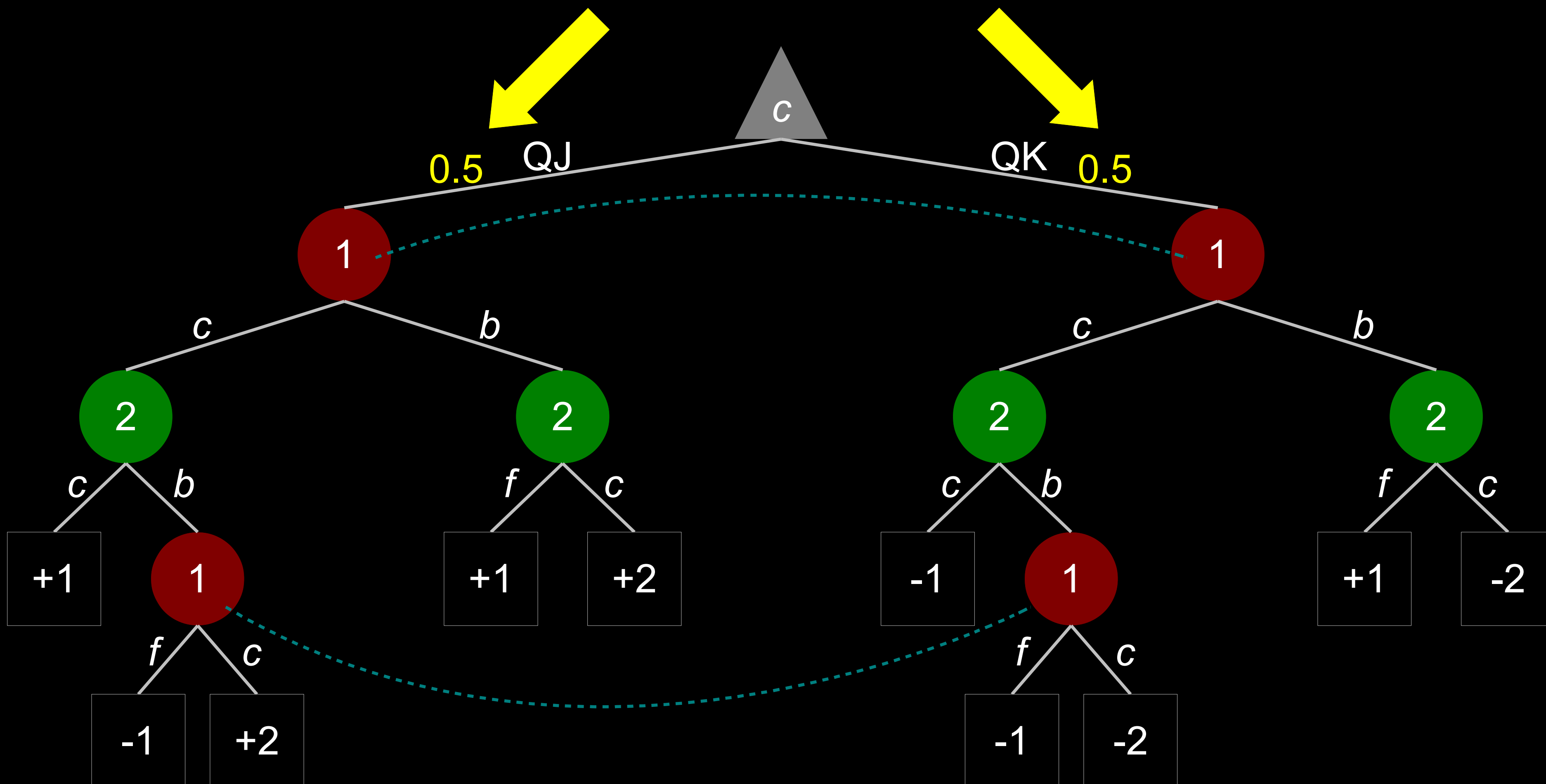
Extensive-Form Game Terminology

Histories belong to player 1, player 2, or chance (assume 2-players)



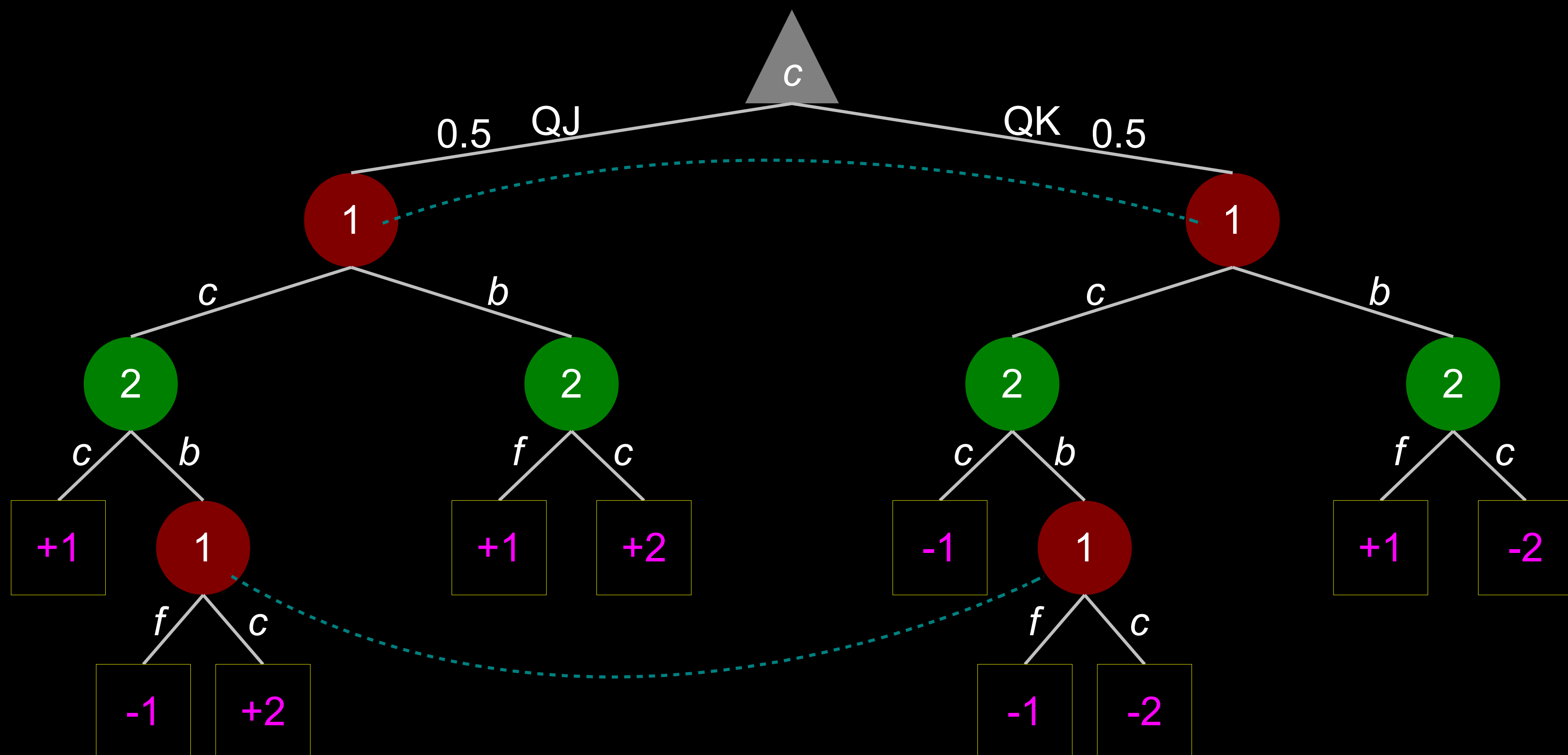
Extensive-Form Game Terminology

Chance's action probabilities are known and fixed



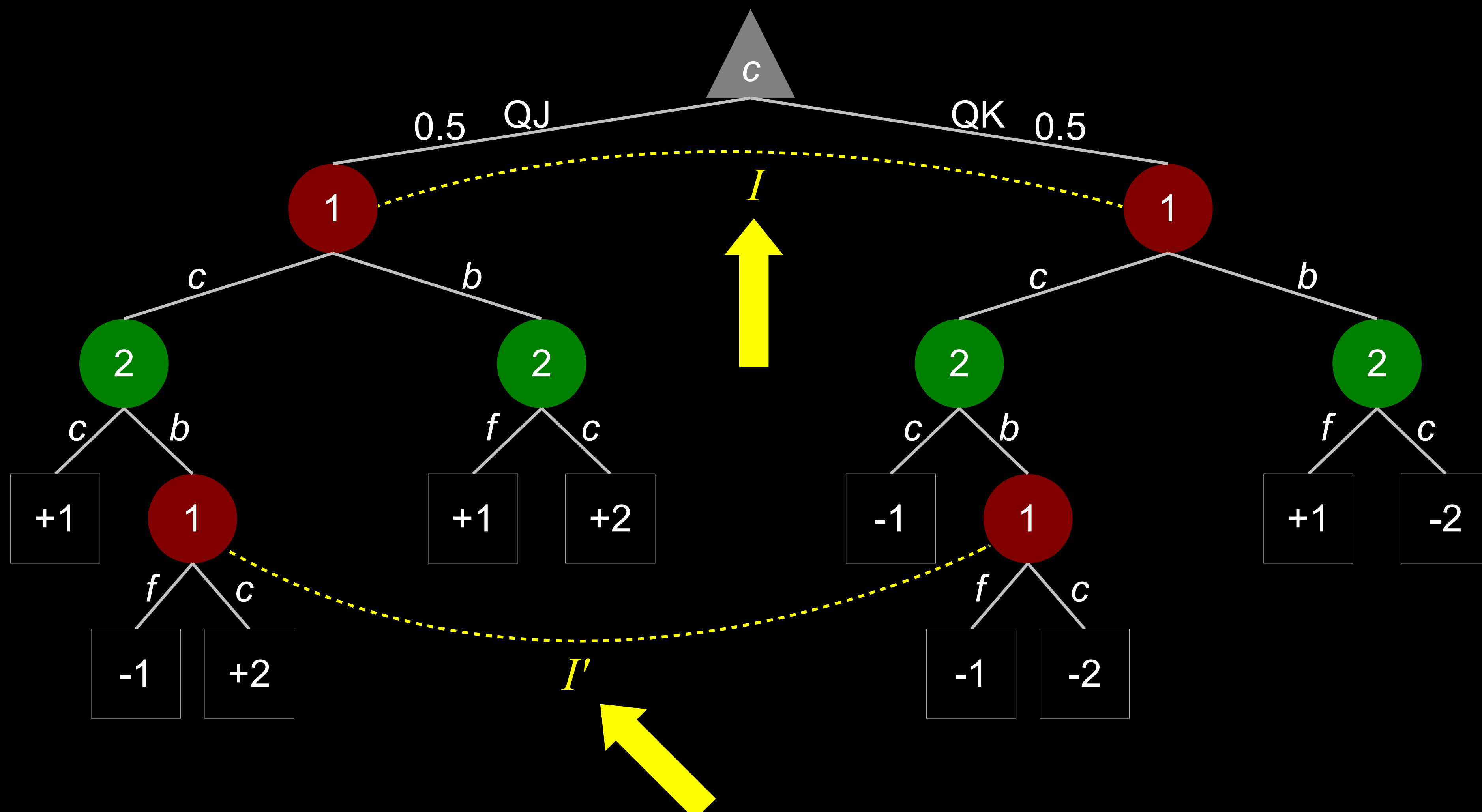
Extensive-Form Game Terminology

Terminal histories have associated utilities (assume **zero-sum**)



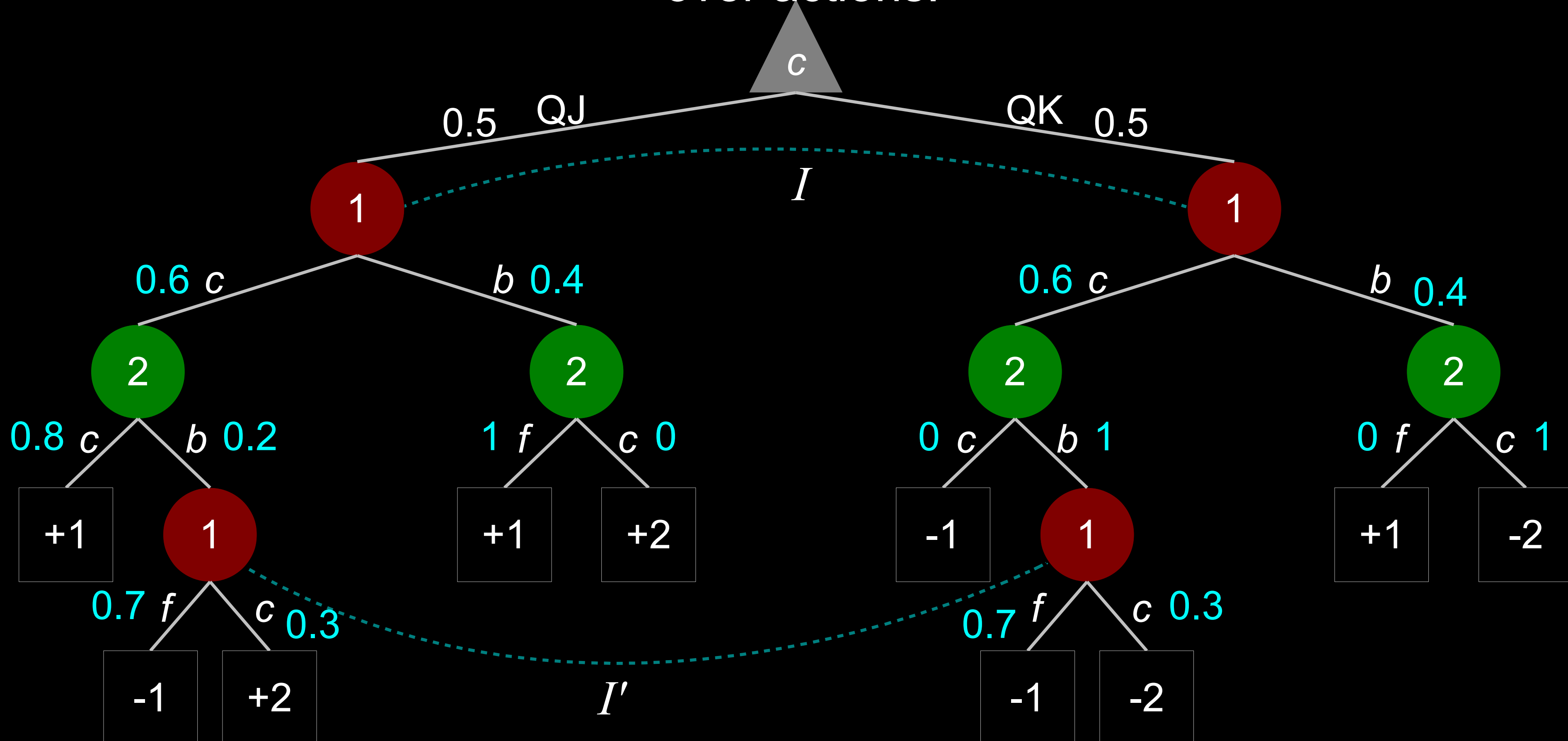
Extensive-Form Game Terminology

Players' histories are partitioned into information sets



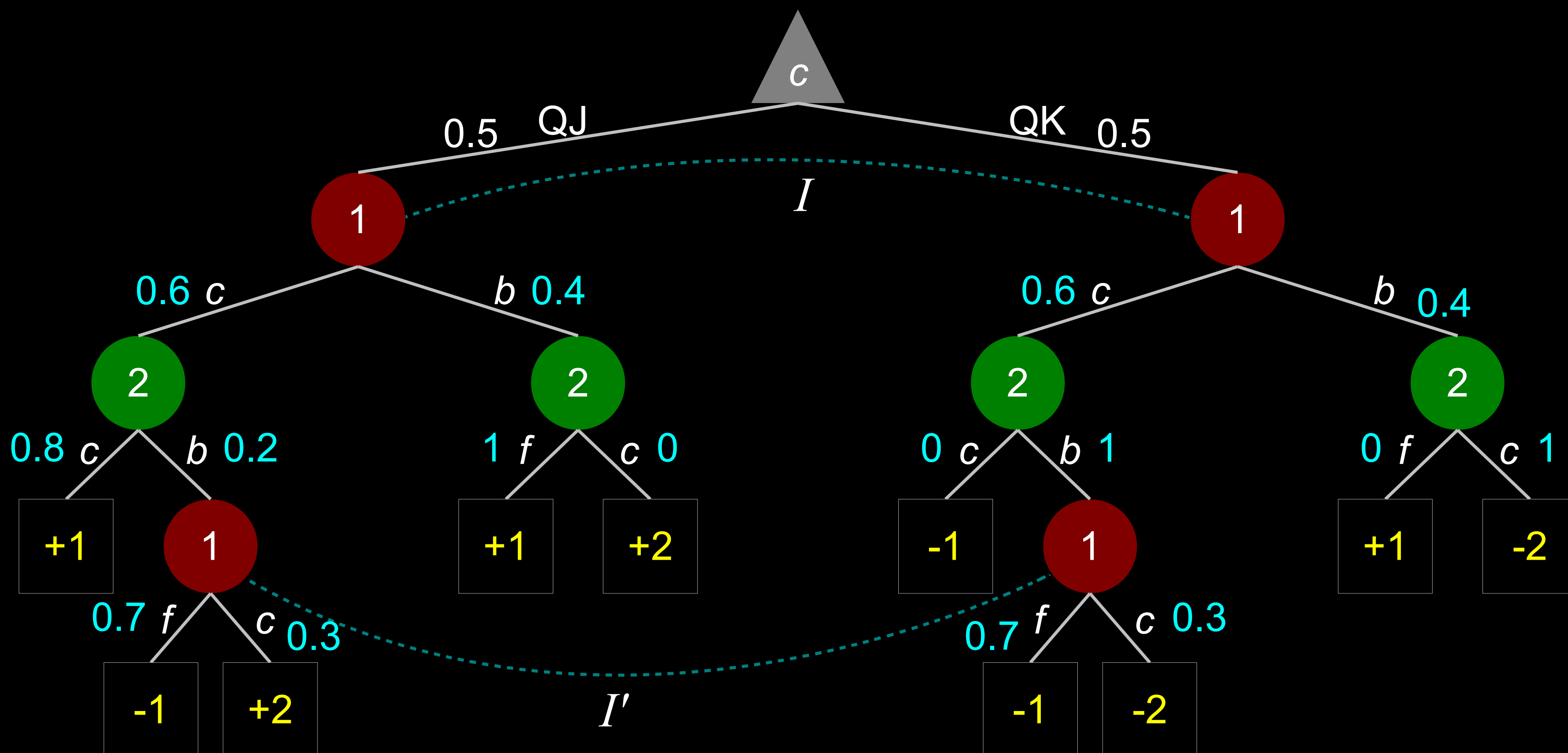
Extensive-Form Game Terminology

Strategy profile $\sigma = (\sigma_1, \sigma_2)$ maps information sets to probability distributions over actions.



Extensive-Form Game Terminology

$u_i(\ \sigma\ )$ is the expected utility for player i if both players play according to σ.



Solution Concepts

- A best response to a player 1 strategy σ_1 is an opponent strategy that maximizes player 2's expected utility:

$$\operatorname*{argmax}_{\sigma_2'} u_2(\sigma_1, \sigma_2')$$

- The best response value, $brv_2(\sigma_1)$, against σ_1 is that expected utility:

$$brv_2(\sigma_1) = \max_{\sigma_2'} u_2(\sigma_1, \sigma_2')$$

- The exploitability of σ, $e(\sigma)$ measures how much σ loses to a worst case opponent when players alternate positions:

$$e(\sigma) = \frac{(brv_1(\sigma_2) + brv_2(\sigma_1))}{2}$$

Solution Concepts

- A Nash equilibrium is a strategy profile σ with zero exploitability:

$$e(\sigma) = 0 \Rightarrow \sigma \text{ is Nash}$$

- An ε-Nash equilibrium is a strategy profile σ that is exploitable for at most ε:

$$e(\sigma) \leq \varepsilon \Rightarrow \sigma \text{ is } \varepsilon\text{-Nash}$$

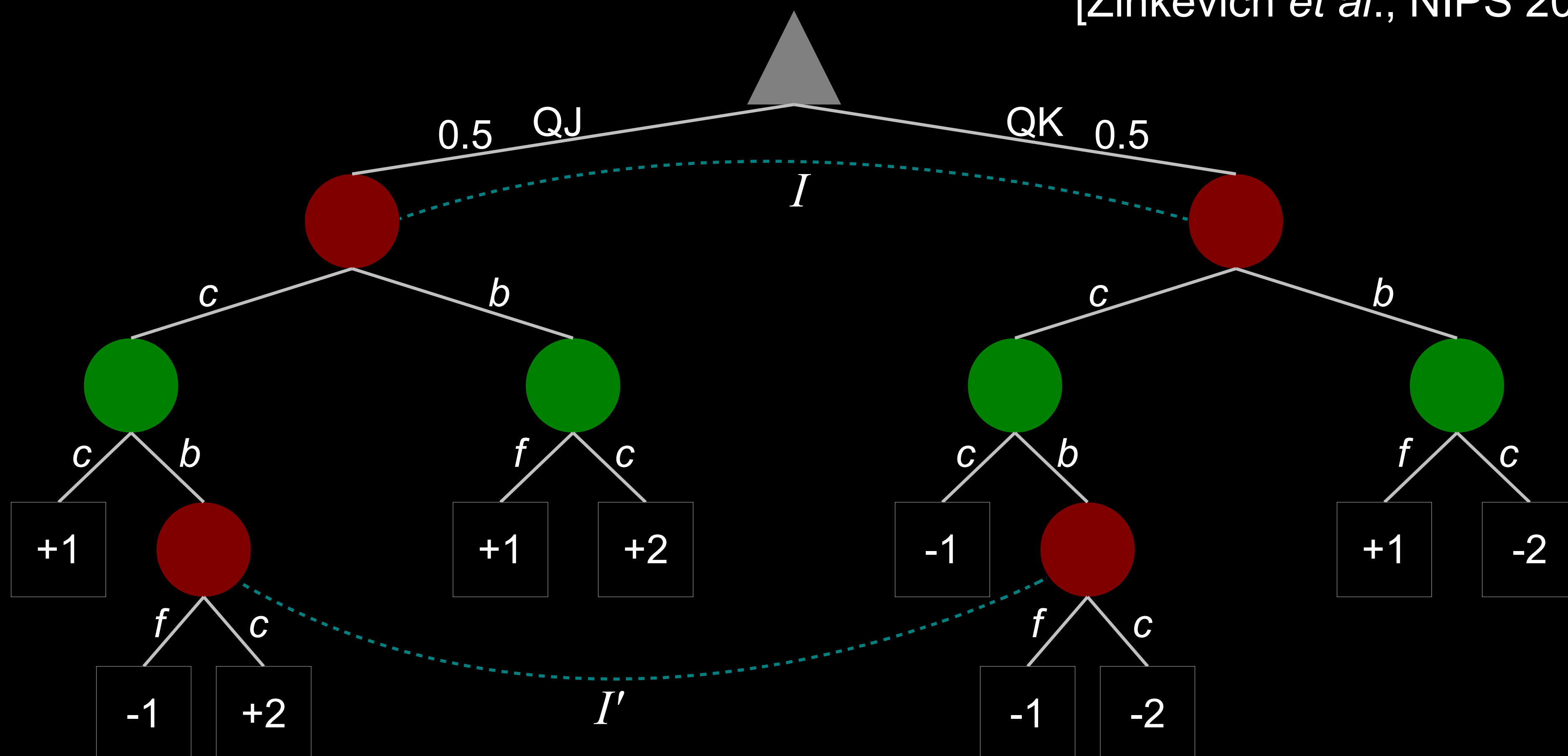
Outline

- Extensive-form Games
  - Examples
  - Terminology
  - Solution concepts
- Counterfactual Regret Minimization (CFR)
  - Base algorithm for solving extensive-form games
  - Older variants
- New, Faster CFR Variants
  - Probing
  - Public Chance Sampling
  - Average Strategy Sampling
- Conclusions and Future Work

“Vanilla” CFR Walk-through

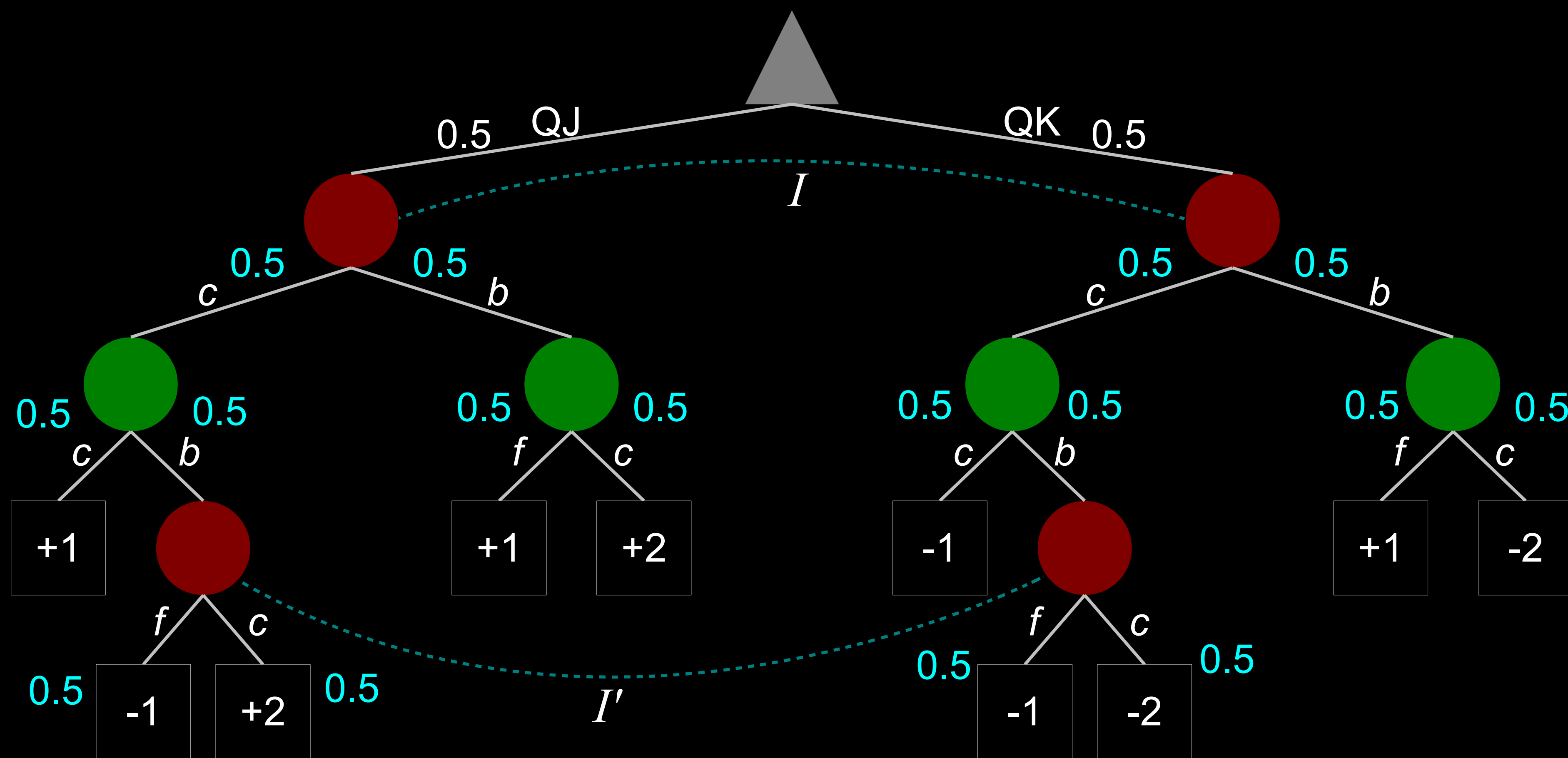
CFR is an iterative algorithm that computes an ε-Nash equilibrium

[Zinkevich *et al.*, NIPS 2007]



"Vanilla" CFR Walk-through

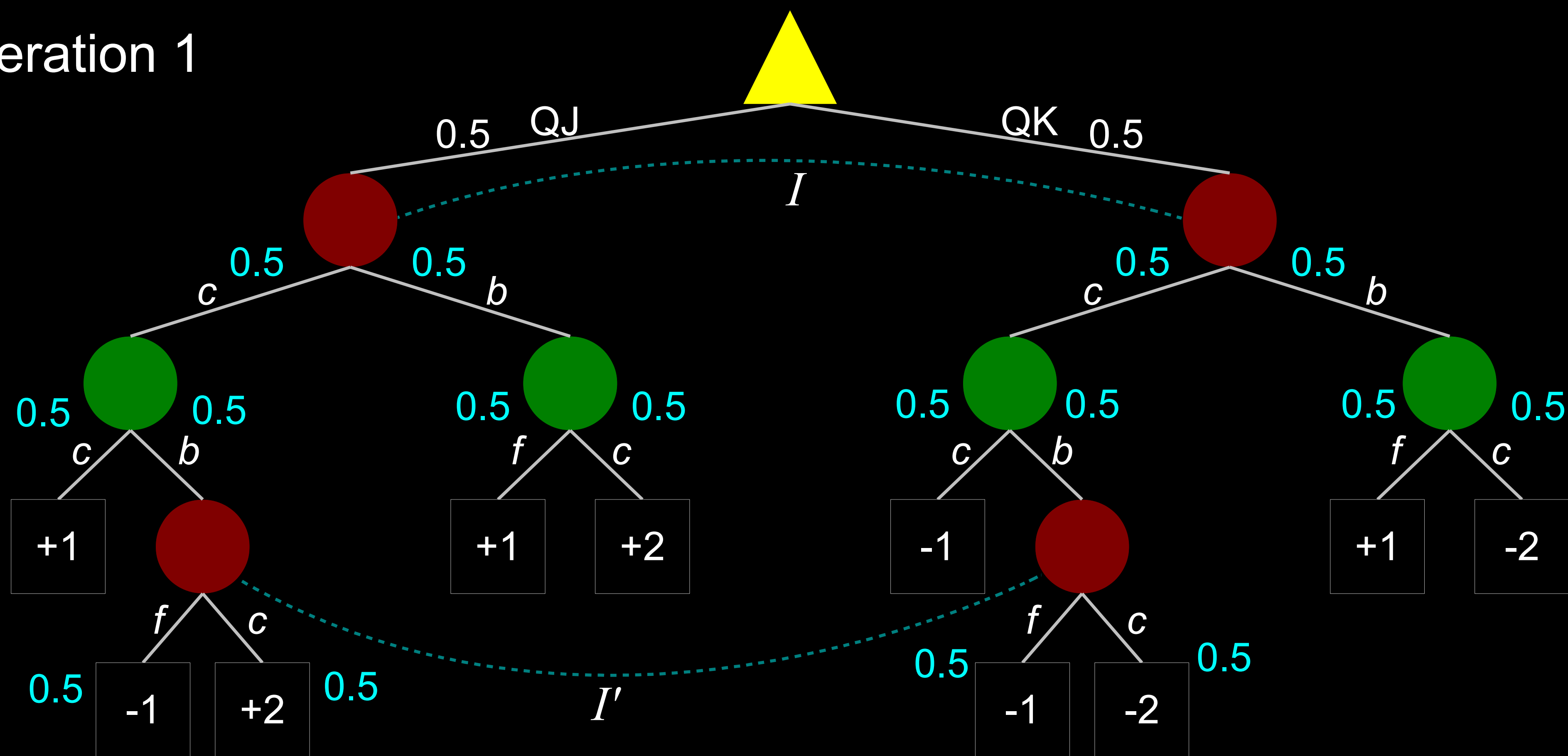
Initialize strategy profile to uniform random



