



Combining Deep Reinforcement Learning and Search for Imperfect-Information Games

Noam Brown*, Anton Bakhtin, Adam Lerer, Qucheng Gong

NOAM BROWN?

현재 **FAIR (Facebook AI Research)** Research Scientist로 Multi-agent AI를 연구 중
그 중에서도 머신러닝과 게임이론을 연결하는 연구를 주로 진행하였음
No limit Poker (**Imperfect Information game**) 문제를 주로 다룸.

2017 nips에서 Safe and Nested Subgame Solving for Imperfect Information game으로 best paper award
Libratus로 Outstanding Achievement in AI에서 Marvin Minsky Medal
Pluribus로 Science지 커버를 장식하기도 함.(Superhuman AI for Multiplayer Poker(2019 Science),
Superhuman AI for Heads-Up No-Limit Poker: Libratus Beats Top Professionals(2017 Science))

<https://www.cs.cmu.edu/~noamb/media.html>





논문이 나온 배경

알파고

AI의 시대를 연 사건들 중 하나.

알파고는 대국 데이터(사람의 데이터)가 사용됨

또한, 전문가 직접 만든 feature가 사용됨

바둑에만 적용될 수 있는 방법



알파고

AI의 시대를 연 사건들 중 하나.

알파고는 대국 데이터(사람의 데이터)가 사용됨

또한, 전문가 직접 만든 feature가 사용됨

바둑에만 적용될 수 있는 방법

완전 정보 게임에서 사람을 능가하는 인공지능



The background of the slide is a dark, blurred image of a Go board with several black and white stones. The text "AlphaGo Zero" is written in a large, white, sans-serif font, and "Starting from scratch" is written below it in a smaller, white, sans-serif font. Thin white lines connect the text to the Go stones on the board.

AlphaGo Zero

Starting from scratch

Chess, Go, Shogi에 모두 적용될 수 있는
알고리즘

방법의 일반화

사람의 데이터 필요 없음

전문가의 feature를 필요로 하지 않음

완전 정보 게임에 한정됨

The background of the slide is a dark, blurred image of a Go board with several black and white stones. The text "AlphaGo Zero" is prominently displayed in a large, white, sans-serif font. Below it, the phrase "Starting from scratch" is written in a smaller, white, sans-serif font. Thin white lines radiate from the text area towards the bottom right, connecting to the list of features.

AlphaGo Zero

Starting from scratch

Chess, Go, Shogi에 모두 적용될 수 있는
알고리즘

방법의 일반화

사람의 데이터 필요 없음

전문가의 feature가 필요 하지 않음

완전 정보 게임에 한정됨

**불완전 정보 게임에도 적용할 수 있는 방법이
필요함**

Imperfect-Information Games



Perfect-Information Games



완전 정보 게임

- 게임의 정보가 참가자 모두에게 공개된 게임
- 숨어 있는 정보가 없다.

예) 바둑, 오목, 체스 등

불완전 정보 게임

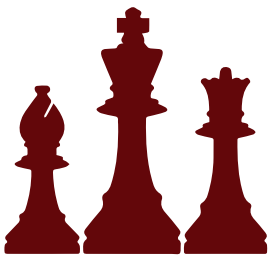
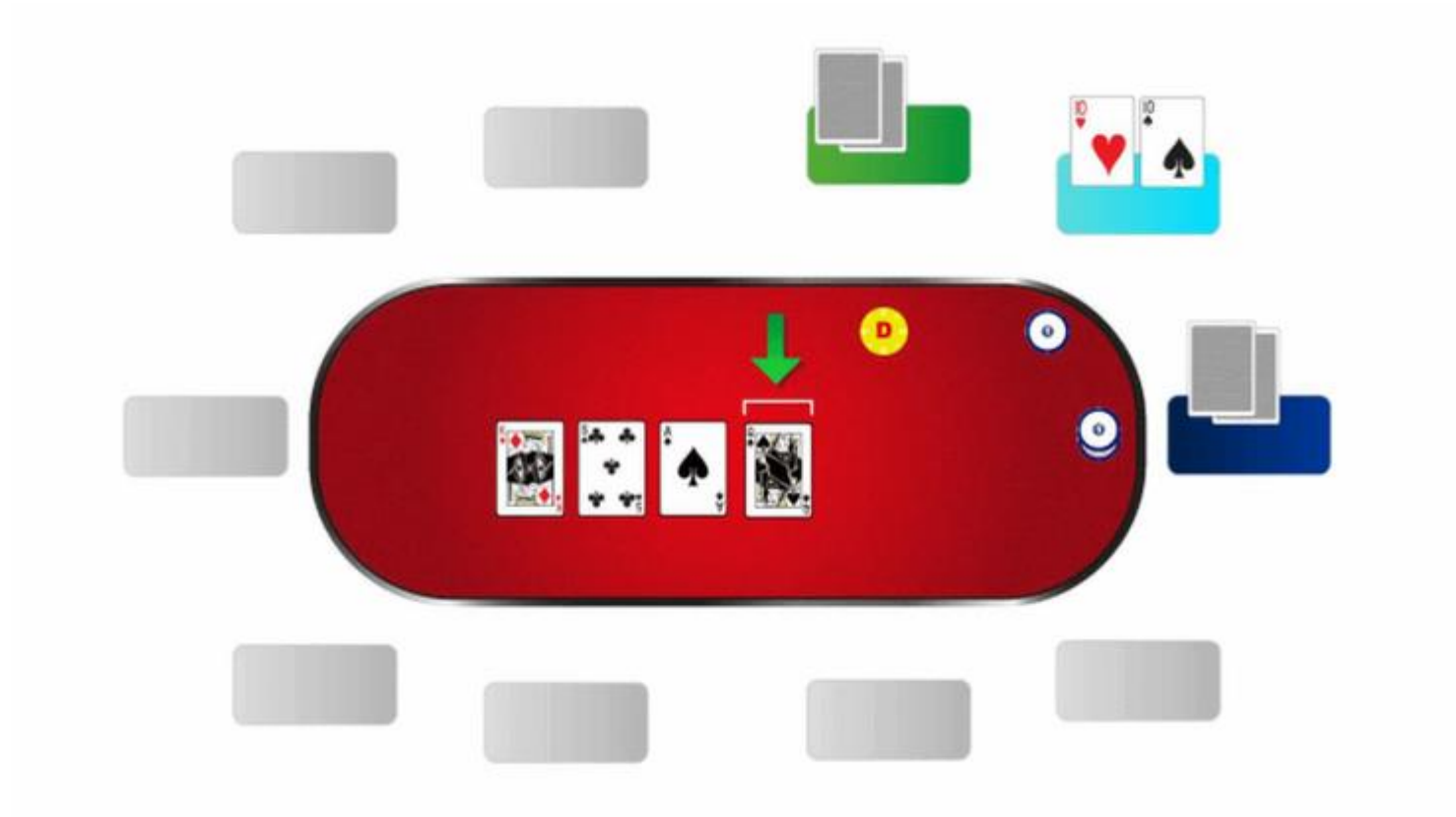
- 확률적으로 정보가 주어진다.
- 모든 정보가 공개되어 있지는 않다.