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

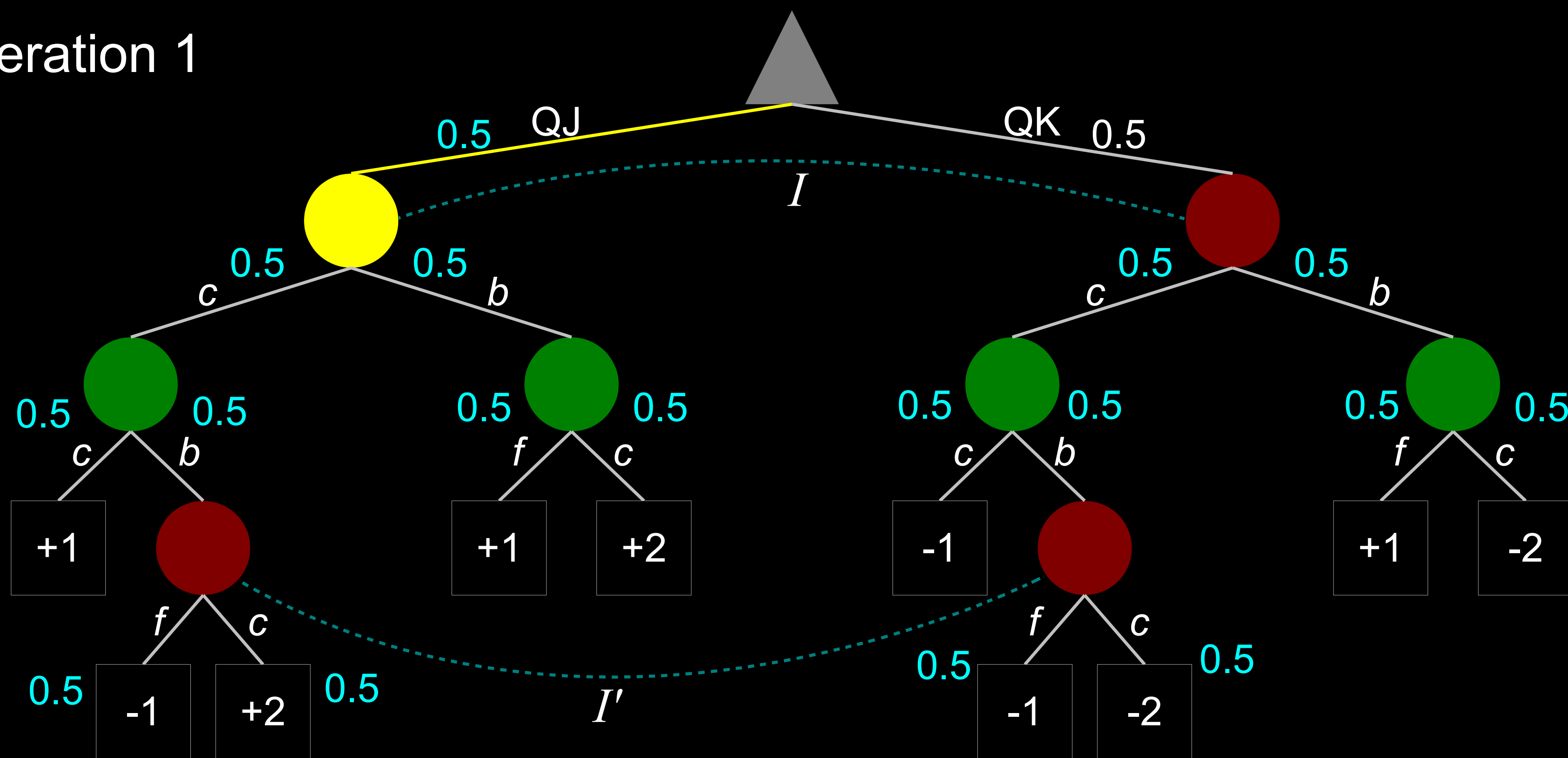
- Iteration 1



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

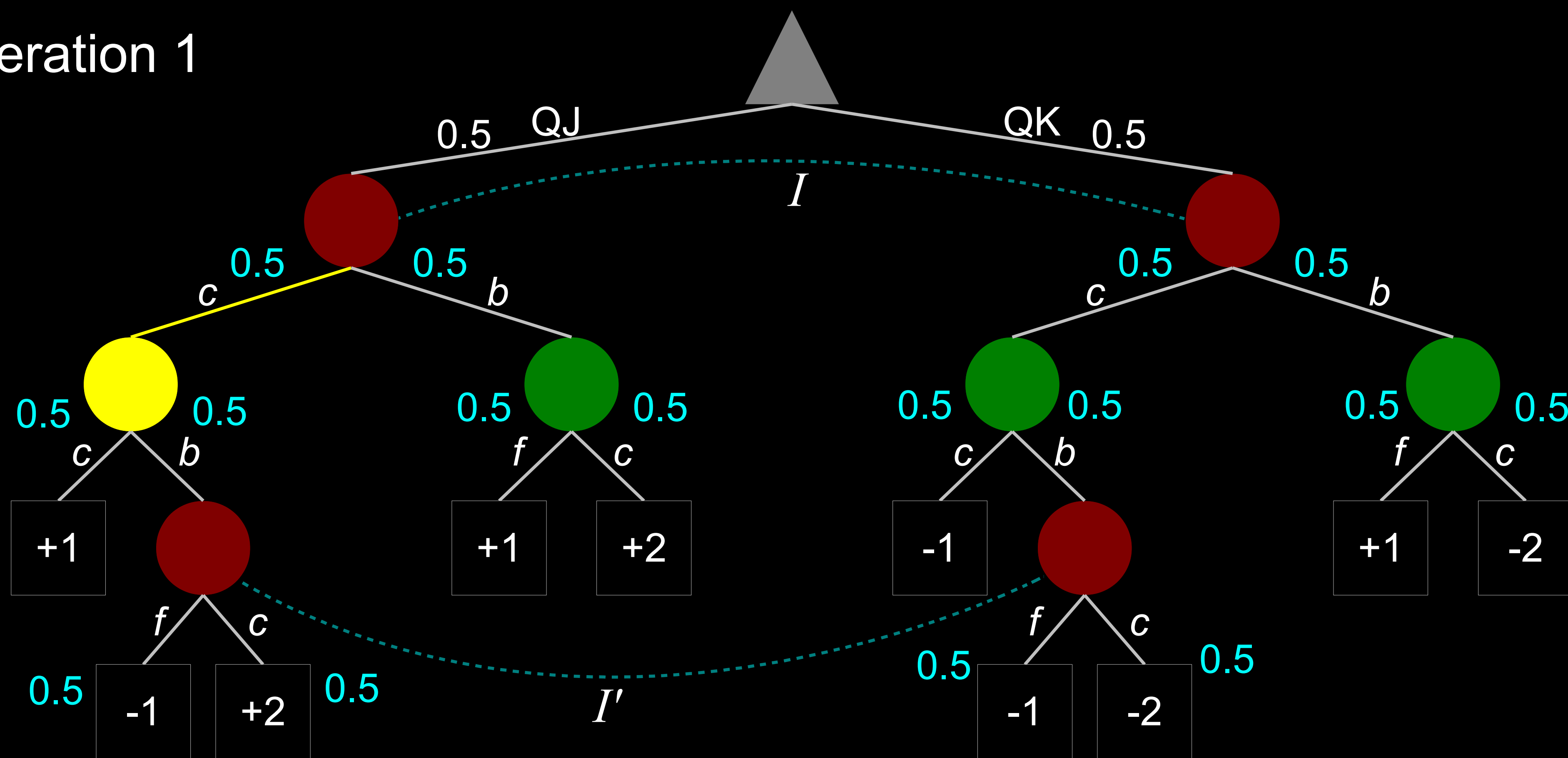
- Iteration 1



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

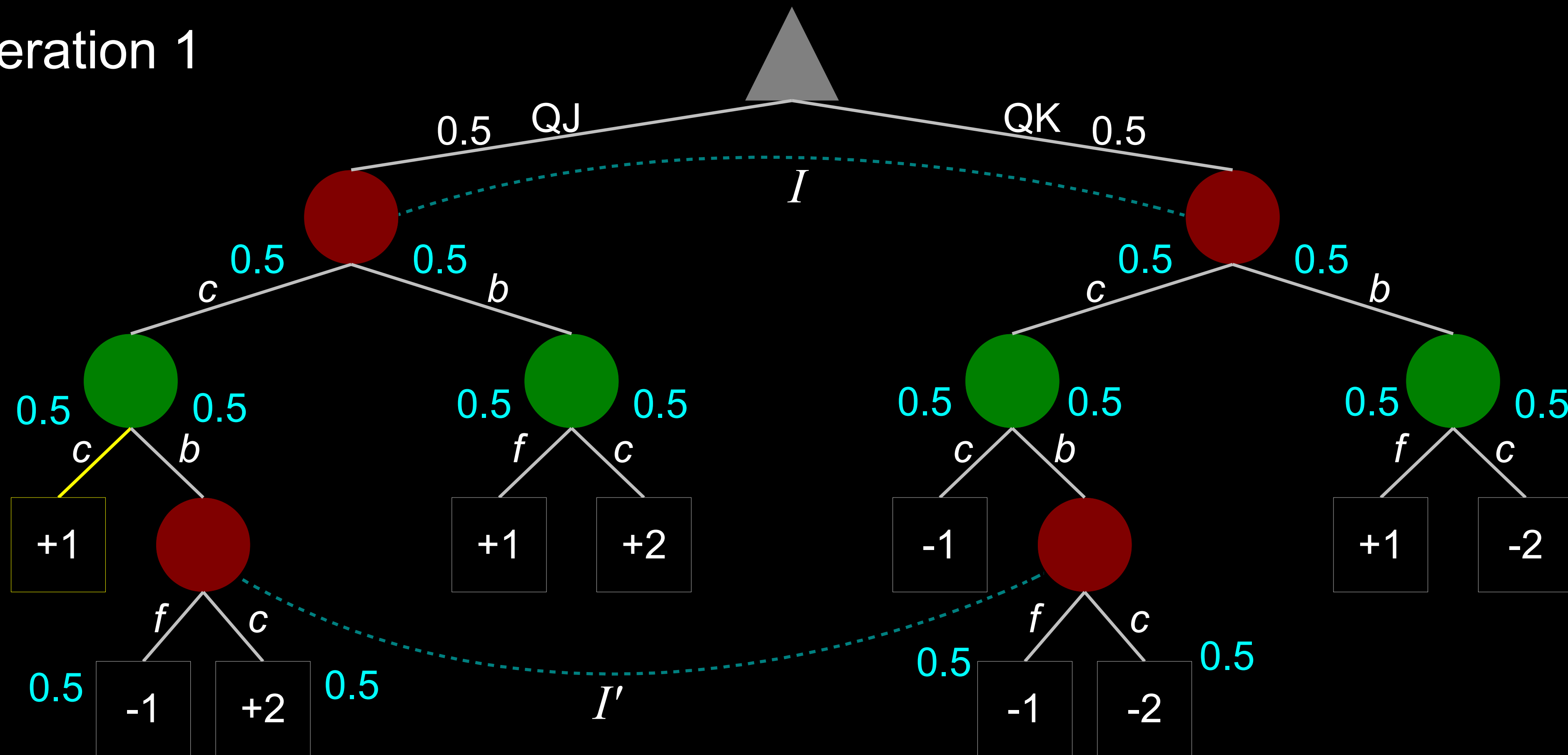
- Iteration 1



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

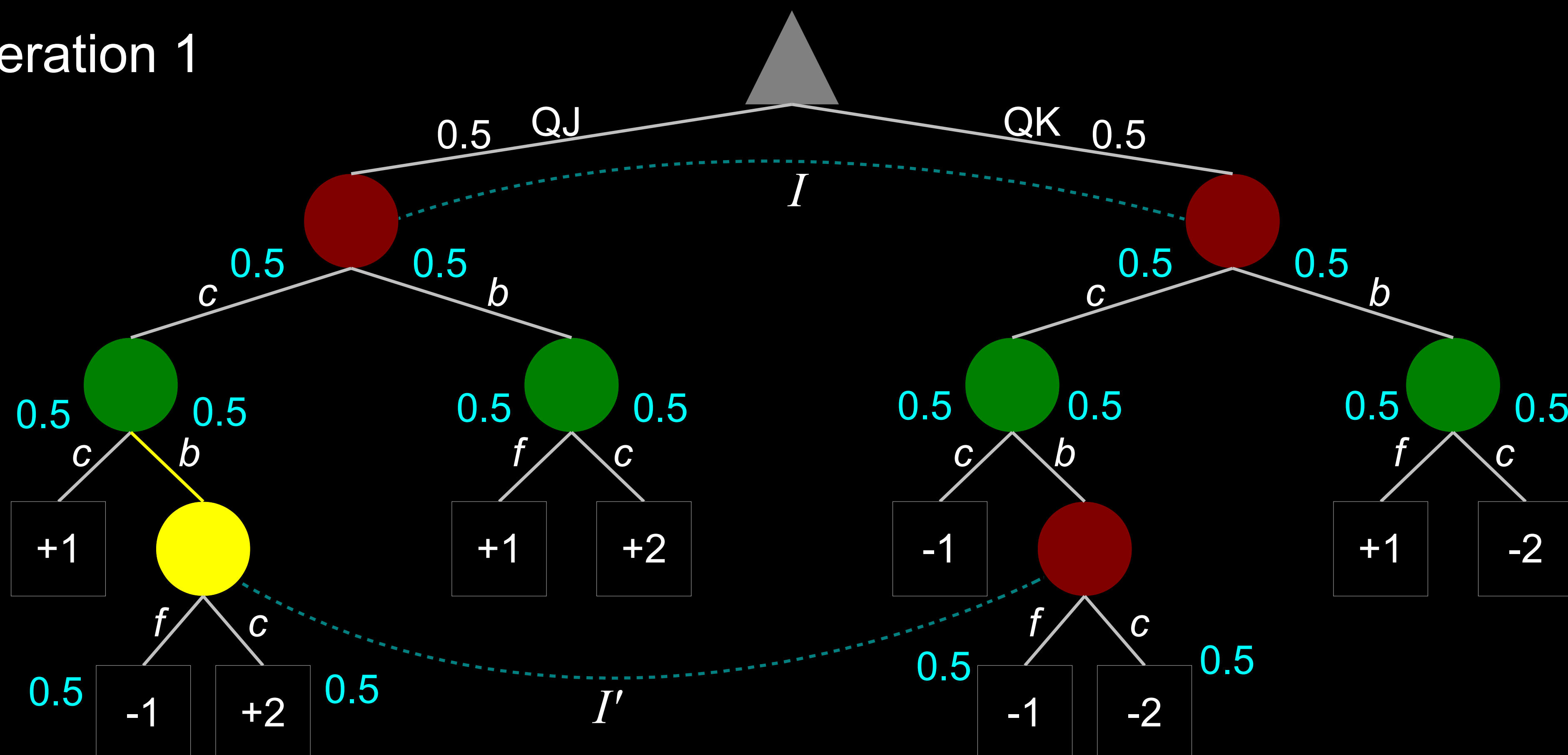
- Iteration 1



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

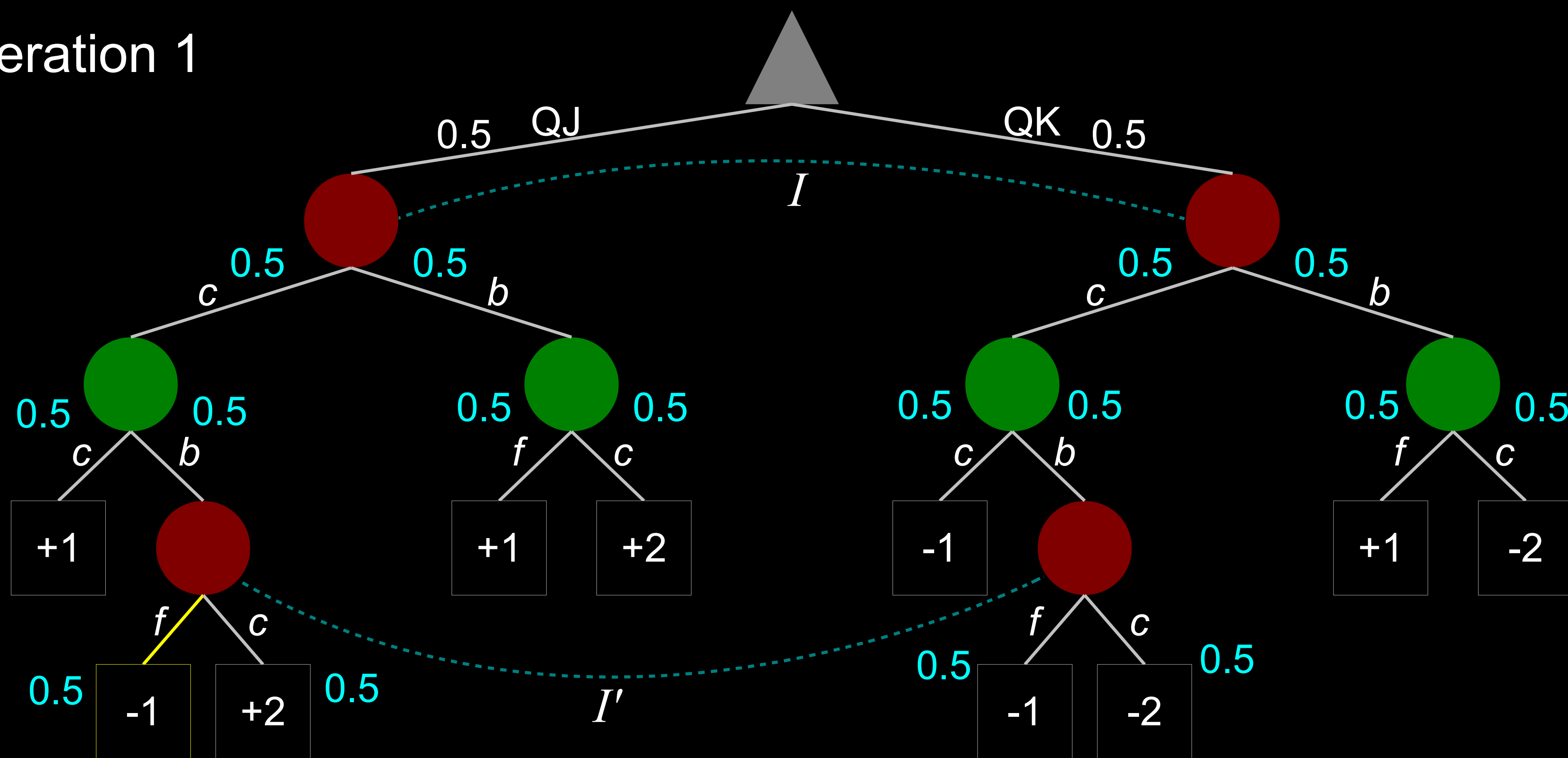
- Iteration 1



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

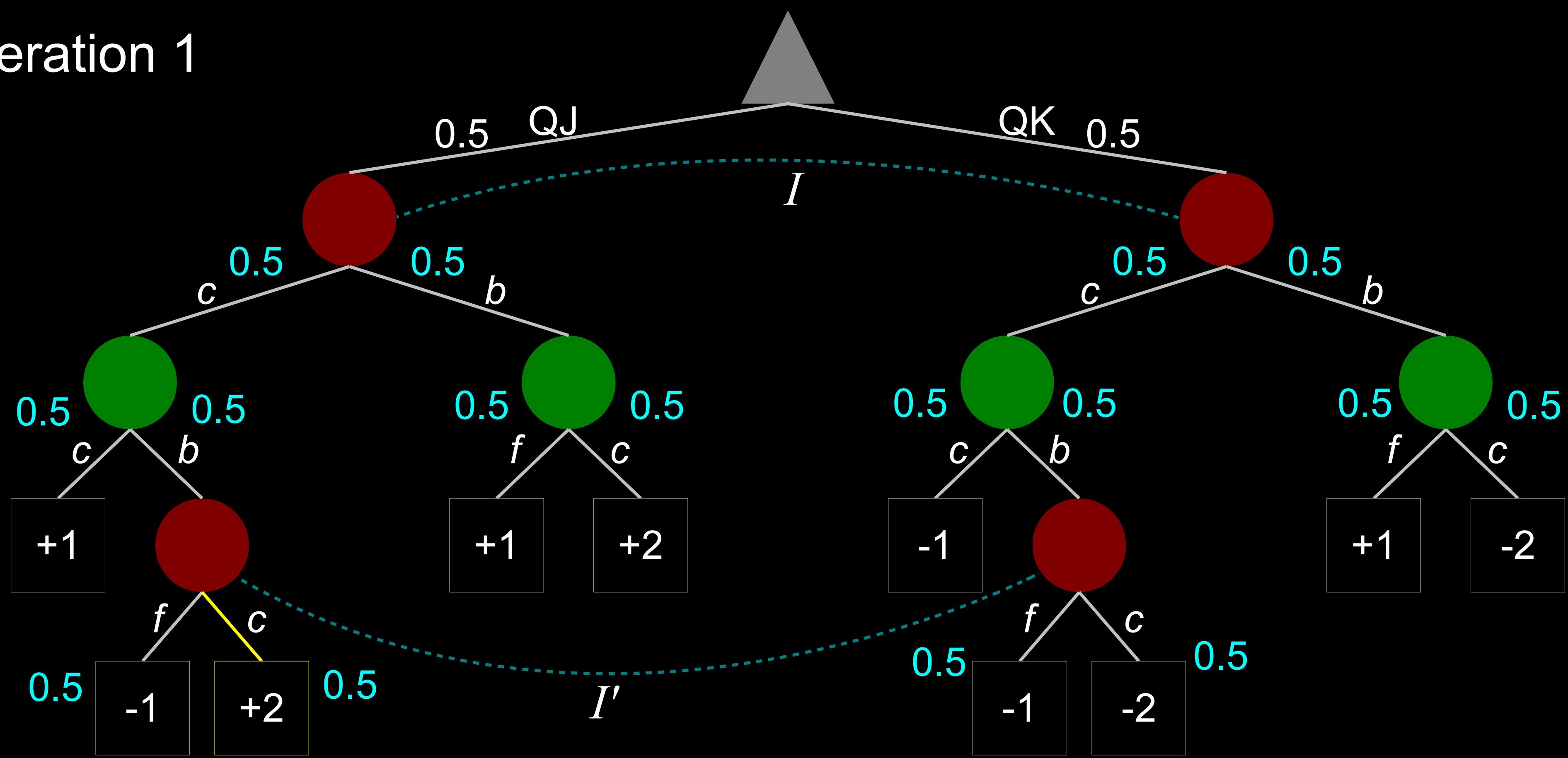
- Iteration 1



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

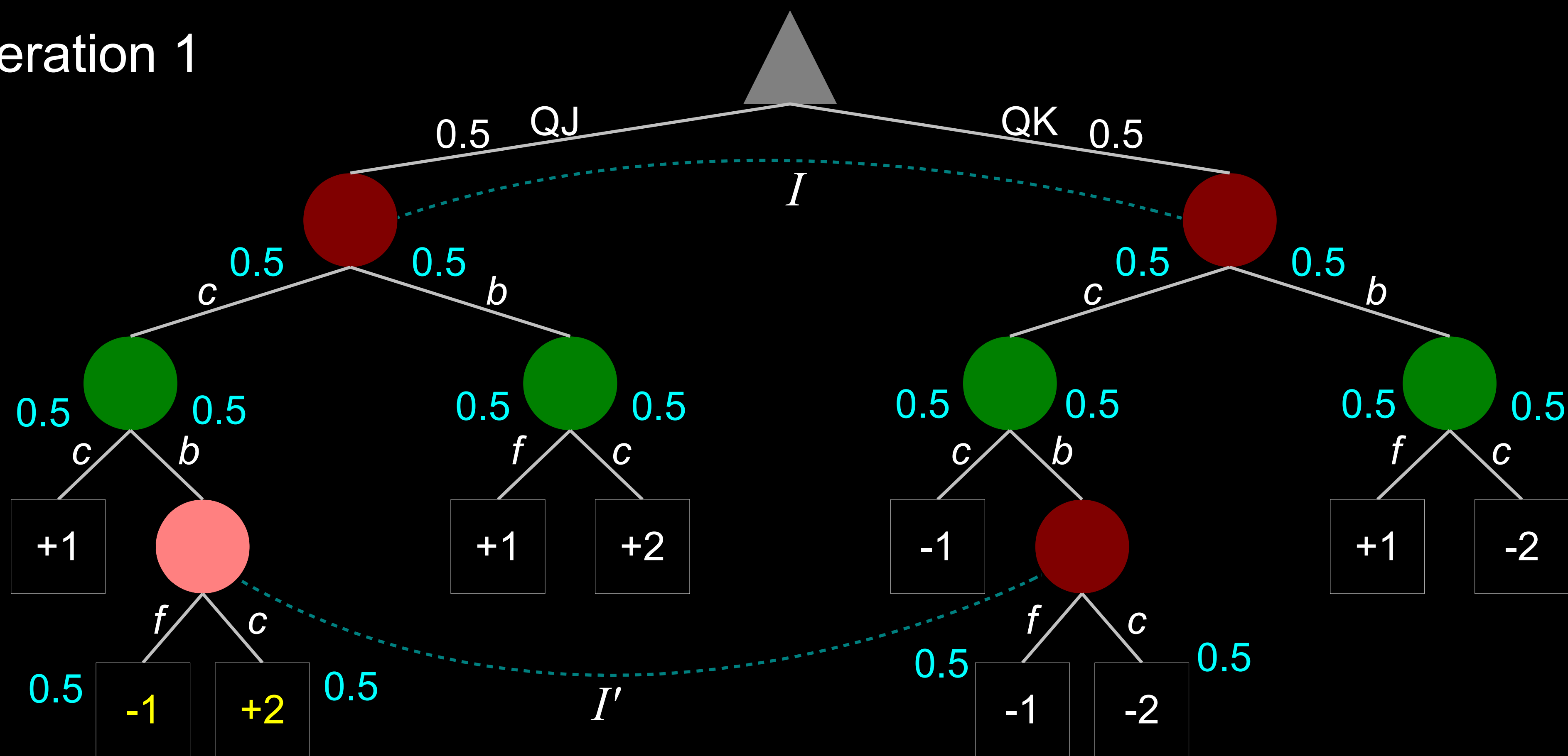
- Iteration 1



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 1

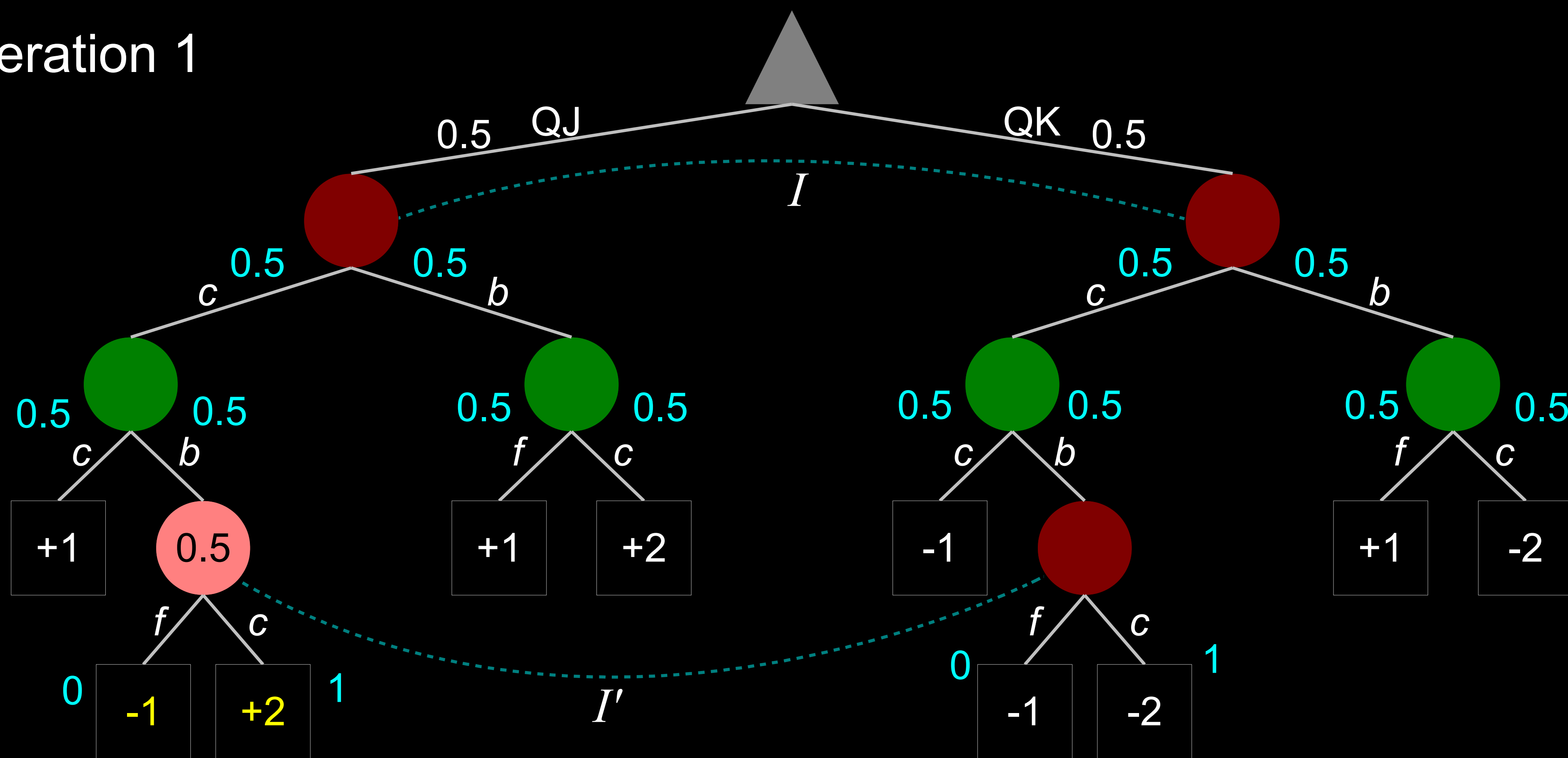


EV = 0.5(-1) + 0.5(+2) = 0.5

“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 1



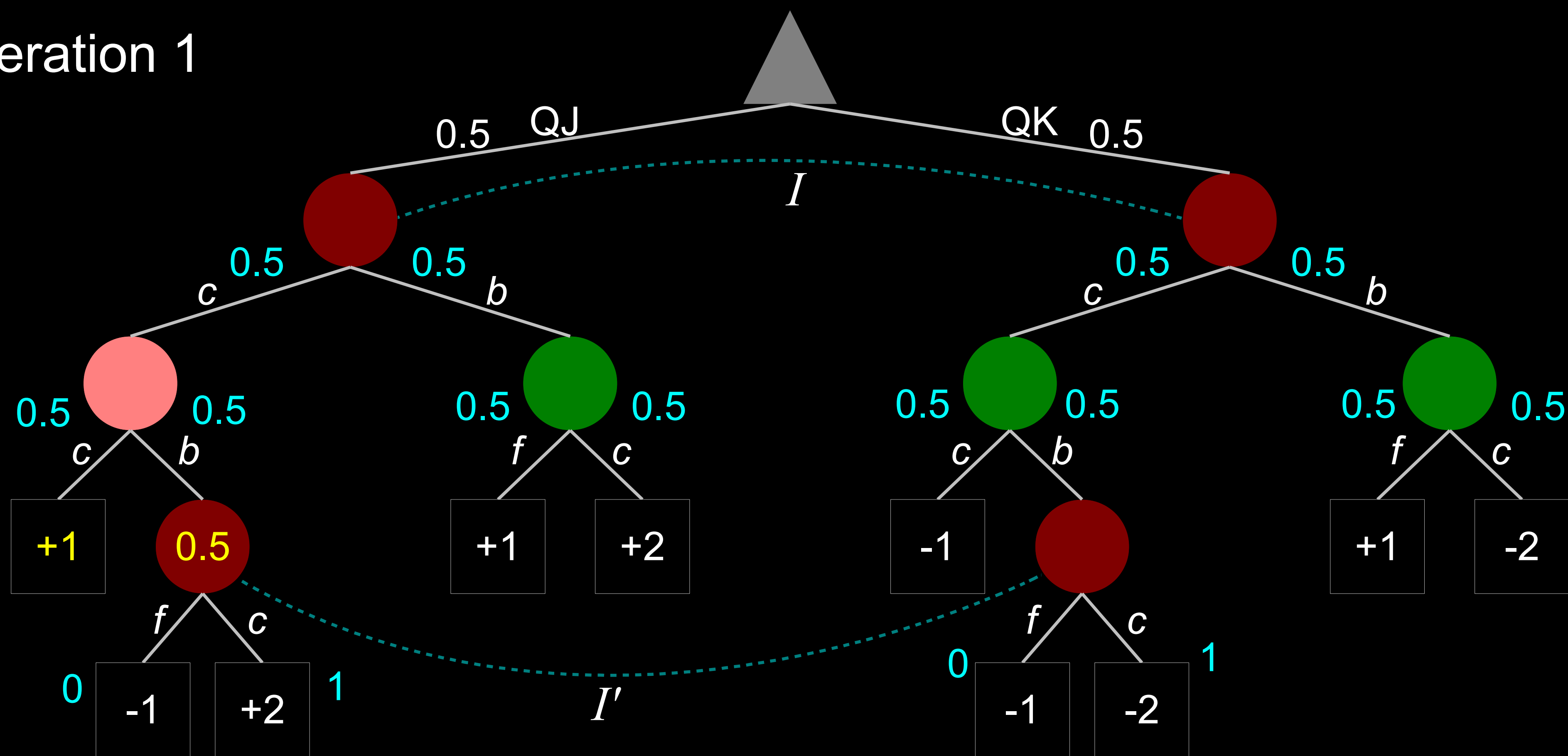
$\sigma_1(f)$: -1 < 0.5

$\sigma_1(c)$: +2 > 0.5

“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 1

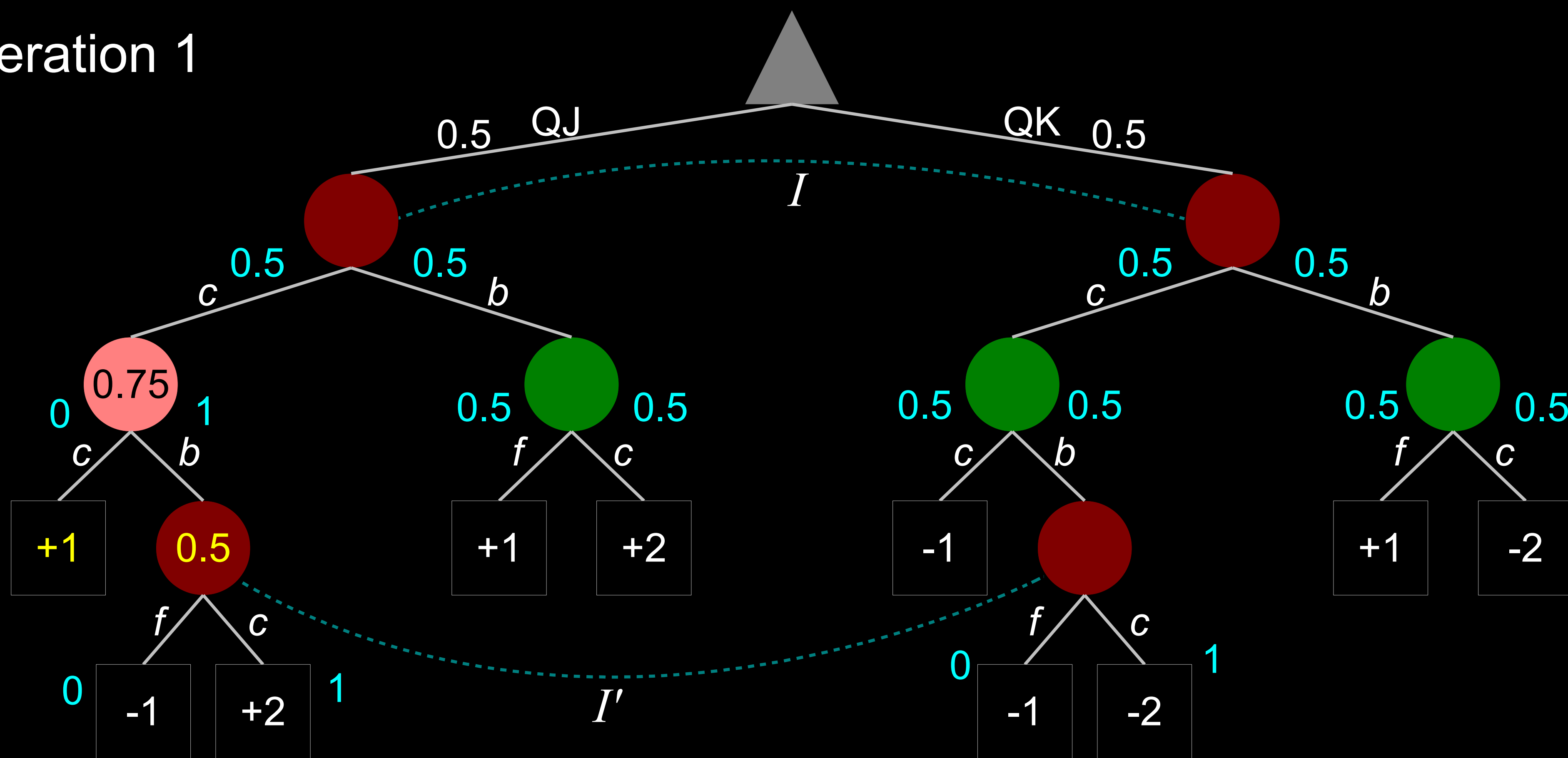


$\text{EV} = 0.5(+1) + 0.5(0.5) = 0.75$

“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 1



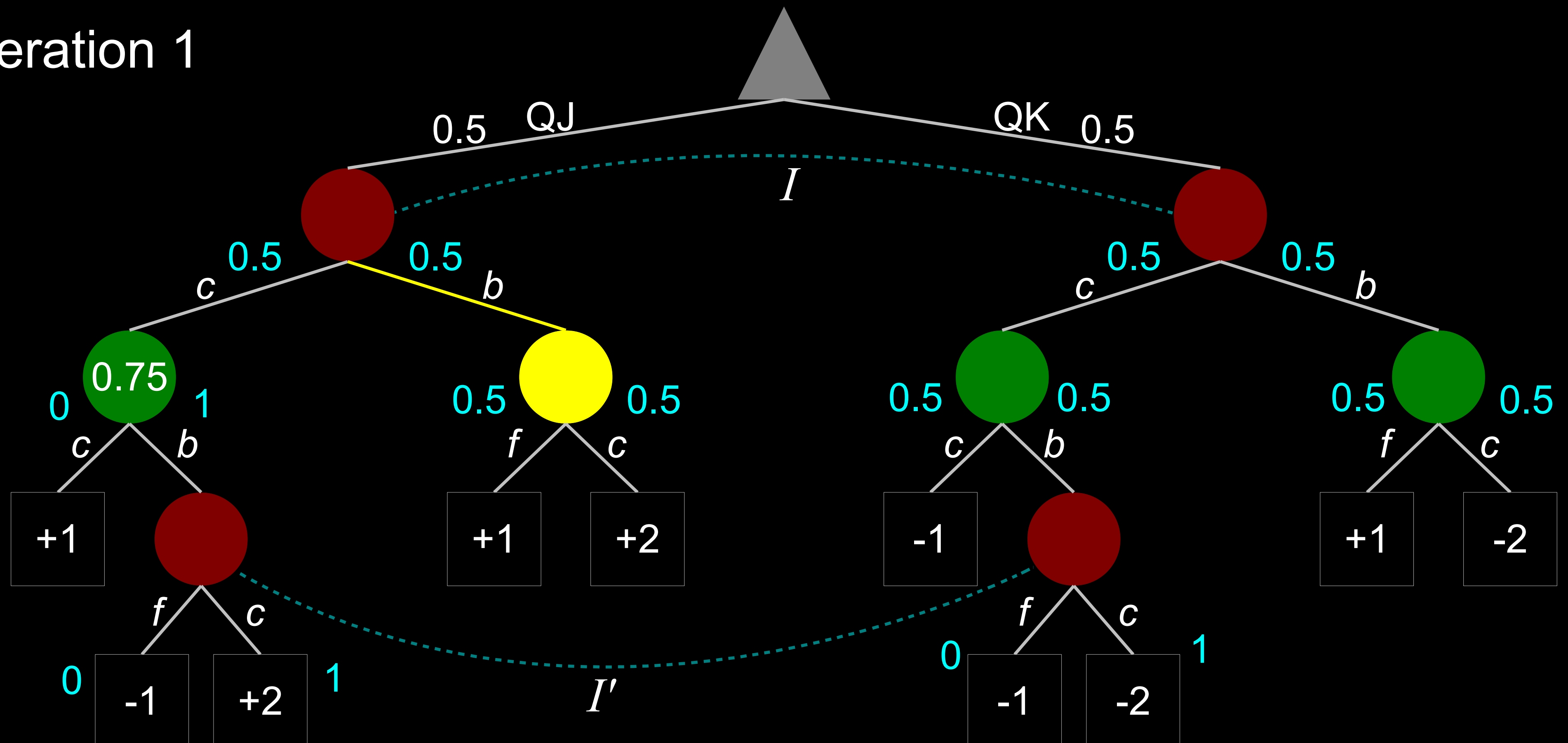
$\sigma_2(\ c\ )$: -1 < -0.75

$\sigma_2(\ b\ )$: -0.5 > -0.75

“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

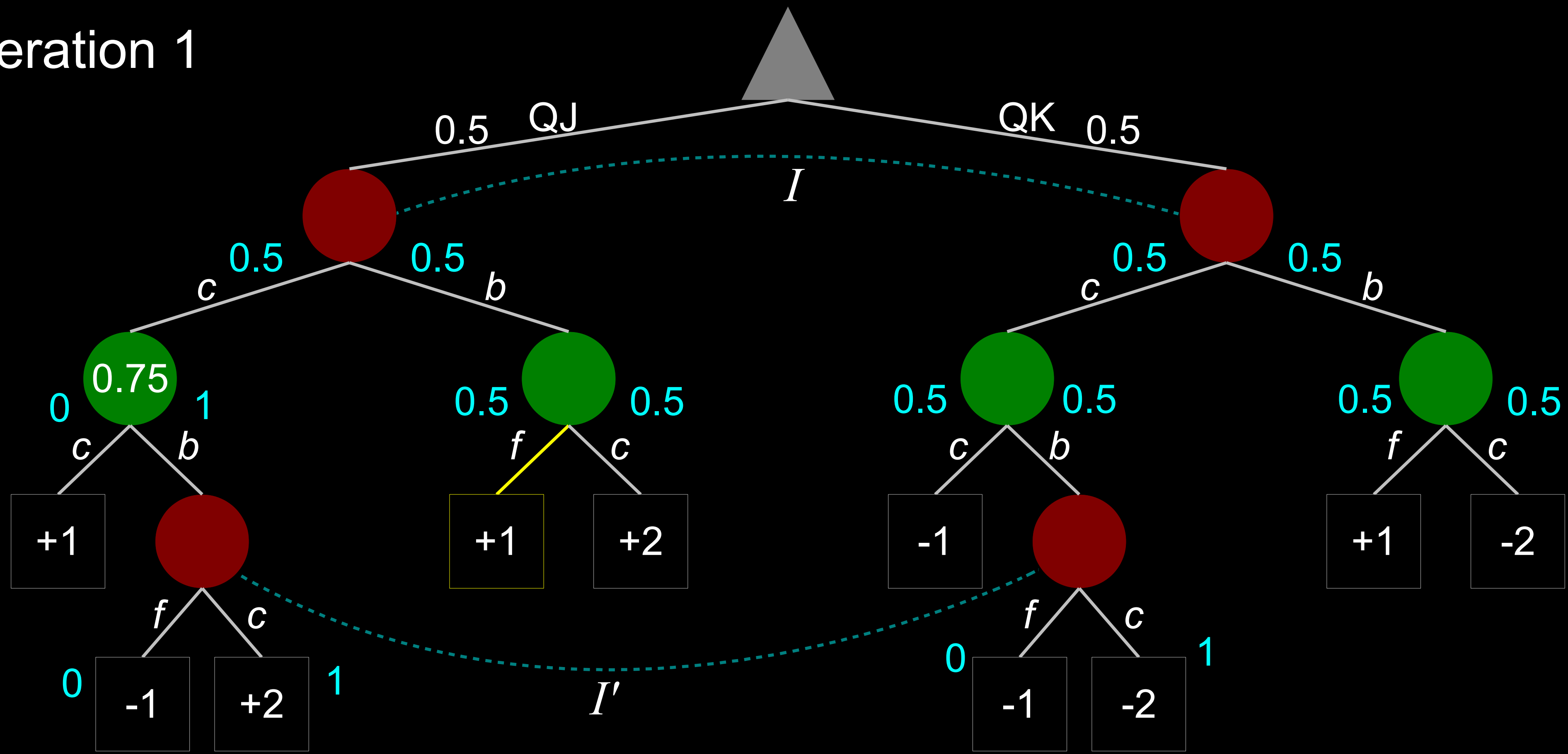
- Iteration 1



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

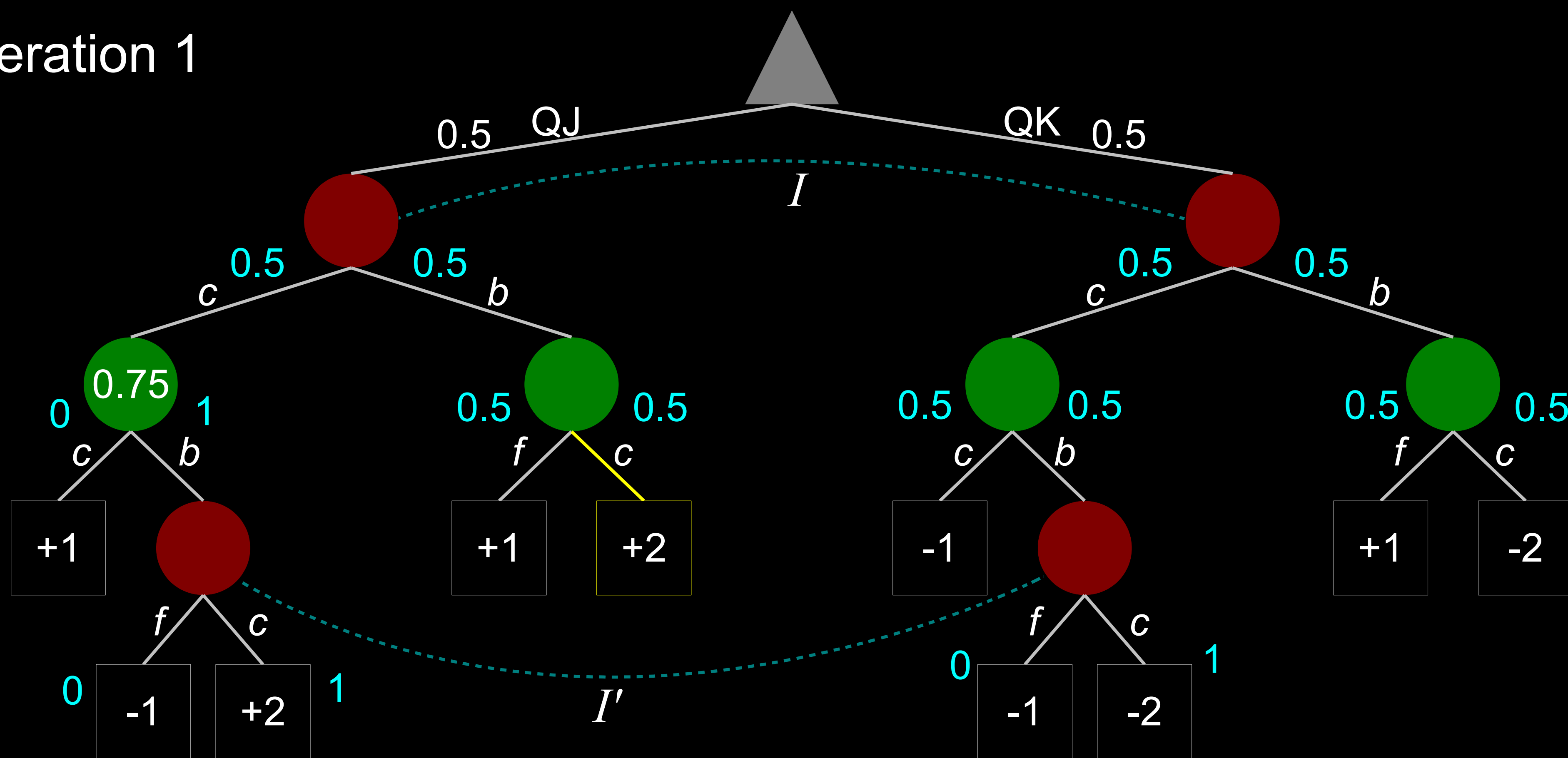
- Iteration 1



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

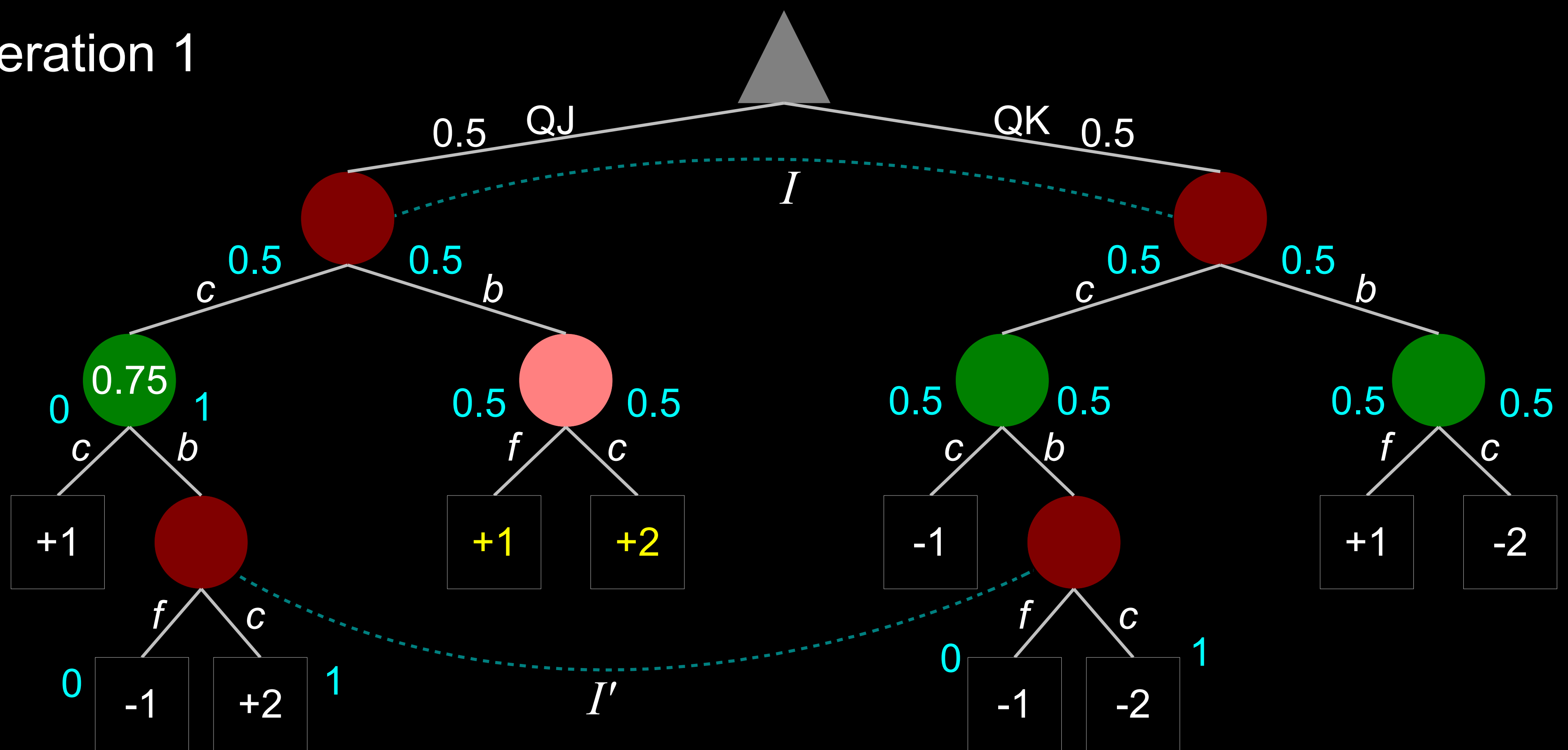
- Iteration 1



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 1

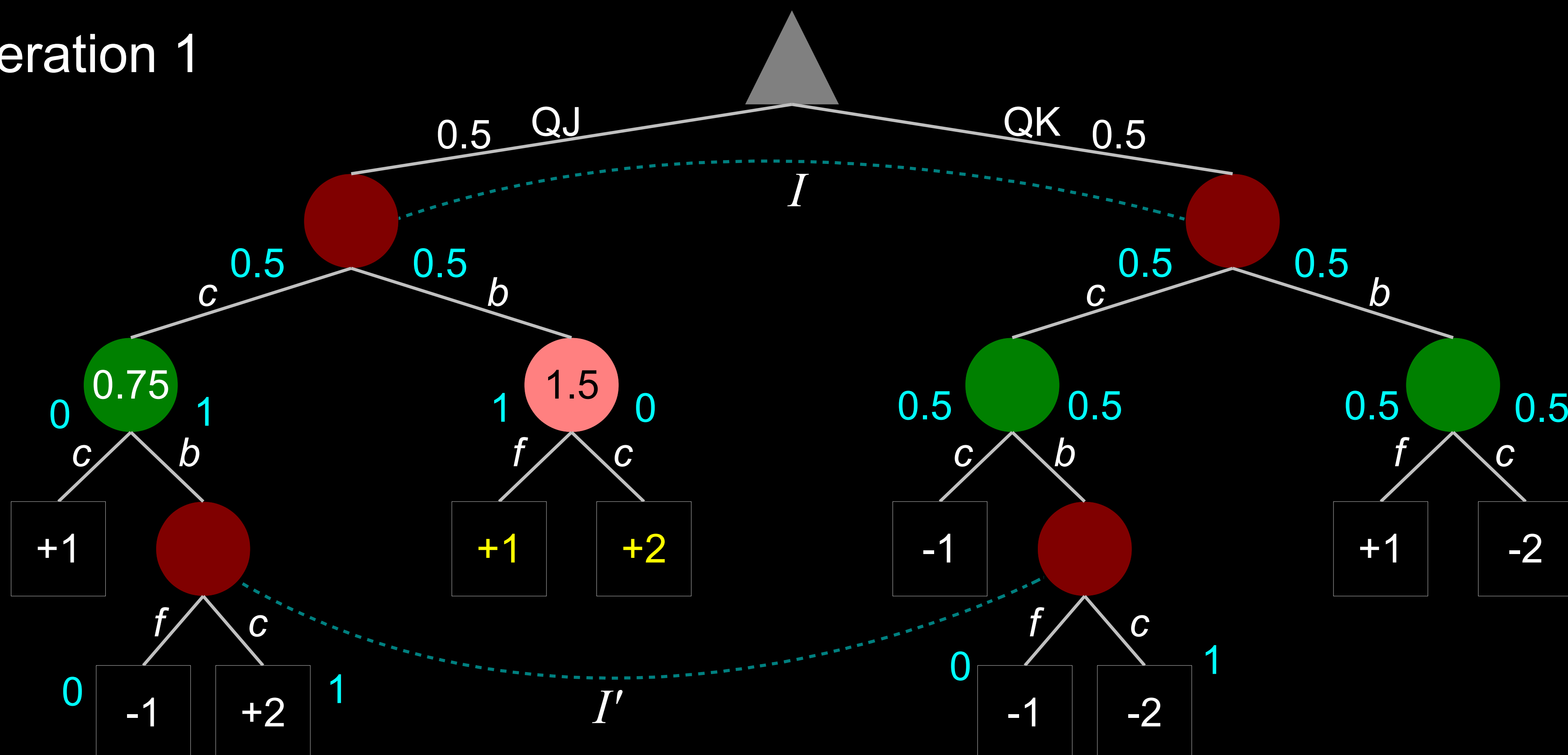


$$EV = 0.5(+1) + 0.5(+2) = 1.5$$

“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 1

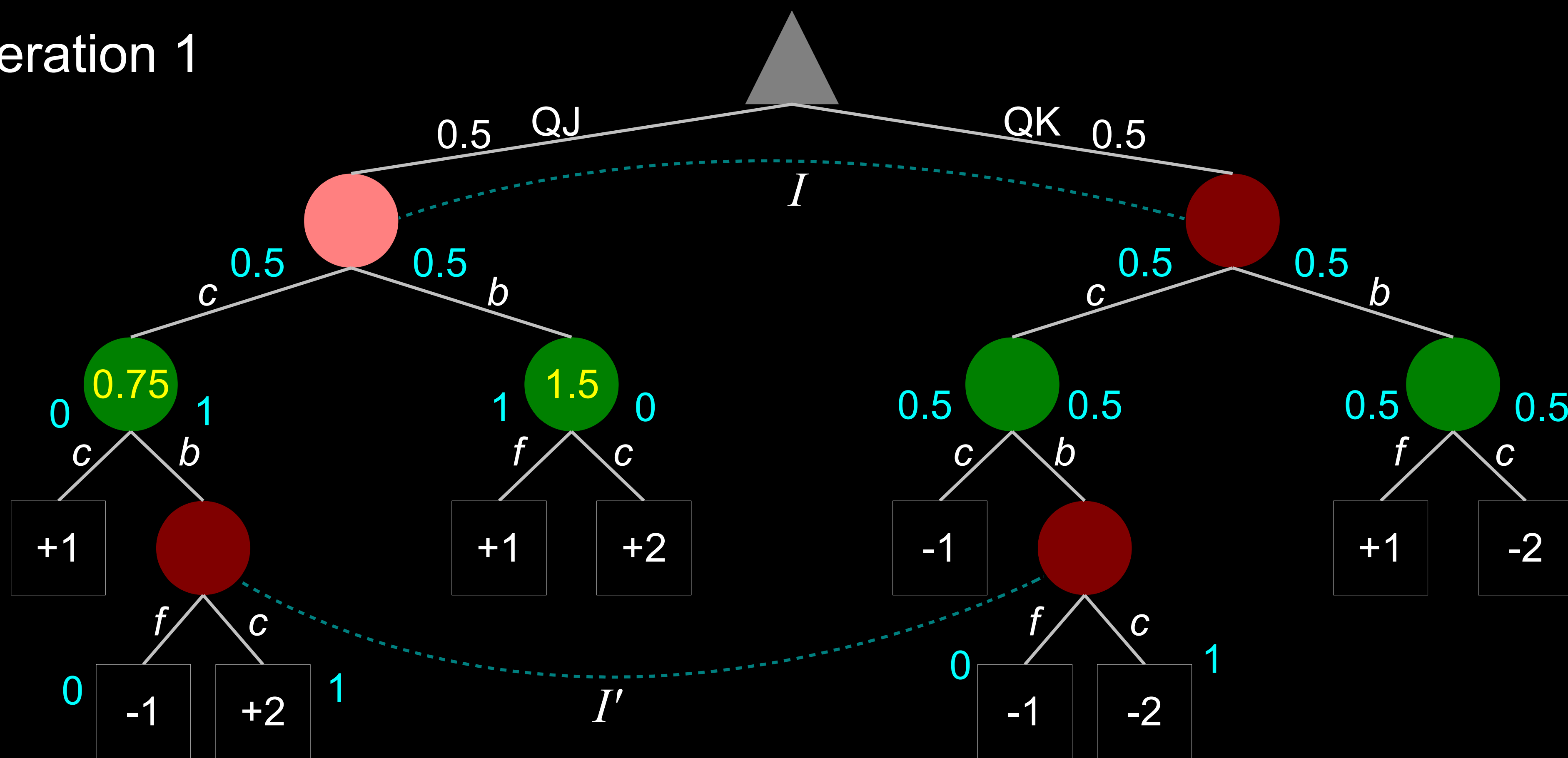


$\sigma_2(\,f\,)$: -1 > -1.5 ⬆ $\sigma_2(\,c\,)$: -2 < -1.5 ⬇

“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 1

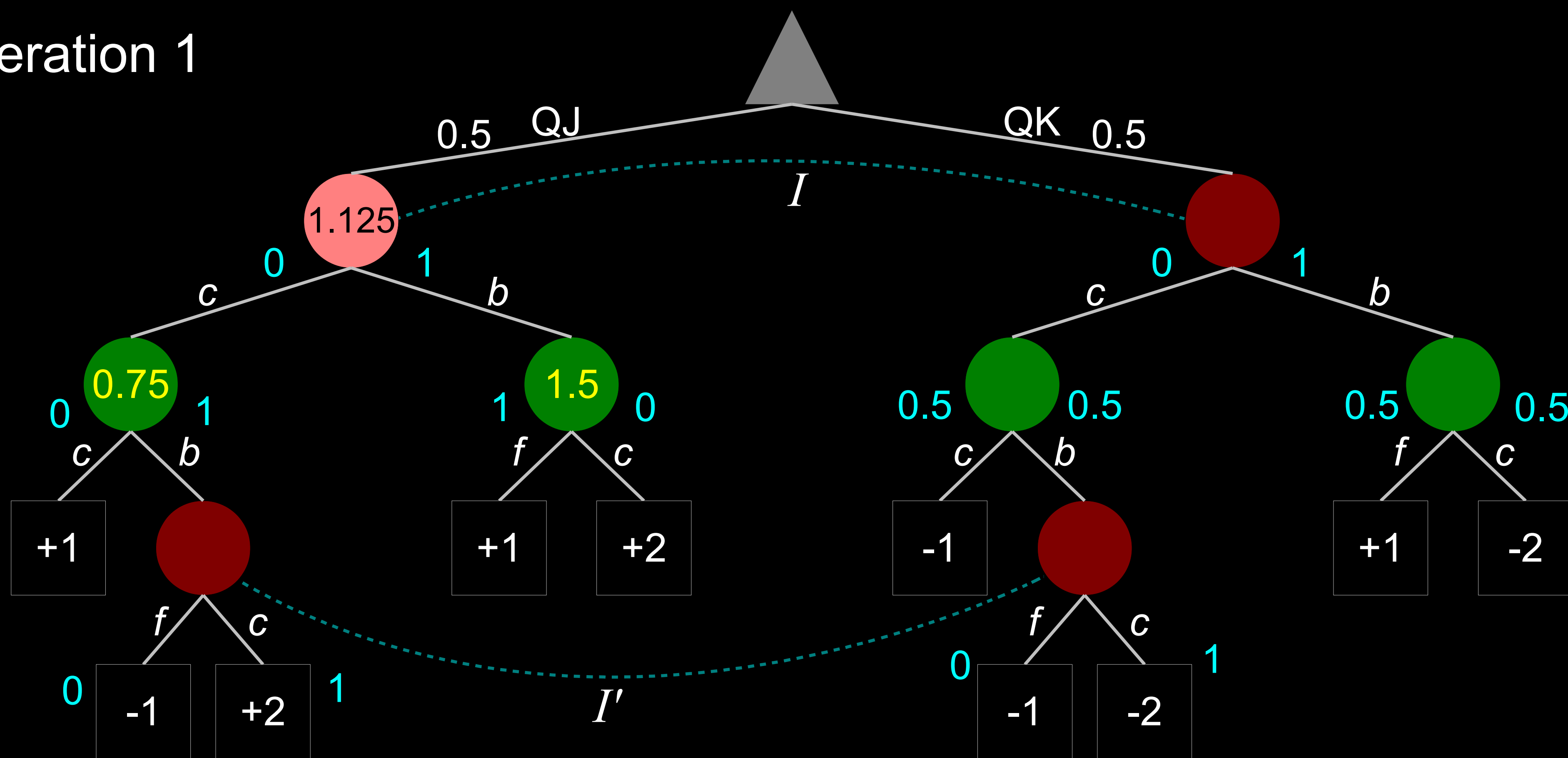


EV = 0.5(0.75) + 0.5(1.5) = 1.125

"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 1

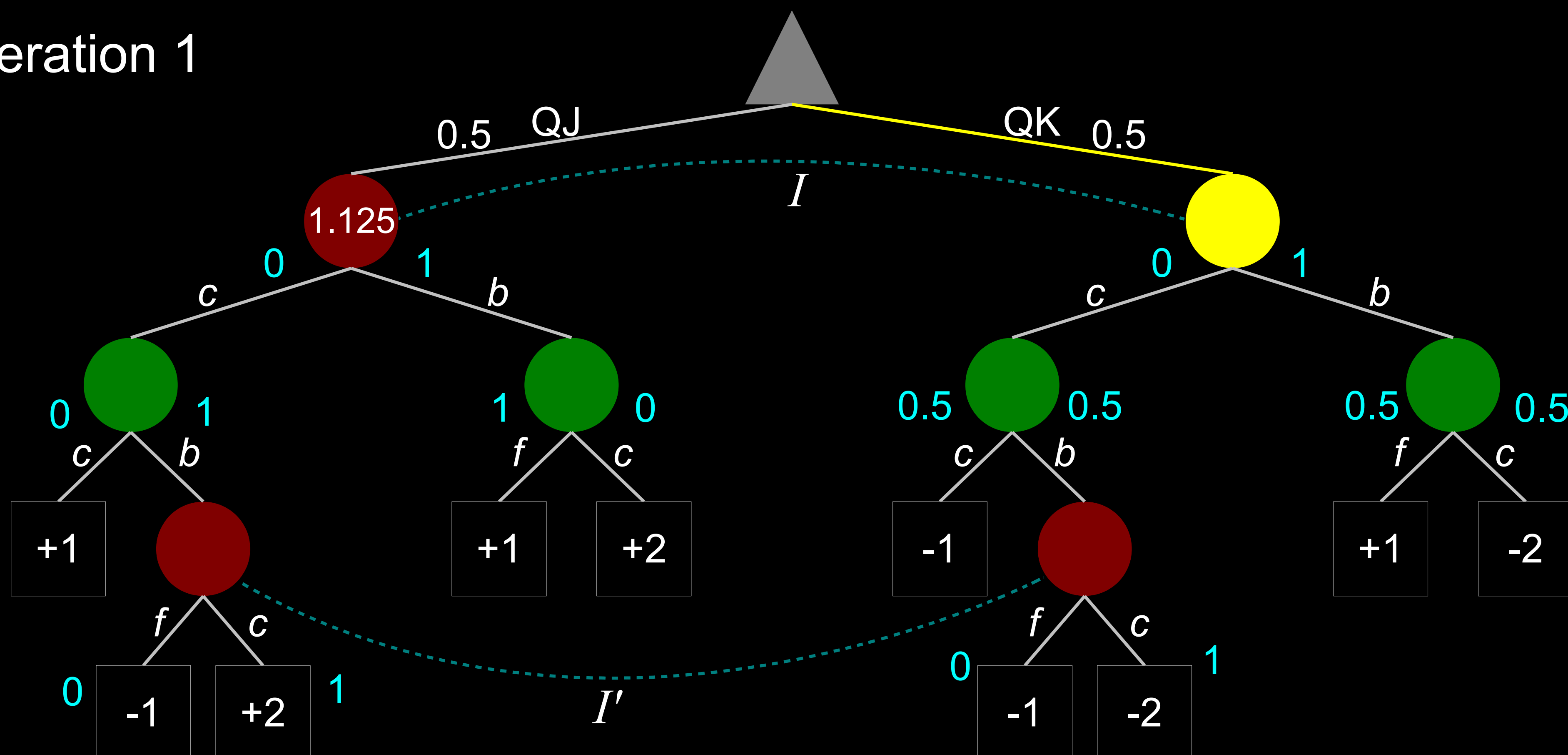


$\sigma_1(\,c\,)$: 0.75 < 1.125 ⬇ $\sigma_1(\,b\,)$: 1.5 > 1.125 ⬆

“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

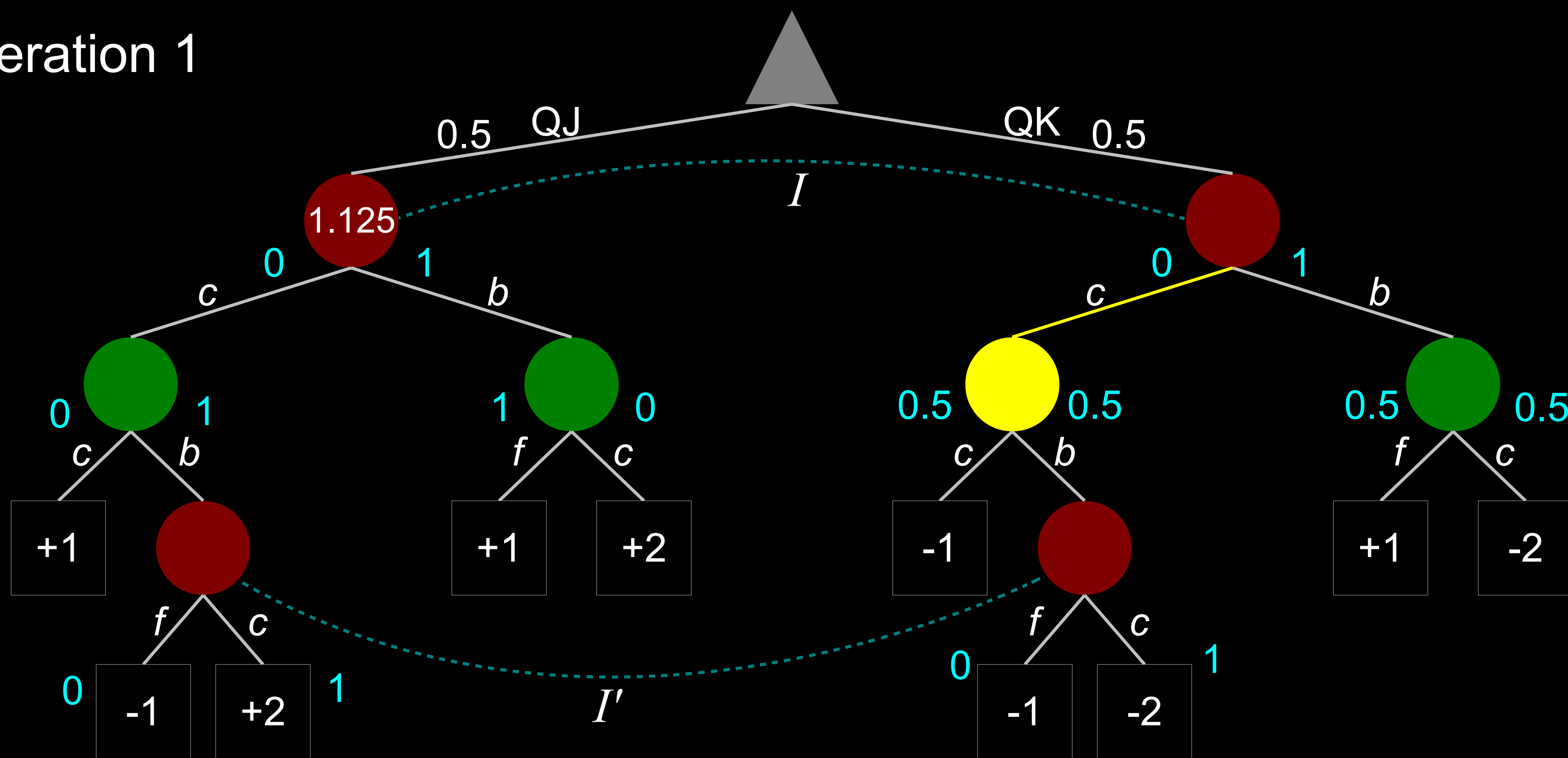
- Iteration 1



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

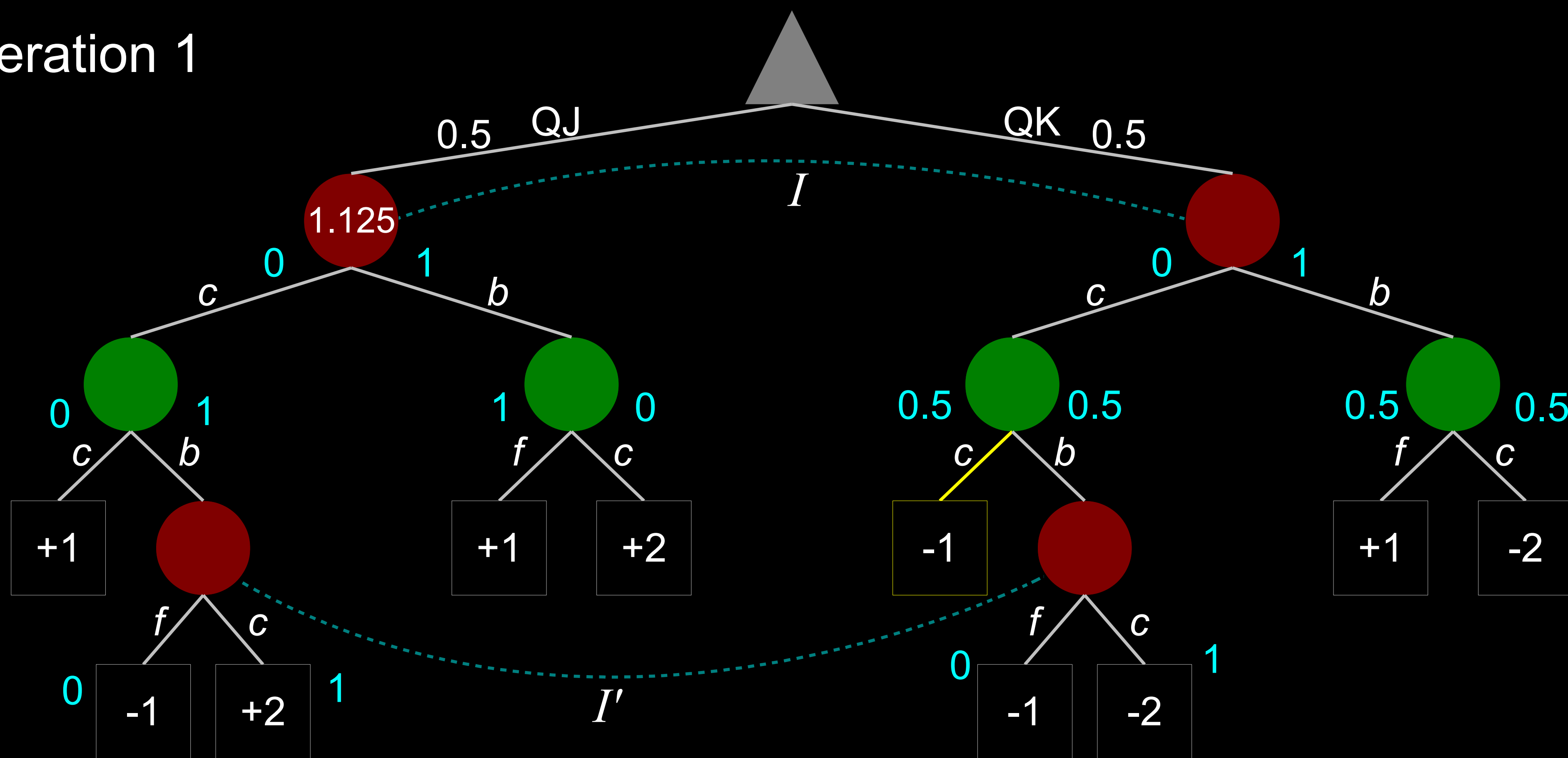
- Iteration 1



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

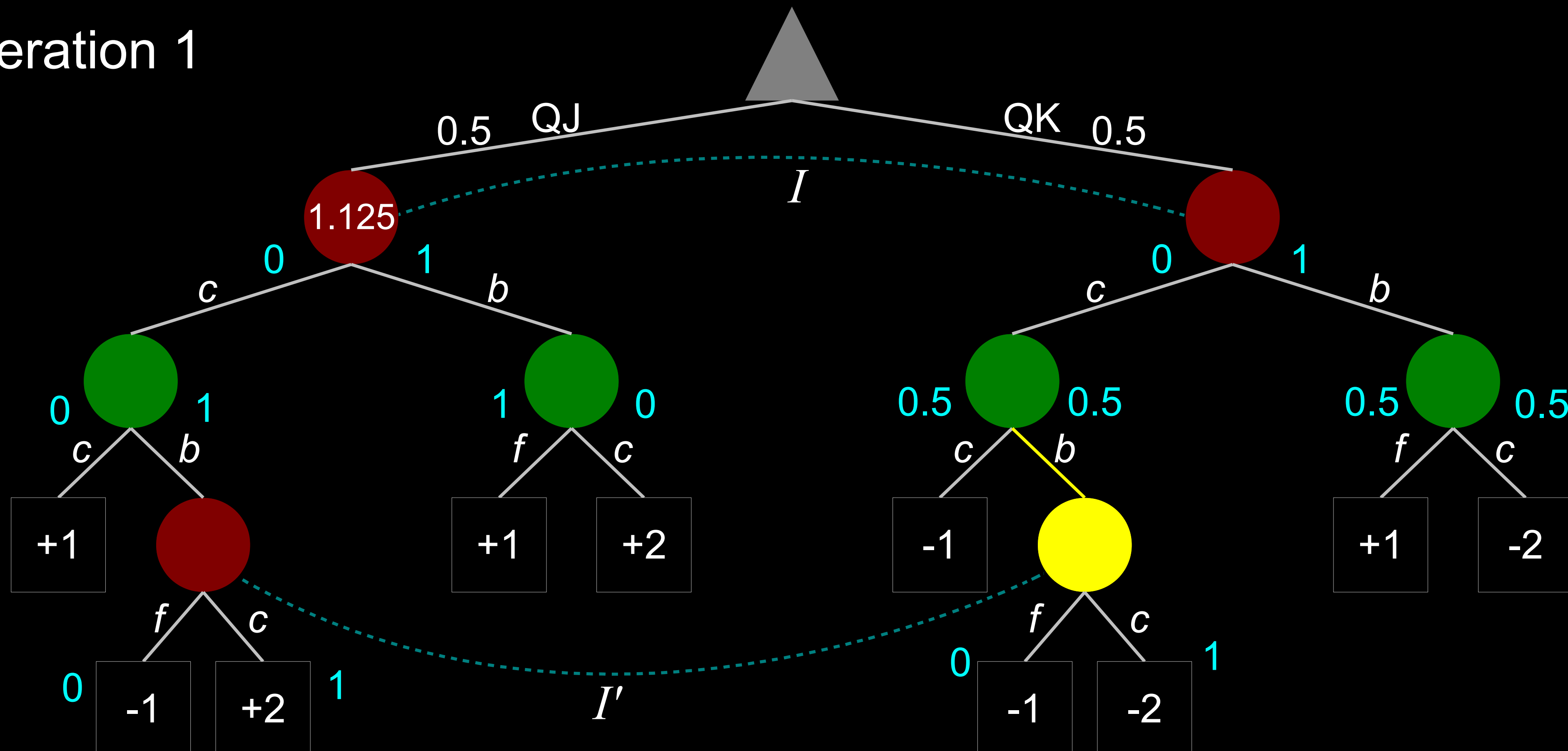
- Iteration 1



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

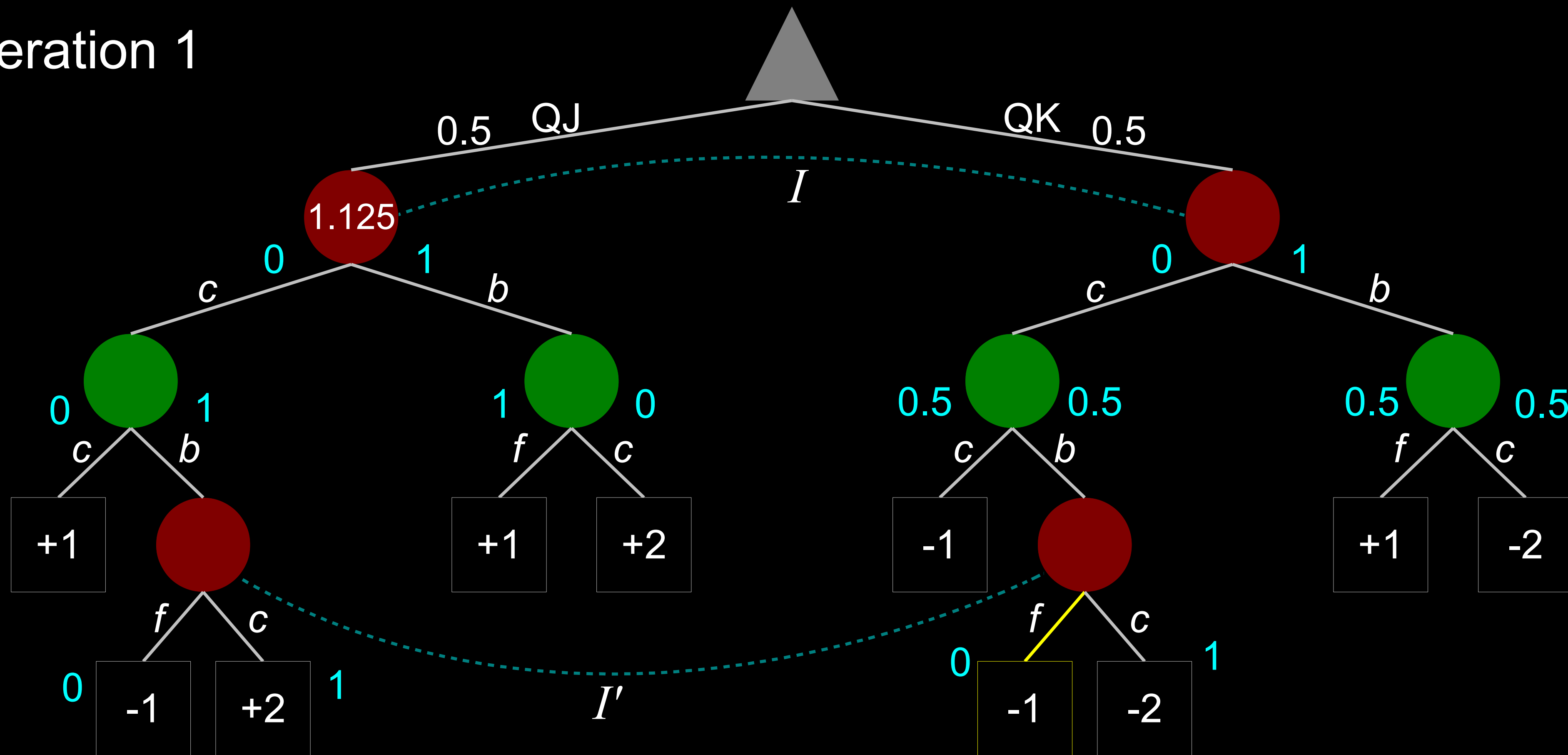
- Iteration 1



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

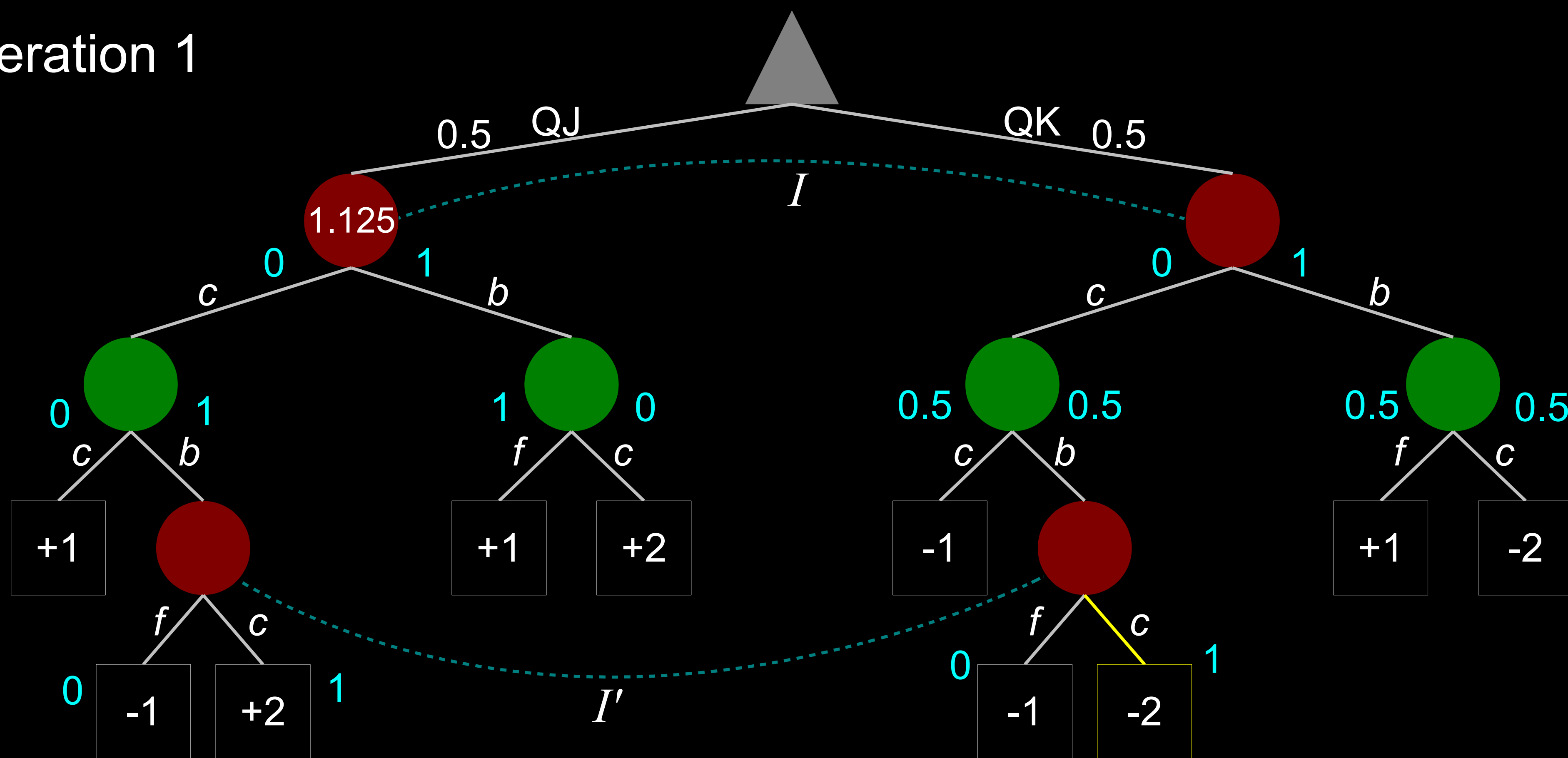
- Iteration 1



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

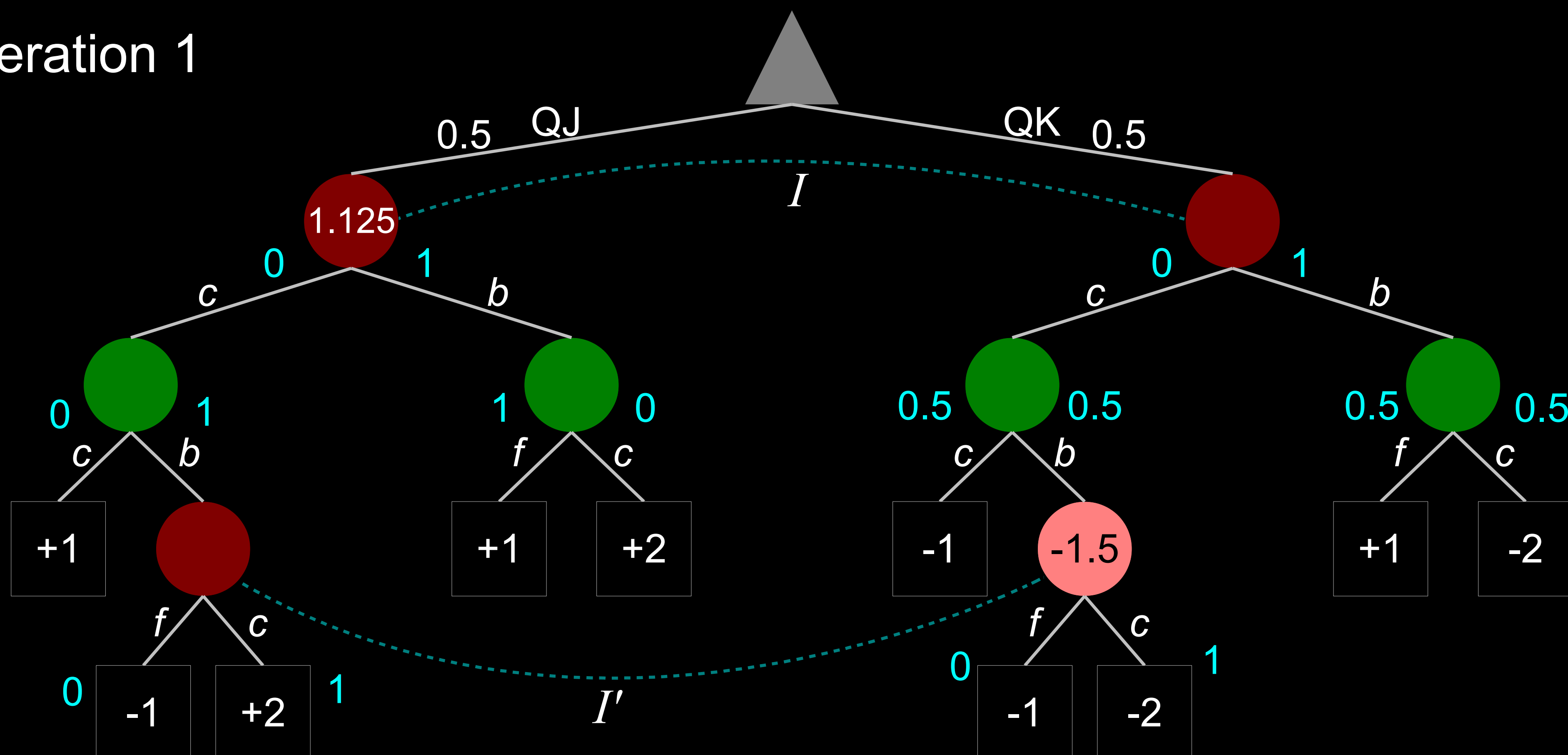
- Iteration 1



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

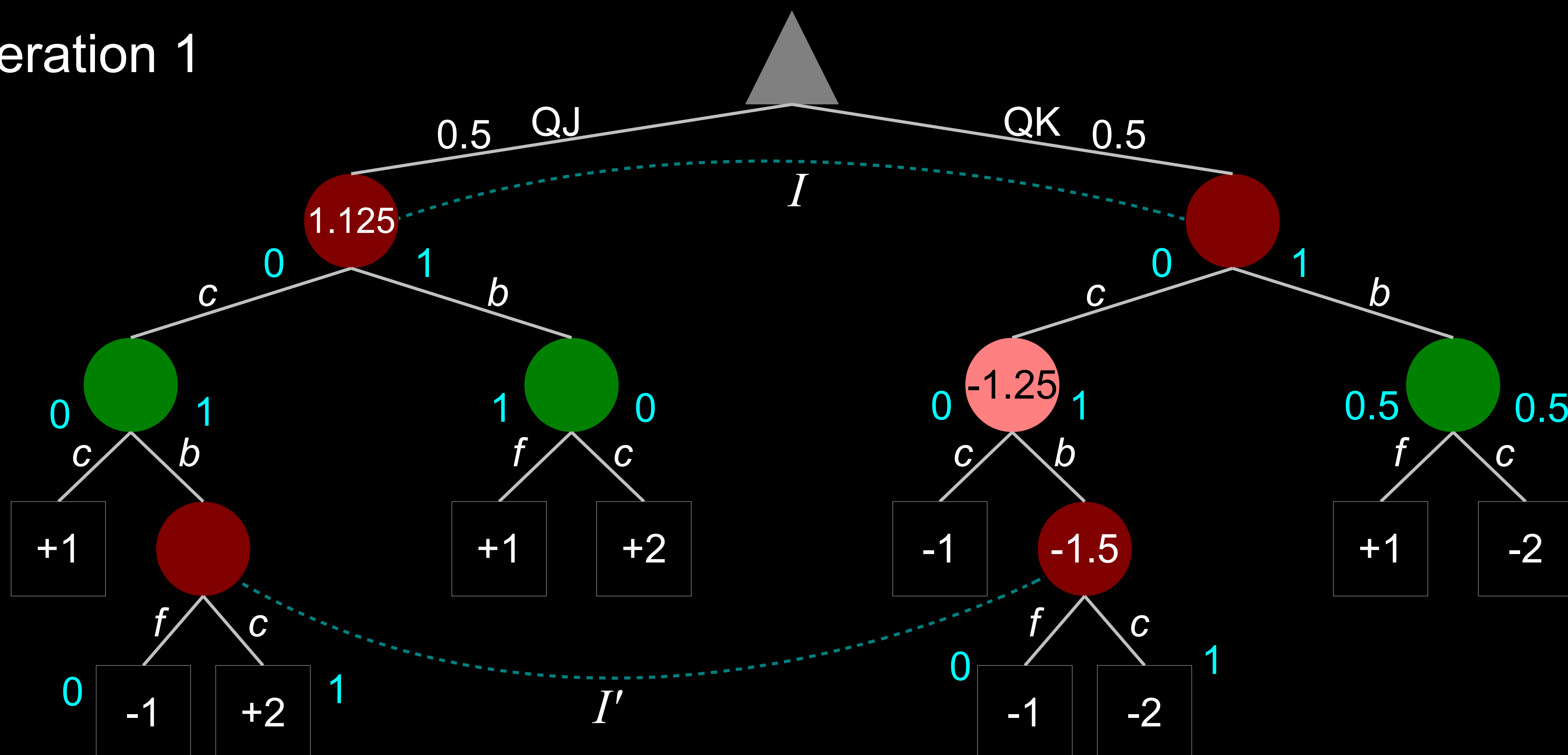
- Iteration 1



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

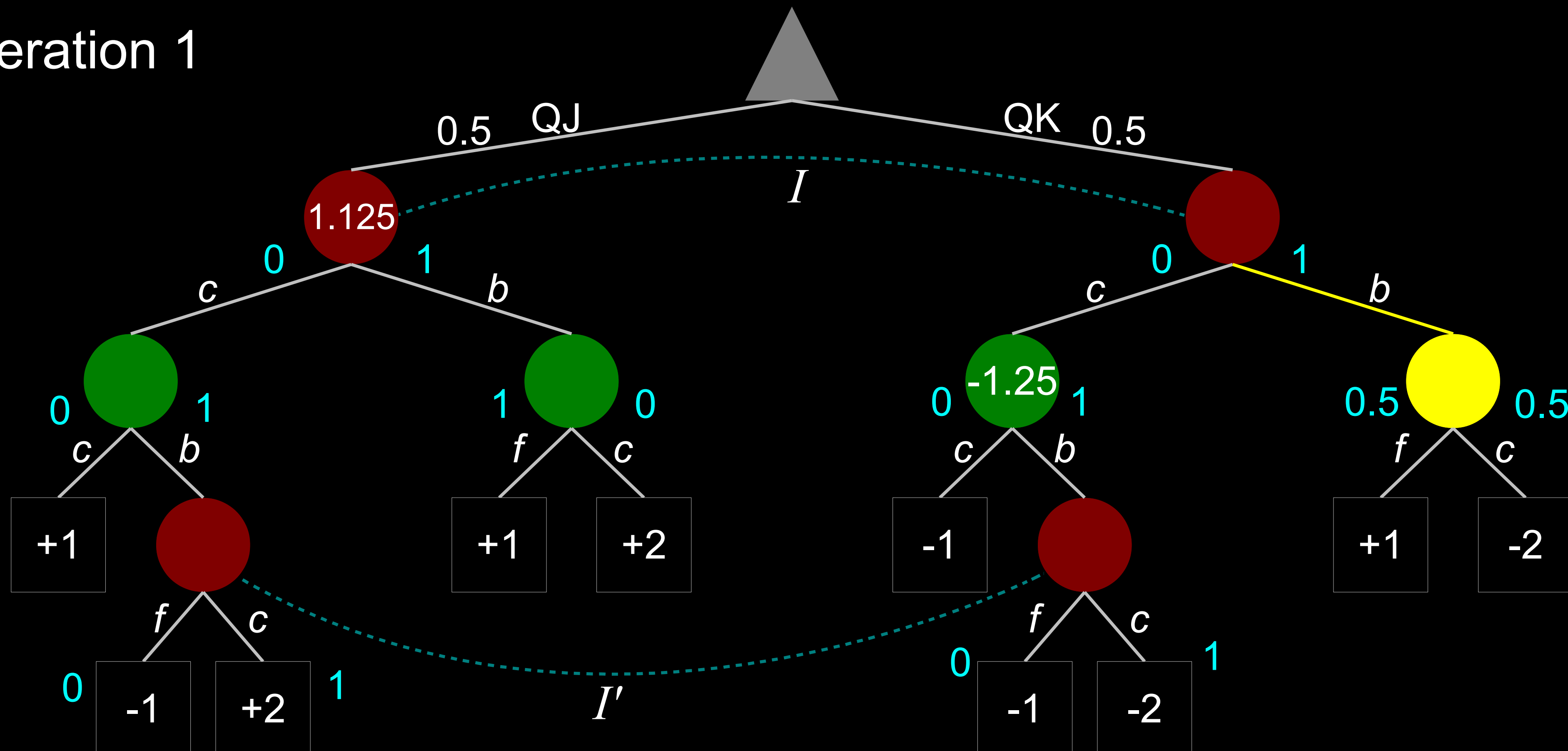
- Iteration 1



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

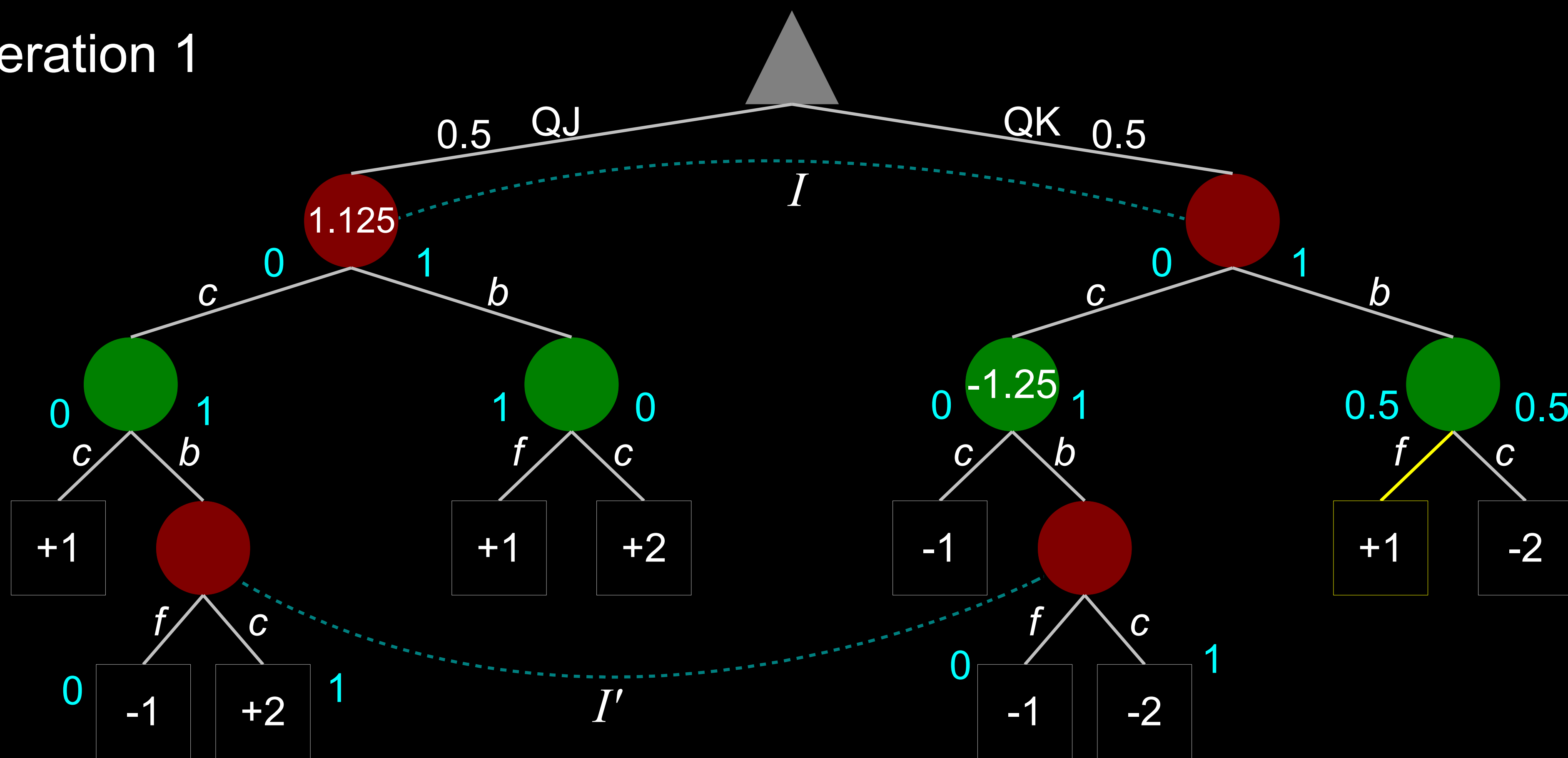
- Iteration 1



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

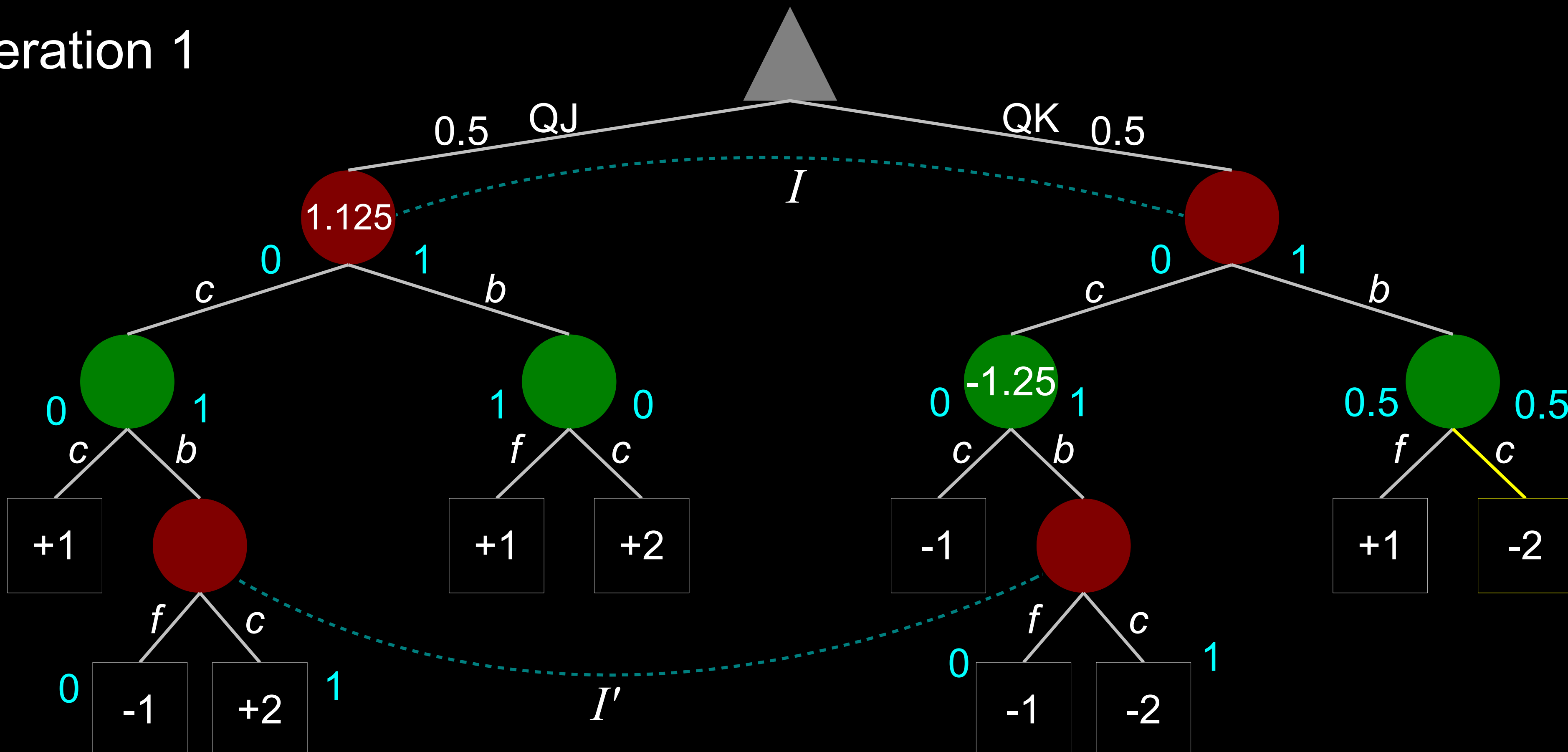
- Iteration 1



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

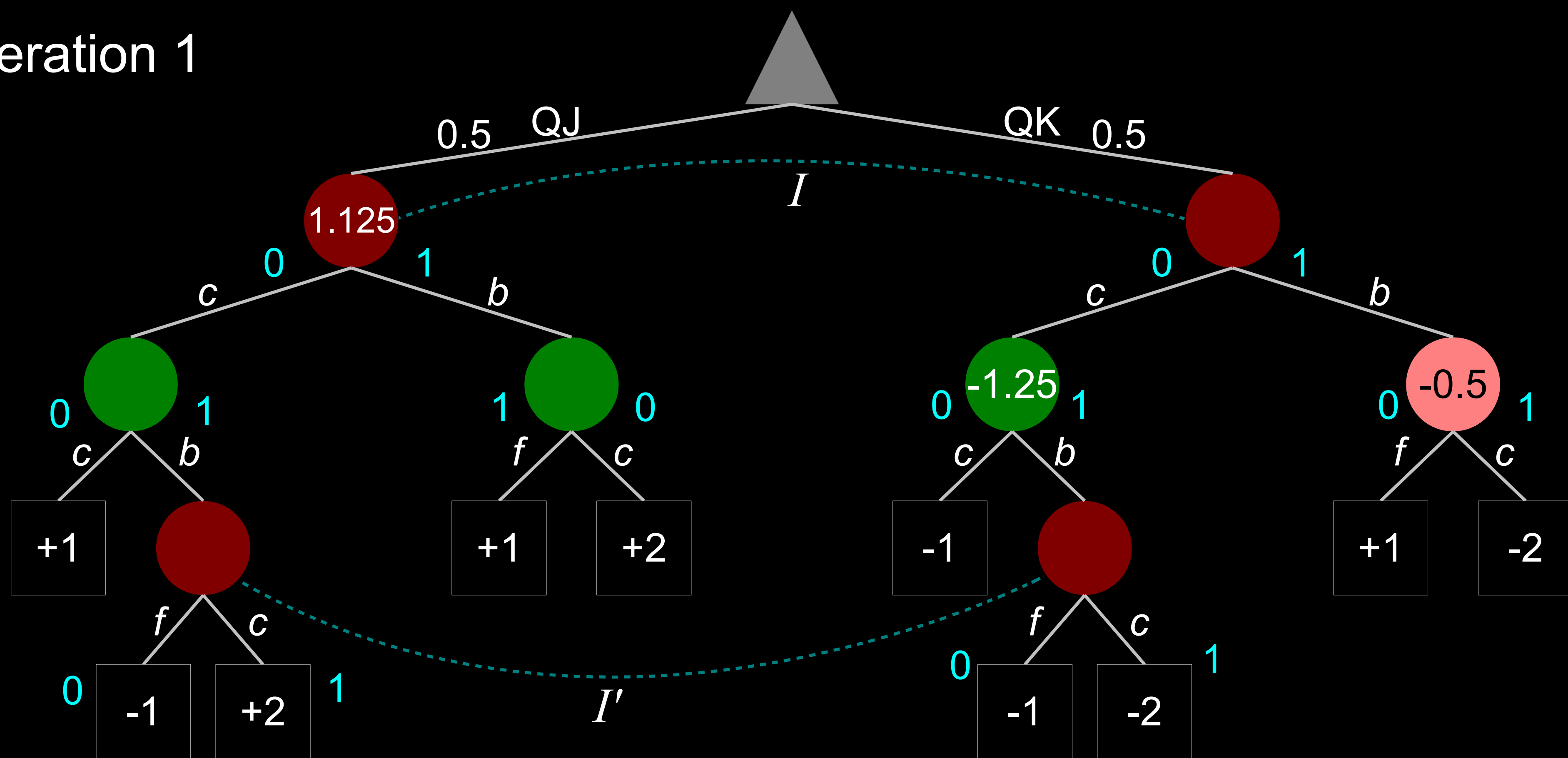
- Iteration 1



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

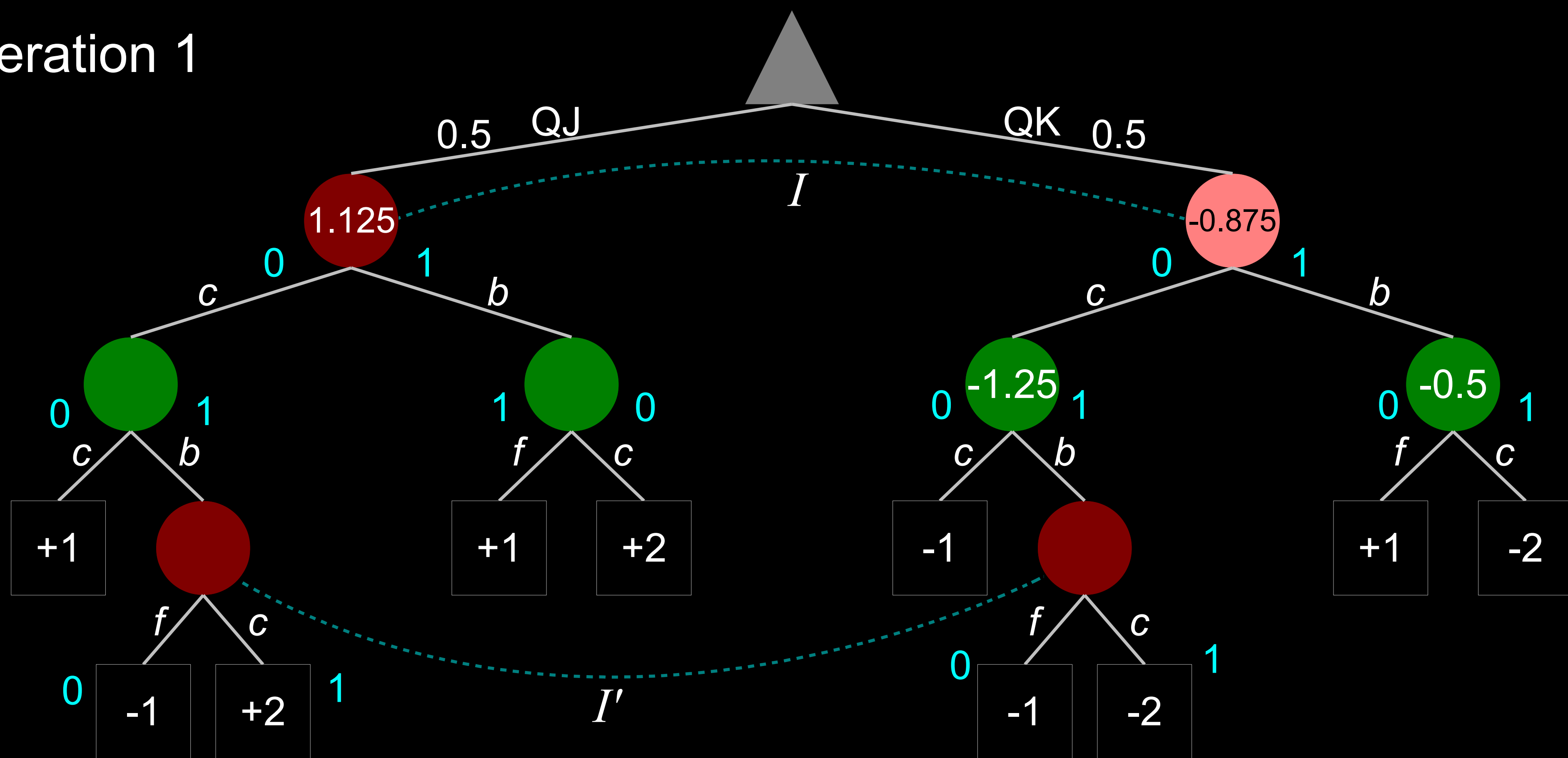
- Iteration 1



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

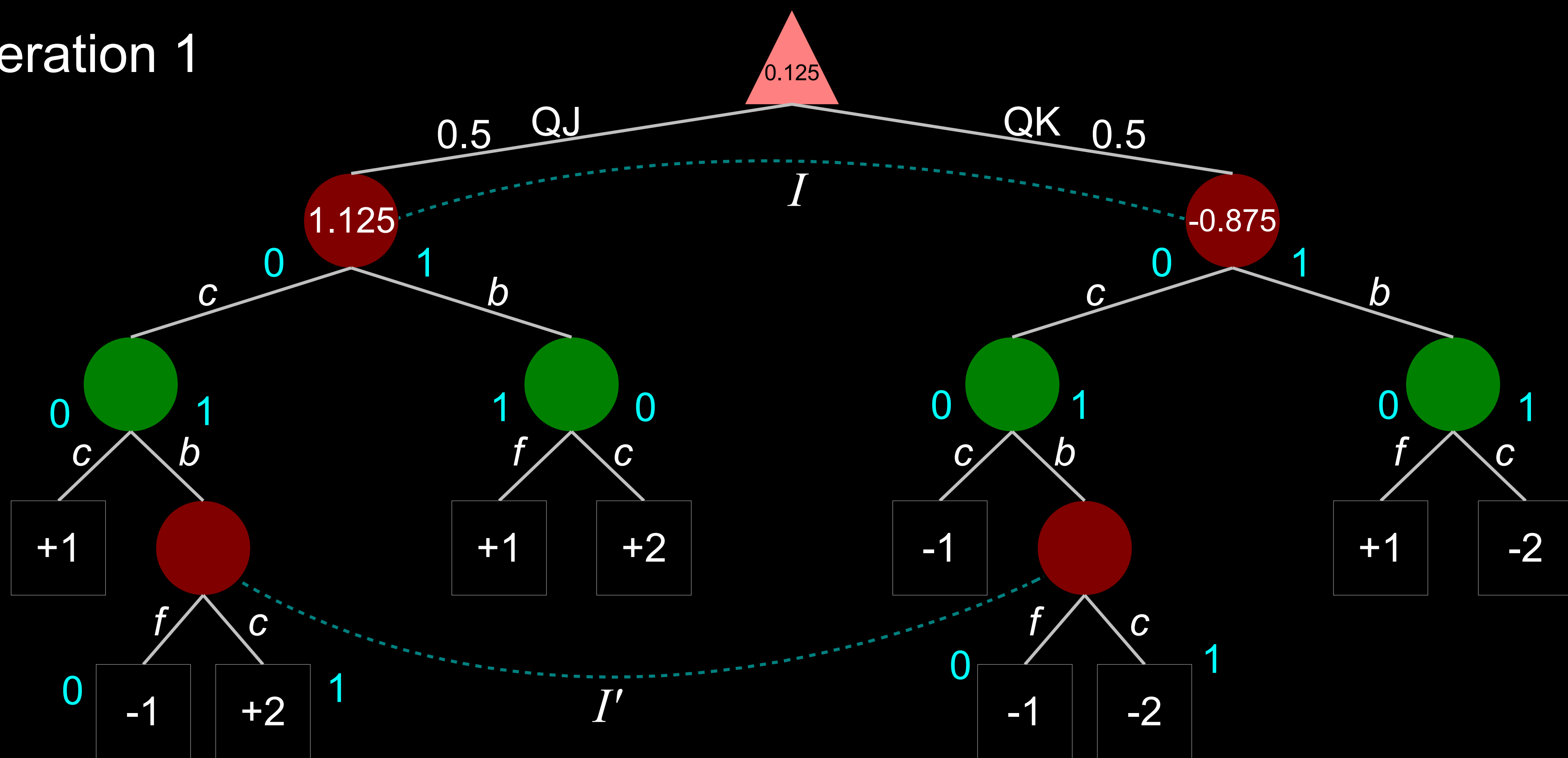
- Iteration 1



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

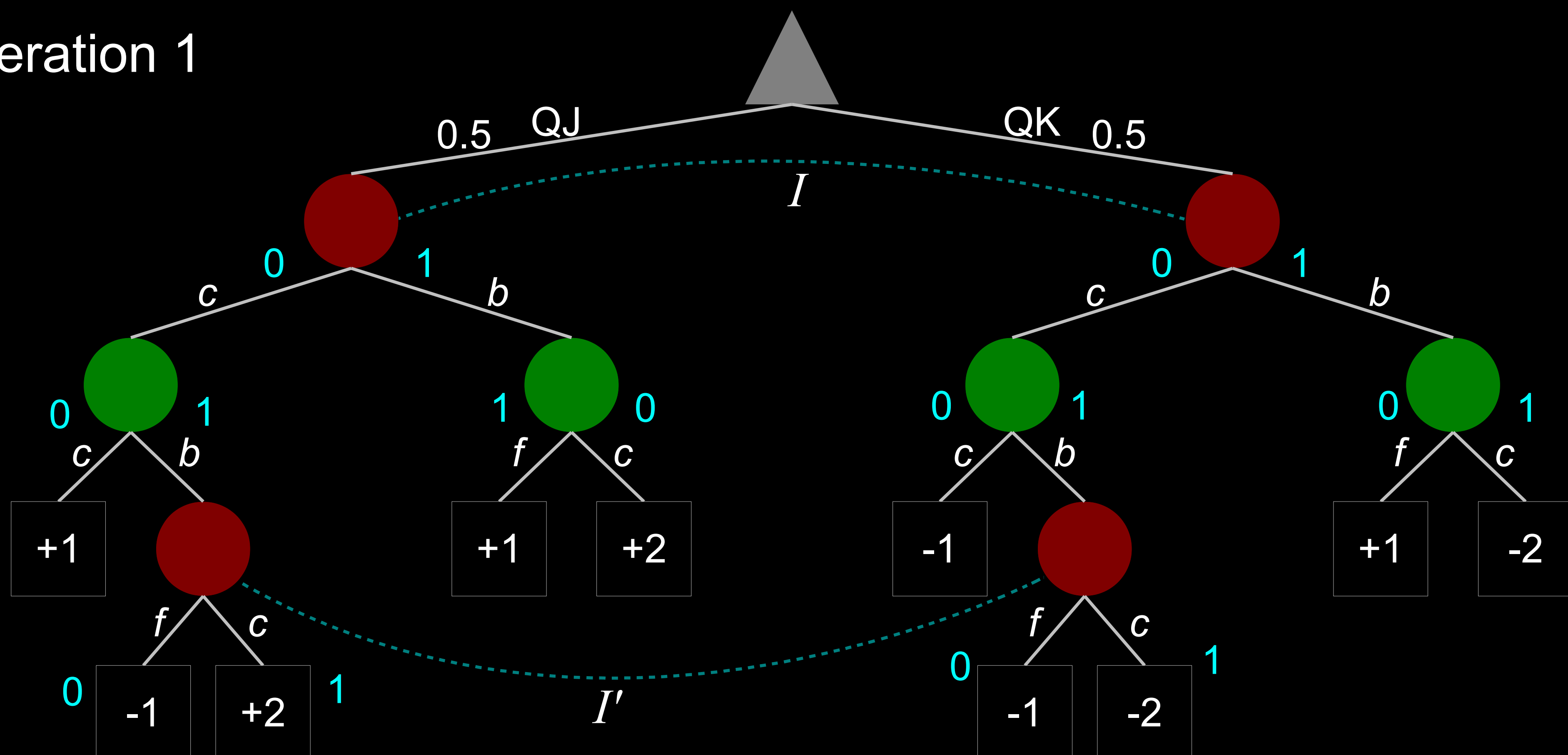
- Iteration 1



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

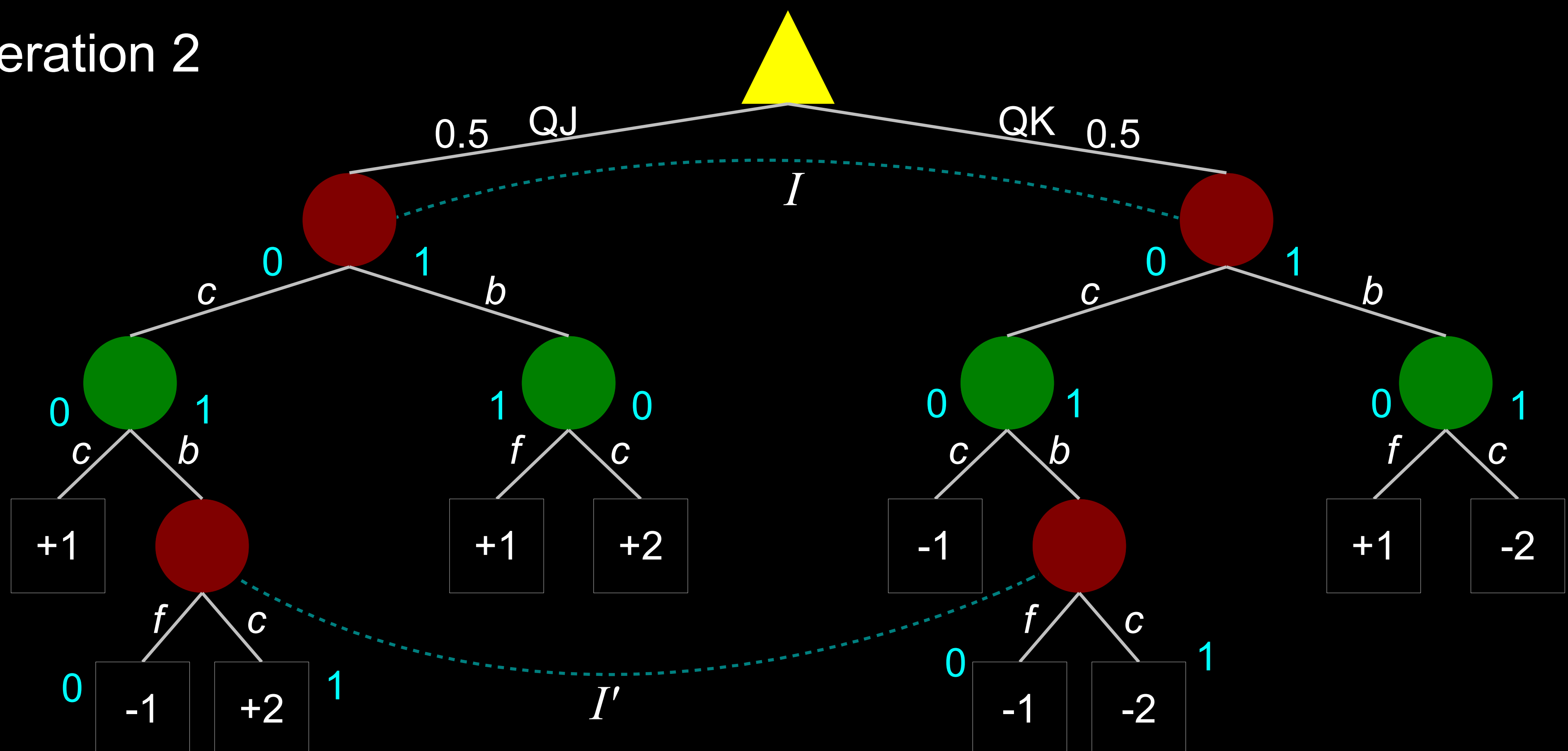
- Iteration 1



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

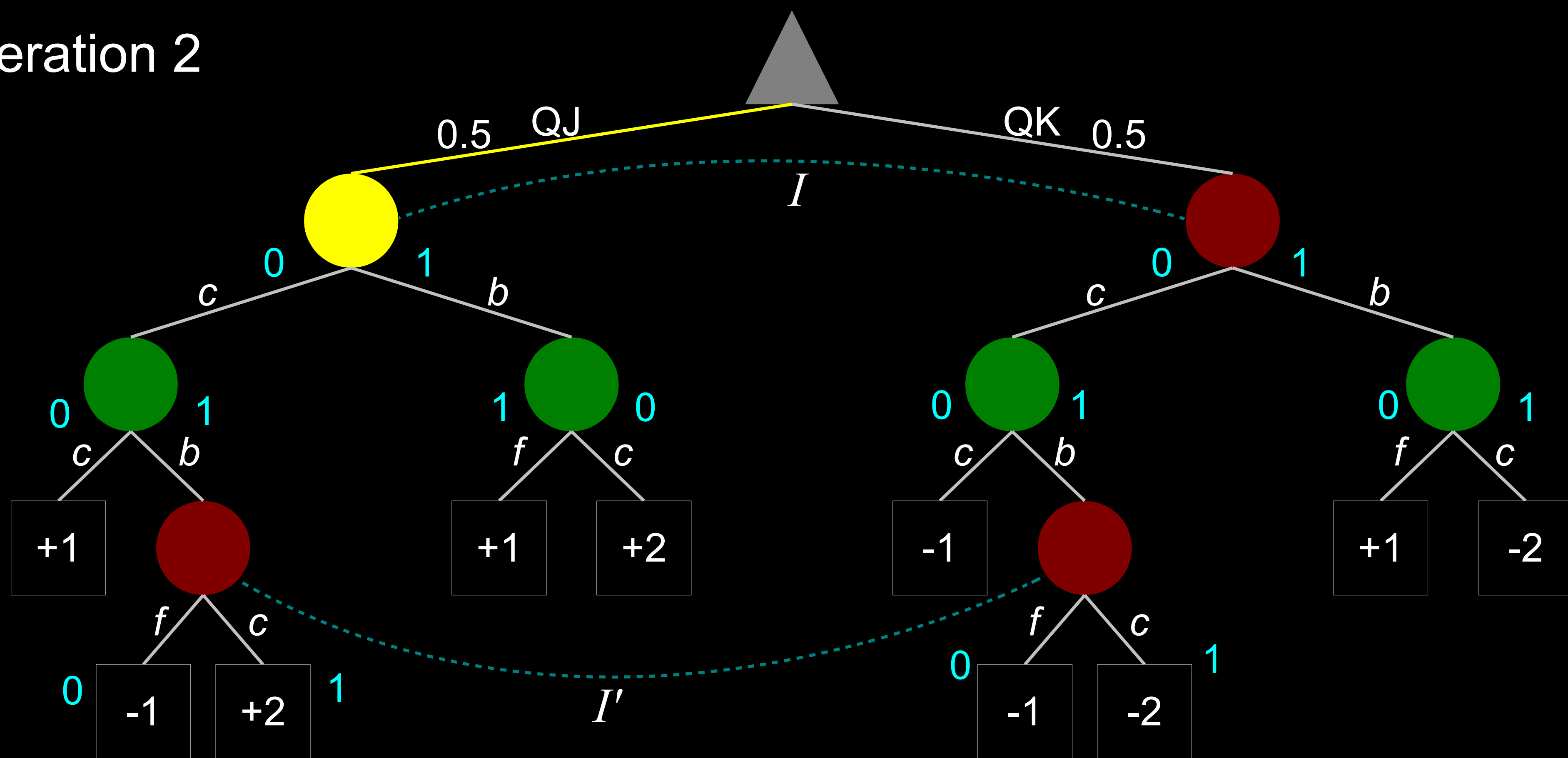
- Iteration 2



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

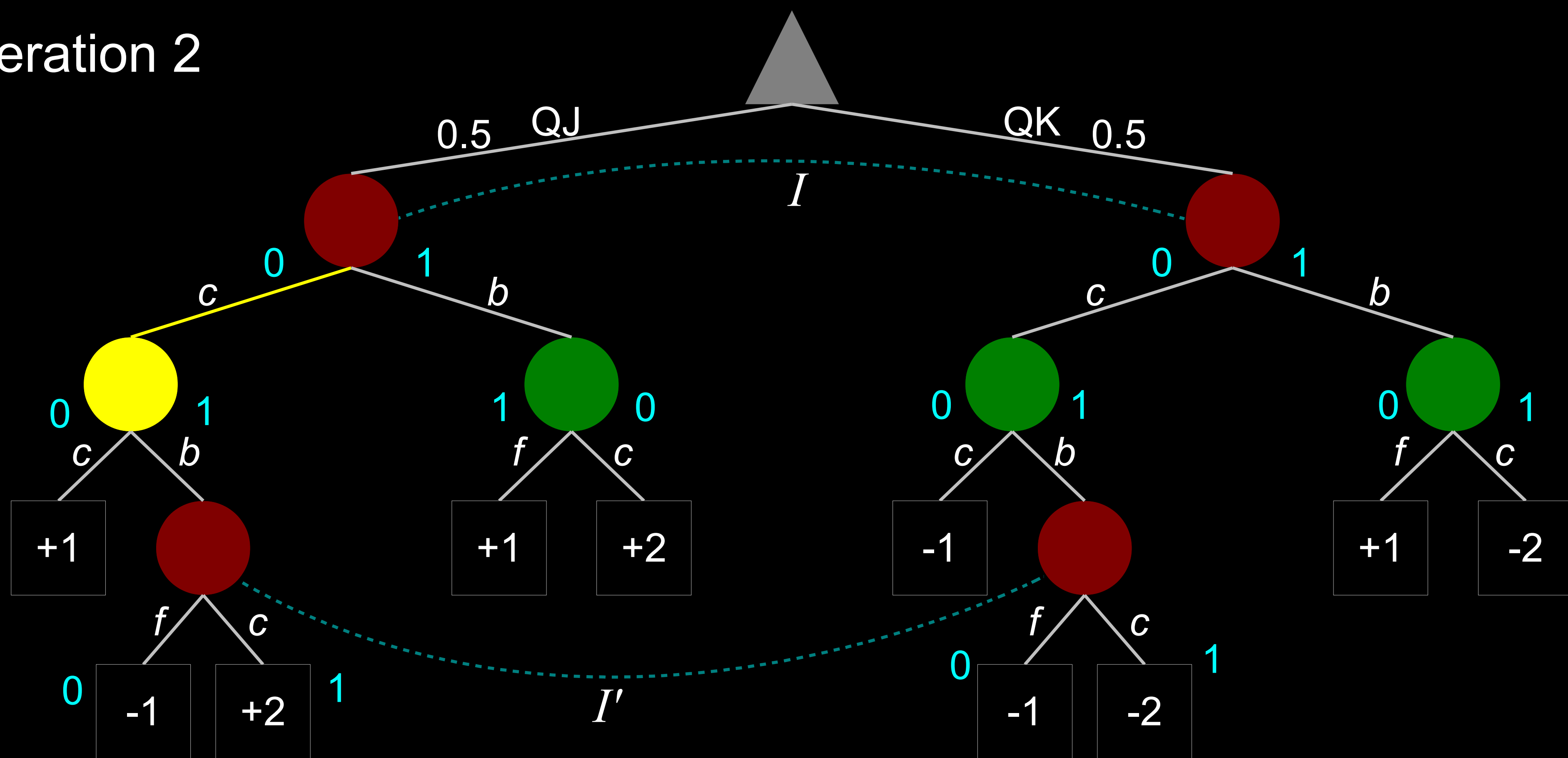
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

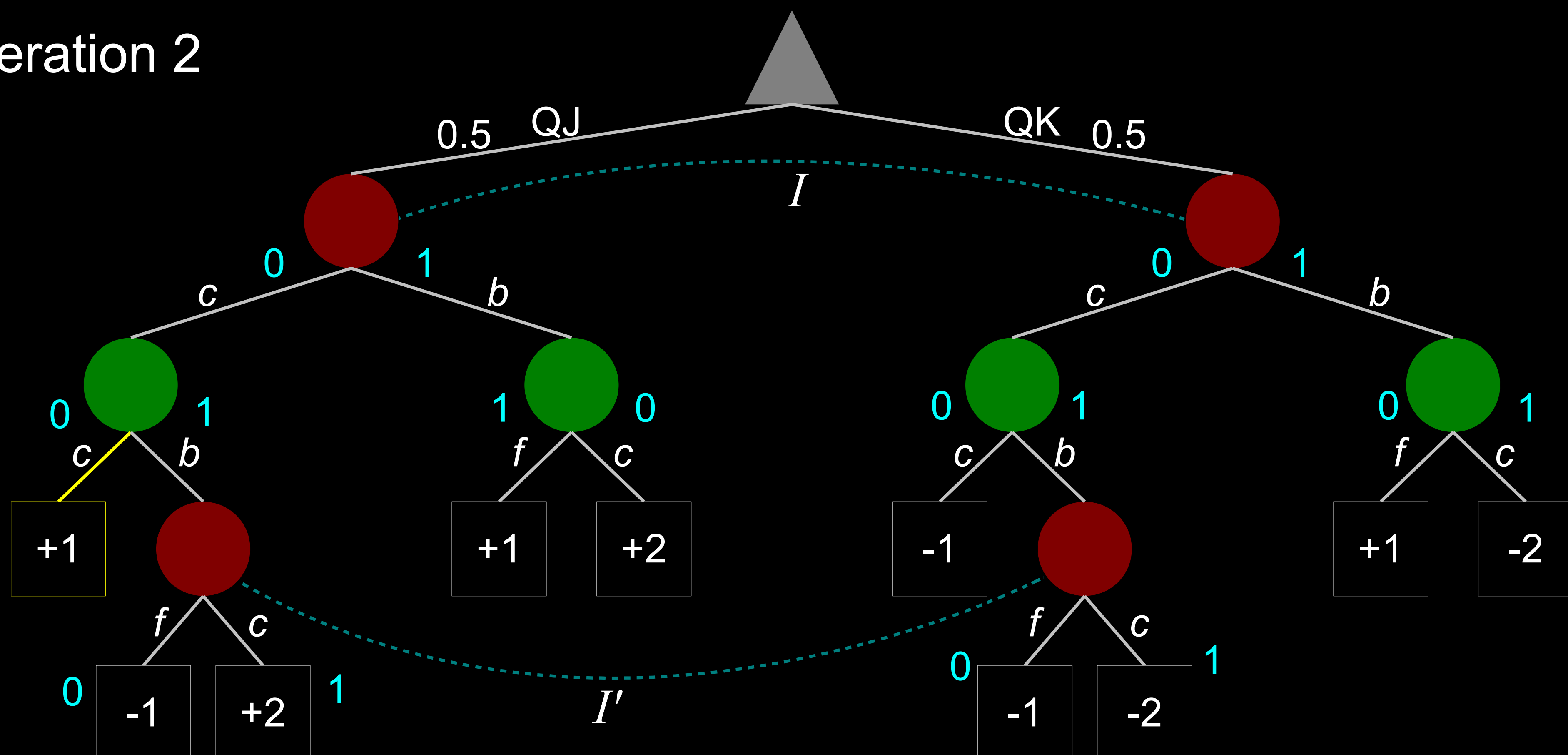
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

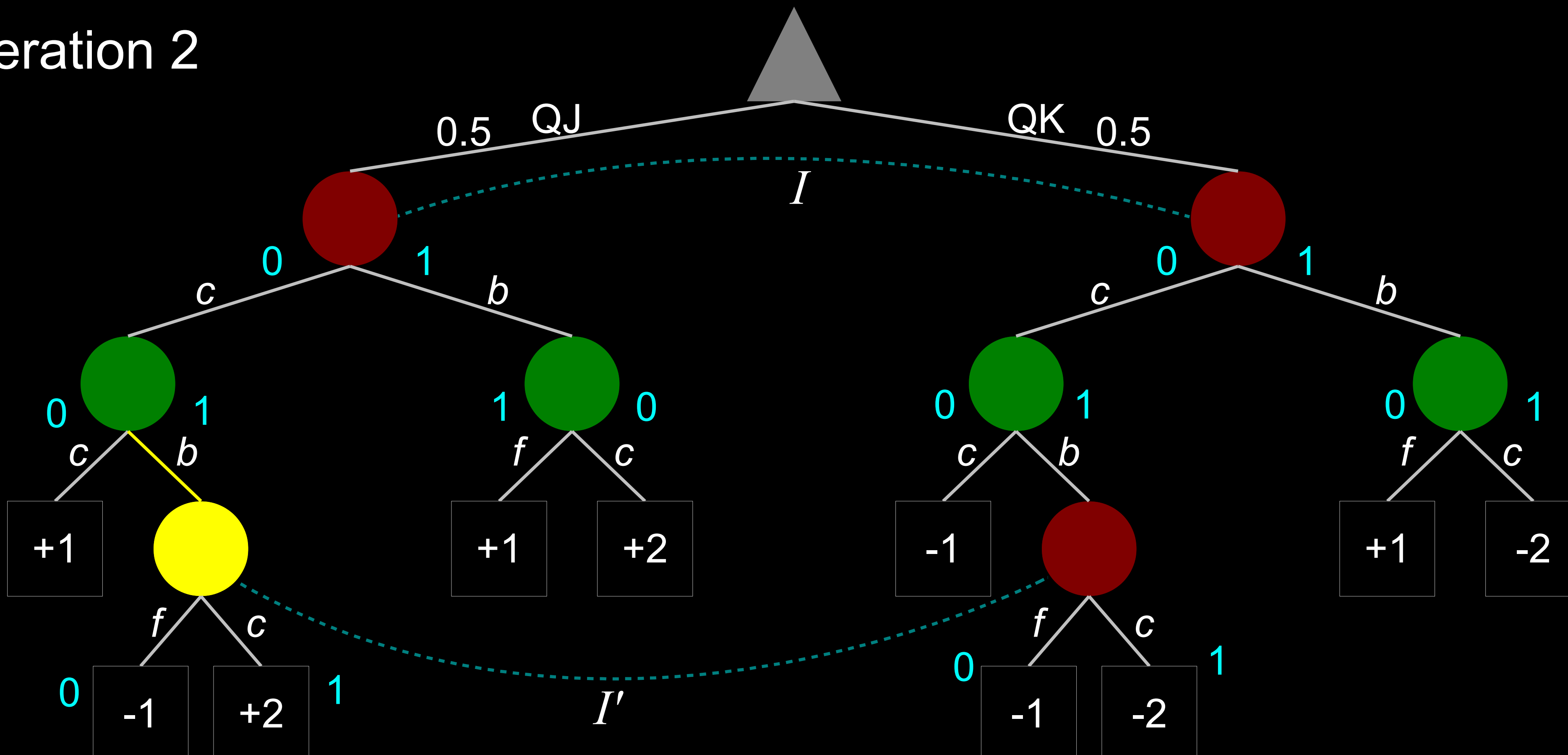
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

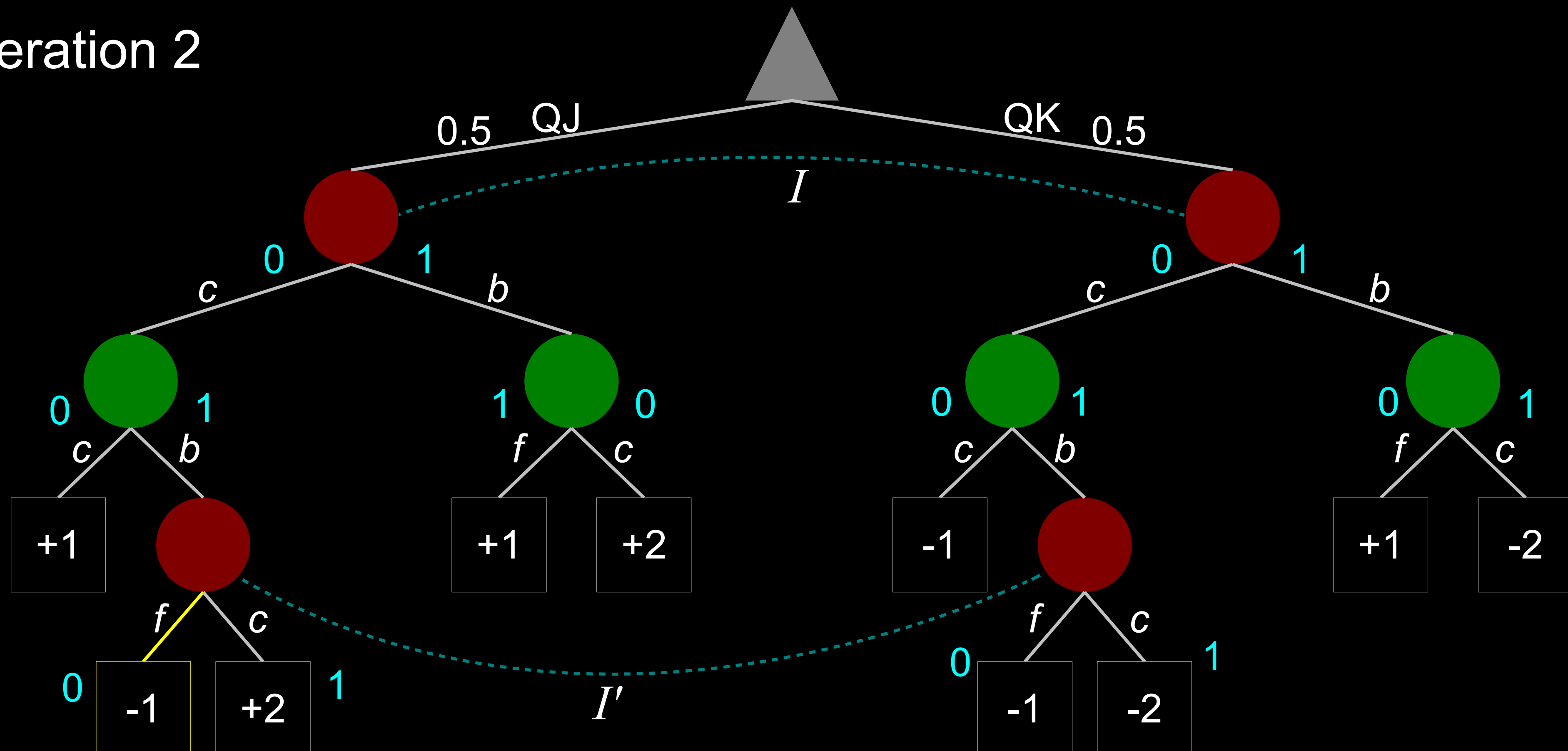
- Iteration 2



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

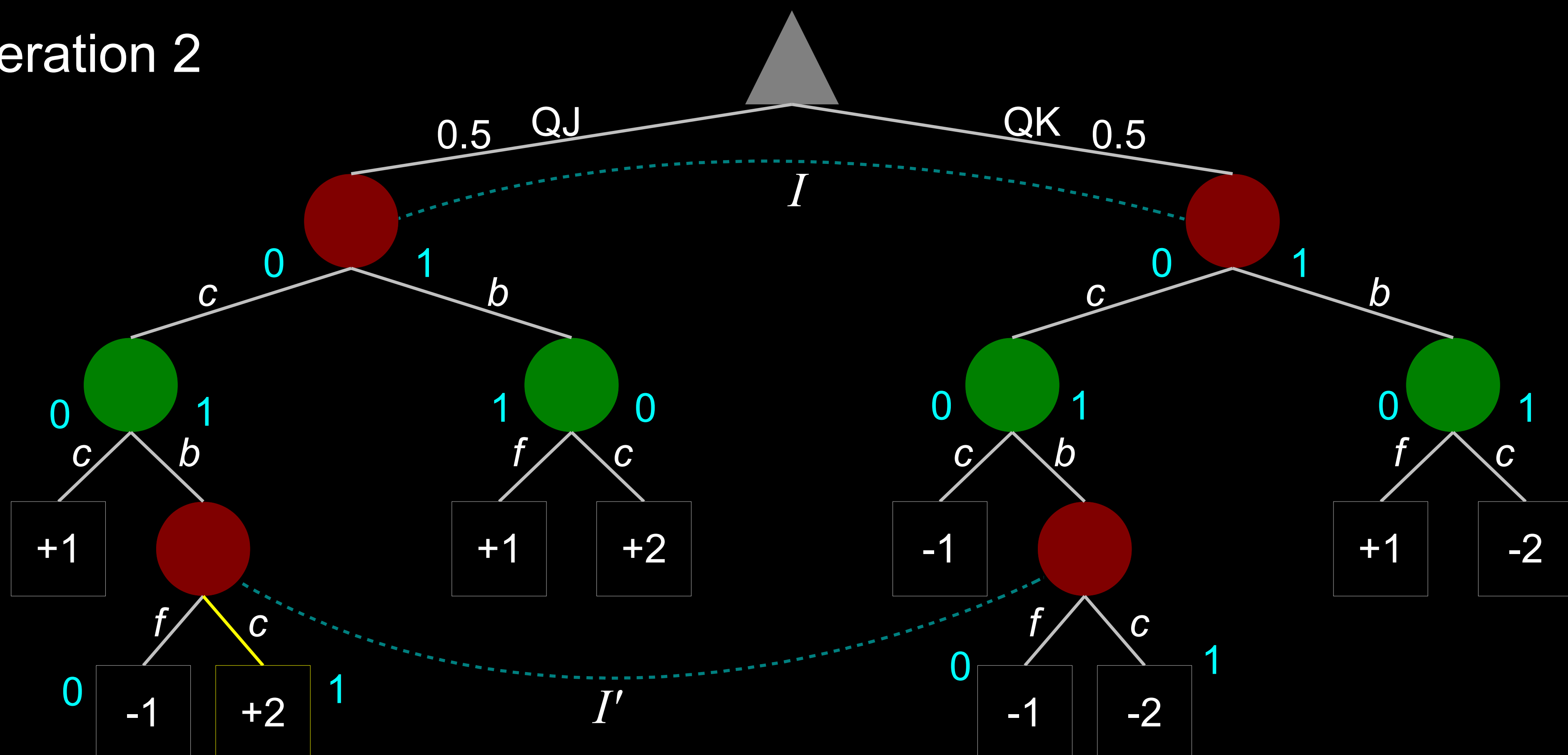
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

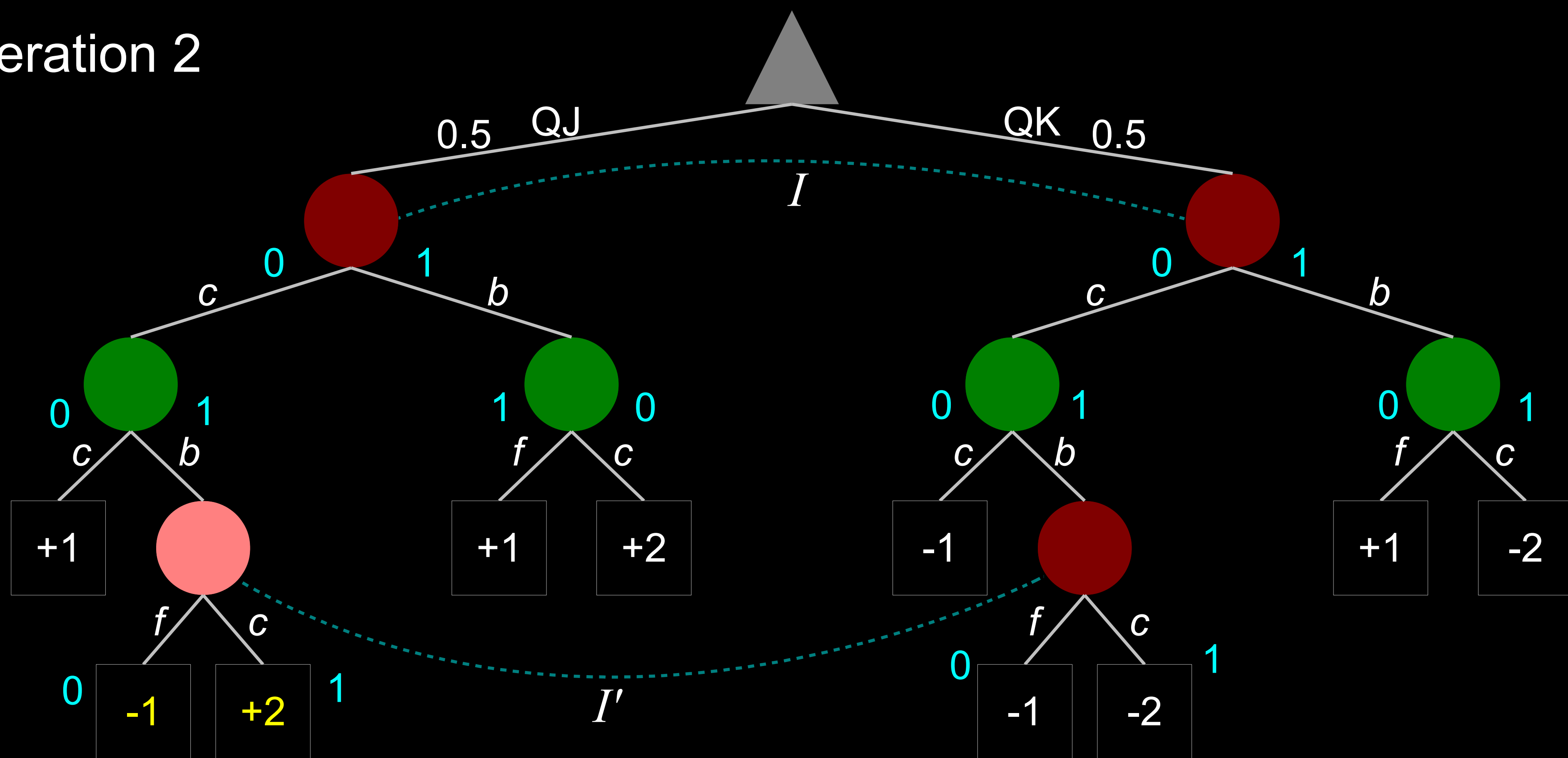
- Iteration 2



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 2

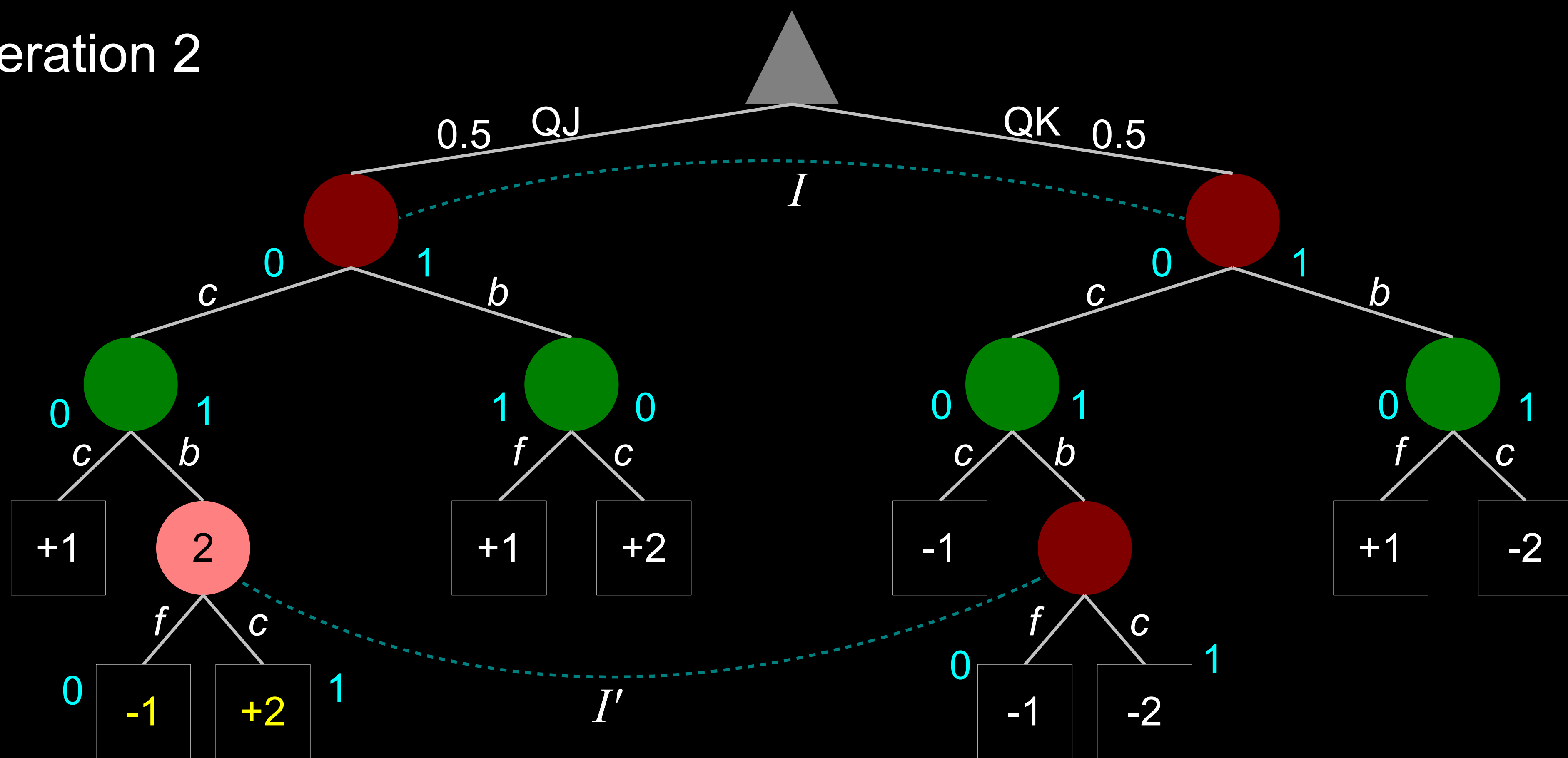


EV = 0(-1) + 1(+2) = 2

“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 2

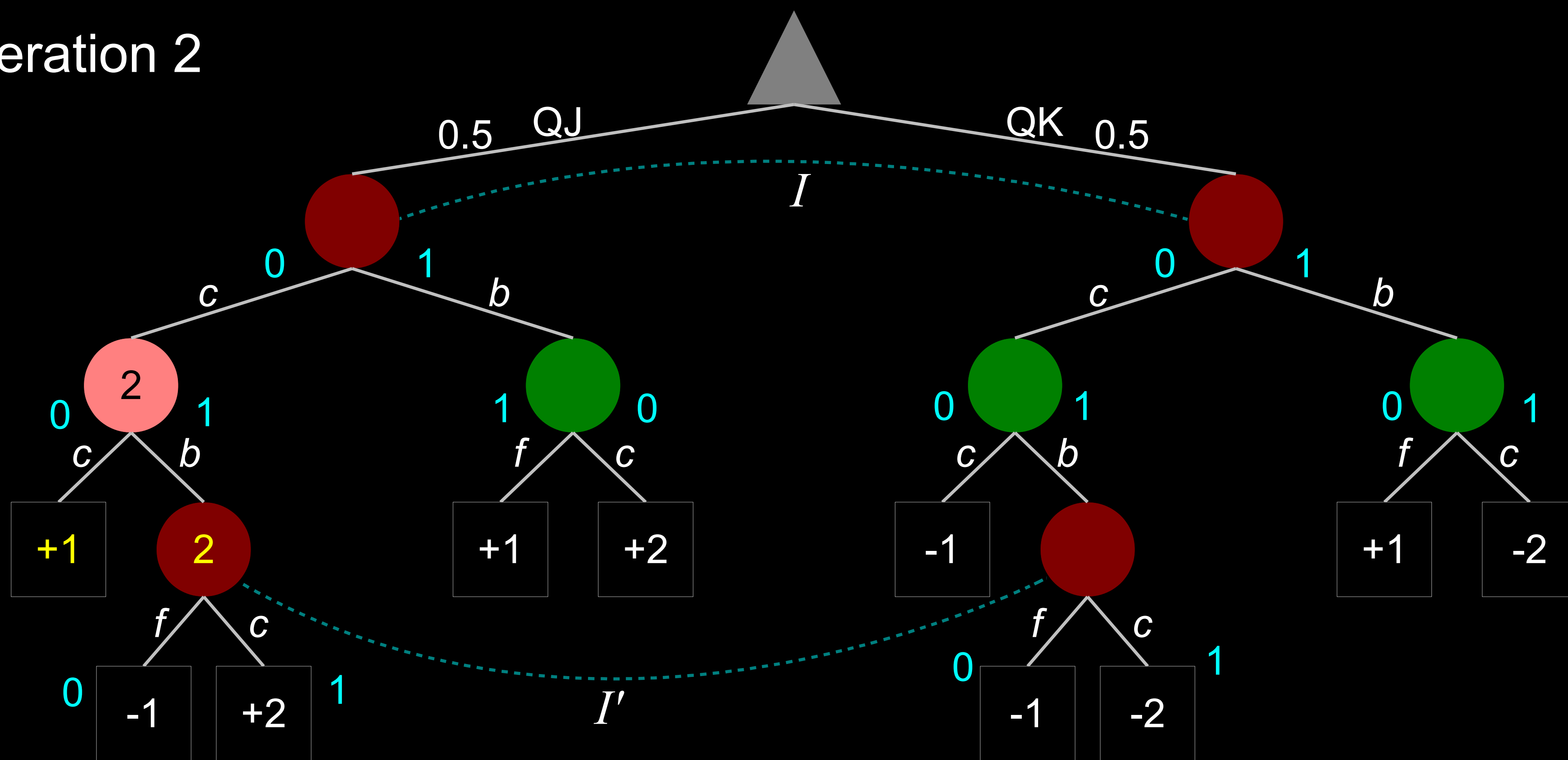


$\sigma_1(\, f\,)$: -1 < 2 ⬇ $\sigma_1(\, c\,)$: +2 = 2 no change

“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 2

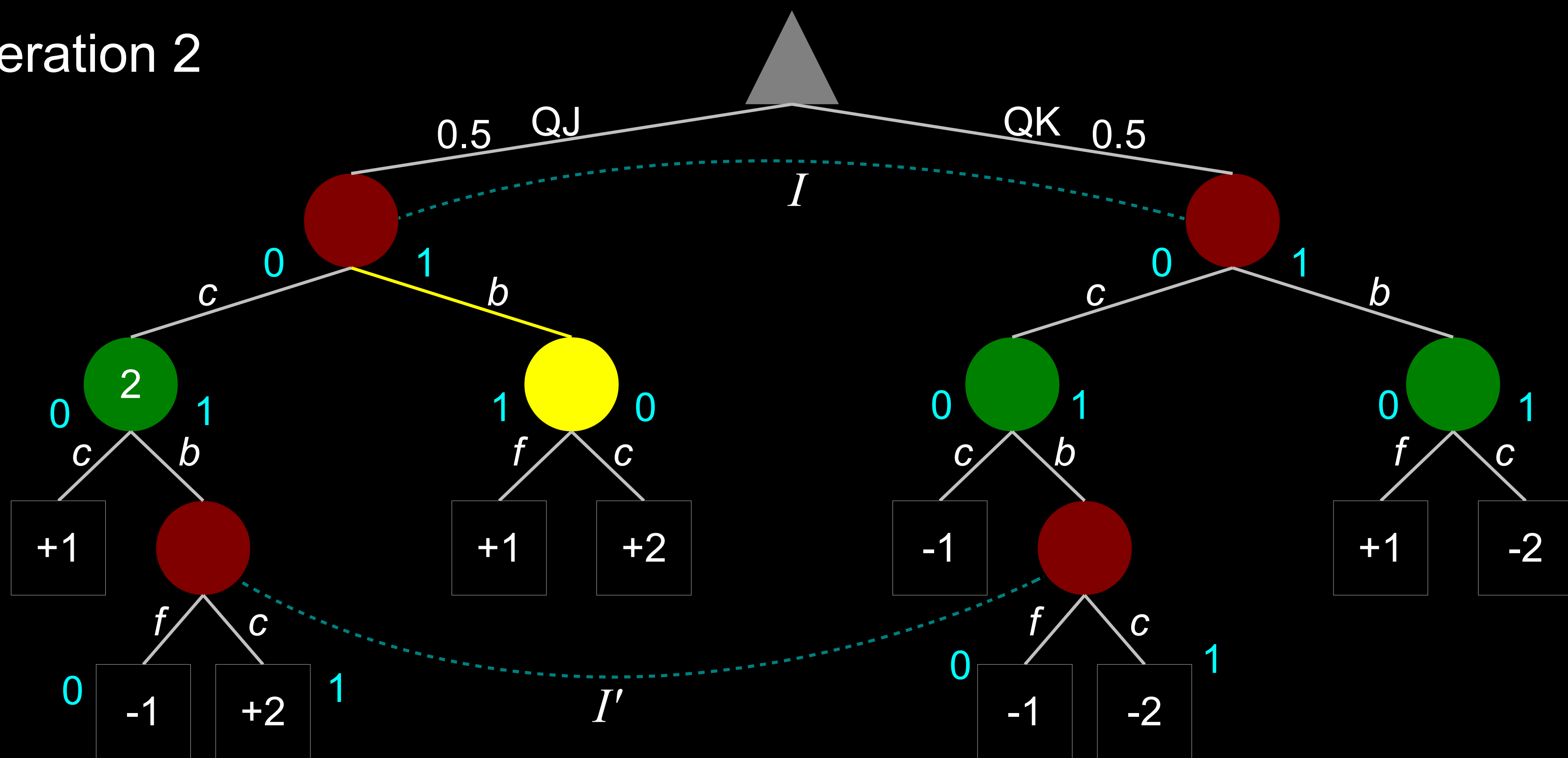


EV = 0(+1) + 1(2) = 2

“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

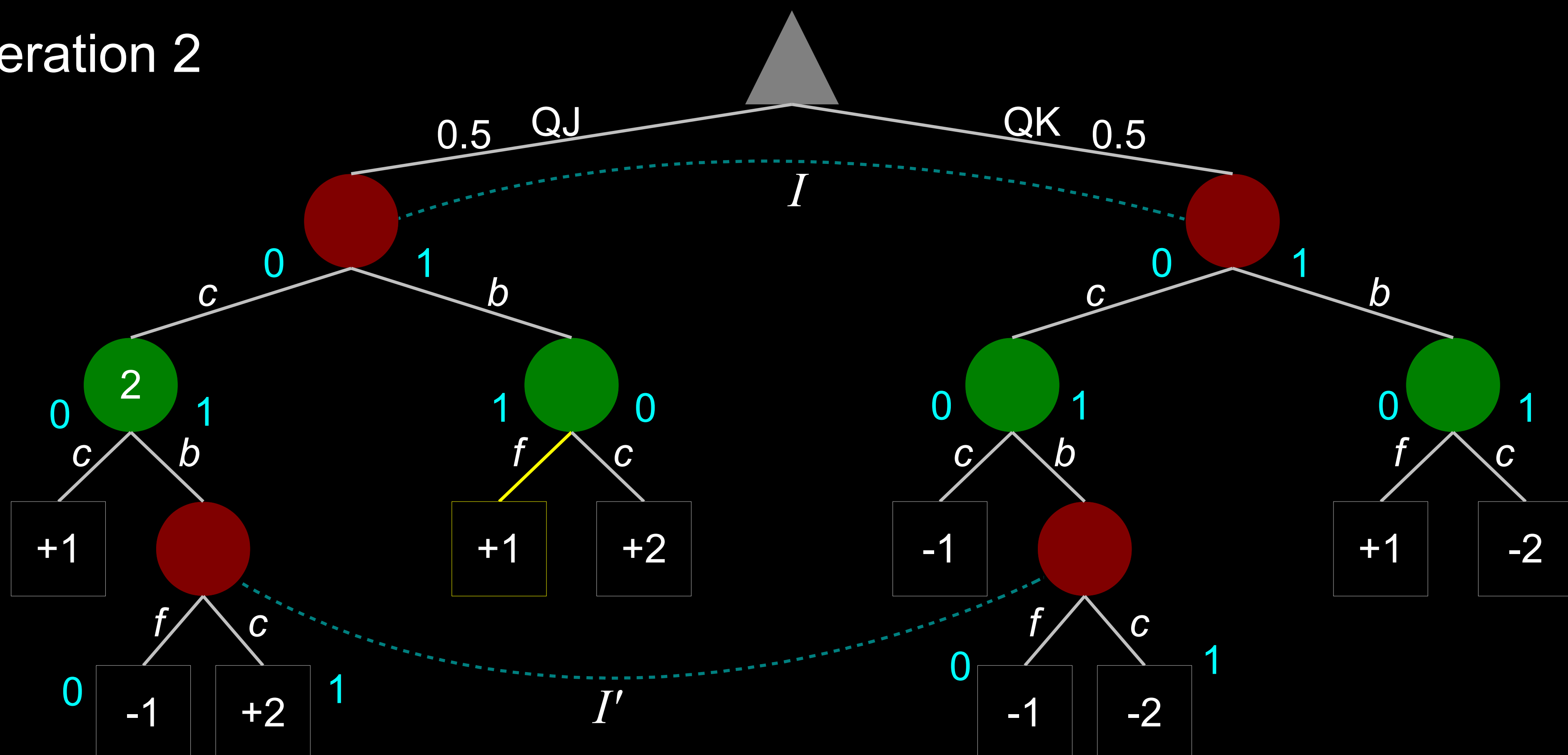
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

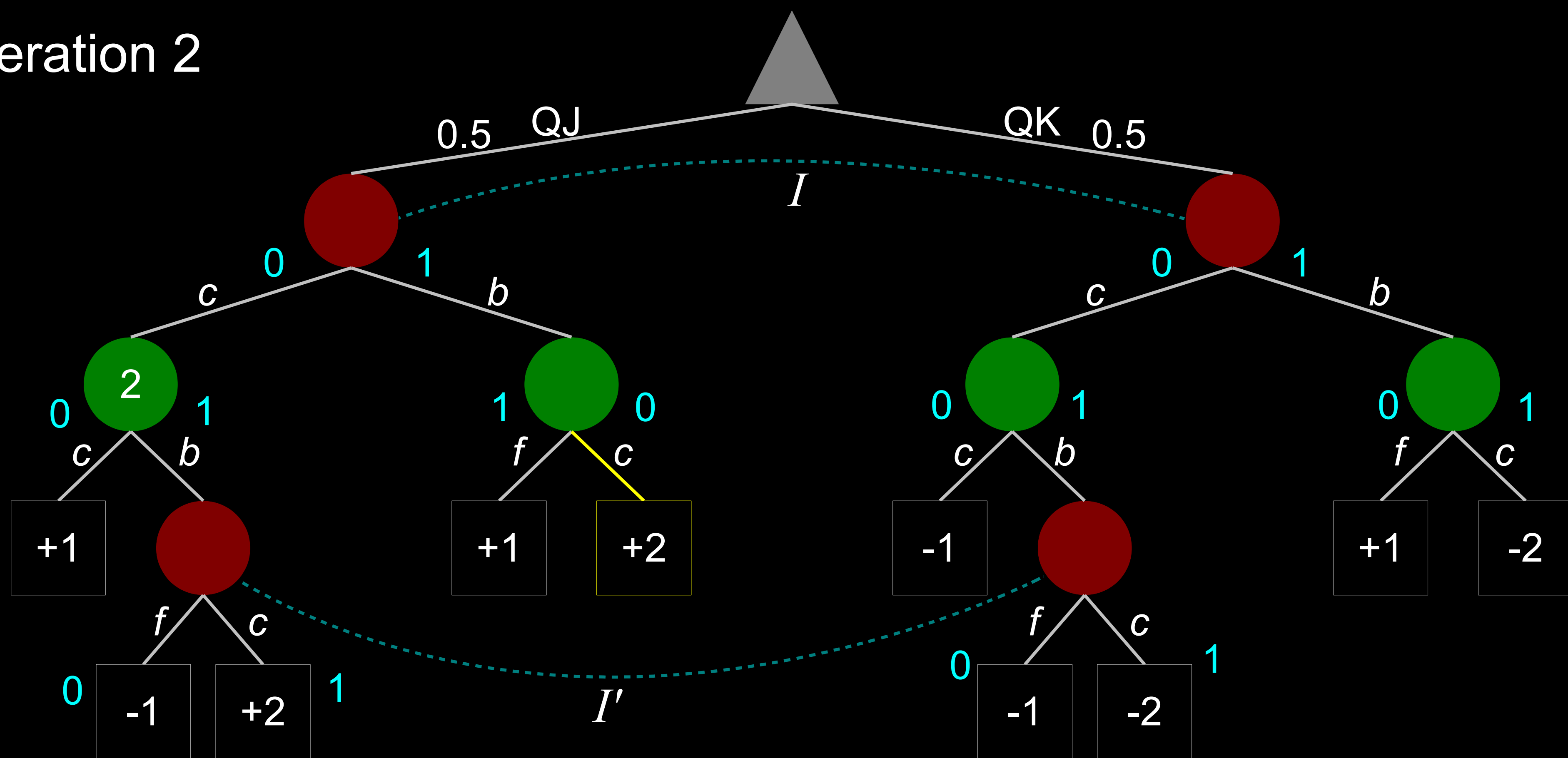
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

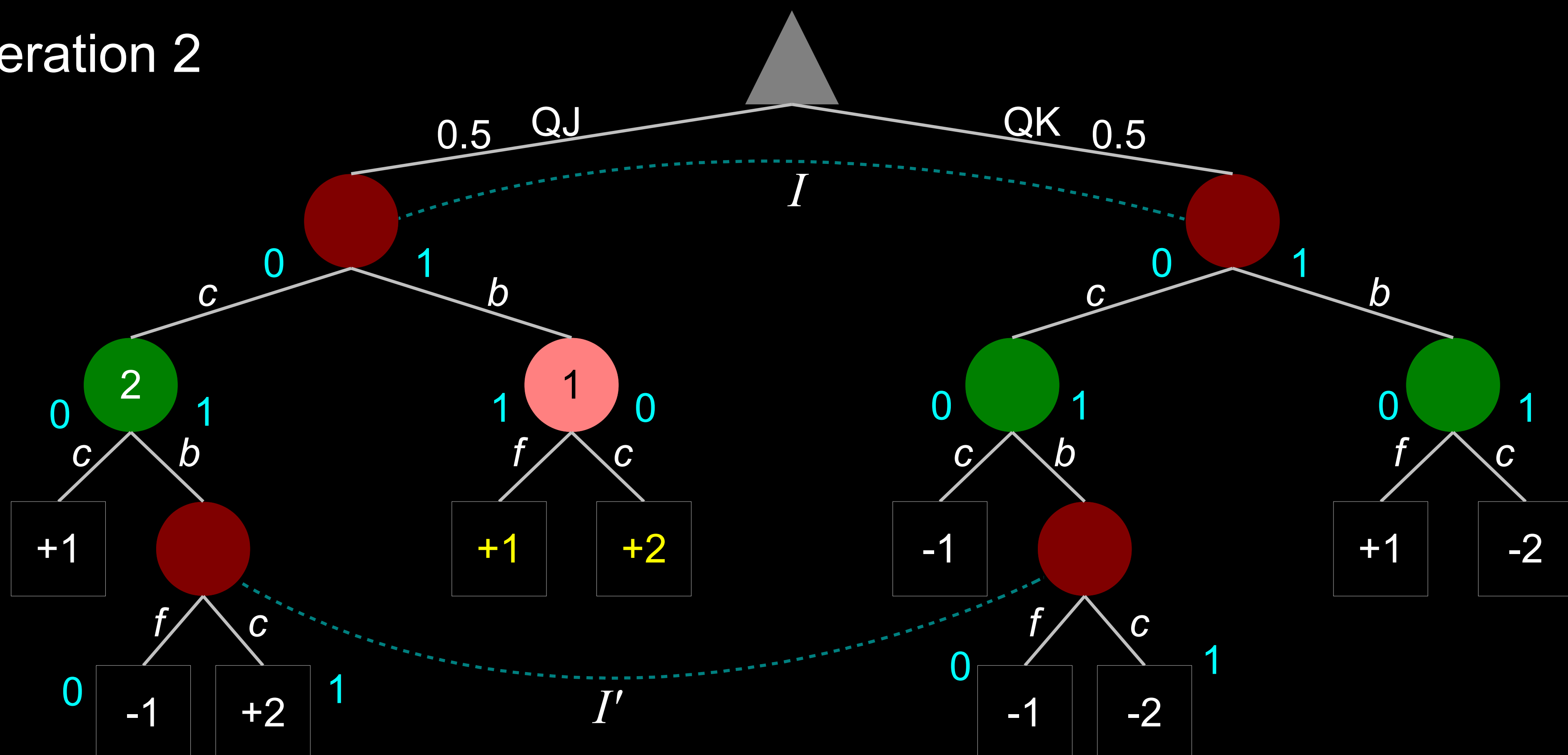
- Iteration 2



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 2

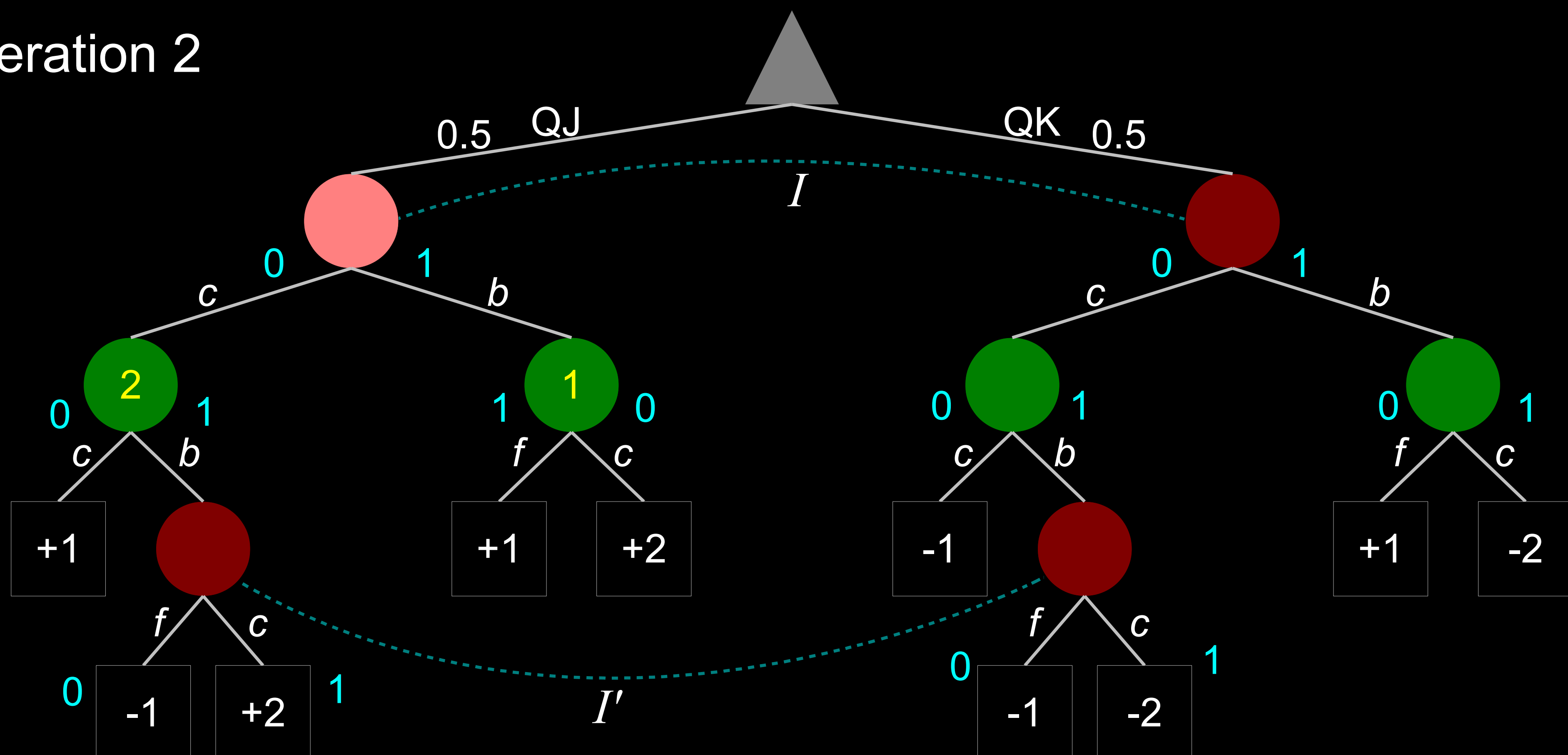


EV = 1(+1) + 0(+2) = 1

"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 2

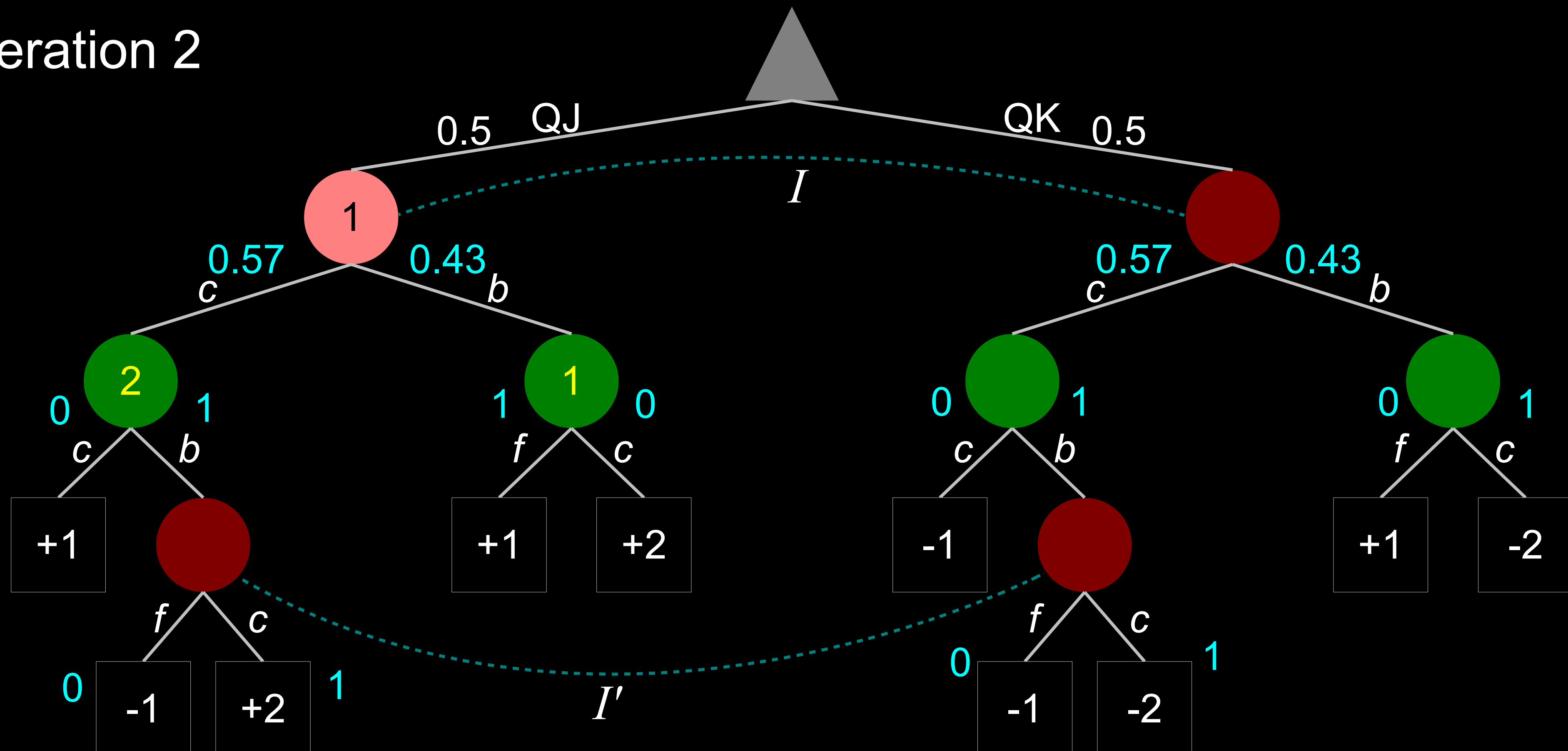


EV = 0(2) + 1(1) = 1

"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 2

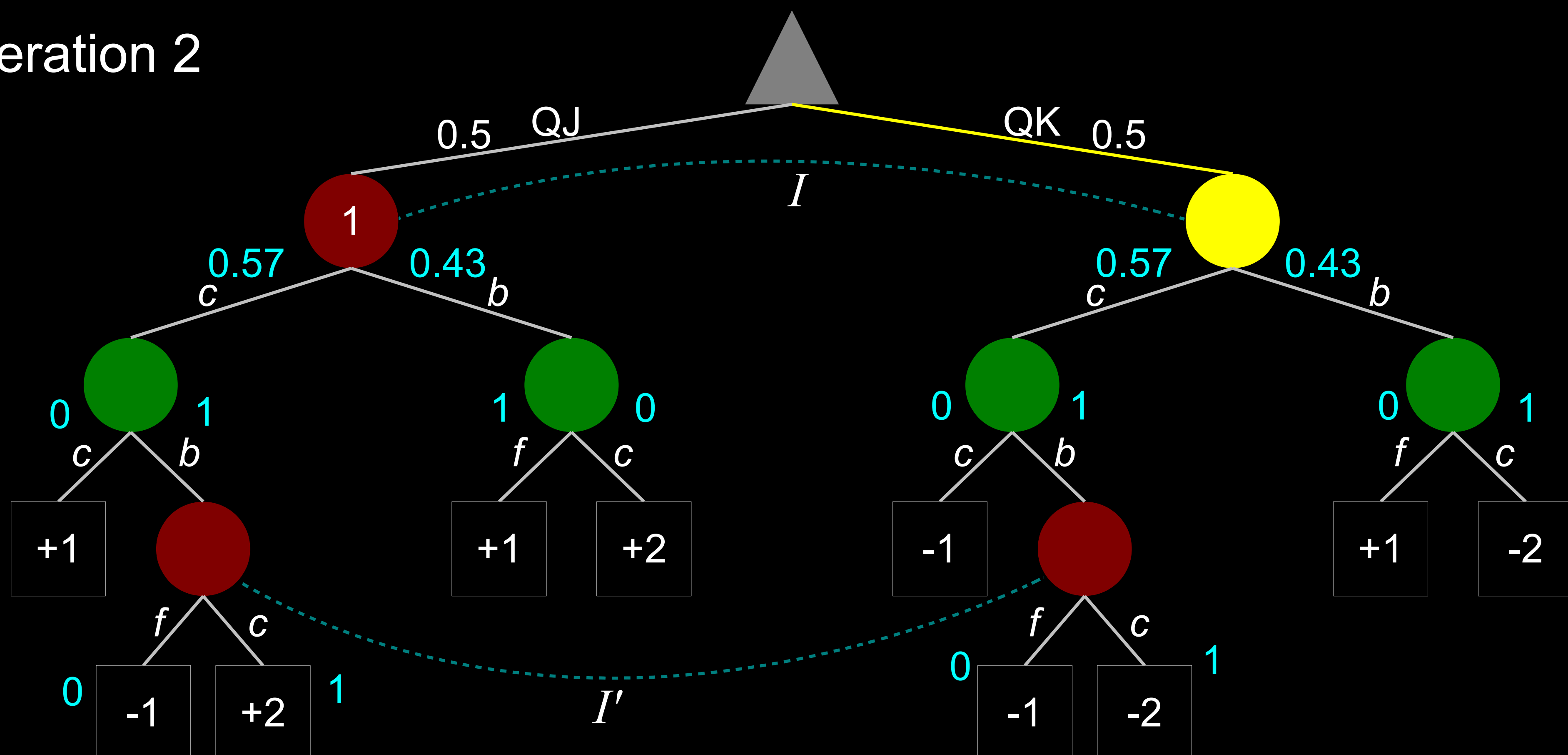


$\sigma_1(c)$: 2 > 1 ⬆ $\sigma_1(b)$: 1 = 1 no change

“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

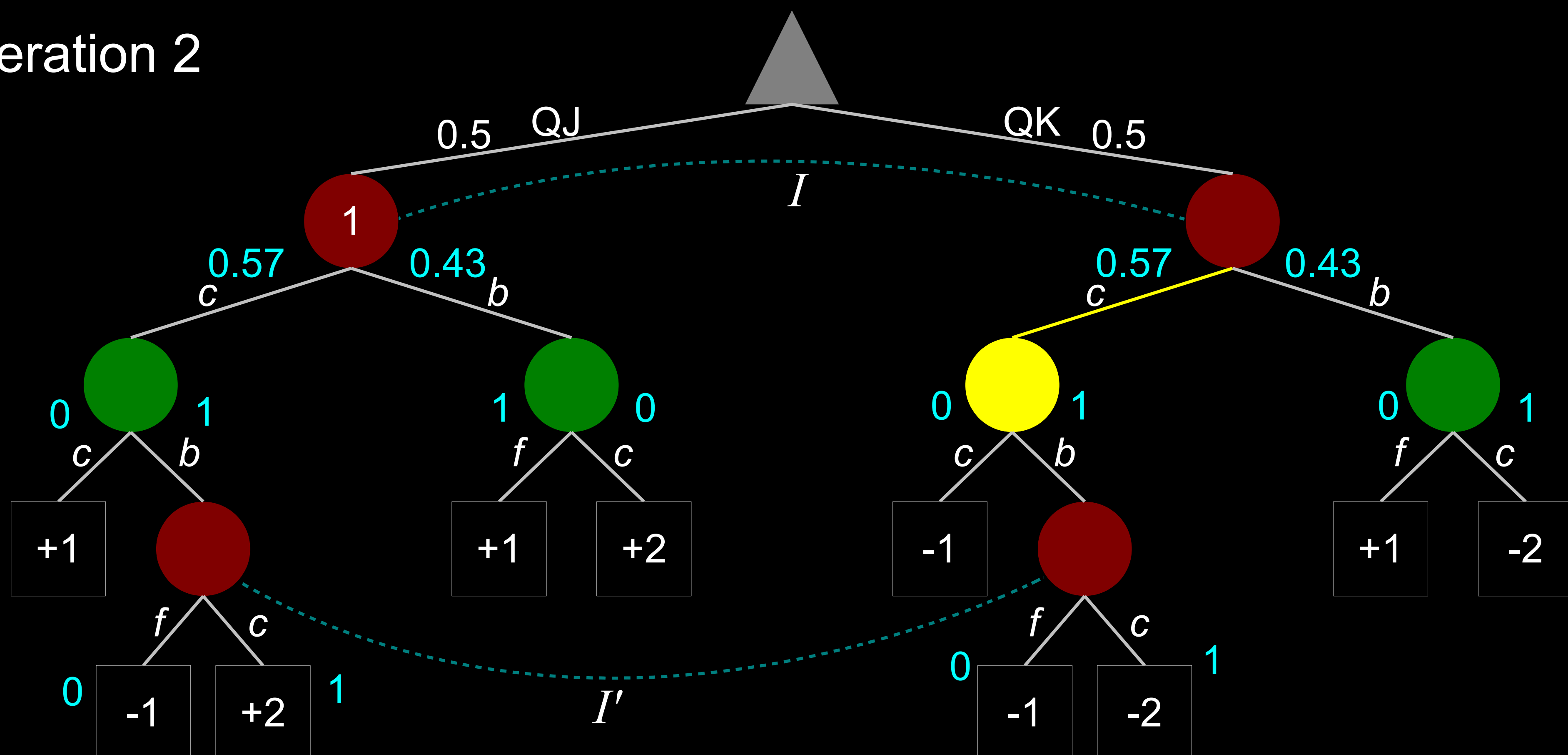
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

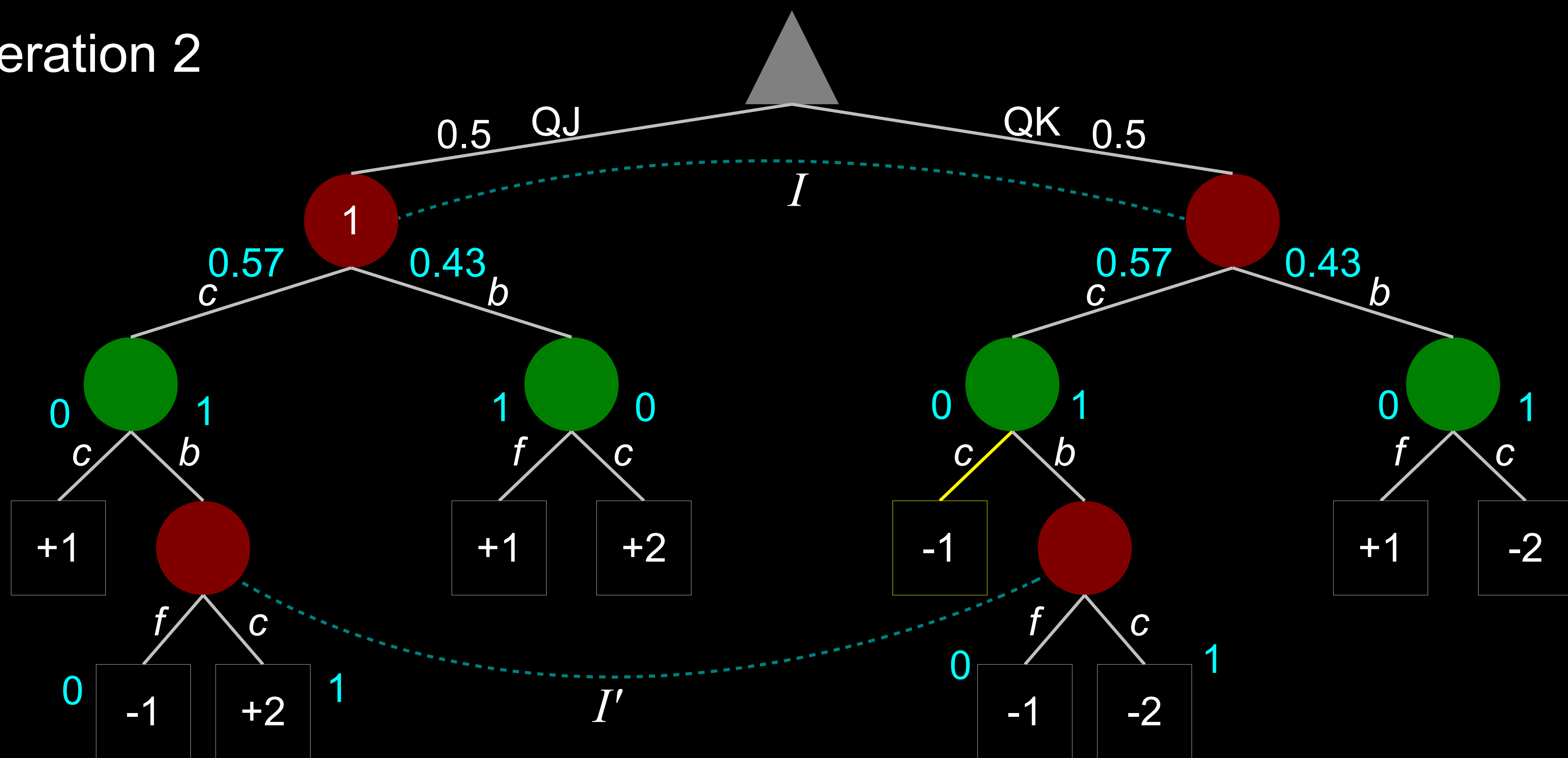
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

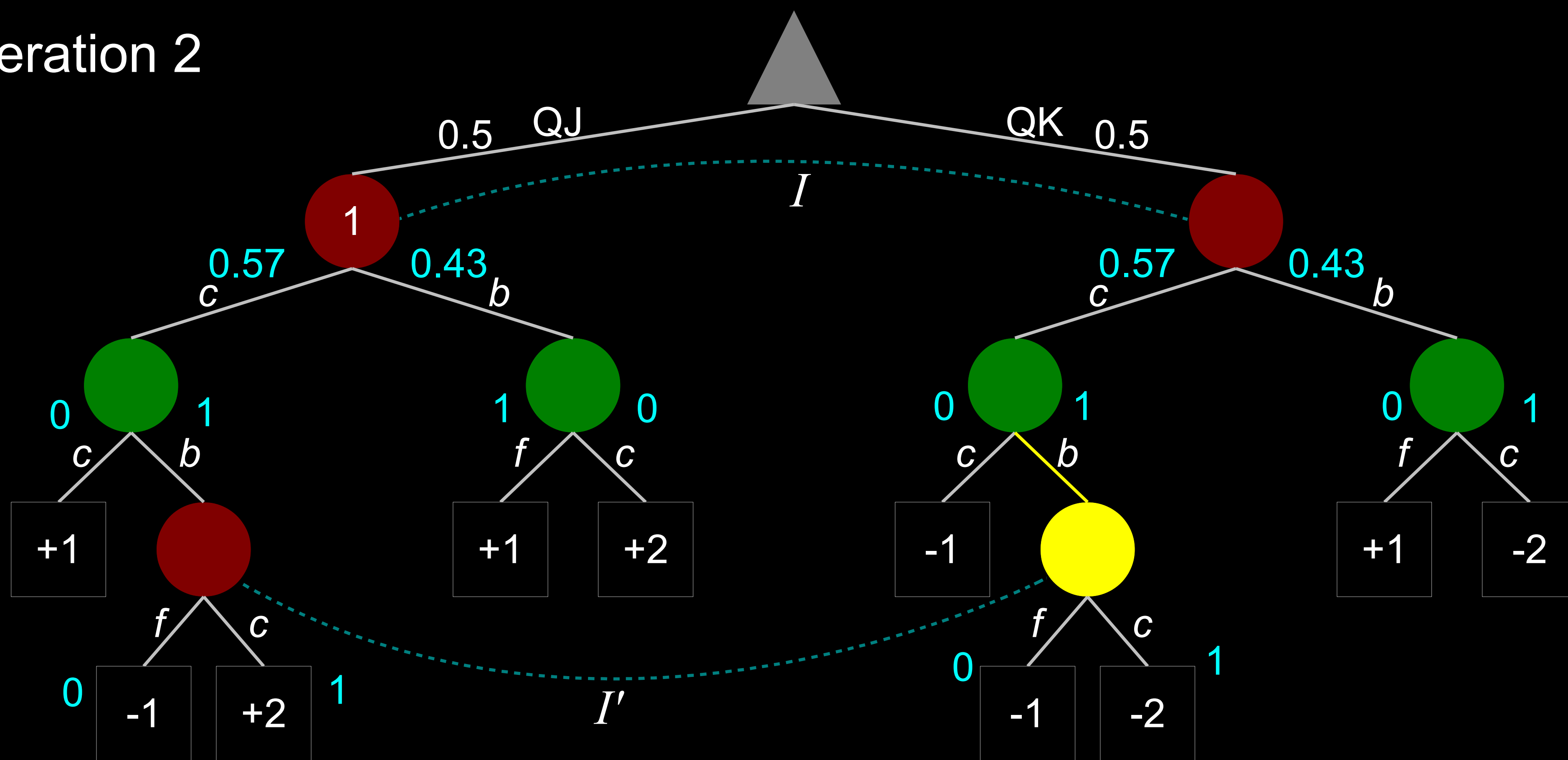
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

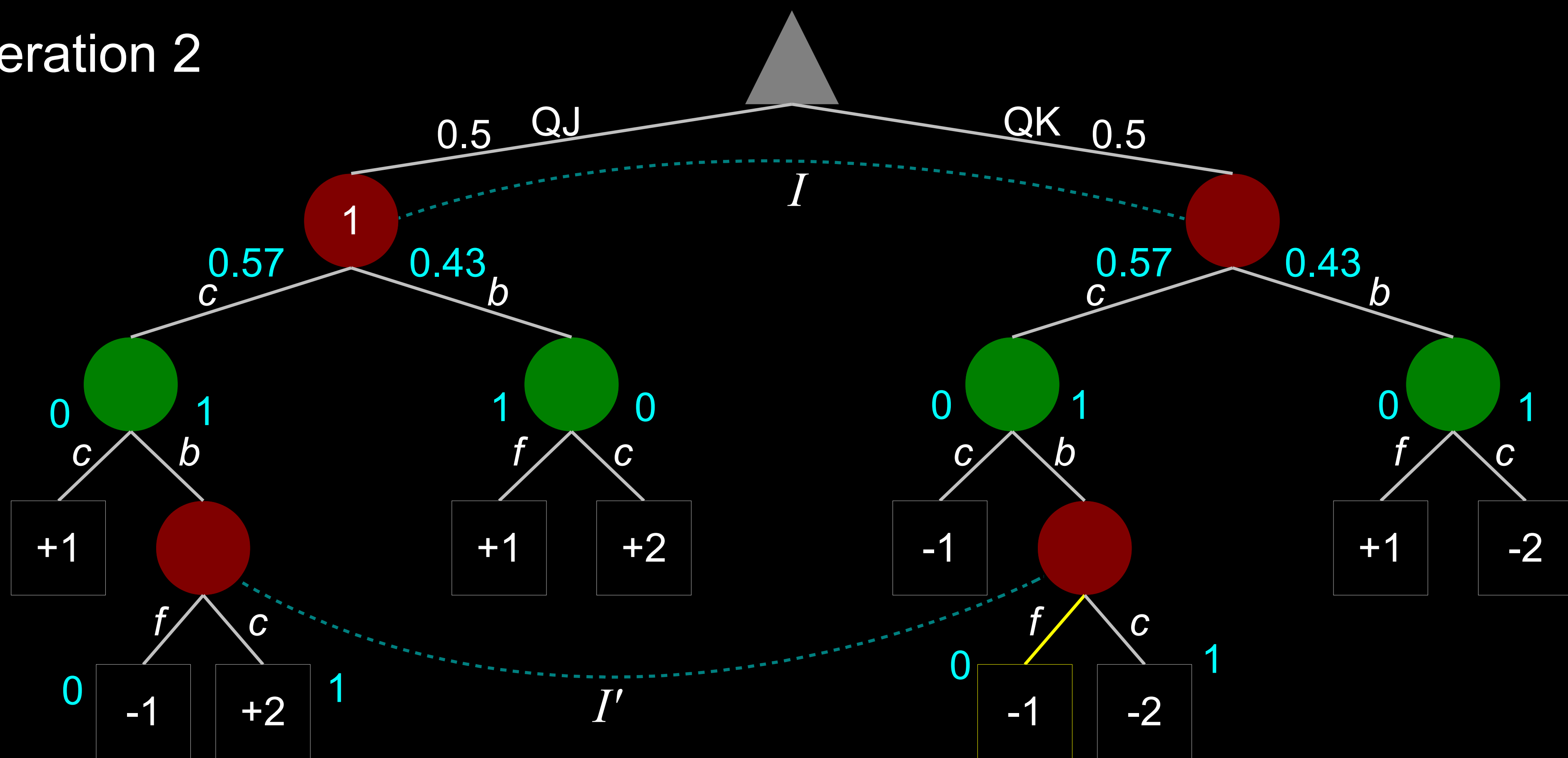
- Iteration 2



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

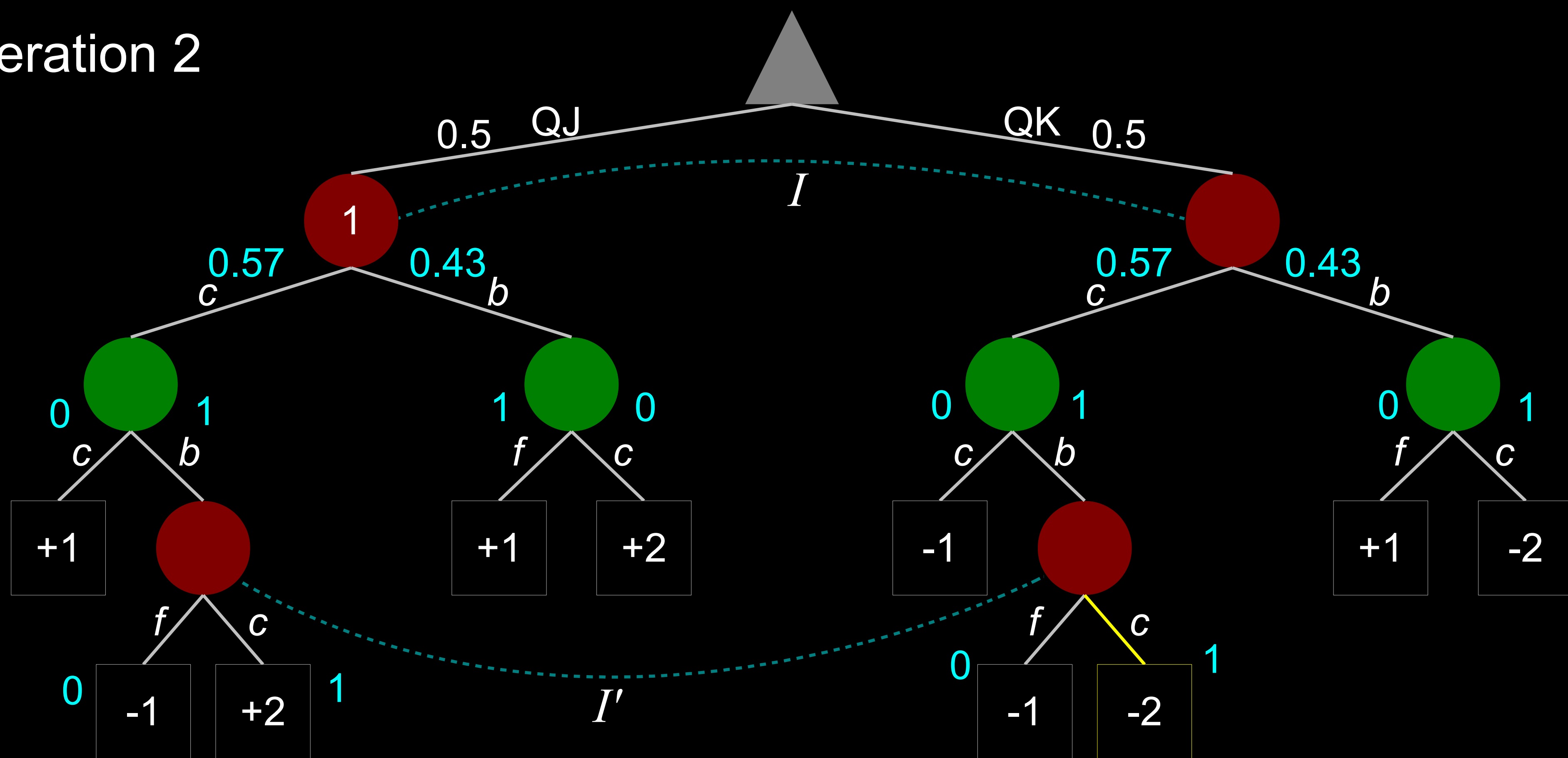
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

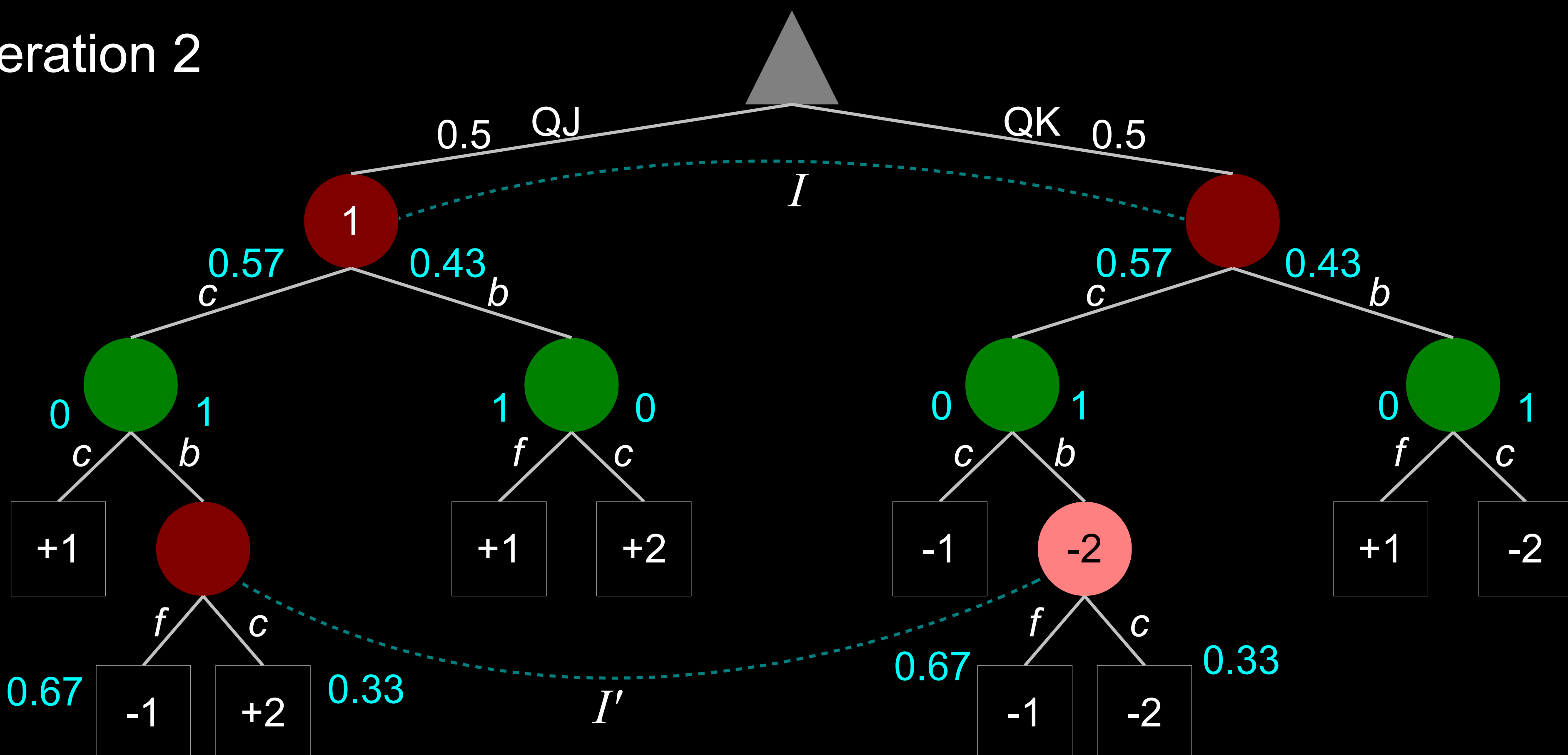
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

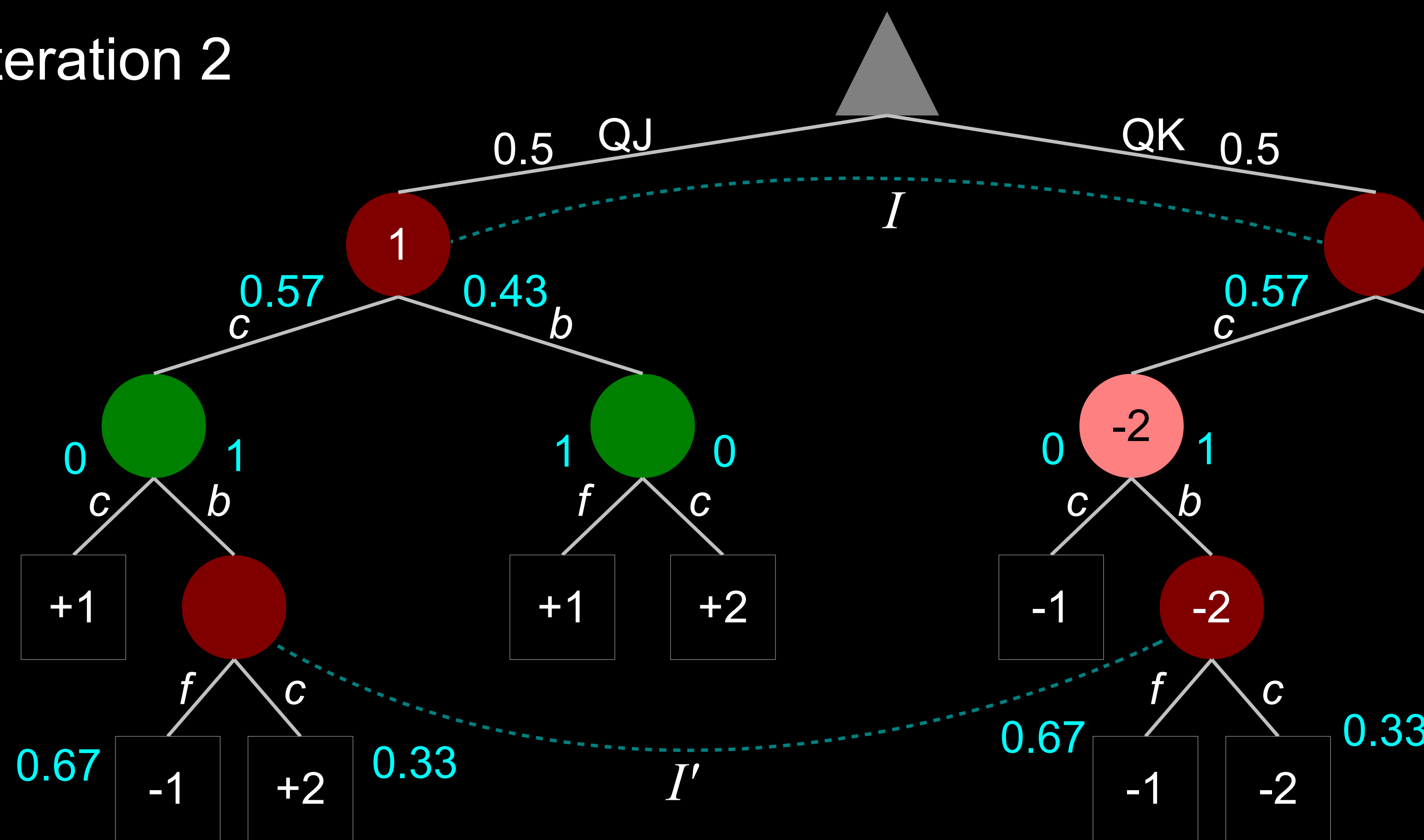
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

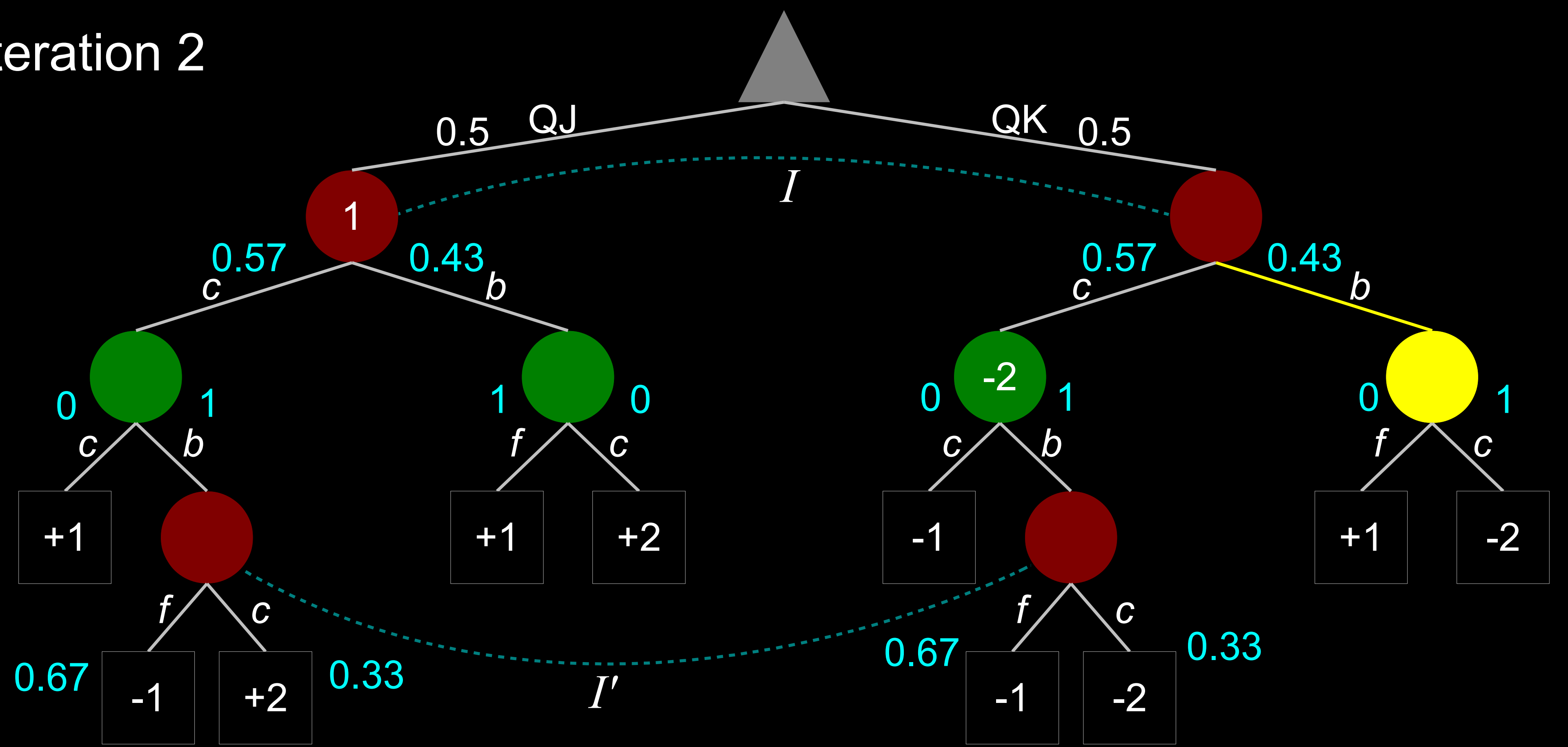
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

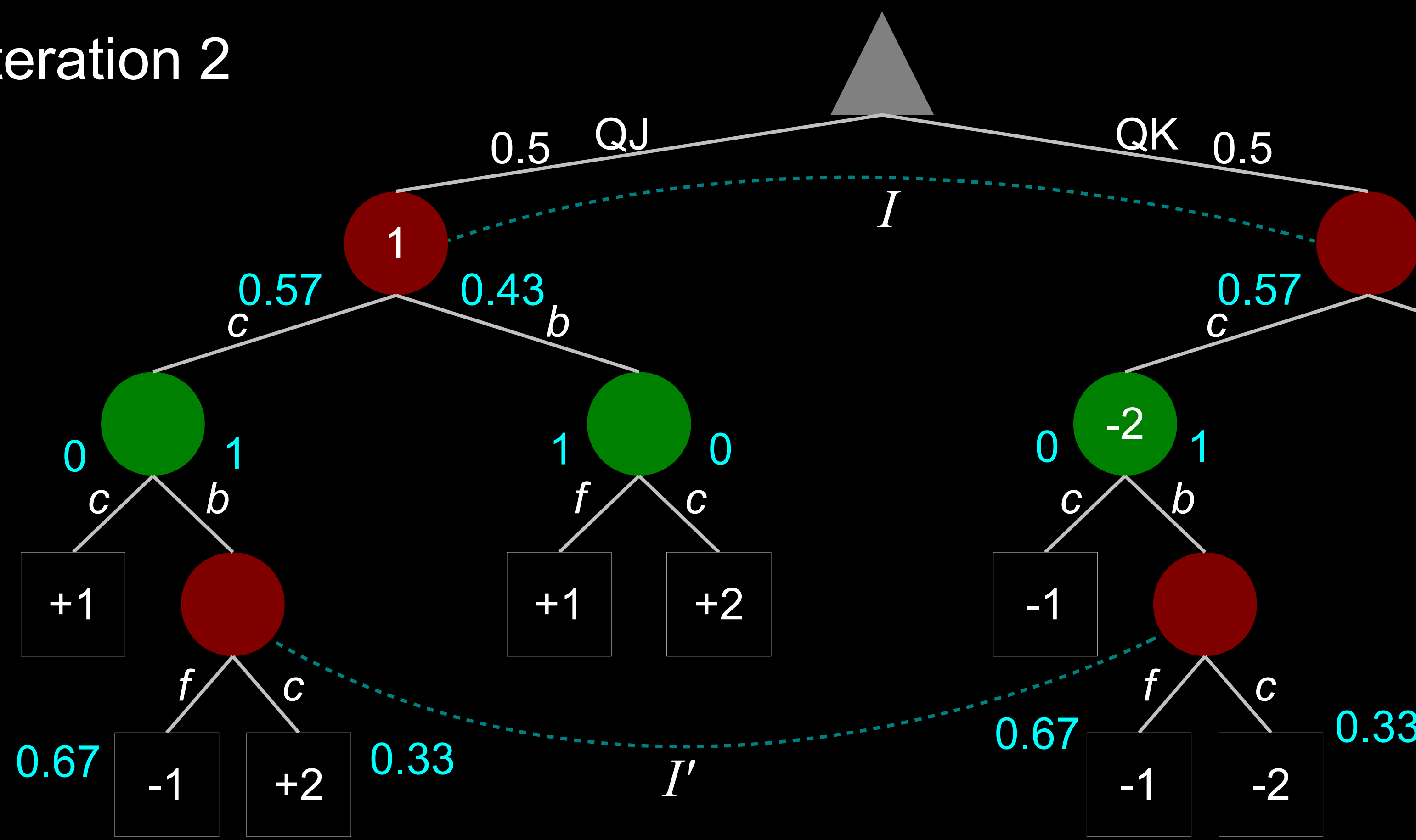
- Iteration 2



"Vanilla" CFR Walk-through

Each iteration, we perform a depth-first tree walk

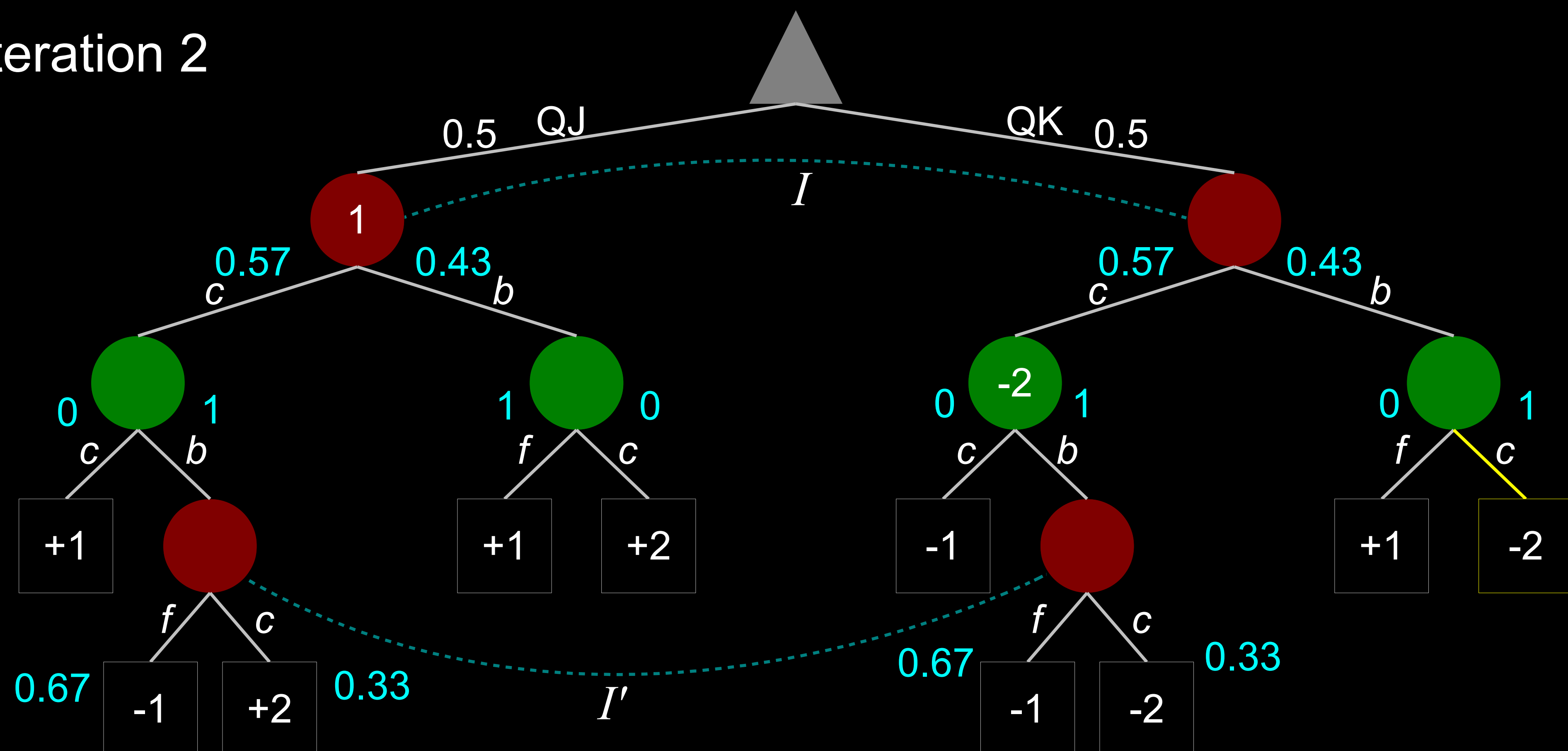
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

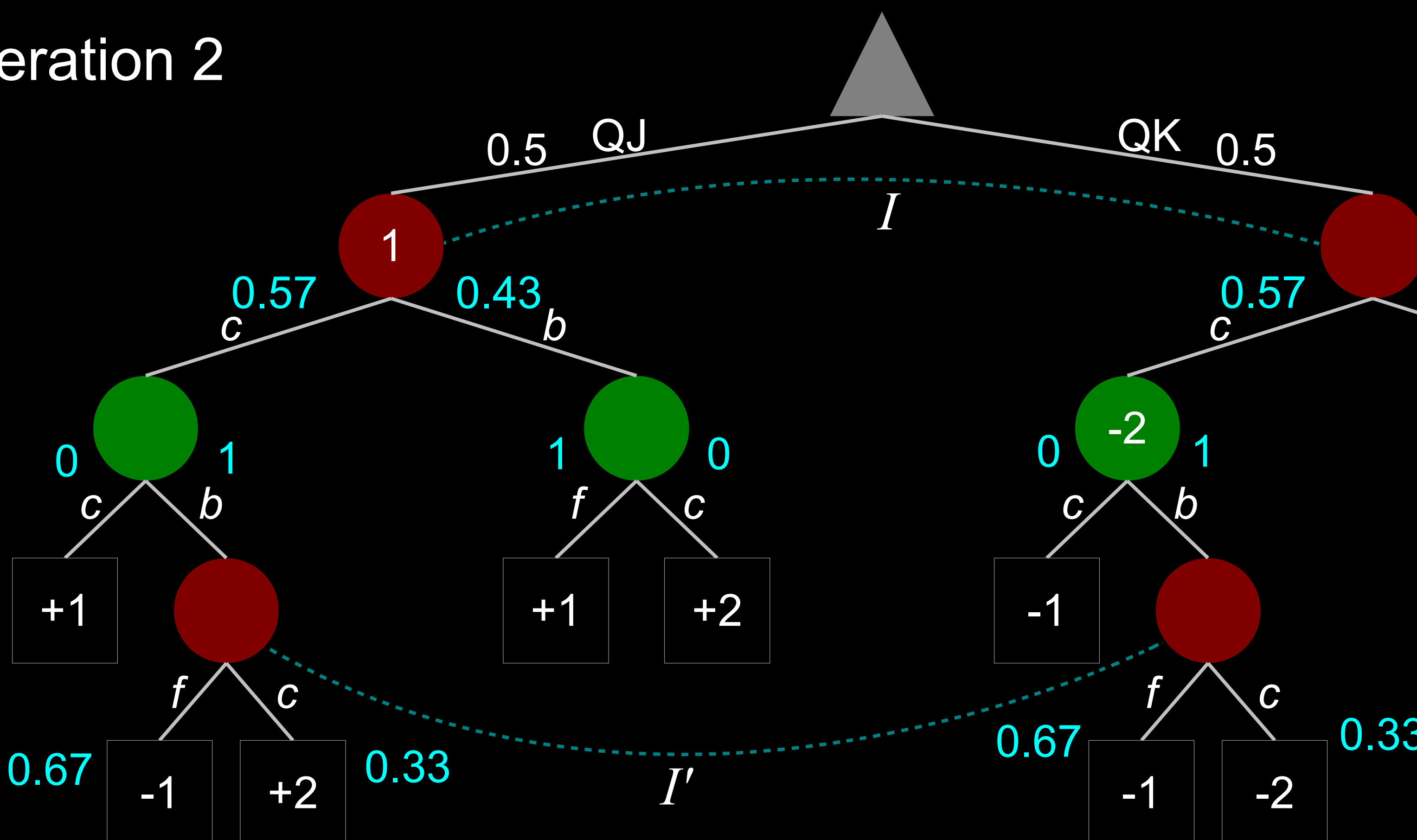
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

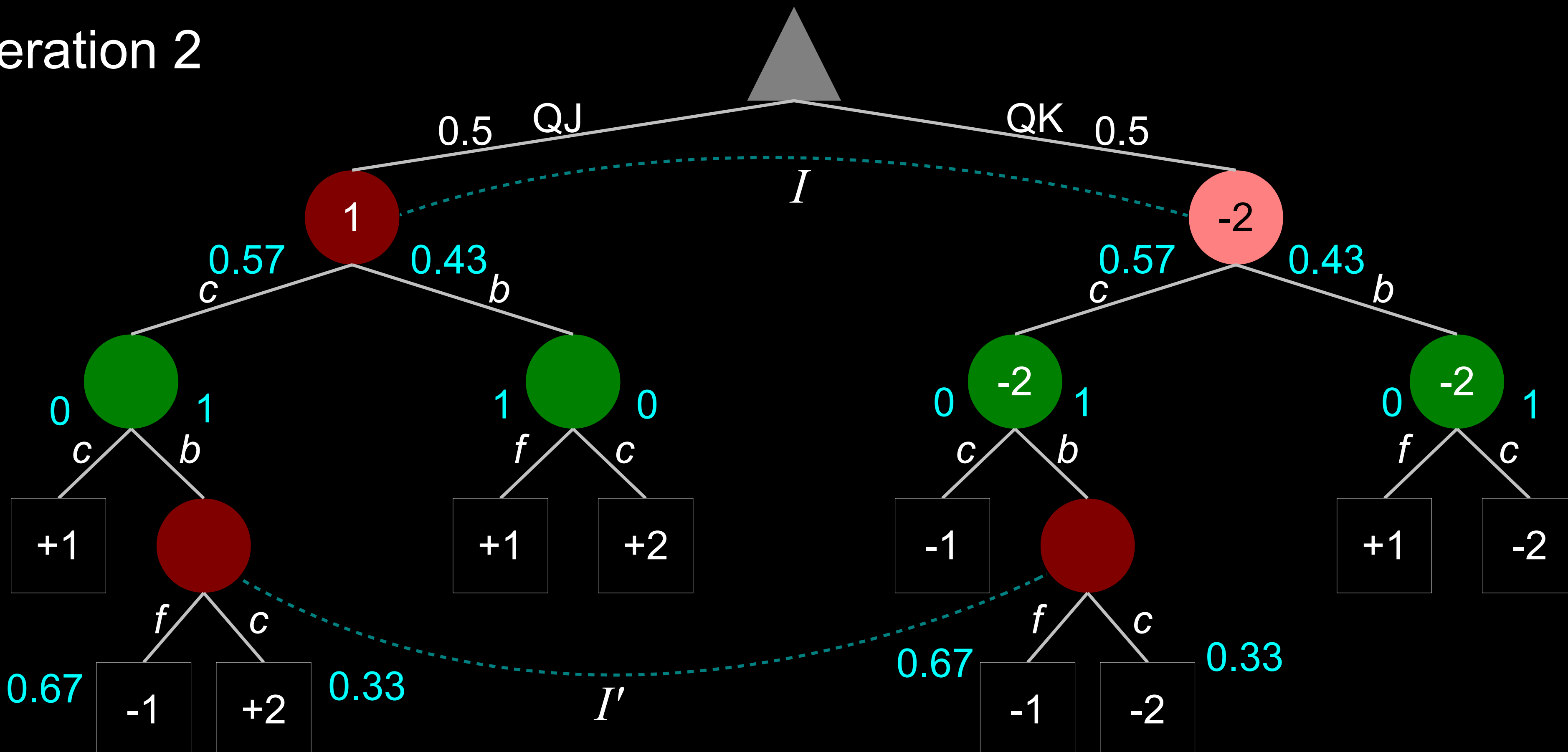
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

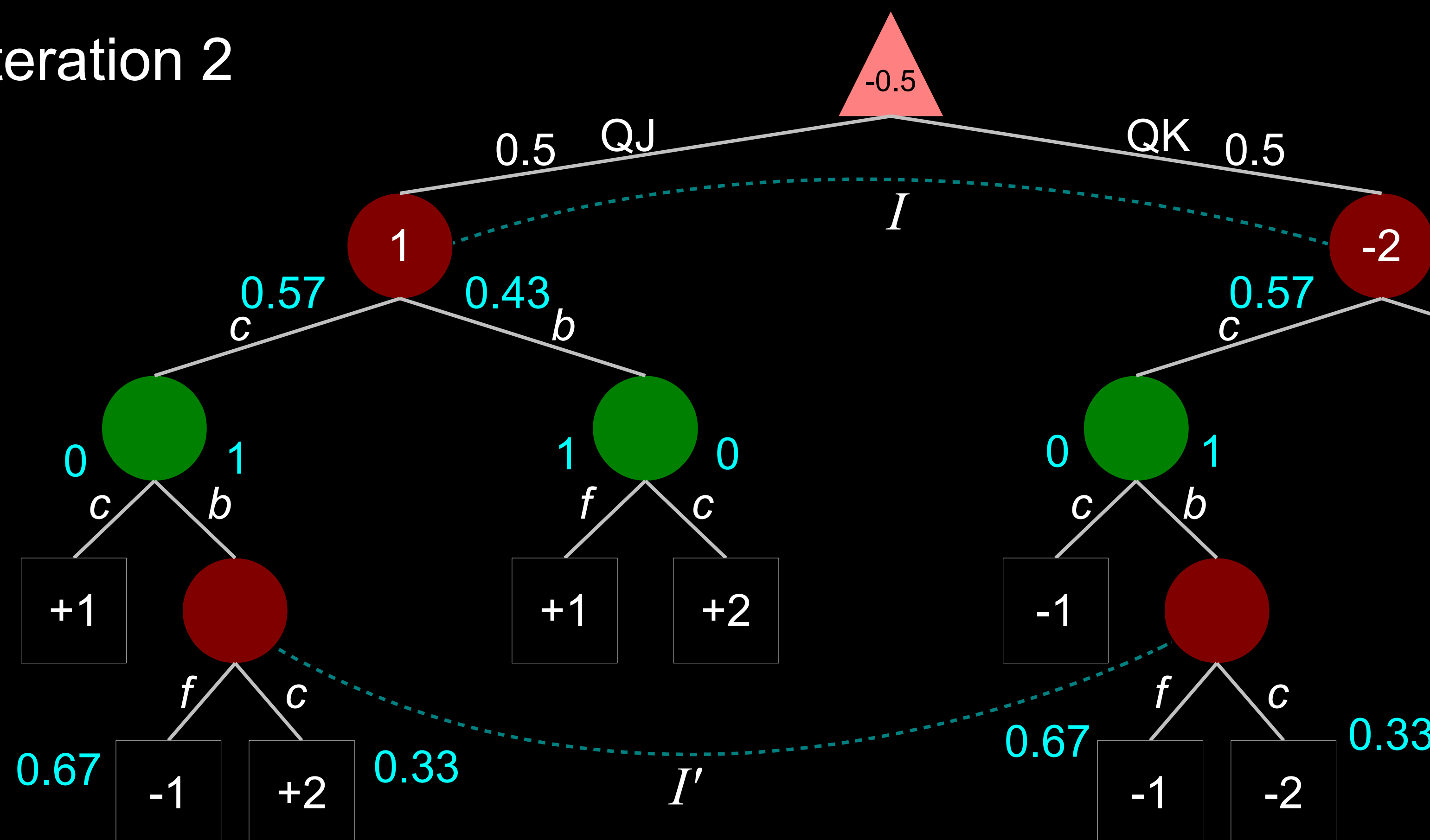
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

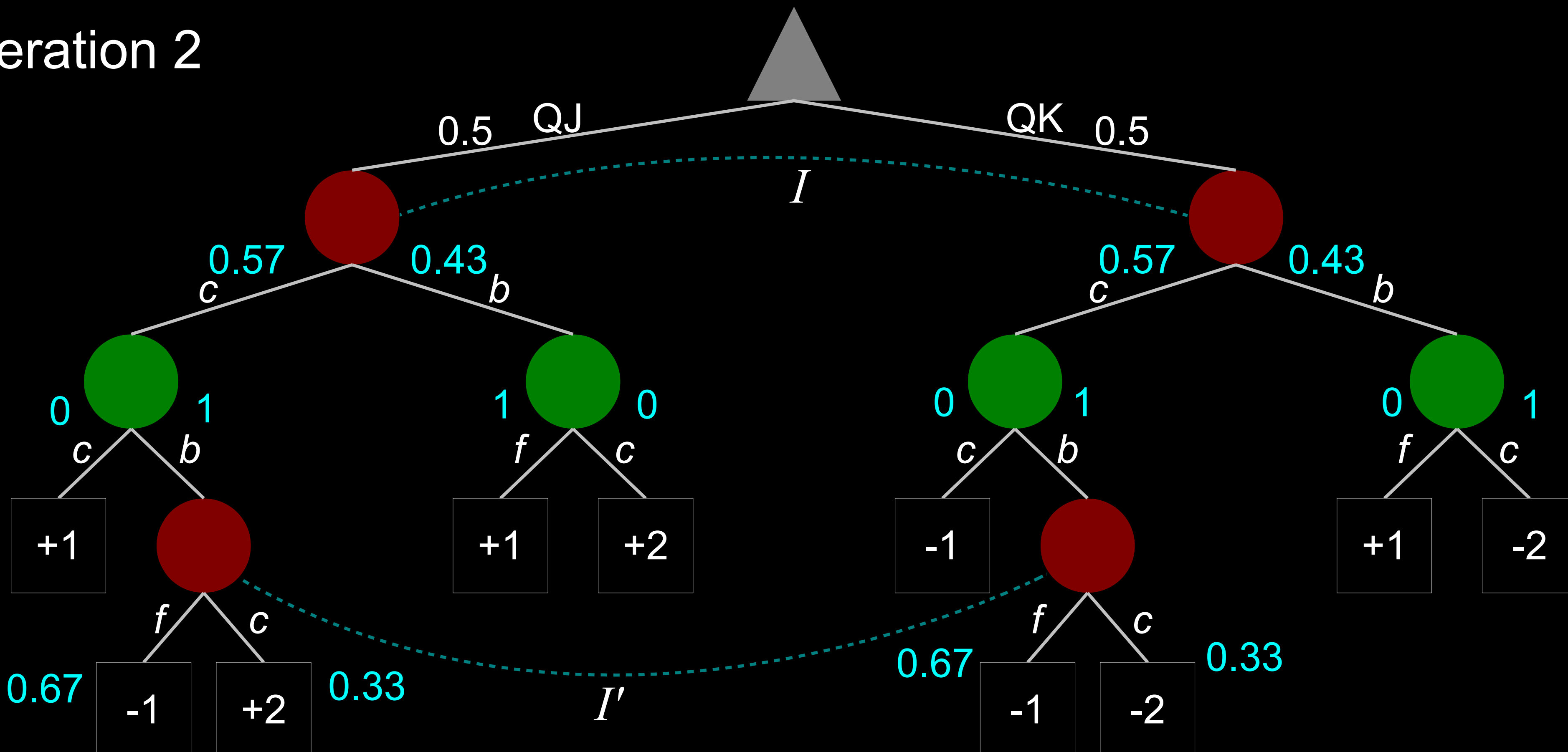
- Iteration 2



“Vanilla” CFR Walk-through

Each iteration, we perform a depth-first tree walk

- Iteration 2



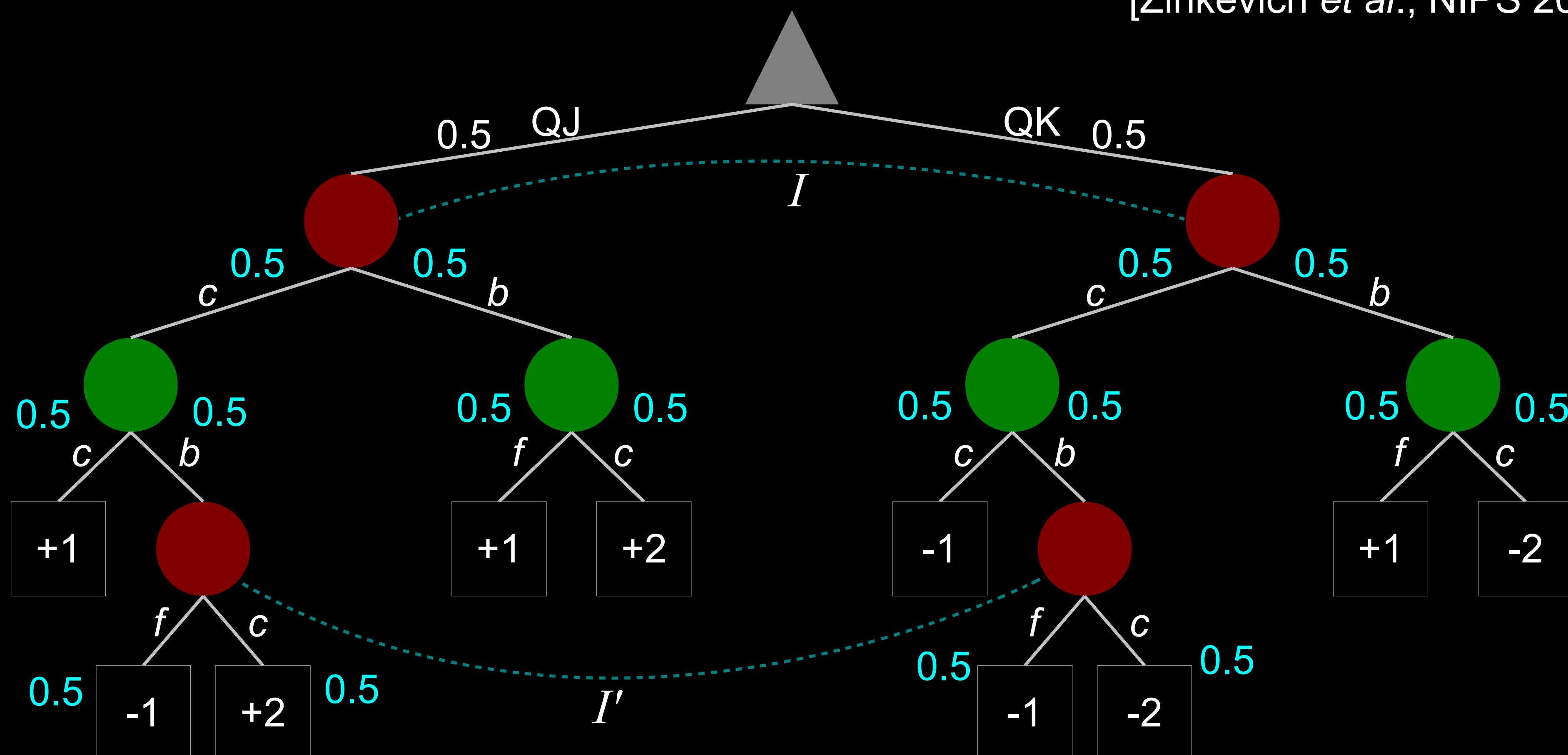
Vanilla CFR Summary

- Vanilla CFR traverses the entire game tree each iteration.
- As the number of iterations approaches infinity, the average of the strategy profiles produced converges to a Nash equilibrium.
- However, each iteration can be very expensive in really large games (think billions of information sets).

First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

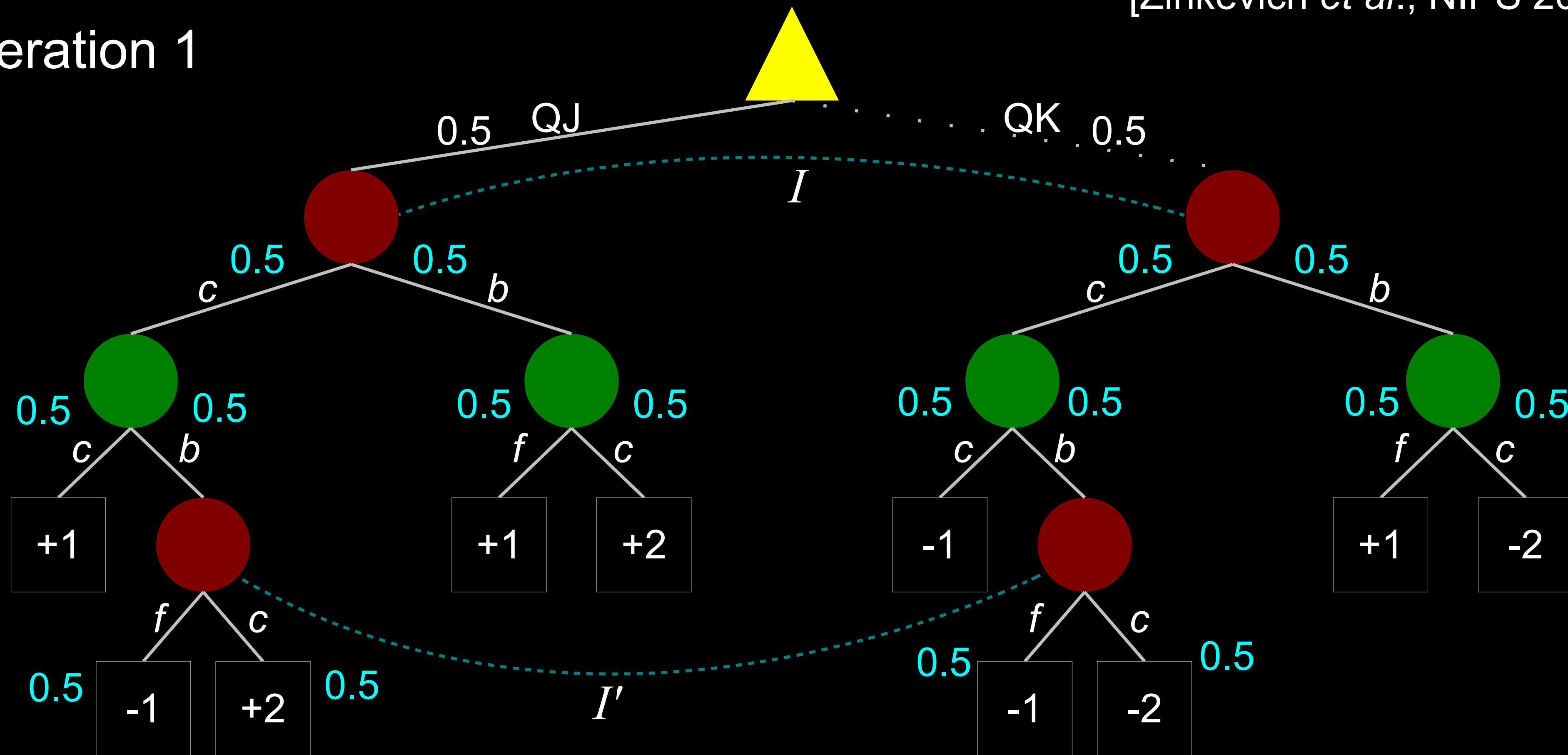


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1

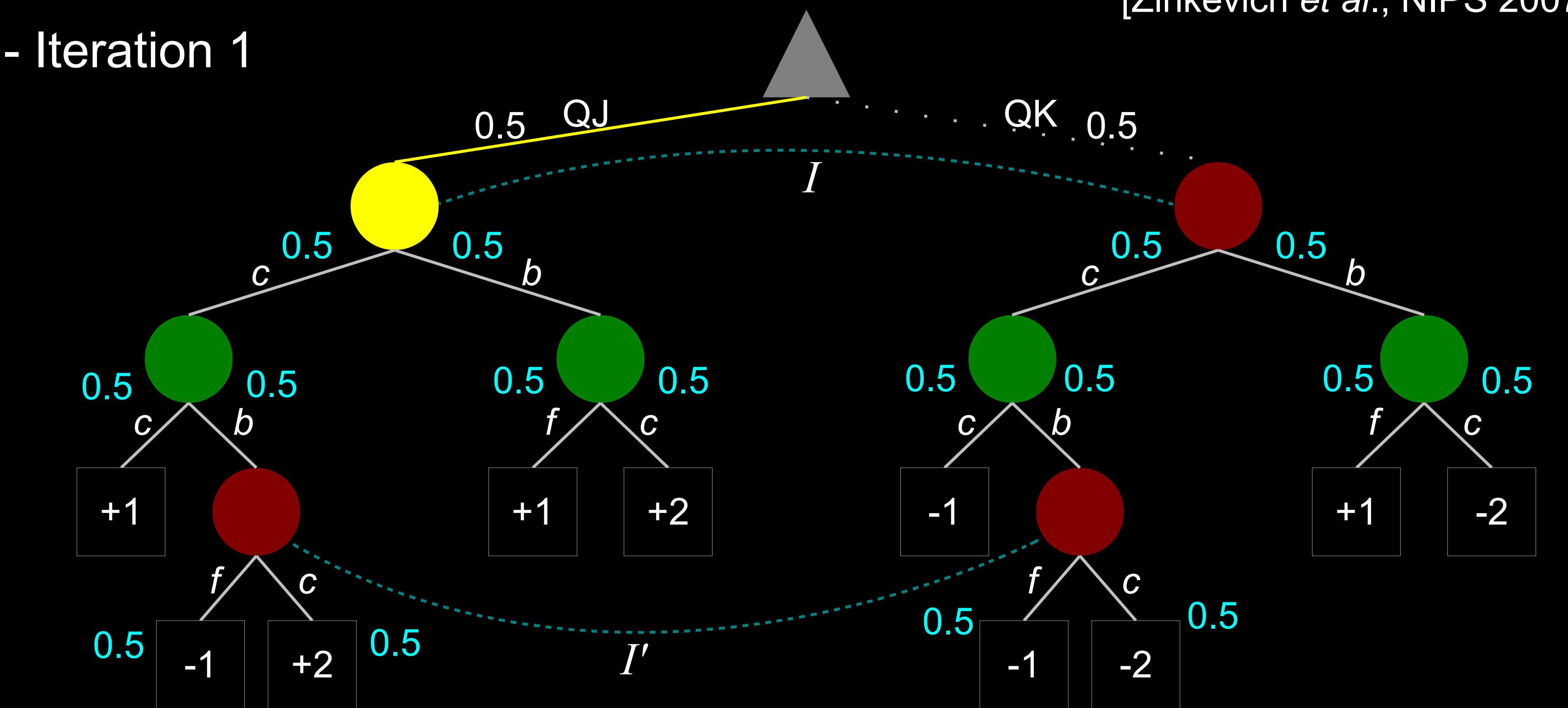


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1

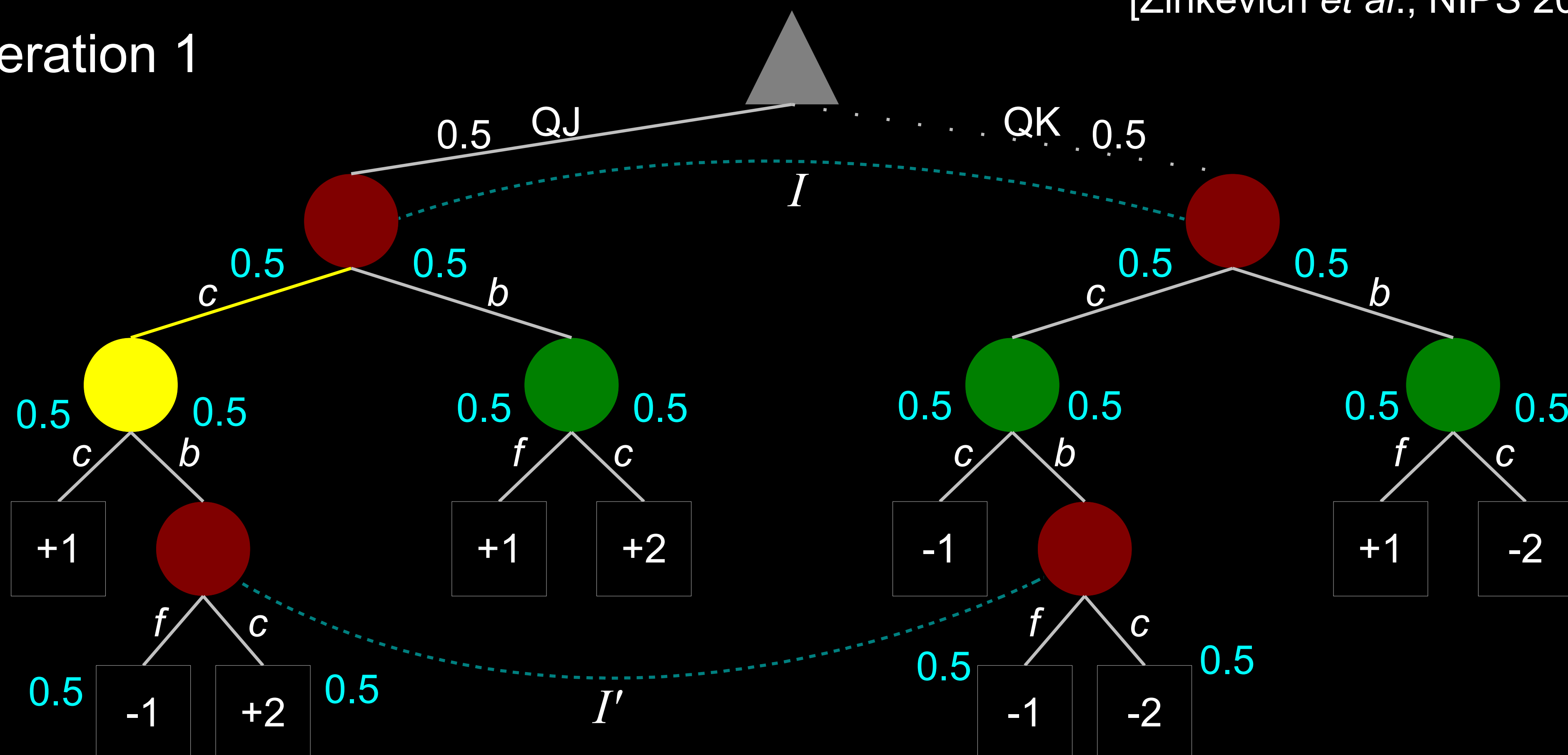


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1

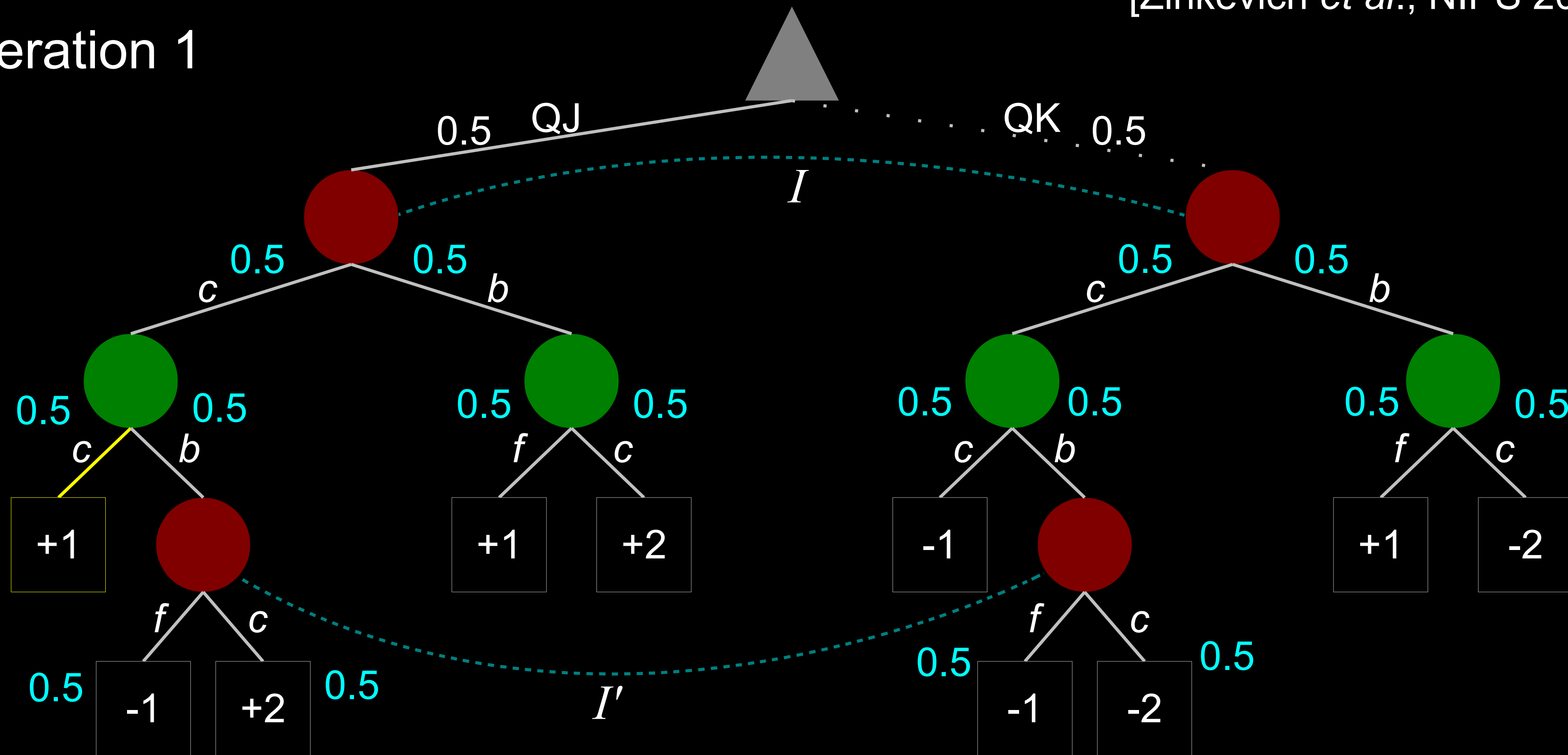


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1

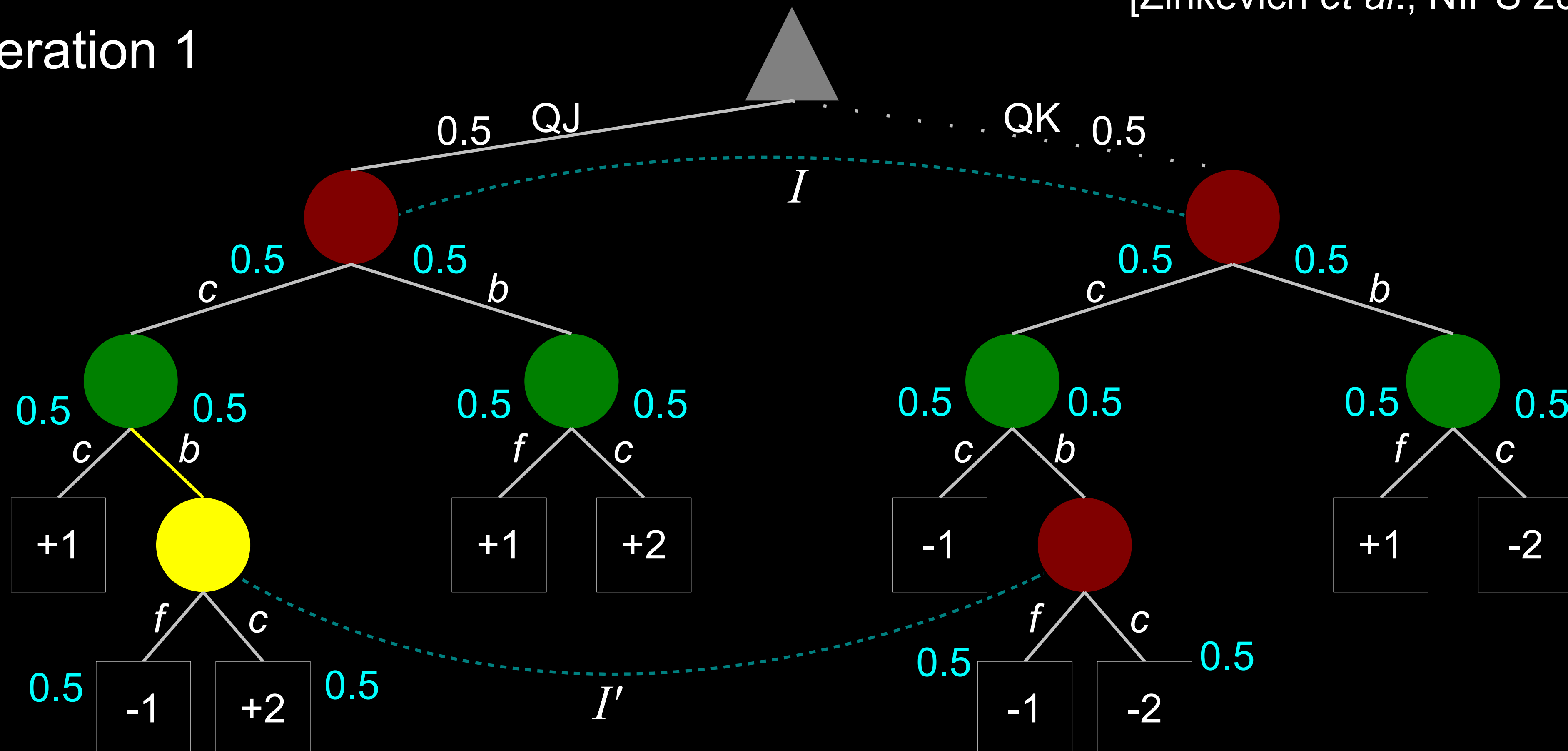


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1

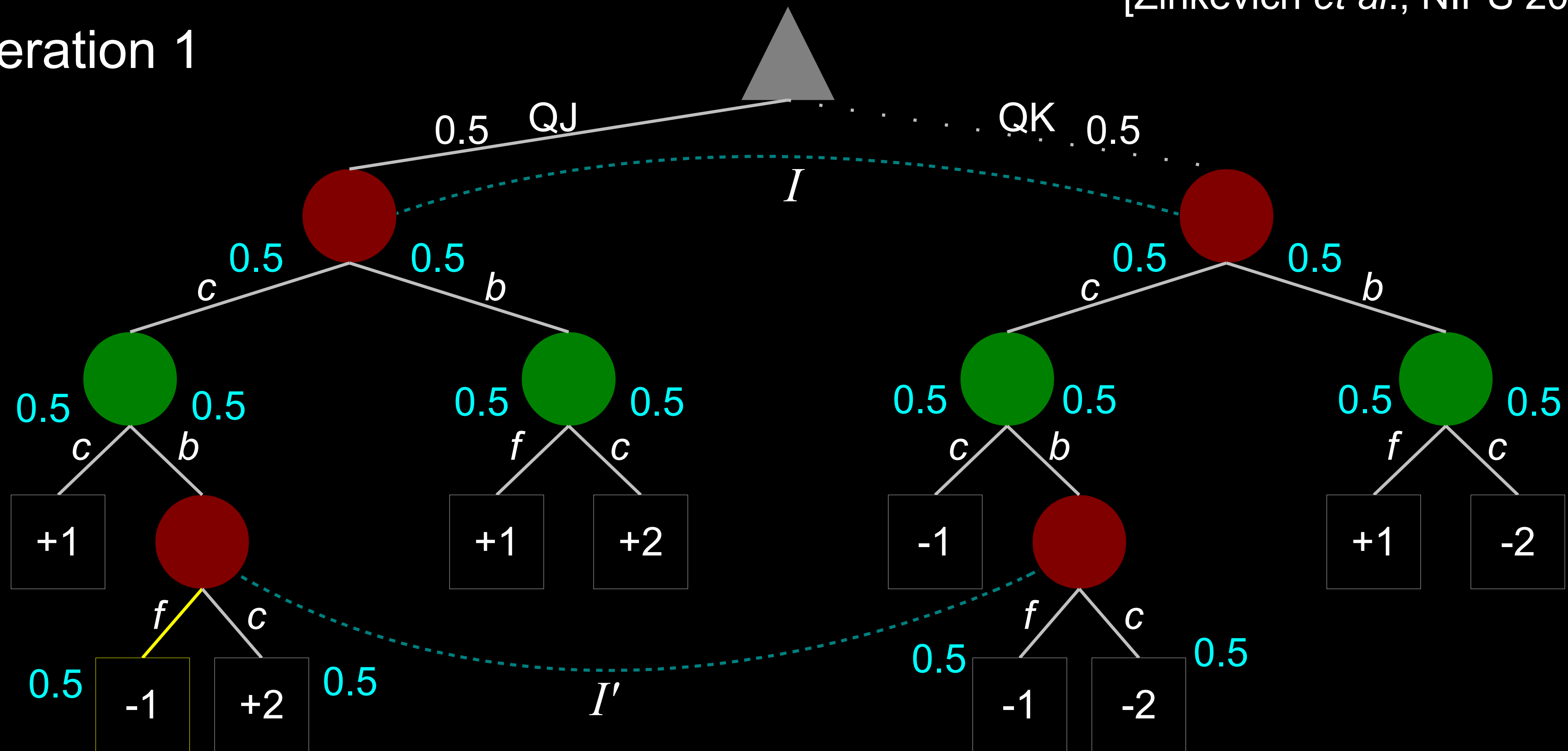


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1

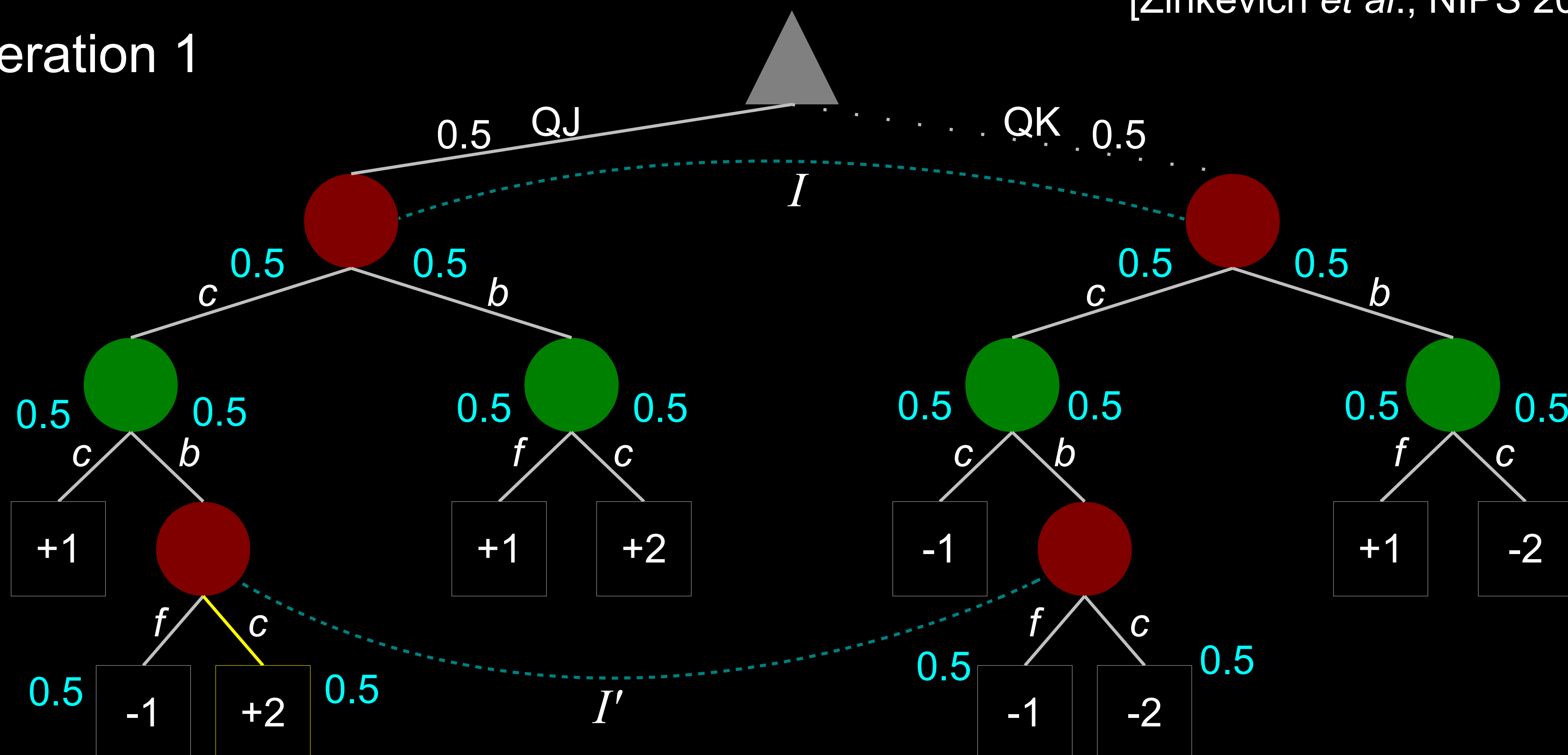


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1

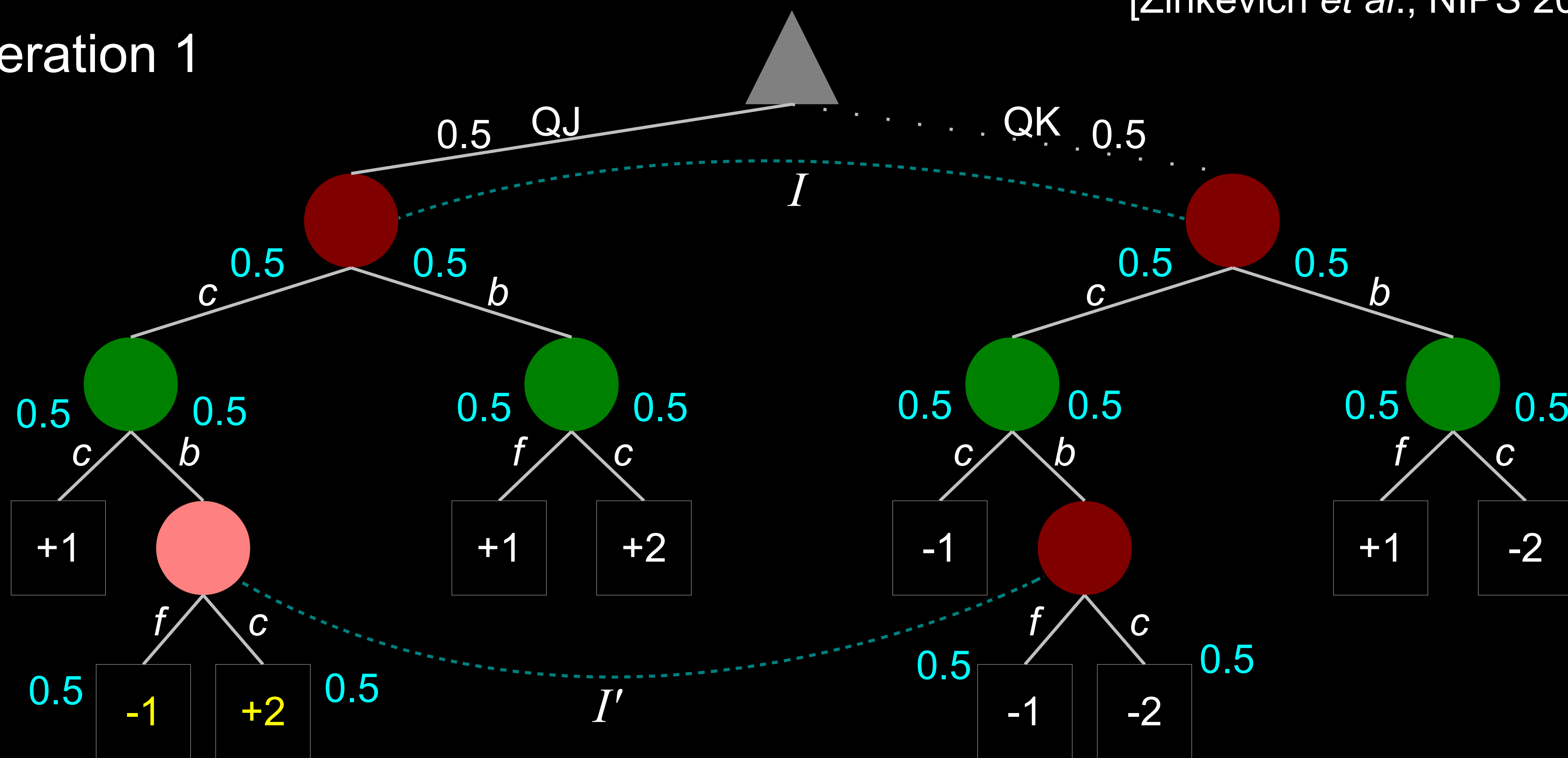


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1



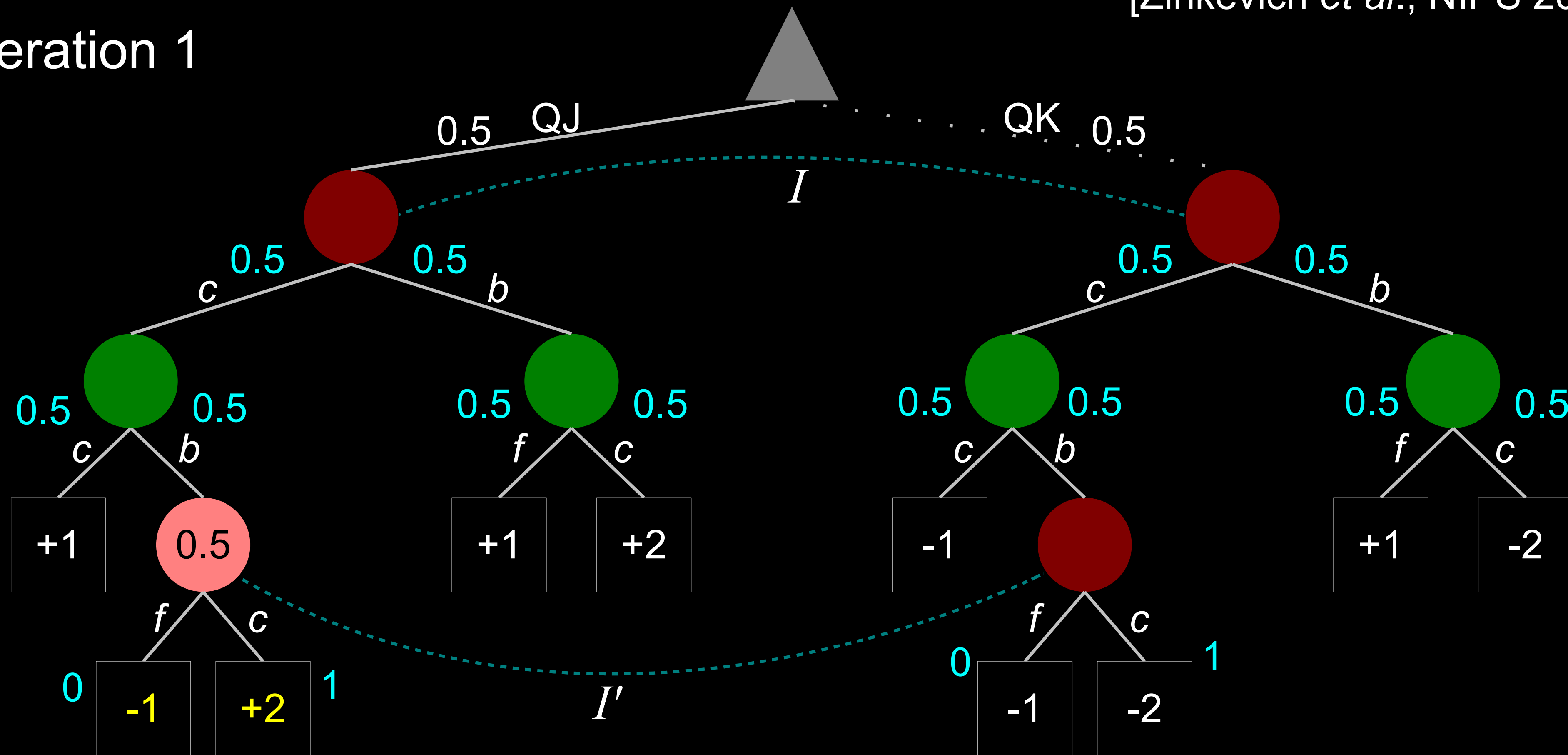
$$EV = 0.5(-1) + 0.5(+2) = 0.5$$

First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1



$\sigma_1(\ f\ )$: -1 < 0.5

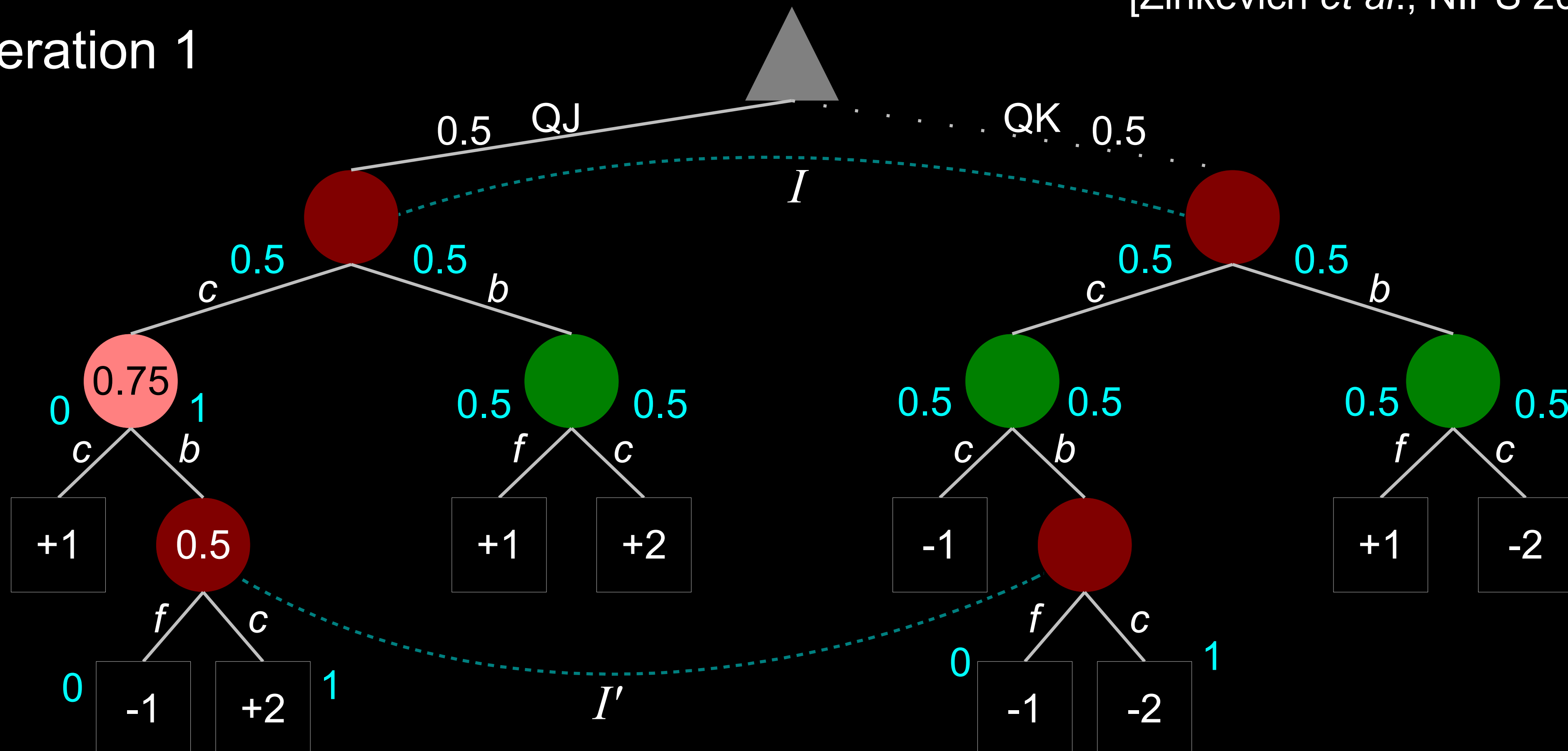
$\sigma_1(\ c\ )$: +2 > 0.5

First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1

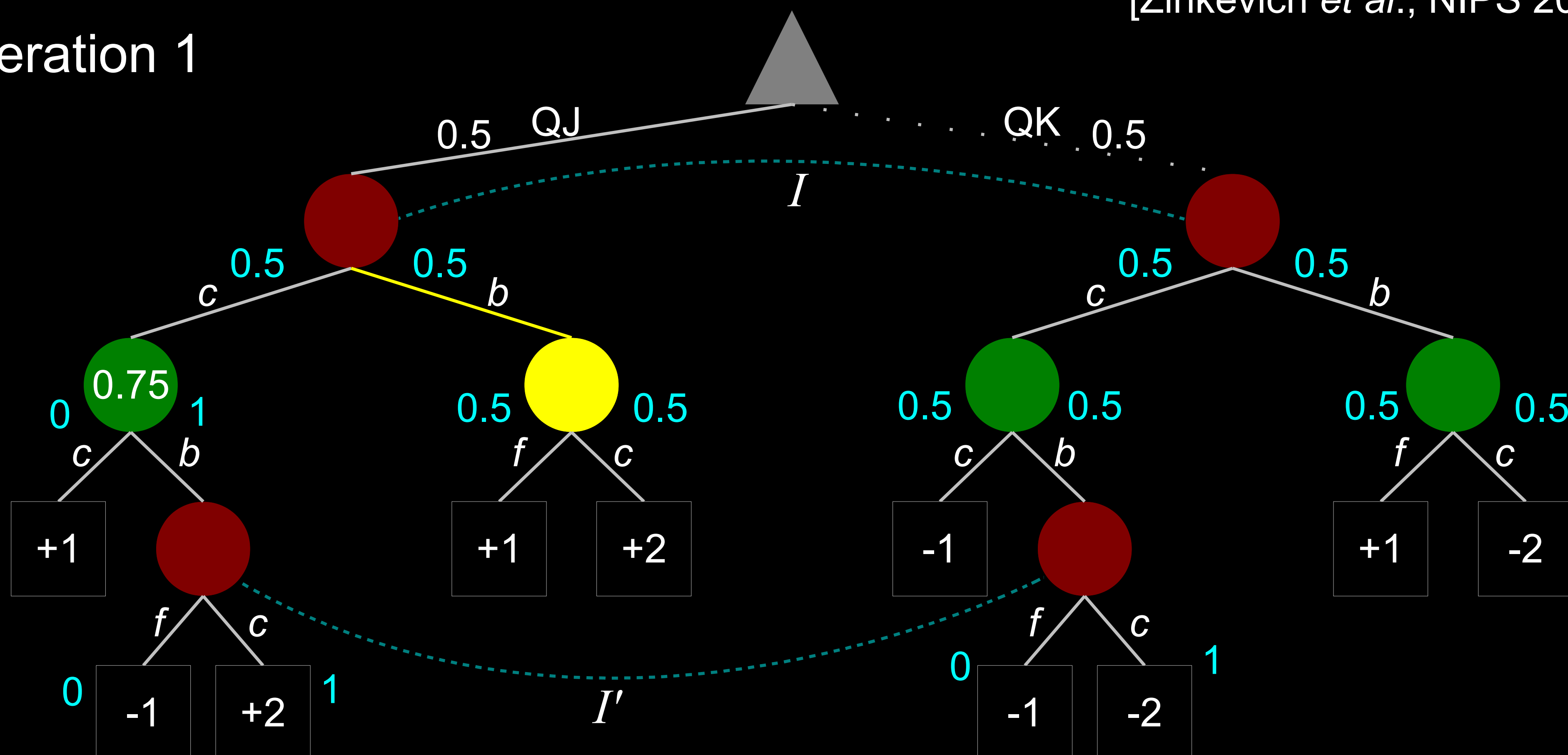


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1

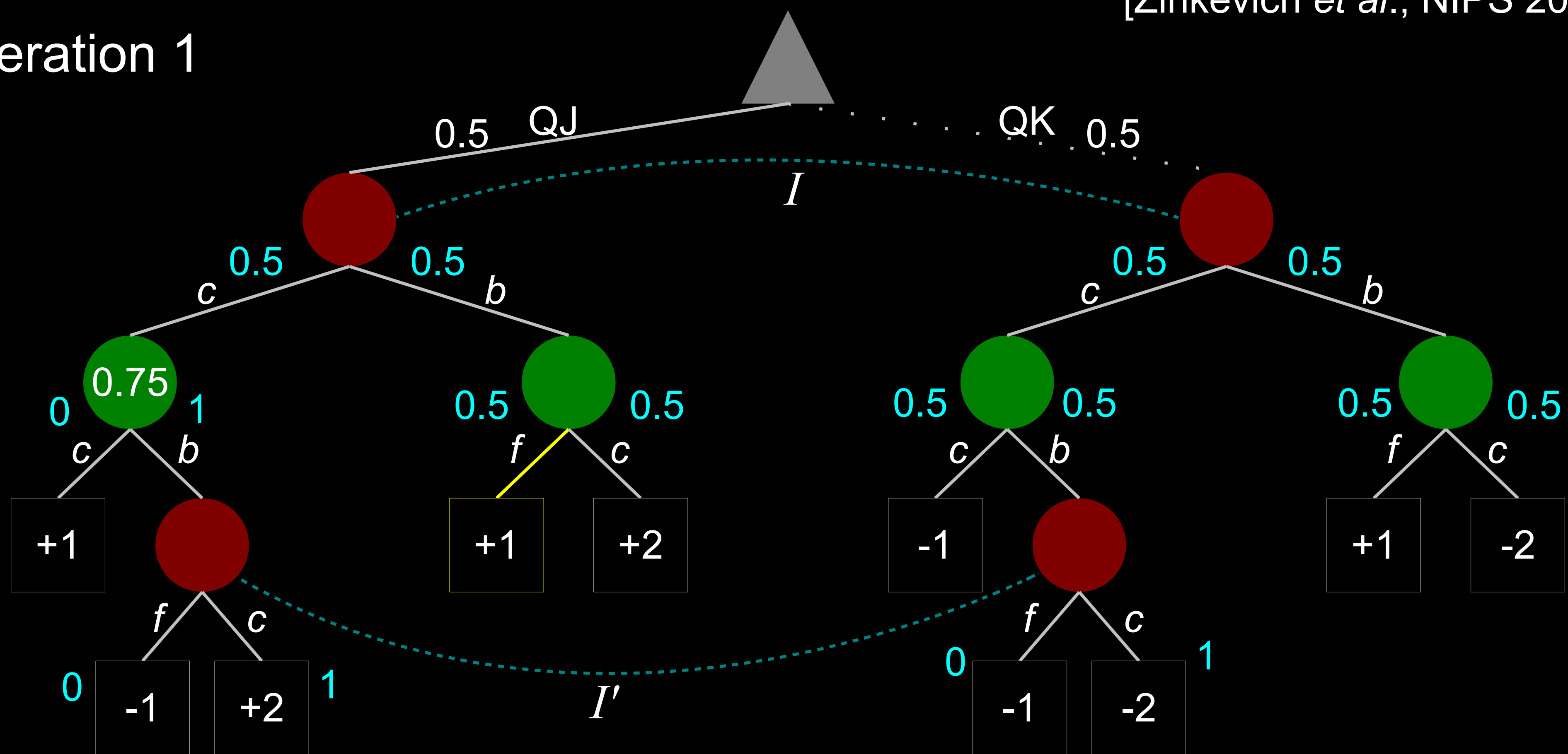


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1

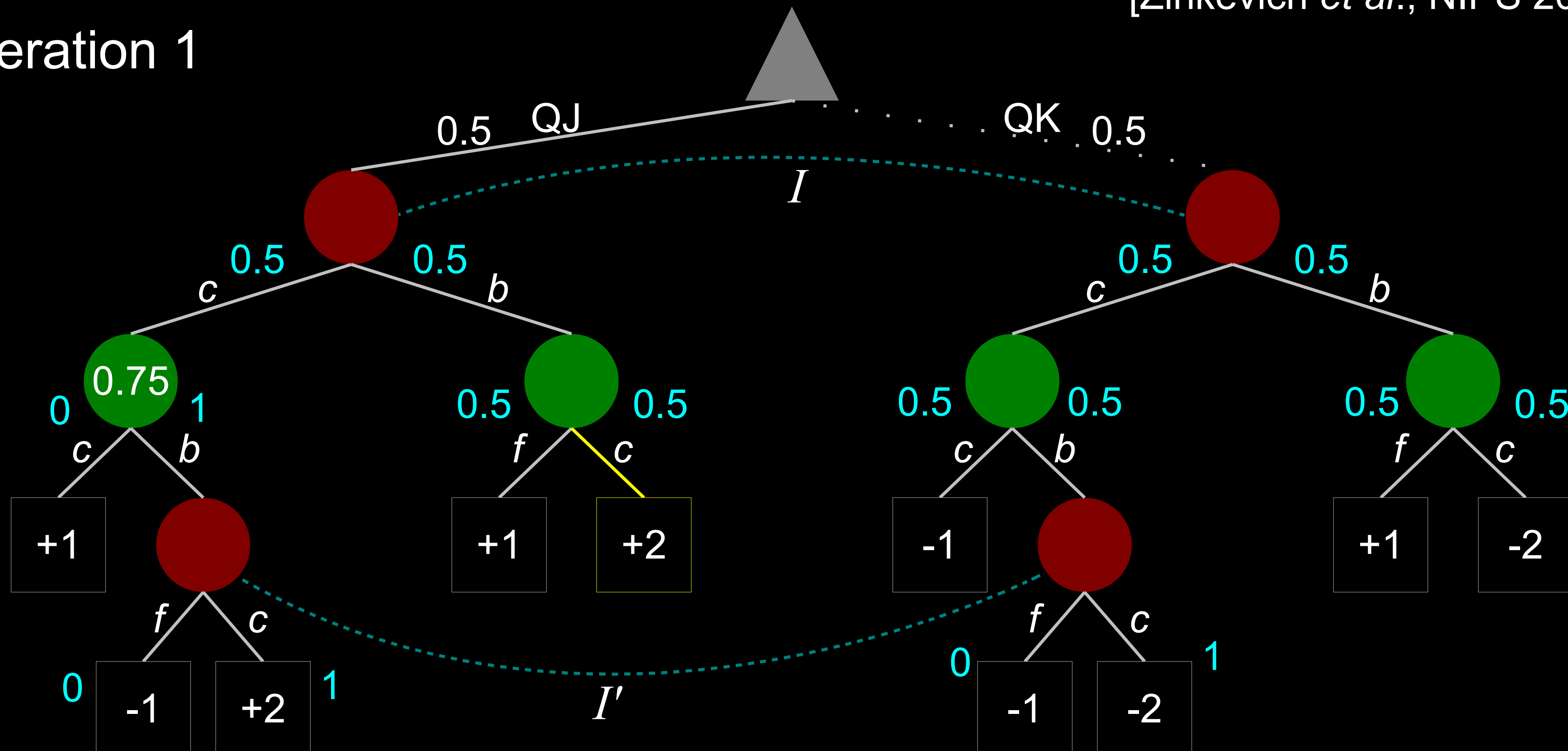


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1

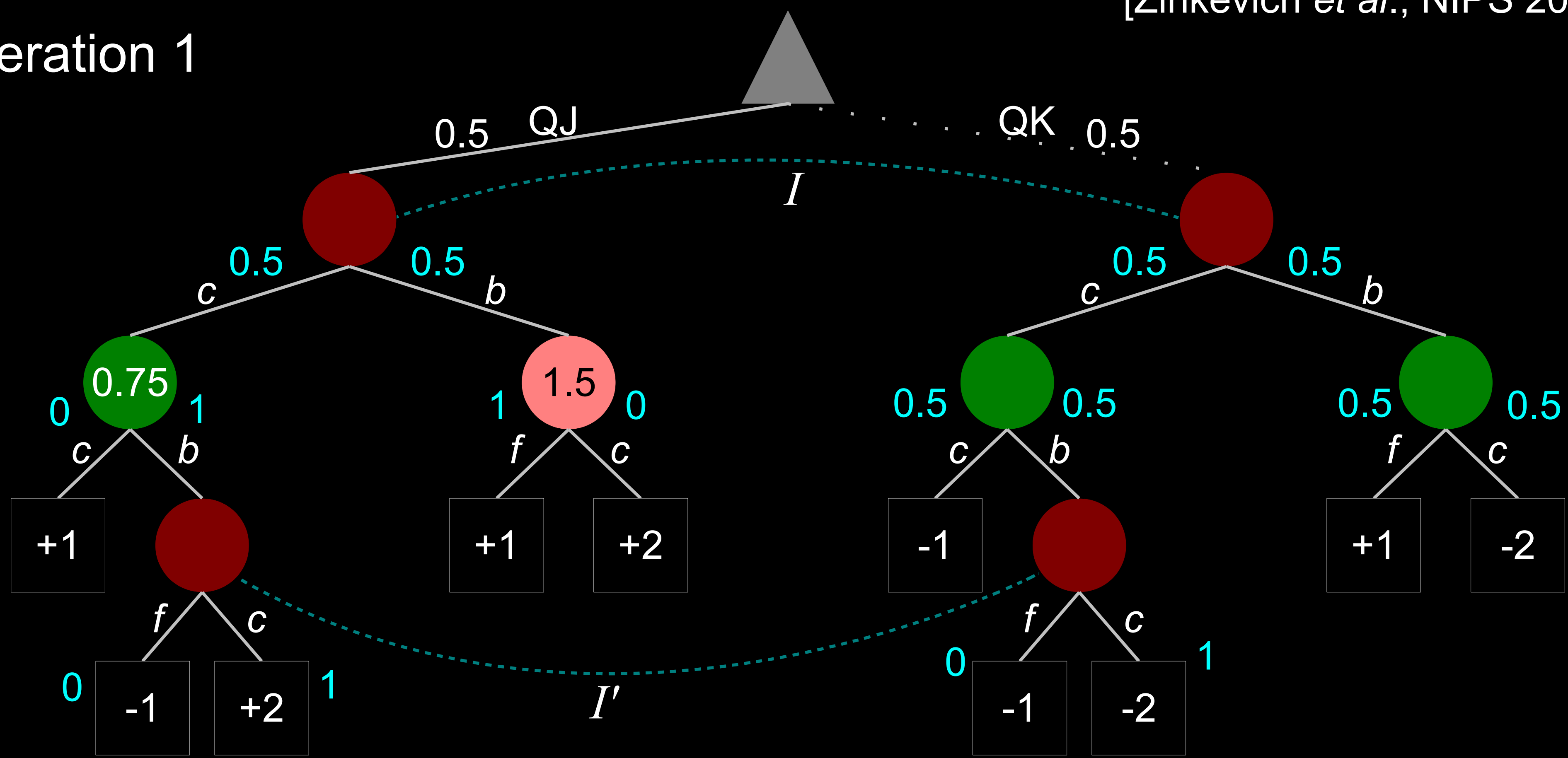


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1

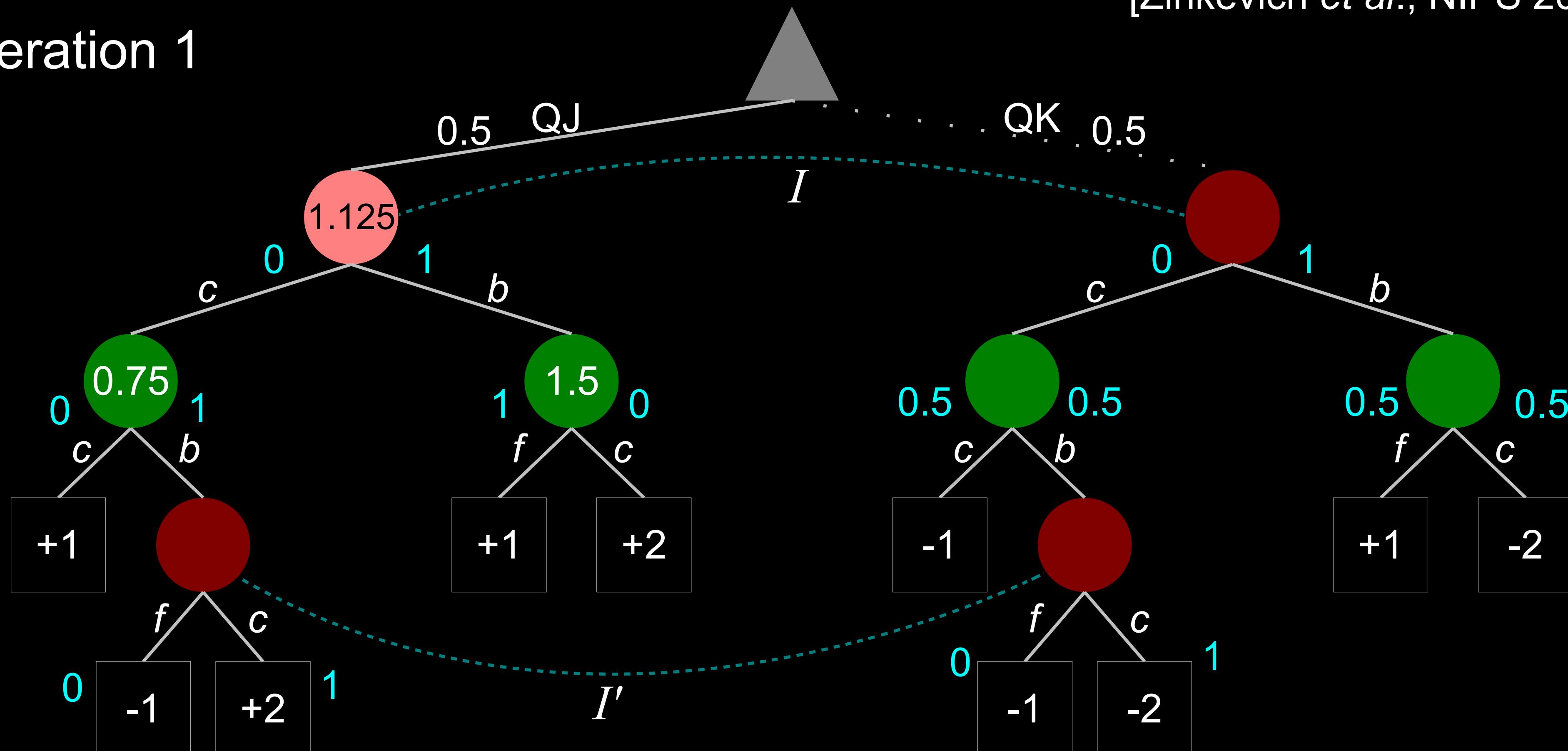


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al*., NIPS 2007]