예) 가위바위보, 텍사스 홀덤 등

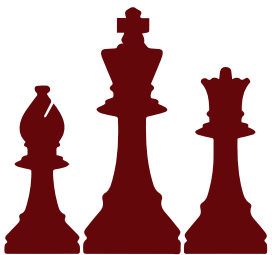
텍사스 홀덤



텍사스 홀덤



텍사스 홀덤



텍사스 홀덤



AI와 게임 이론에 있어서 오랫동안 도전 되어 온 문제.

2017: AI는 2명의 no-limit hold'em에서 사람을 이김.

2019: AI는 6명의 no-limit hold'em에서 사람을 이김

알파고/알파제로와는 다른 방식으로 발전되어 왔음



ReBeL

완전 정보 게임과 불완전 정보 게임에 모두 적용될 수 있는 알고리즘

ReBeL (Recursive Belief-based Learning)

- 2 player zero-sum game에서 내쉬 균형으로 수렴함
- 2 player no-limit hold'em poker에서 사람을 능가함
- 기존 포커 AI는 사전 지식이 사용되는 반면에 ReBeL은 자가 학습이 가능함
- 완전 정보 게임에서 ReBeL은 알파제로와 유사해짐



ReBeL

완전 정보 게임과 불완전 정보 게임에 모두 적용될 수 있는 알고리즘

ReBeL (Recursive Belief-based Learning)

- 2 player zero-sum game에서 내쉬 균형으로 수렴함
- 2 player no-limit hold'em poker에서 사람을 능가함
- 기존 포커 AI는 사전 지식이 사용되는 반면에 ReBeL은 자가 학습이 가능함
- 완전 정보 게임에서 ReBeL은 알파제로와 유사해짐

알파제로를 불완전 정보 게임으로 일반화



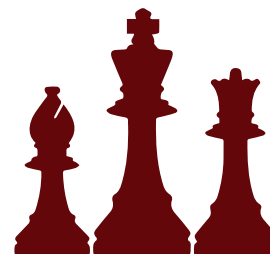
알파제로

완전 정보 게임에서는 **State value**가 유일하게 나옵니다.
(특정 지점으로부터 두 플레이어가 최적의 플레이를 할 경우)

Value network는 State를 input으로 받고 State value를
output하는 함수로 모델링 됩니다.

알파제로에서 Value network는 자가 학습을 통해 학습하게
됩니다.

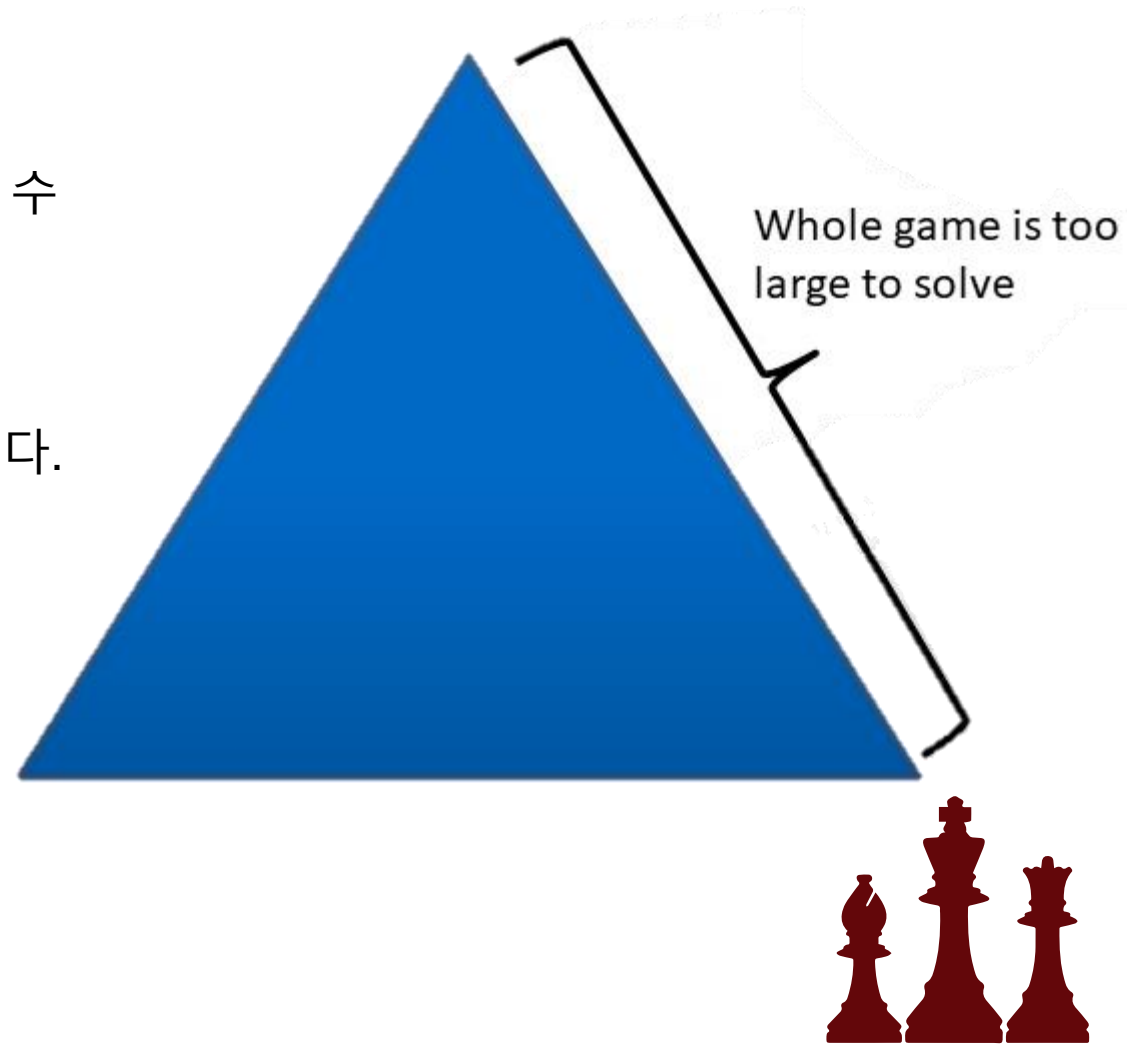
$$f_{white}(\text{State}) = 1$$



알파제로

이론적으로, Backward Induction만이 완전 정보 게임을 풀 수 있다.

그러나, 실제로는 너무 연산량이 많기 때문에 이는 쉽지 않다.



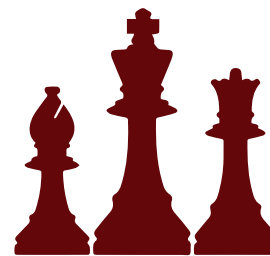
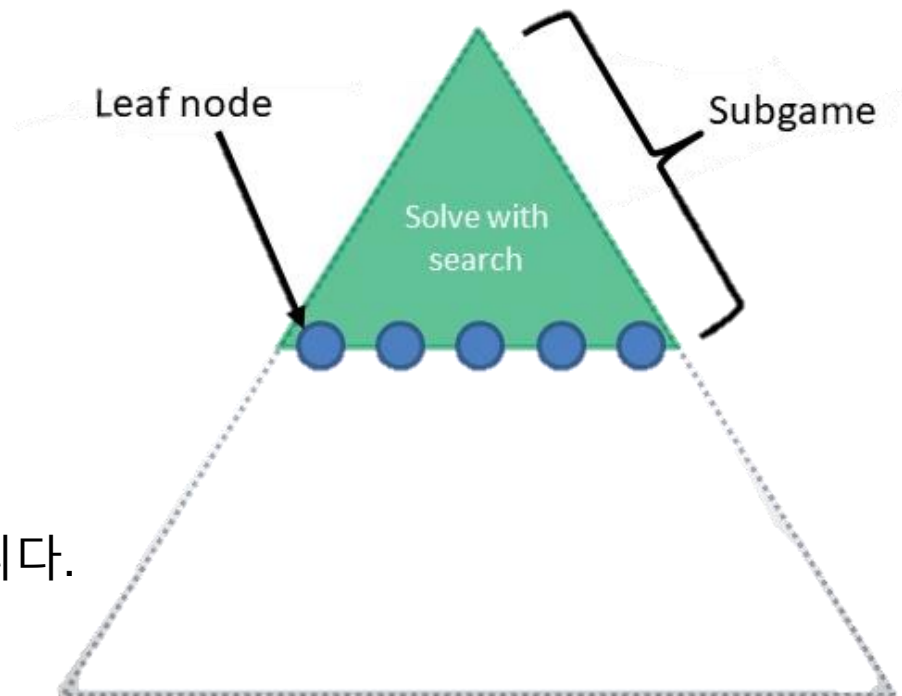
알파제로

대신에, AI는 Search라는 것을 하게 됩니다.

1. 앞에 10수를 미리 보고
2. Value network를 통해 상태의 가치를 추정합니다.
3. State Value들을 이용하여 Backward Induction을 진행합니다.

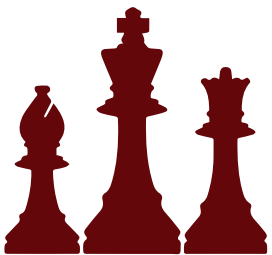
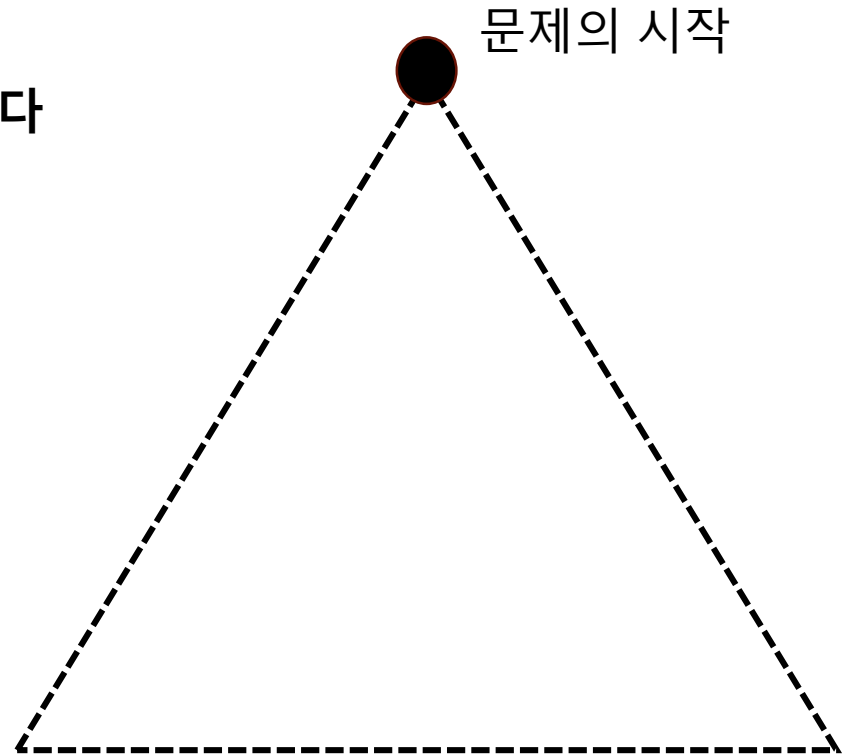
다른 말로 한다면 Subgame을 푸는 것이라고 할 수 있습니다.

Value network가 완벽하다면 최적의 행동을 하게 될 것입니다.