- Iteration 1

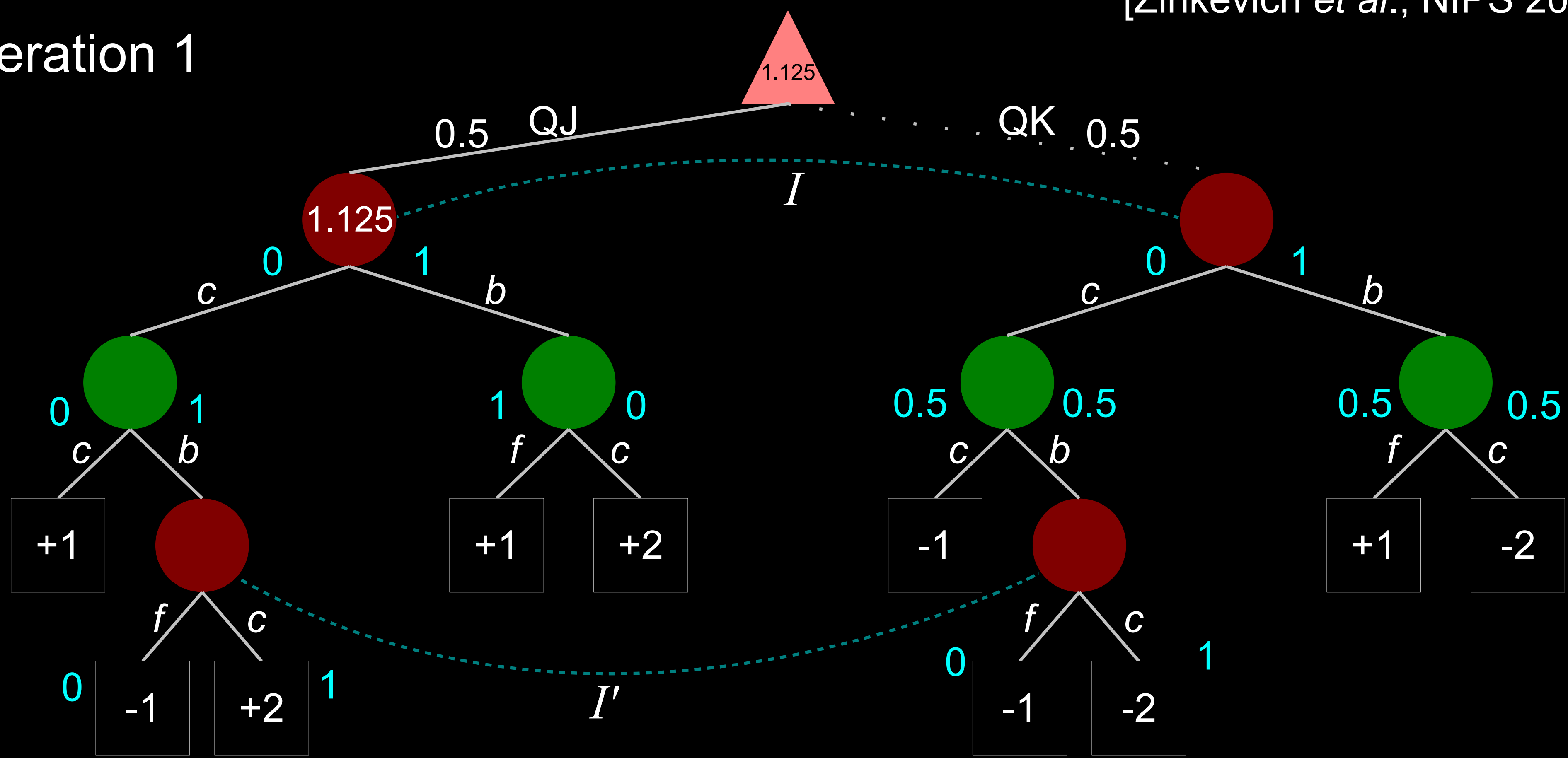


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1

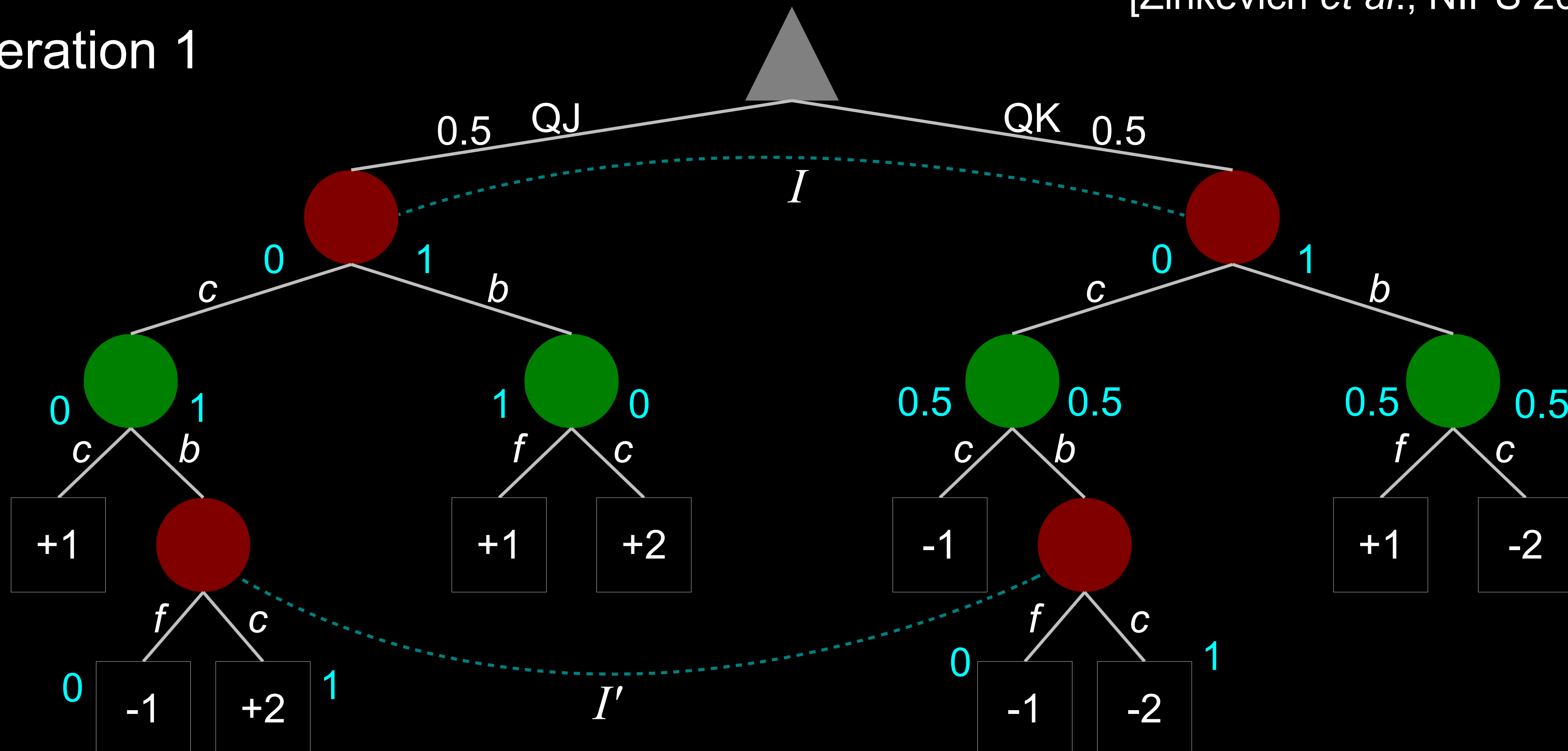


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 1

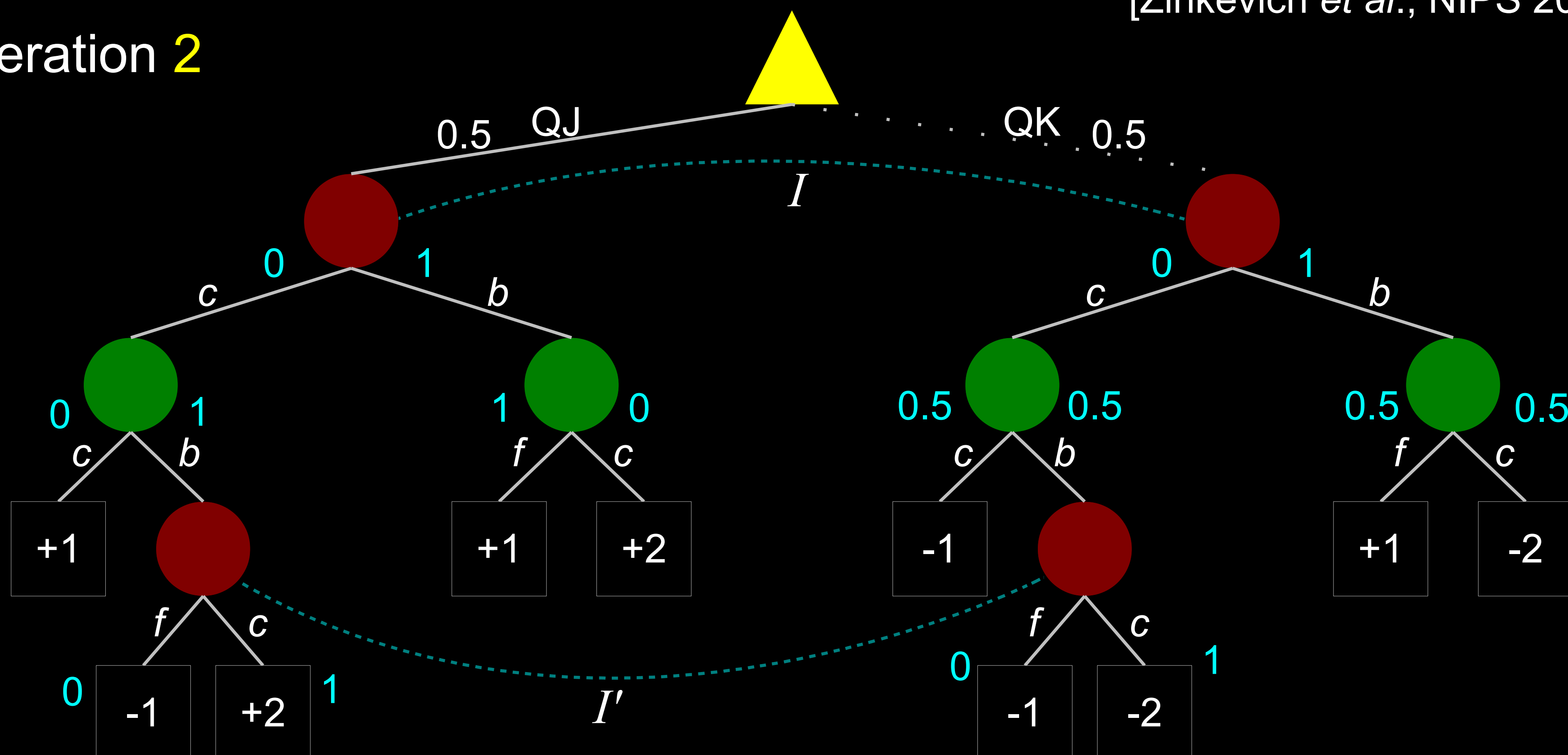


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al*., NIPS 2007]

- Iteration 2

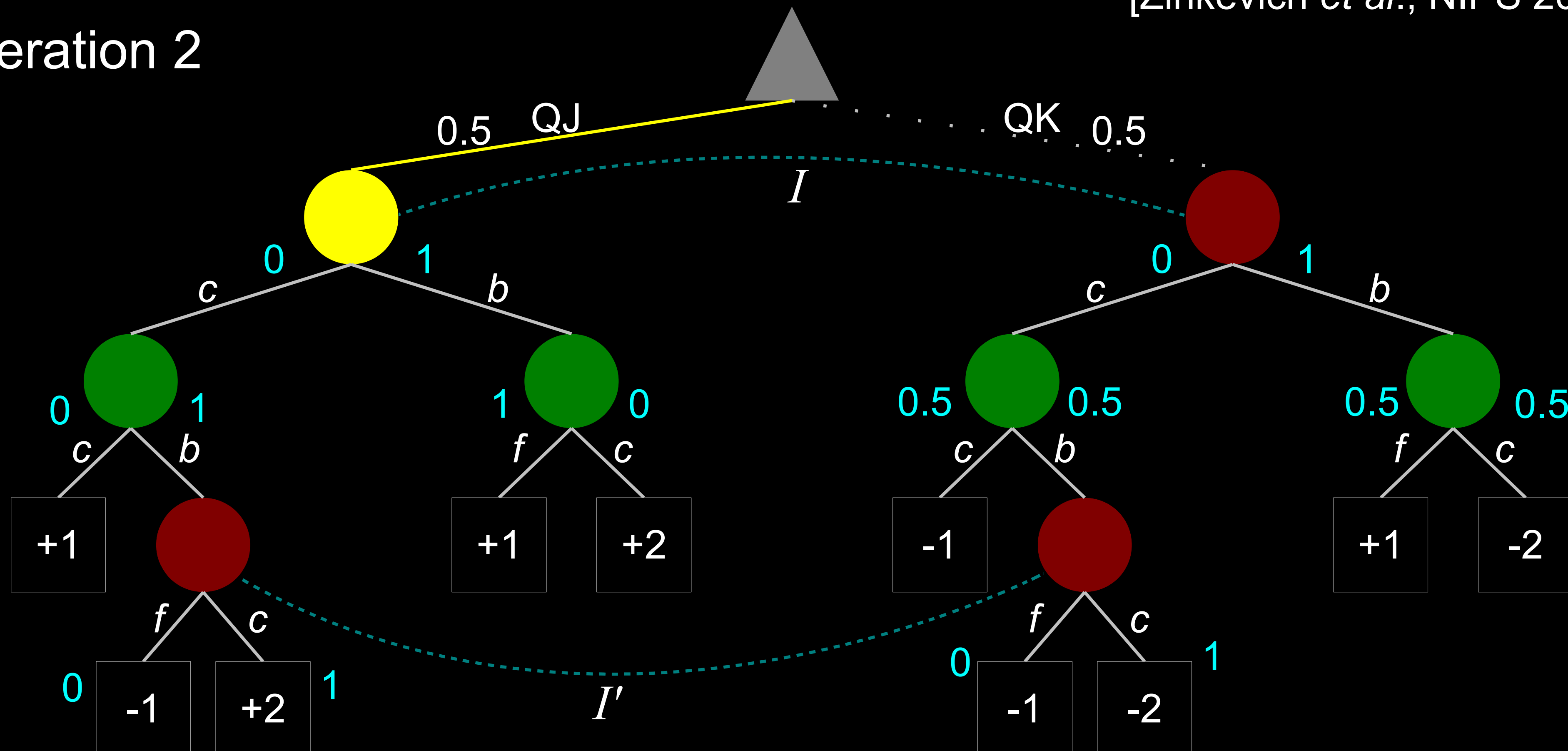


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

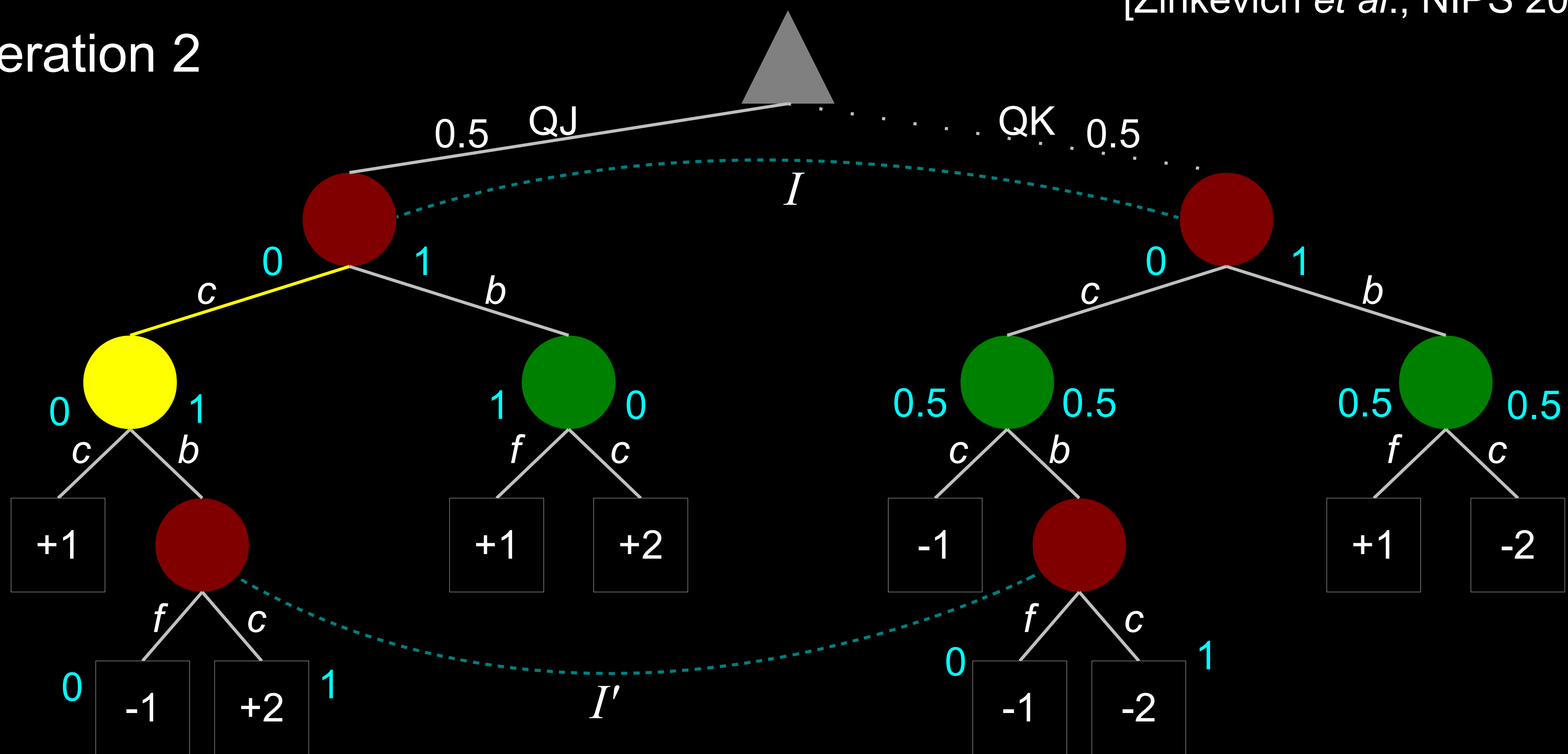


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

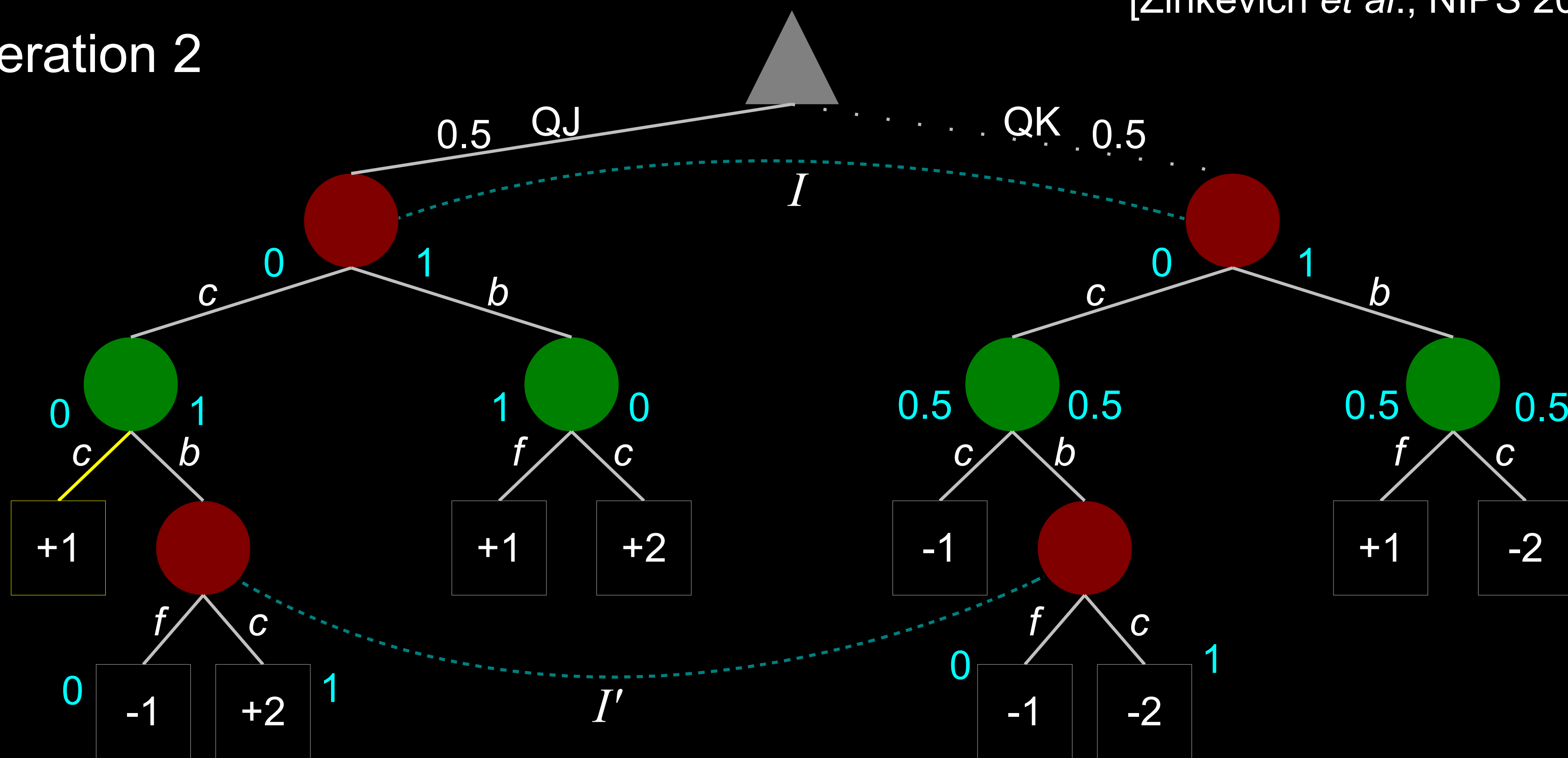


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

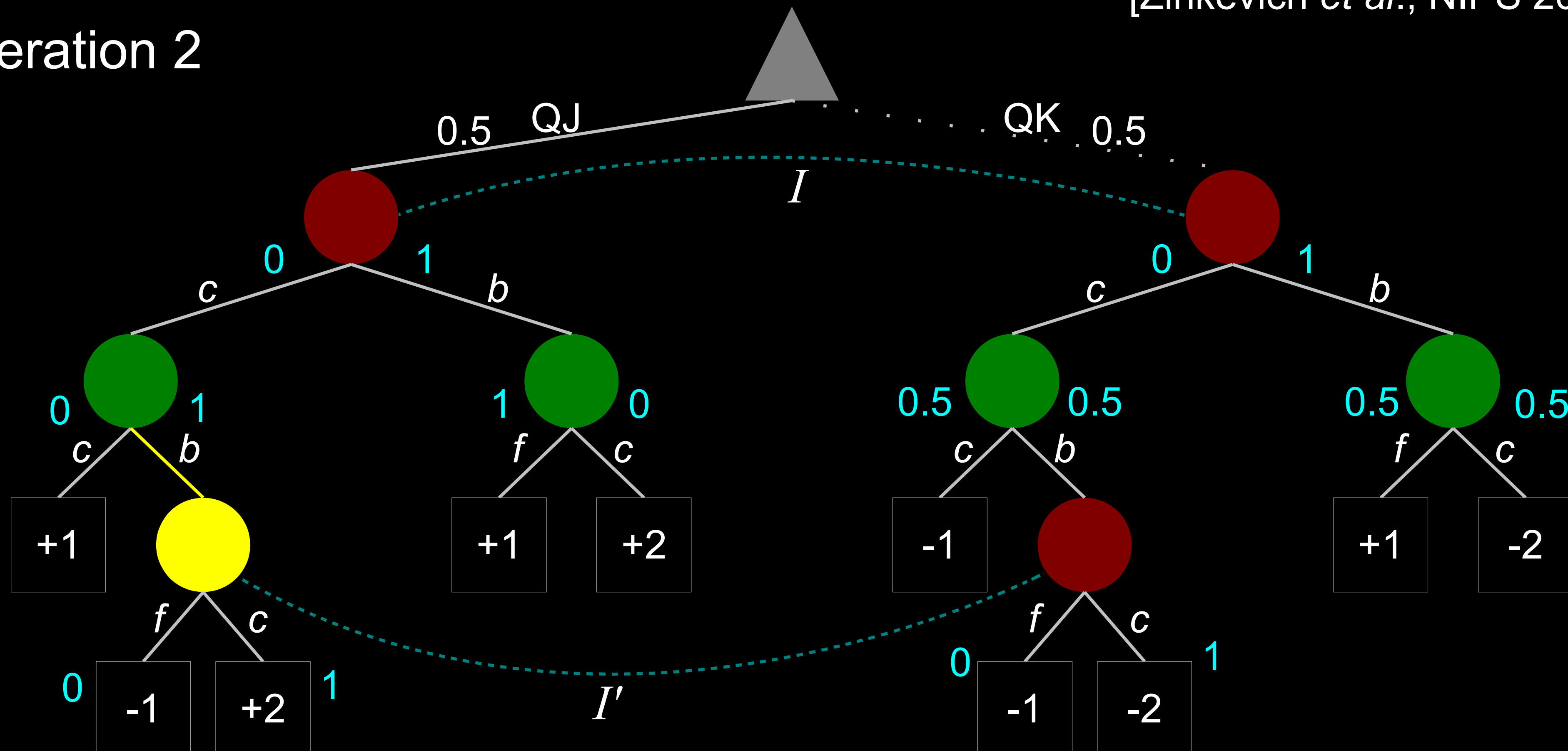


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

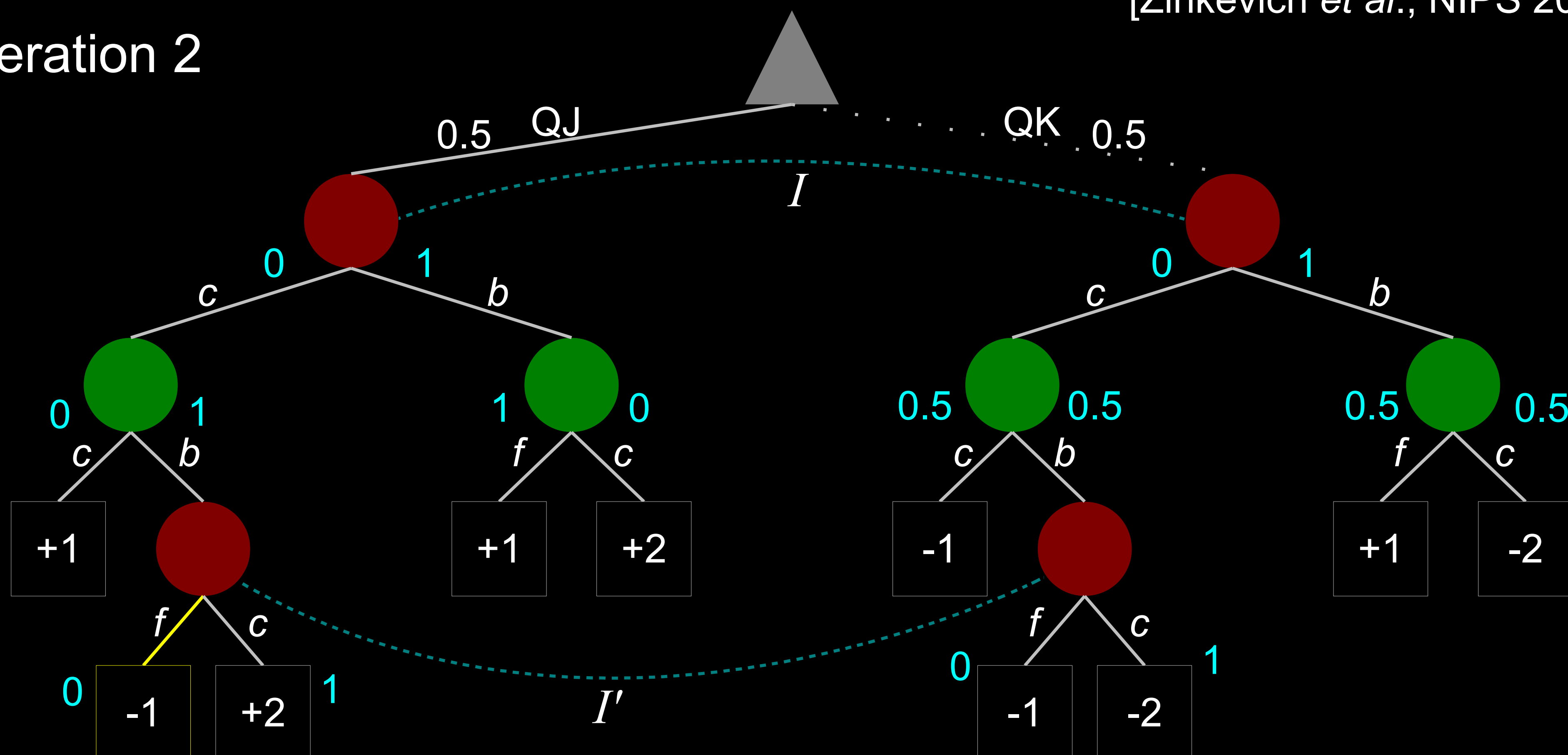


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

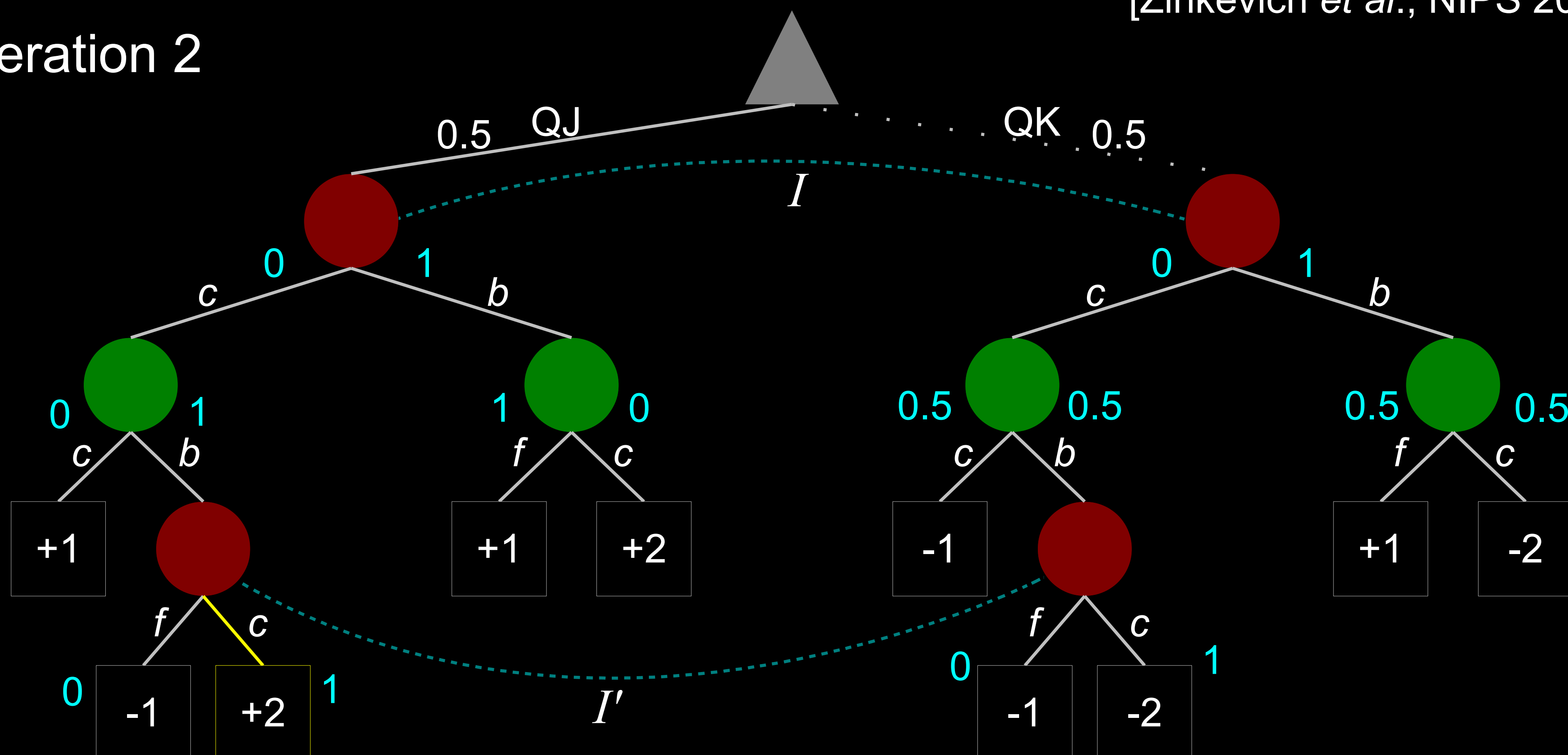


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

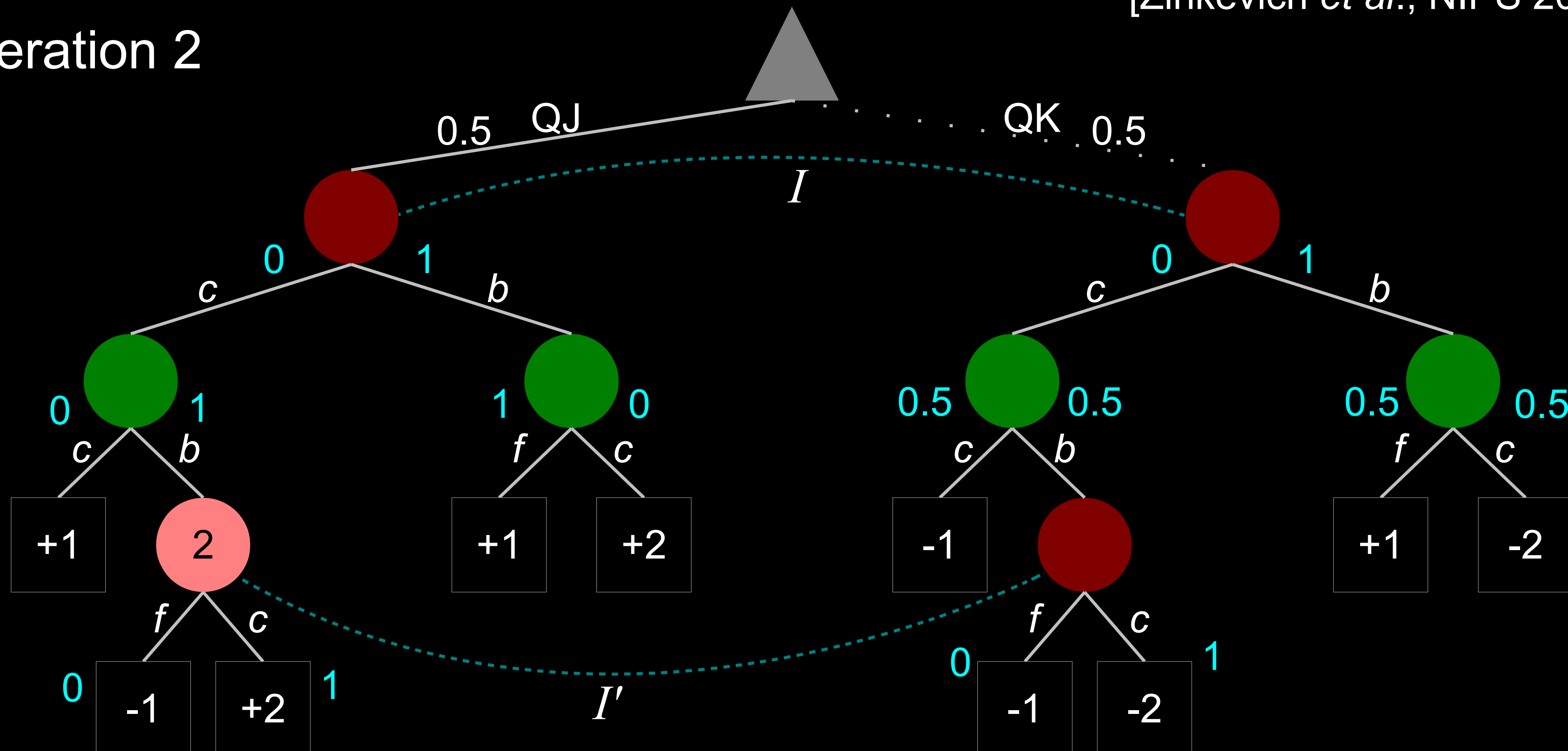


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

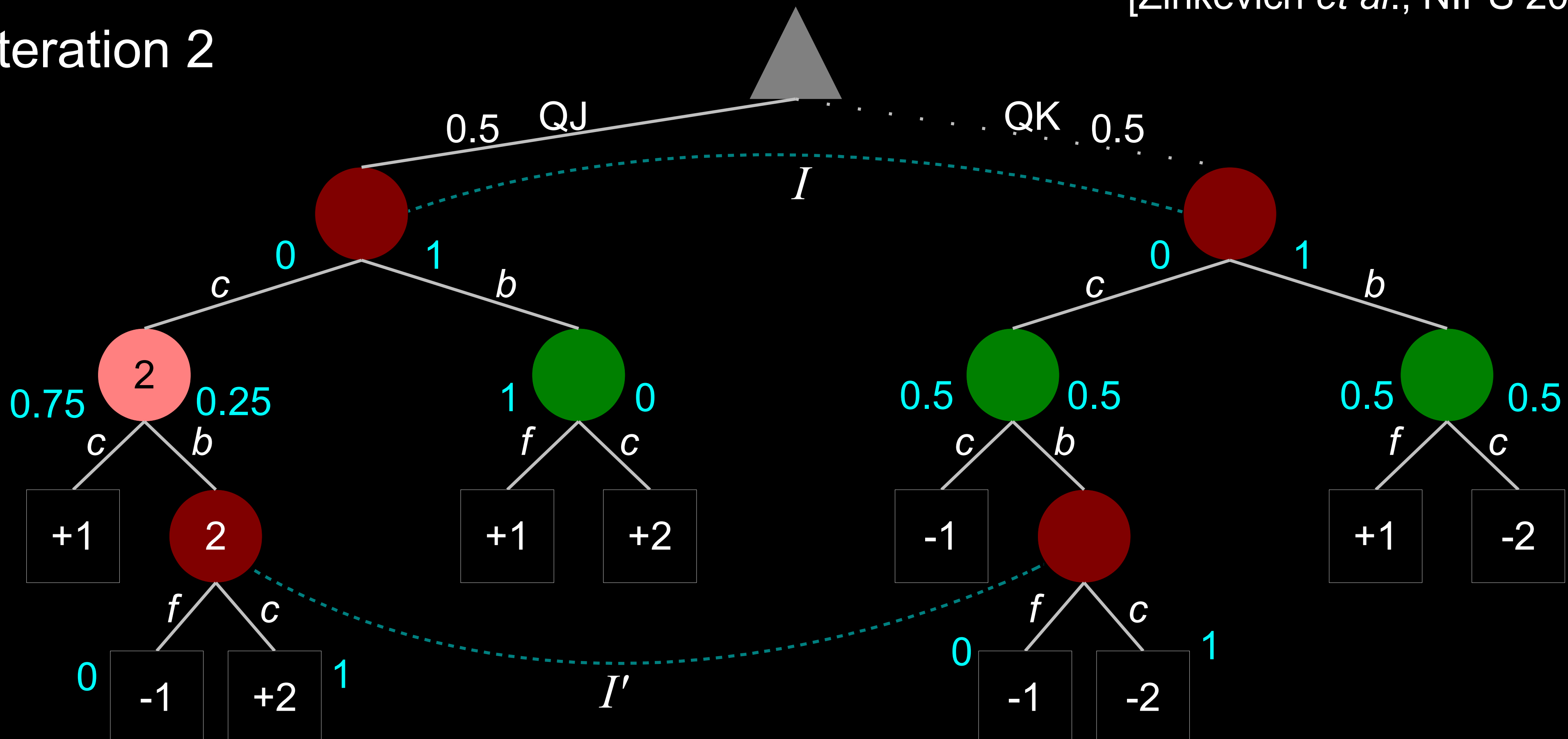


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

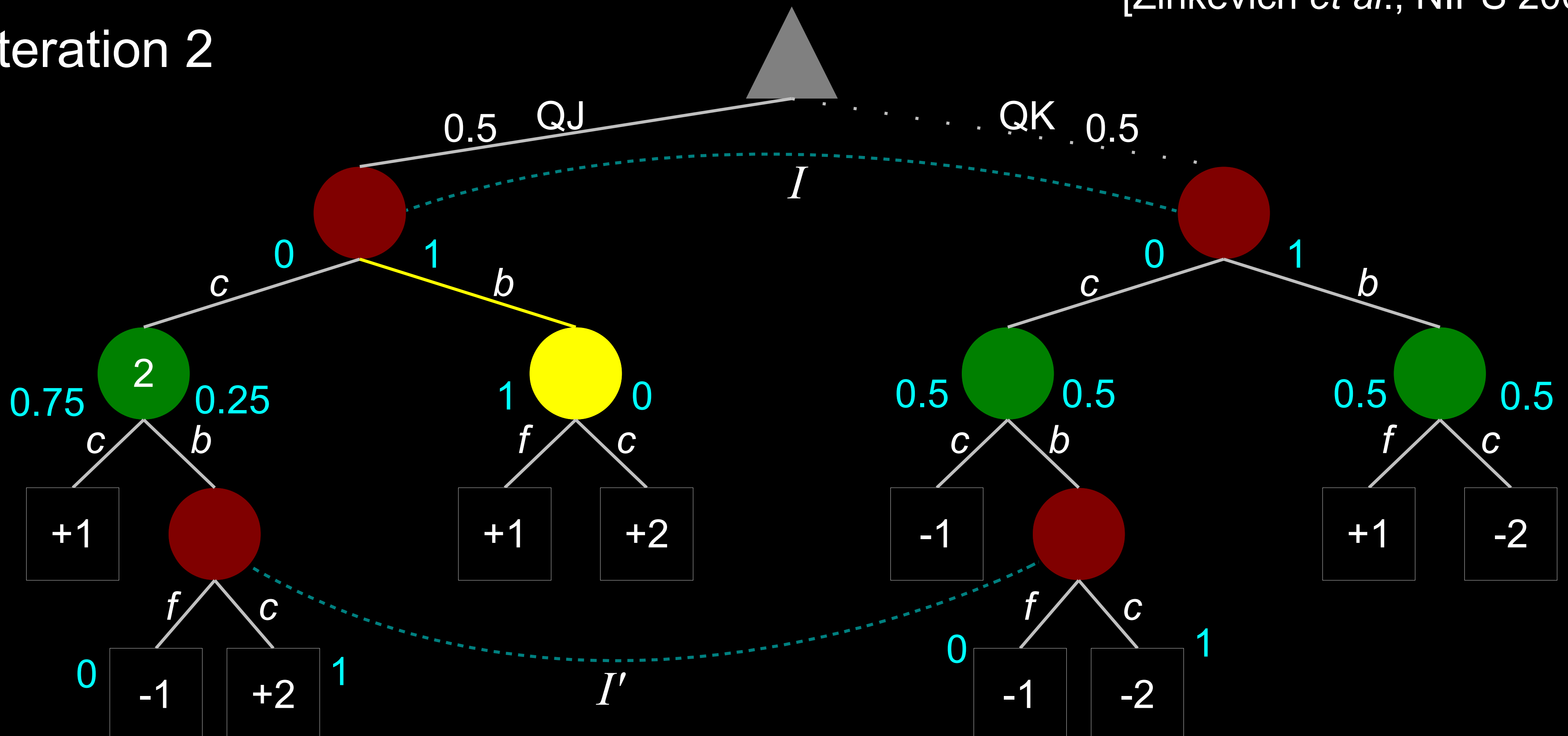


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

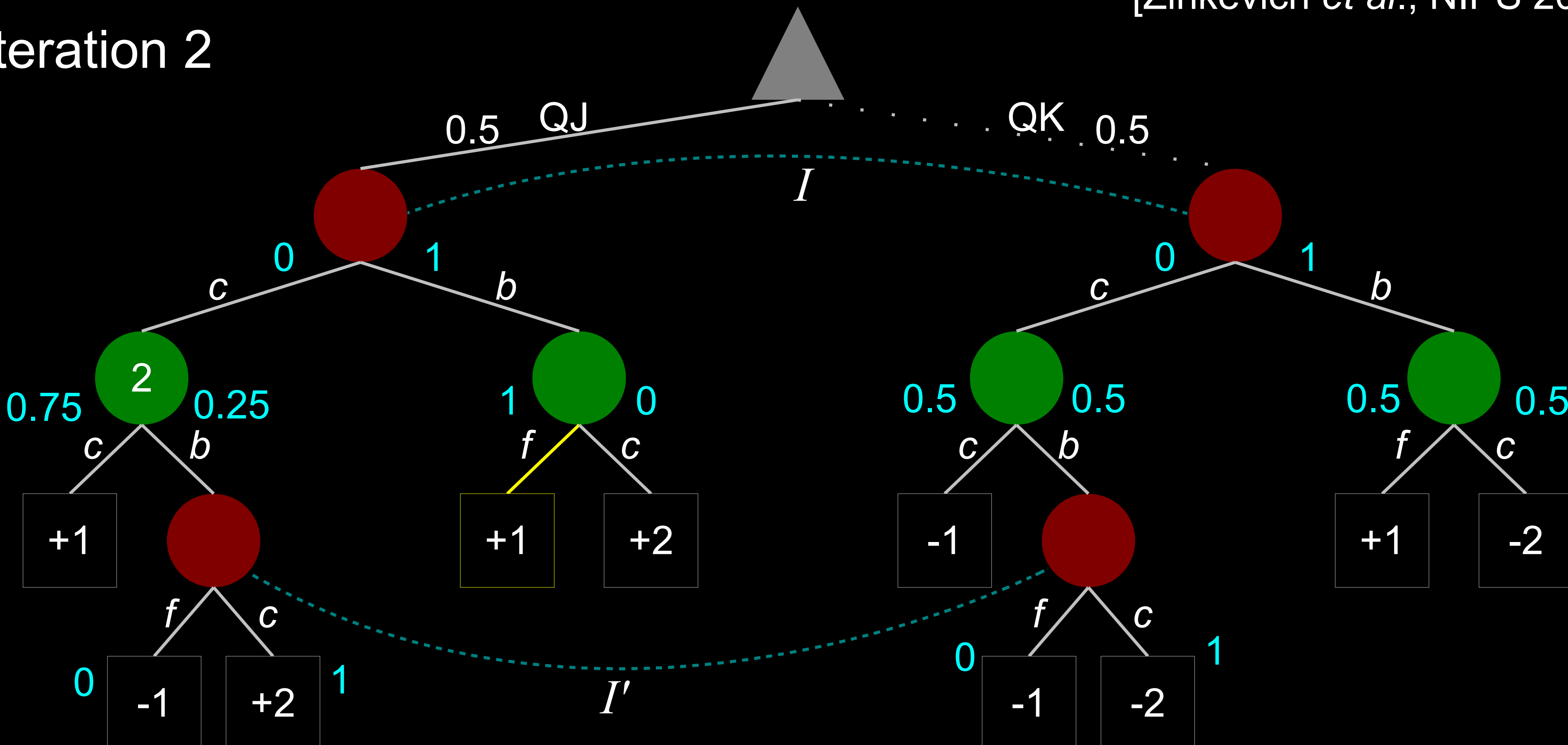


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

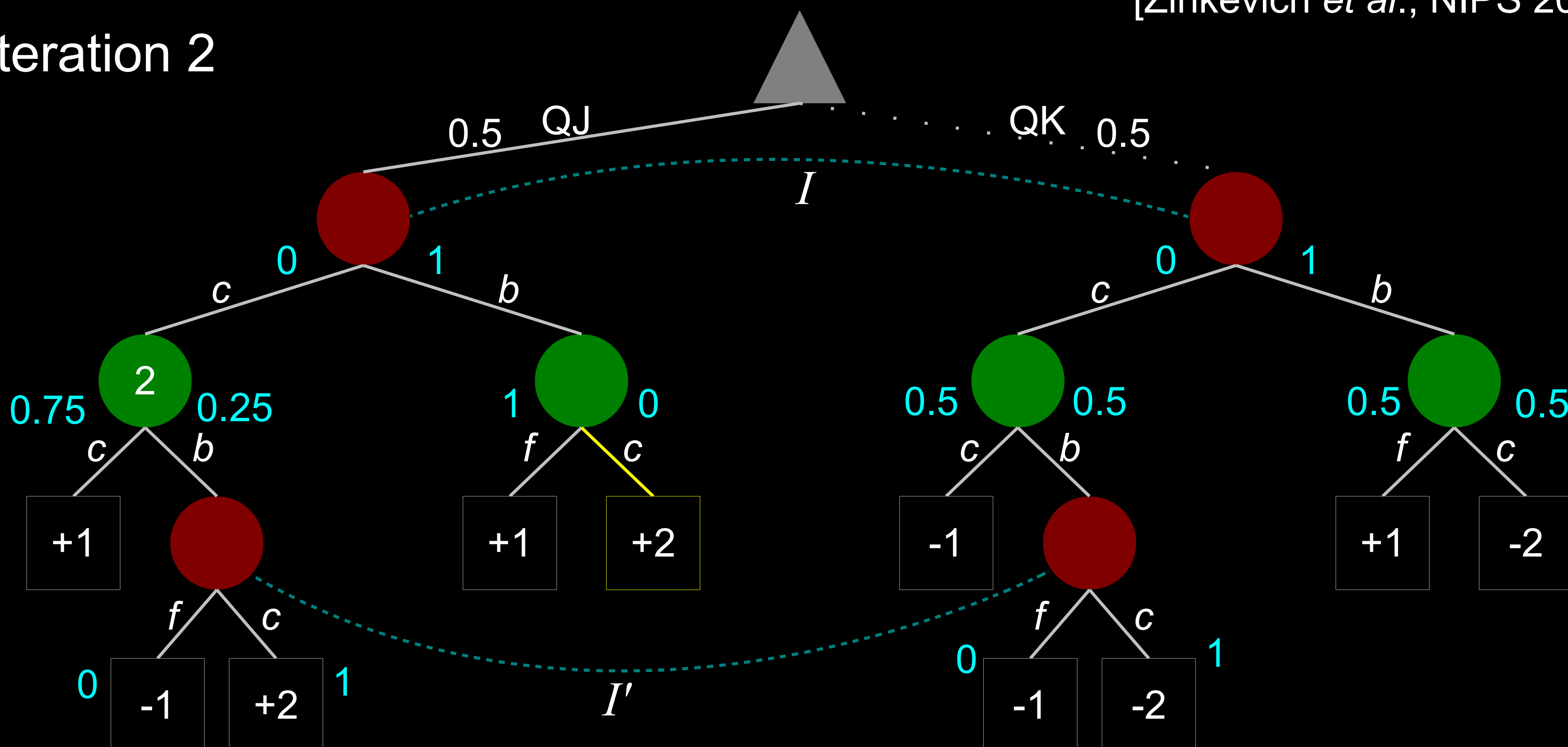


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

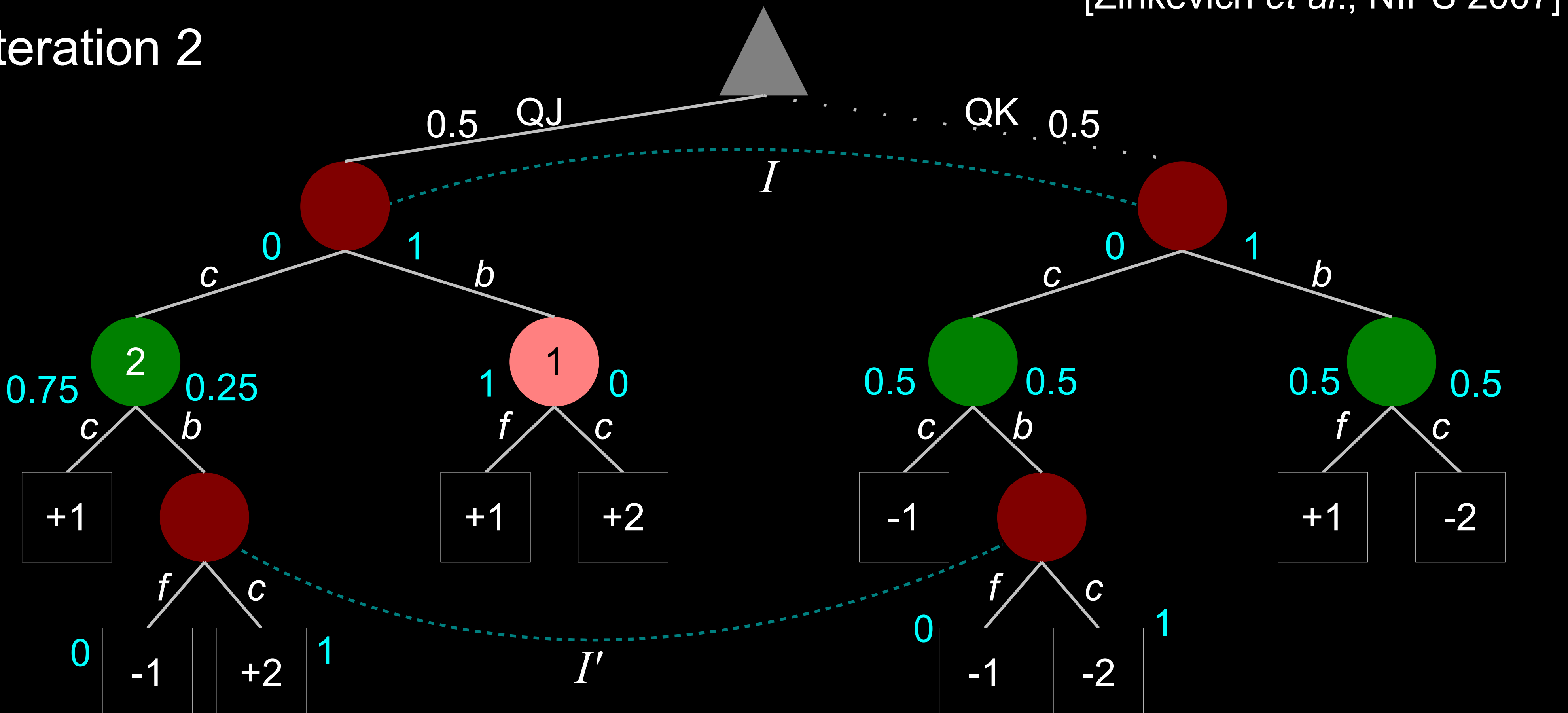


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

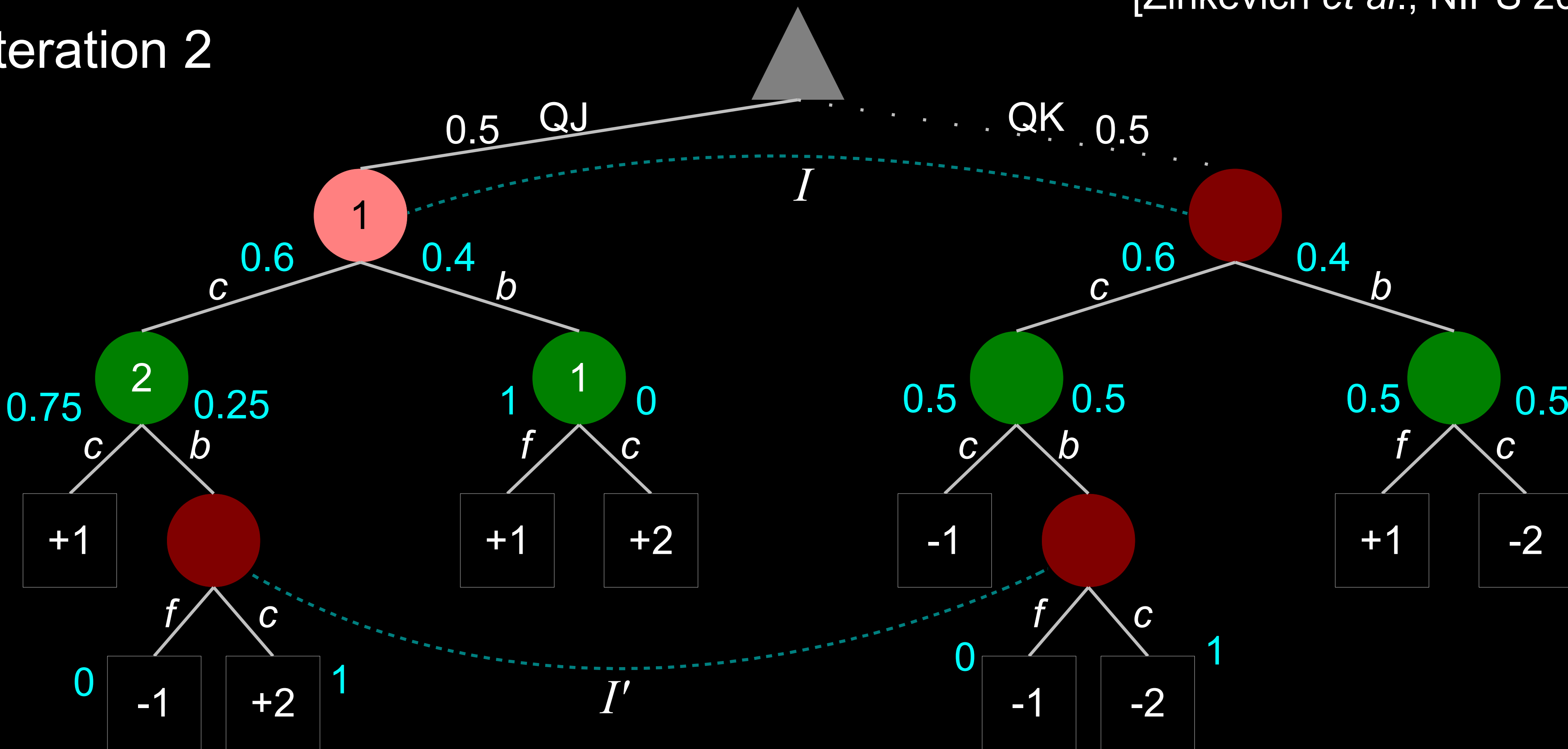


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

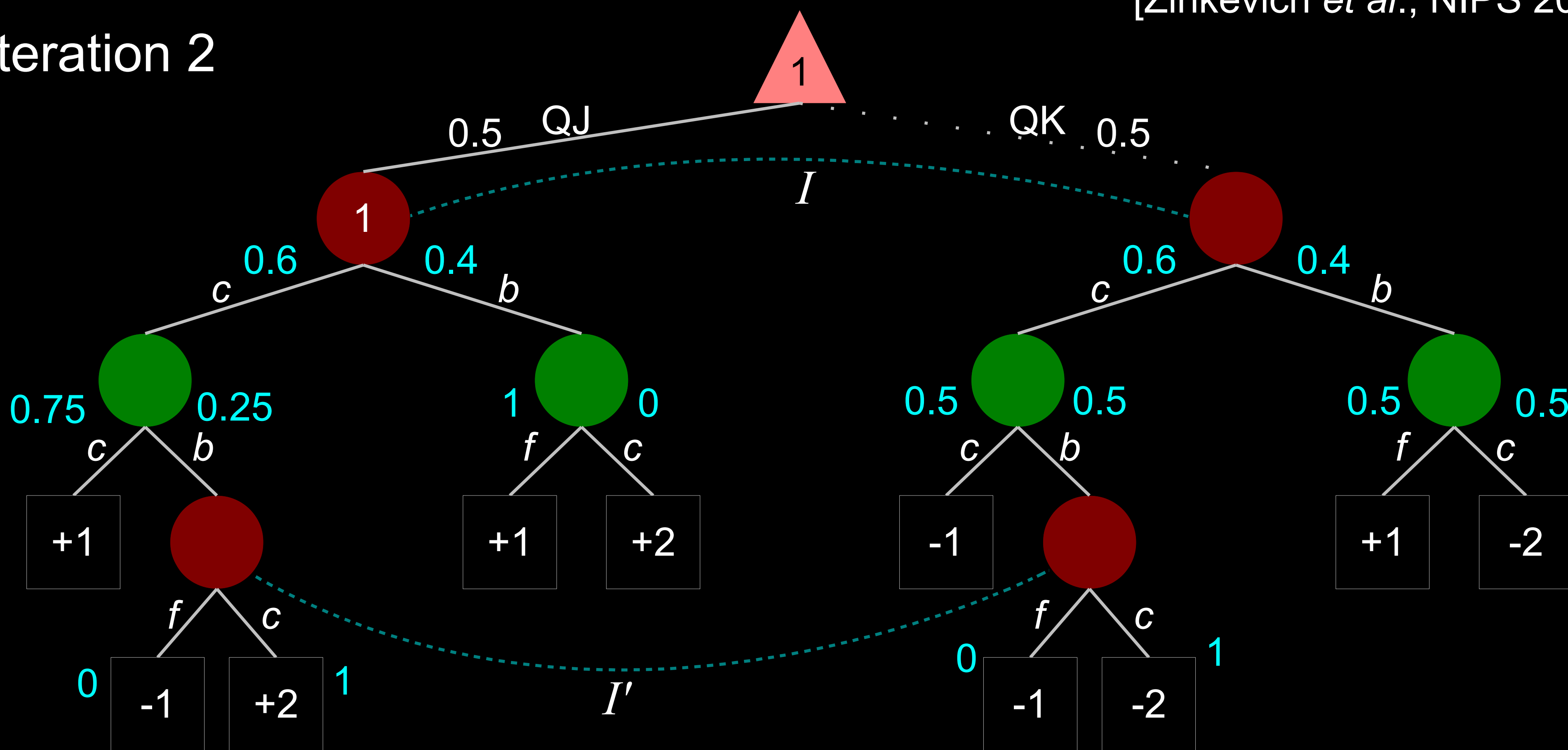


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

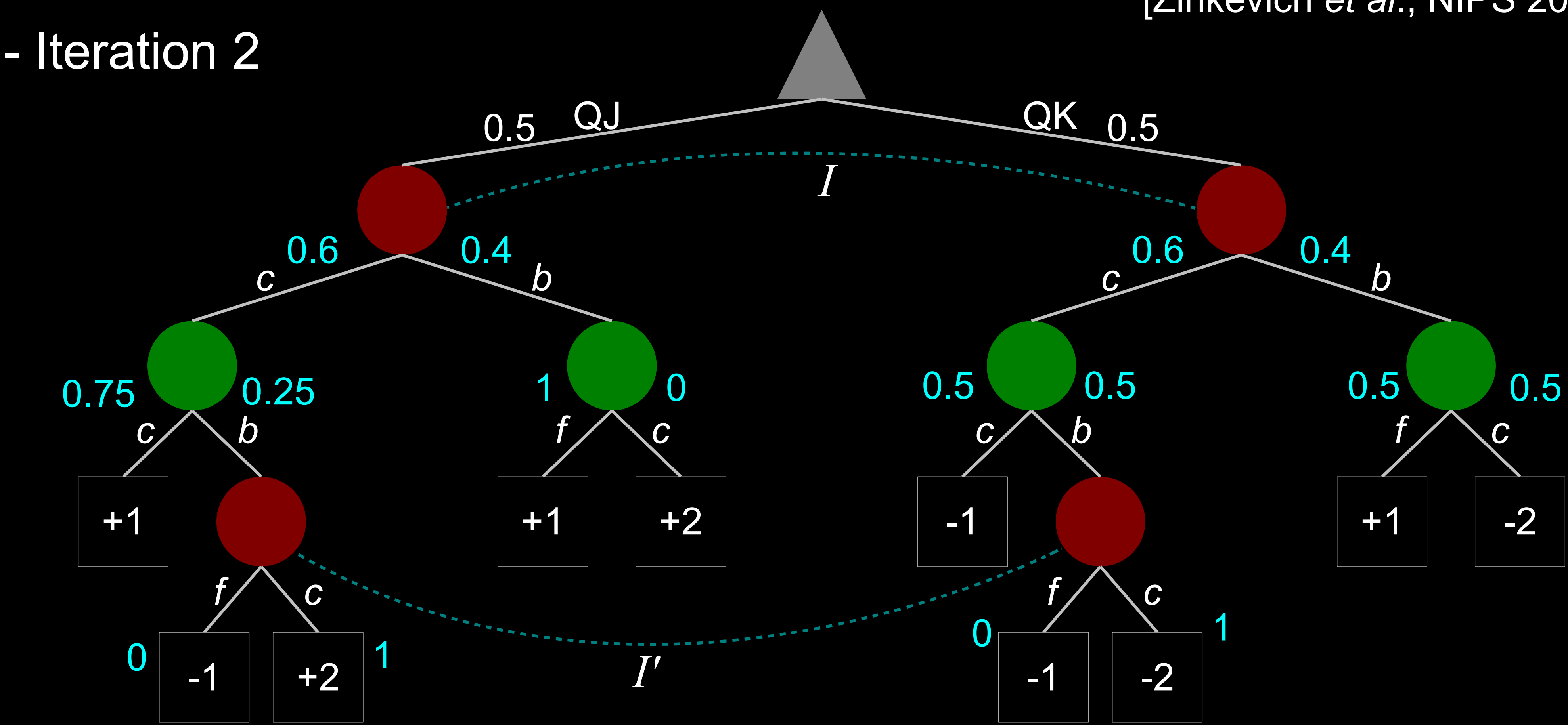


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 2

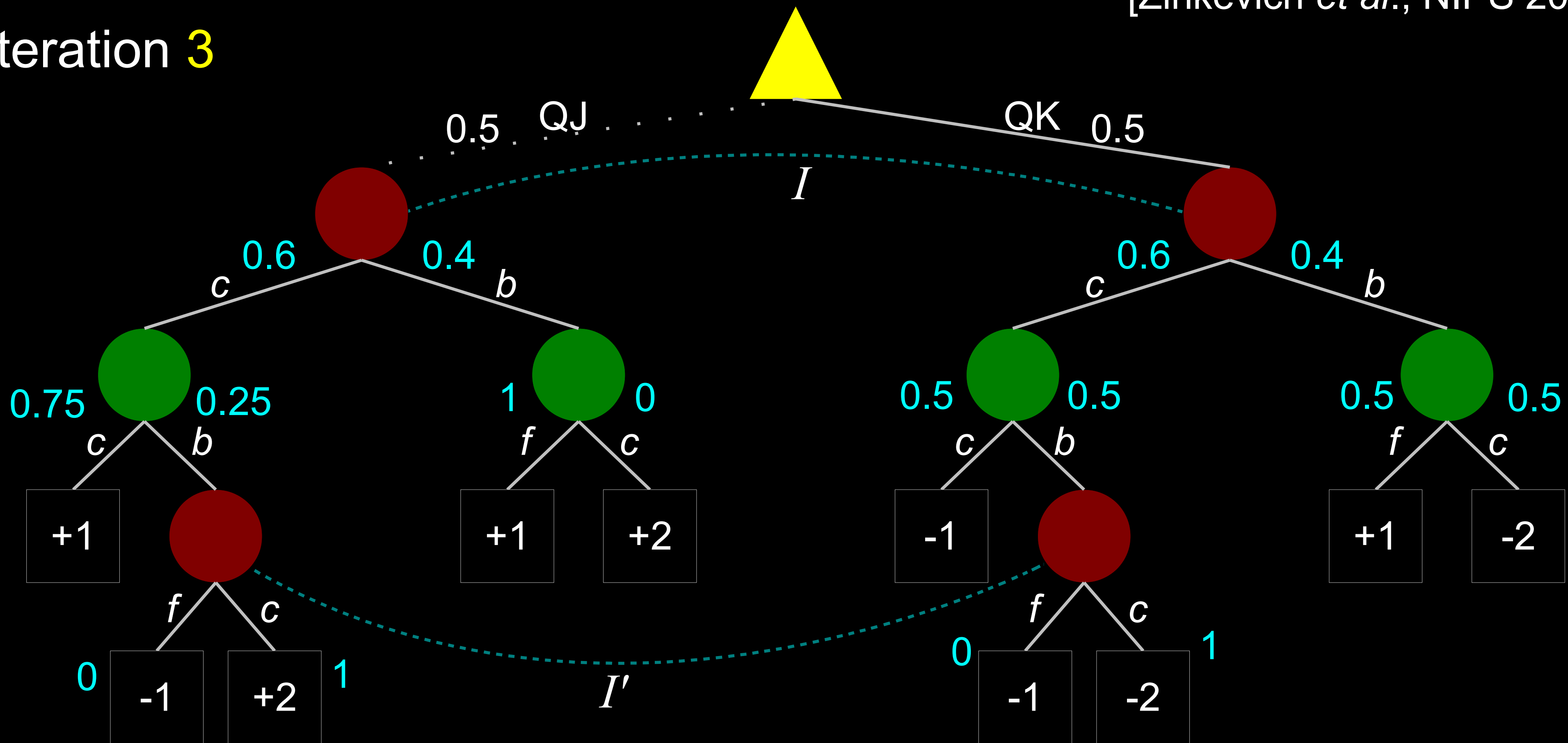


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 3

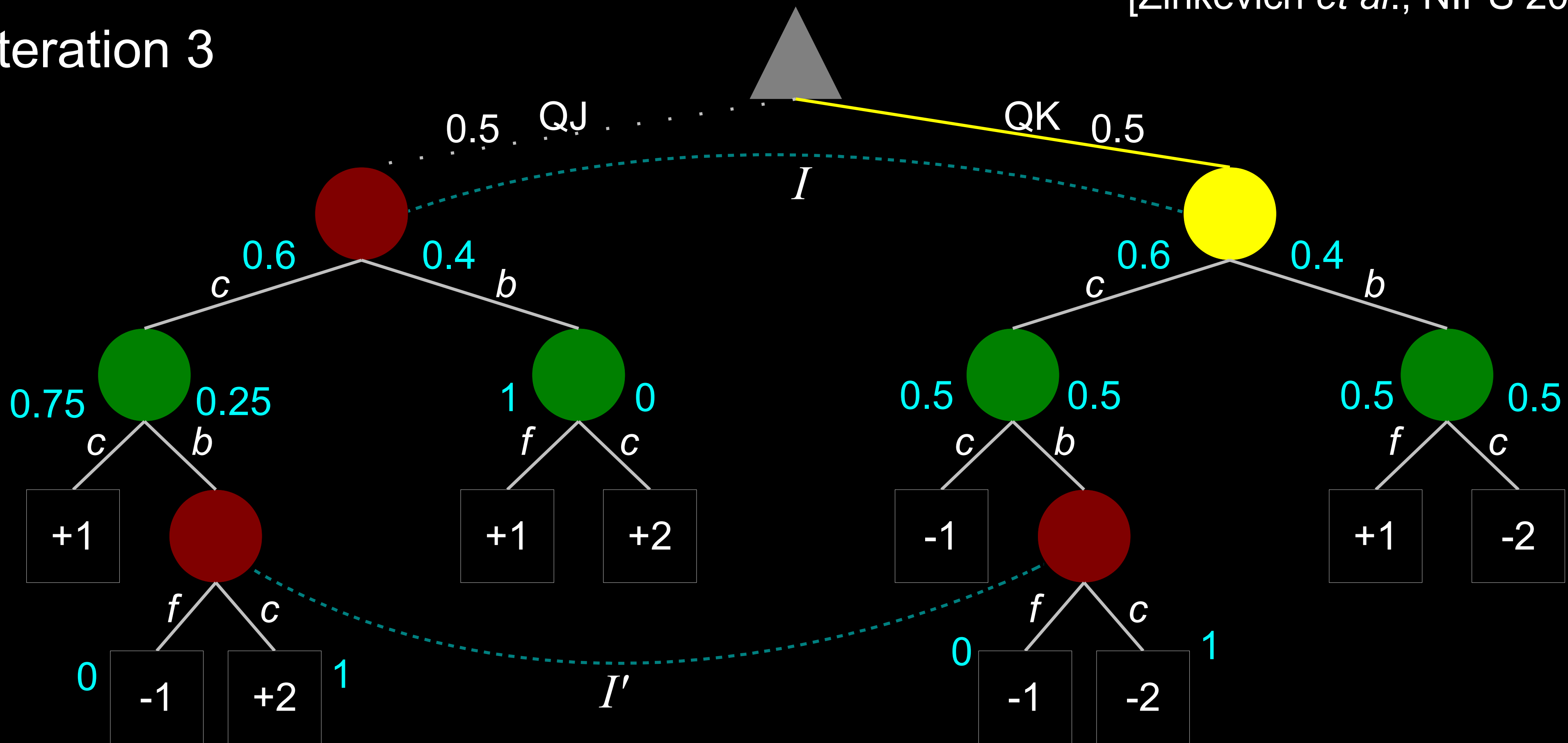


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 3

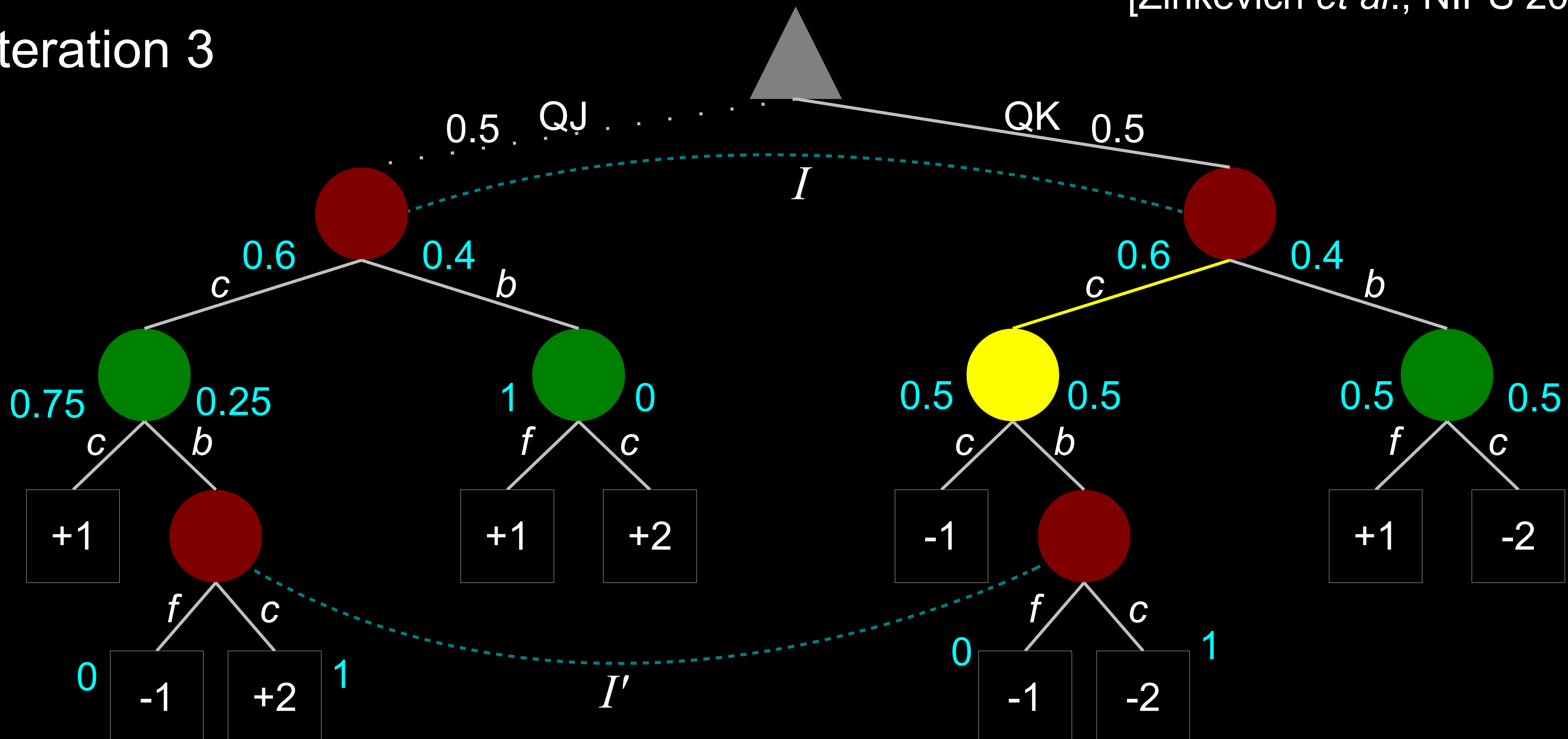


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 3

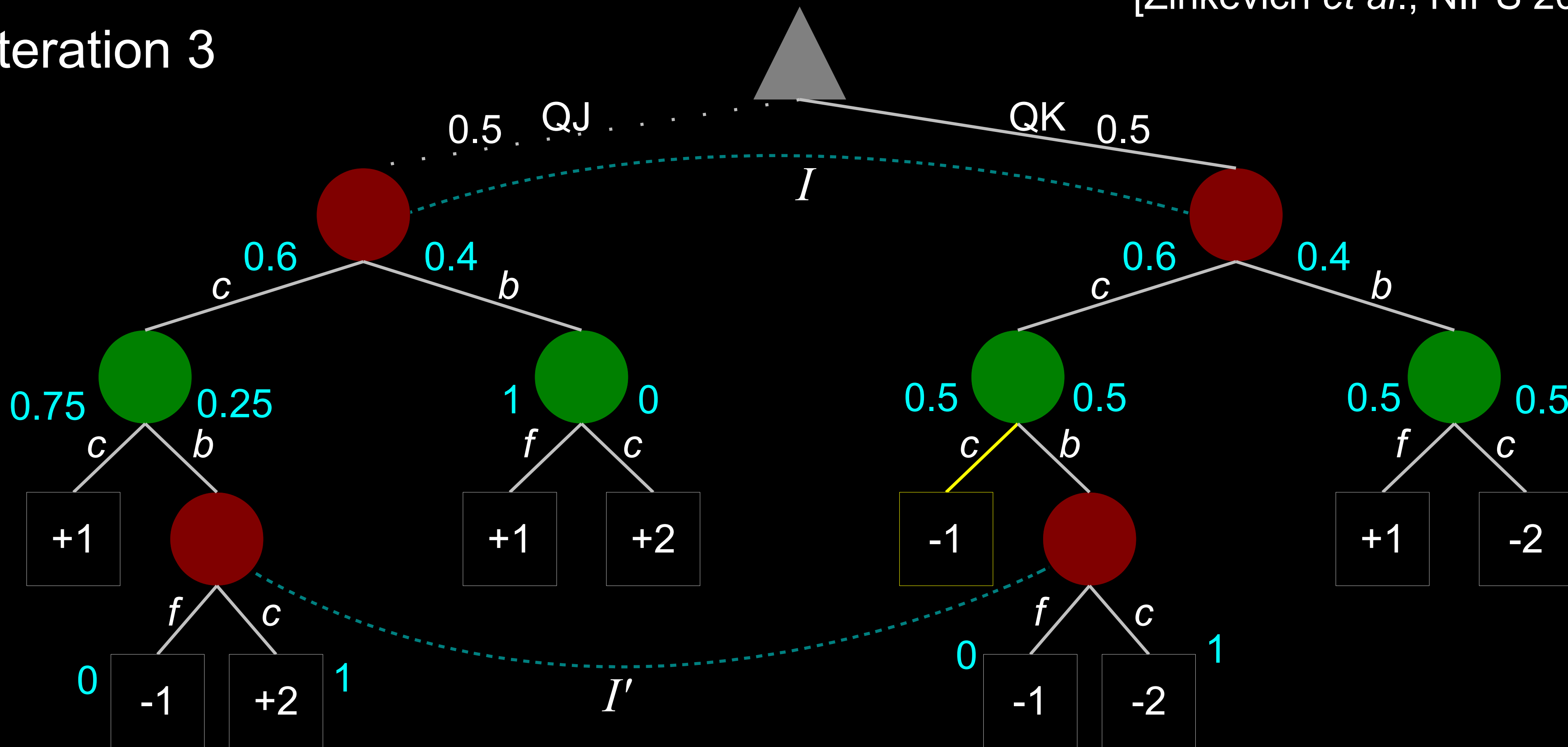


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 3

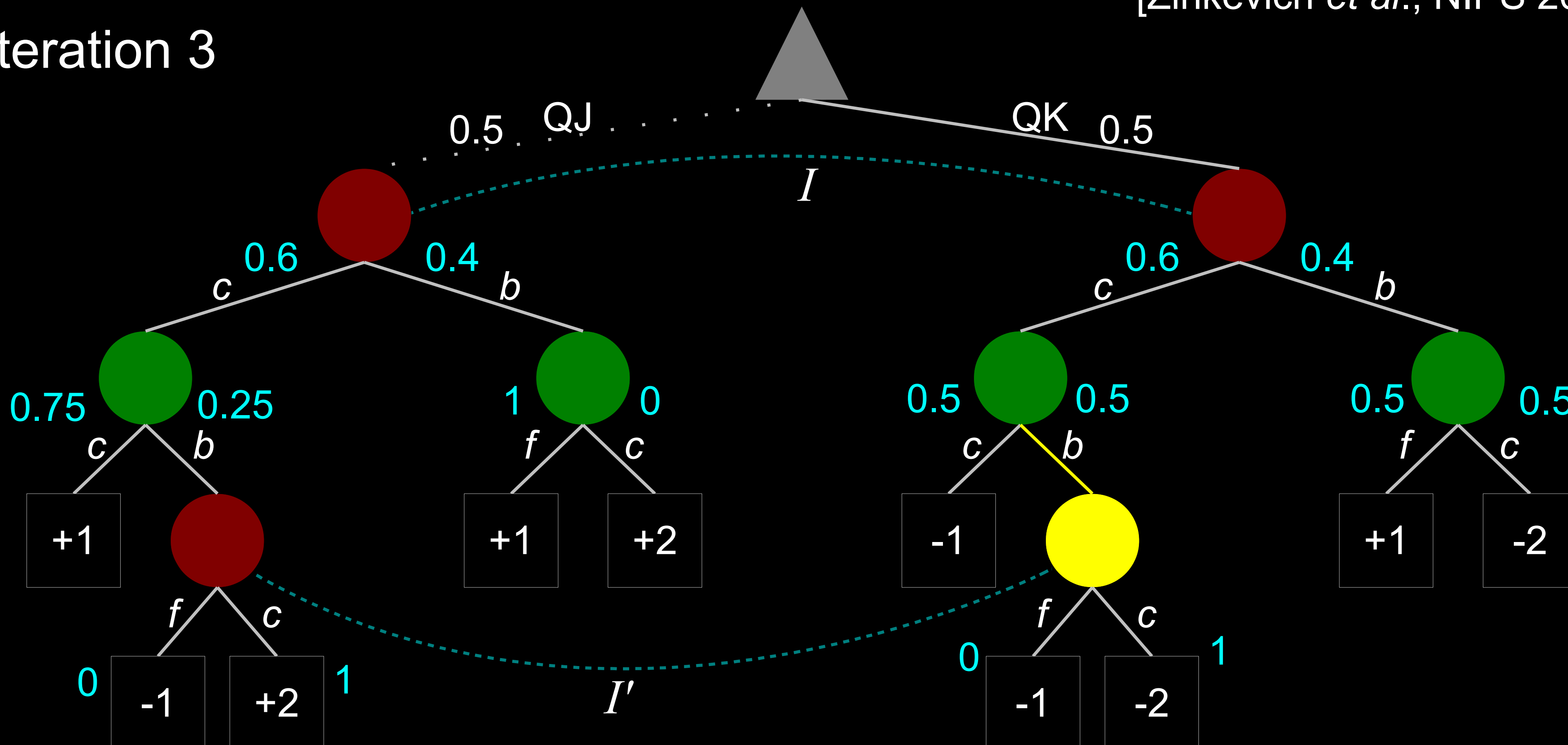


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 3

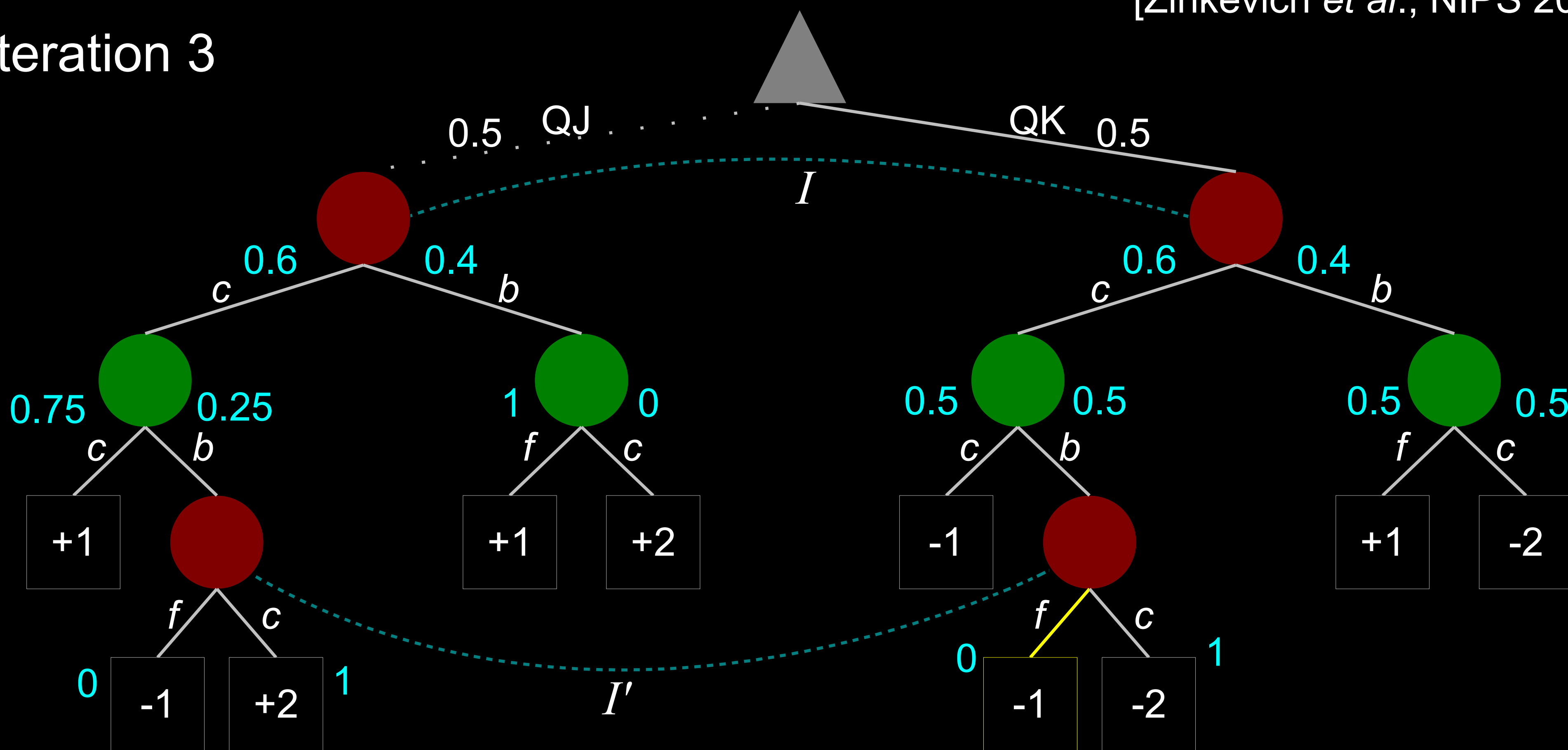


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 3

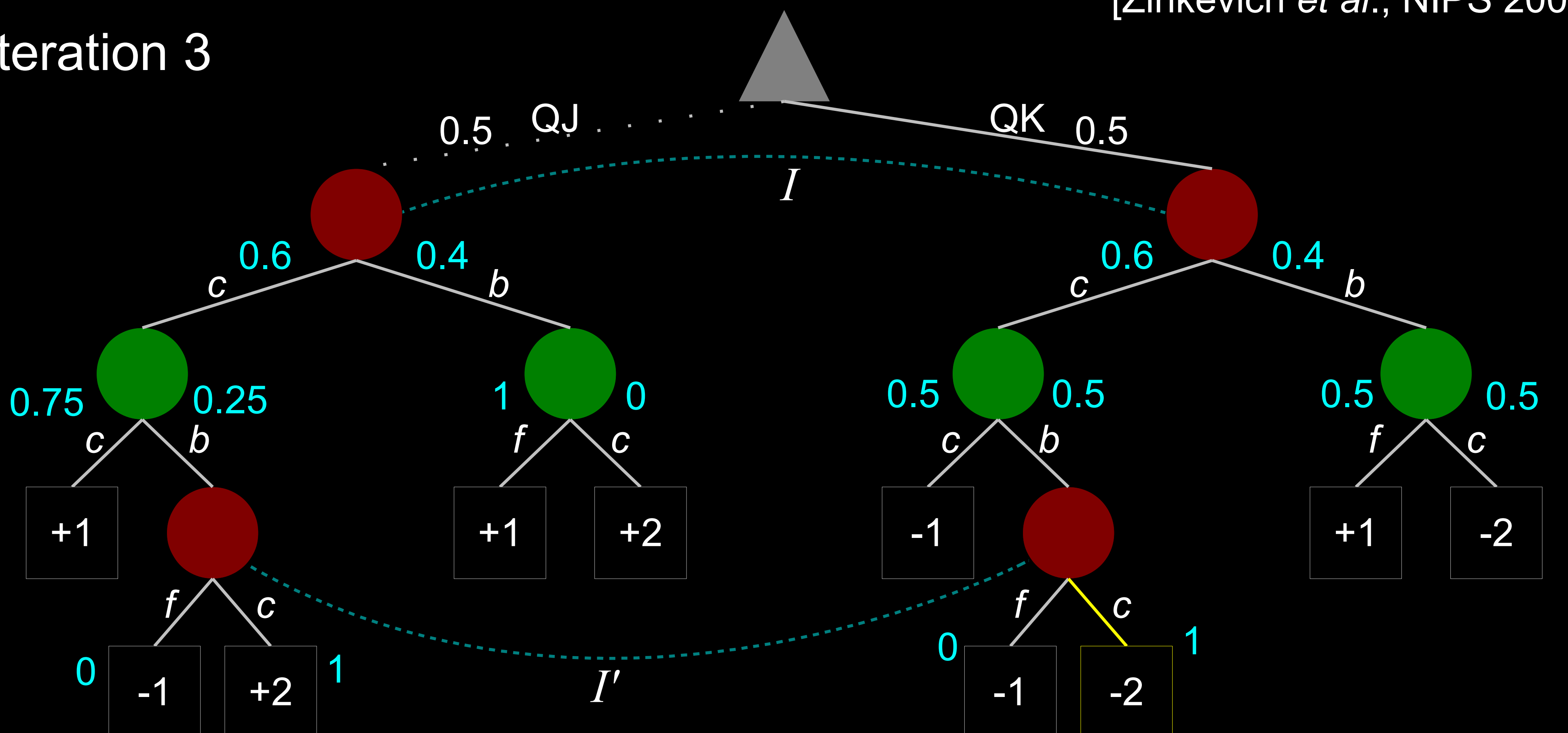


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al*., NIPS 2007]