알파제로

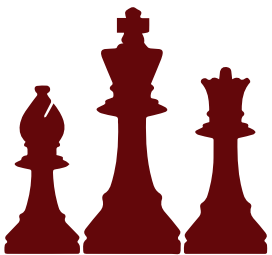
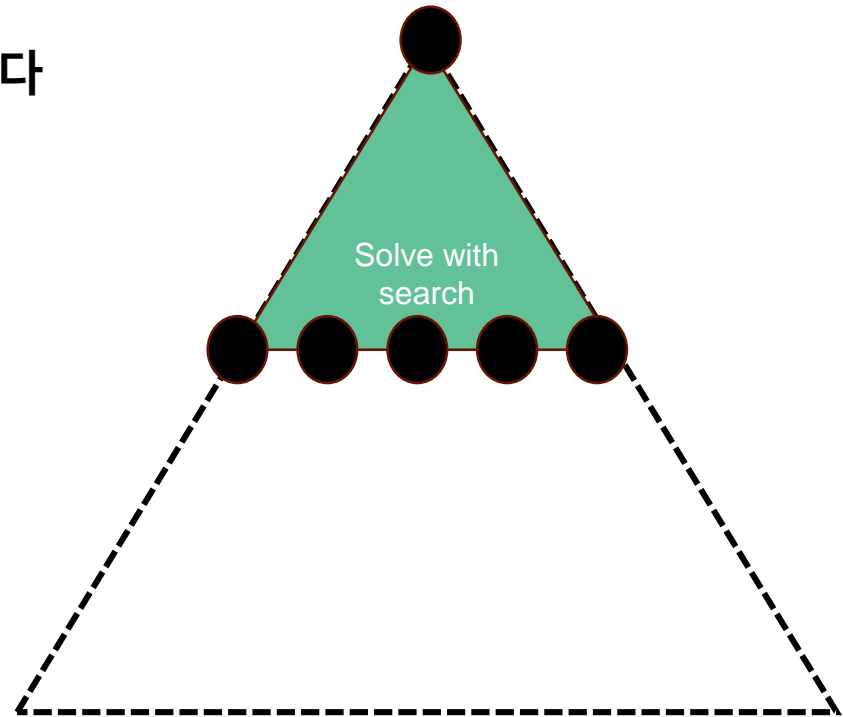
Agent가 action을 할 때마다 서브게임을 만들고 이를 풉니다
- Leaf node의 가치를 Value net을 기반으로 정해줍니다.



알파제로

Agent가 action을 할 때마다 서브게임을 만들고 이를 풉니다

- Leaf node의 가치를 Value net을 기반으로 정해줍니다.

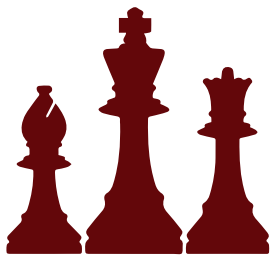
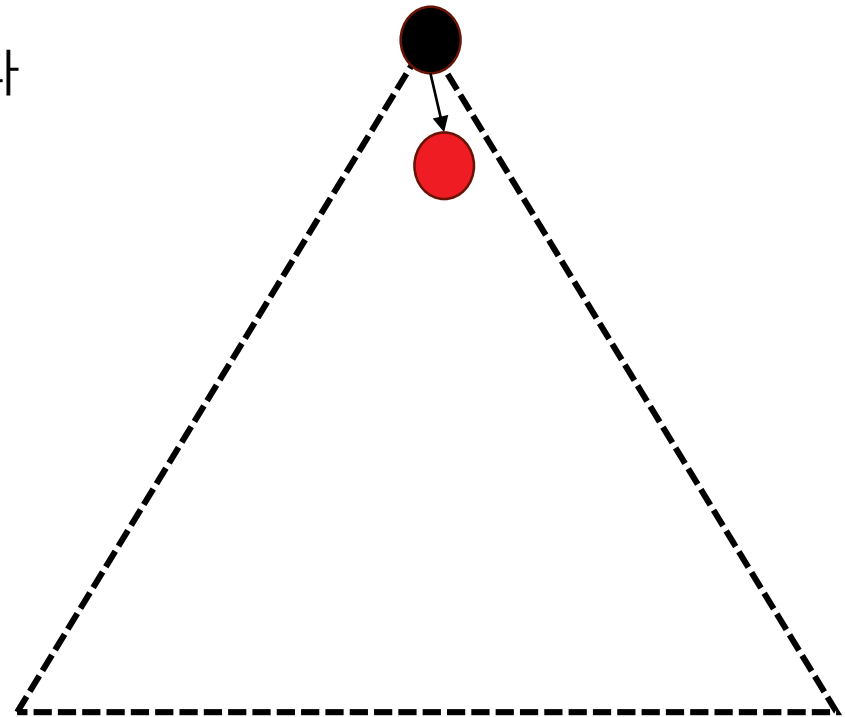


알파제로

Agent가 action을 할 때마다 서브게임을 만들고 이를 풉니다

- Leaf node의 가치를 Value net을 기반으로 정해줍니다.

서브게임의 해답을 갖고 다음 Action을 행해줍니다.



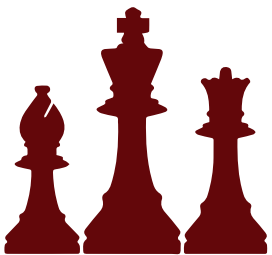
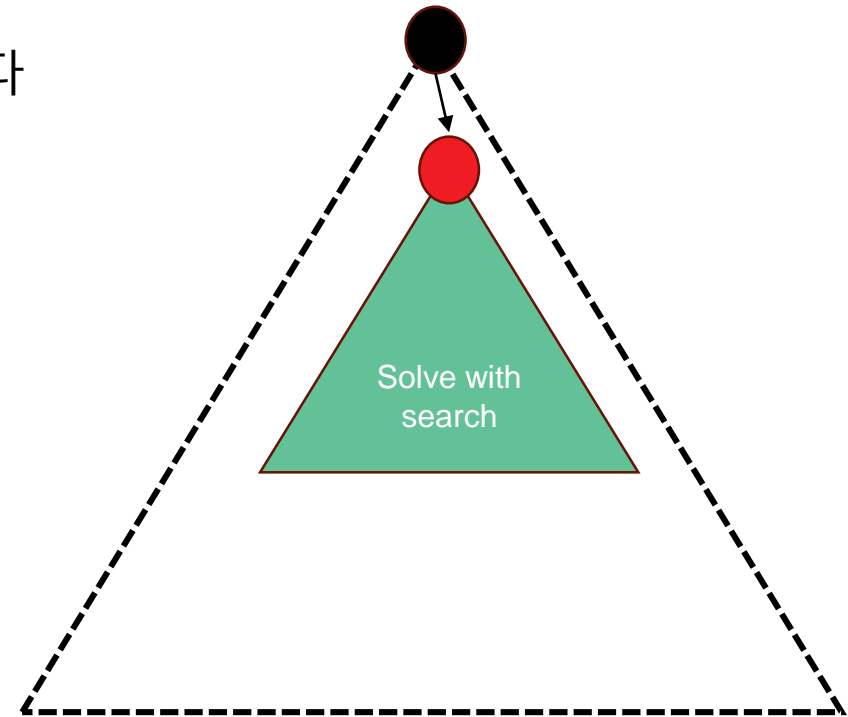
알파제로

Agent가 action을 할 때마다 서브게임을 만들고 이를 풉니다

- Leaf node의 가치를 Value net을 기반으로 정해줍니다.

서브게임의 해답을 갖고 다음 Action을 행해줍니다.

다시 서브 게임을 만들고 풀고 Action을 행해줍니다.



알파제로

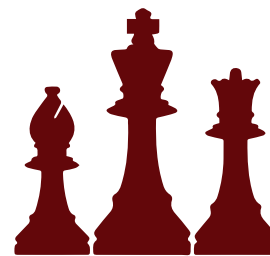
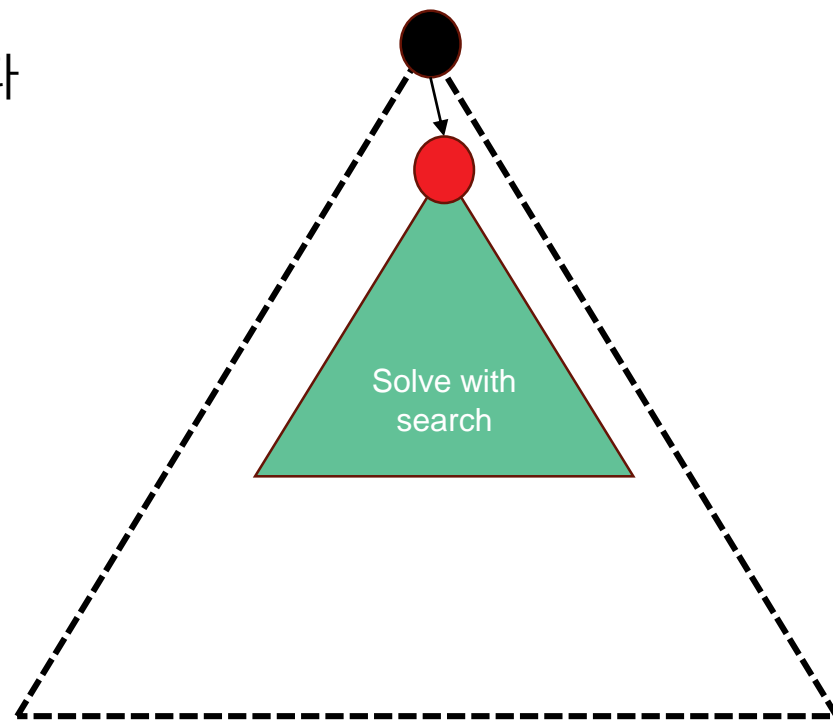
Agent가 action을 할 때마다 서브게임을 만들고 이를 풉니다

- Leaf node의 가치를 Value net을 기반으로 정해줍니다.

서브게임의 해답을 갖고 다음 Action을 행해줍니다.

다시 서브 게임을 만들고 풀고 Action을 행해줍니다.

(이를 게임이 끝날 때까지 반복해줍니다.)



알파제로

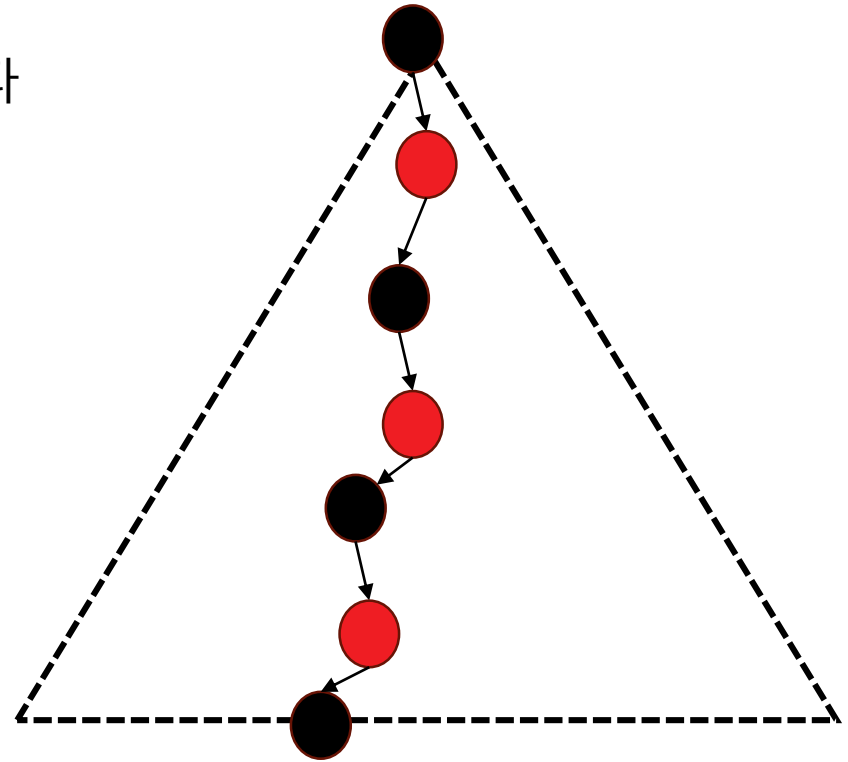
Agent가 action을 할 때마다 서브게임을 만들고 이를 풉니다

- Leaf node의 가치를 Value net을 기반으로 정해줍니다.