- Iteration 3

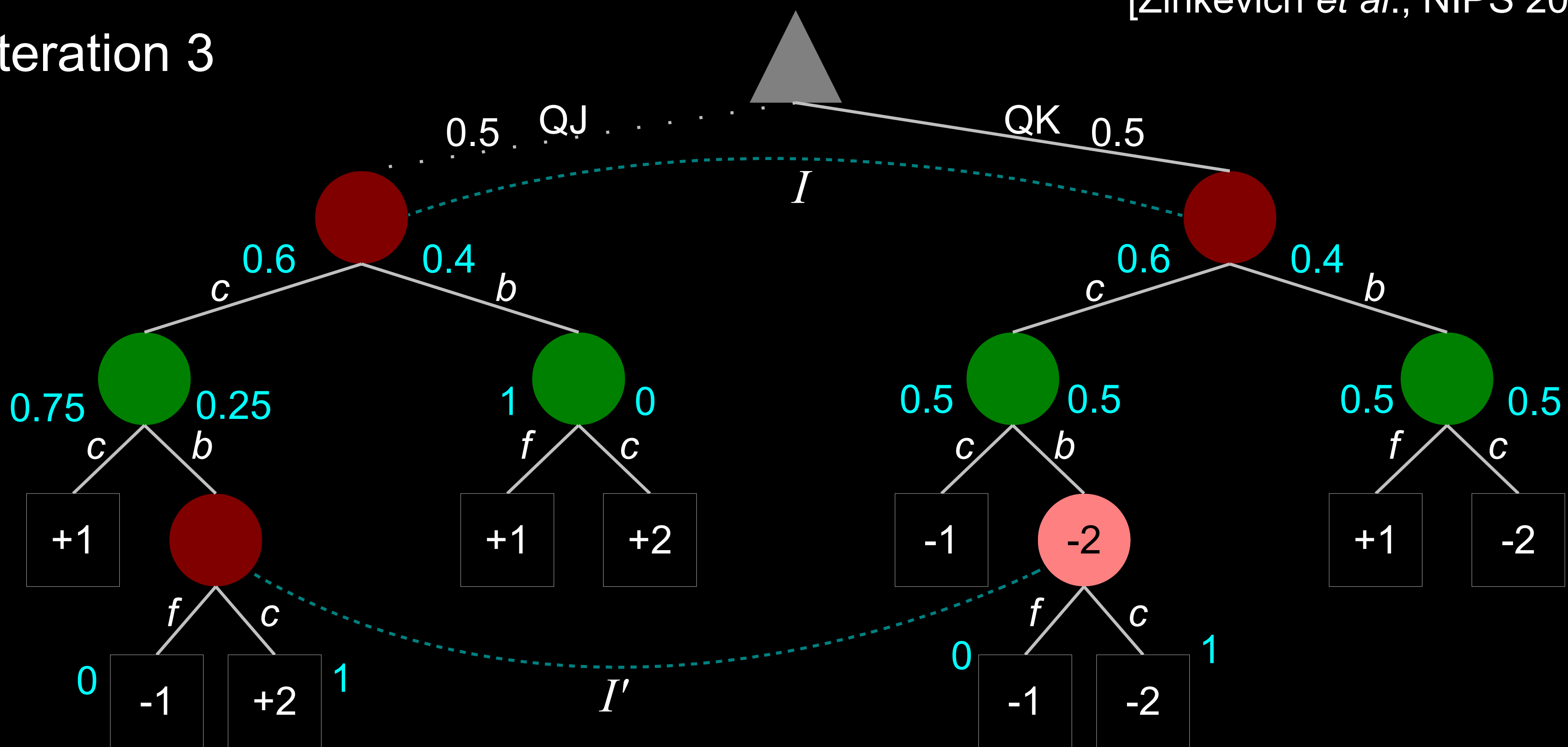


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 3

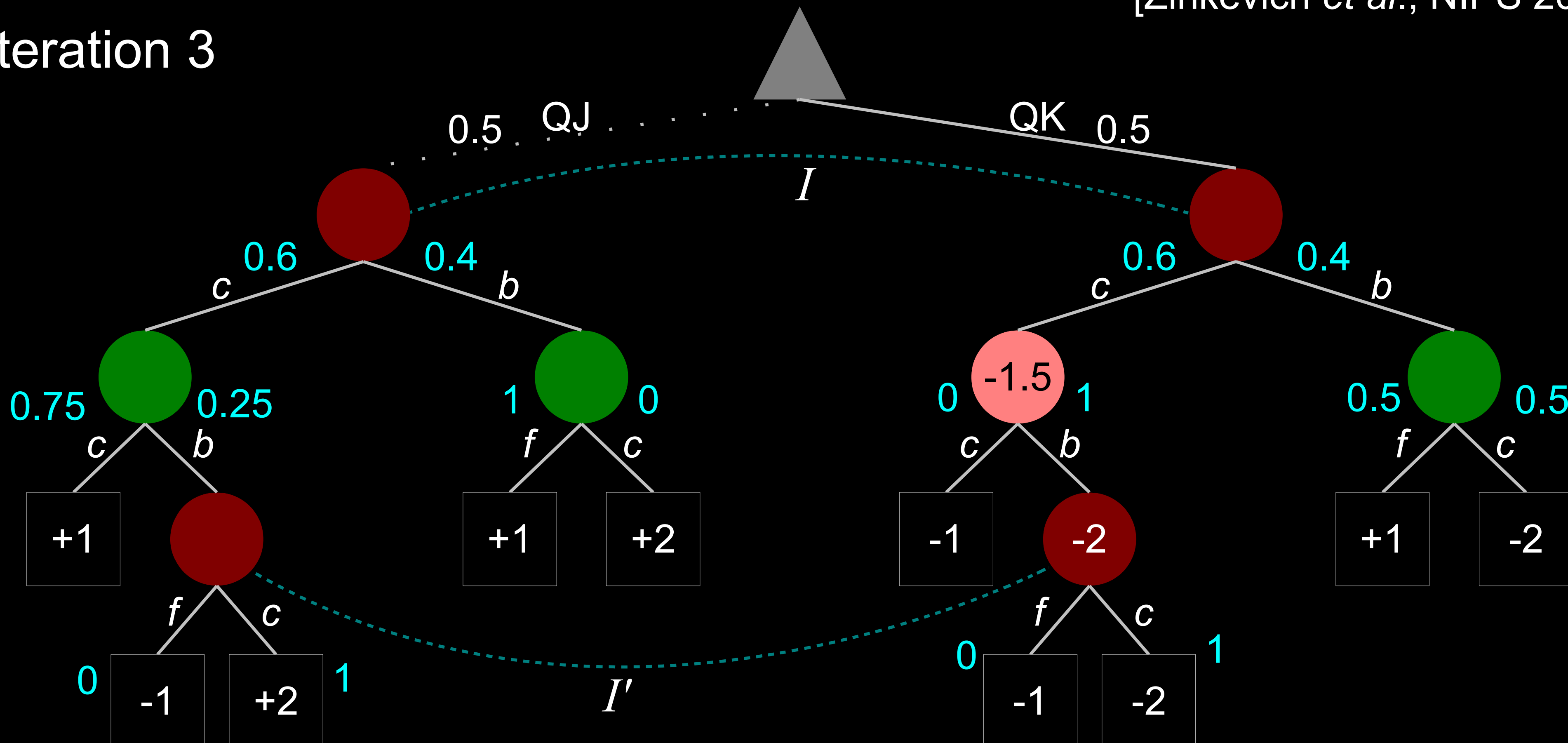


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 3

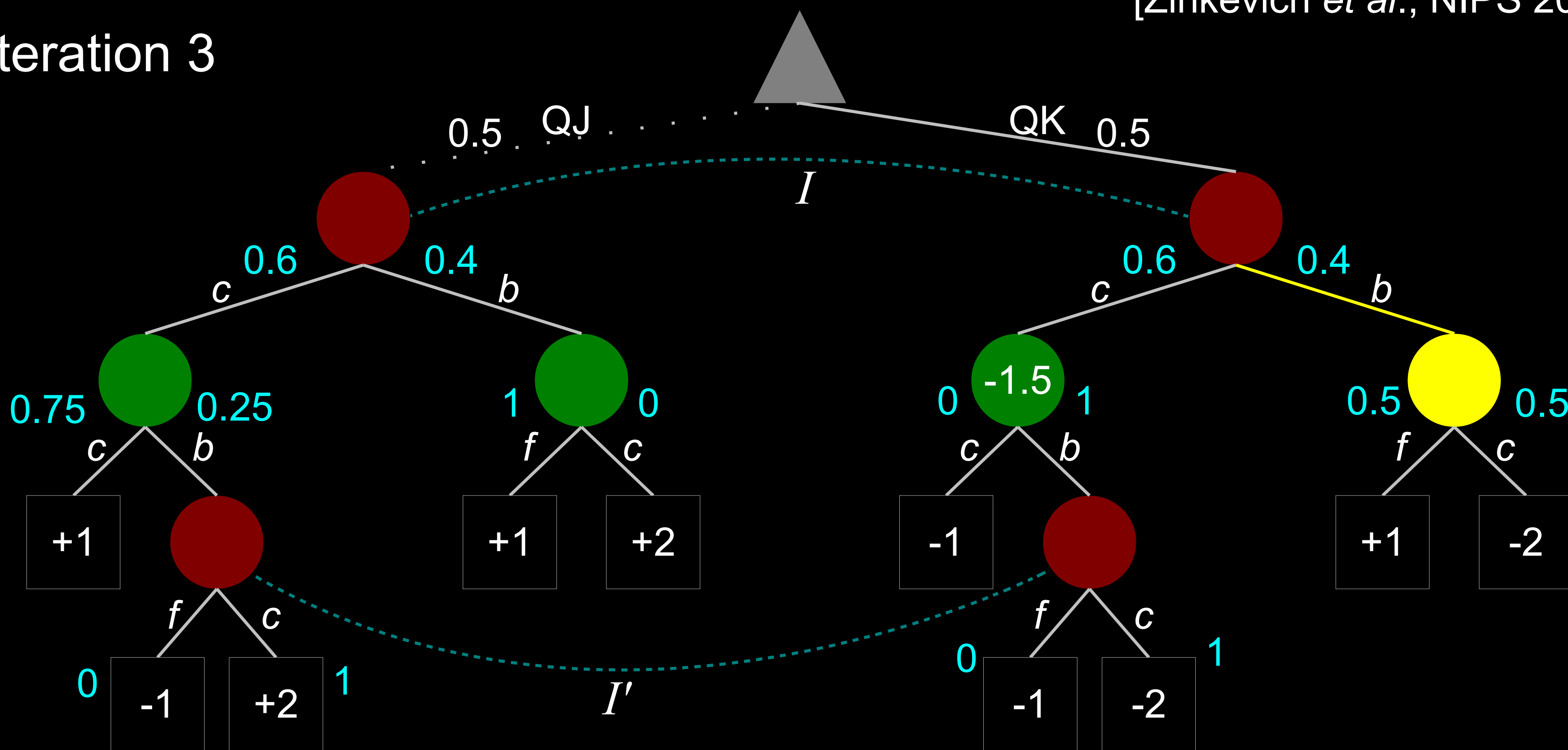


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 3

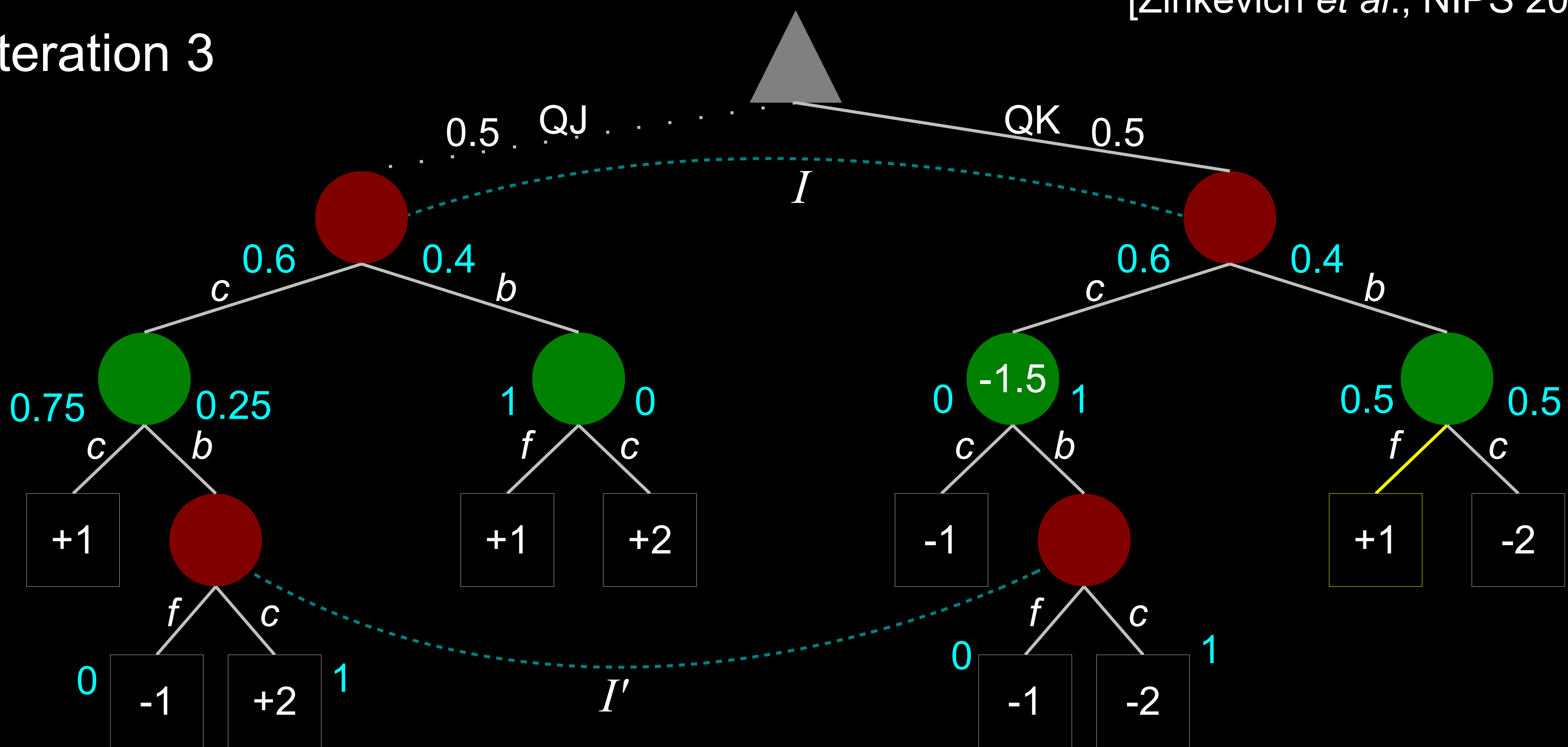


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 3

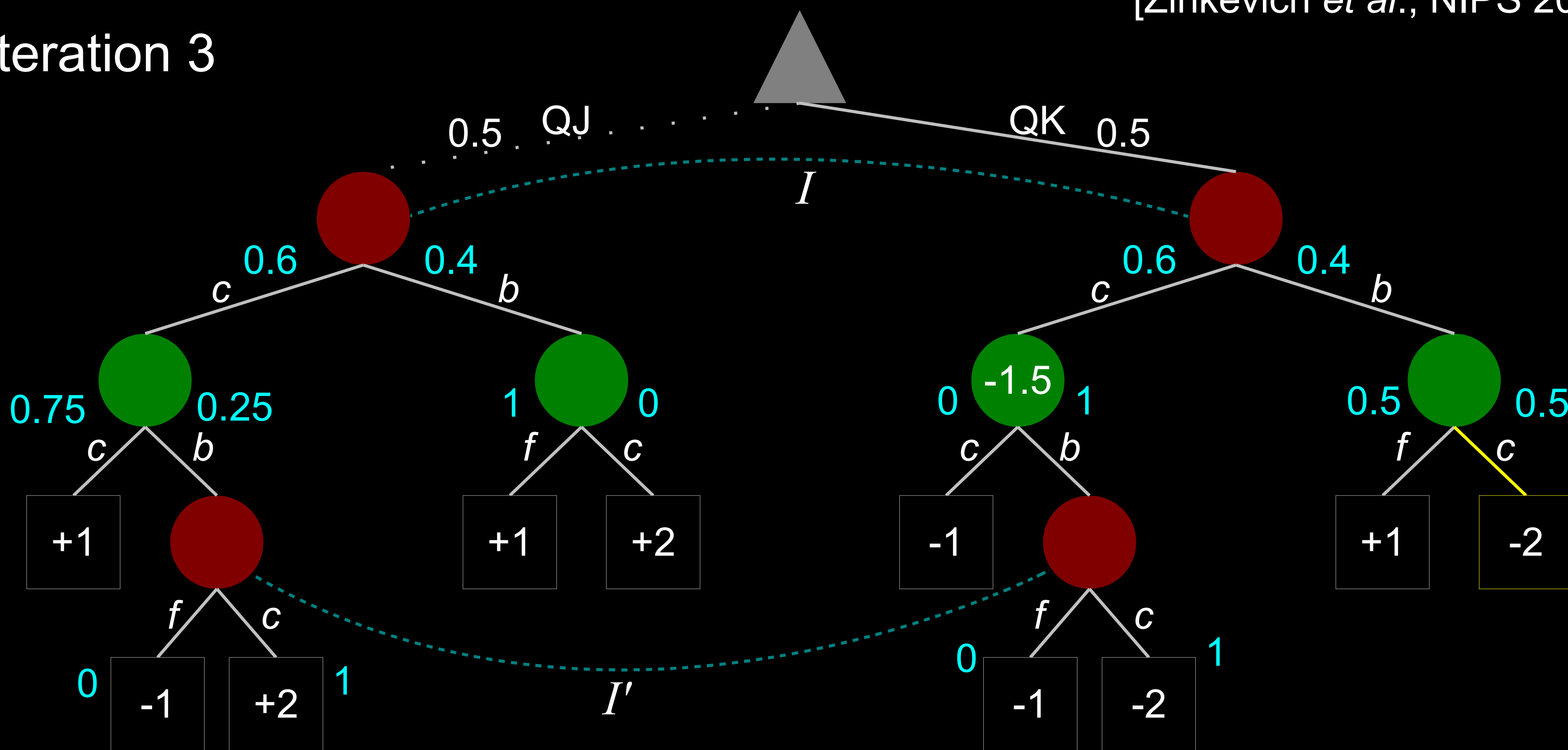


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 3

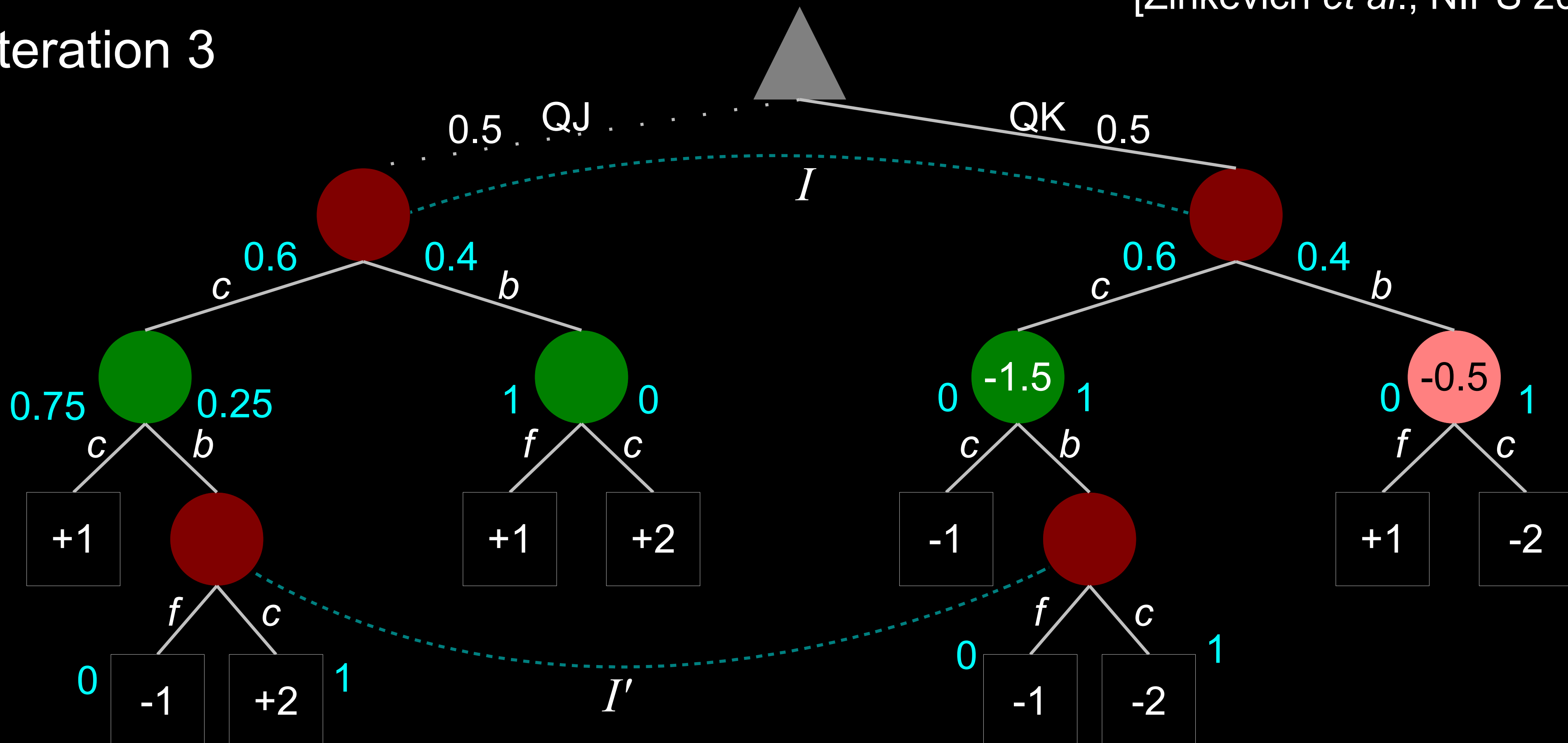


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 3

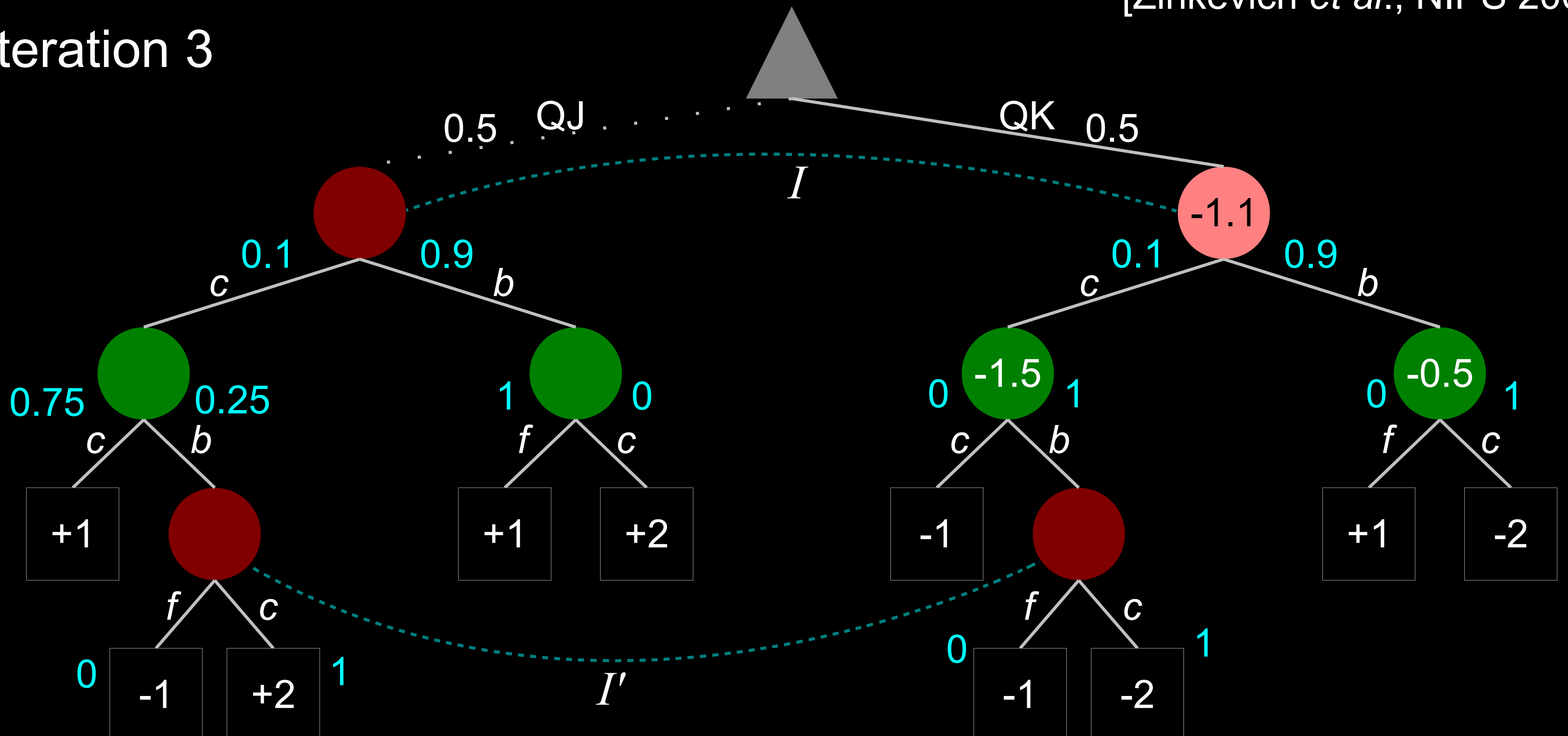


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 3

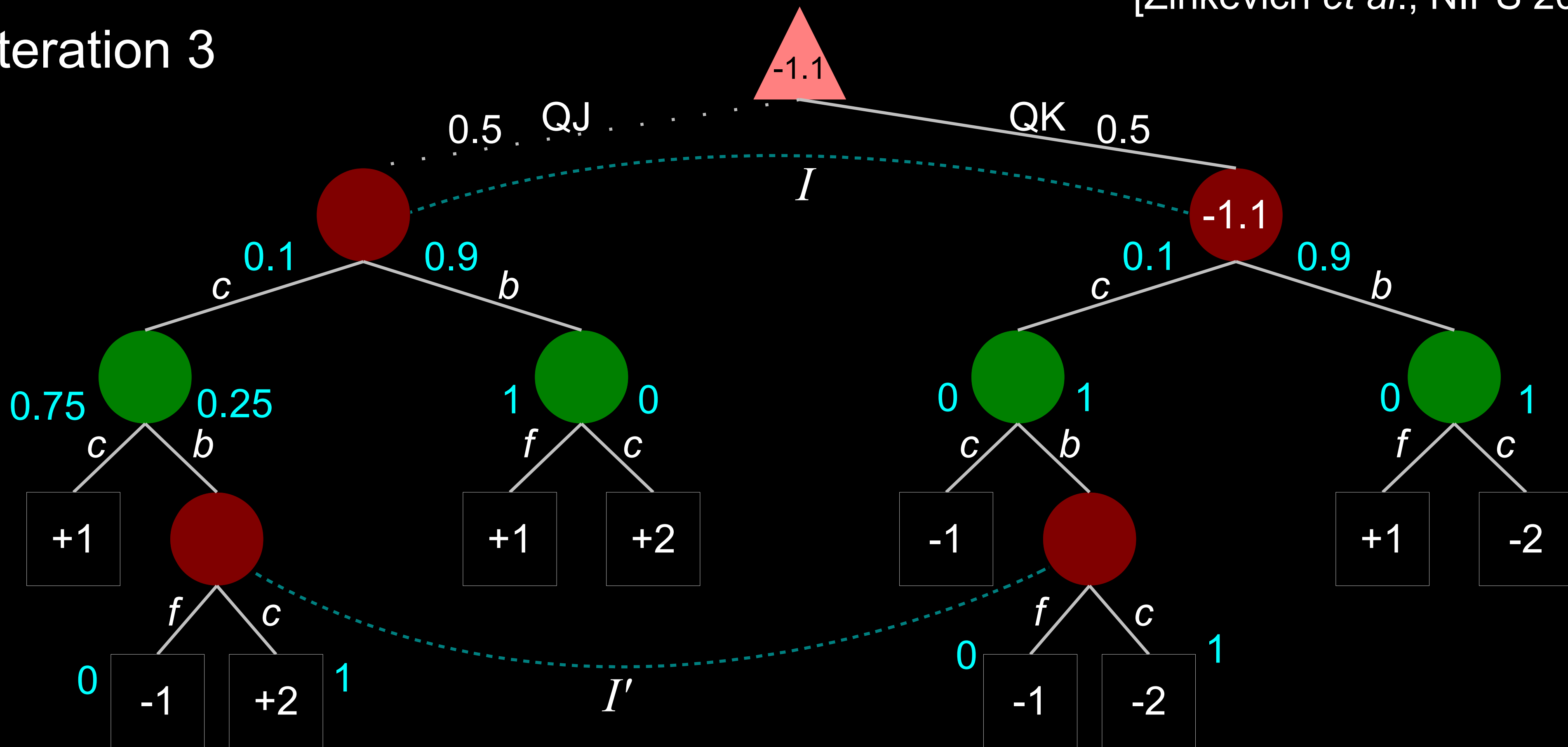


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

[Zinkevich *et al.*, NIPS 2007]

- Iteration 3

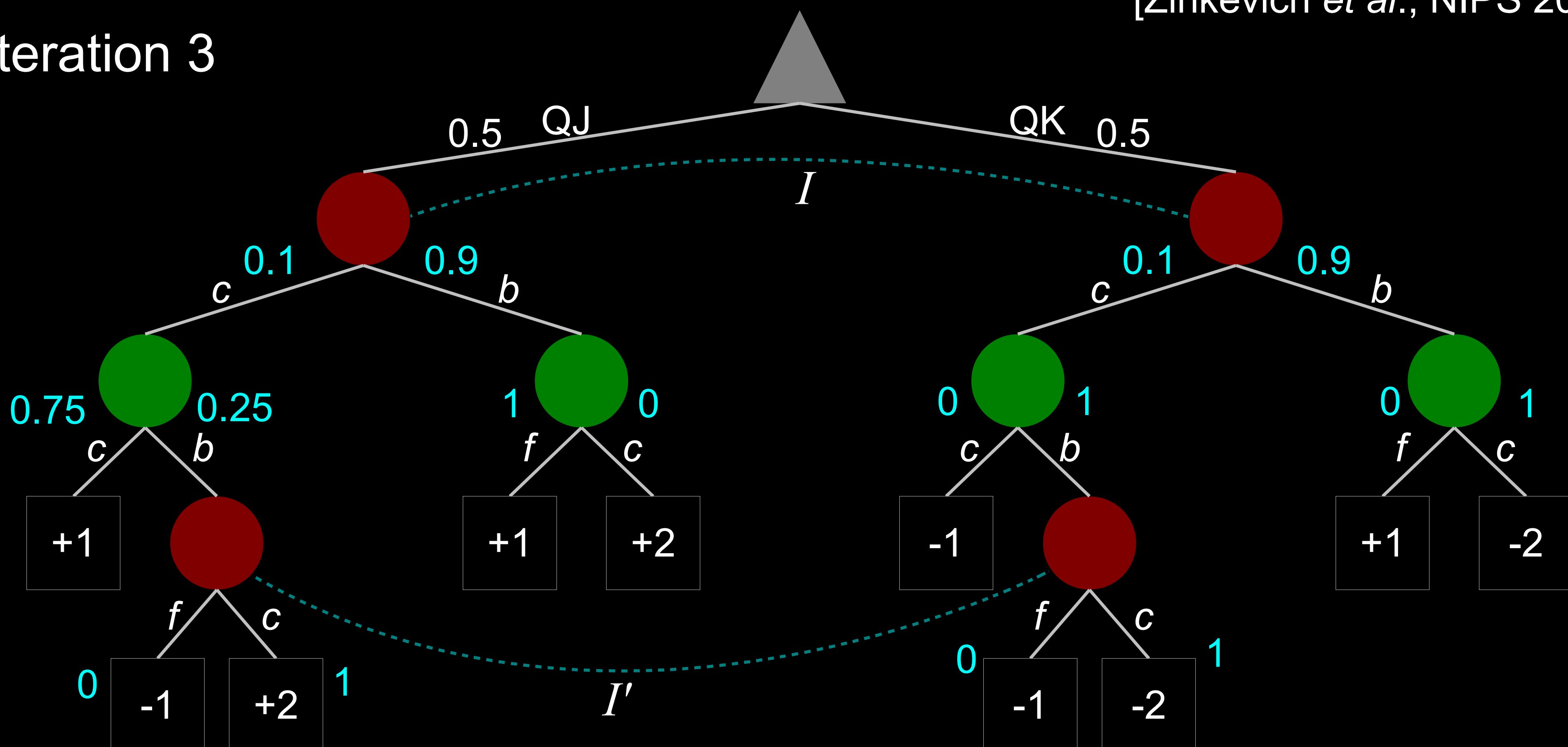


First Variant: Chance Sampling

At each chance node, traverse only one action per iteration

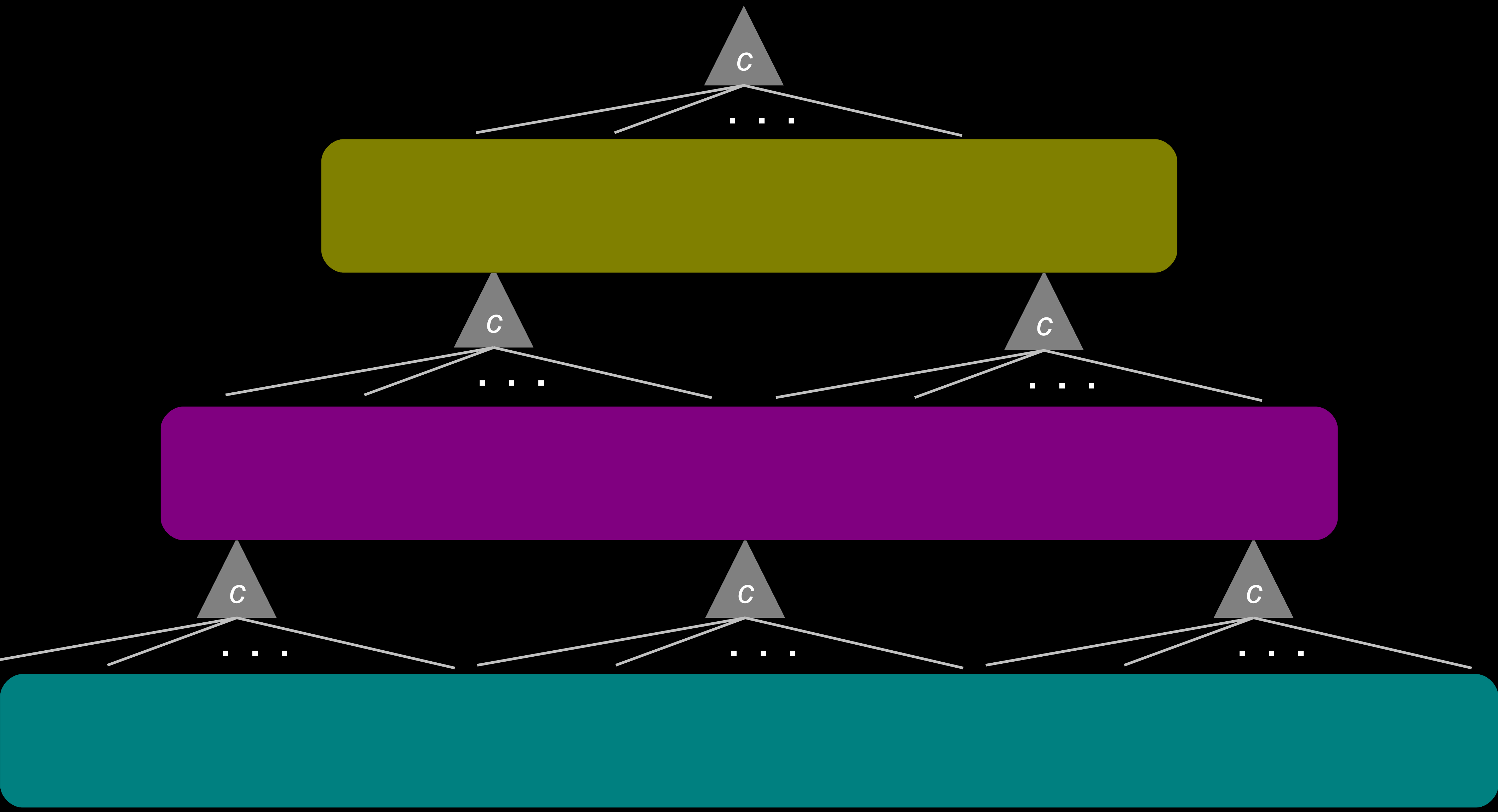
[Zinkevich *et al.*, NIPS 2007]

- Iteration 3



First Variant: Chance Sampling

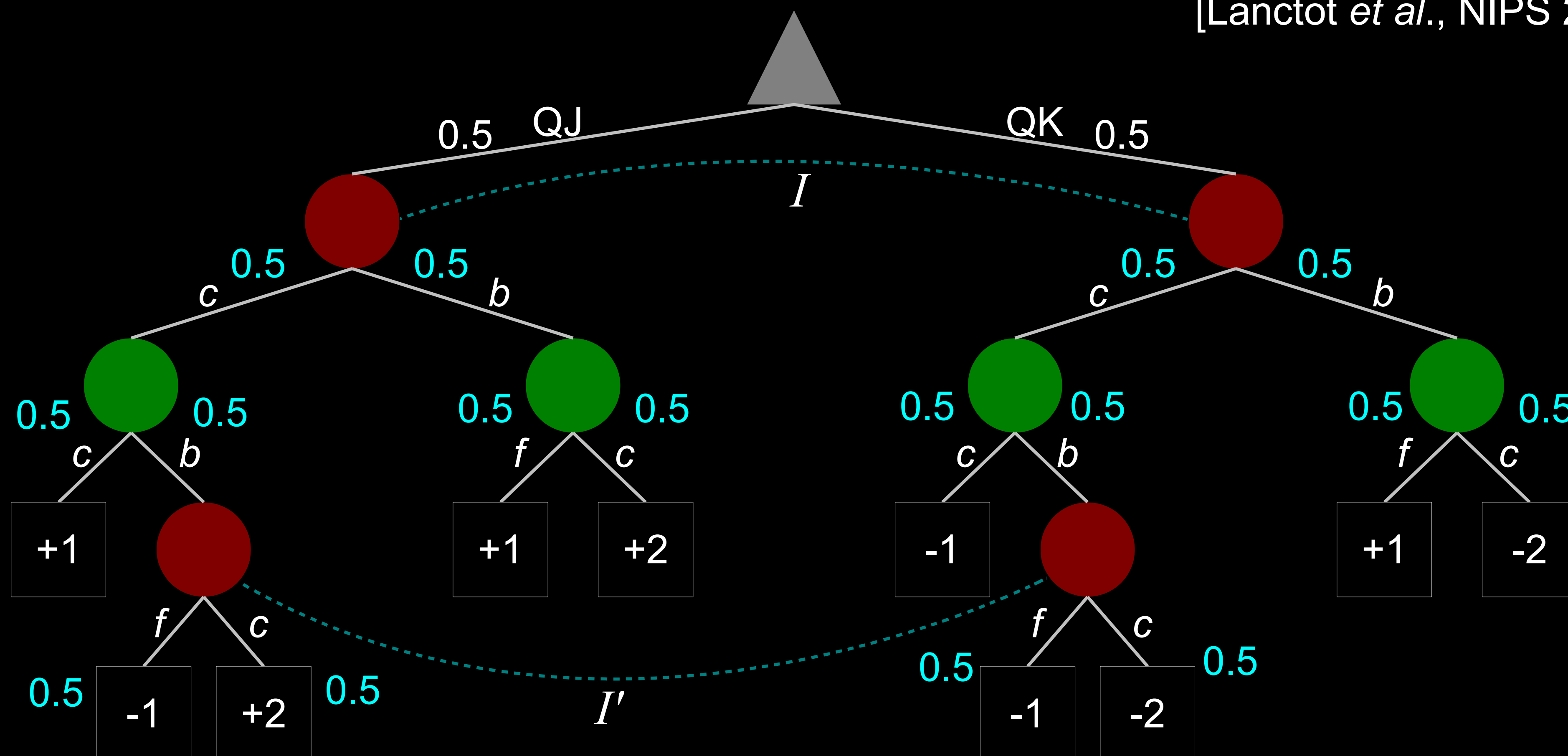
At each chance node, traverse only one action per iteration



Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al*., NIPS 2009]

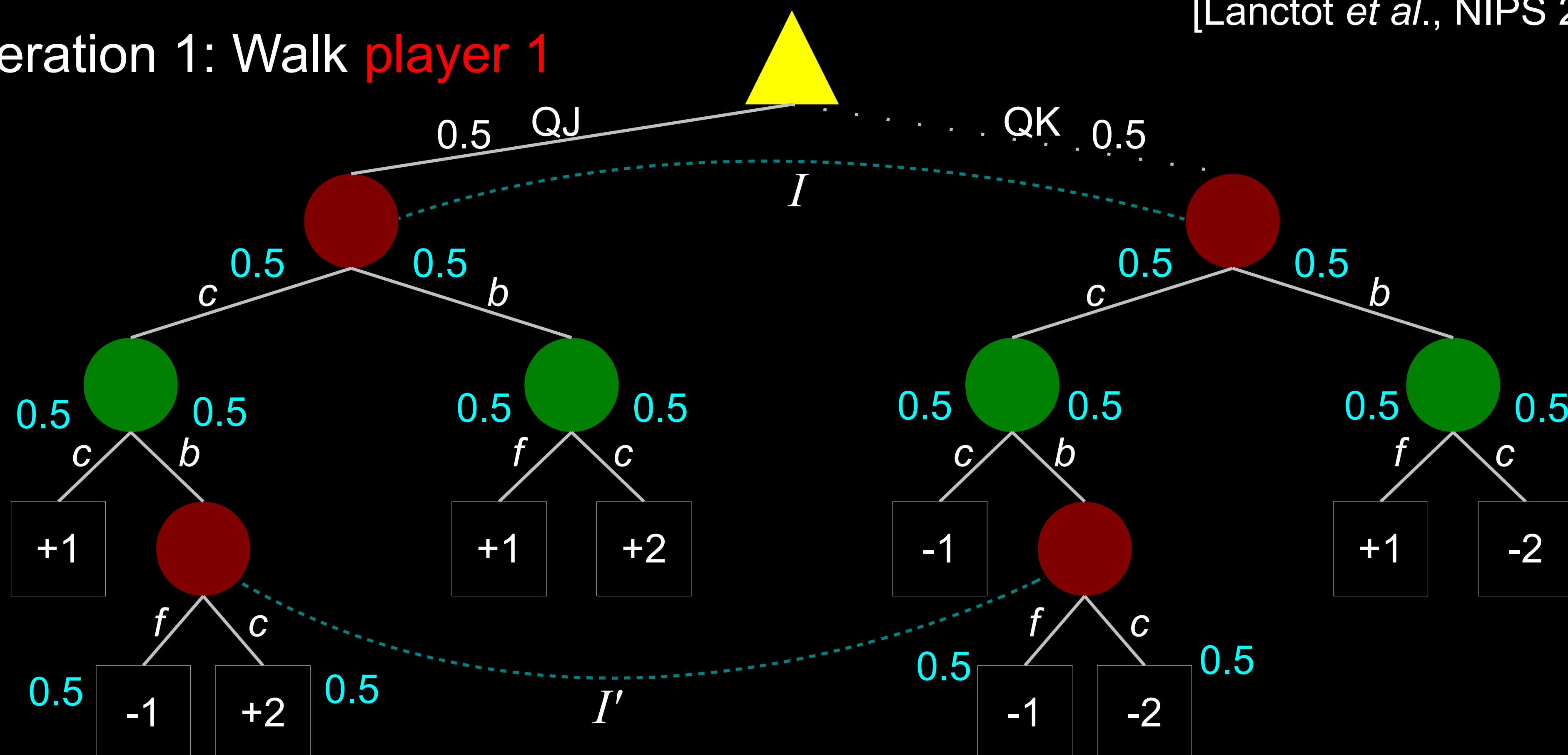


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 1

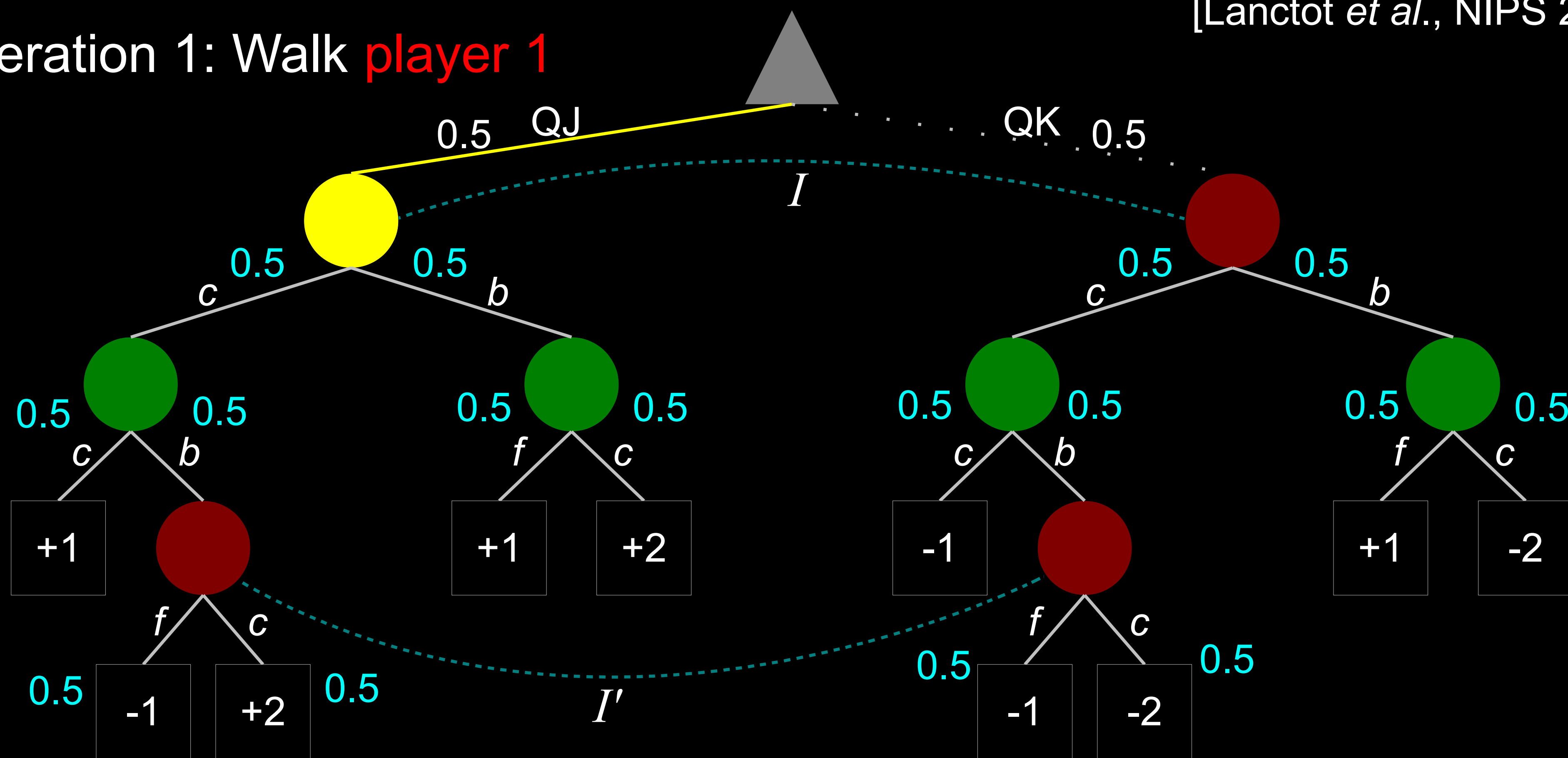


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al*., NIPS 2009]

- Iteration 1: Walk player 1

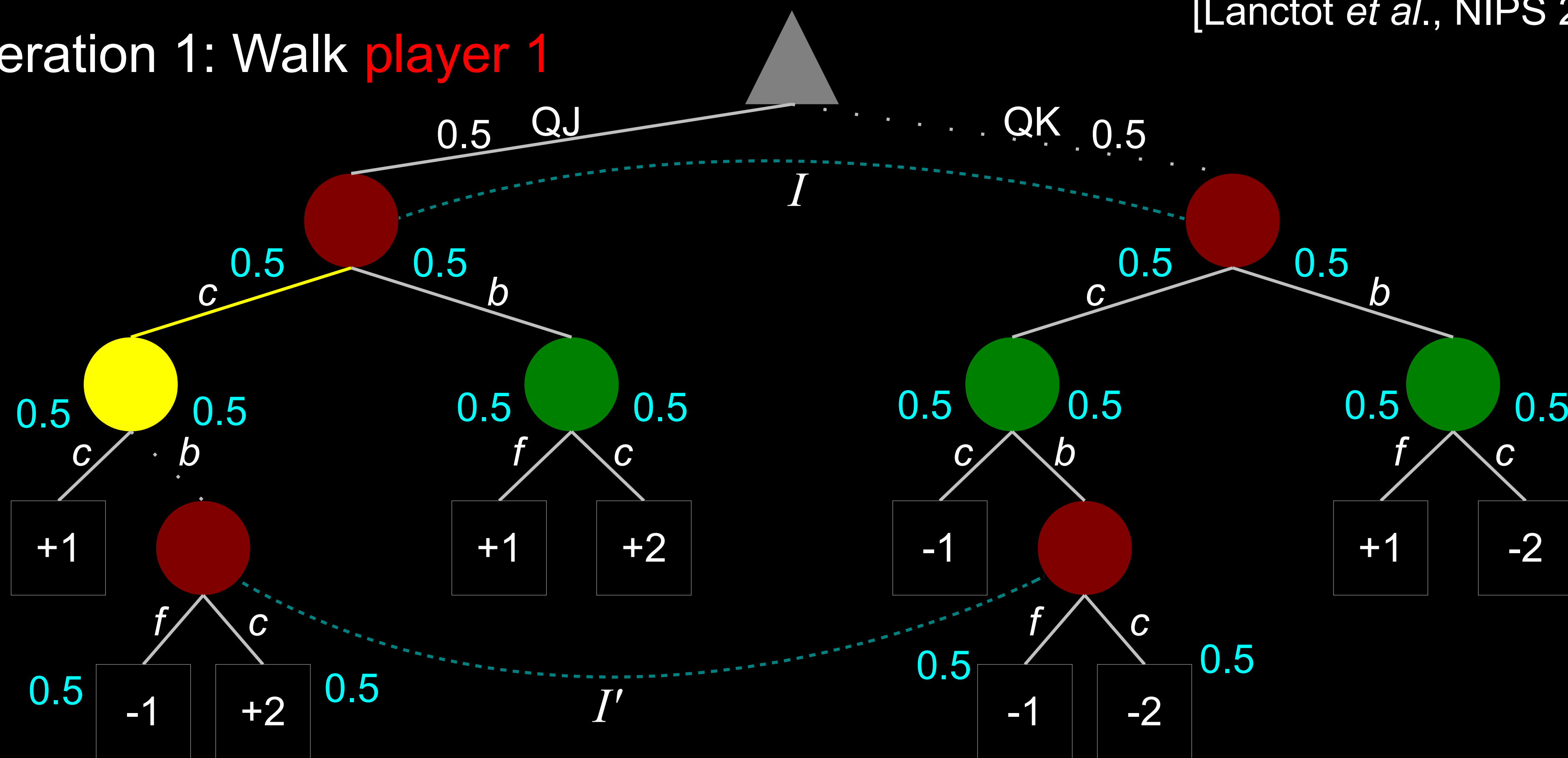


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 1

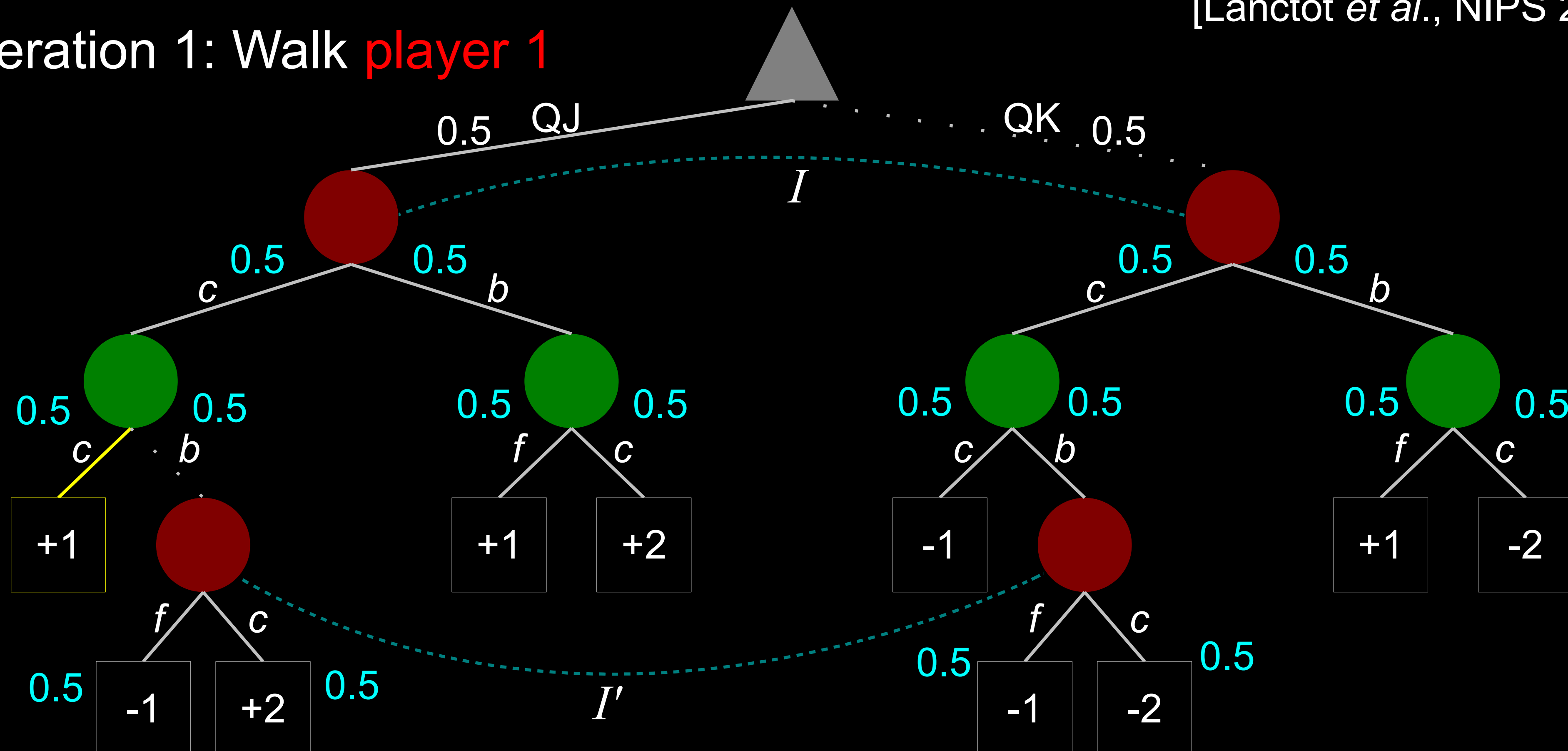


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 1

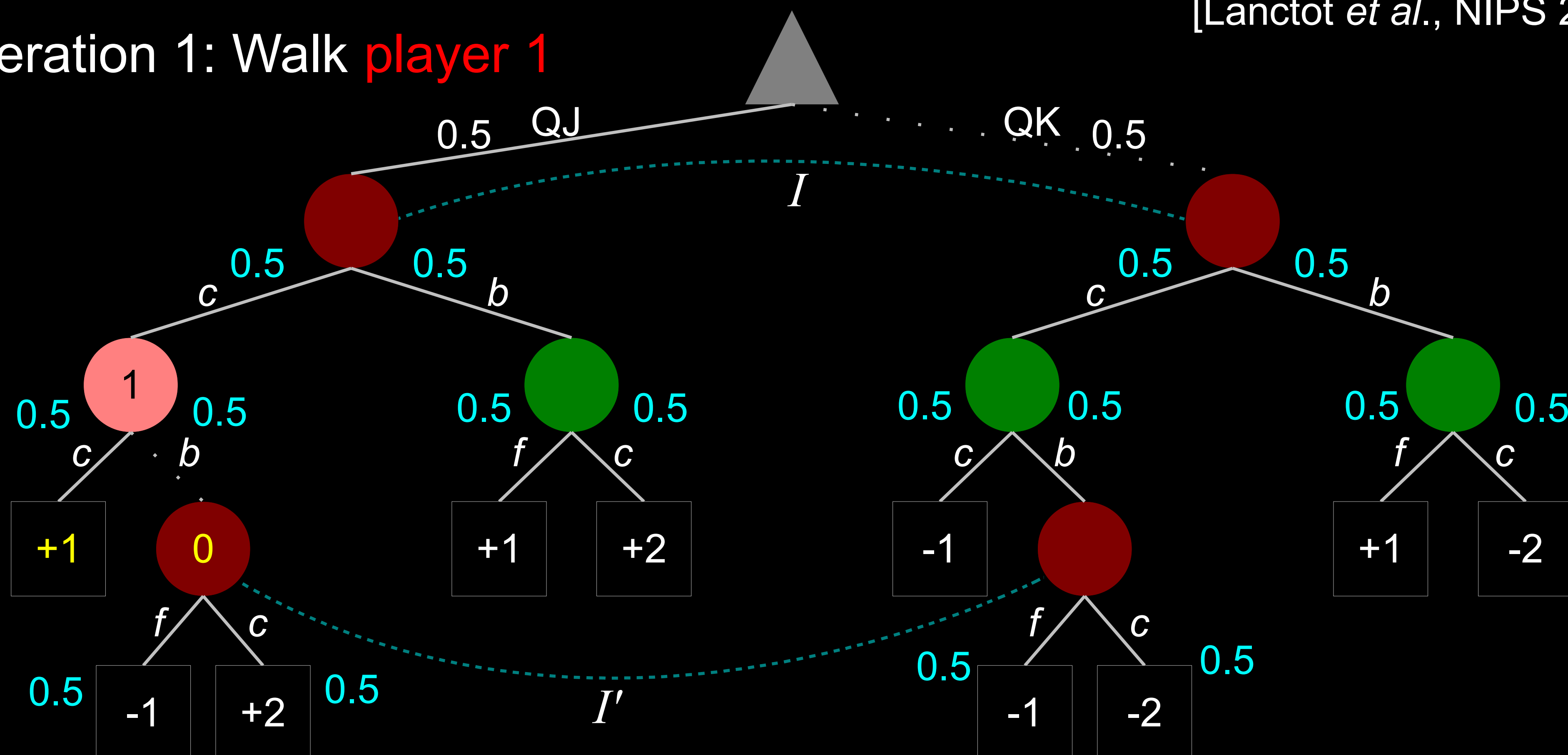


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 1



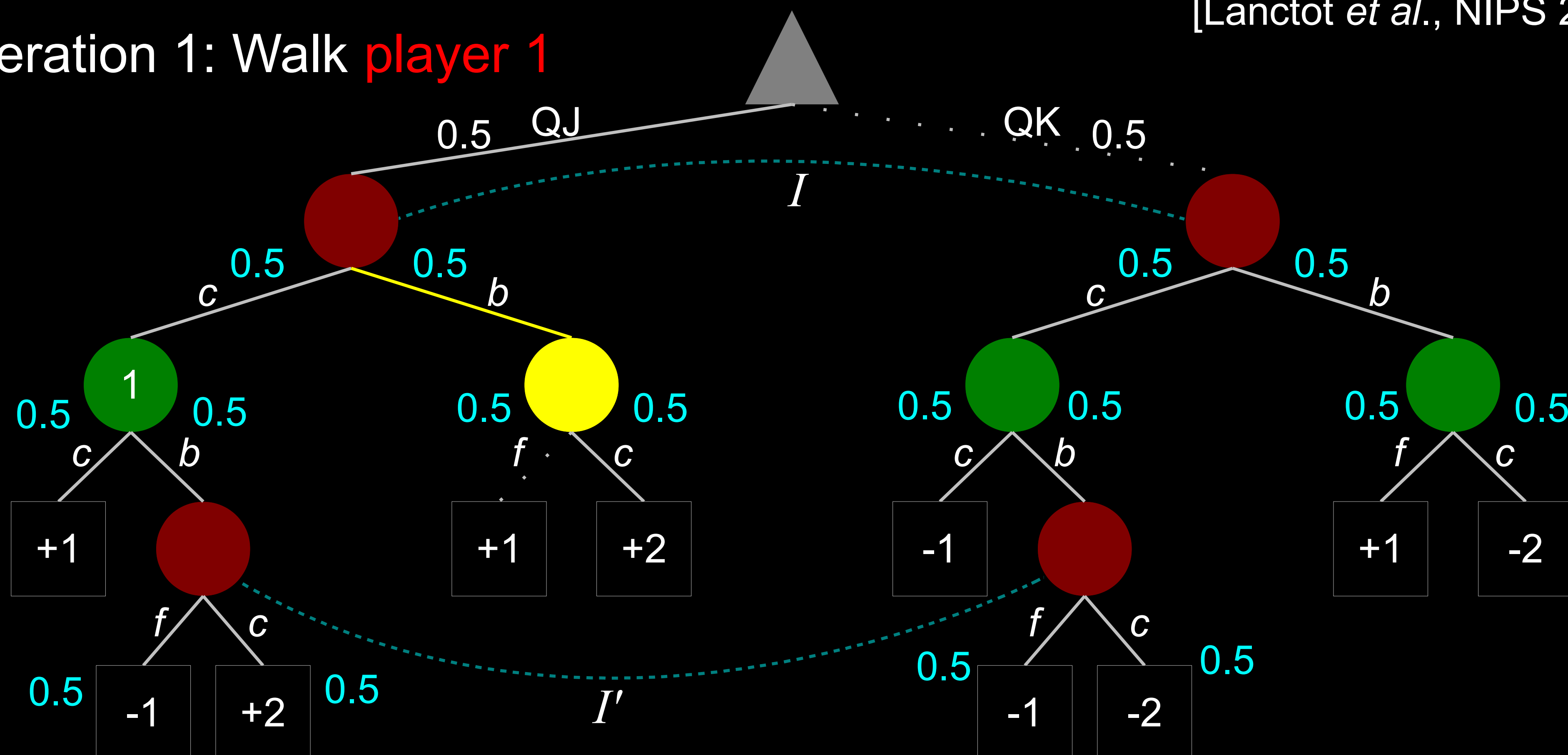
EV = 0.5(+1) + 0.5(0) / probability of sampling *c*

Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 1

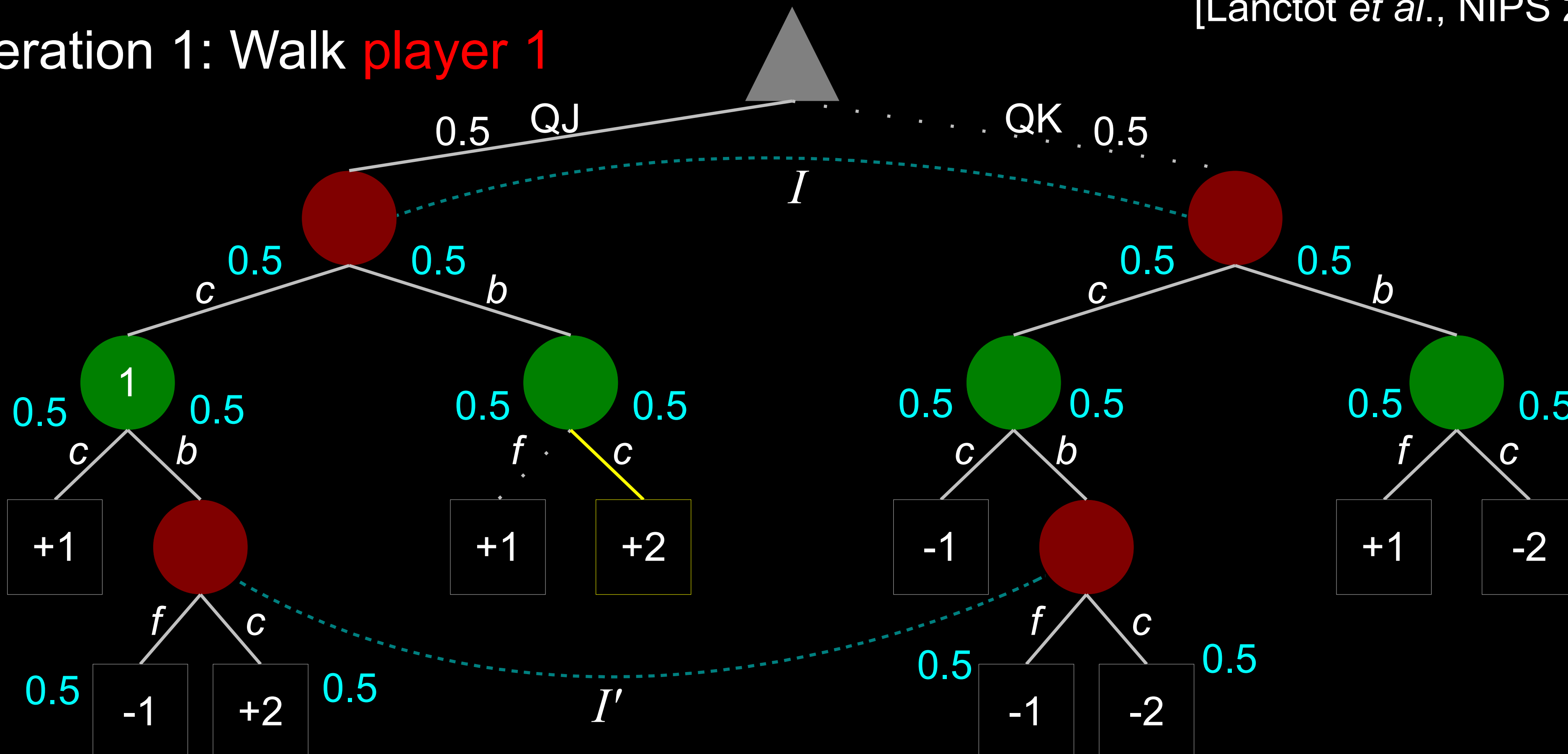


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 1

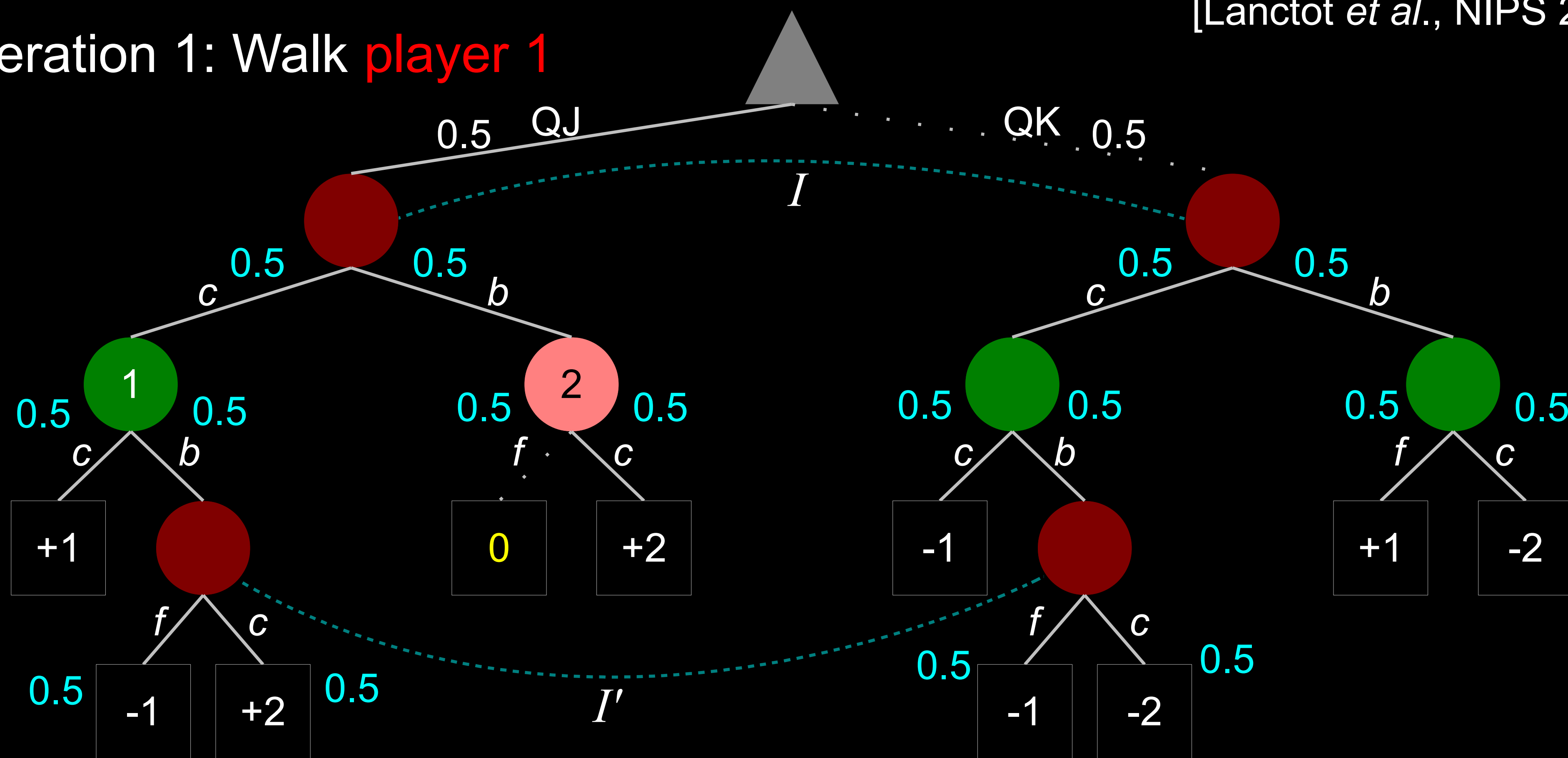


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 1

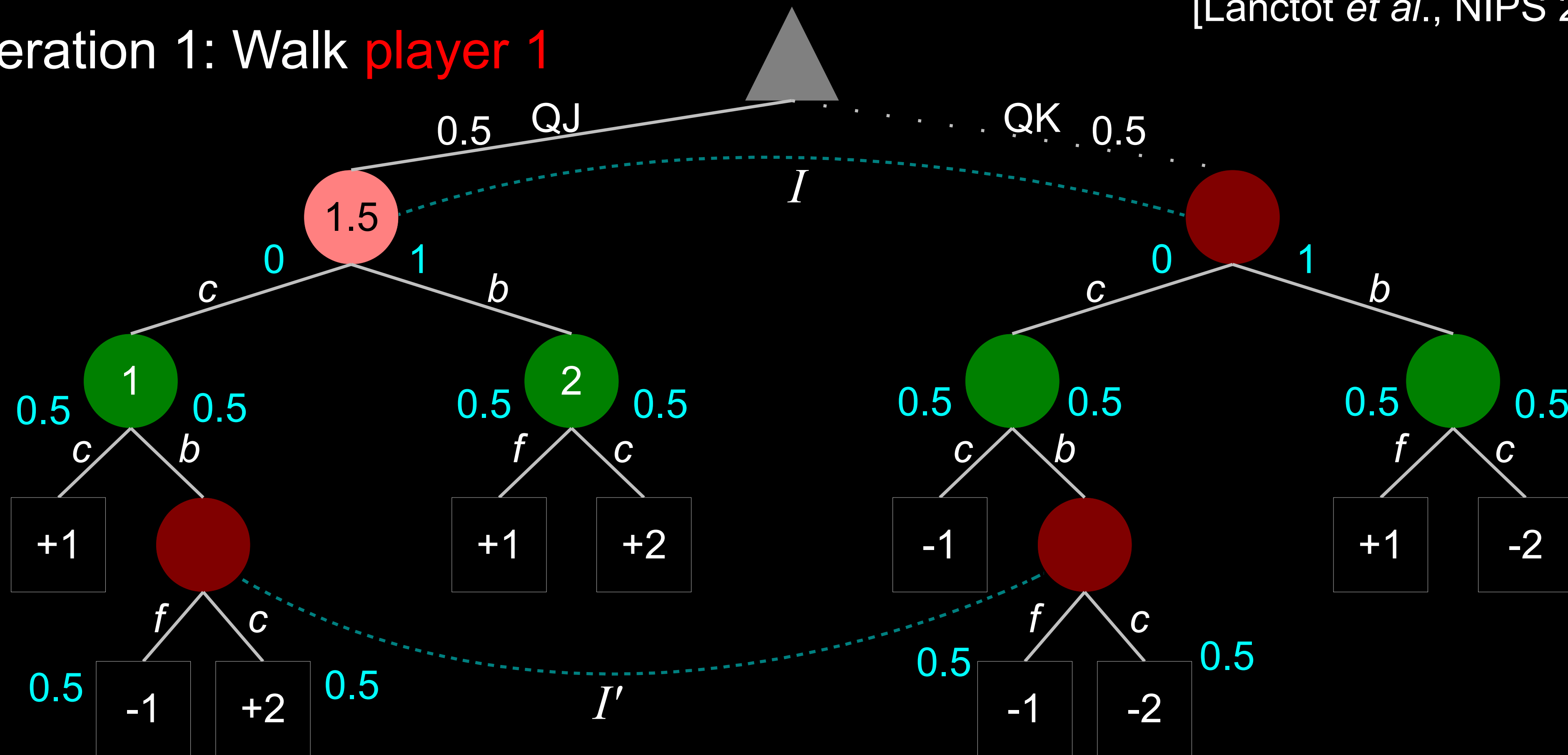


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 1

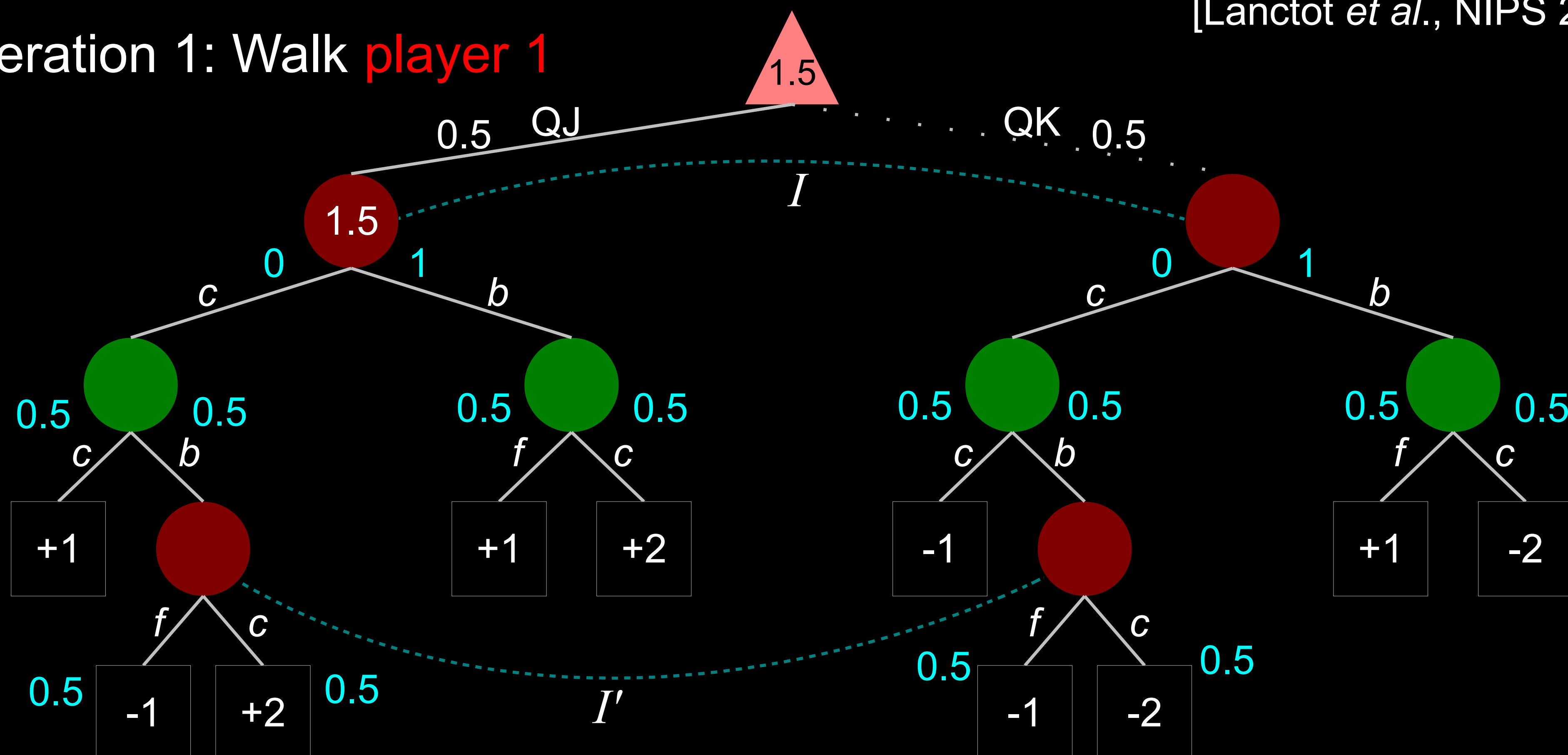


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 1

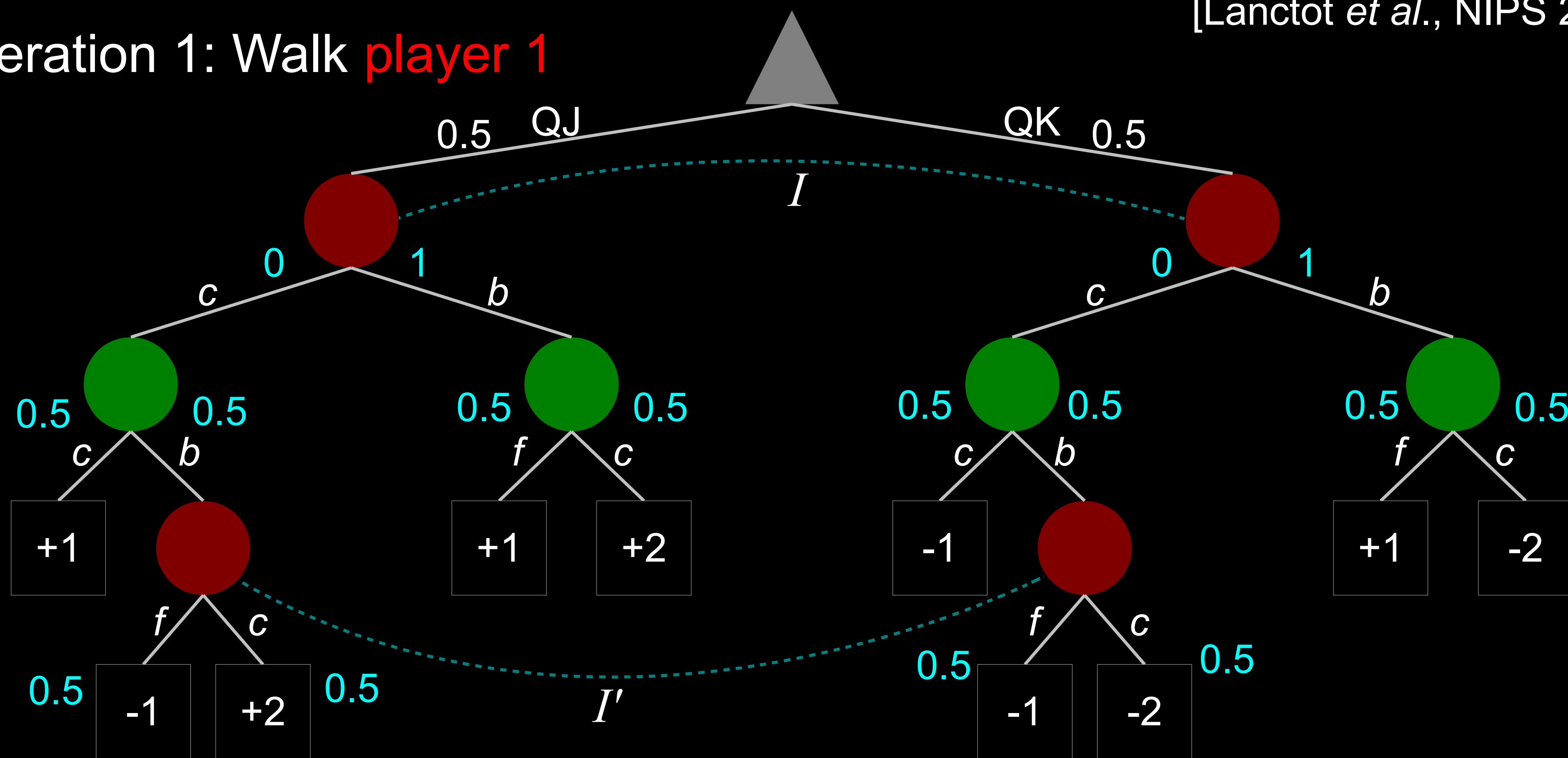


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 1

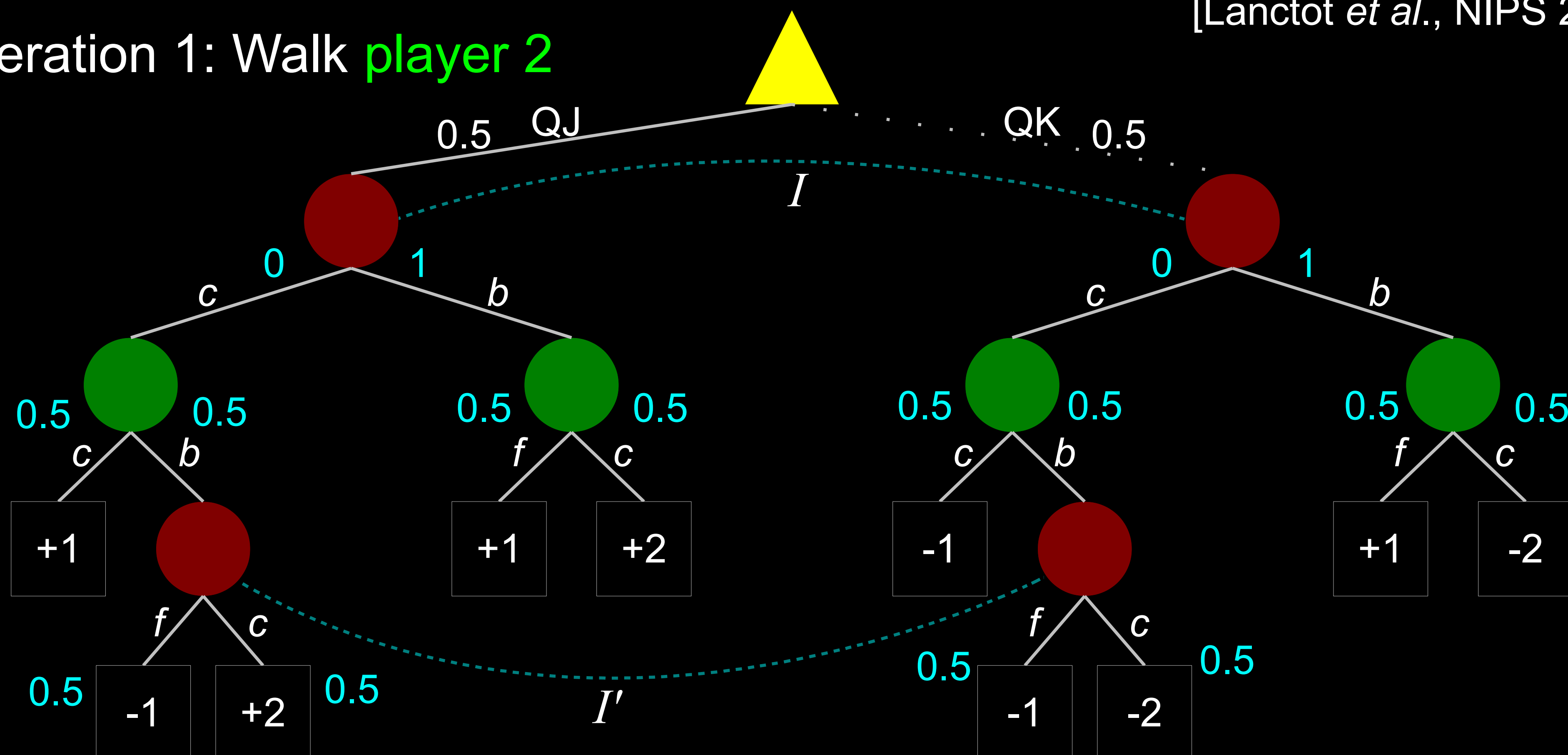


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al*., NIPS 2009]

- Iteration 1: Walk player 2

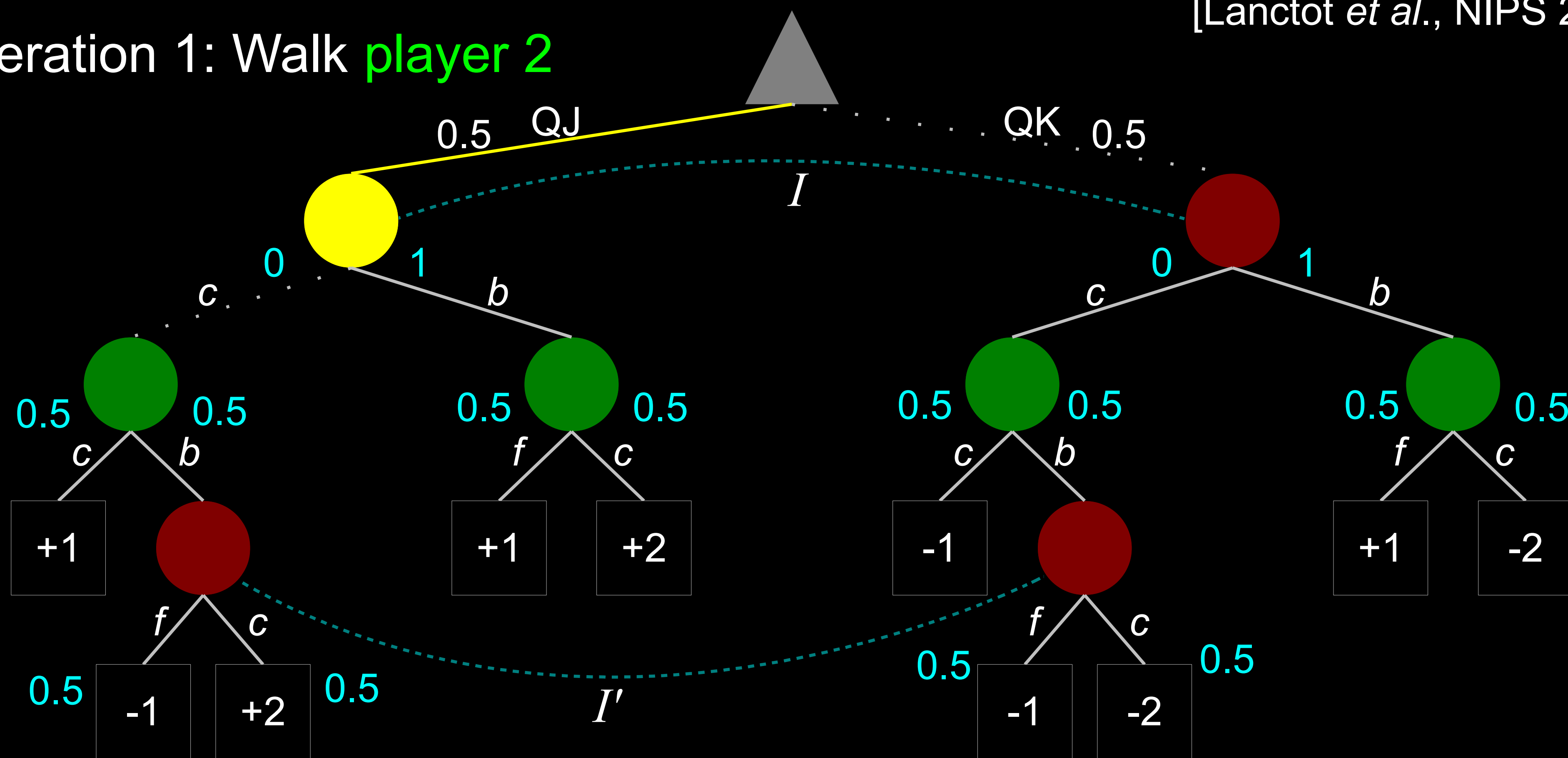


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 2

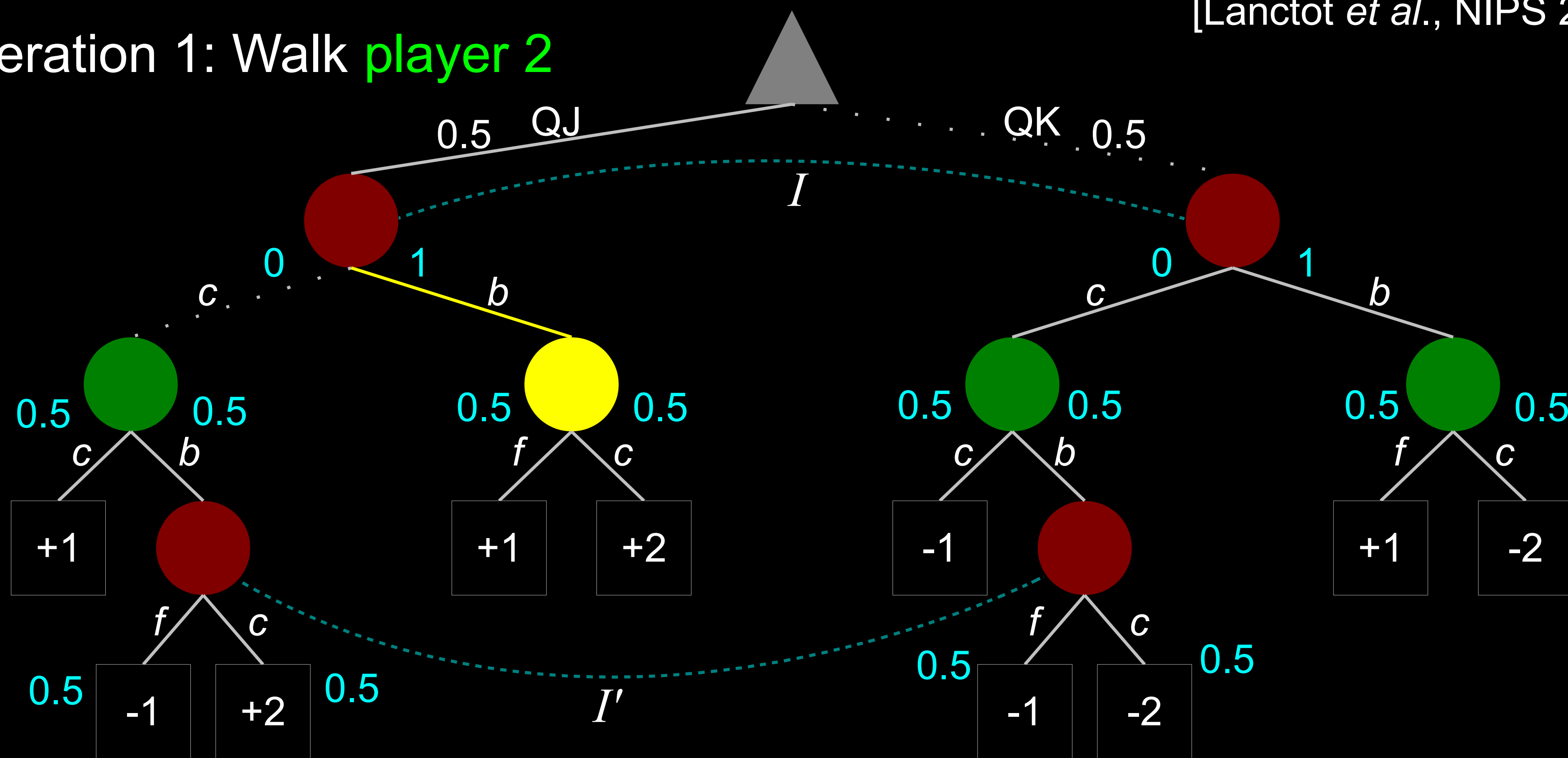


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 2

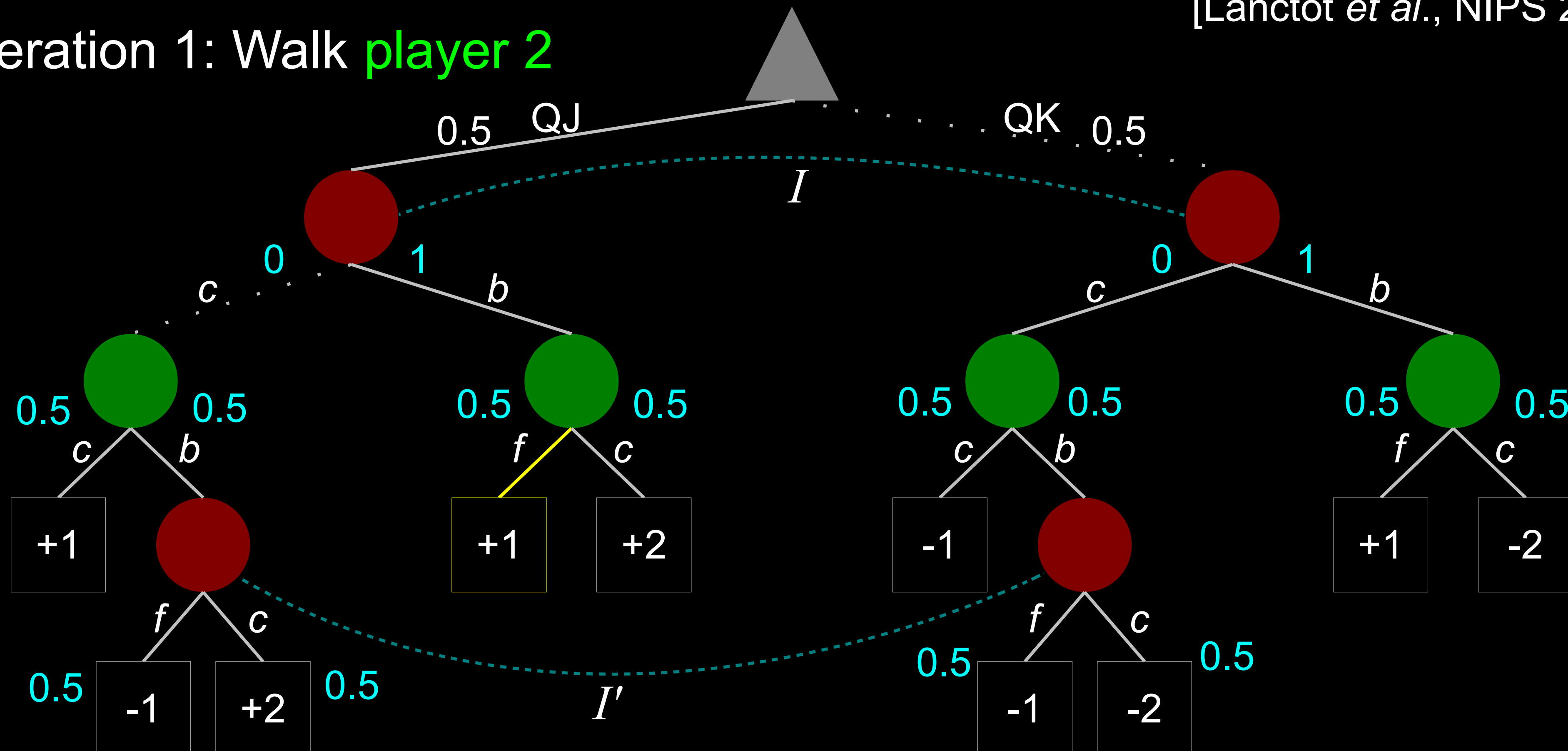


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 2

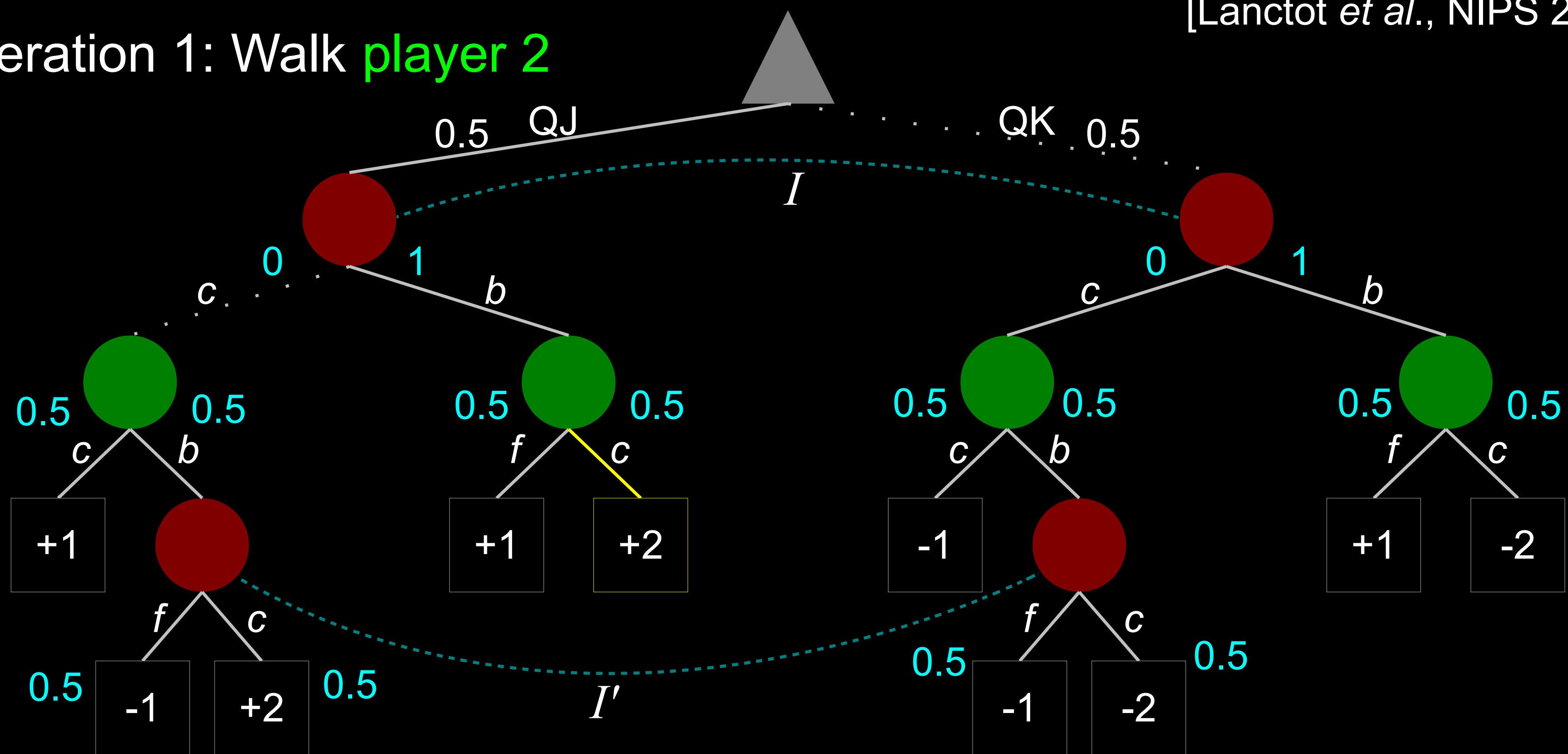


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 2

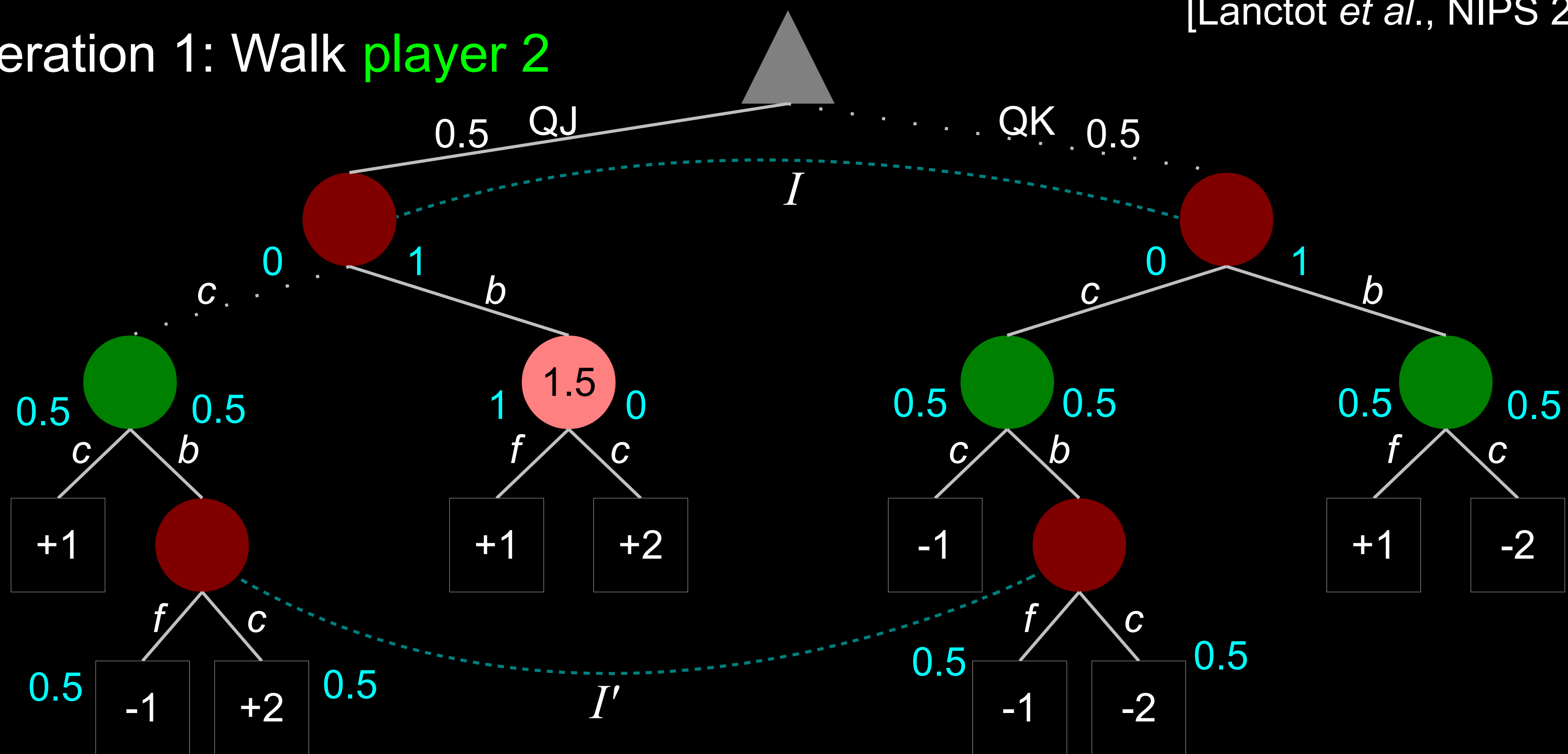


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 2

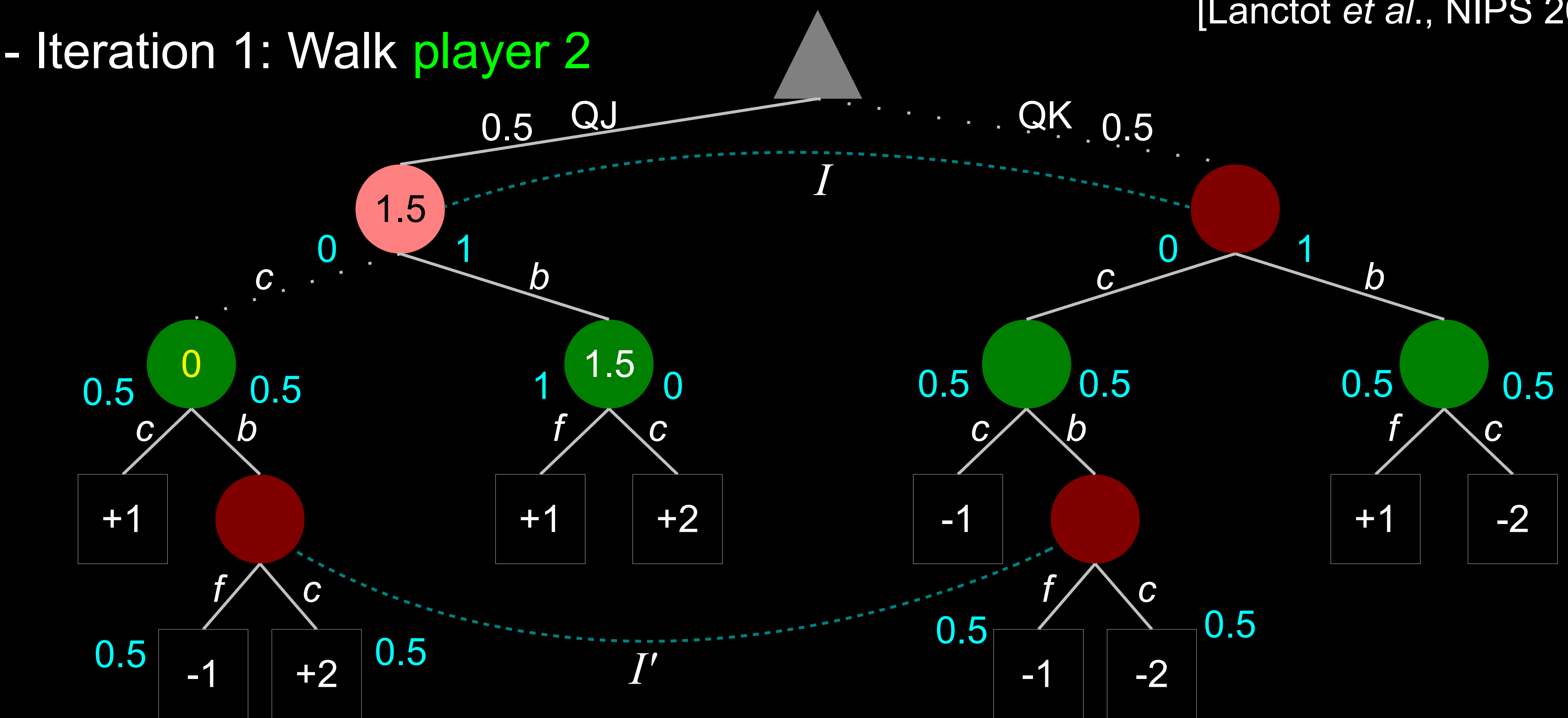


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 2

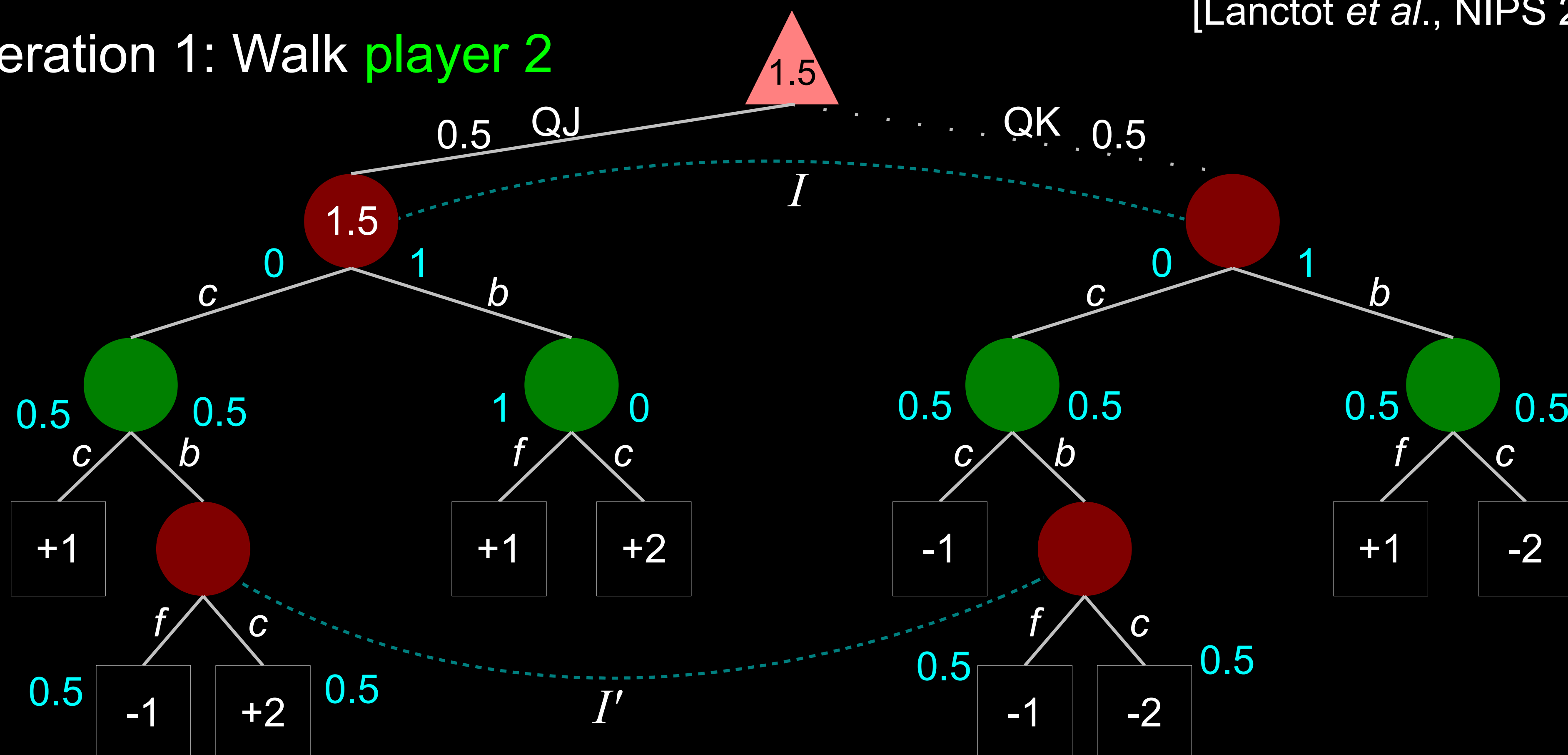


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 2

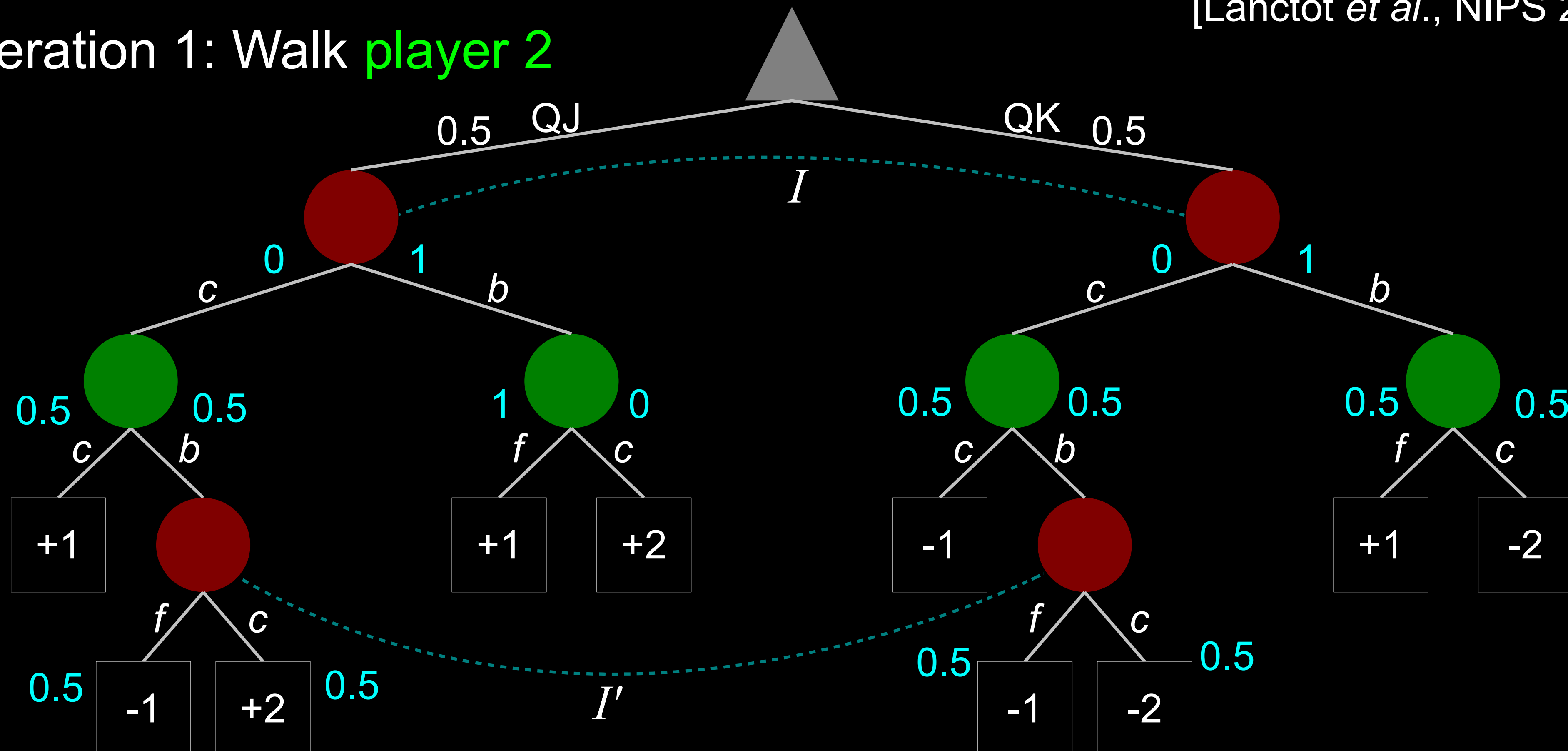


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1: Walk player 2

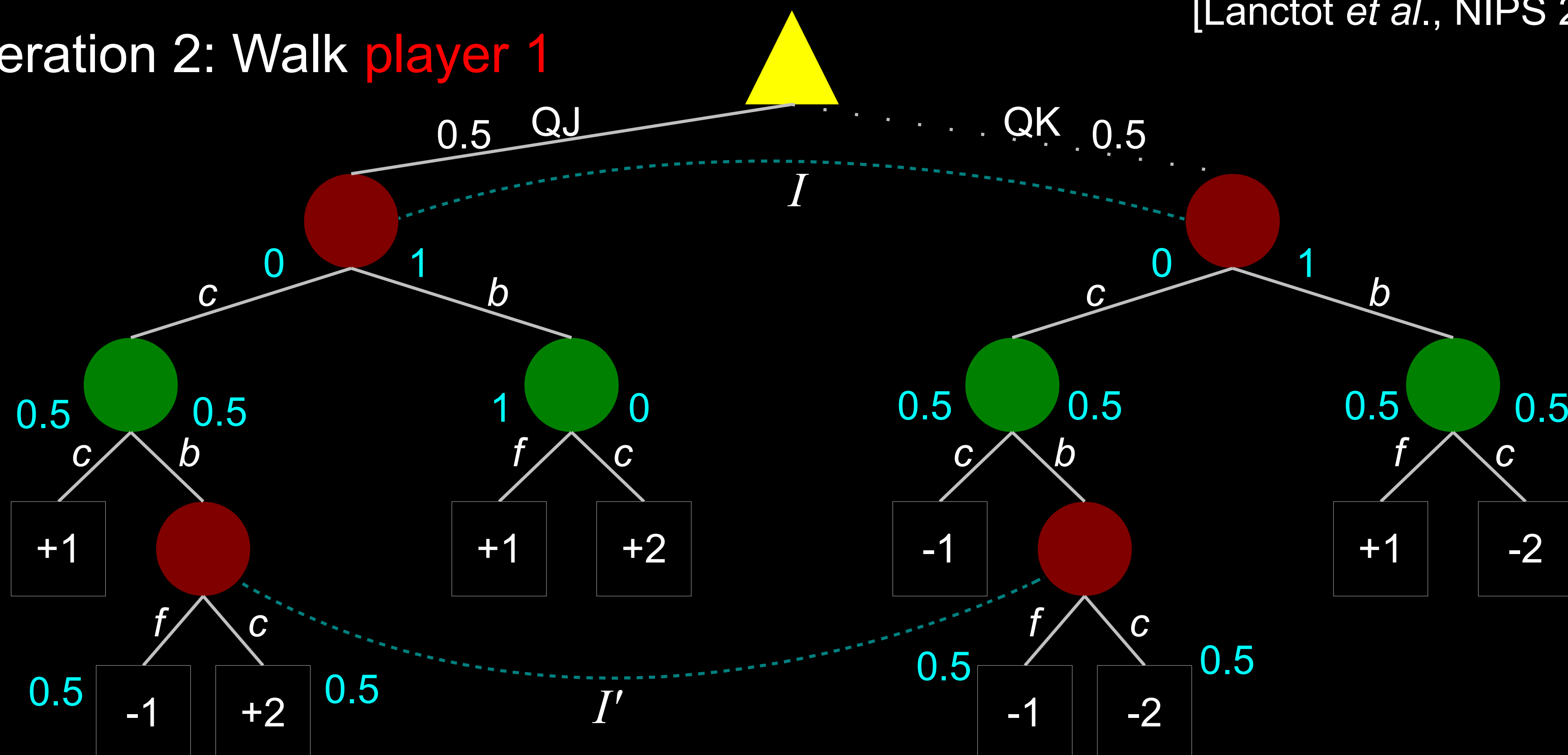


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al*., NIPS 2009]