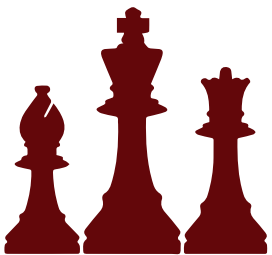
서브게임의 해답을 갖고 다음 Action을 행해줍니다.

다시 서브 게임을 만들고 풀고 Action을 행해줍니다.

(이를 게임이 끝날 때까지 반복해줍니다.)

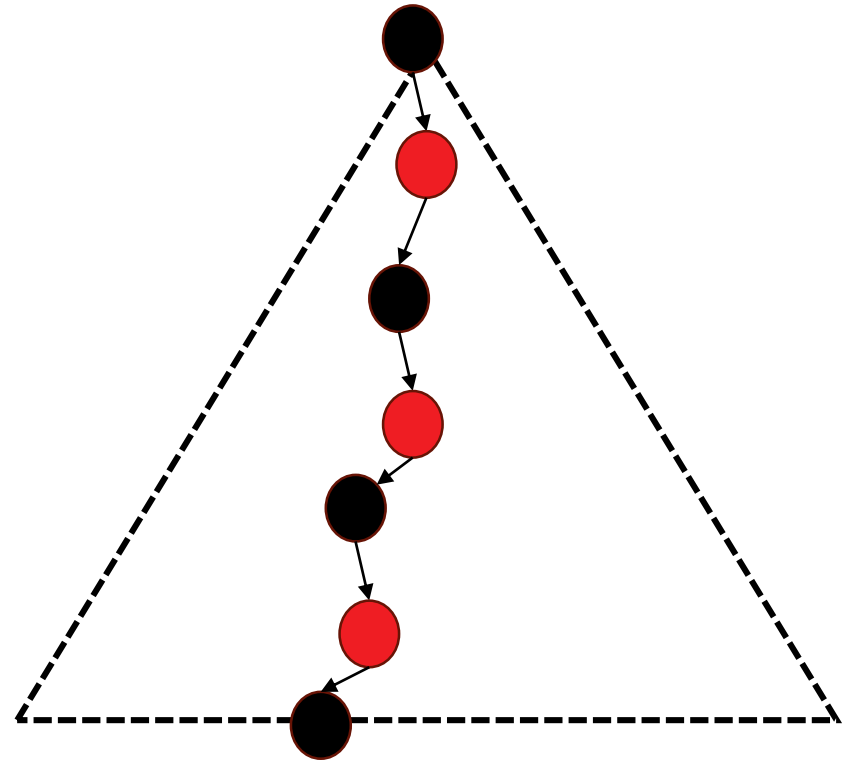


Black Win



알파제로

약간의 Random Exploration을 통해 알파제로는 결국 모든 State에 도달하게 되고, 모든 State의 진짜 가치를 파악할 수 있게 됩니다.



Black Win



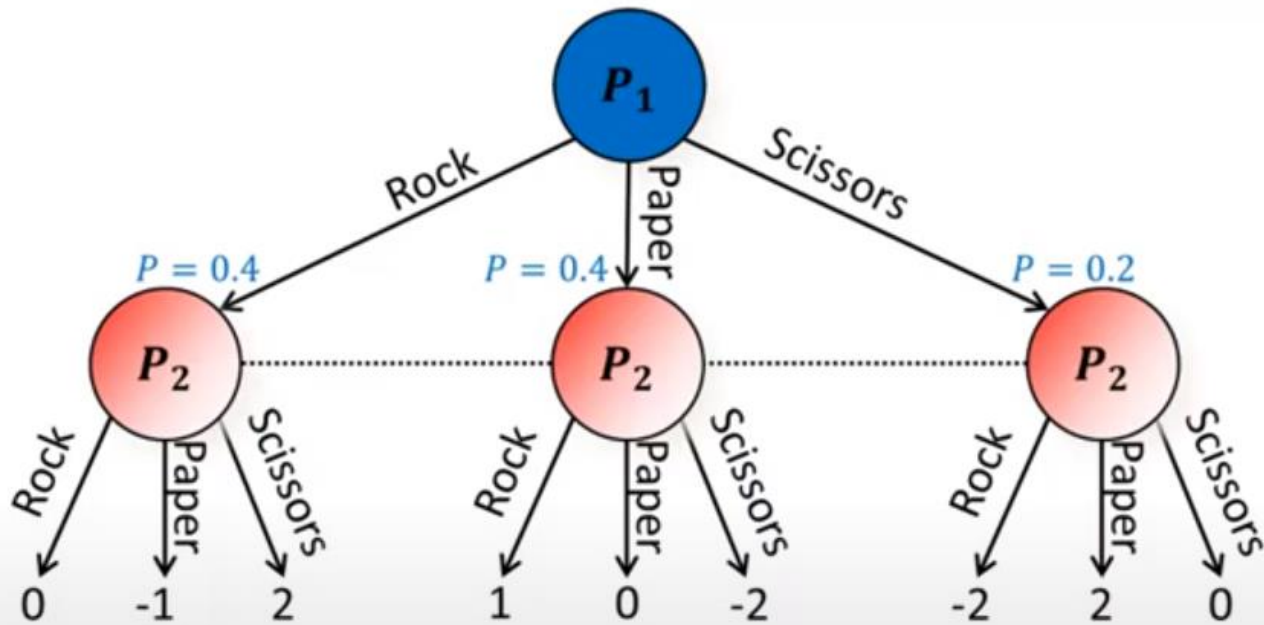
Imperfect information game에서는 State가 유일한 값이 아닙니다

따라서, 알파제로를 그대로 적용할 수 없습니다.



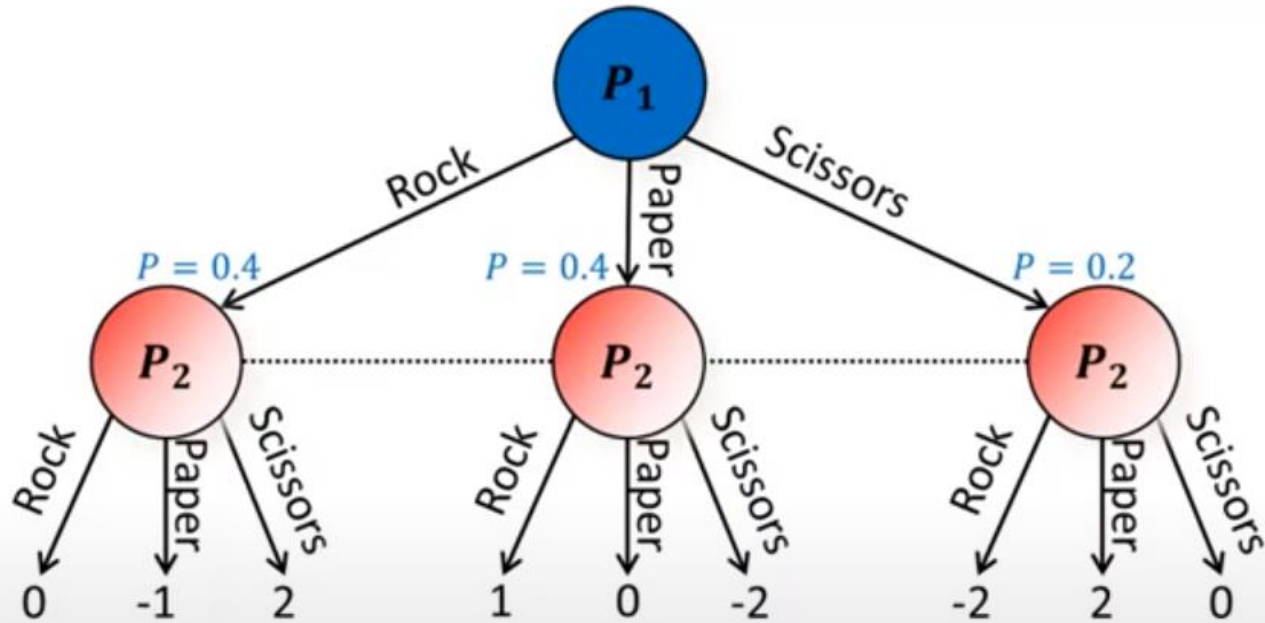
불완전 정보 게임에서 서치

Rock-Paper-Scissors+

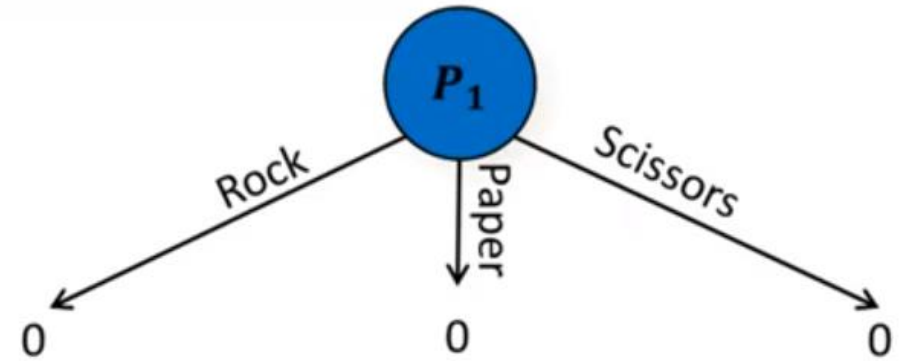


불완전 정보 게임에서 서치

Rock-Paper-Scissors+

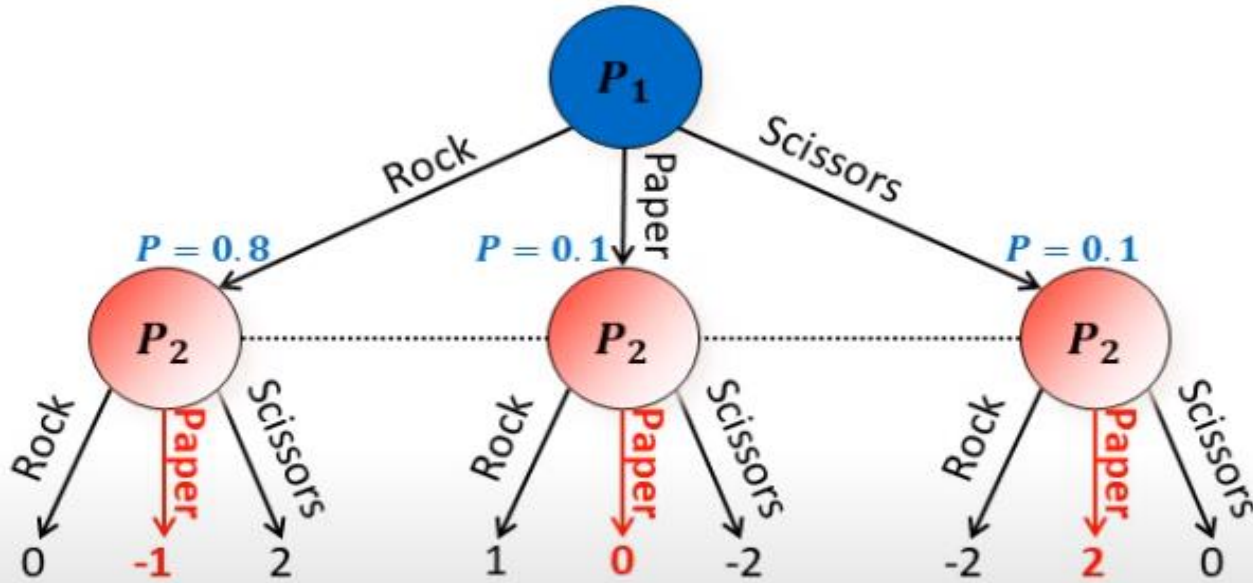


Depth-Limited Rock-Paper-Scissors+

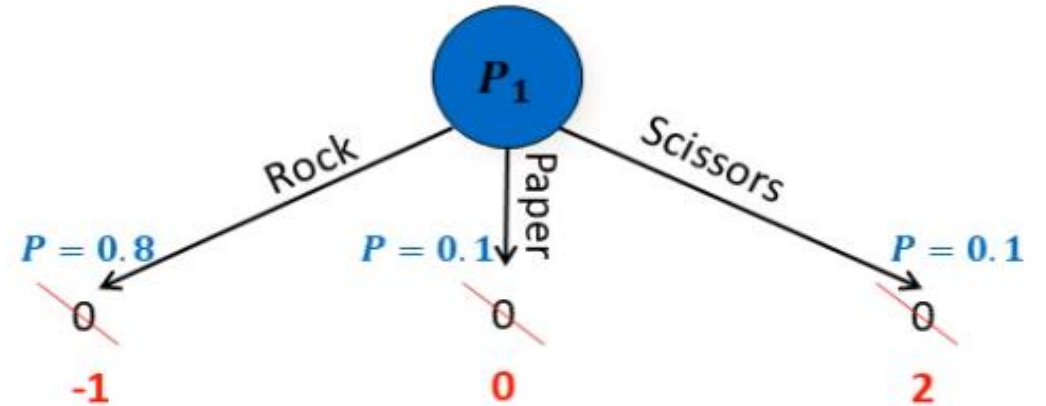


불완전 정보 게임에서 서치

Rock-Paper-Scissors+