- Iteration 2: Walk player 1

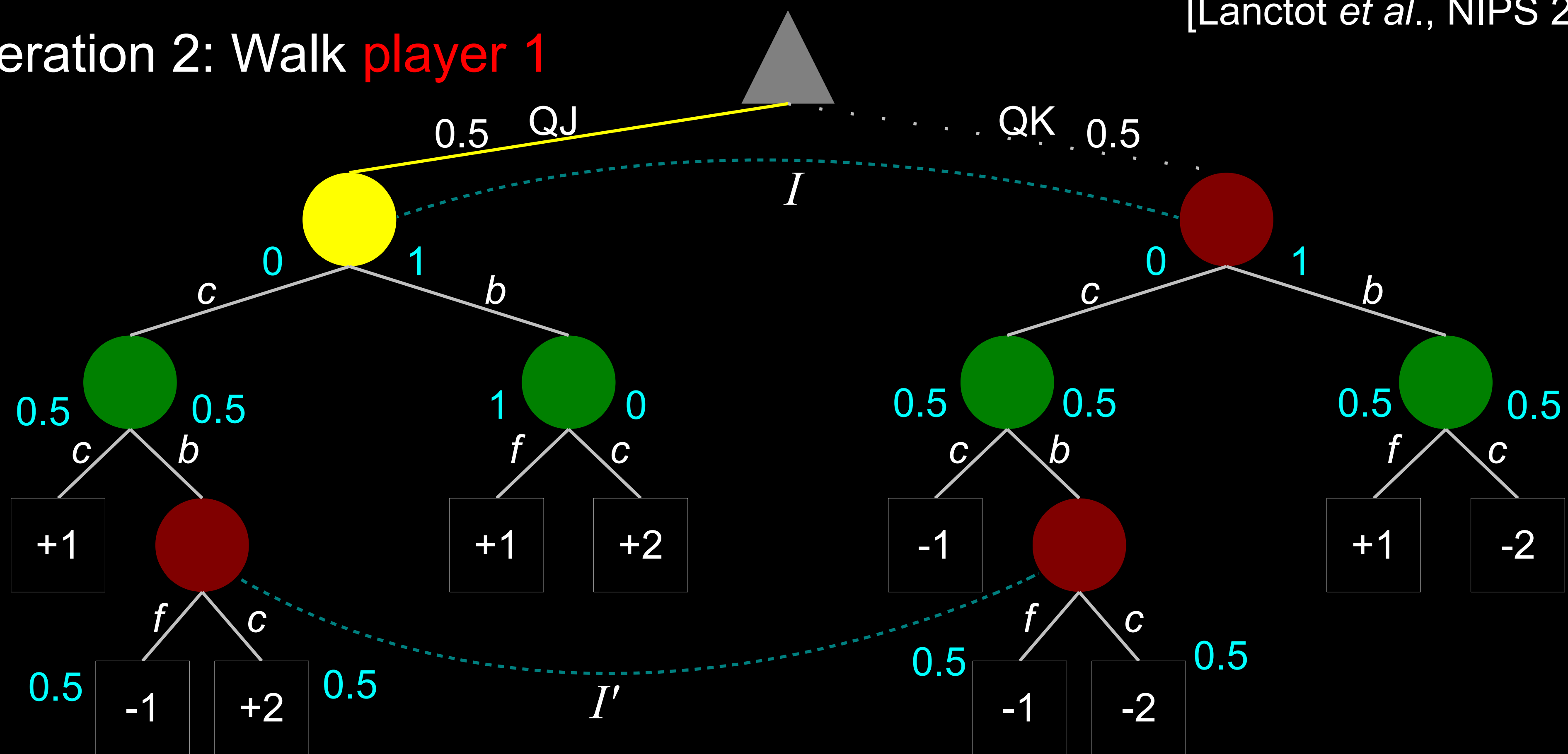


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 1

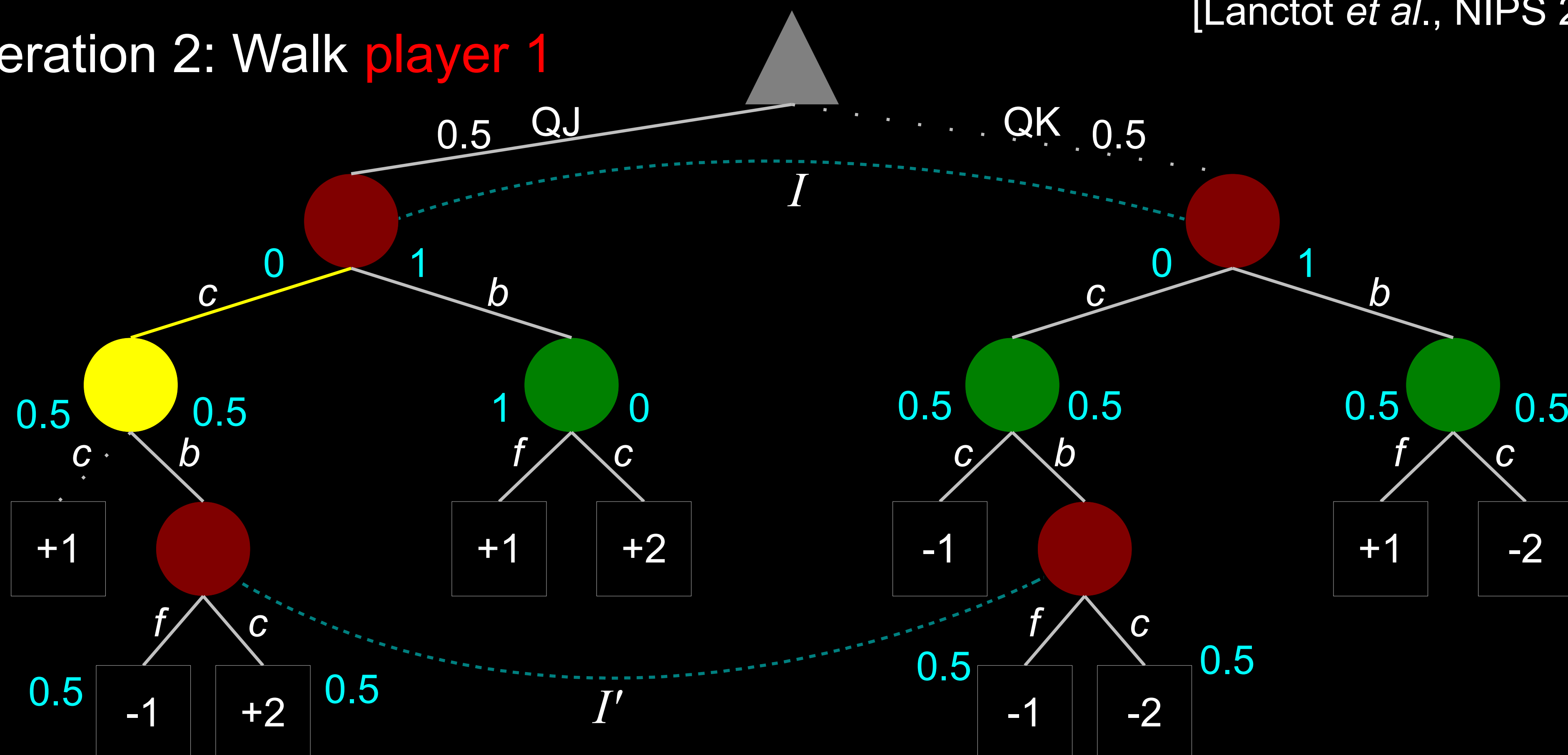


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 1

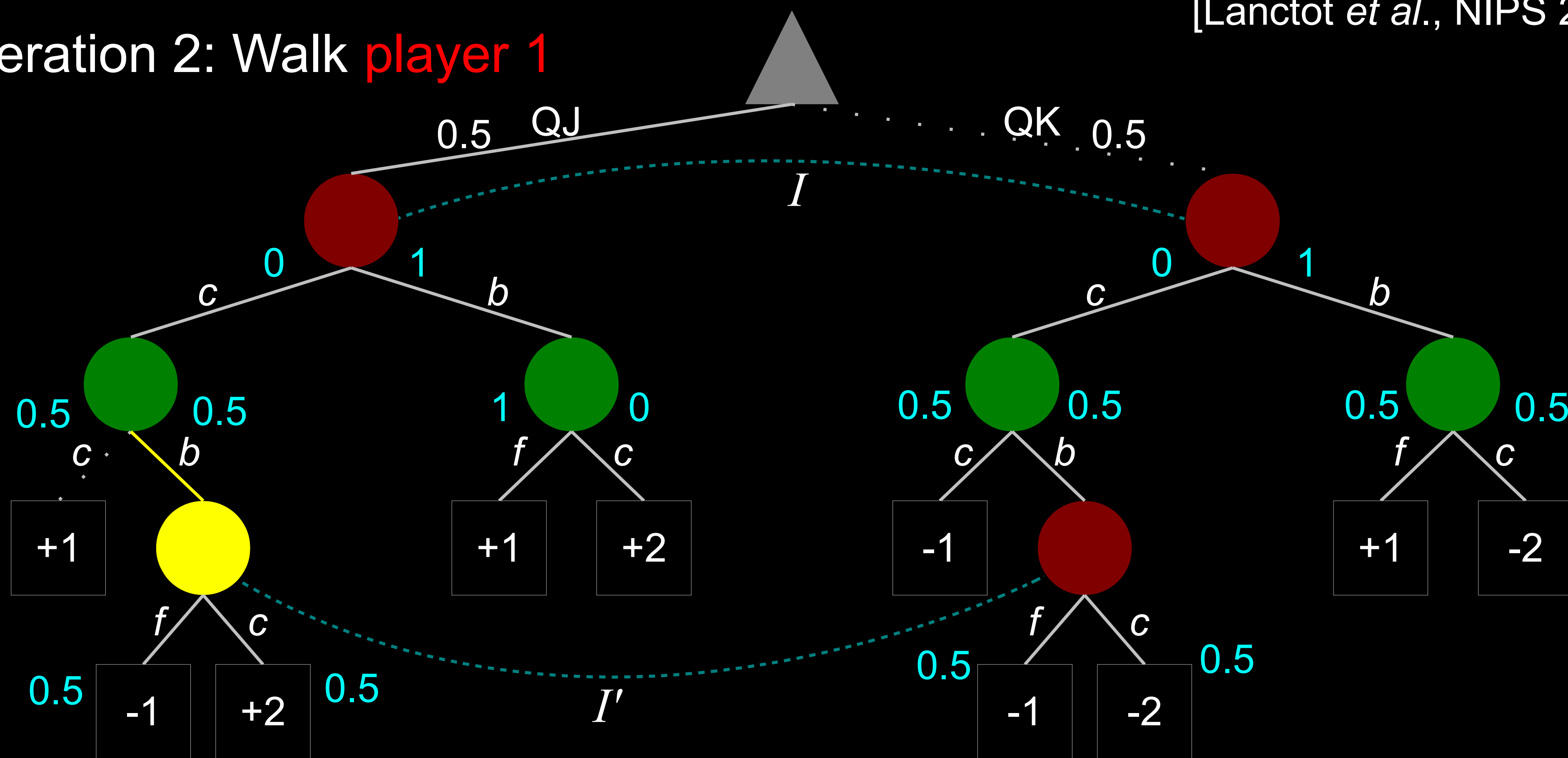


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 1

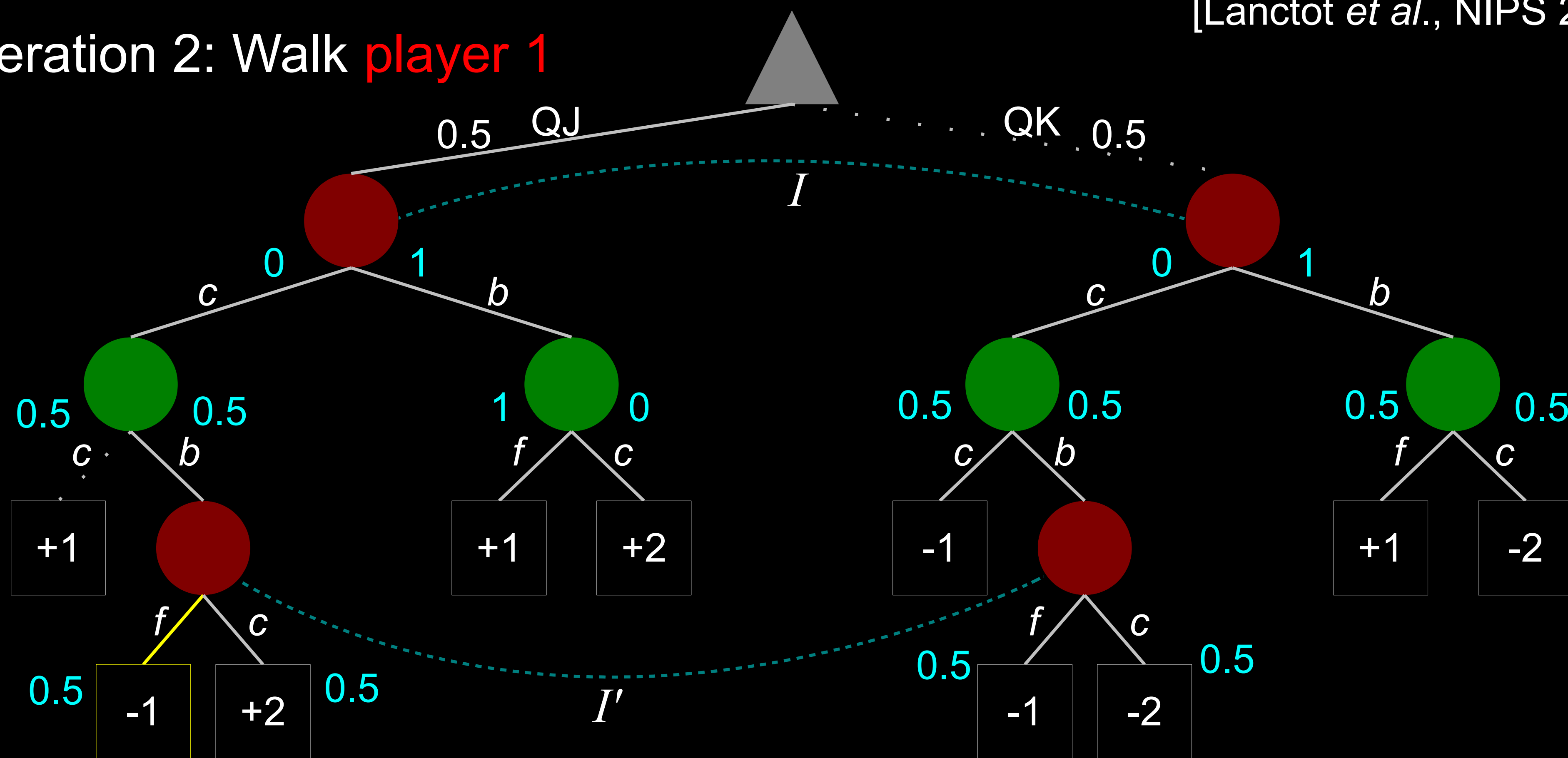


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 1

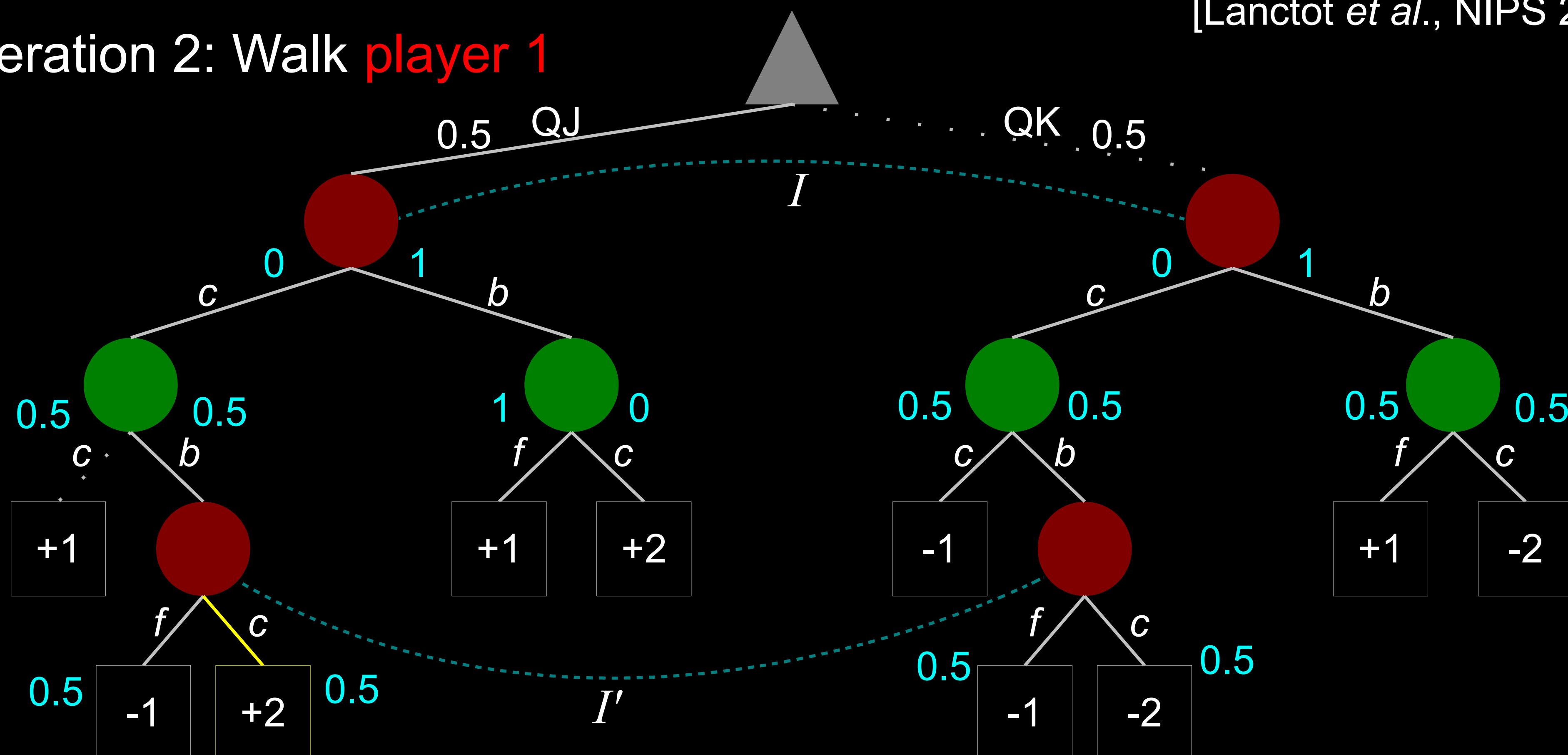


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al*., NIPS 2009]

- Iteration 2: Walk player 1

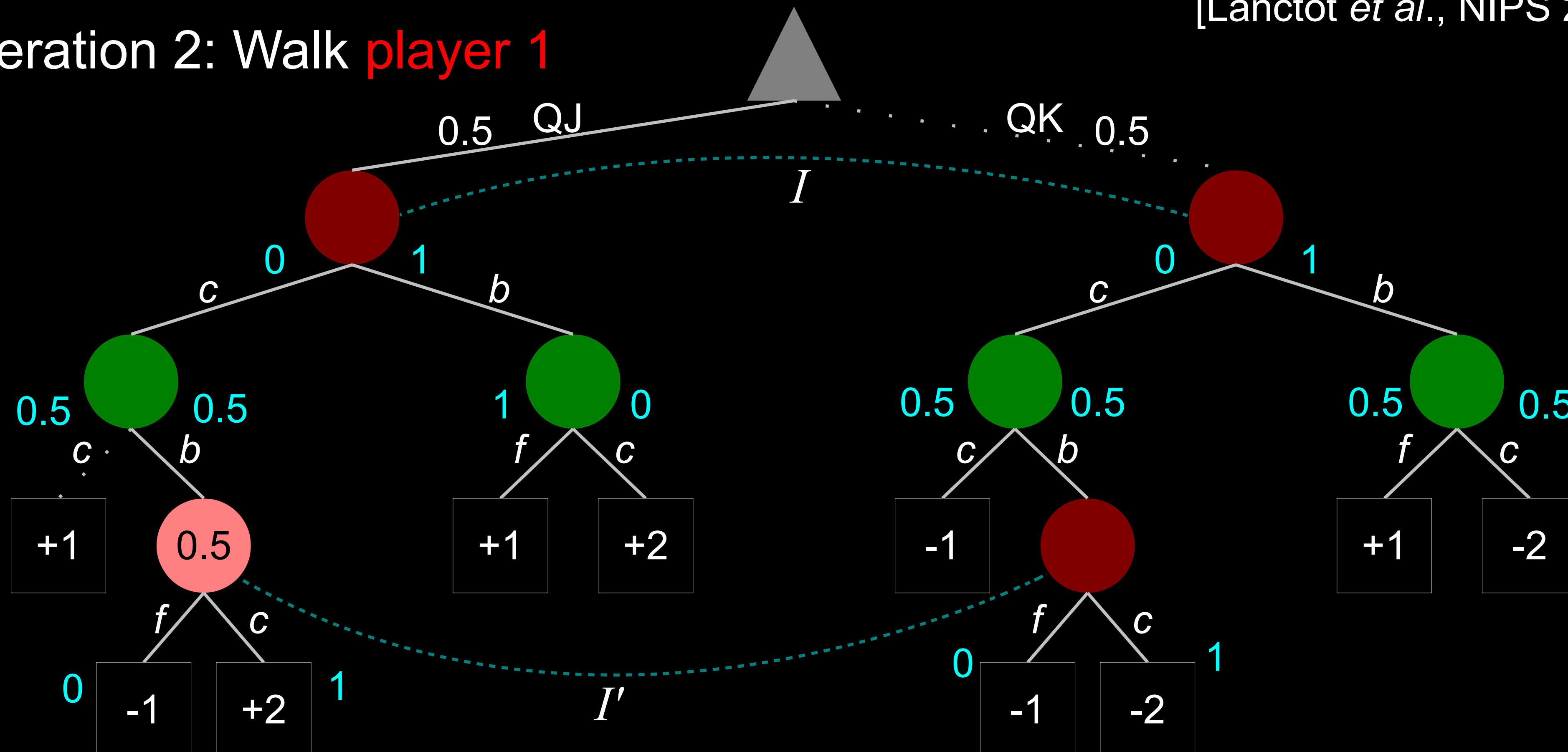


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 1

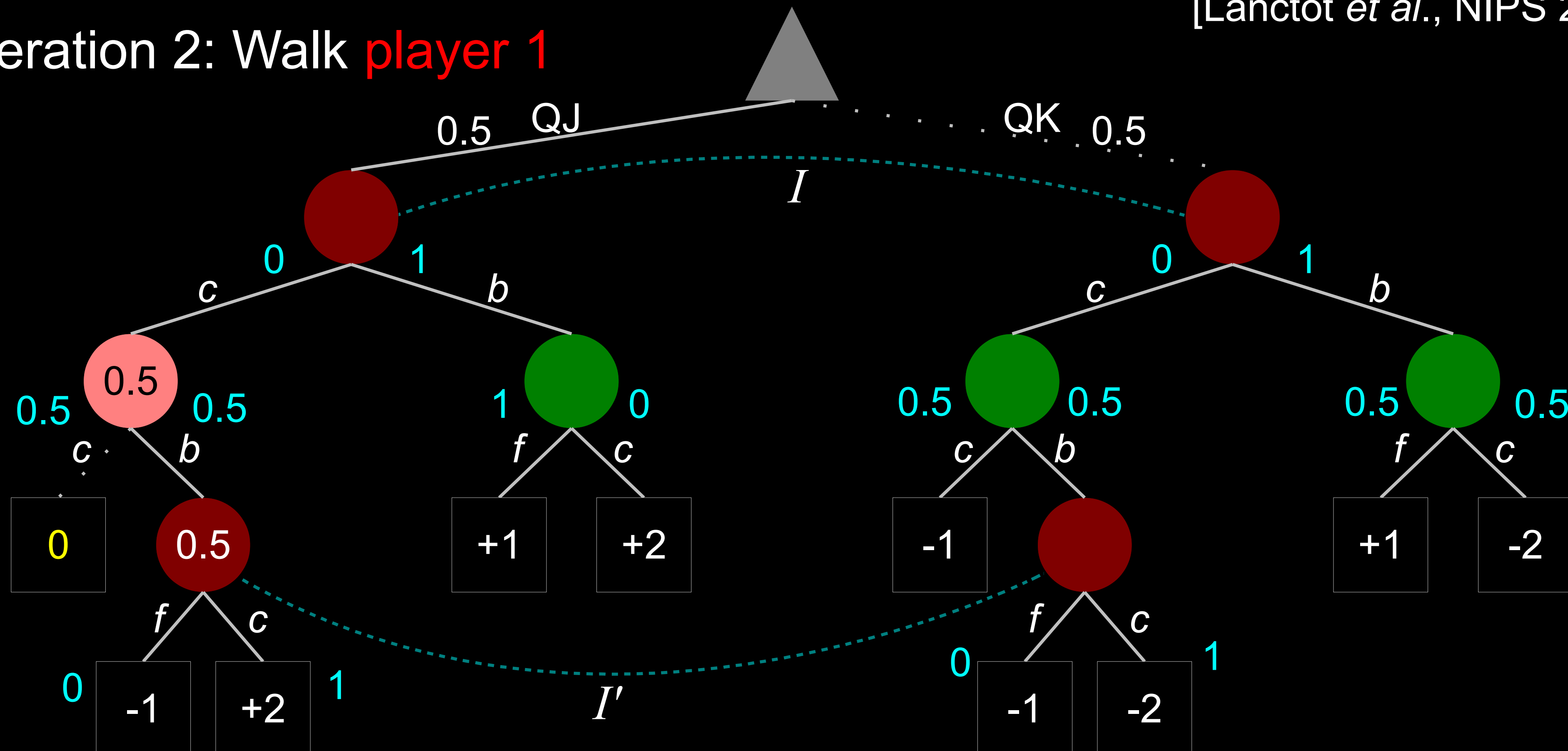


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 1

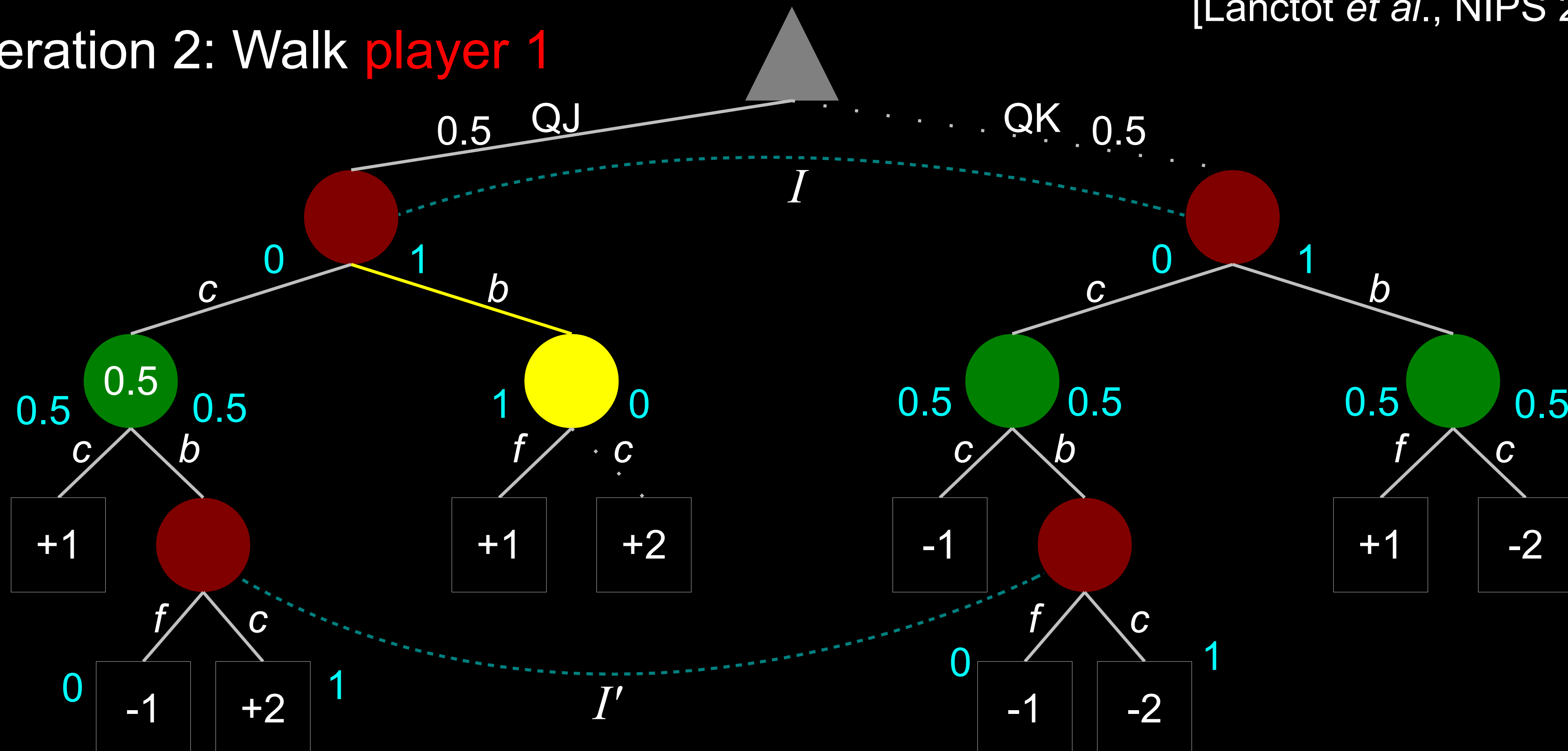


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 1

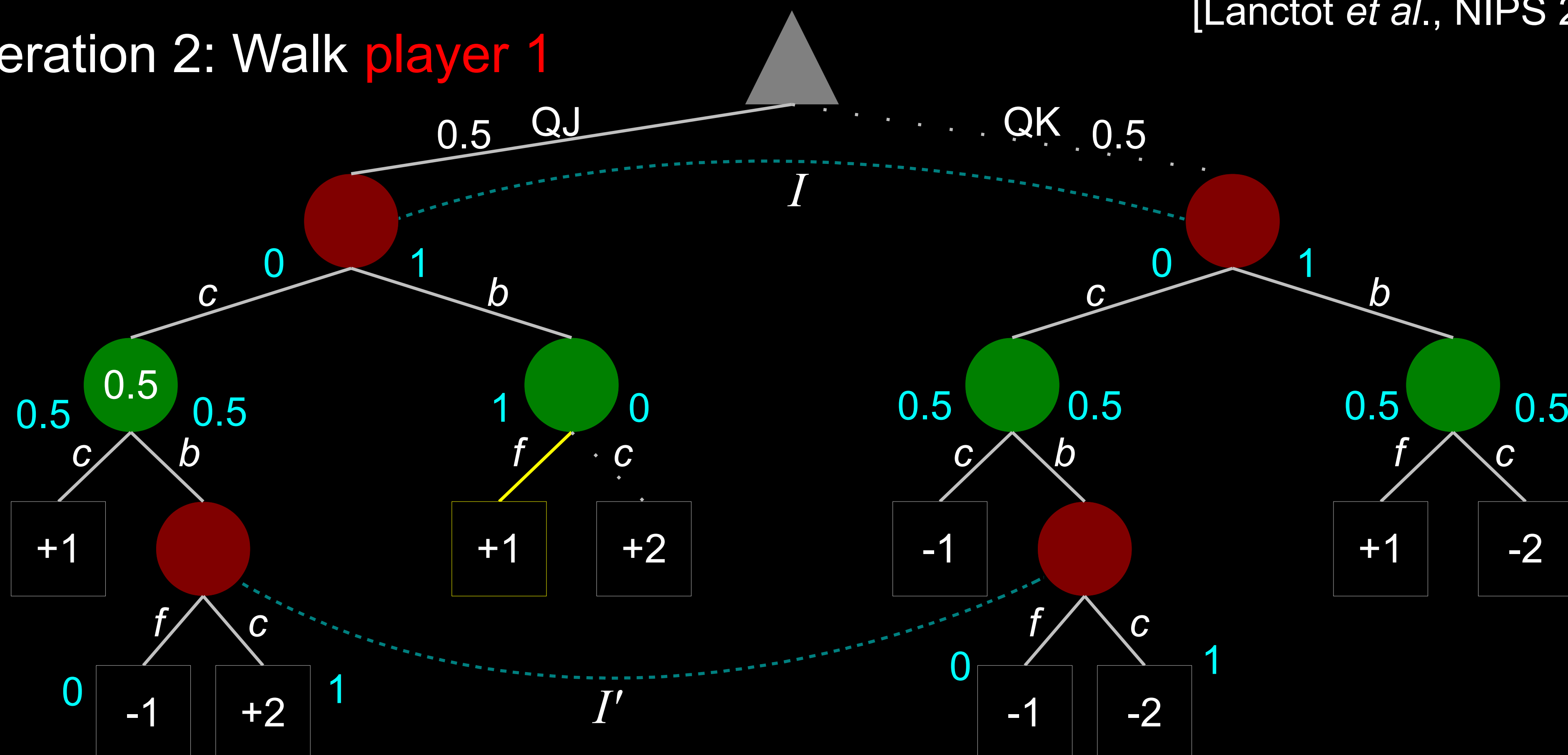


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 1

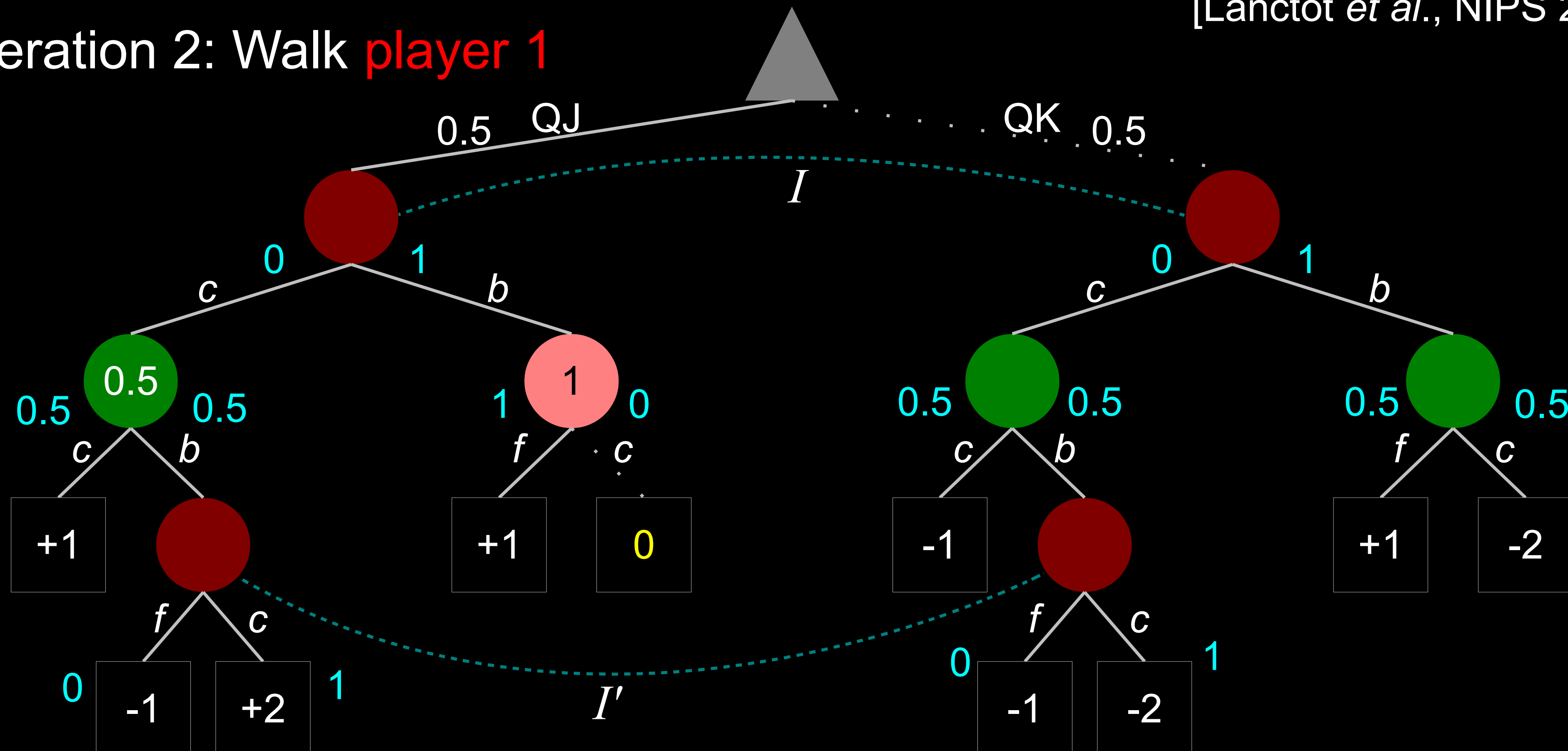


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 1

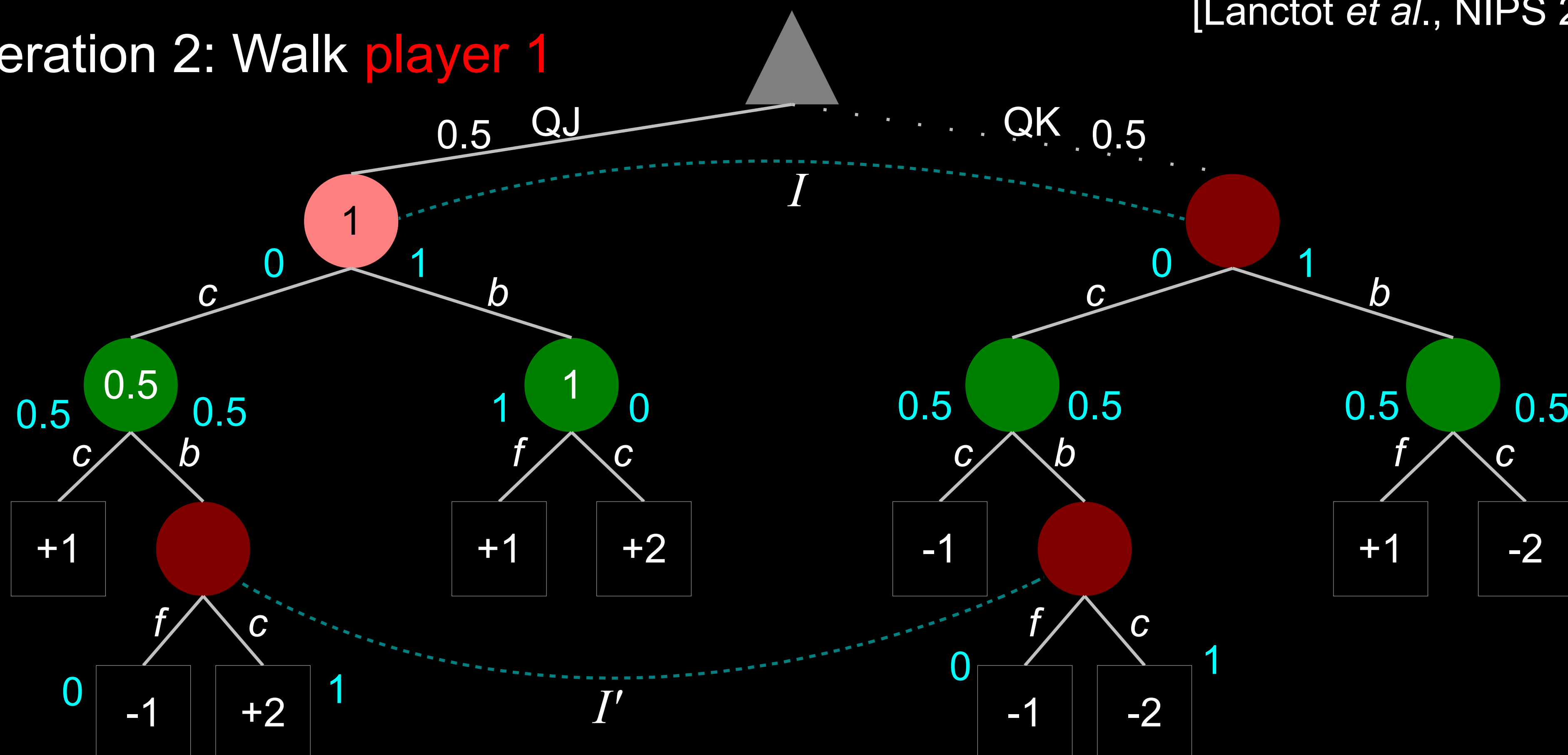


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 1

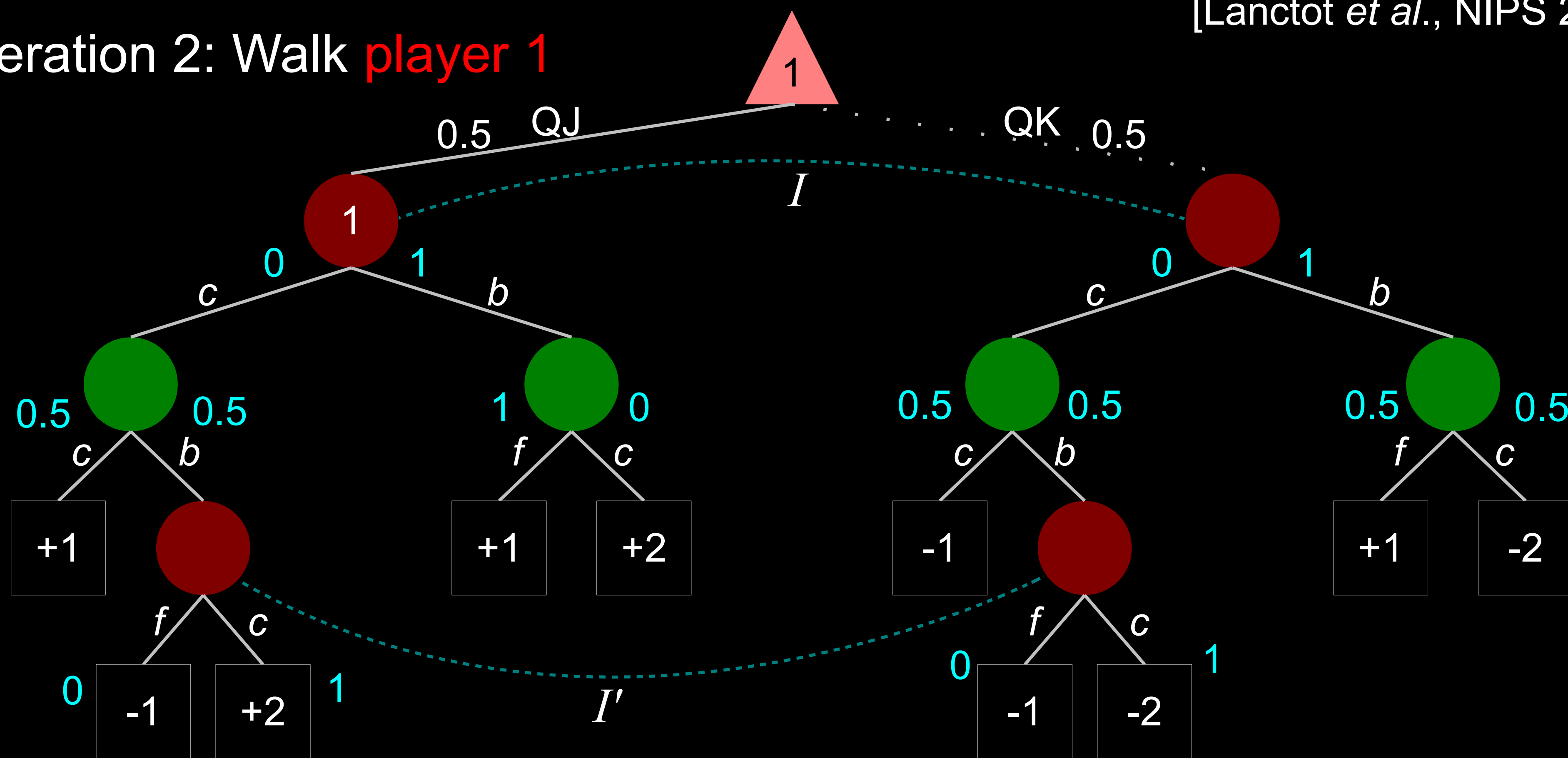


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 1

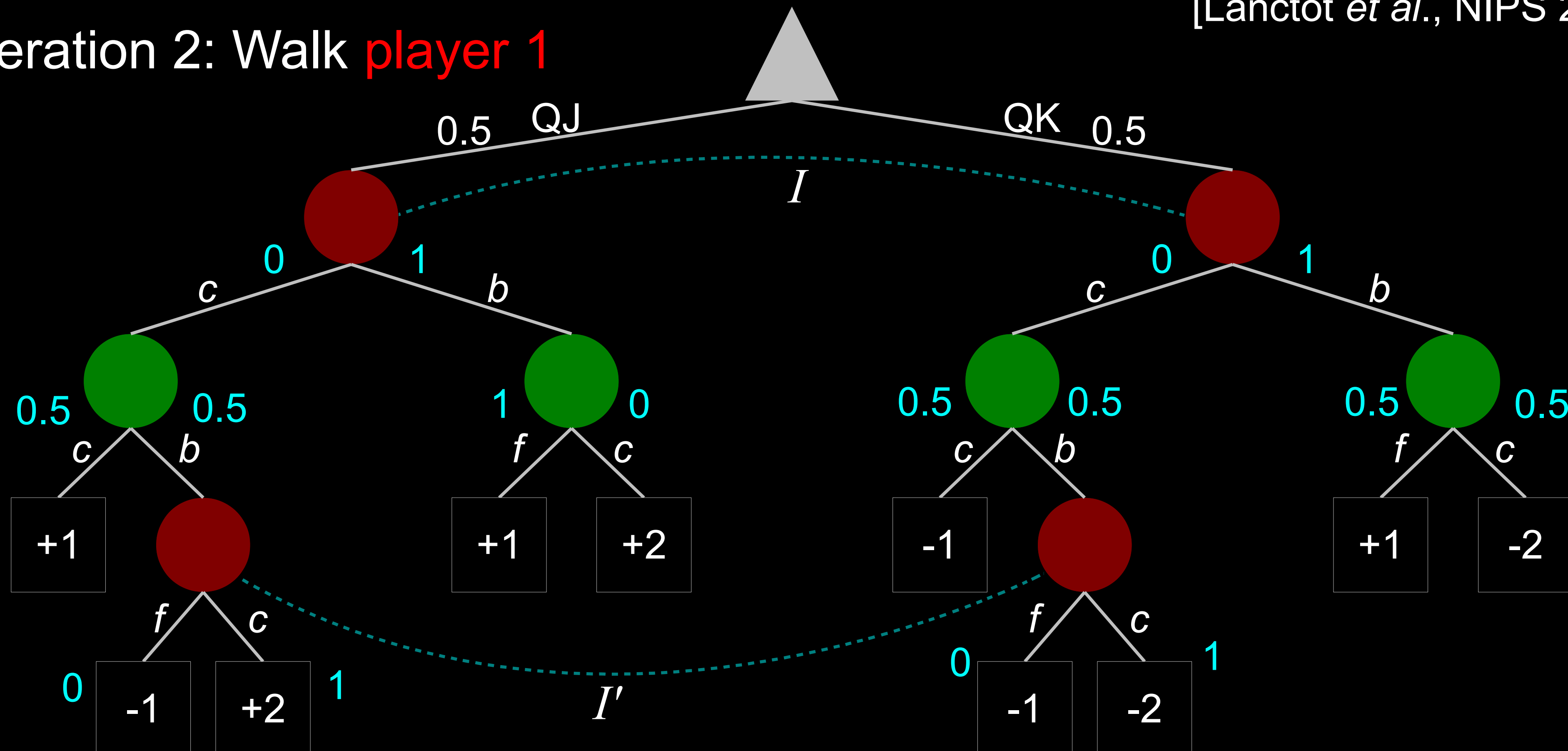


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 1

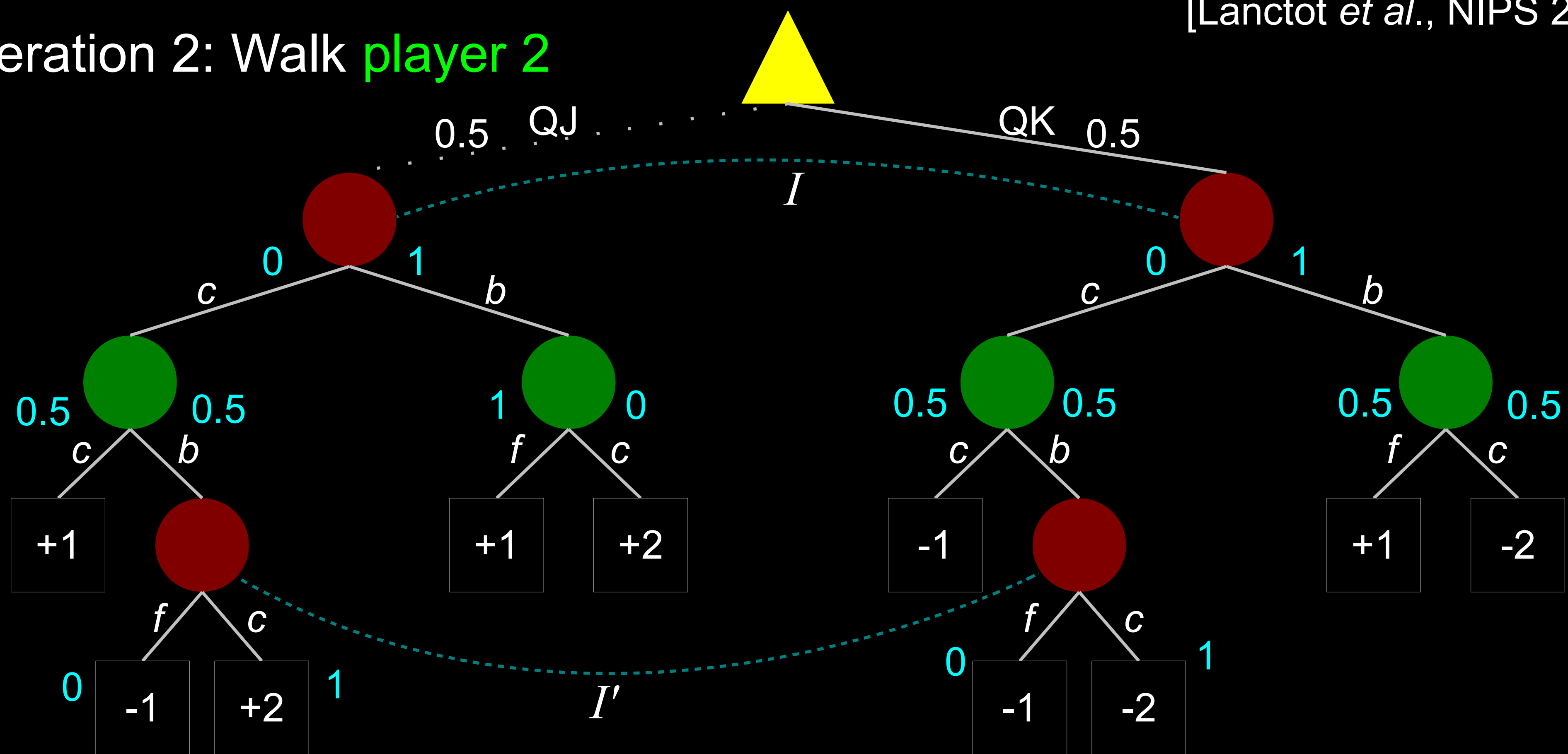


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 2

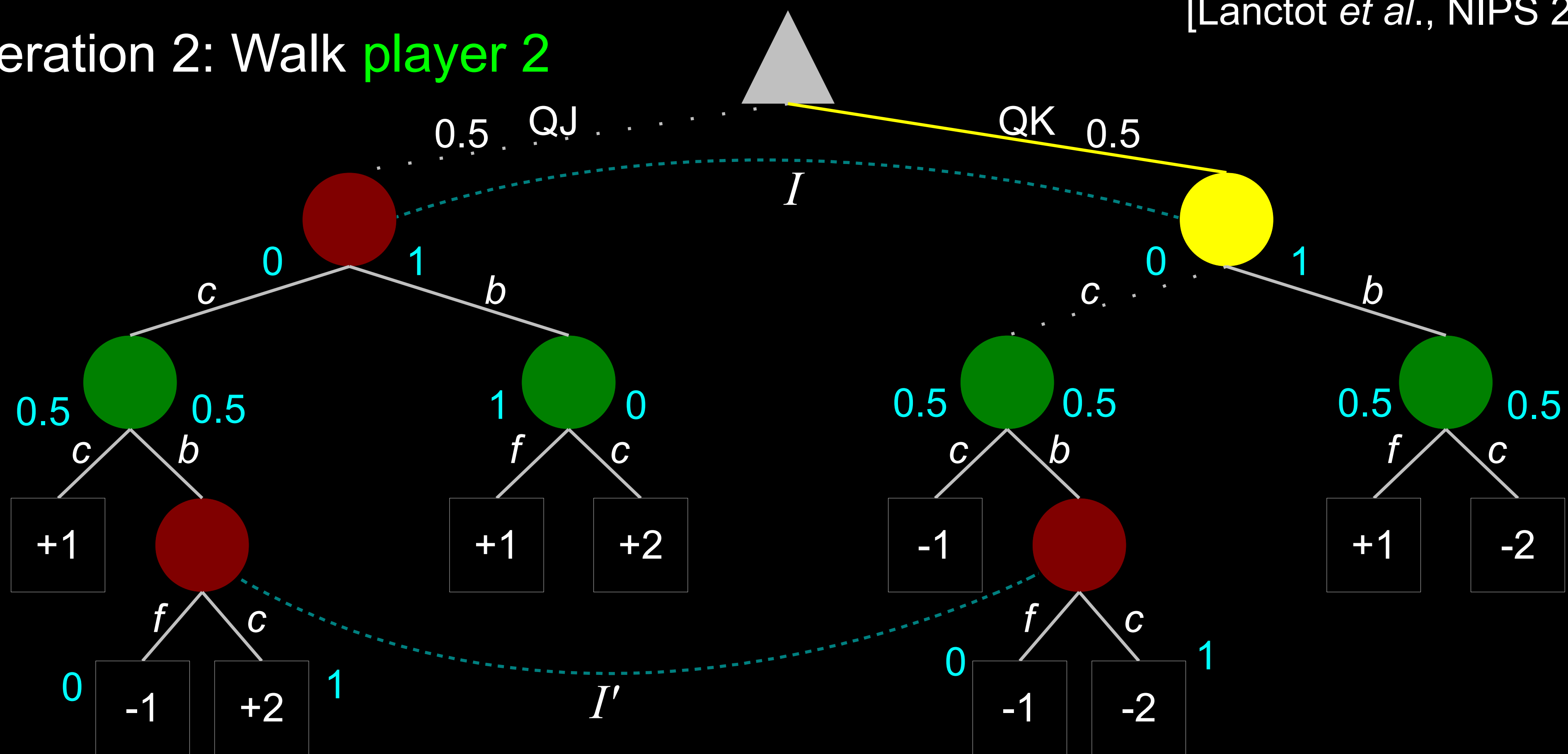


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 2

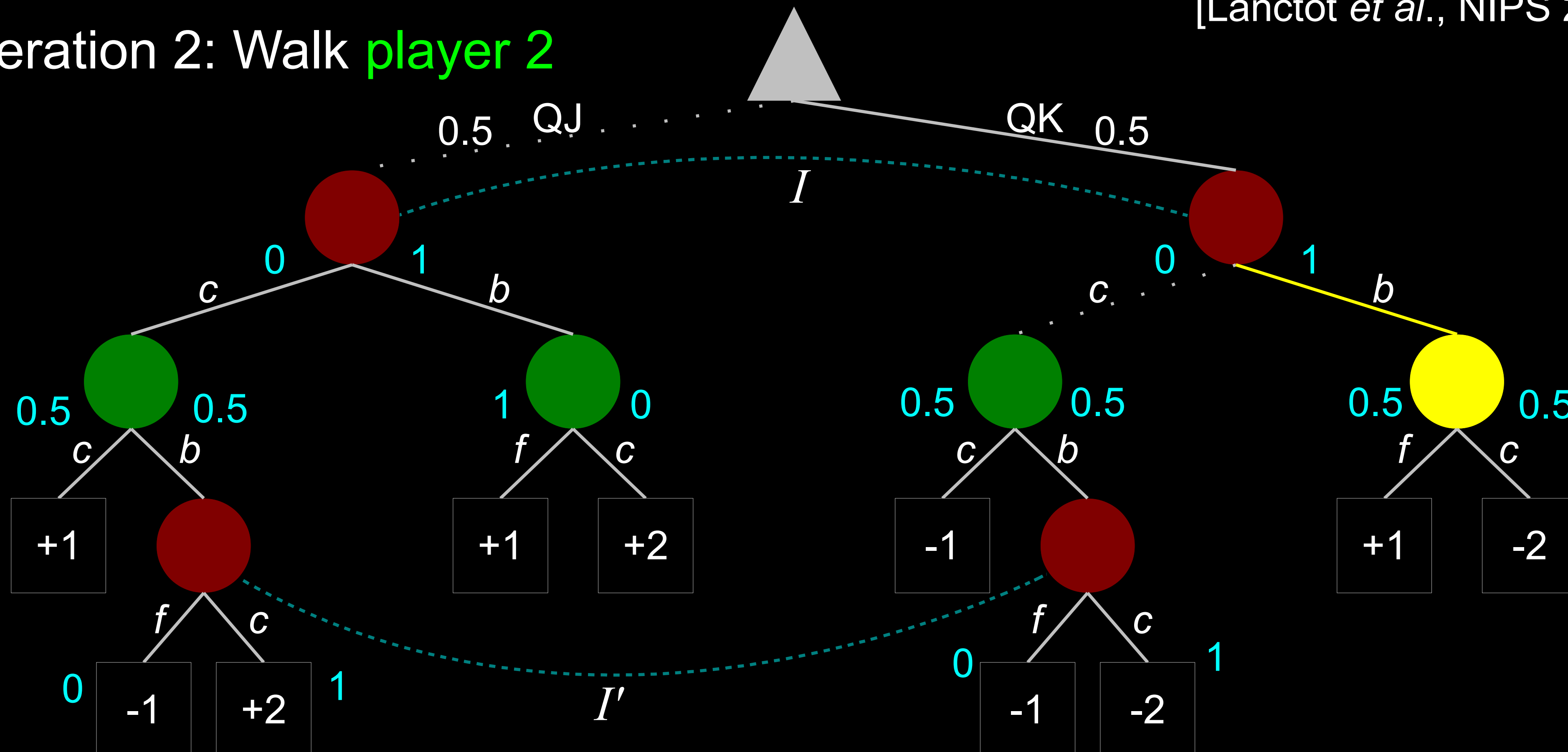


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 2

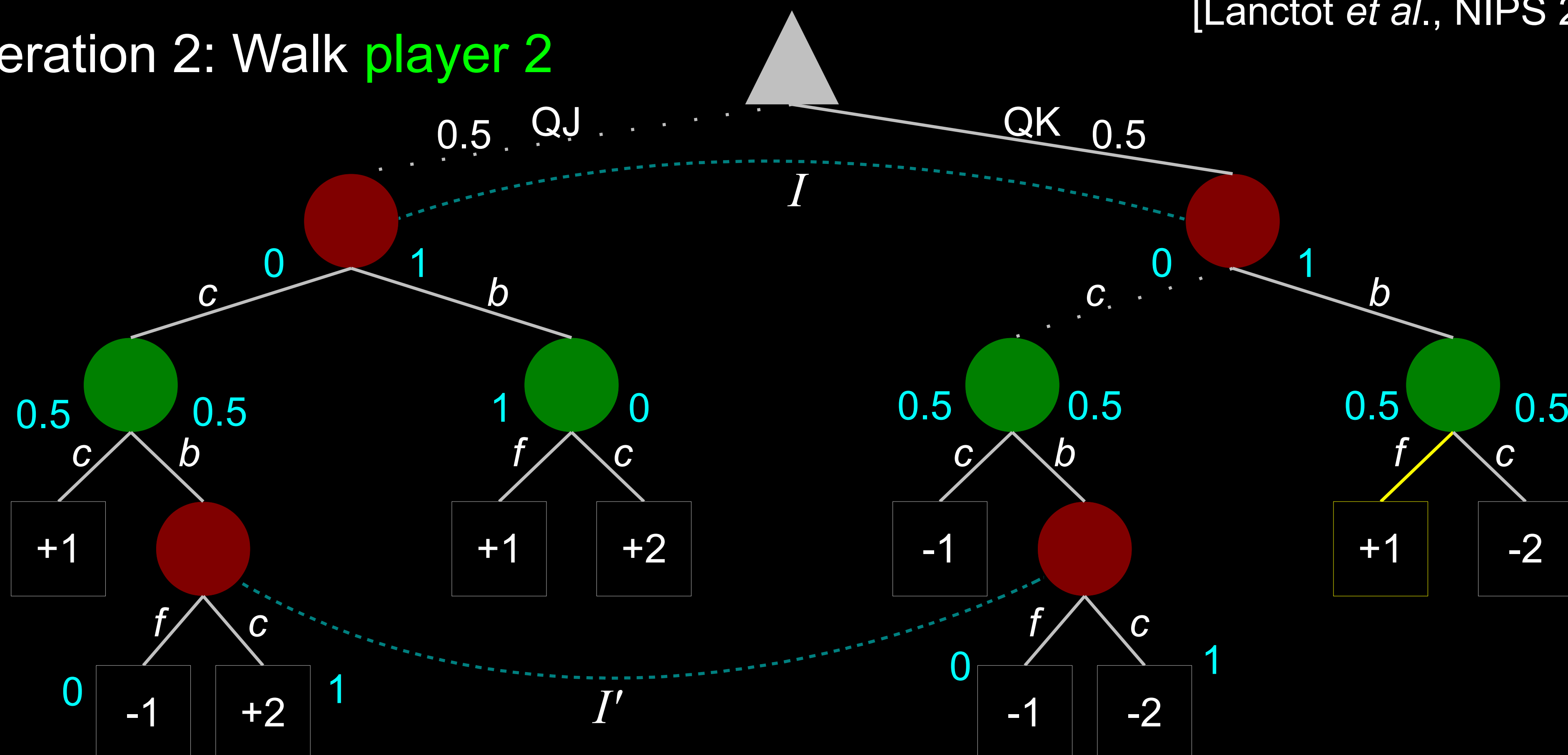


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 2

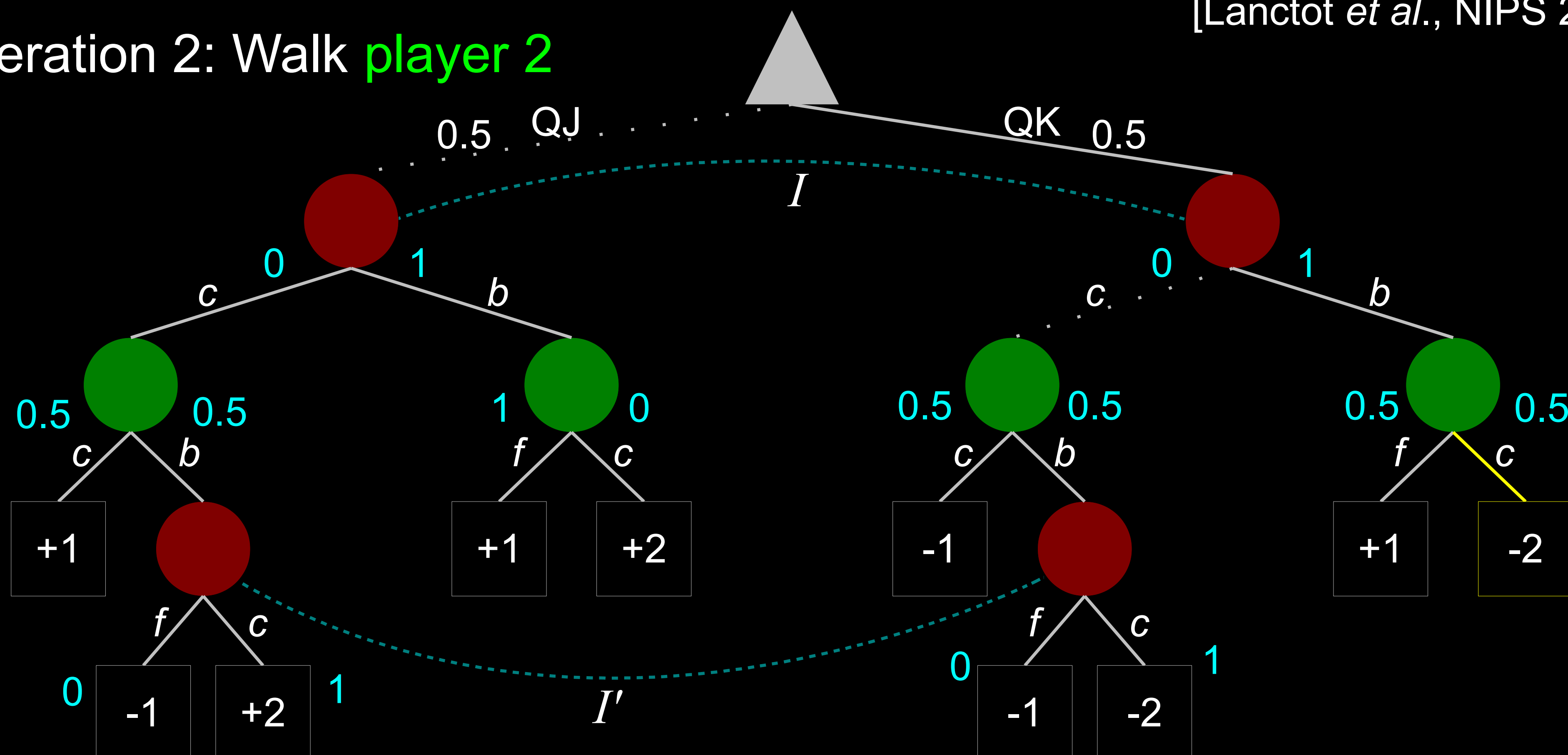


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al*., NIPS 2009]

- Iteration 2: Walk player 2

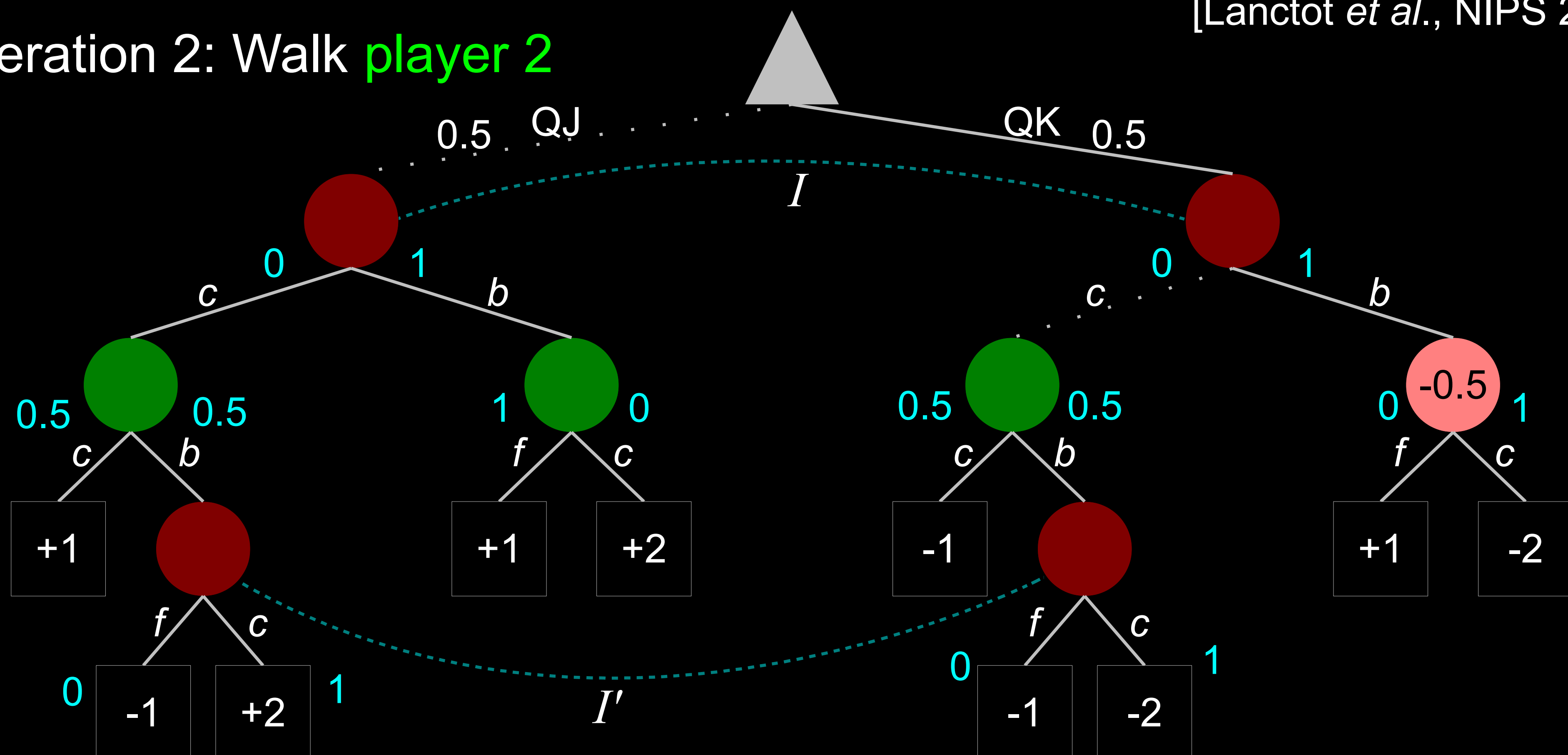


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 2

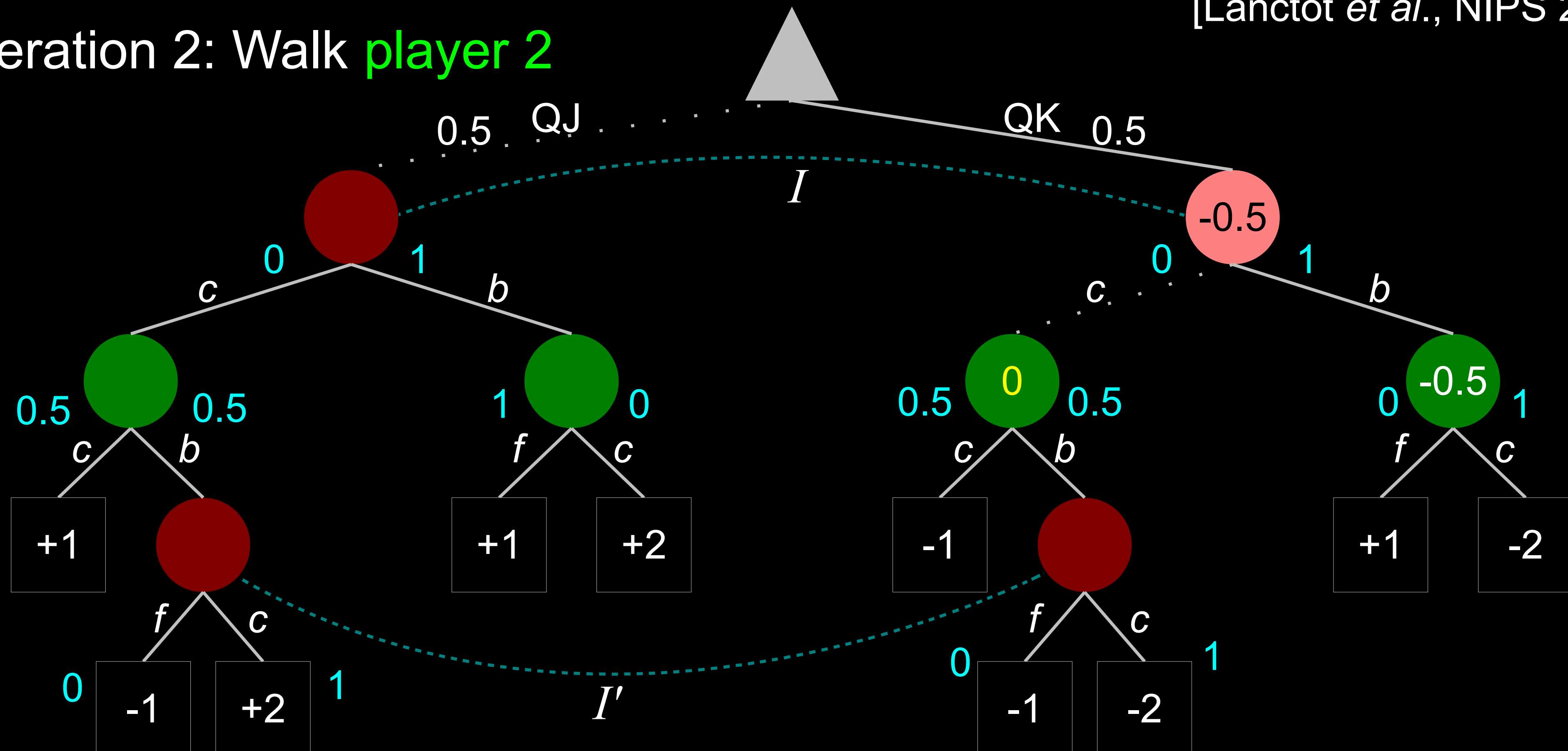


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 2

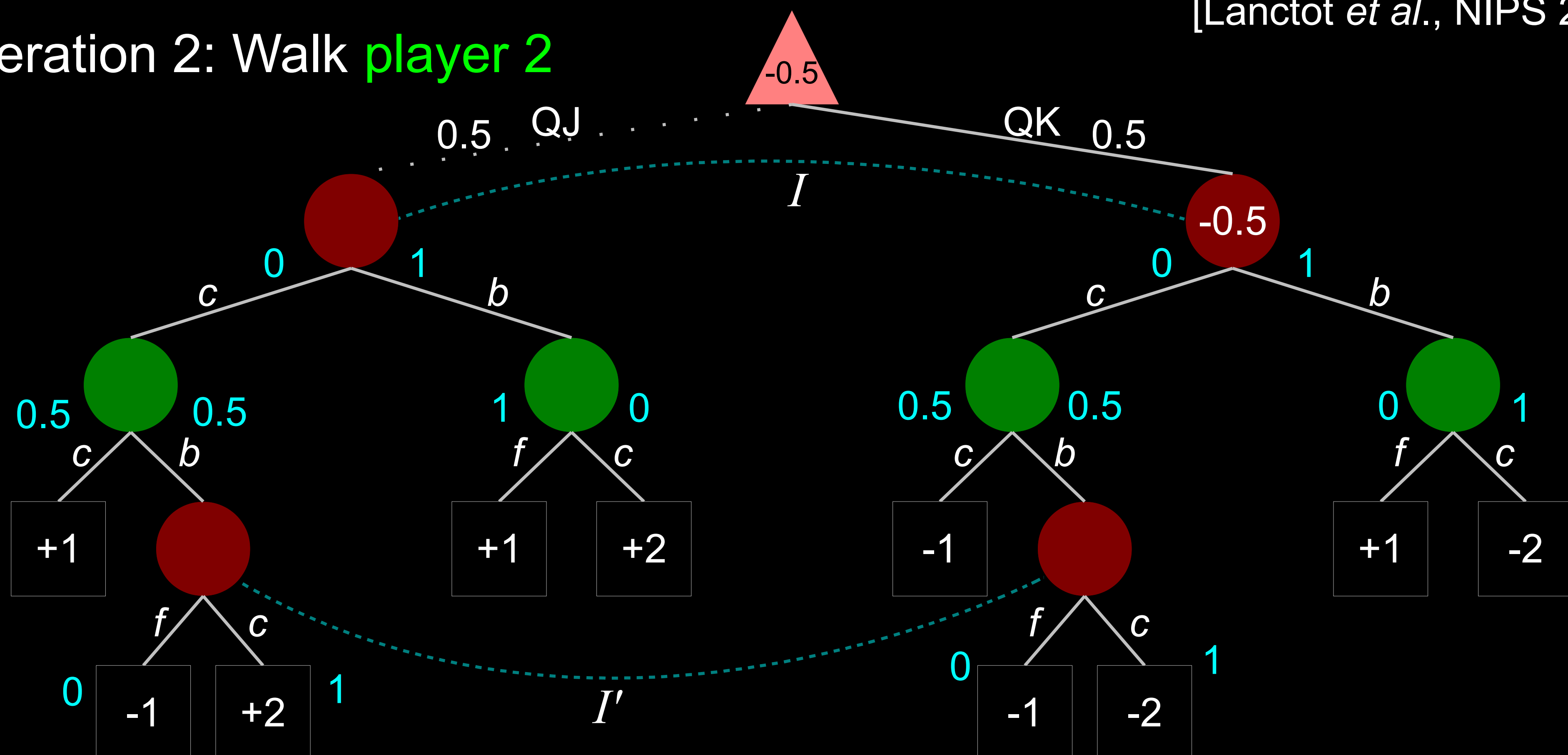


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2: Walk player 2

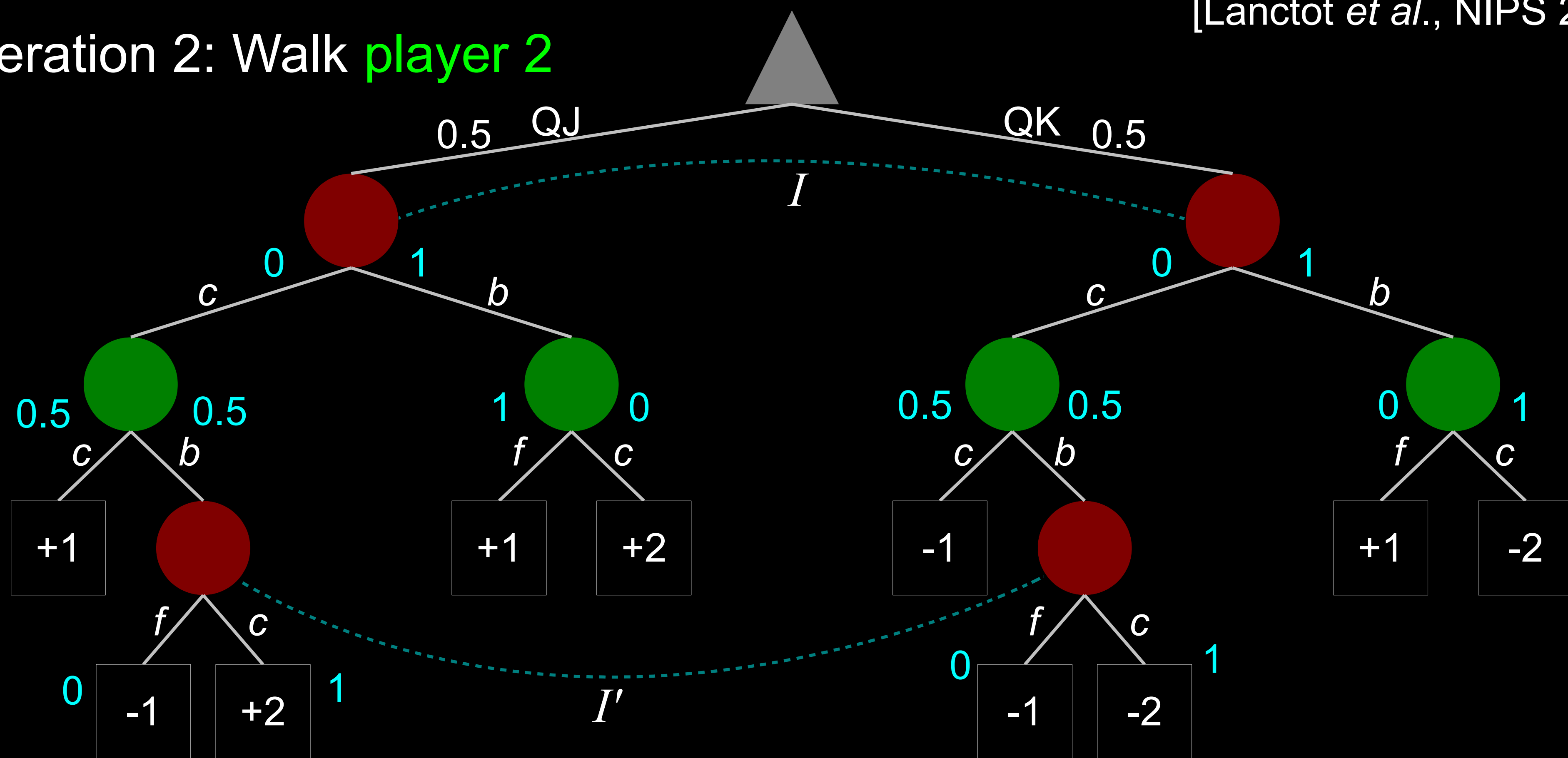


Other Variants: External Sampling

At each chance or opponent node, traverse only one action per iteration

[Lanctot *et al*., NIPS 2009]

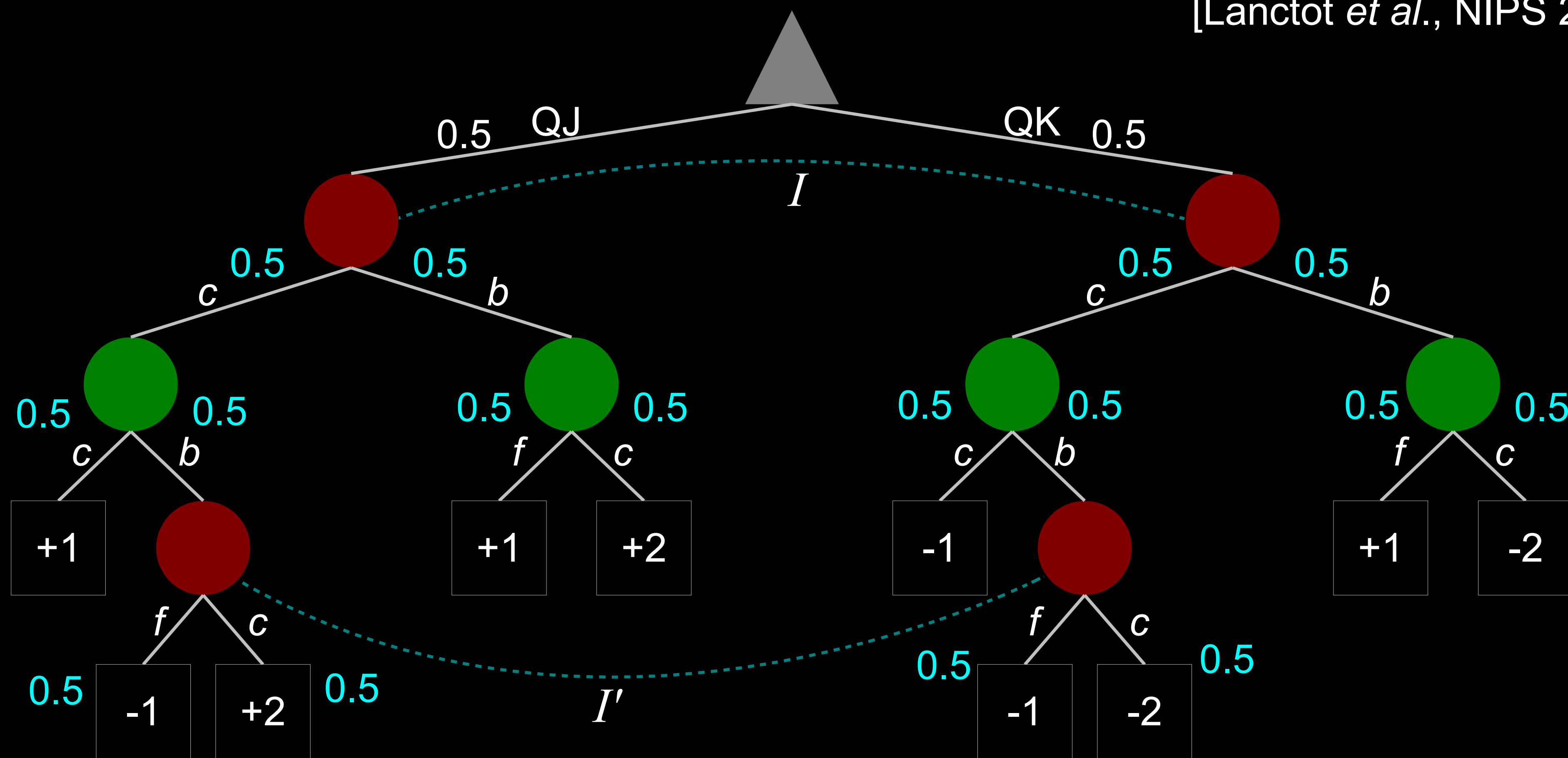
- Iteration 2: Walk player 2



Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

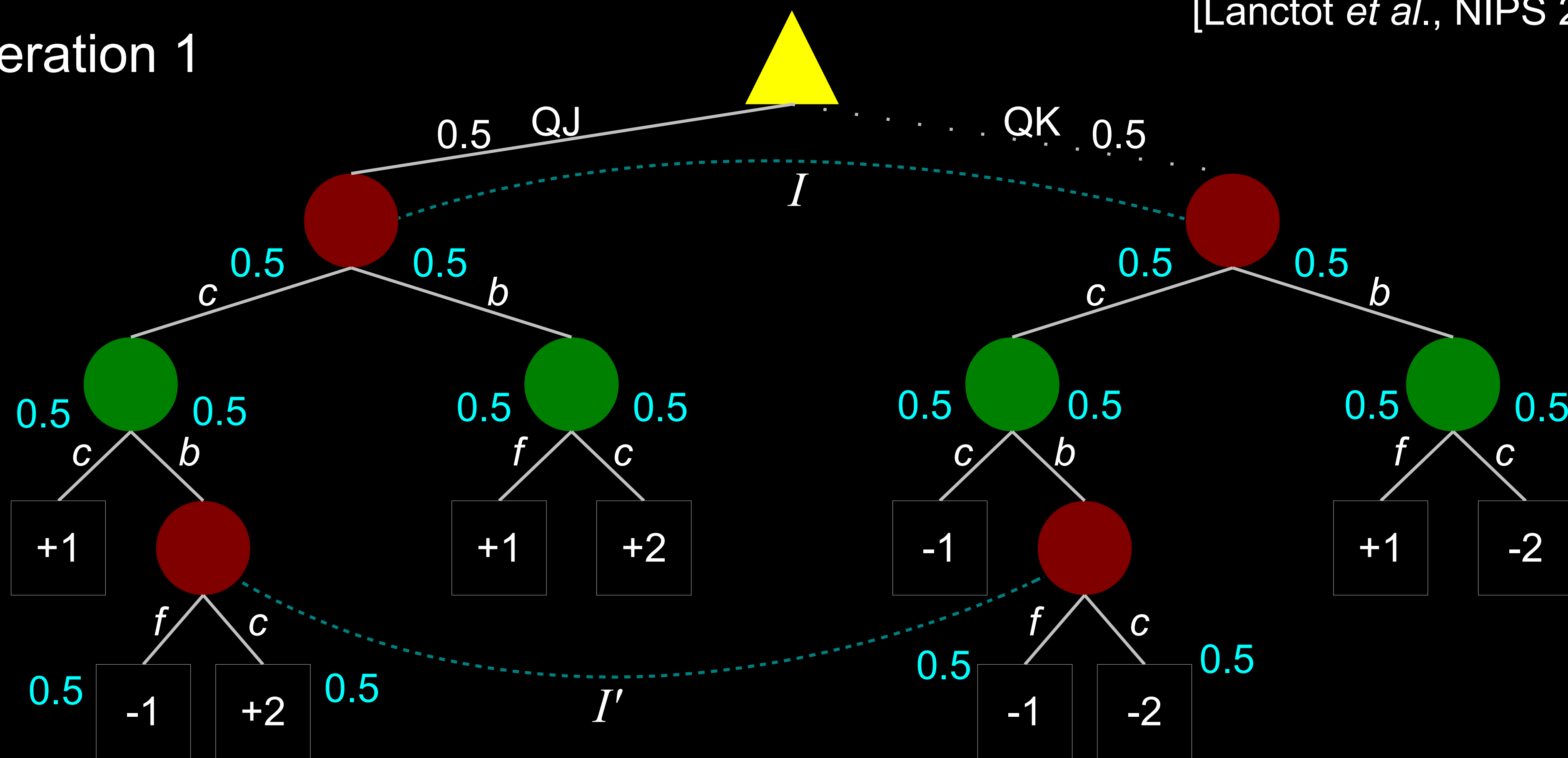


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1

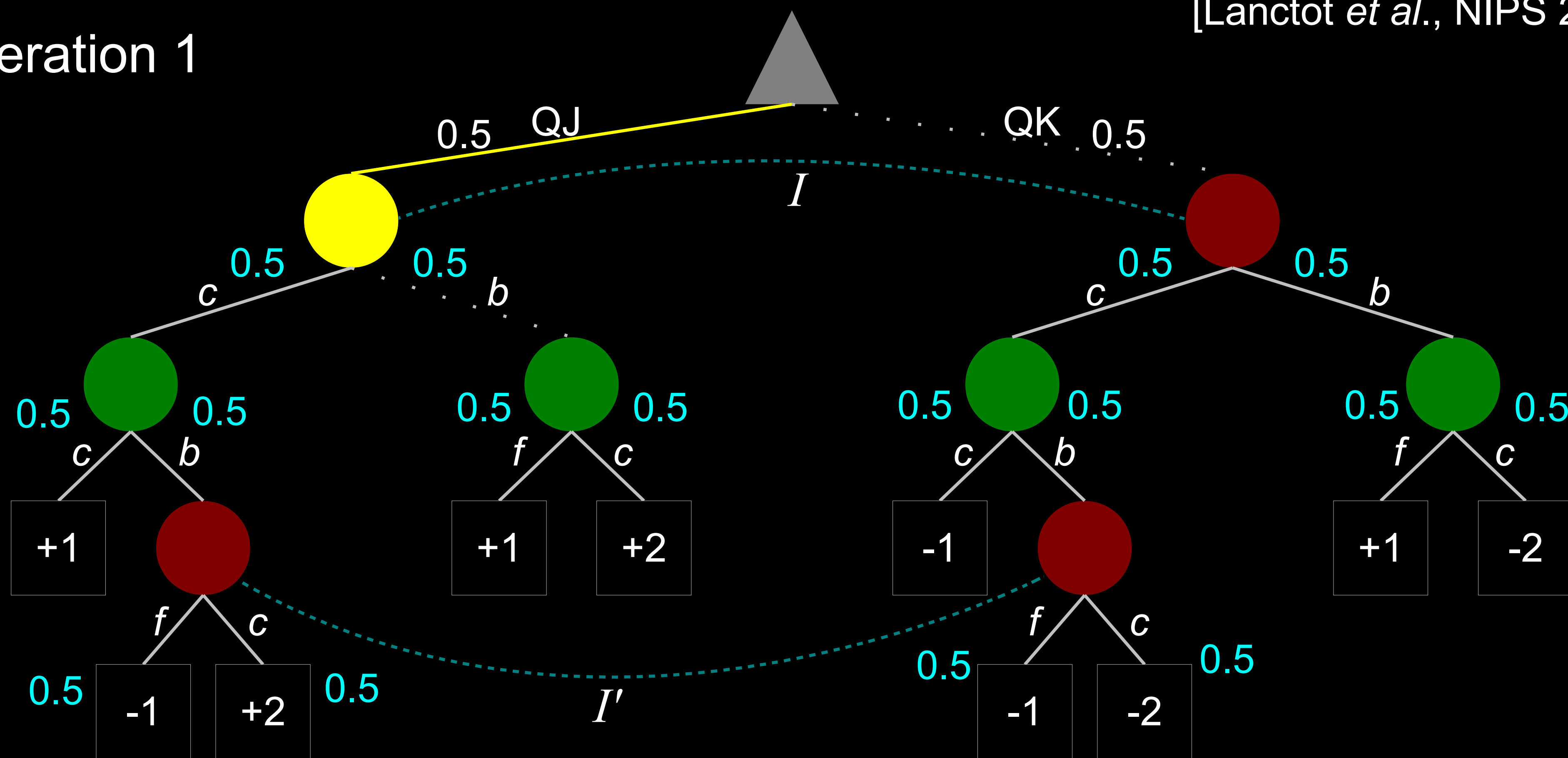


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1

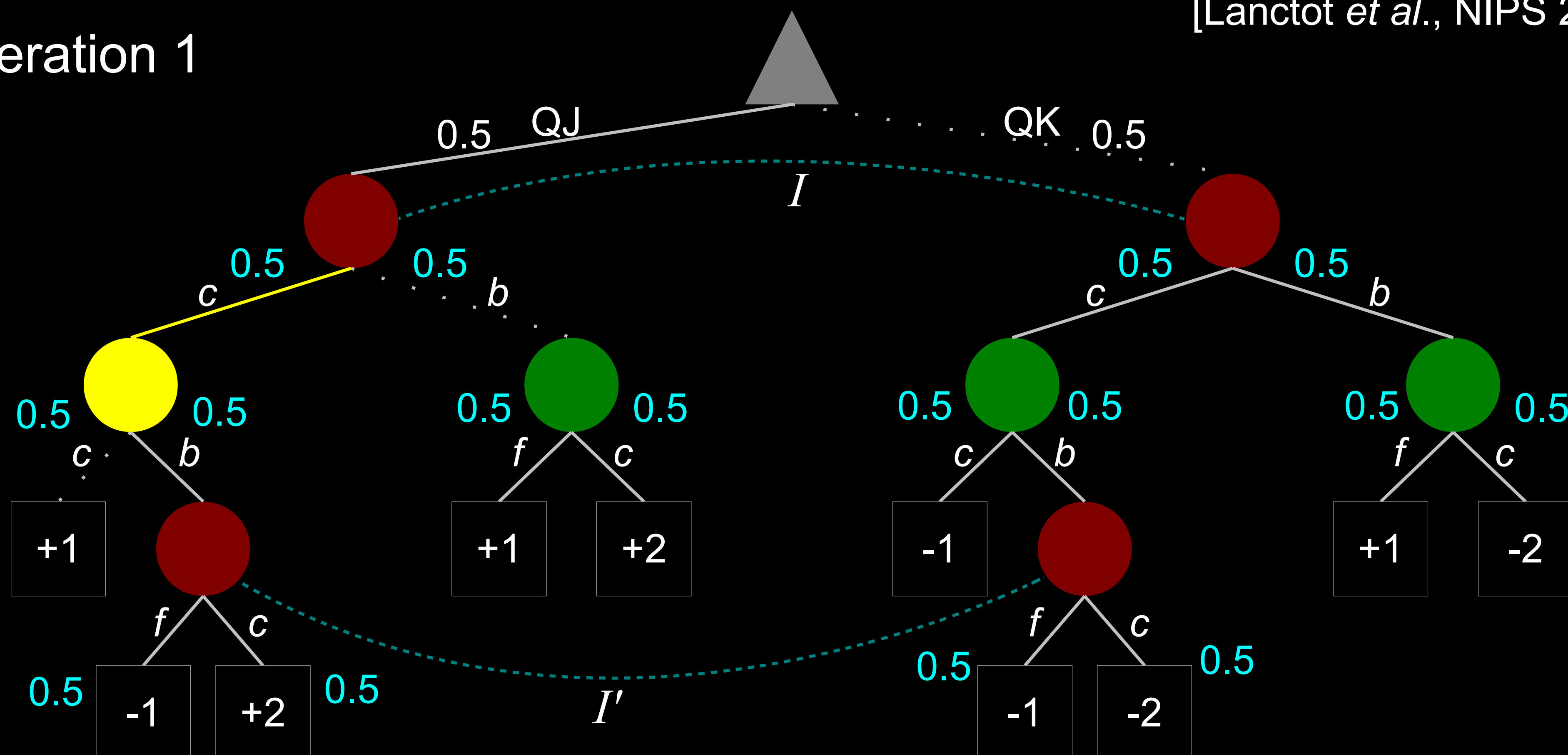


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1

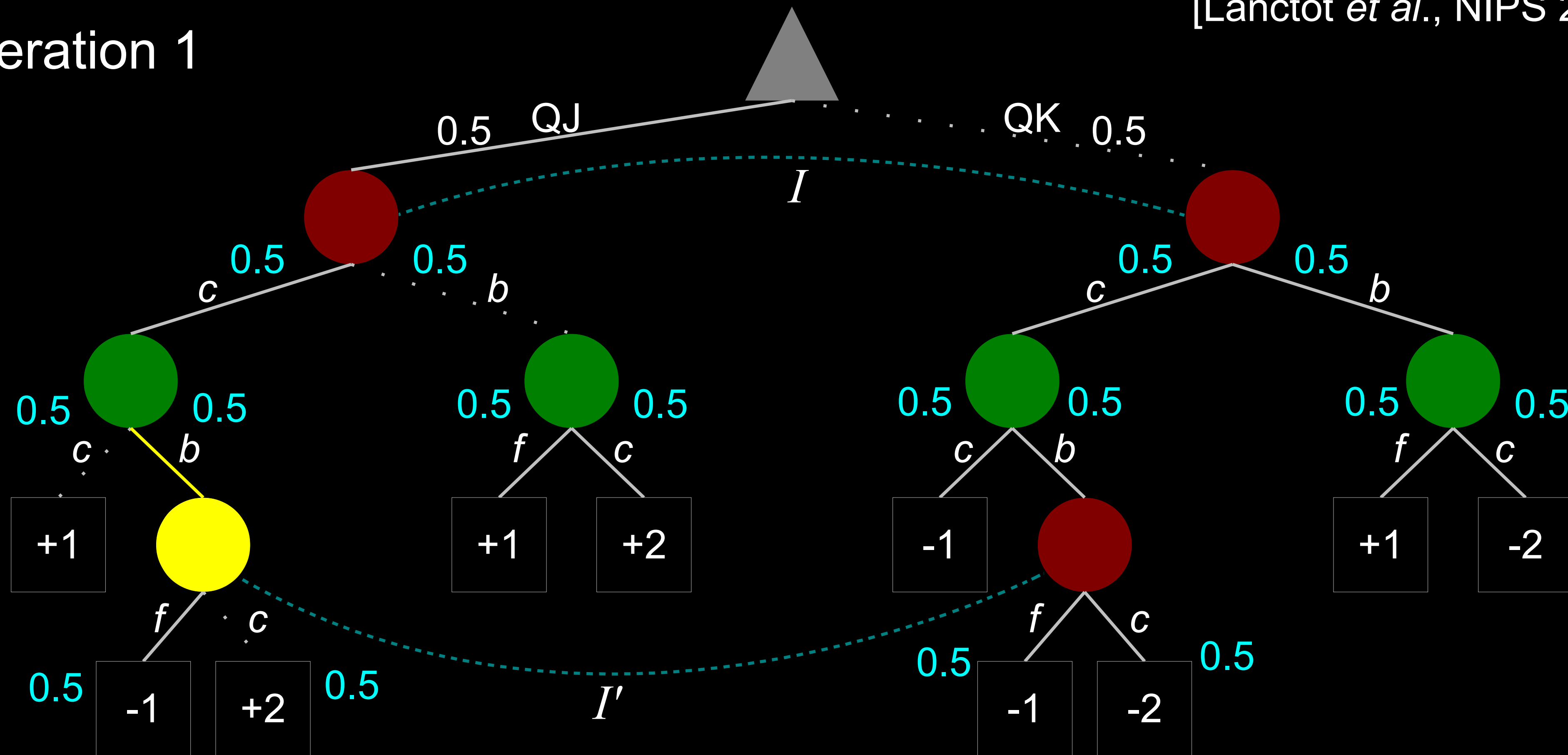


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1

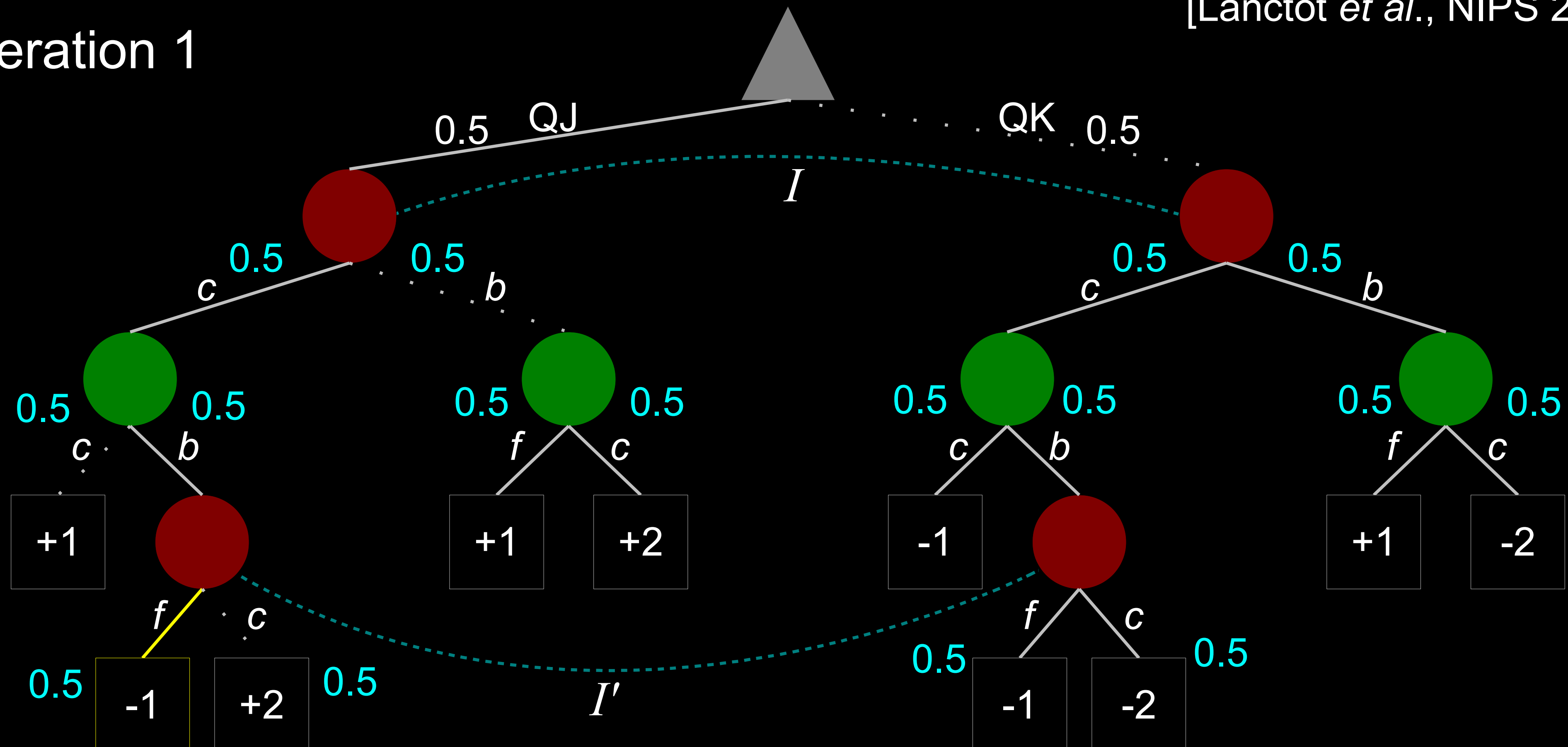


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1

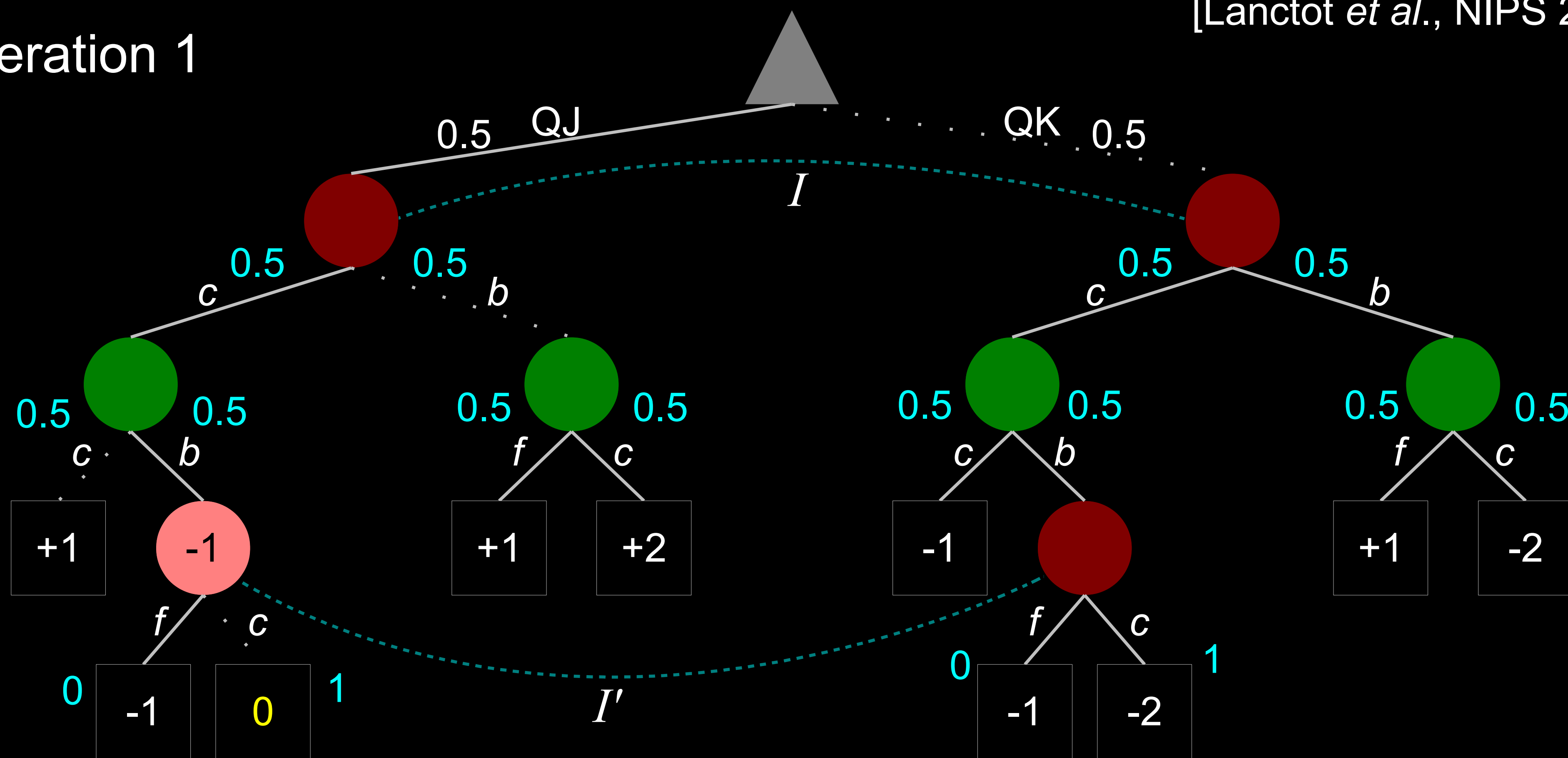


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1

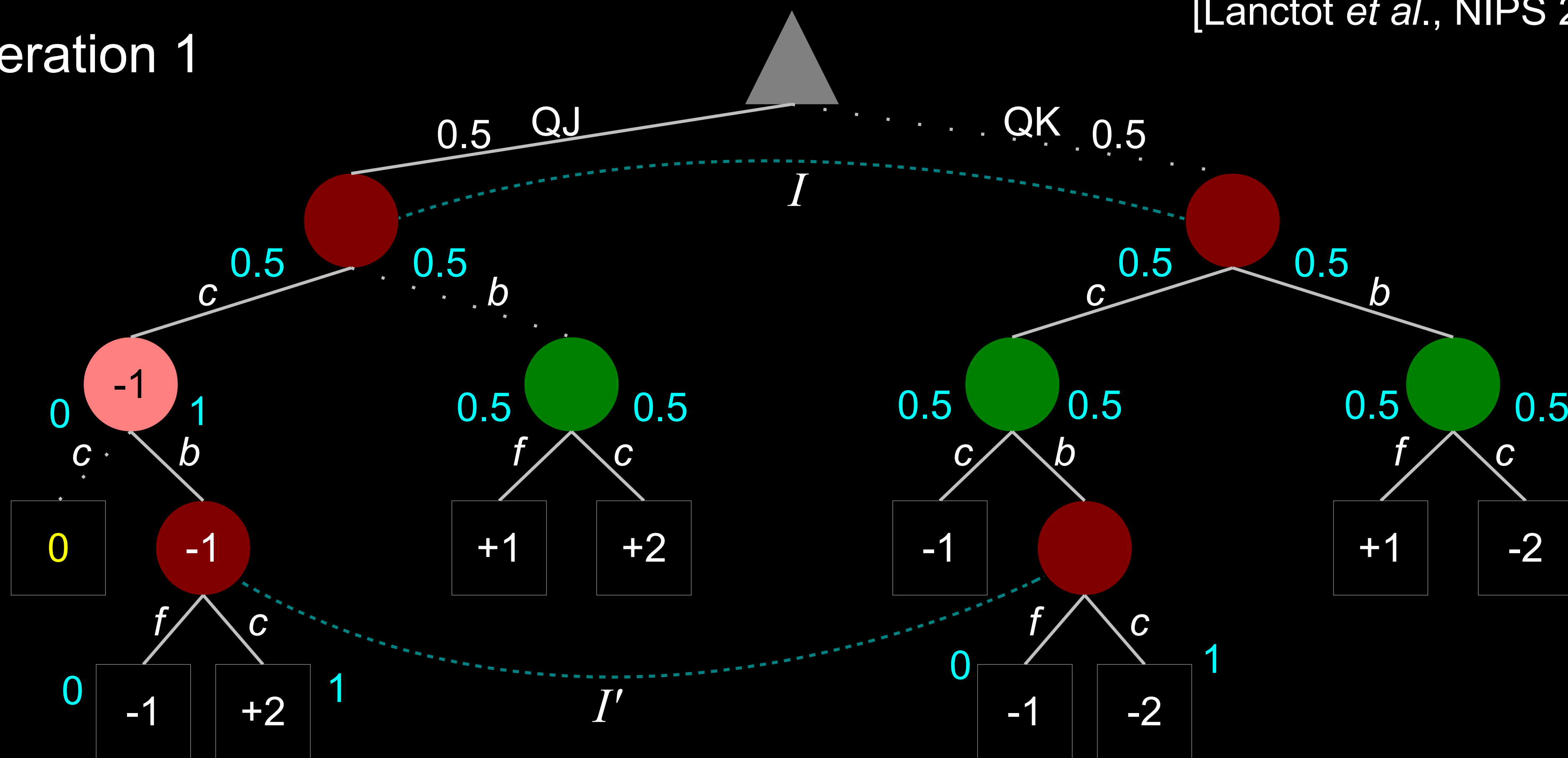


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1

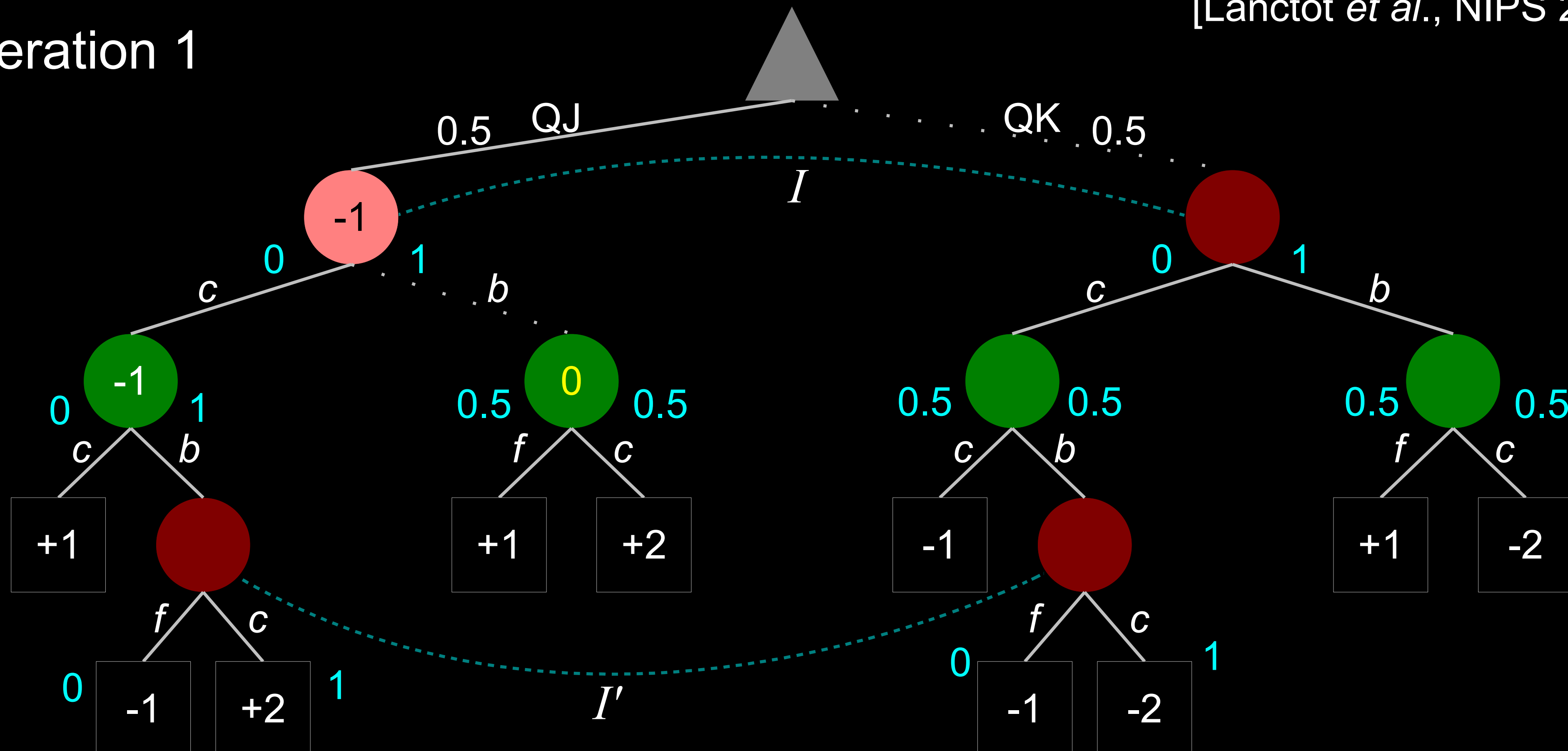


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1

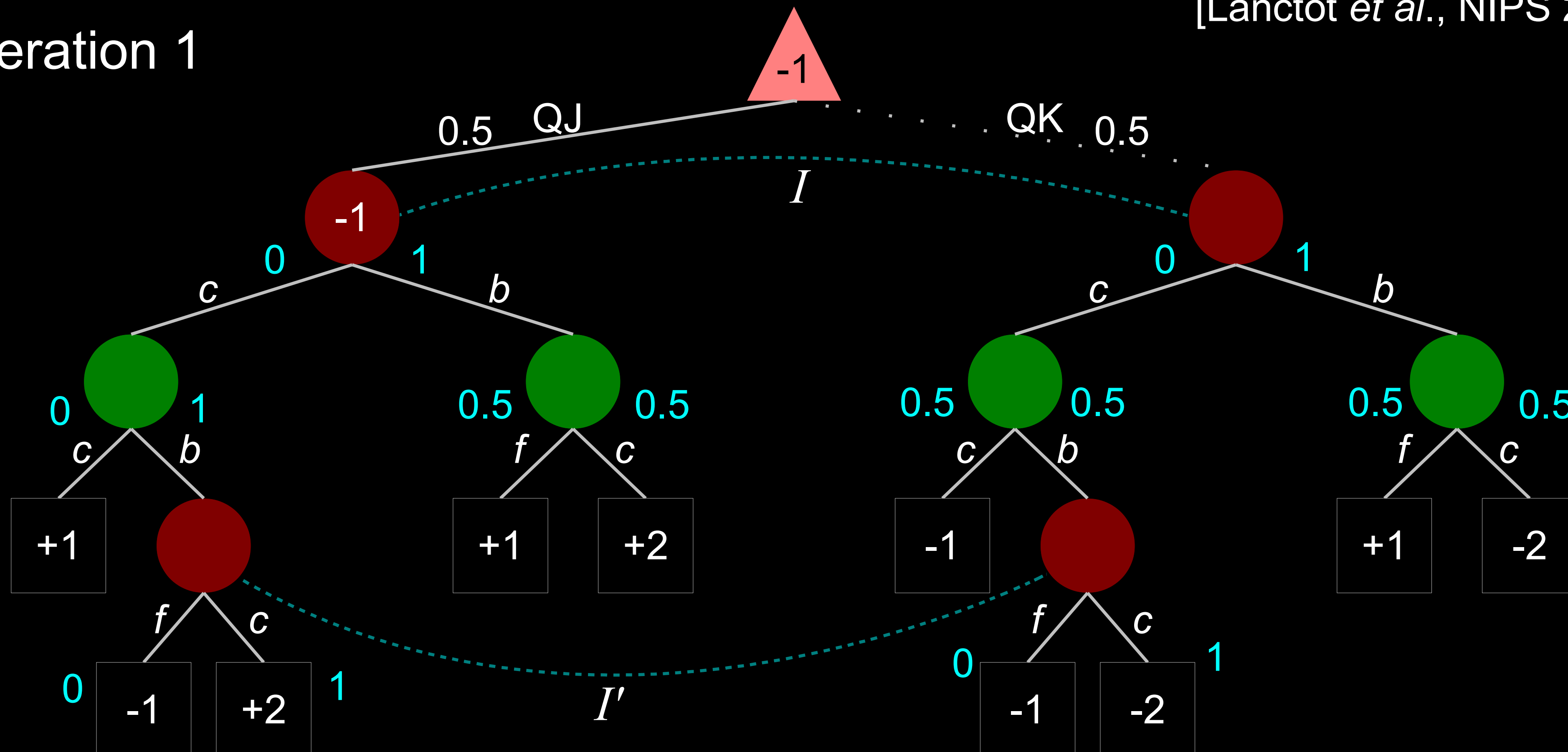


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1

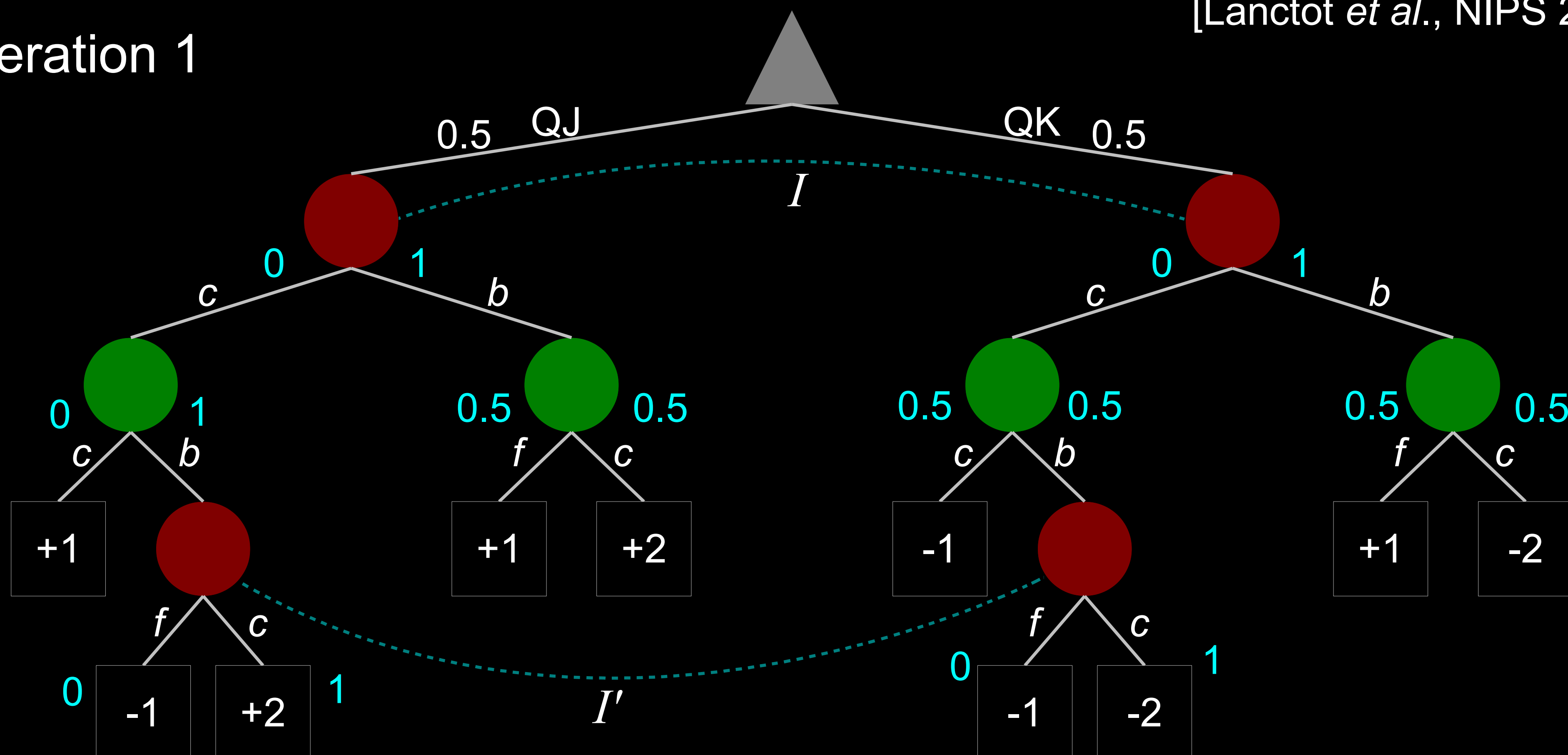


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 1

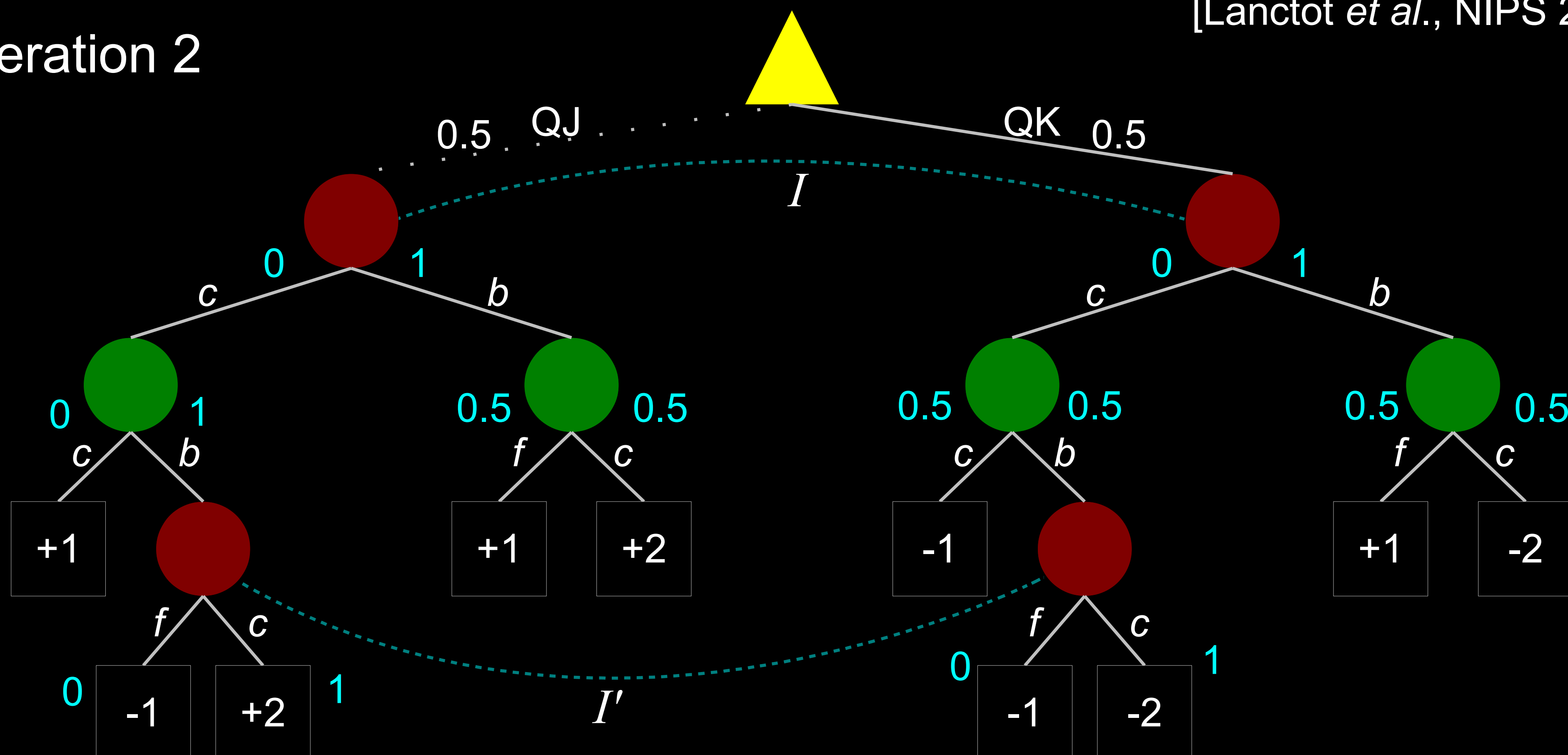


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al*., NIPS 2009]

- Iteration 2

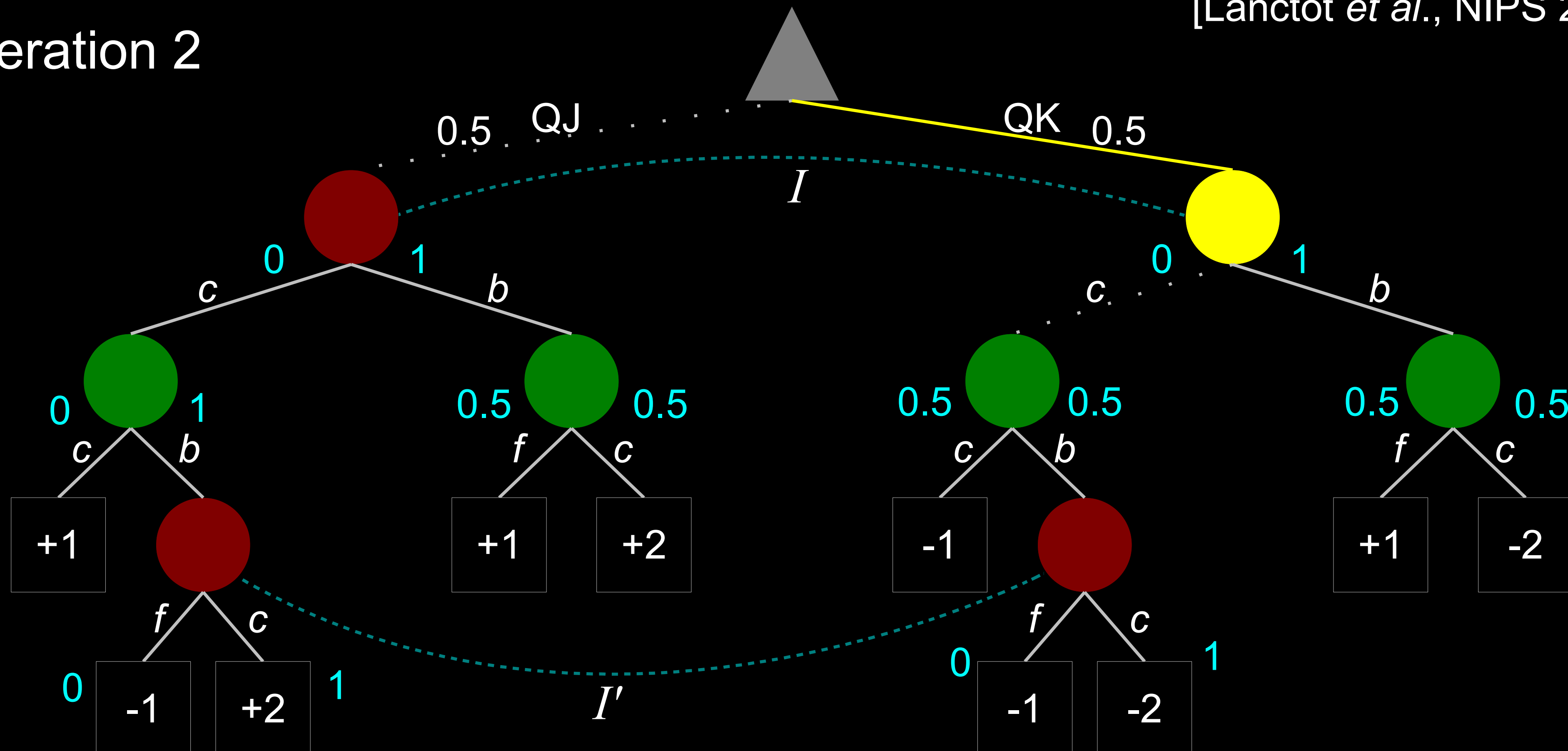


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2

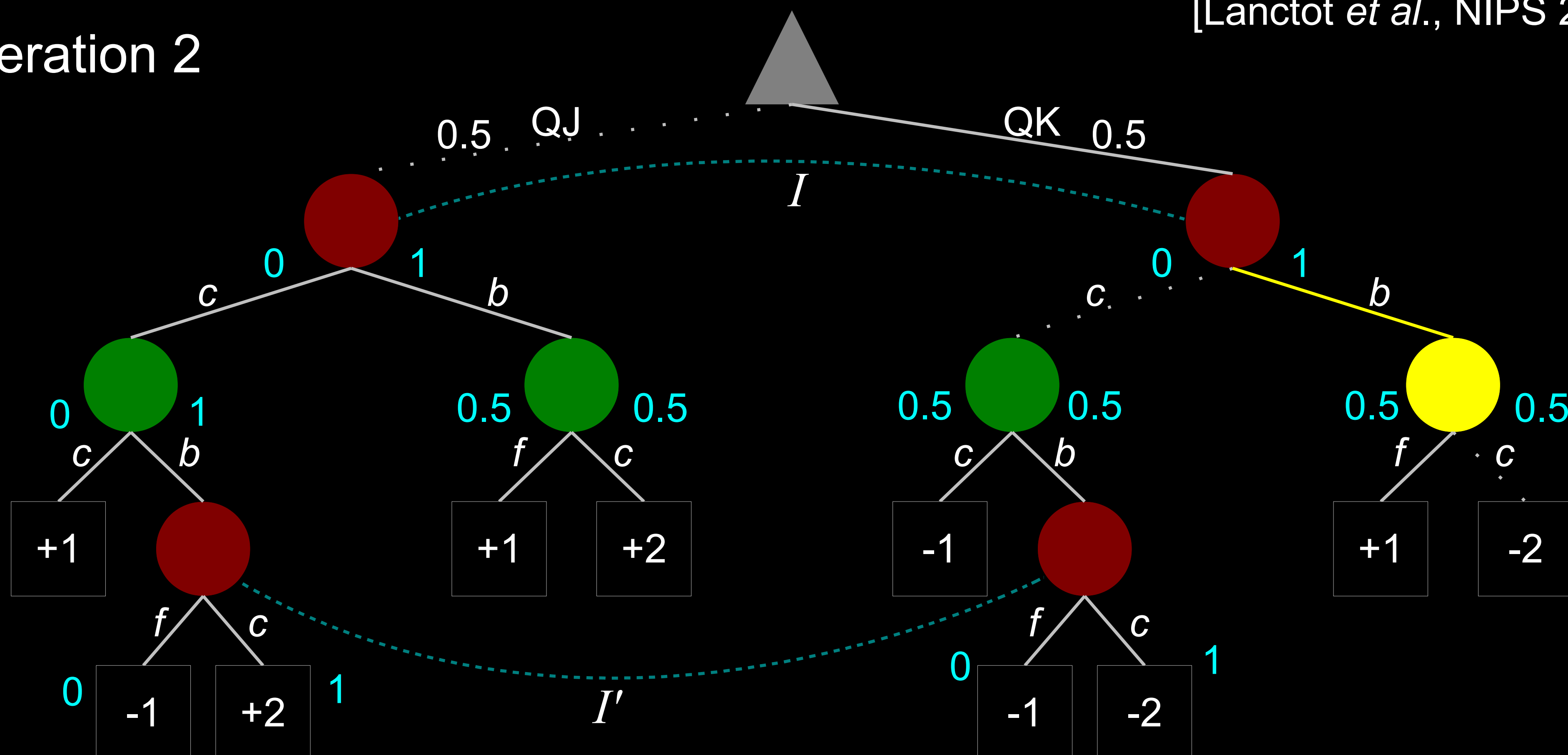


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2

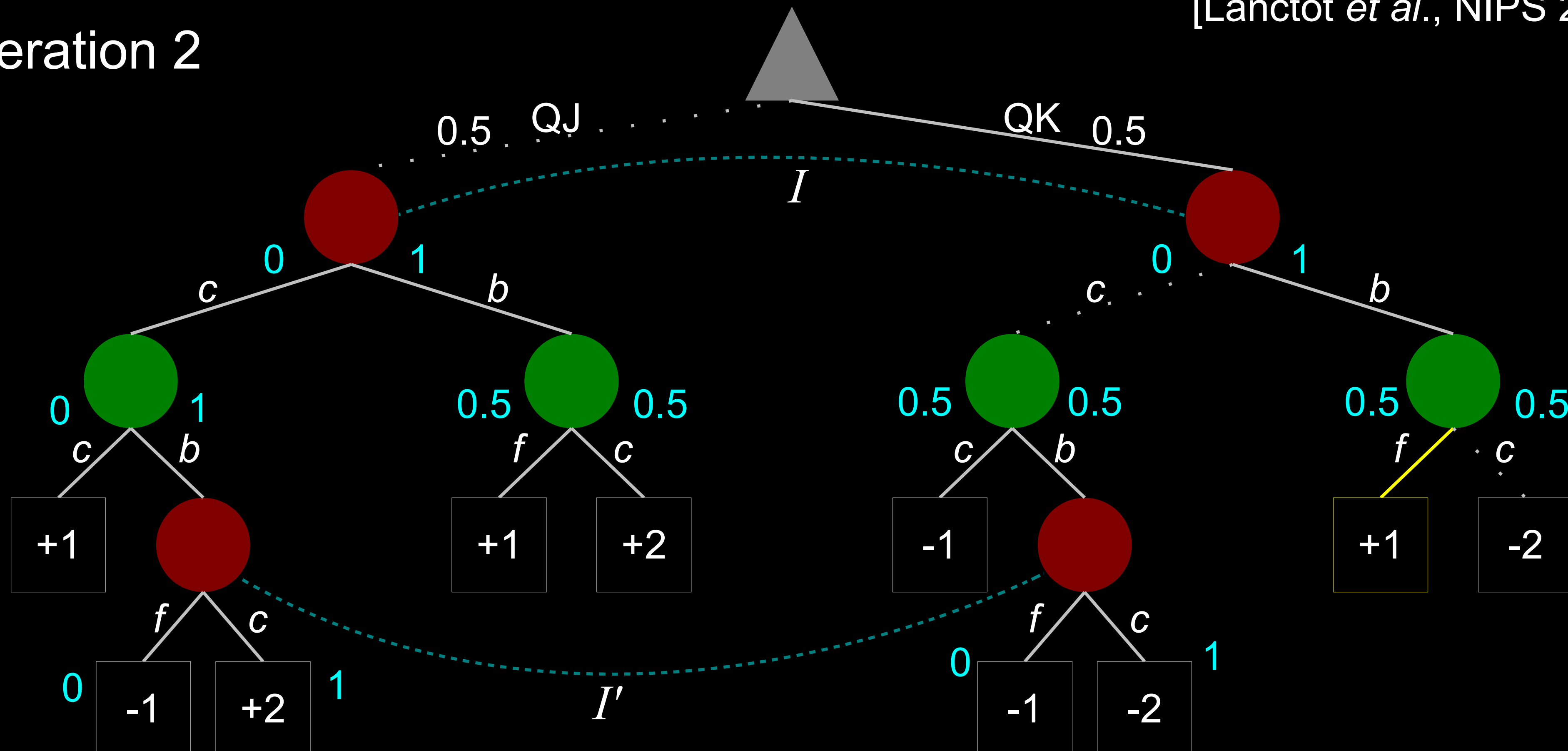


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2

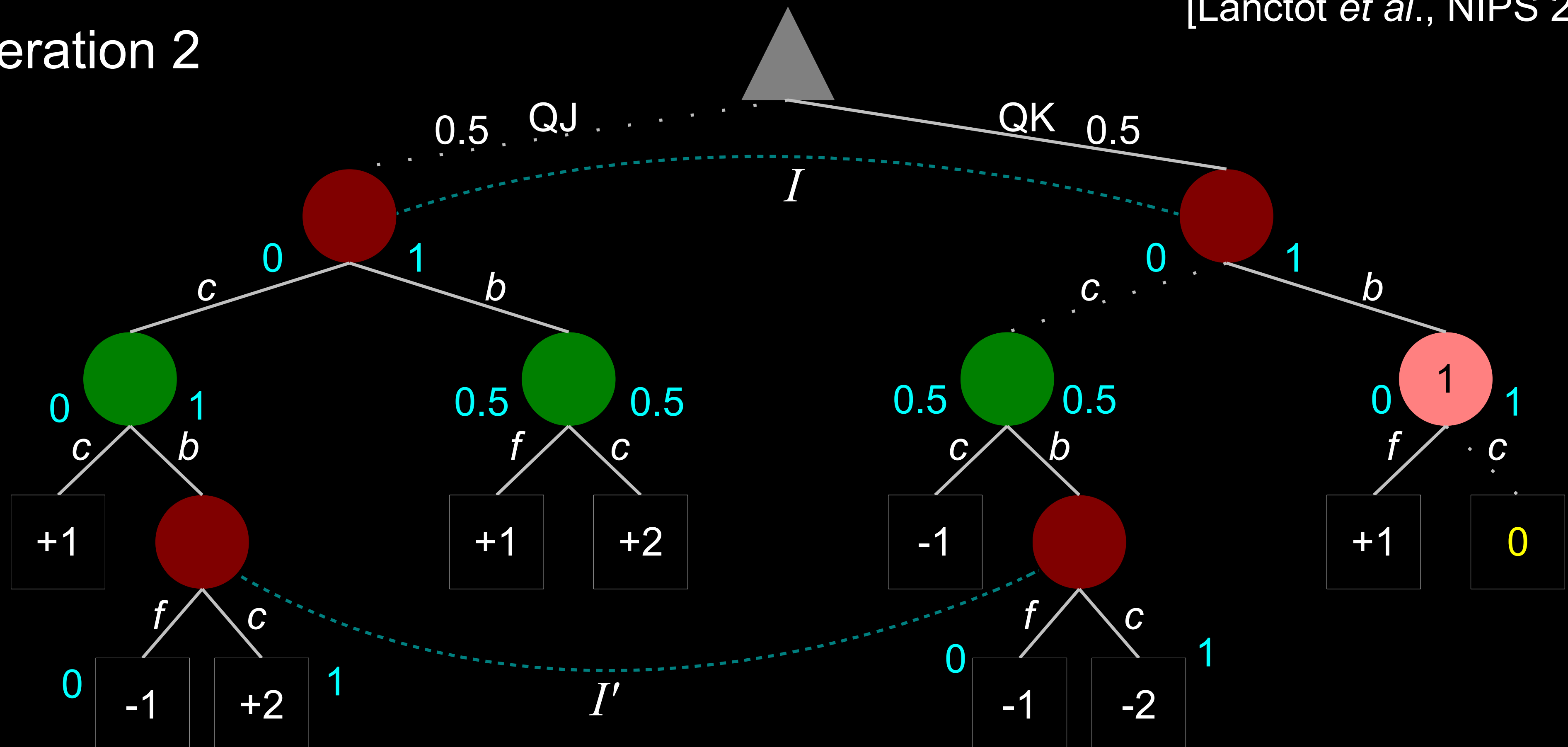


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2

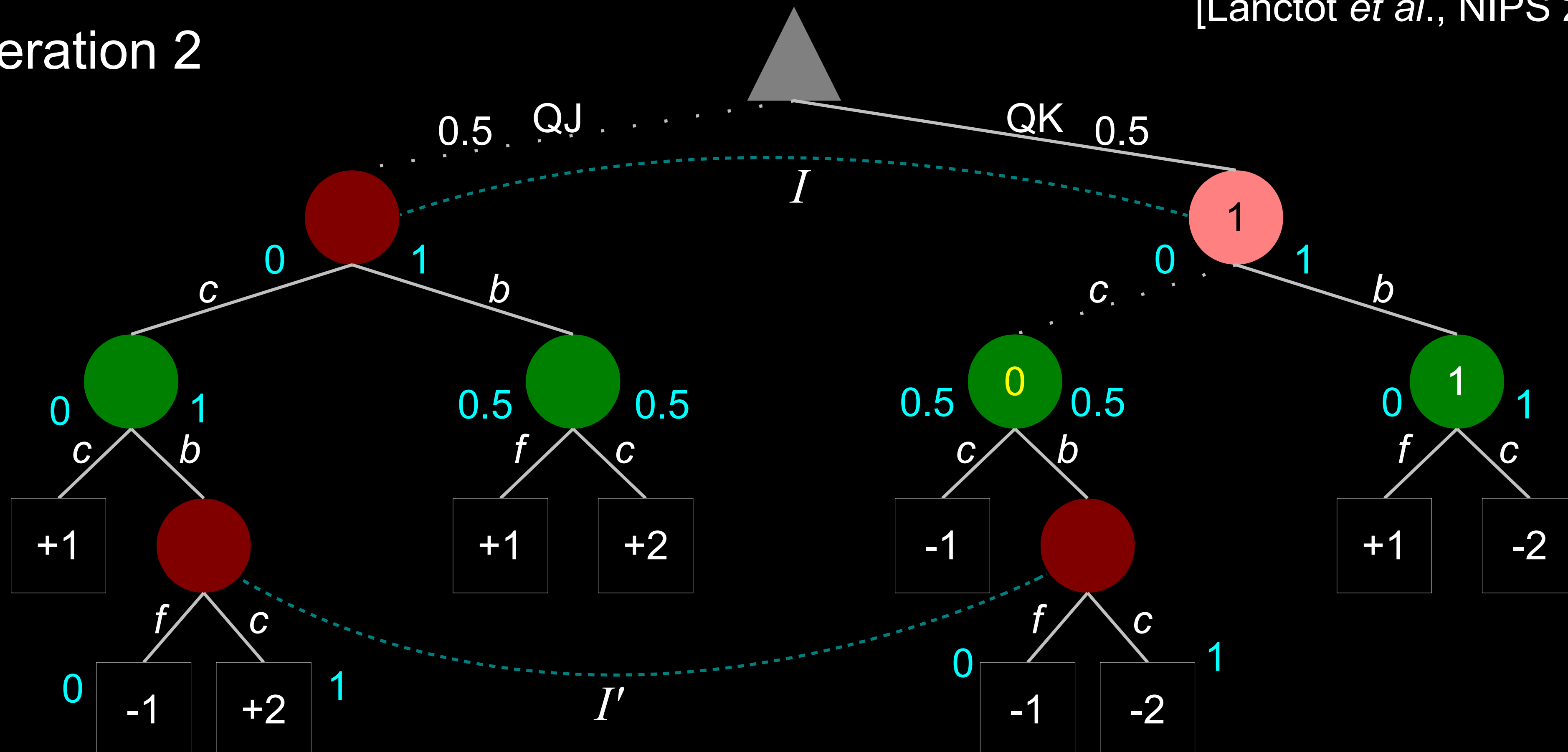


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2

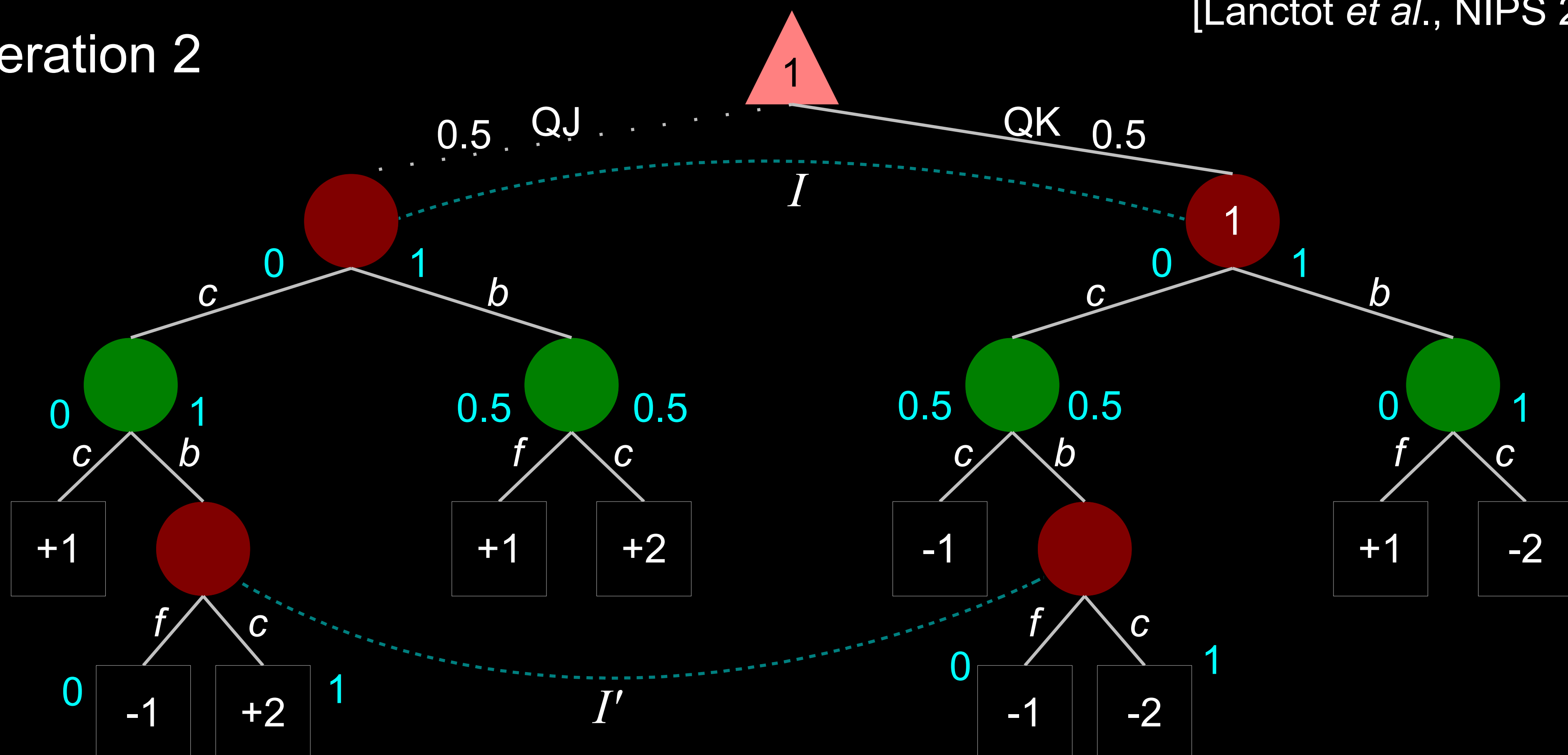


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2

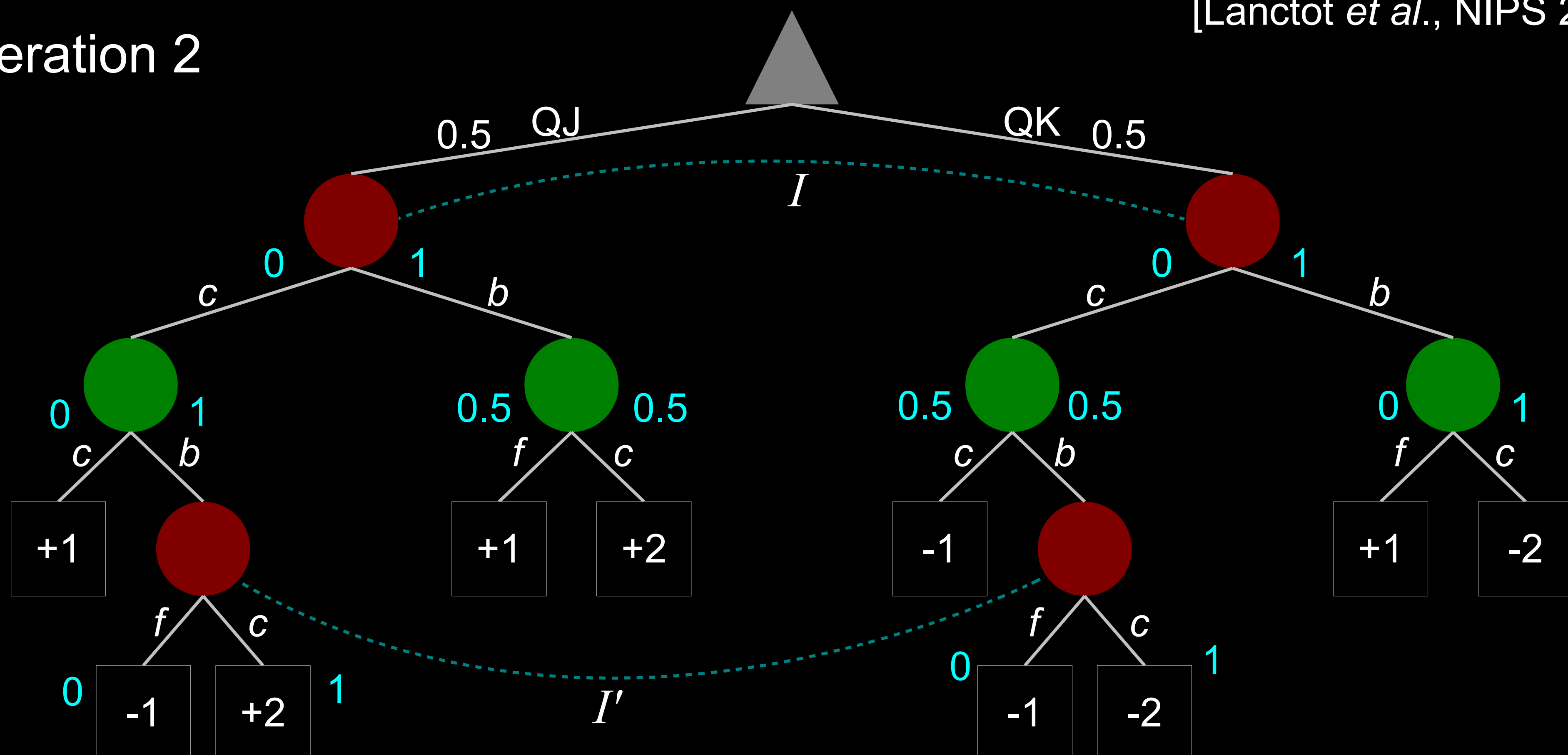


Other Variants: Outcome Sampling

At every node, traverse only one action per iteration

[Lanctot *et al.*, NIPS 2009]

- Iteration 2



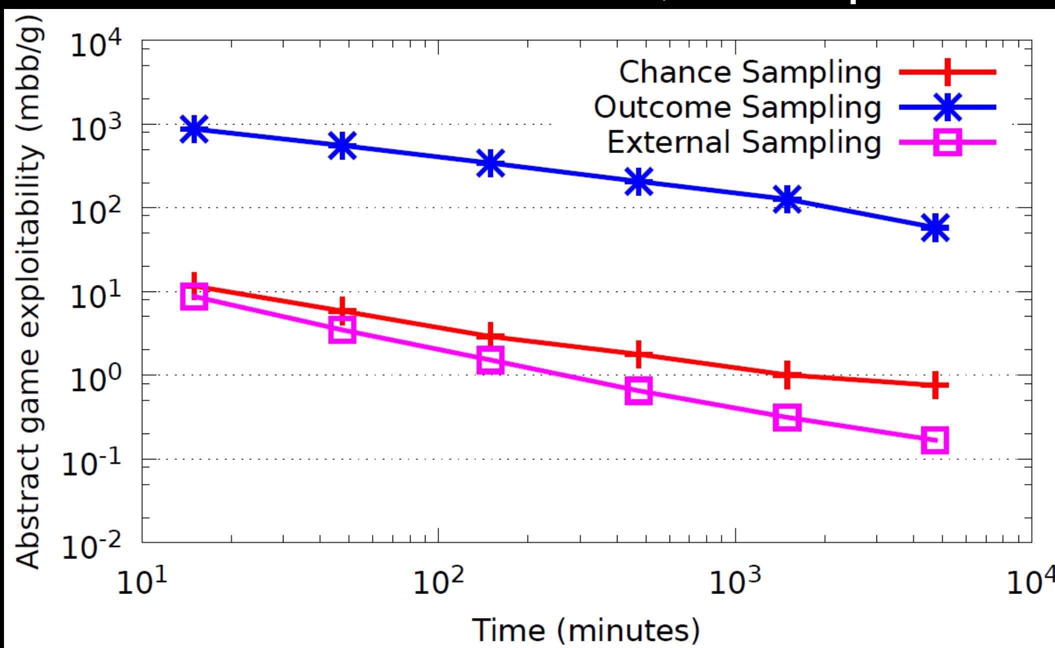
Sampling Variants

- Chance Sampling, External Sampling, and Outcome Sampling all fall under a general Monte Carlo CFR (MCCFR) framework.
- In general, one can choose any scheme for sampling actions (can be domain specific).
- All actions not sampled are always assumed to contribute zero value.

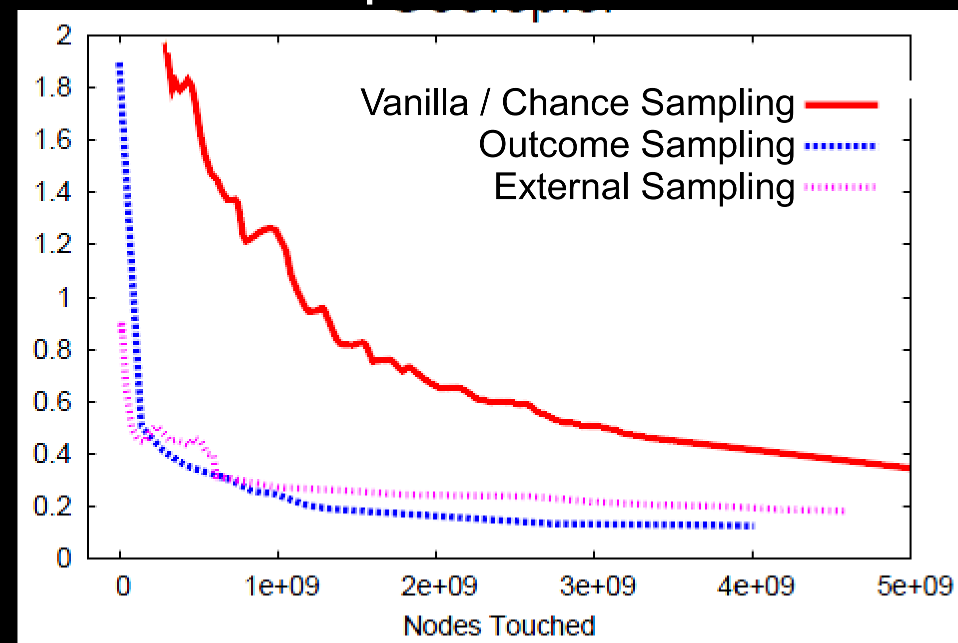
- Iterations required: Vanilla < Chance ≤ External < Outcome
- Time per iteration: Vanilla > Chance > External > Outcome

Chance vs. External vs. Outcome Sampling

2-round No-limit Hold'em, 30 chip stacks



Goofspiel with 6 cards



- Card abstraction applied to reduce chance branching factor to 5 at each chance node

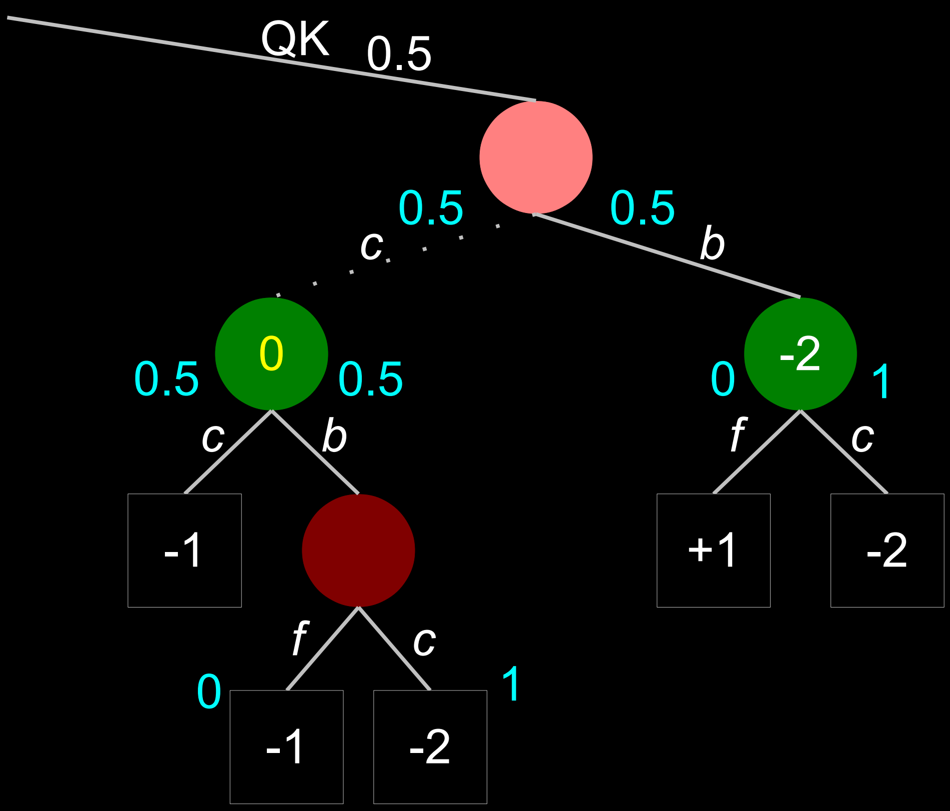
[Lanctot *et al.*, NIPS 2009]

Outline

- Extensive-form Games
  - Examples
  - Terminology
  - Solution concepts
- Counterfactual Regret Minimization (CFR)
  - Base algorithm for solving extensive-form games
  - Older variants
- New, Faster CFR Variants
  - Probing
  - Public Chance Sampling
  - Average Strategy Sampling
- Conclusions and Future Work

Sampling in General

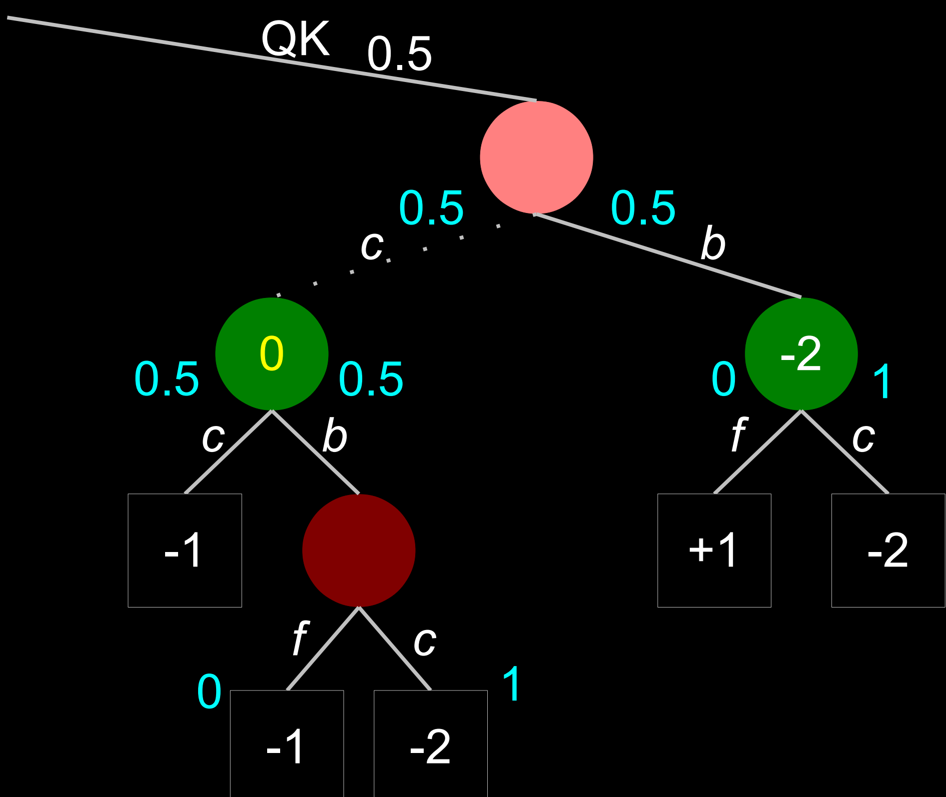
Example: Outcome Sampling



EV = 0.5(0) + 0.5(-2) / probability of sampling *b*

Sampling in General

Example: Outcome Sampling



· EV is a sampled value and is an unbiased (equal in expectation) estimate of the true value at this node [Lanctot *et al*., NIPS 2009].

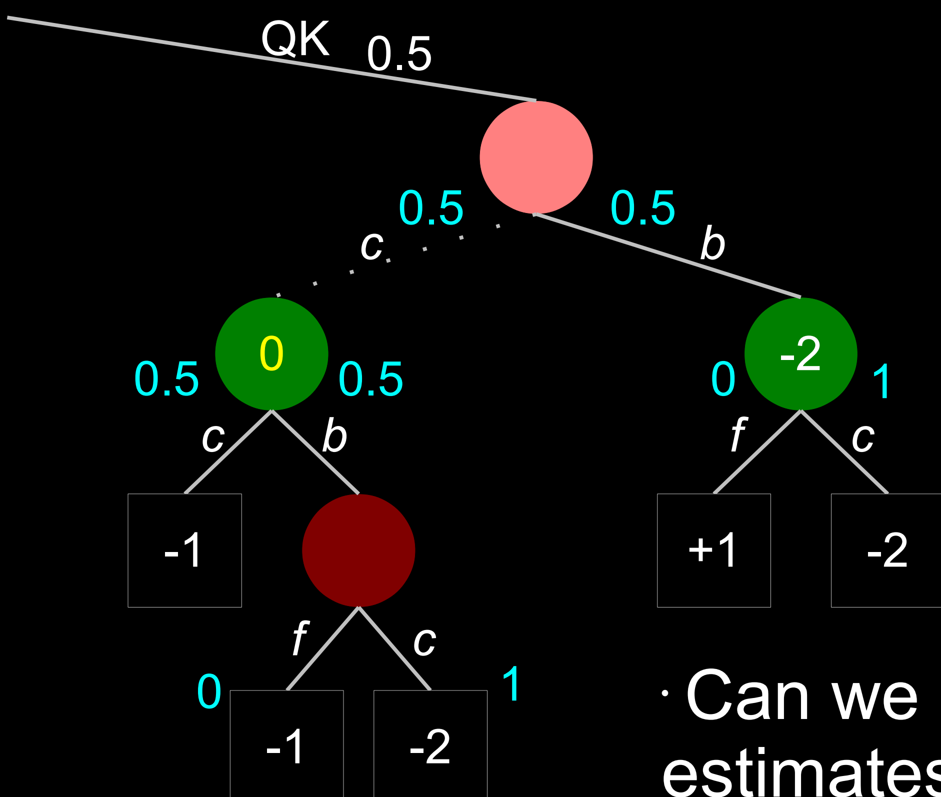
· However, EV is a noisy estimate of the true value (variance).

· Can prove that estimates with lower variance gives better bound on number of iterations required to converge to a given solution quality [G *et al*., AAAI 2012].

EV = 0.5(0) + 0.5(-2) / probability of sampling *b*

Sampling in General

Example: Outcome Sampling



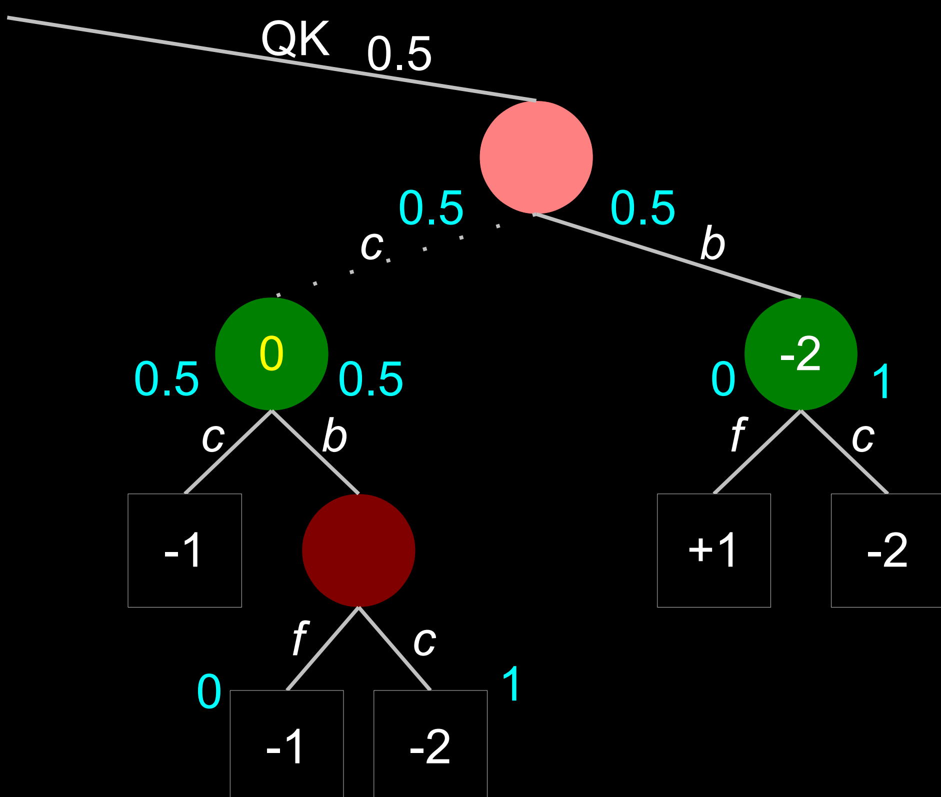
- EV is a sampled value and is an unbiased (equal in expectation) estimate of the true value at this node [Lanctot *et al*., NIPS 2009].
- However, EV is a noisy estimate of the true value (variance).
- Can prove that estimates with lower variance gives better bound on number of iterations required to converge to a given solution quality [G *et al*., AAAI 2012].

- Can we **quickly** produce unbiased estimates of EV with lower variance?

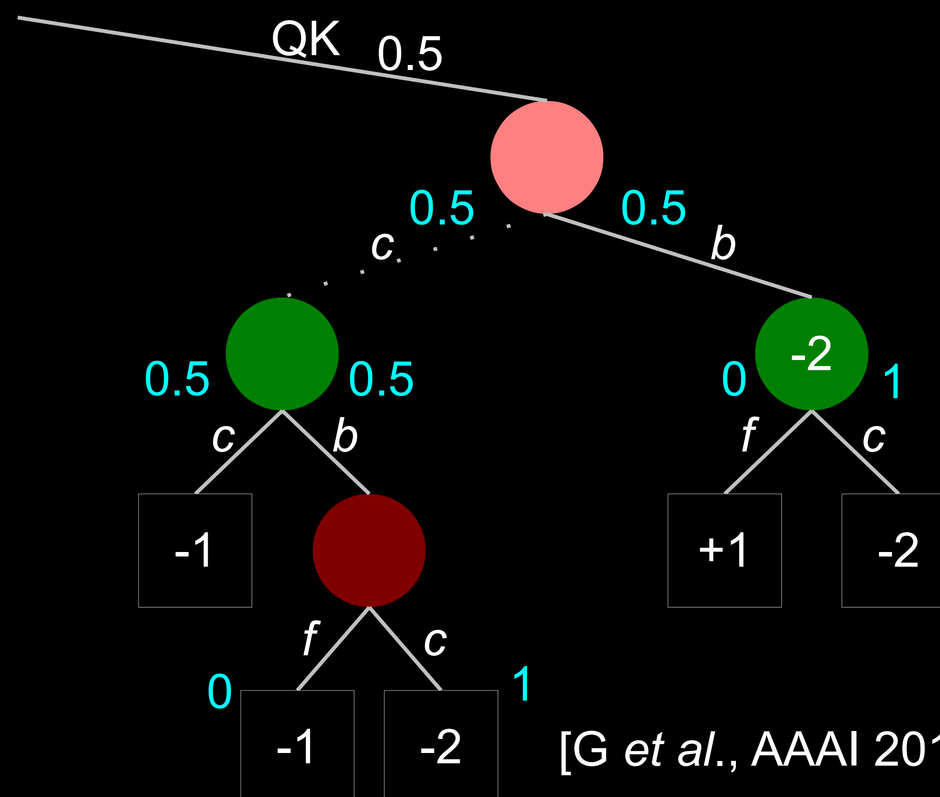
EV = 0.5(0) + 0.5(-2) / probability of sampling *b*

Probing

Example: Outcome Sampling



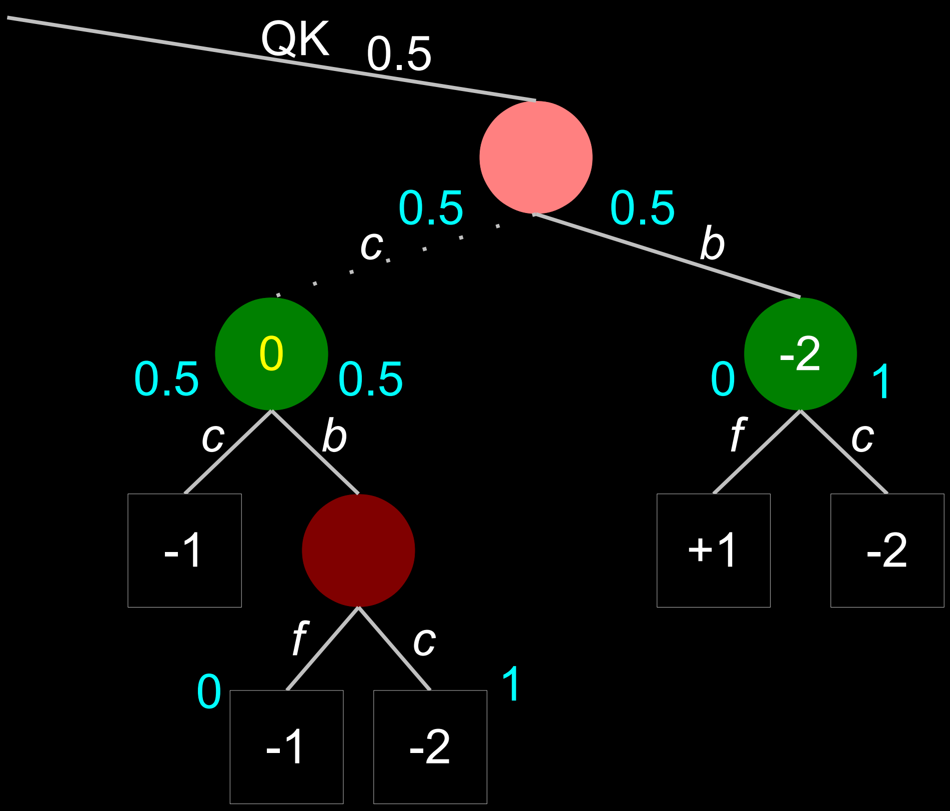
Example: Outcome + Probing



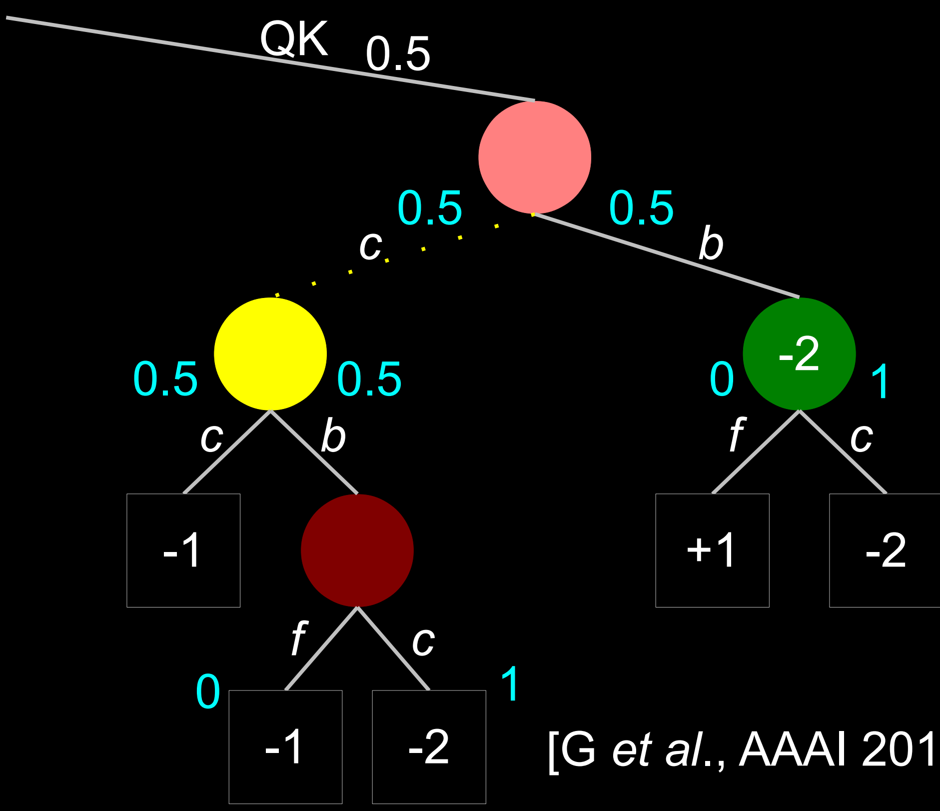
[G *et al.*, AAAI 2012]

Probing

Example: Outcome Sampling



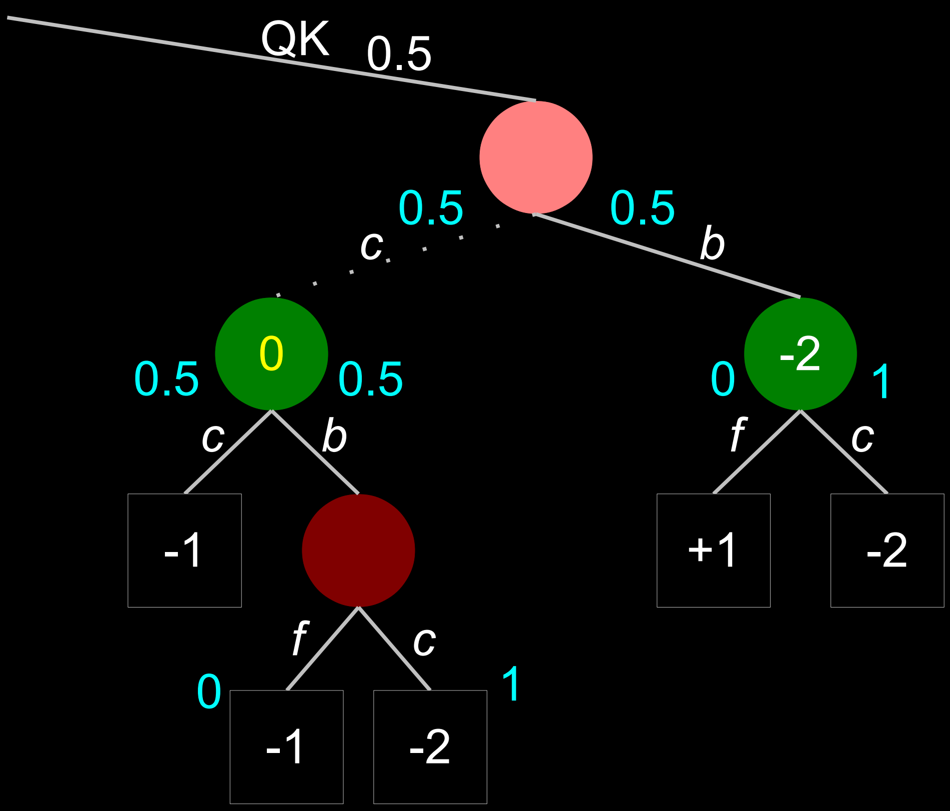
Example: Outcome + Probing



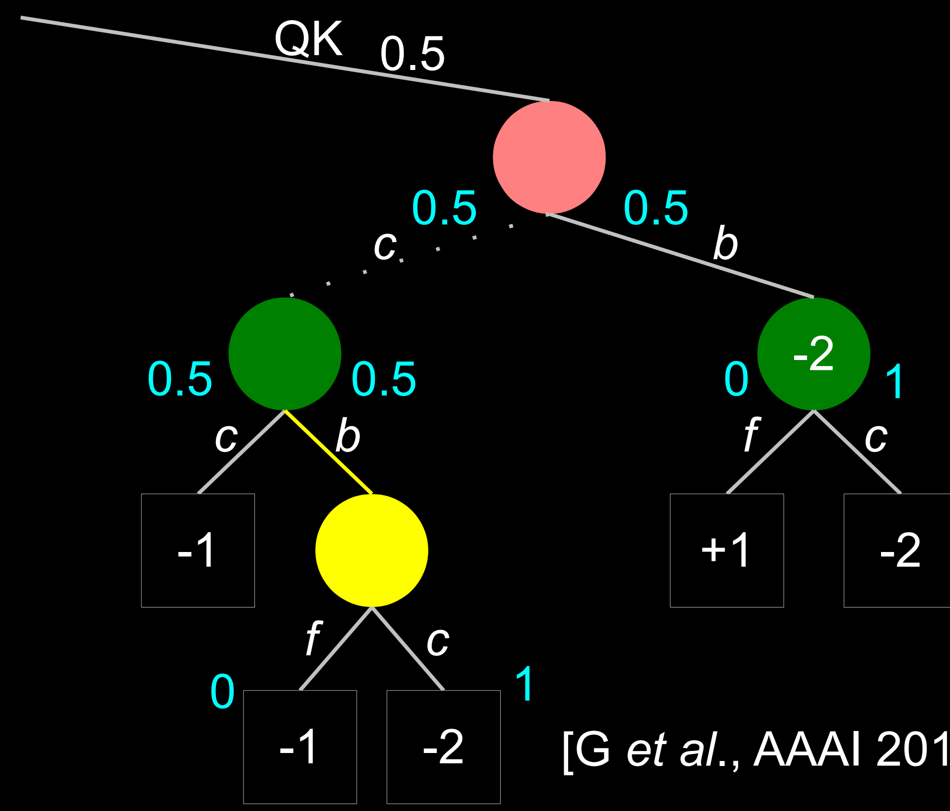
[G *et al.*, AAAI 2012]

Probing

Example: Outcome Sampling



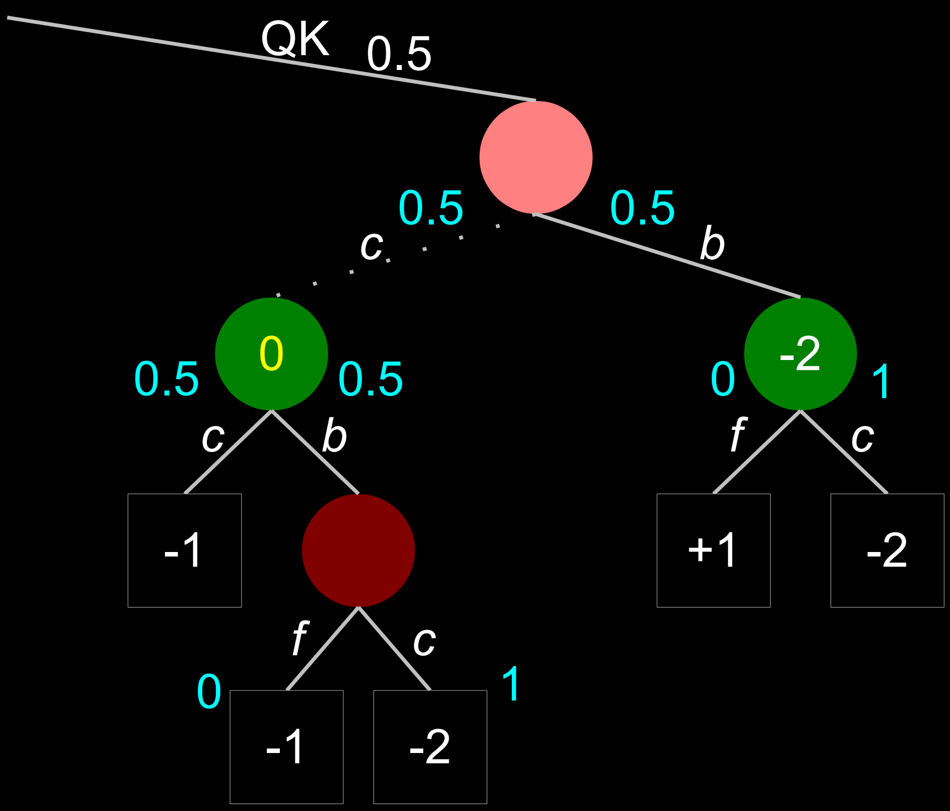
Example: Outcome + Probing



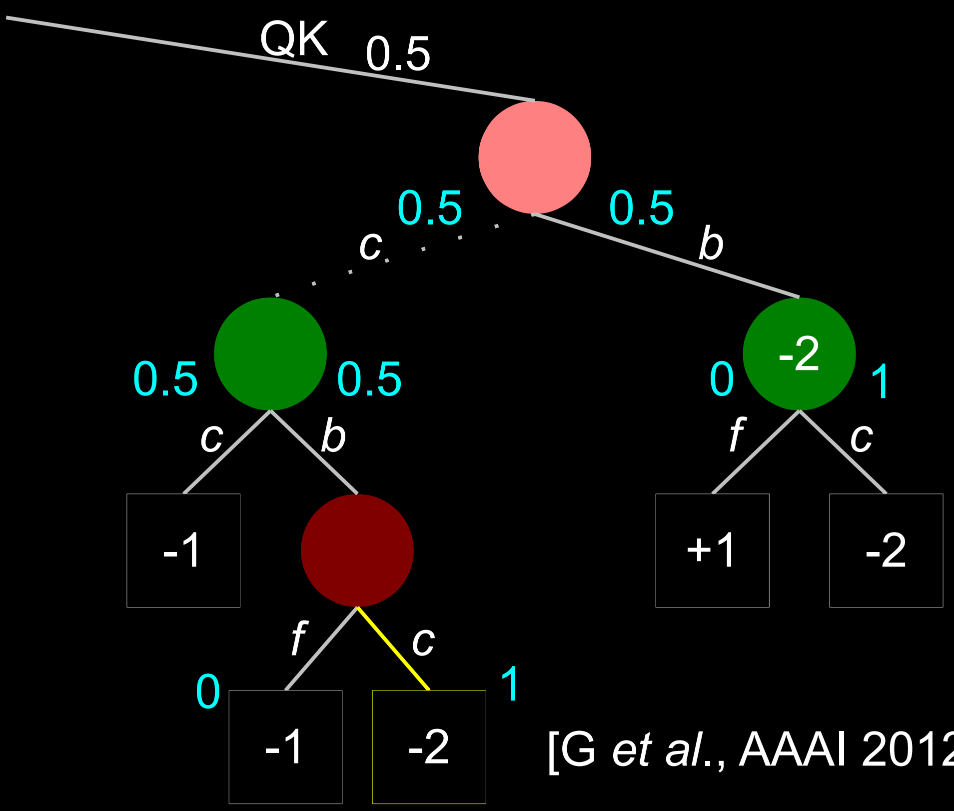
[G *et al.*, AAAI 2012]

Probing

Example: Outcome Sampling



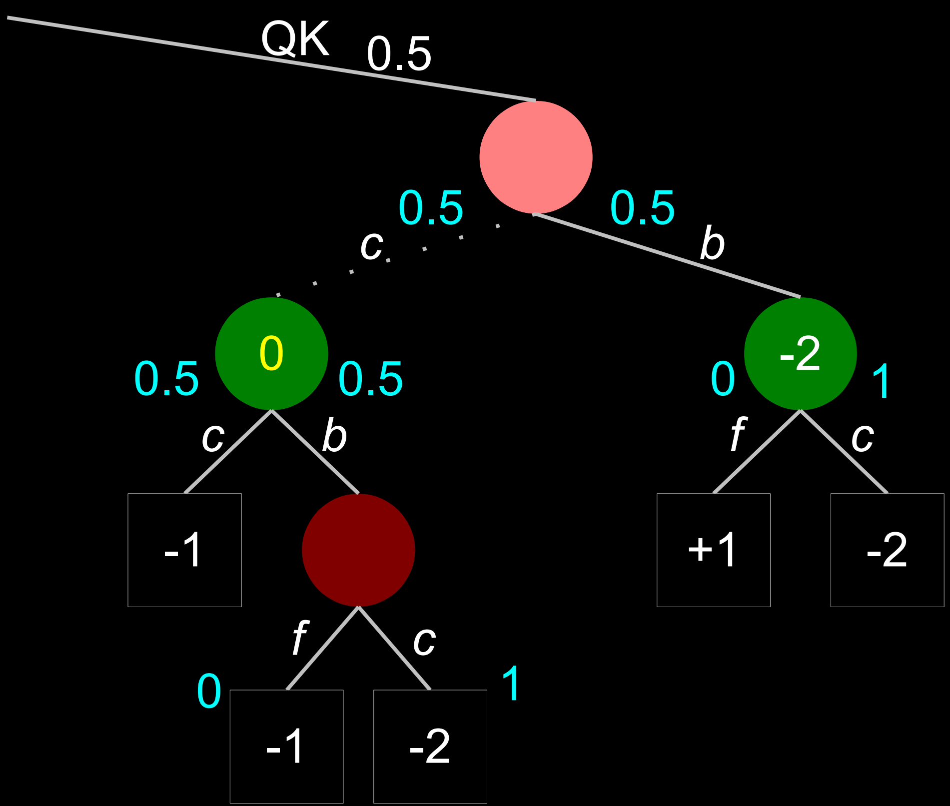
Example: Outcome + Probing



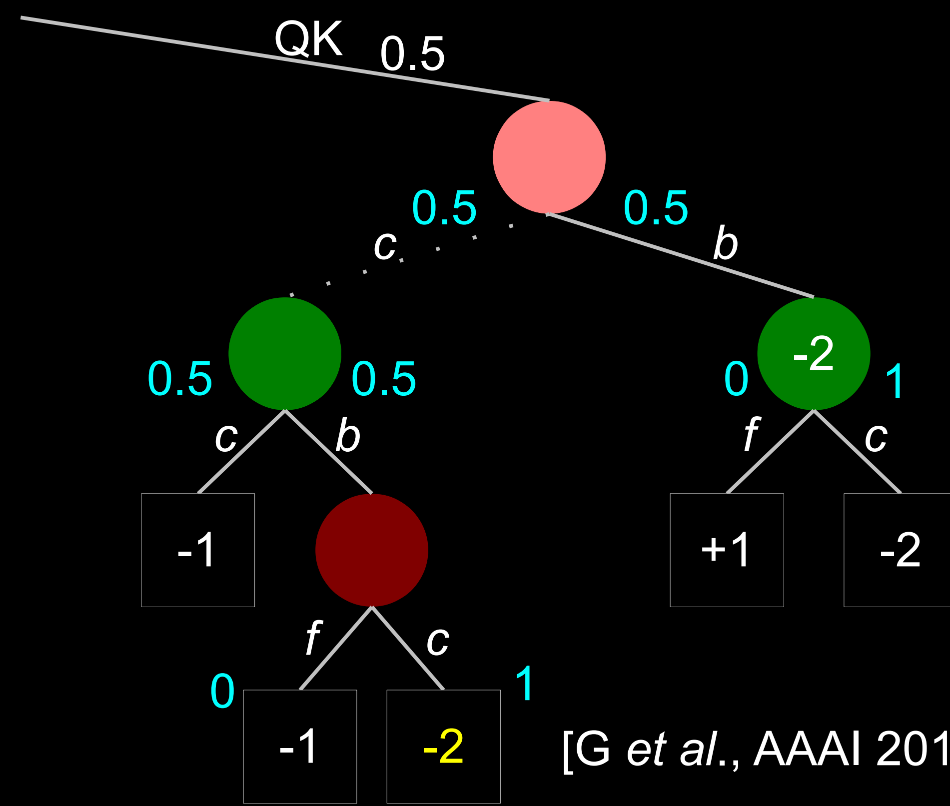
[G *et al.*, AAAI 2012]

Probing

Example: Outcome Sampling



Example: Outcome + Probing



[G *et al.*, AAAI 2012]

EV = 0.5(-2) + 0.5(-2)

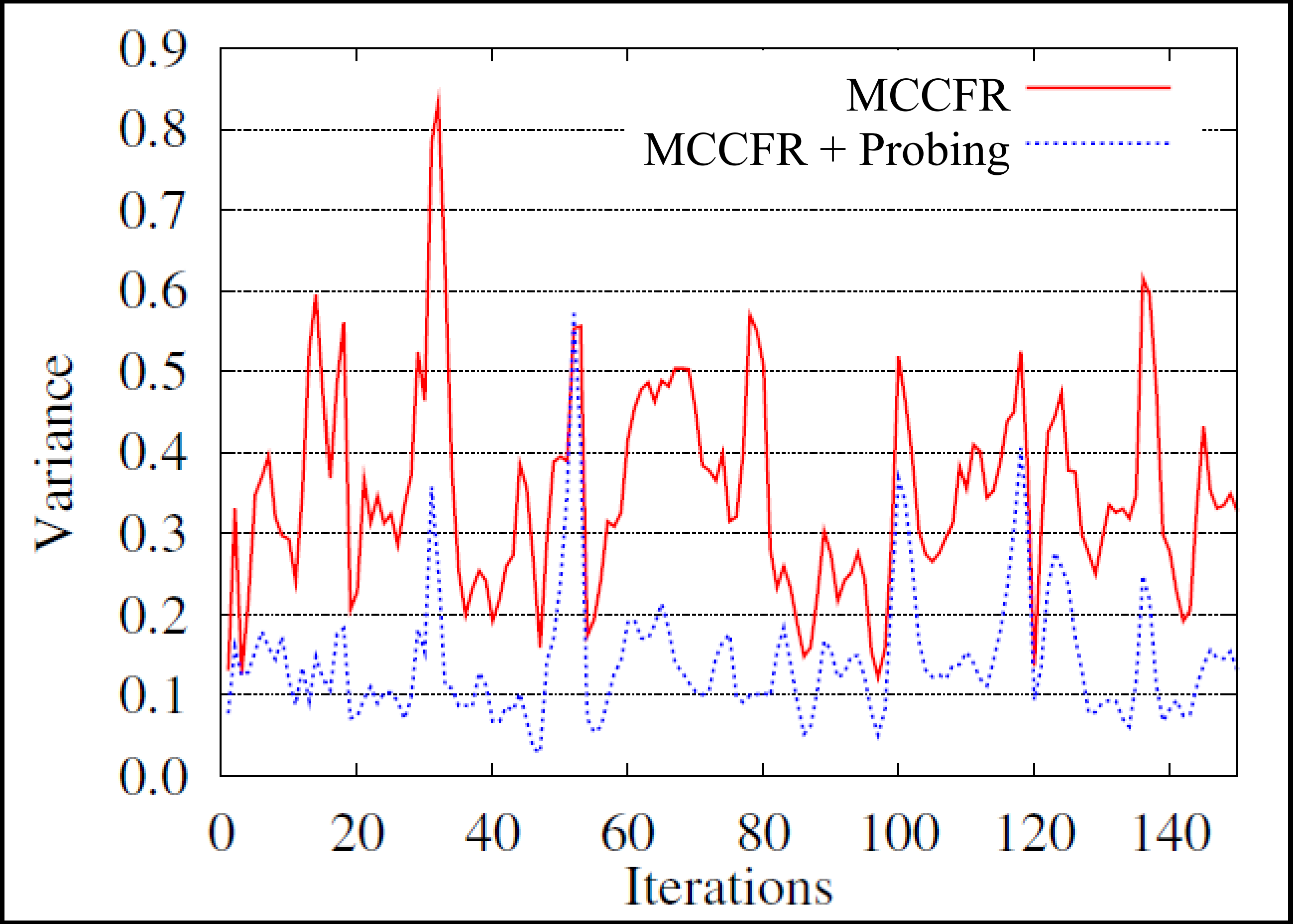
Probing

- Probing can be added to any Monte Carlo sampling algorithm.
  - Time per iteration: MCCFR < MCCFR + Probing (barely)
  - Iterations required: MCCFR > MCCFR + Probing
- Probing algorithms live outside of the Monte Carlo family of algorithms (but still provably converge).
- Probing is more expensive than simply assigning zero value, but no updates are performed during probe traversal (cheaper than regular traversal).
- All of our probing experiments use domain-specific sampling schemes.

Probing Experiments

Goofspiel with 6 cards

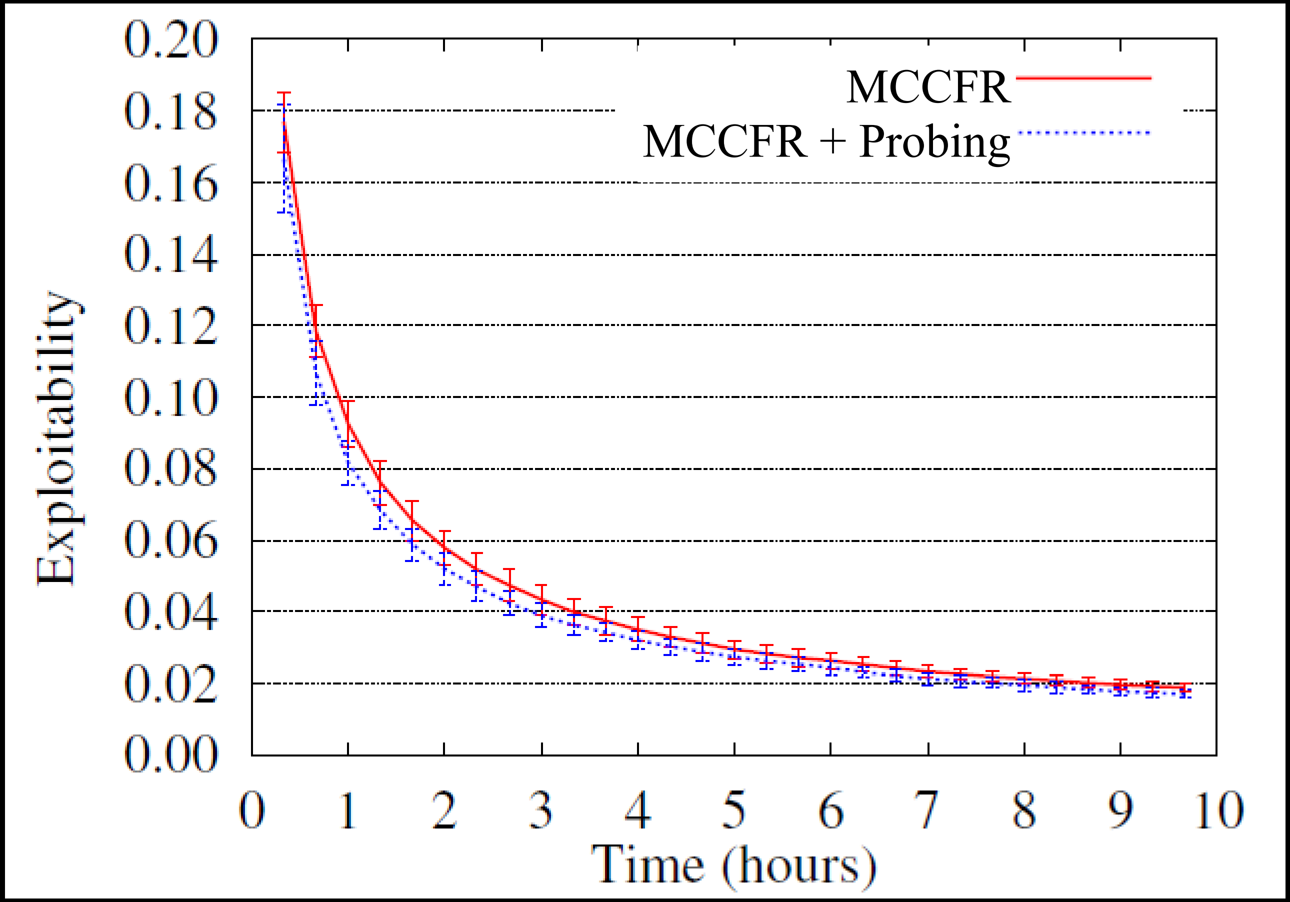
[G *et al.*, AAAI 2012]



MCCFR = Always sample the highest and lowest cards remaining, sample all other actions with probability 0.5.

Probing Experiments

Bluff with 2 dice each

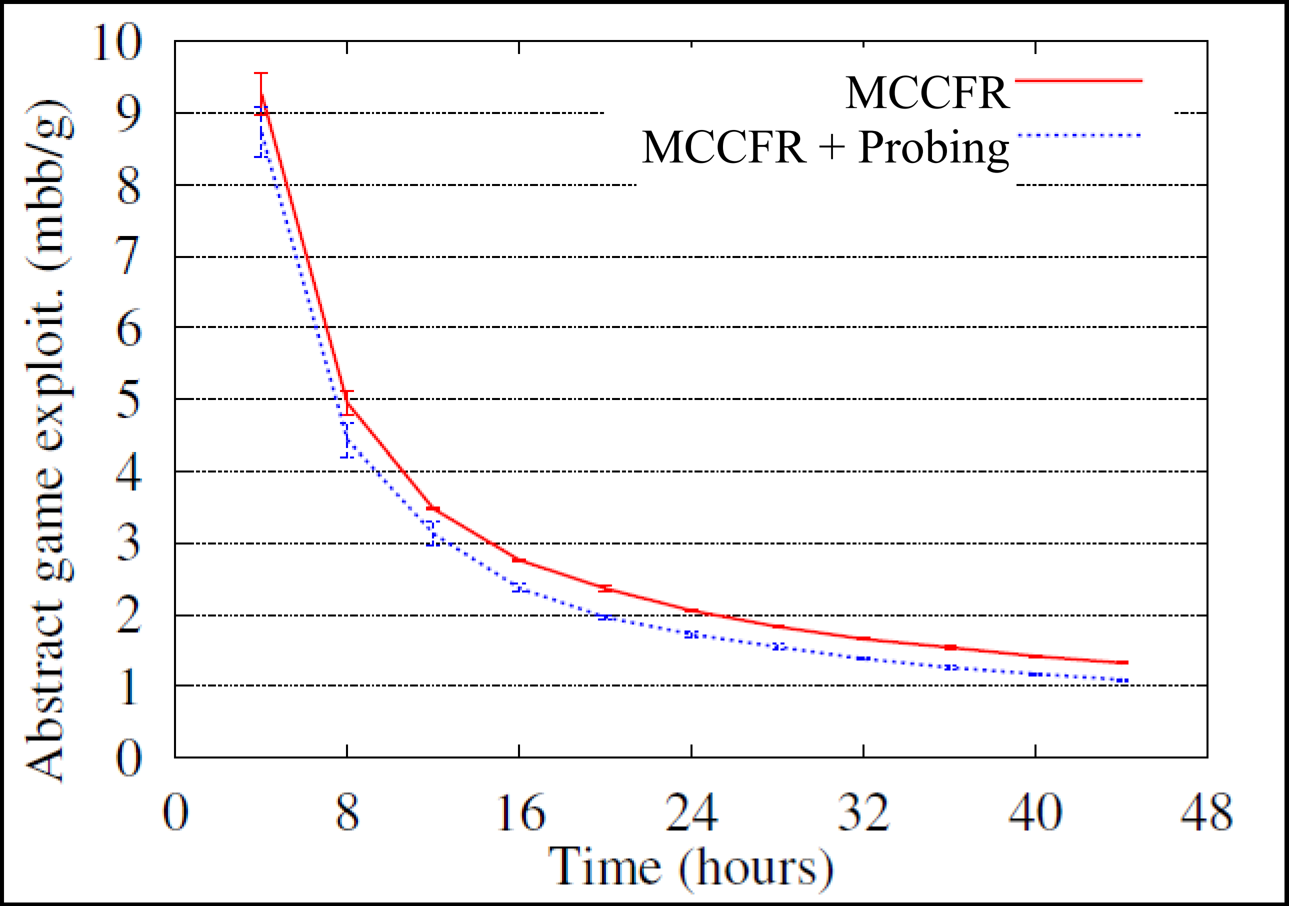


[G *et al*., AAAI 2012]

MCCFR = Always sample 1-5, 2-5, 1-6, 2-6, and for each die-face *x* rolled, *n*-*x* for all valid *n*. All other actions sampled with probability 0.5.

Probing Experiments

Limit Texas Hold'em with card abstraction (10s)

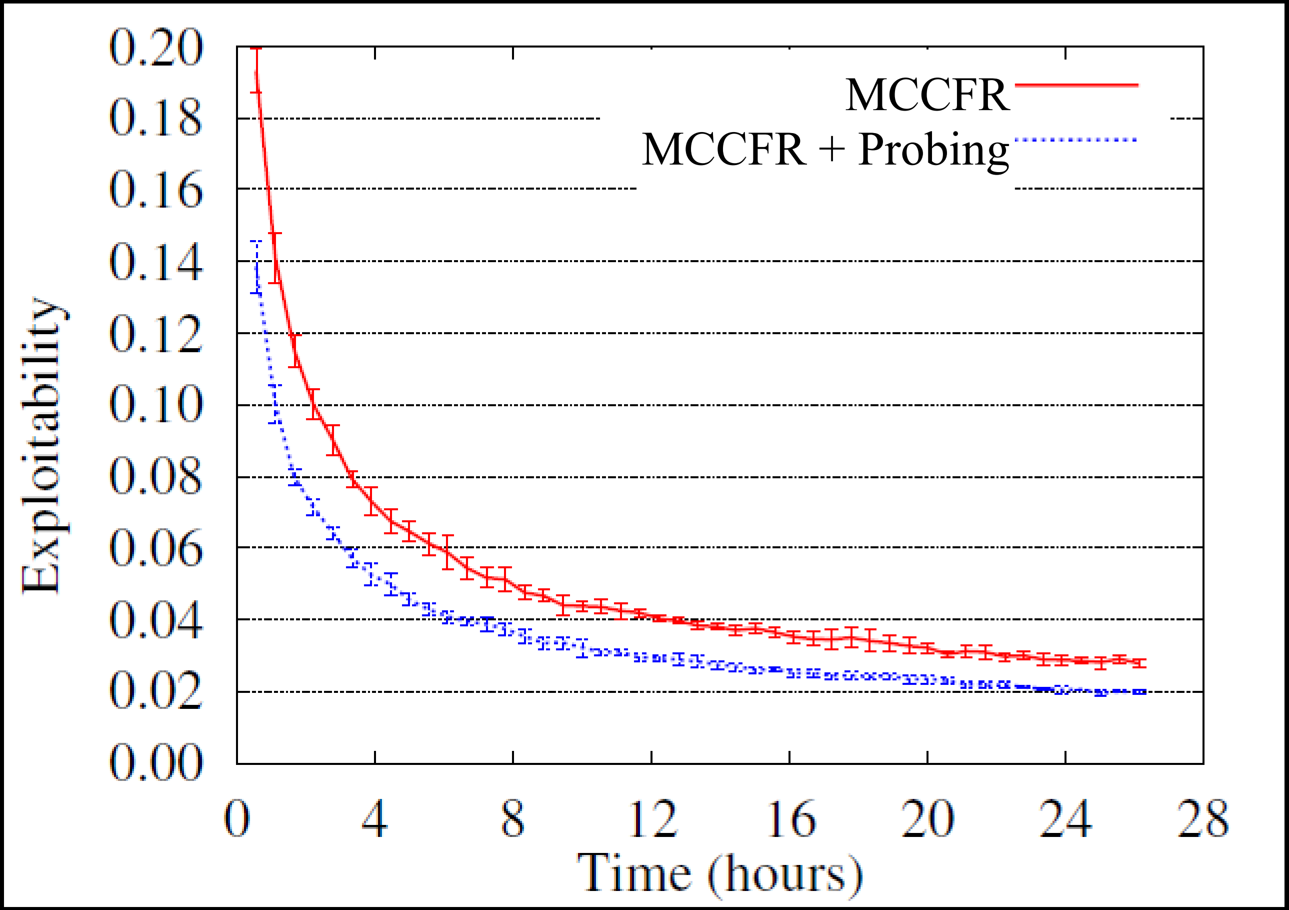


[G *et al*., AAAI 2012]

MCCFR = Always sample fold and raise, sample call with probability 0.5.

Probing Experiments

Goofspiel with 7 cards



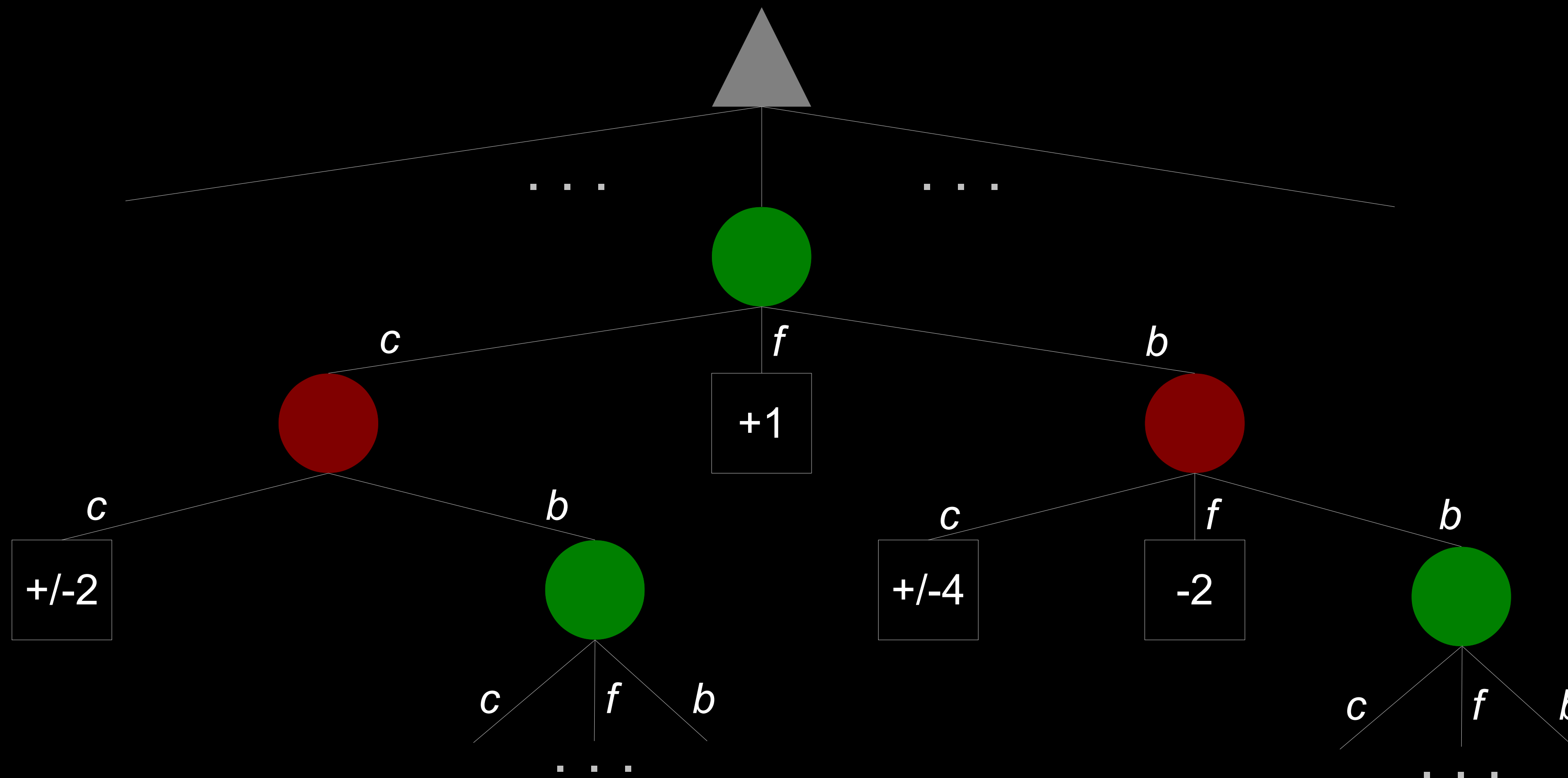
[G *et al*., AAAI 2012]

MCCFR = Always sample the highest and lowest cards remaining, sample all other actions with probability 0.5.

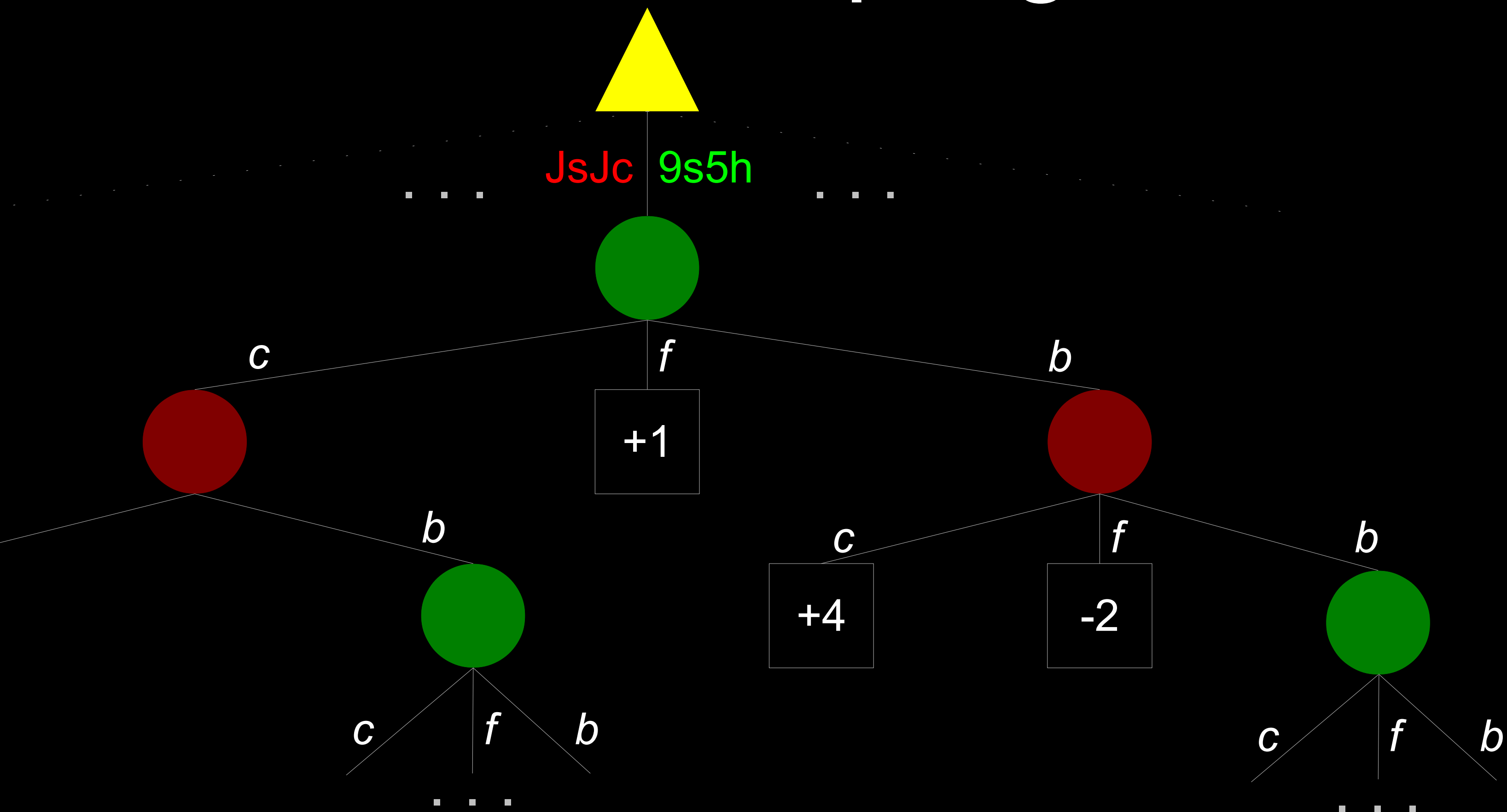
Probing Experiments

- Limitation: Probing does not help when combined with Chance Sampling or External Sampling.
  - These are the best sampling algorithms for Bluff and many poker games.
  - Adding probing to our poker-specific MCCFR algorithm is still slower than Chance Sampling and External Sampling.

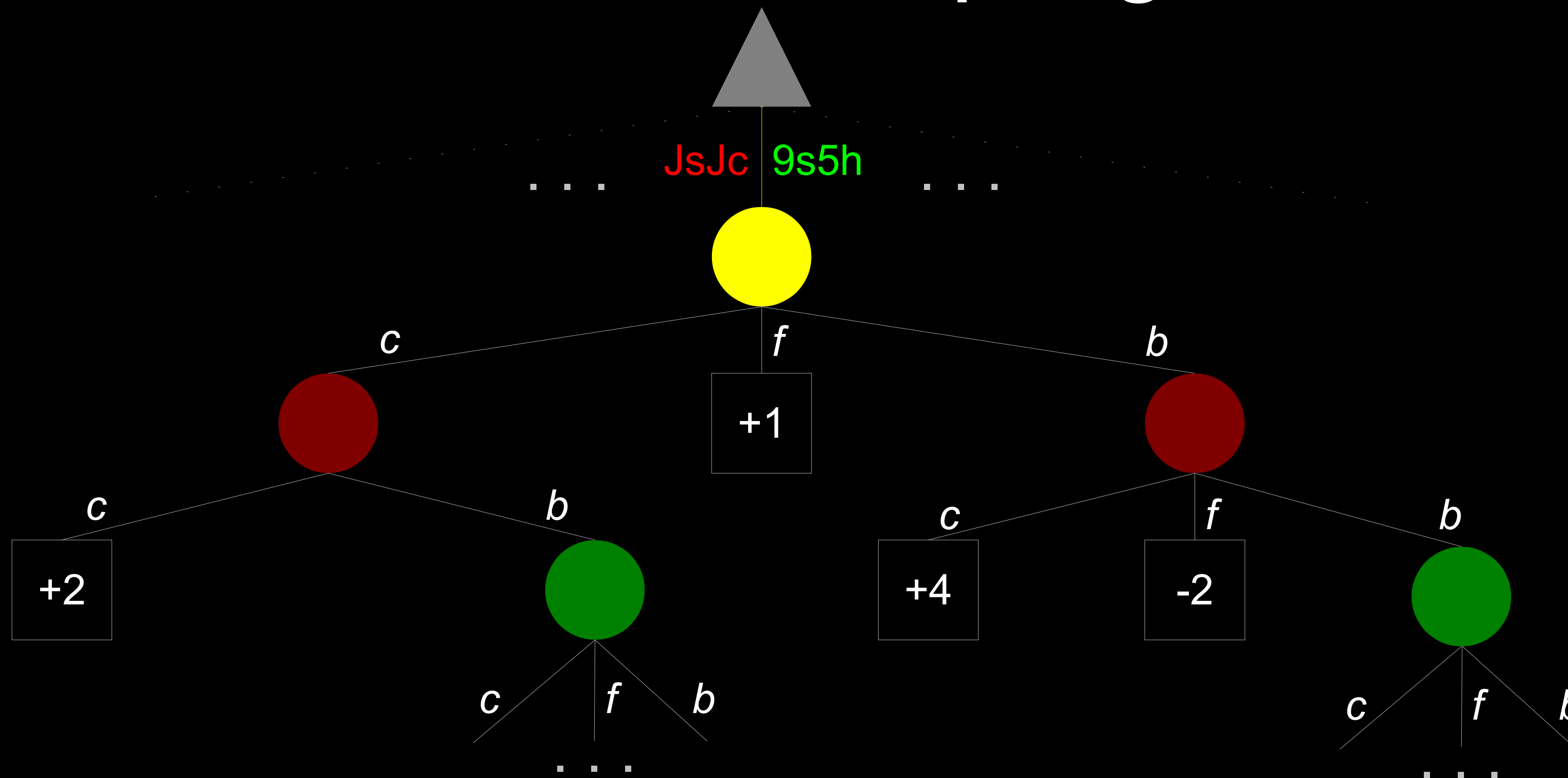
1-Round Limit Hold'em



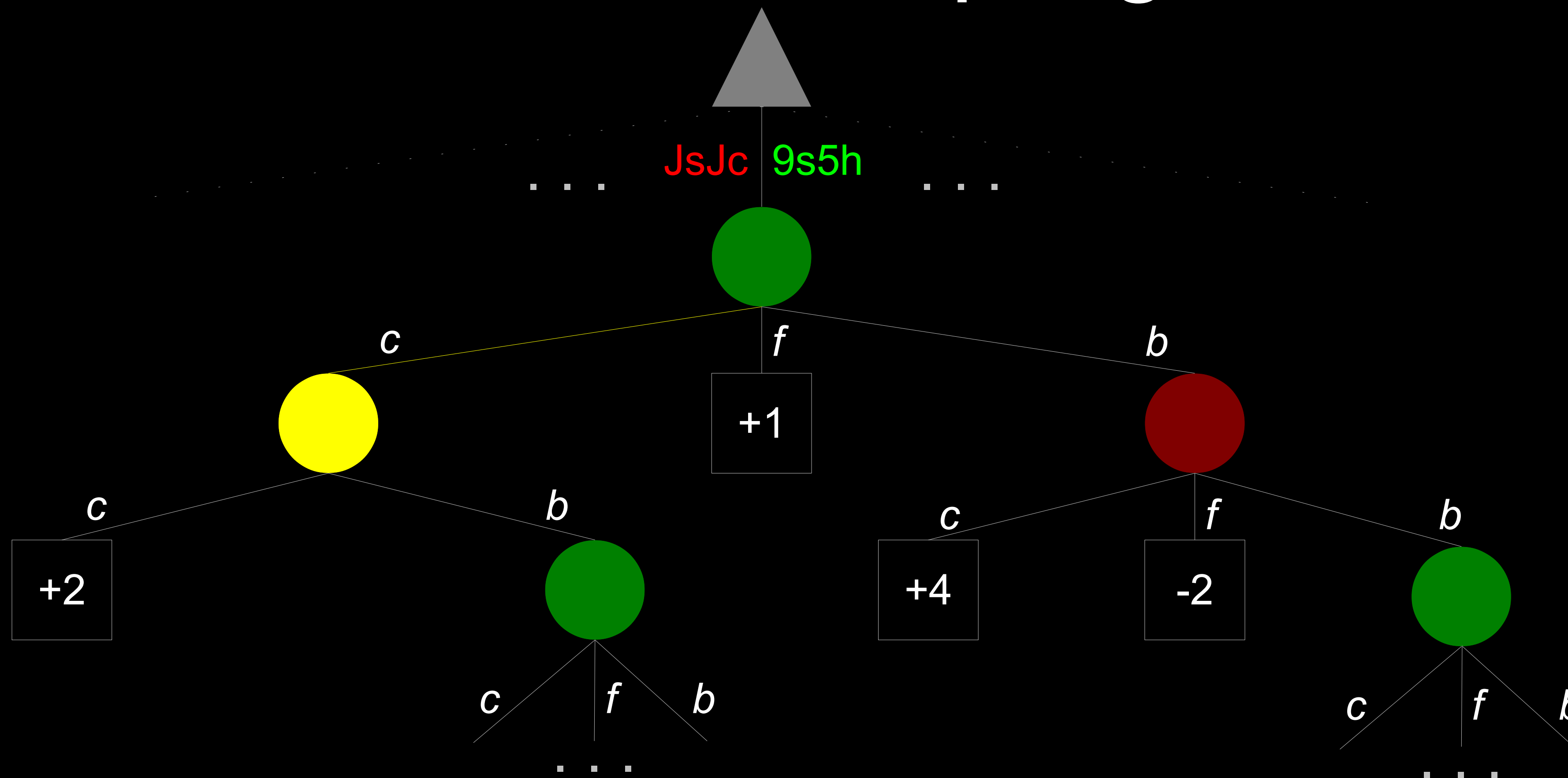
Chance Sampling



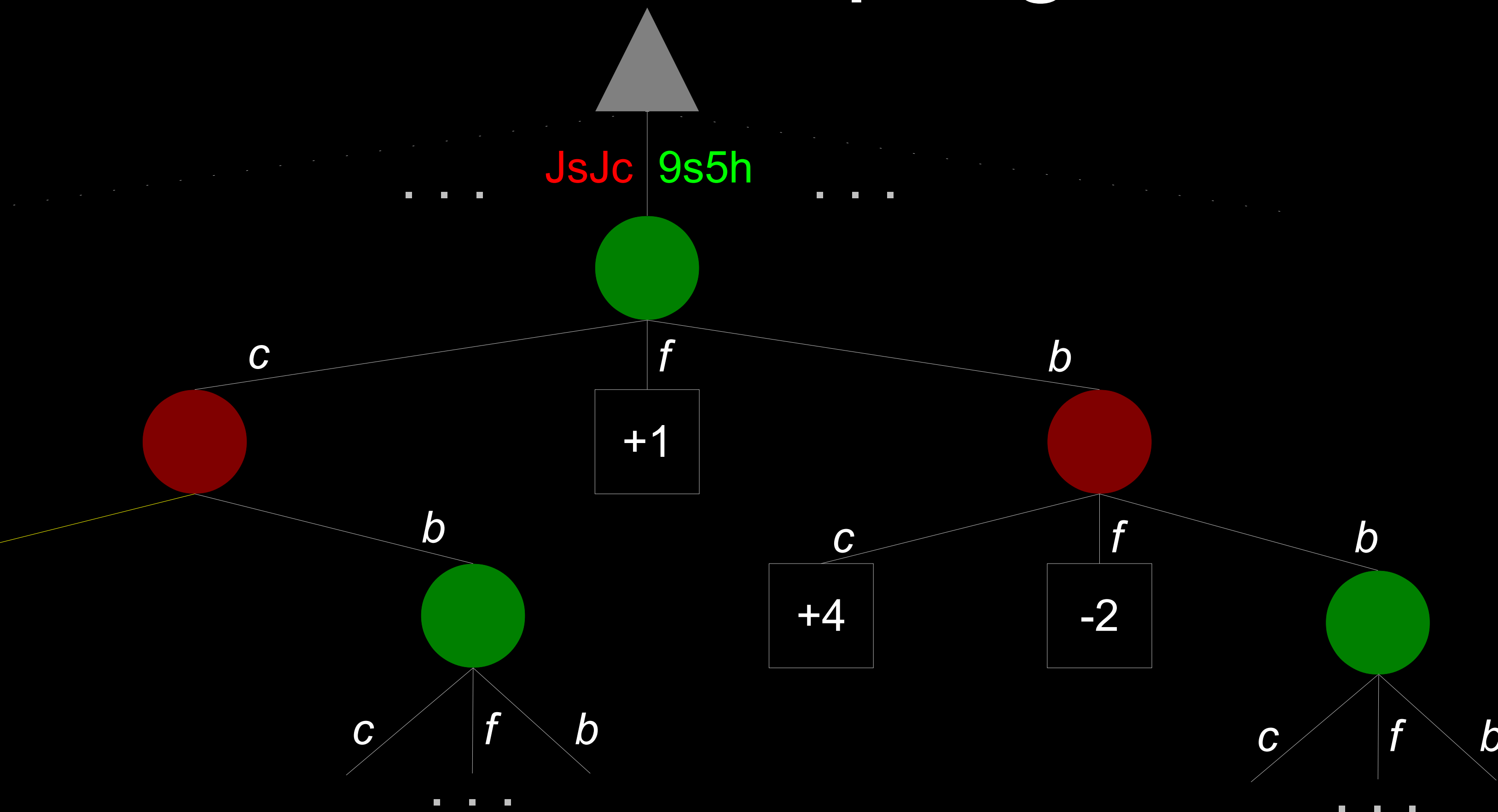
Chance Sampling



Chance Sampling



Chance Sampling

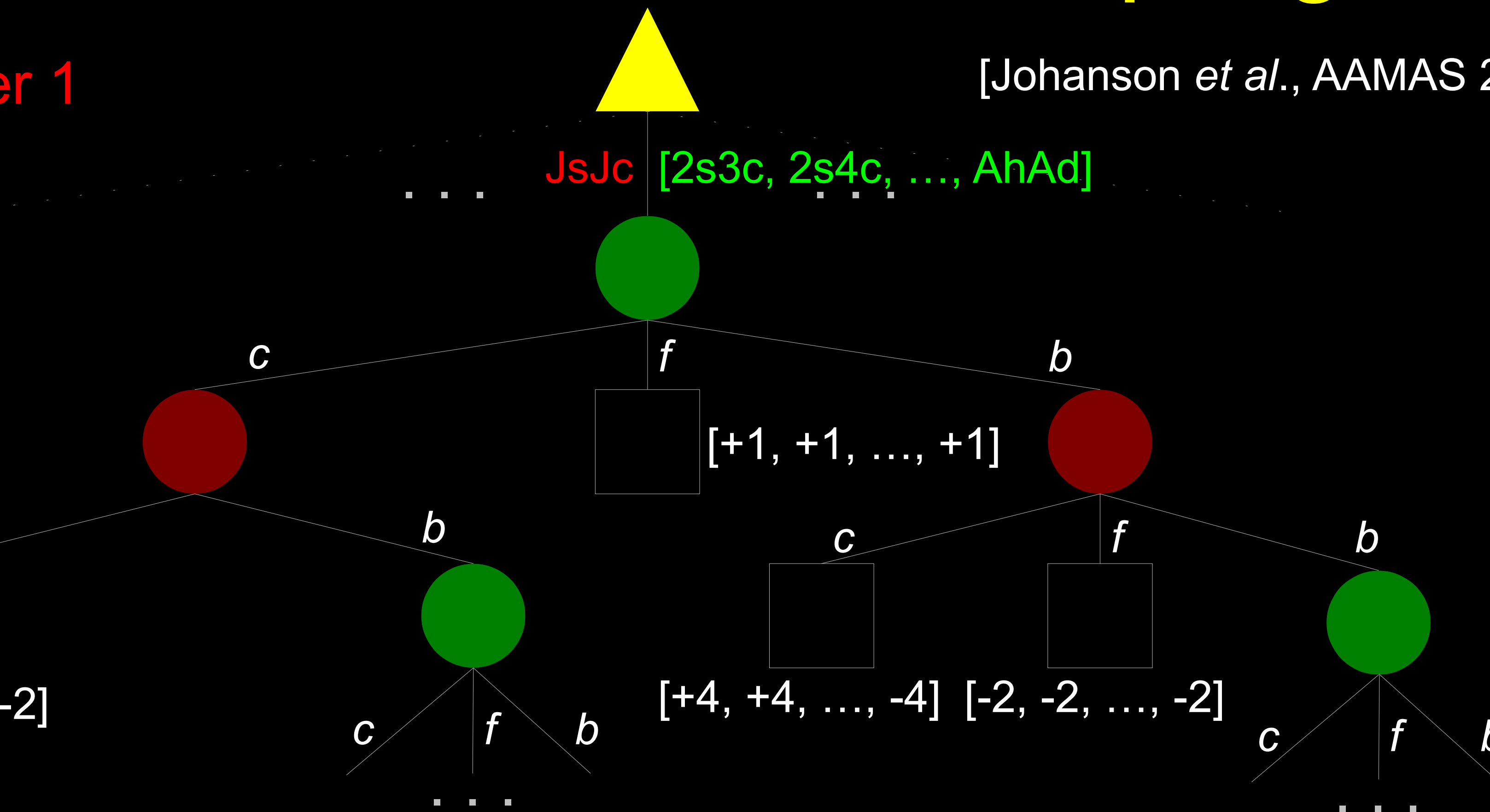


Utility(JsJc) = +2

Self-Public Chance Sampling

- Walk player 1

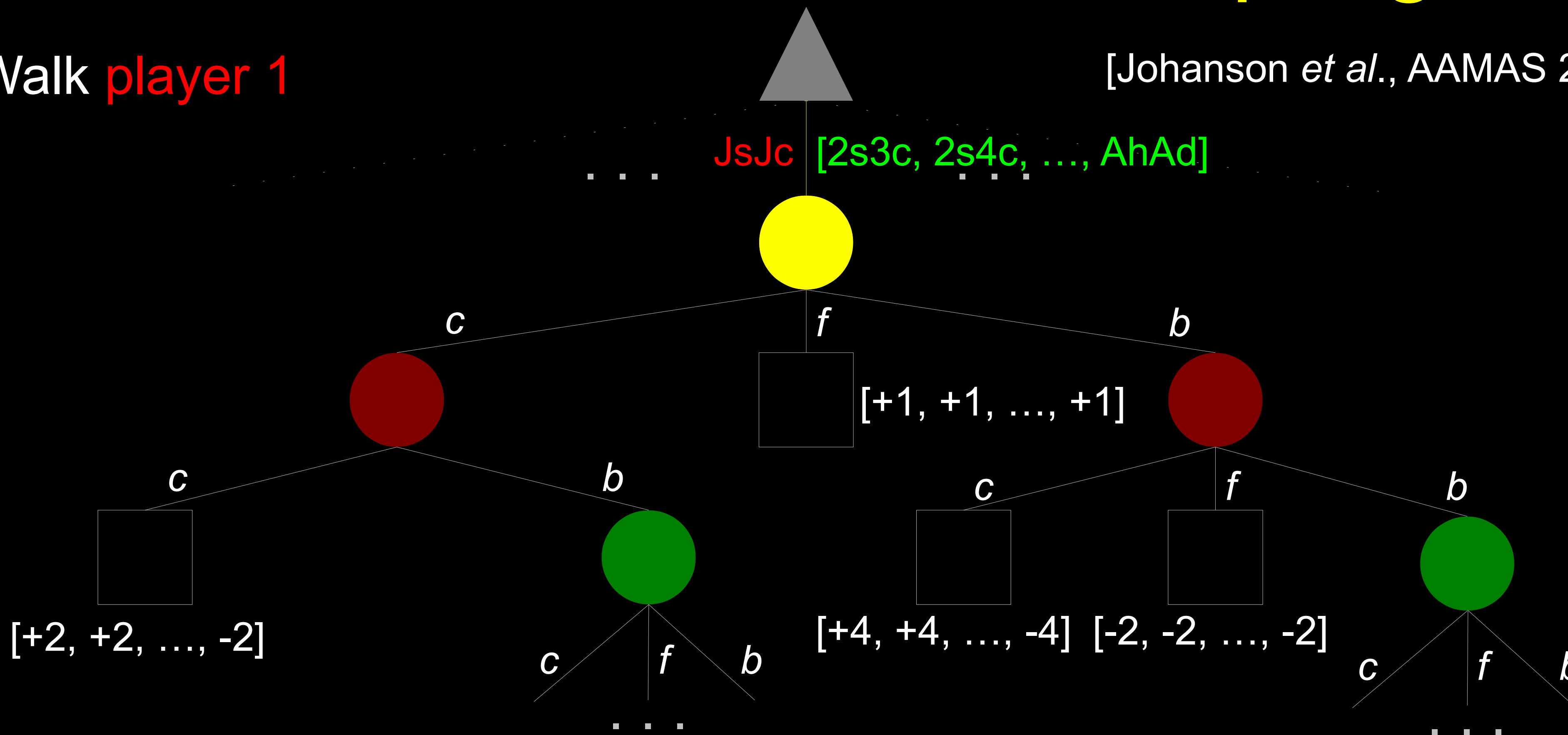
[Johanson *et al*., AAMAS 2012]



Self-Public Chance Sampling

- Walk player 1

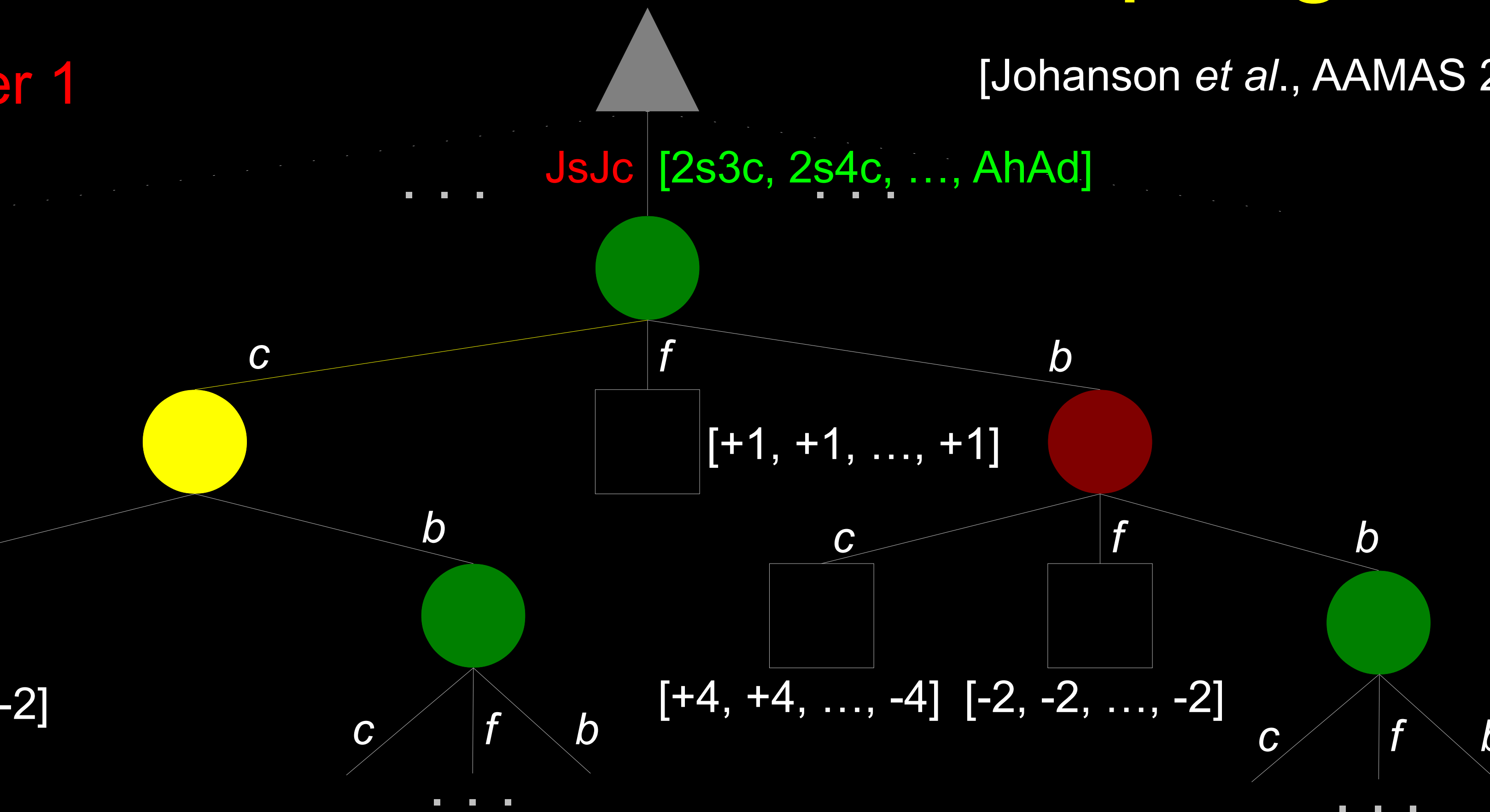
[Johanson *et al.*, AAMAS 2012]



Self-Public Chance Sampling

- Walk player 1

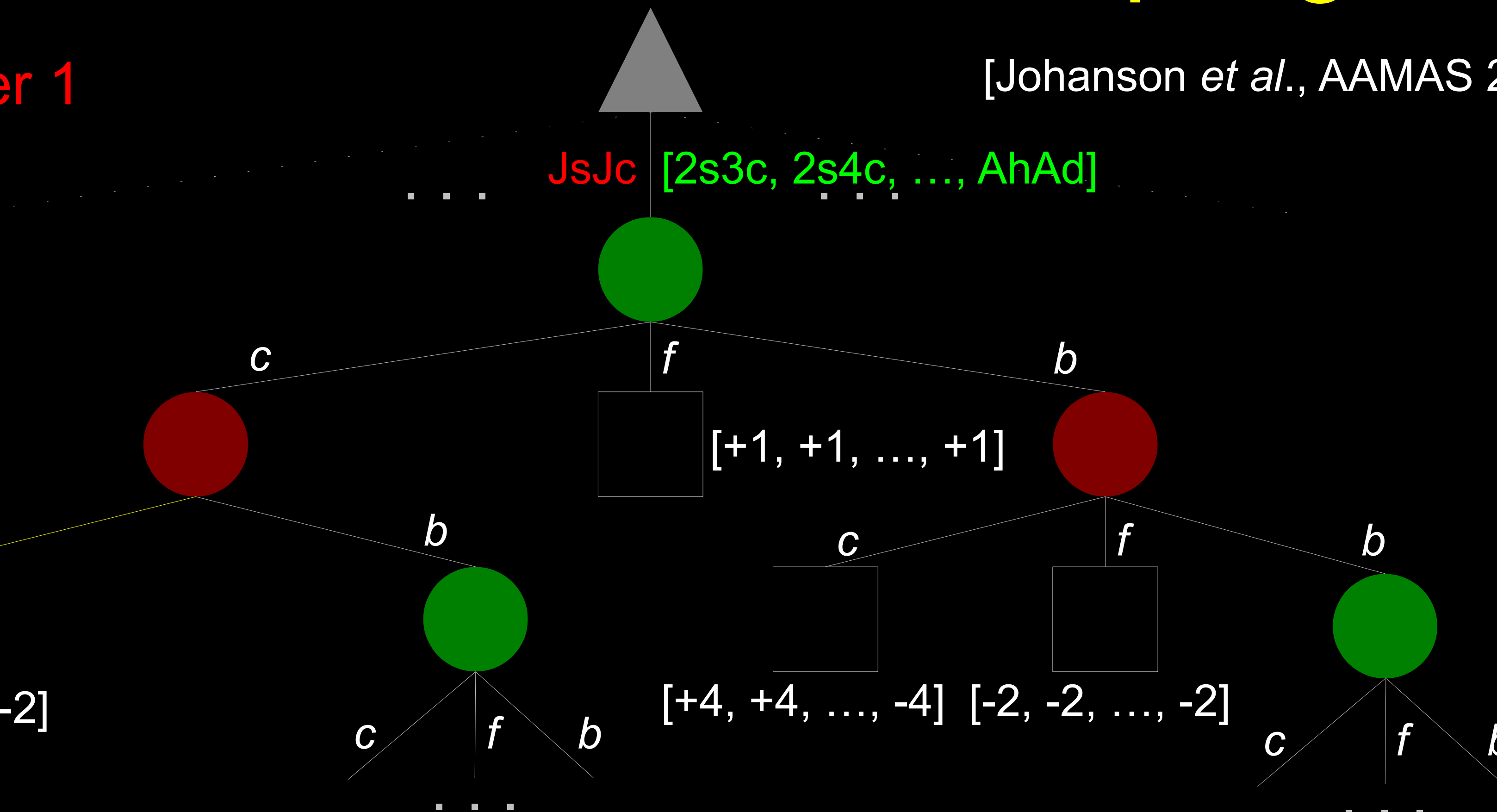
[Johanson *et al.*, AAMAS 2012]



Self-Public Chance Sampling

- Walk player 1

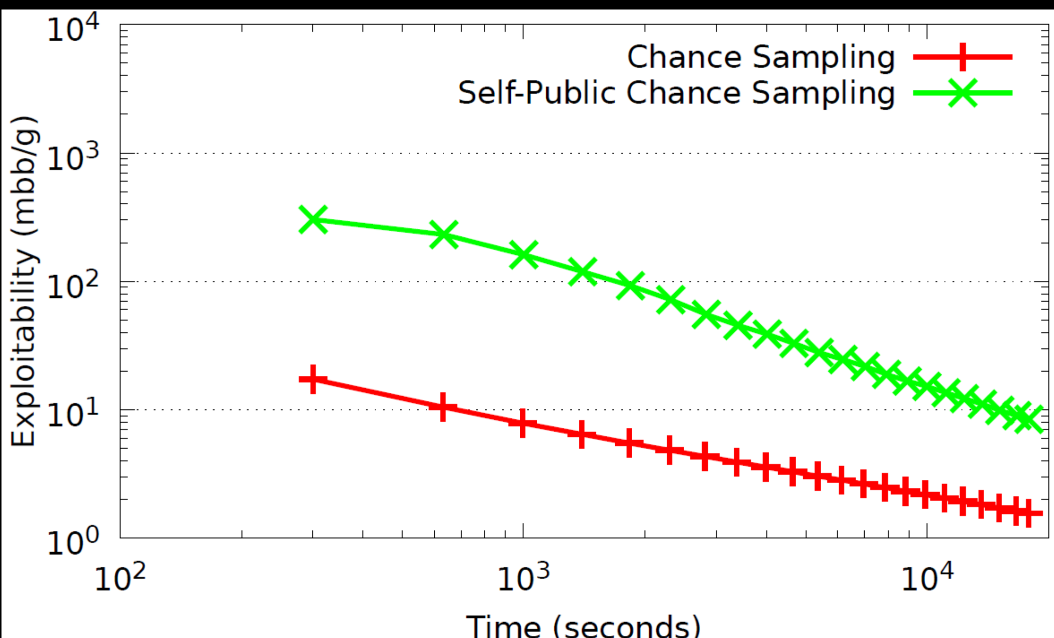
[Johanson *et al*., AAMAS 2012]



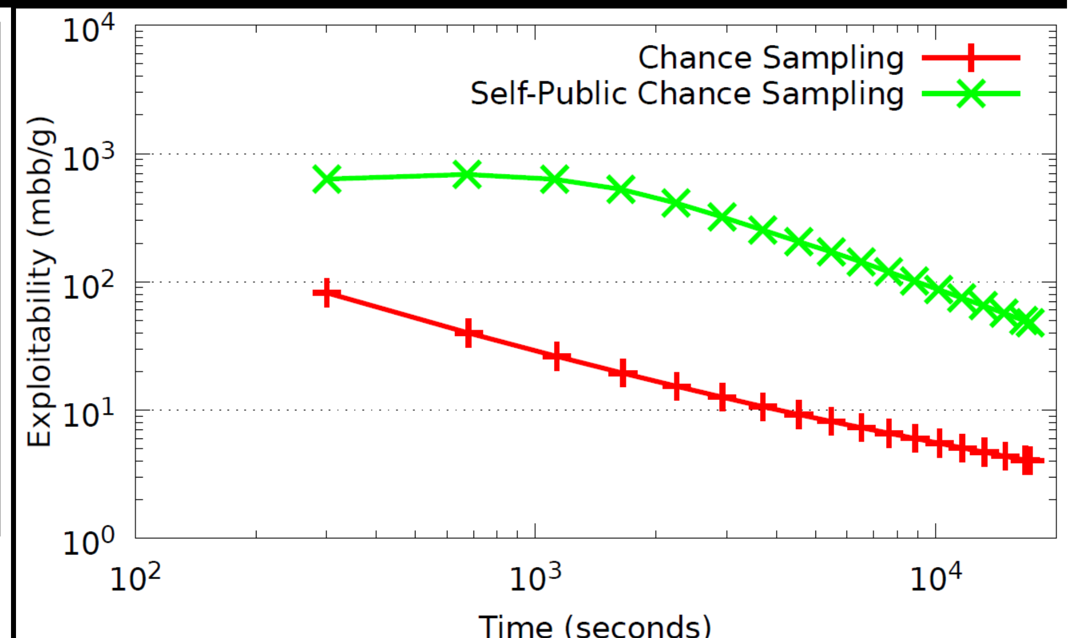
Utility(JsJc) = +2(probability opponent reaches with 2s3c)
+2(probability opponent reaches with 2s4c)
+ … -2(probability opponent reaches with AhAd)
= O(*n*) computation

Self-Public Chance Sampling Experiments

[2-1] Limit Hold'em



[2-4] Limit Hold'em



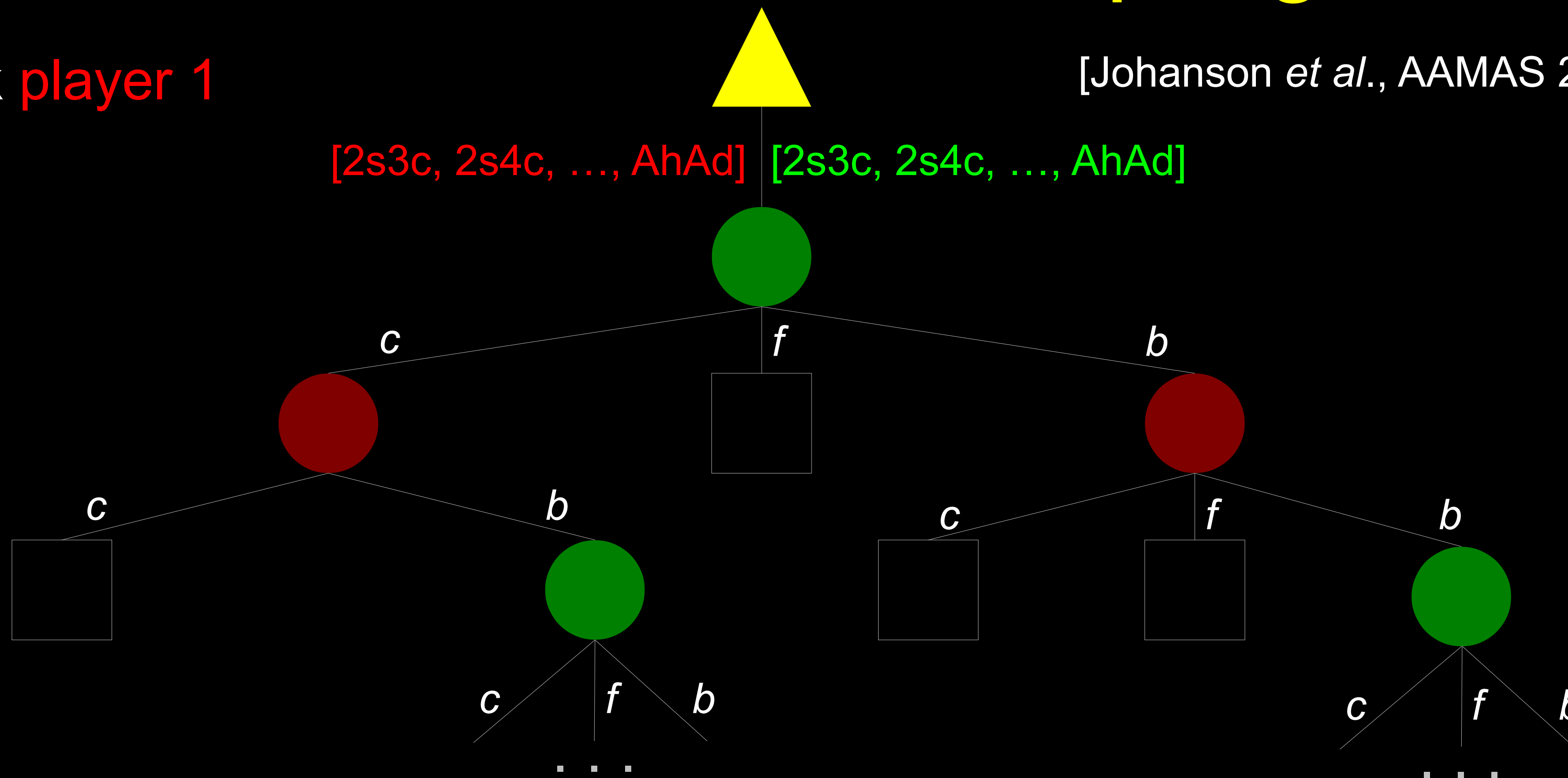
- Self-Public Chance Sampling requires fewer iterations than Chance Sampling (lower variance estimates).
- Iterations of Self-Public Chance Sampling are too expensive (O(*n*) operations at terminal nodes).

[Johanson *et al.*, AAMAS 2012]

Public Chance Sampling

- Walk player 1

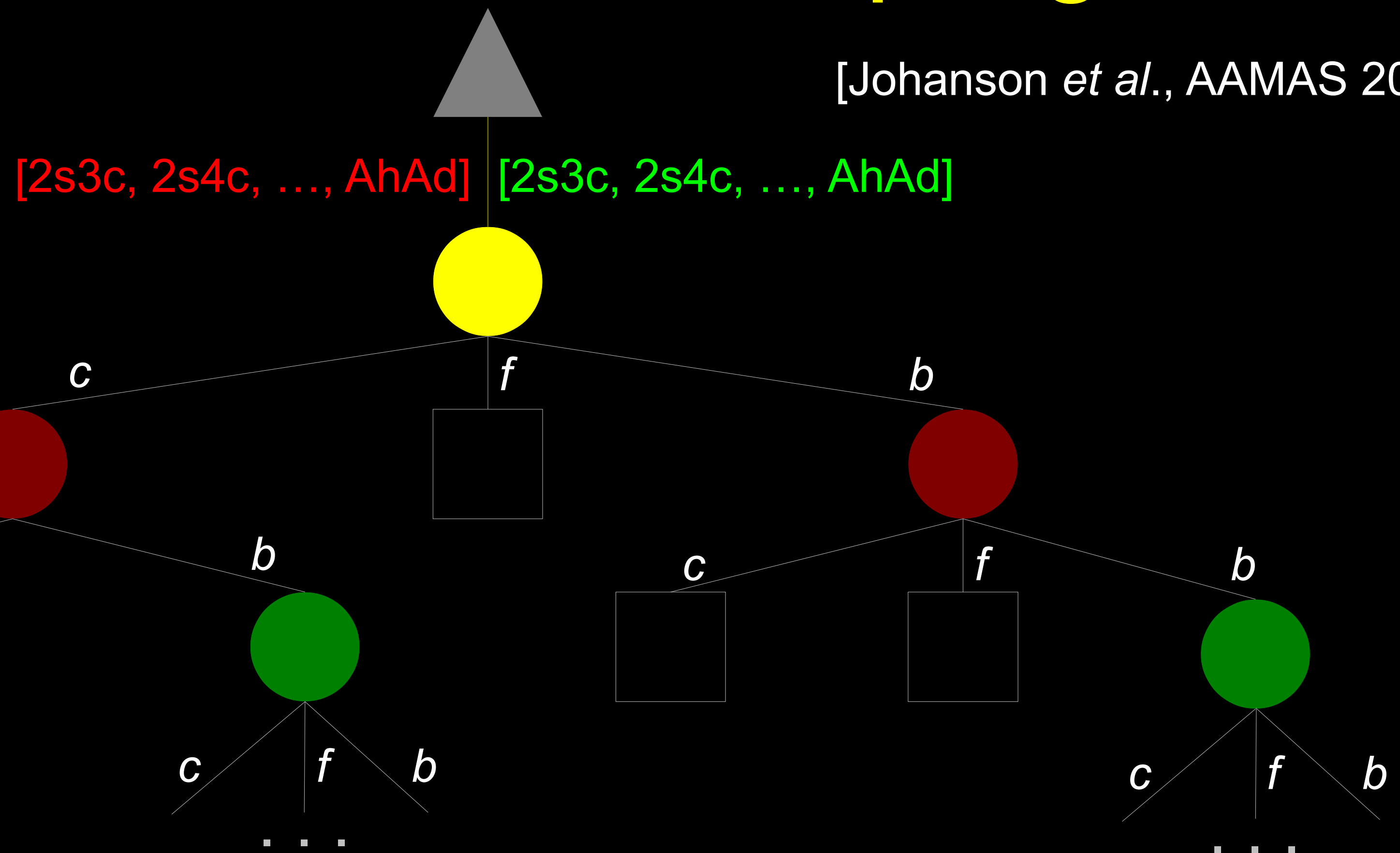
[Johanson *et al.*, AAMAS 2012]



Public Chance Sampling

- Walk player 1

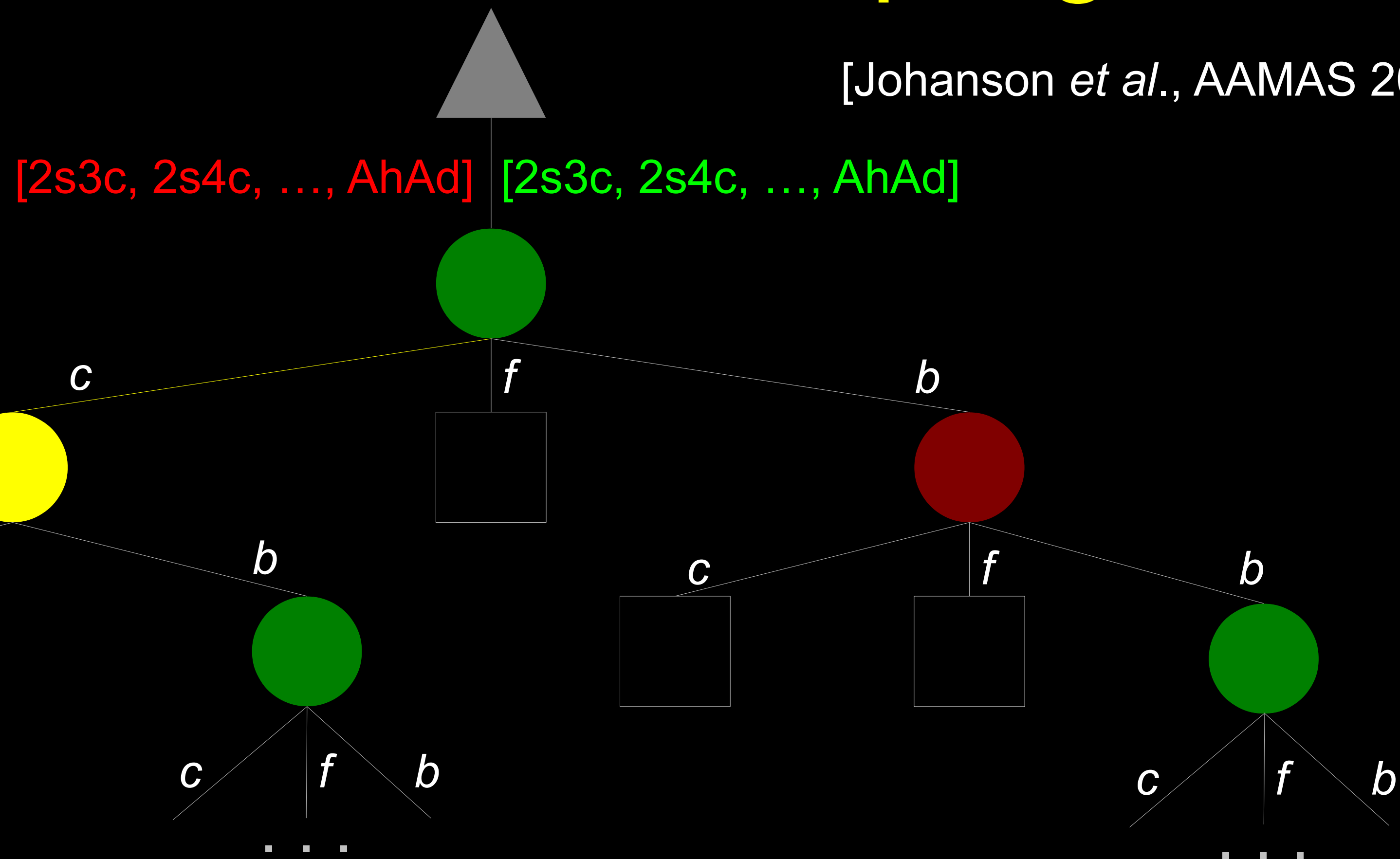
[Johanson *et al.*, AAMAS 2012]



Public Chance Sampling

- Walk player 1

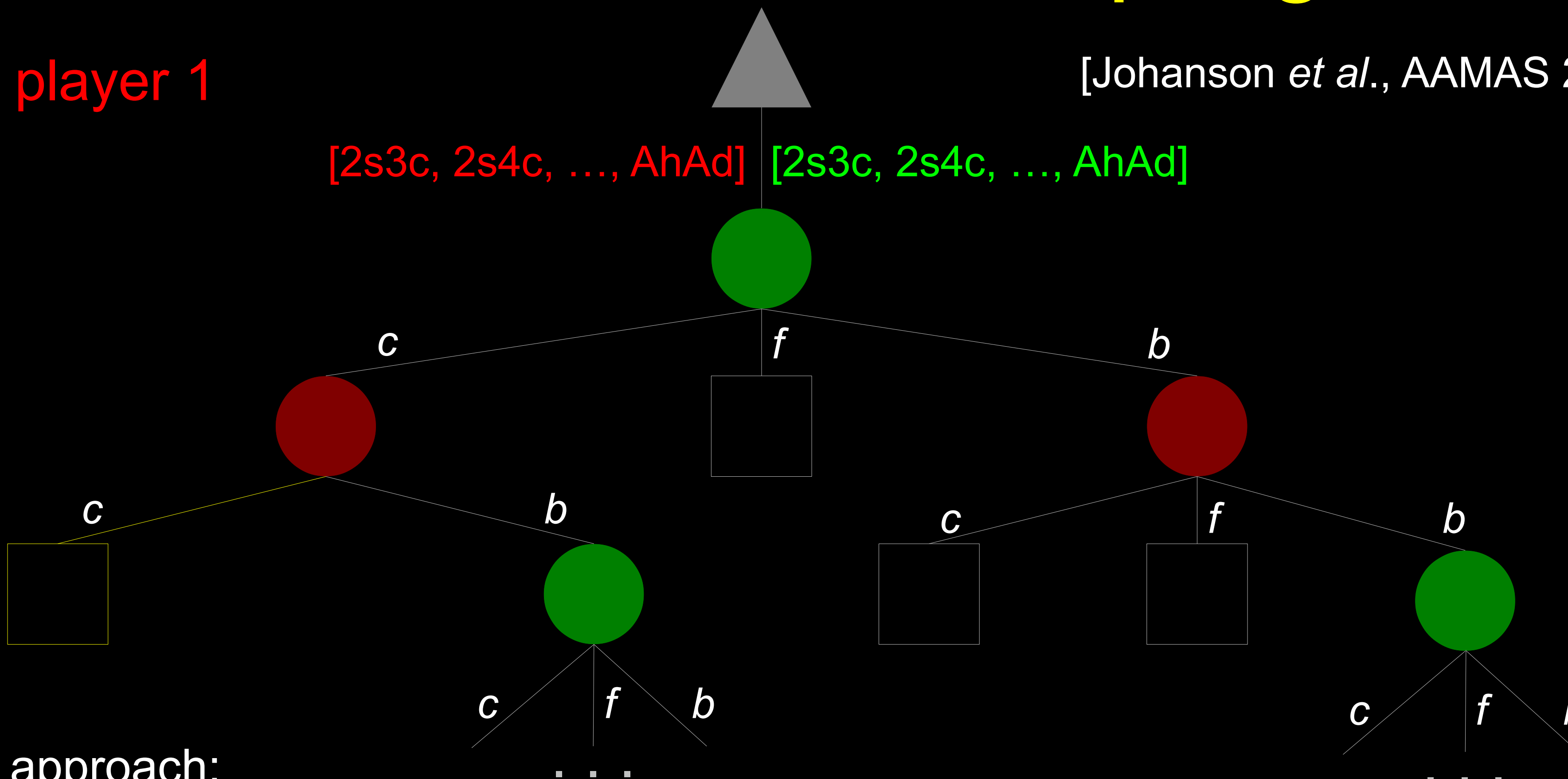
[Johanson *et al.*, AAMAS 2012]



Public Chance Sampling

- Walk player 1

[Johanson *et al.*, AAMAS 2012]



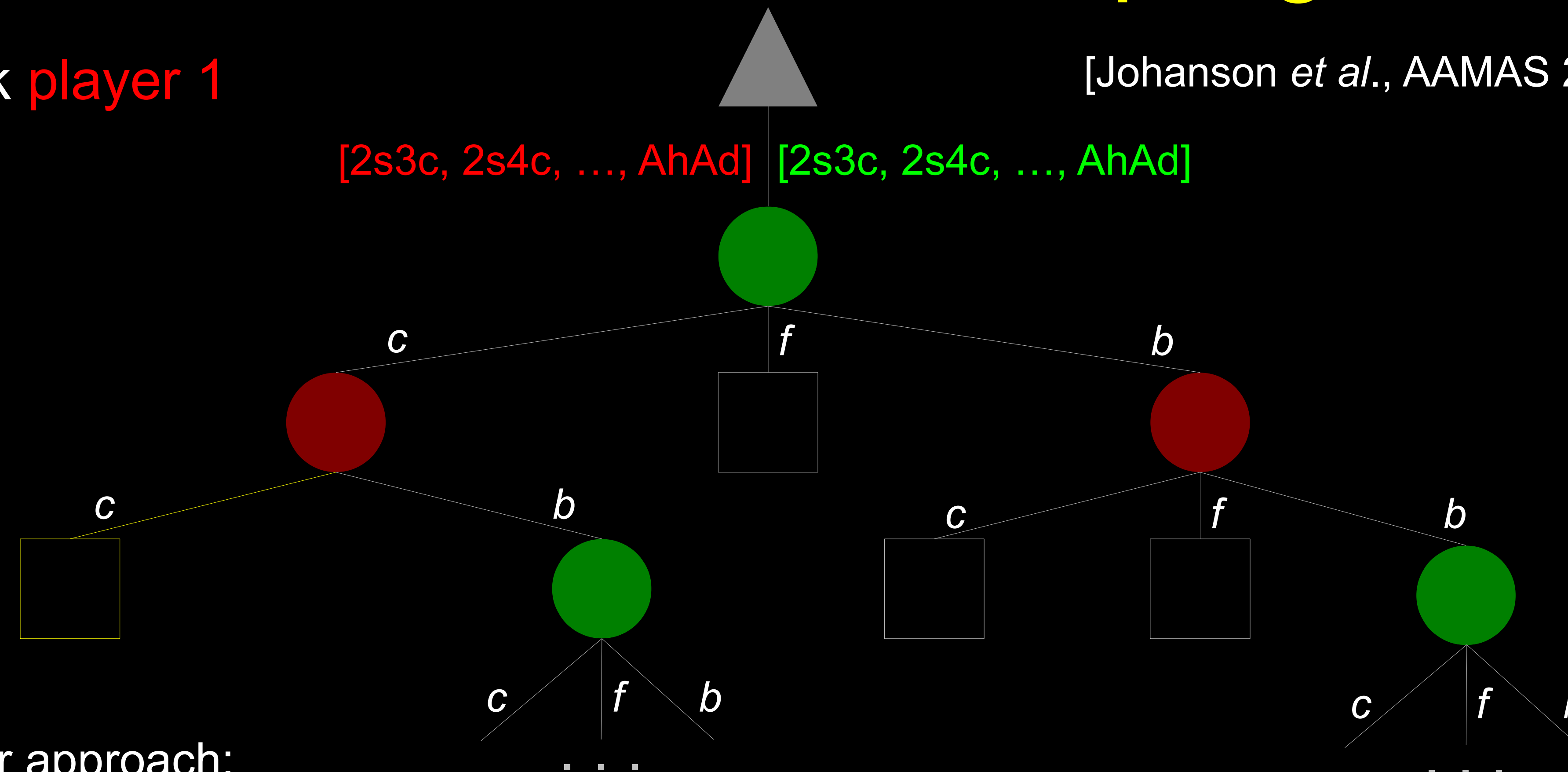
Naïve approach:
Utility(2s3c) = O(n) computation
Utility(2s4c) = O(n) computation
...
Utility(AhAd) = O(n) computation

O(n^2) computation

Public Chance Sampling

- Walk player 1

[Johanson *et al.*, AAMAS 2012]



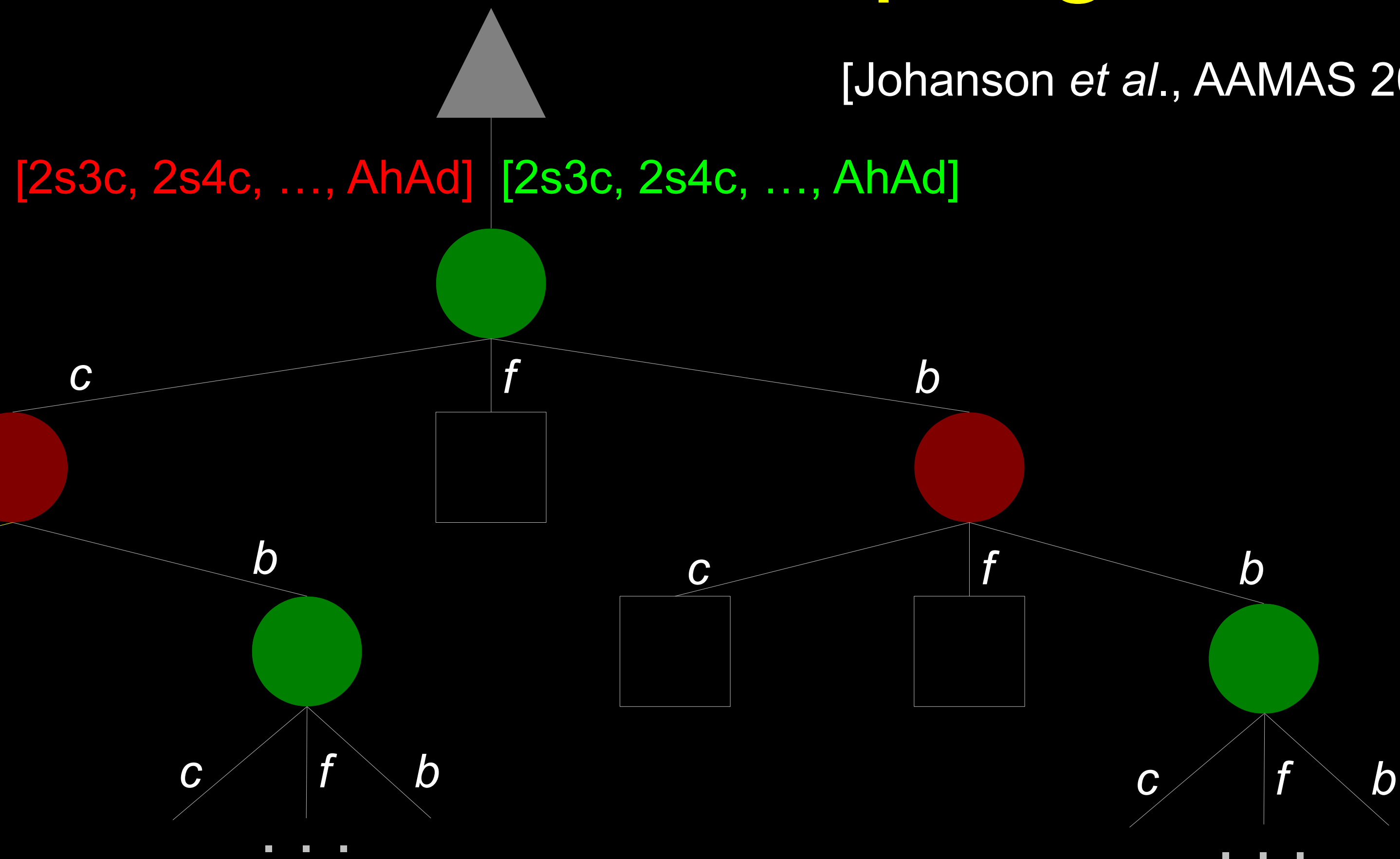
Better approach:

1. p = probability opponent reaches with any hand = O(n) computation

Public Chance Sampling

- Walk player 1

[Johanson *et al.*, AAMAS 2012]



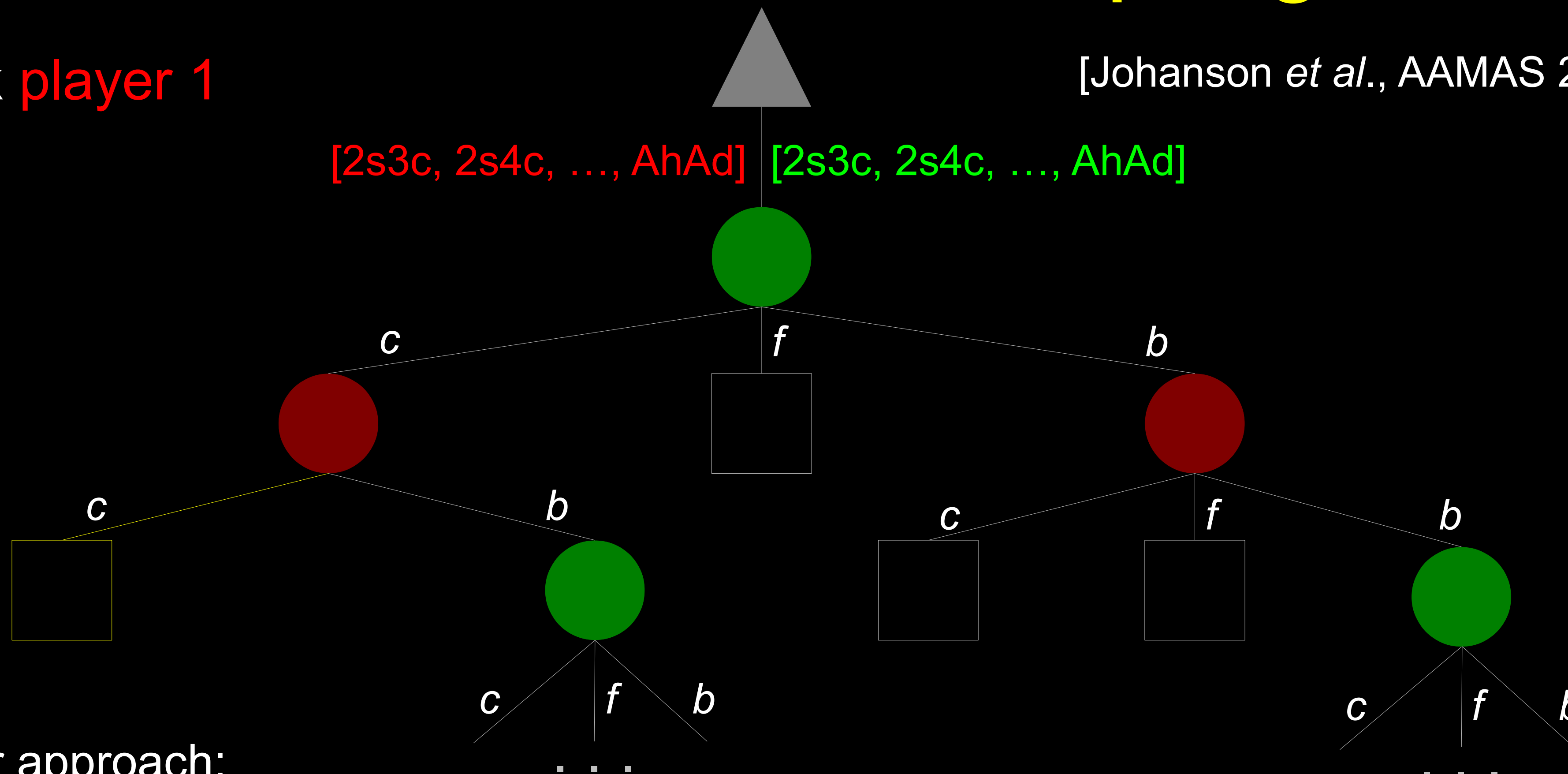
Better approach:

1. p = probability opponent reaches with any hand = $O(n)$ computation
2. w = 0 (probability opponent reaches and player 1 has better hand)

Public Chance Sampling

- Walk player 1

[Johanson *et al.*, AAMAS 2012]



Better approach:

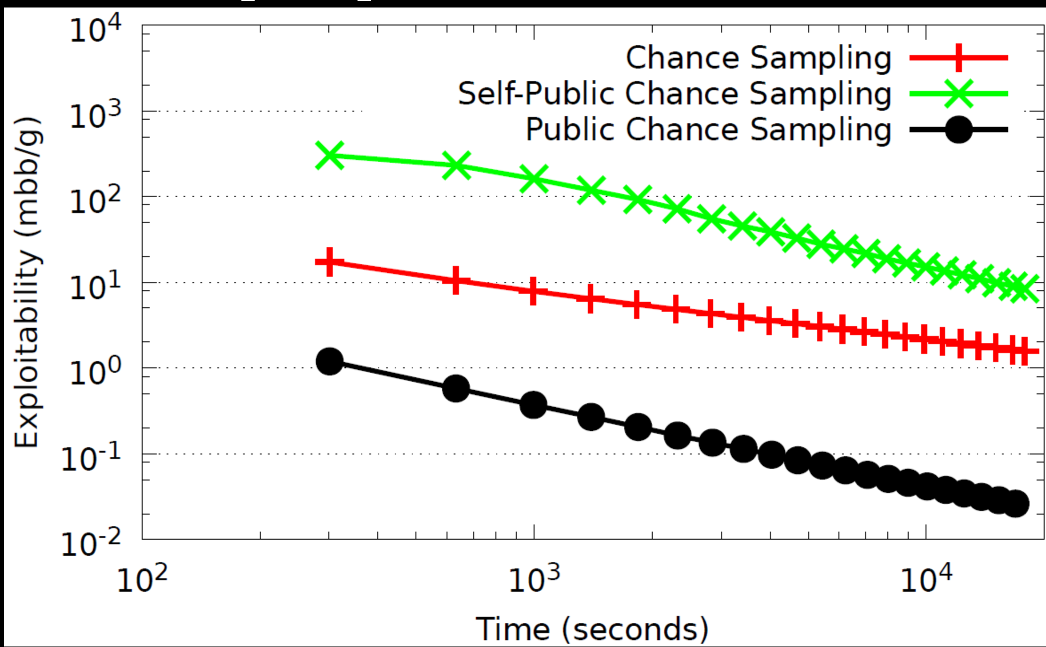
1. p = probability opponent reaches with any hand = $O(n)$ computation
2. w = 0 (probability opponent reaches and player 1 has better hand)
3. for *hand* in [2s3c, 2s3h, …, AhAd] (ordered worst to best) do
   Utility(*hand*) = $+2w - 2(p - w)$
   w += probability opponent reaches with *hand*
   end for

$O(n)$ computation

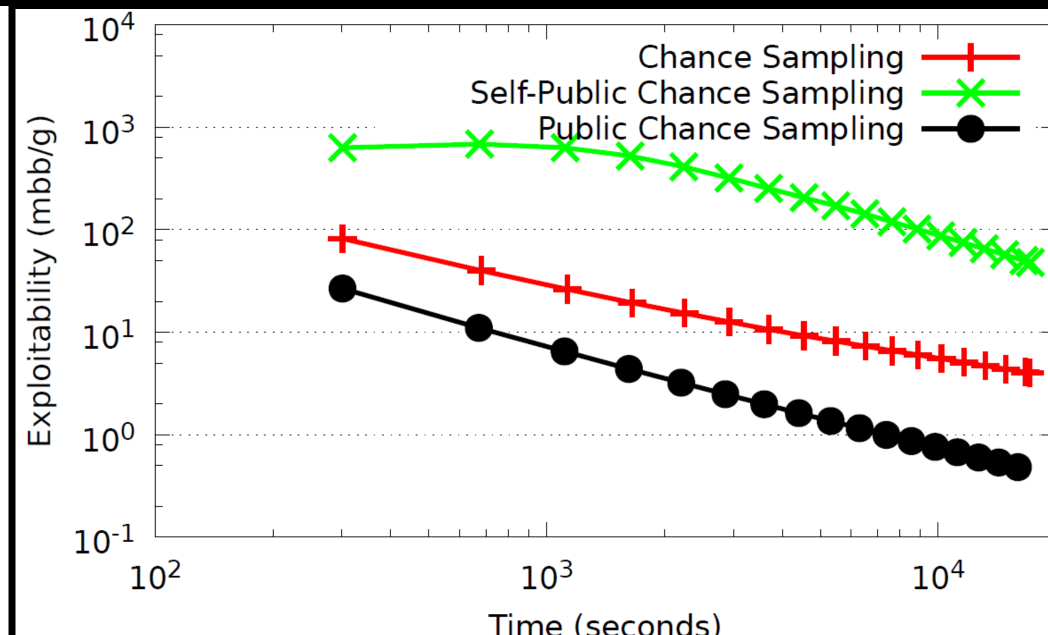
[Johanson *et al.*, IJCAI 2011]

Public Chance Sampling Experiments

[2-1] Limit Hold'em



[2-4] Limit Hold'em

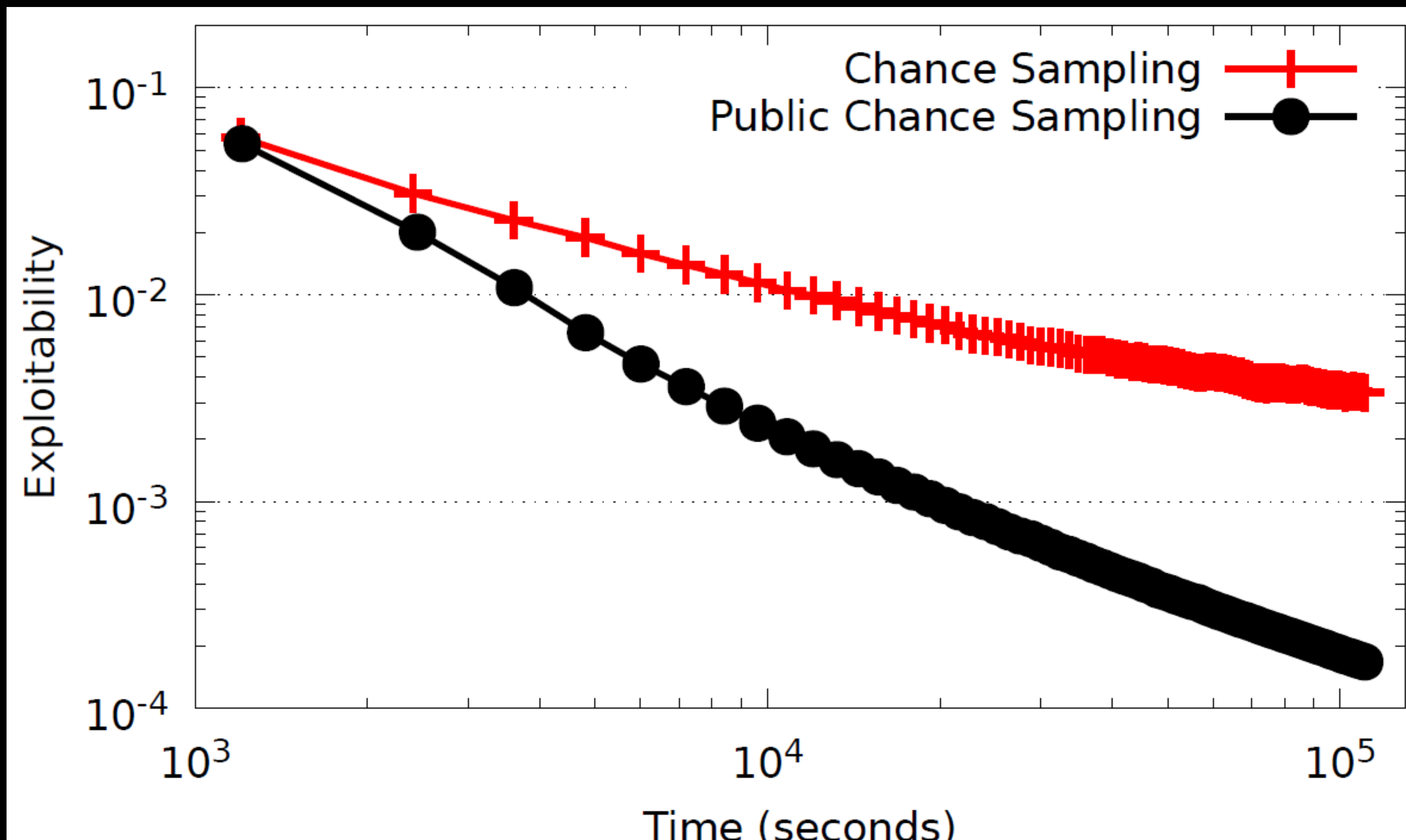


- Public Chance Sampling iterations are equally as expensive as Self-Public Chance Sampling iterations.
- At terminal nodes, Public Chance Sampling does $O(n^2)$ work in $O(n)$ time, and thus requires fewer iterations than Self-Public Chance Sampling.

[Johanson *et al.*, AAMAS 2012]

Public Chance Sampling Experiments

Bluff with 2 dice each



- $O(n^2)$ to $O(n)$ trick can also be applied to Bluff.

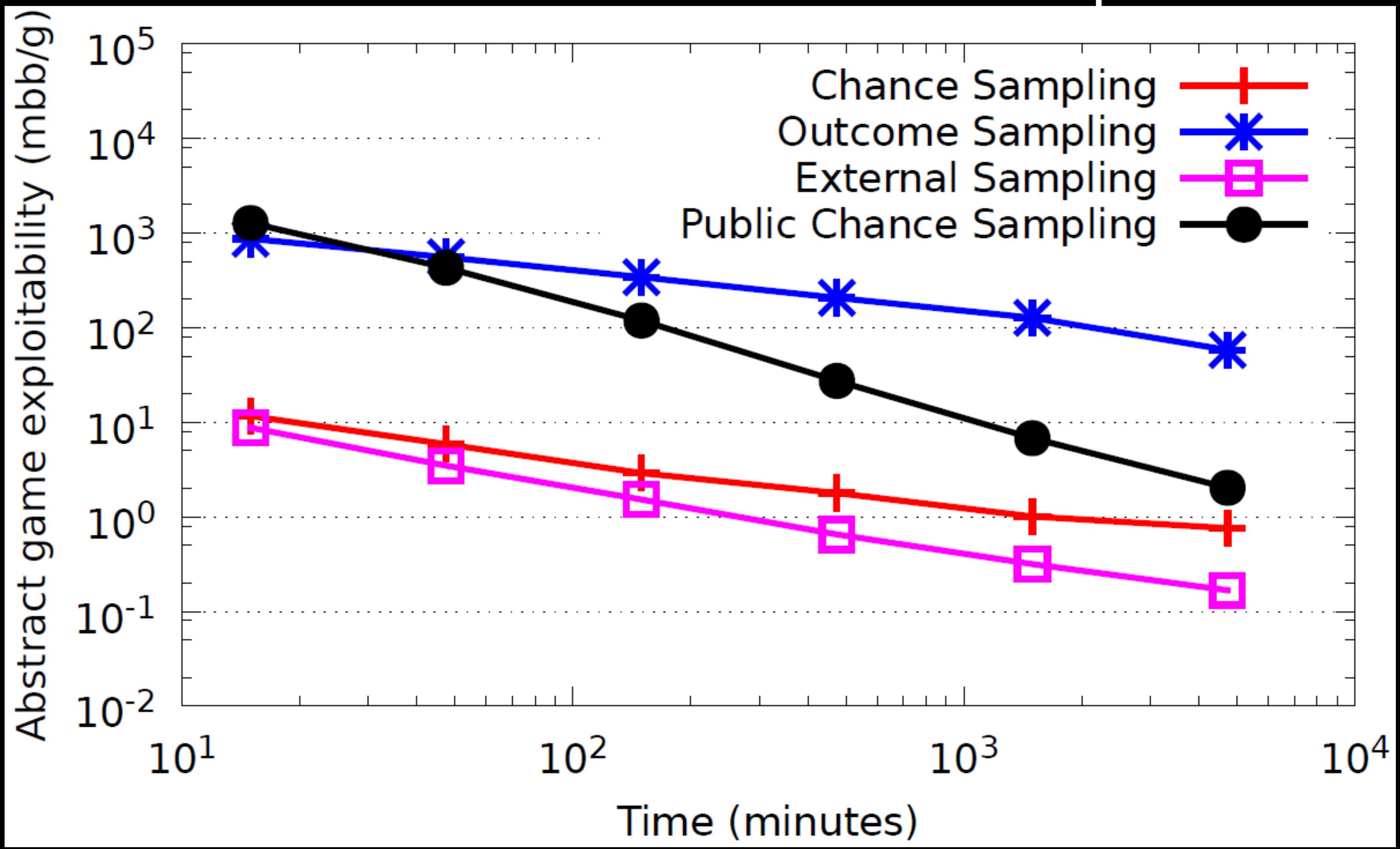
[Johanson *et al.*, AAMAS 2012]

Public Chance Sampling

- Works well in games where players have many possible private states (large *n*).
- Limitation: Does not combine well with actions sampling.
  - Action sampling is very useful in no-limit poker games...

Public Chance Sampling

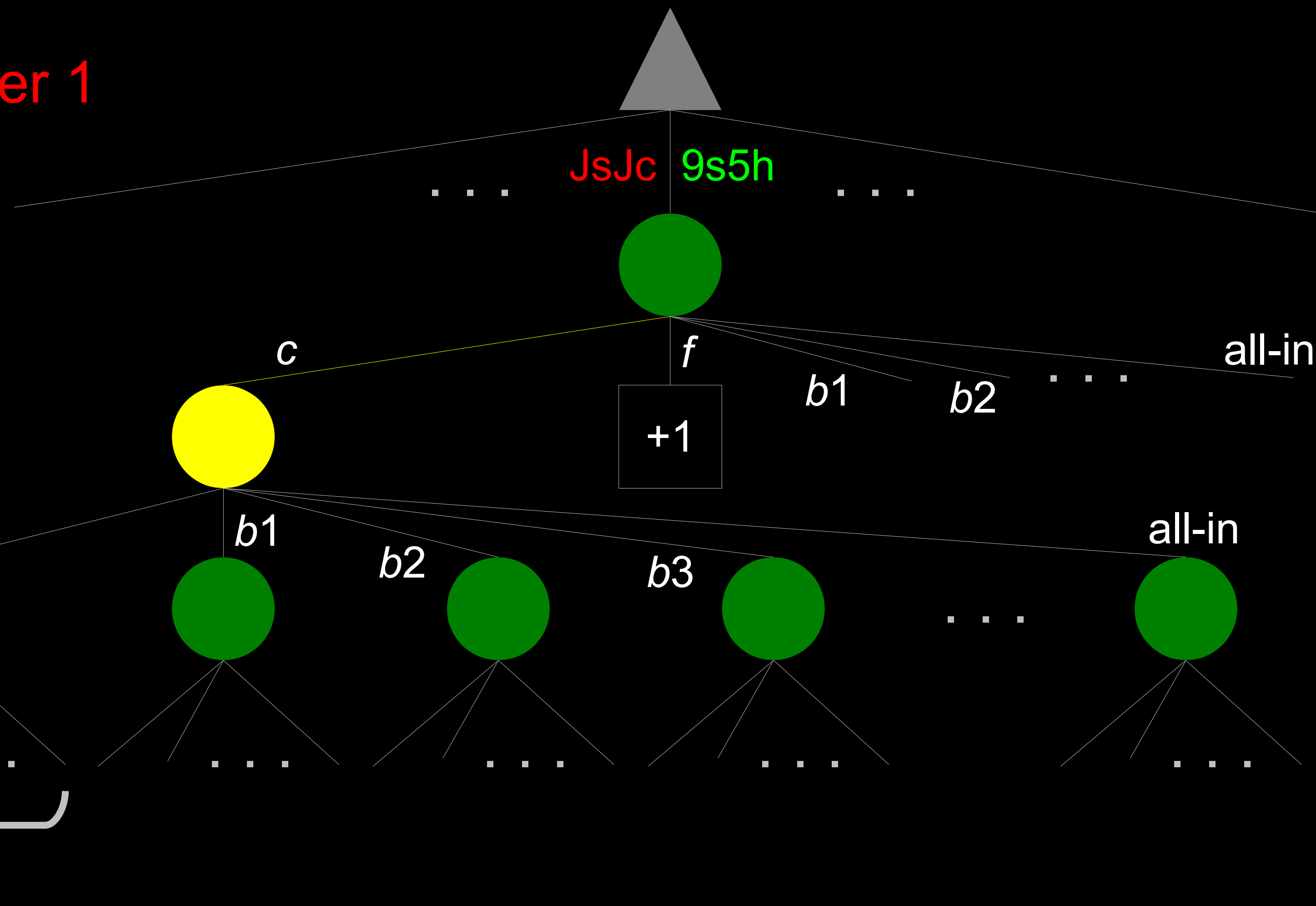
2 round No-limit Hold'em with 30 chip stacks



- Card abstraction applied to reduce chance branching factor to 5
- **Question**: Can we beat External Sampling?

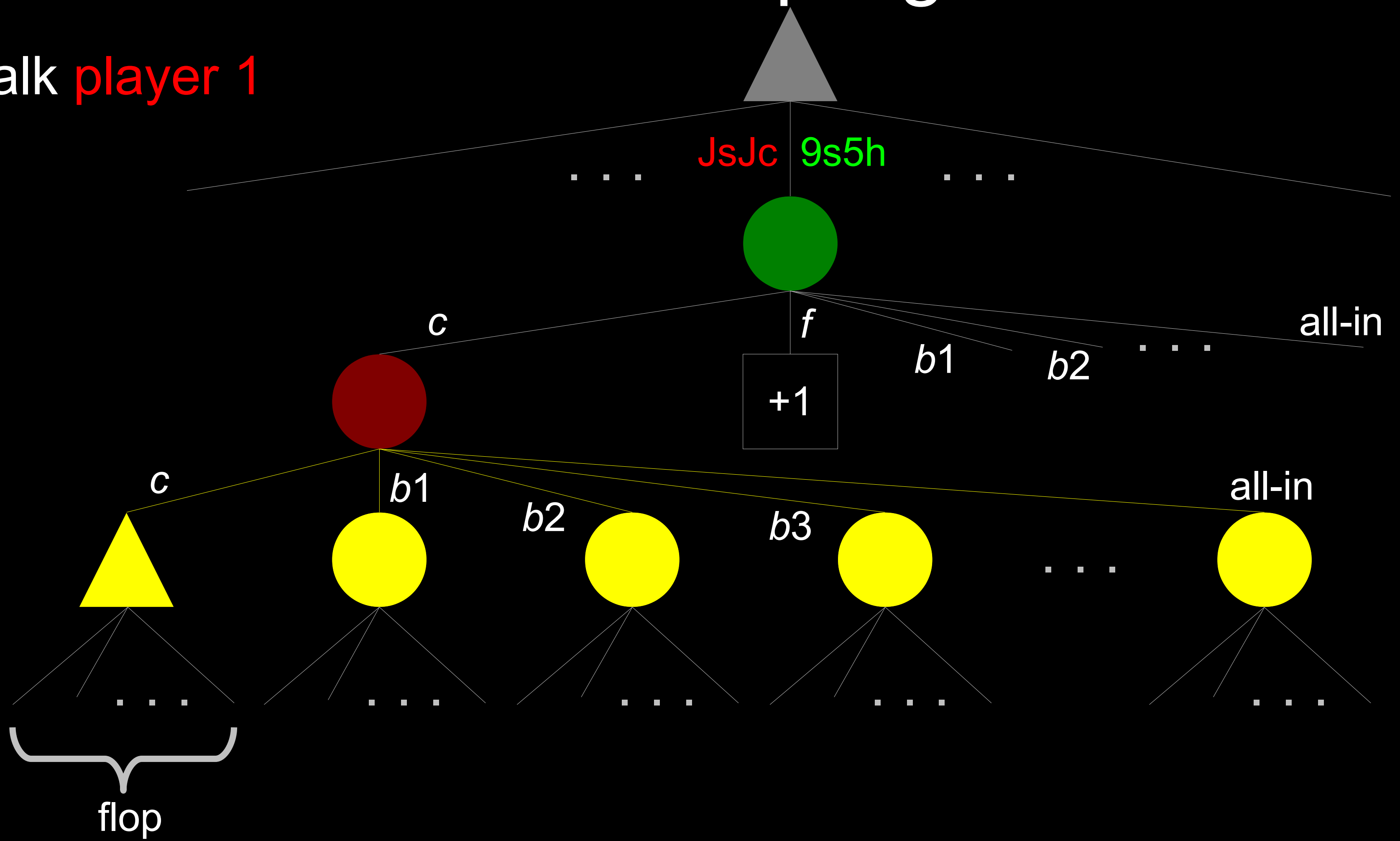
No-limit Poker

- Walk player 1



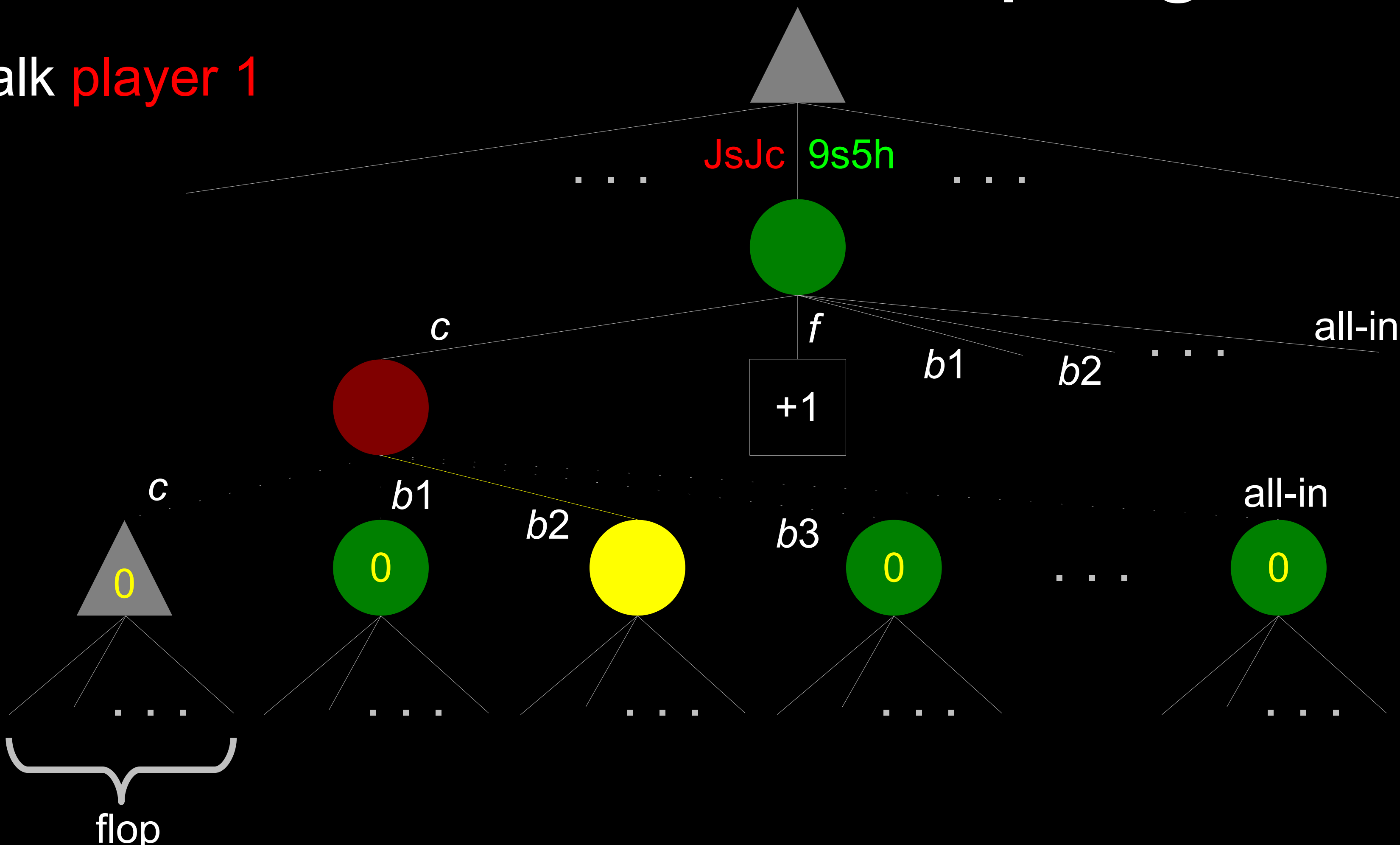
Vanilla / Public Chance / Chance / External Sampling

- Walk player 1



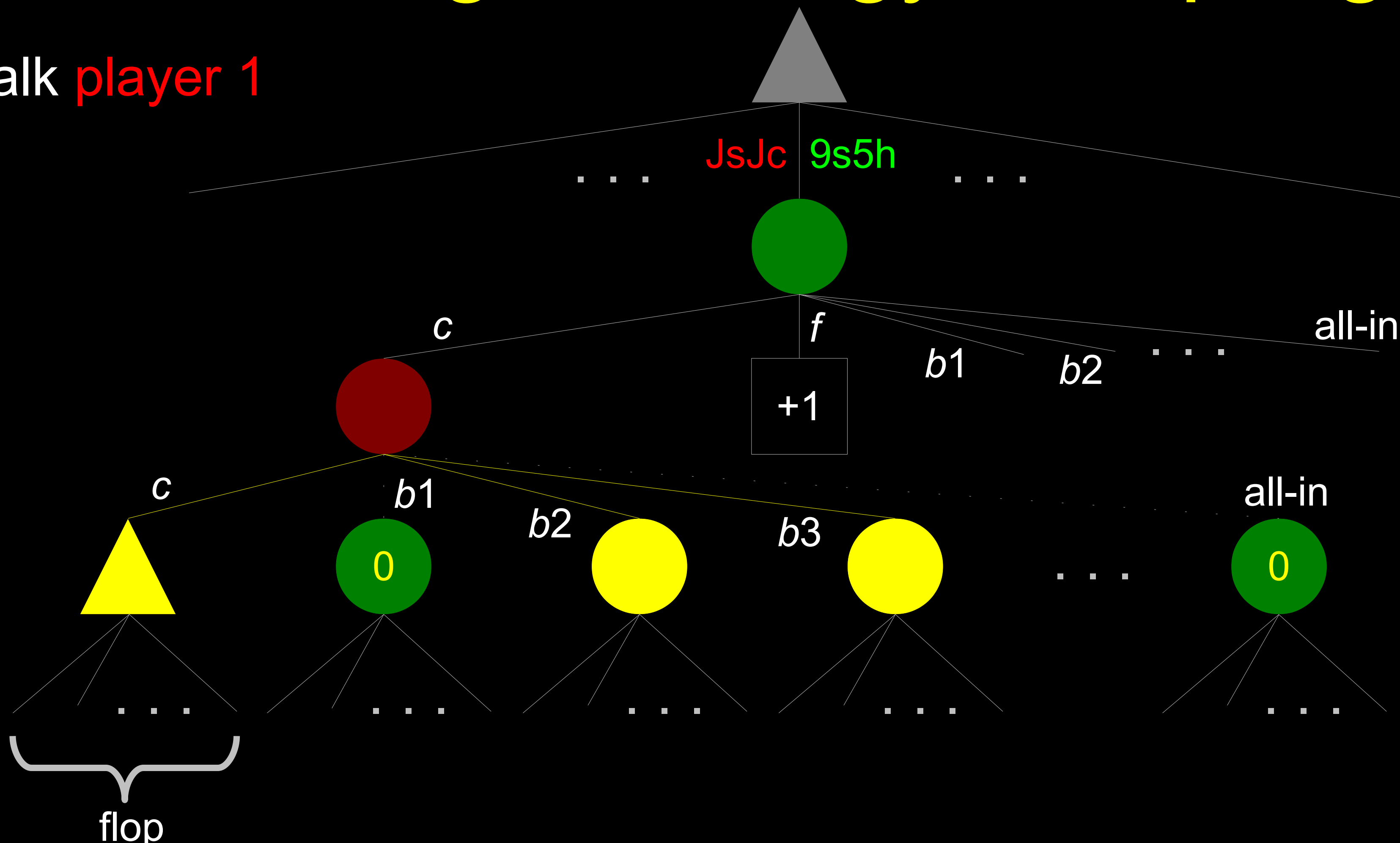
Outcome Sampling

- Walk player 1



Average Strategy Sampling

- Walk player 1



P[sample action *a*] ≈ **P**[average strategy plays *a*]

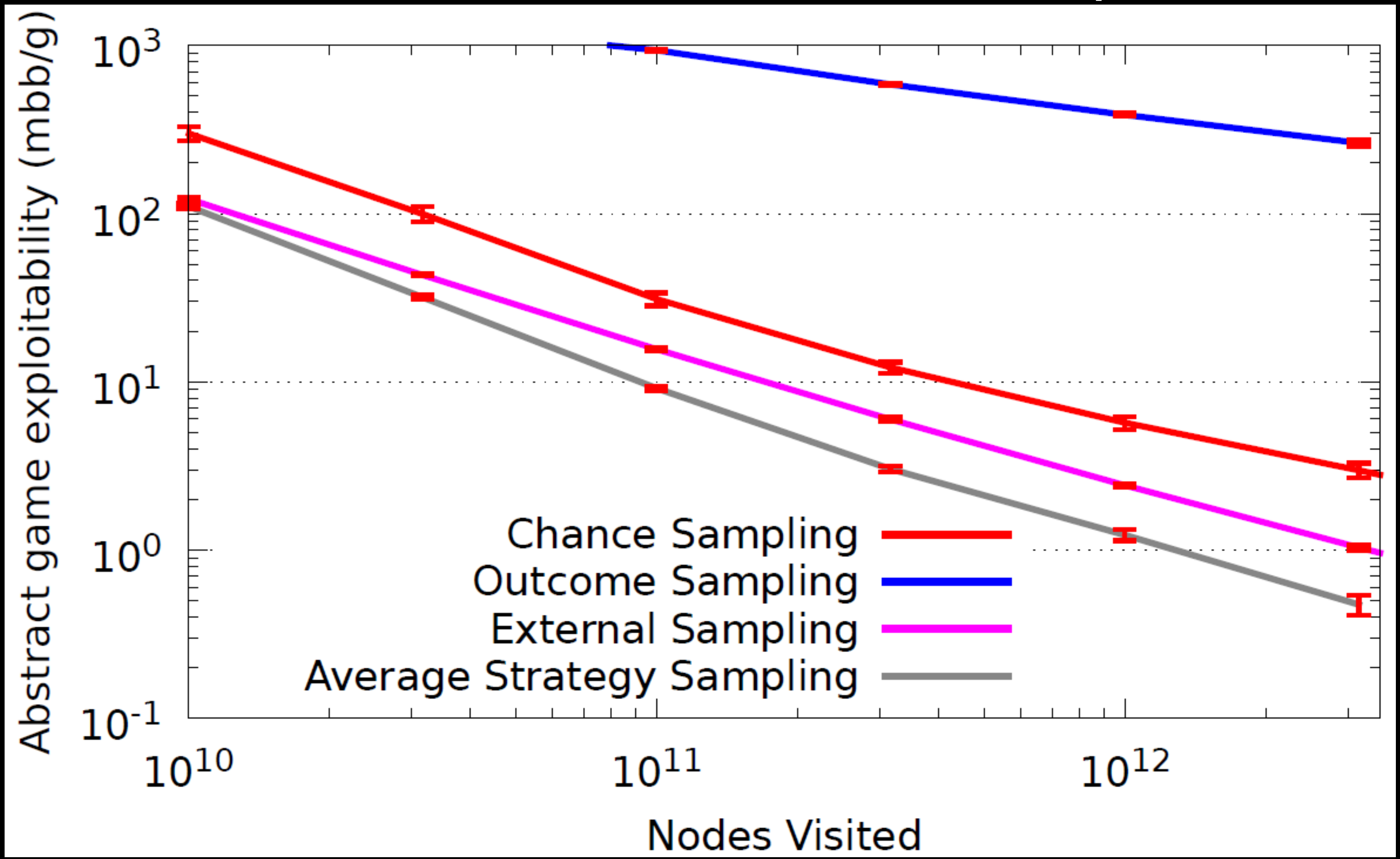
Average Strategy Sampling

- At the end of the day, the strategy that we will actually use is the average strategy.
- Average Strategy Sampling samples more often towards those information sets that our average strategy reaches, and so more often updates action probabilities at the information sets we reach in practice.
- Theoretical arguments suggest that this is a good thing to do.
- For chance and opponent nodes, we follow external sampling rules.

- Iterations required: Chance ≤ External < Average Strategy < Outcome
- Time per iteration: Chance > External > Average Strategy > Outcome

Average Strategy Sampling Experiments

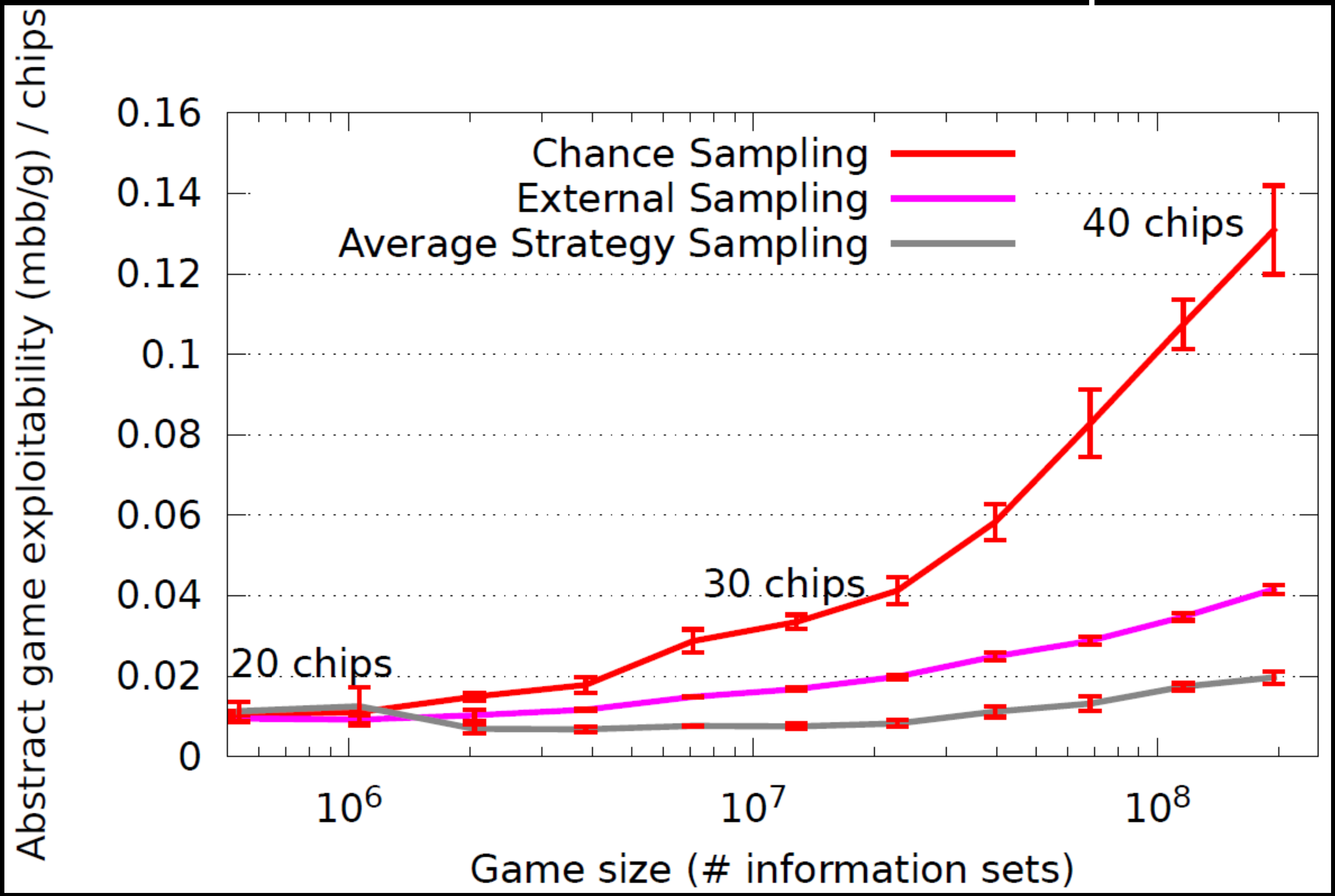
2 round No-limit Hold'em with 36 chip stacks



- Card abstraction applied to reduce chance branching factor to 5

Average Strategy Sampling Experiments

2 round No-limit Hold'em with 20-40 chip stacks



- Card abstraction applied to reduce chance branching factor to 5
- Each algorithm run for approximately 3.16 trillion nodes visited

Conclusions

- We have developed new, fast sampling variants of Counterfactual Regret Minimization for solving large, 2-player zero-sum extensive-form games.
  - Probing: Reduce variance of estimates by sampling a terminal node for non-sampled actions.
  - Public Chance Sampling: Reduce variance of estimates and achieve $O(n^2)$ work in $O(n)$ time by consider all possible private states for both players.
  - Average Strategy Sampling: Produce fast iterations by sampling a subset of actions of the actions for the current player according to the player's average strategy.
- Future work: Are these sampling variants fast enough so that we can apply a modification of CFR on-line to exploit the opponent's tendencies?