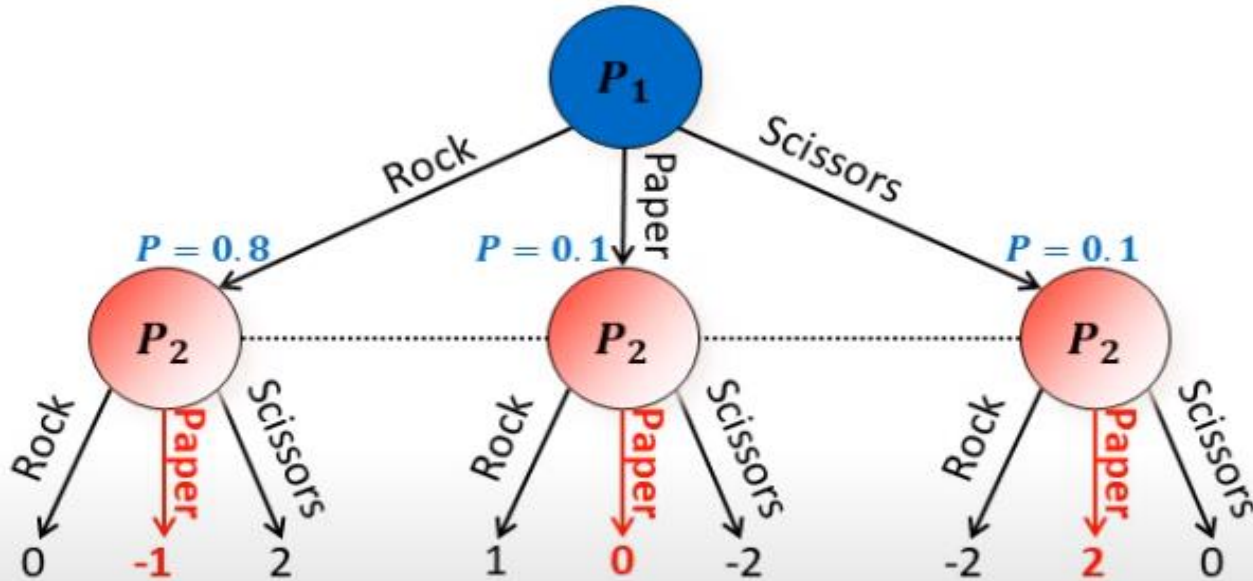
Depth-Limited Rock-Paper-Scissors+



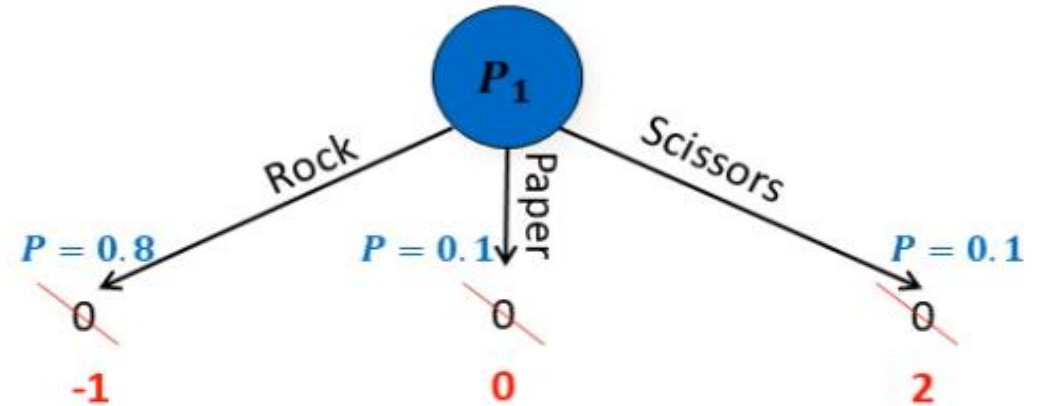
가정: 전체 Policy는 Common한 Knowledge이다. 그러나, 랜덤 프로세스의 결과는 Common Knowledge가 아니다.

불완전 정보 게임에서 서치

Rock-Paper-Scissors+

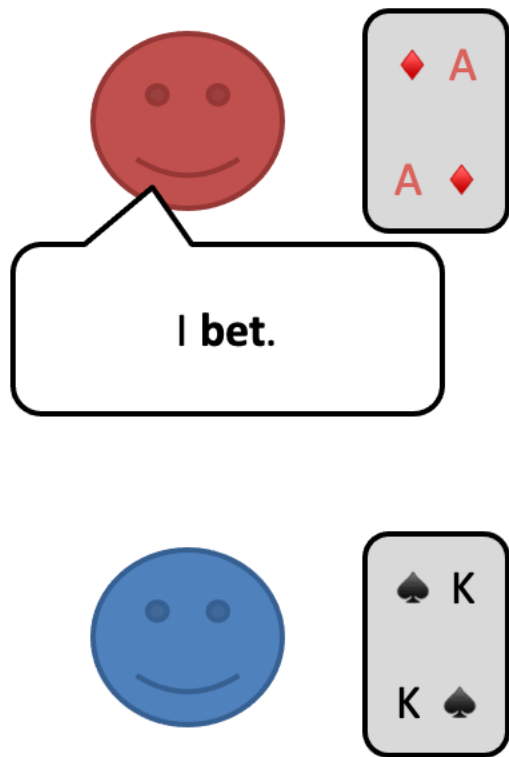


Depth-Limited Rock-Paper-Scissors+

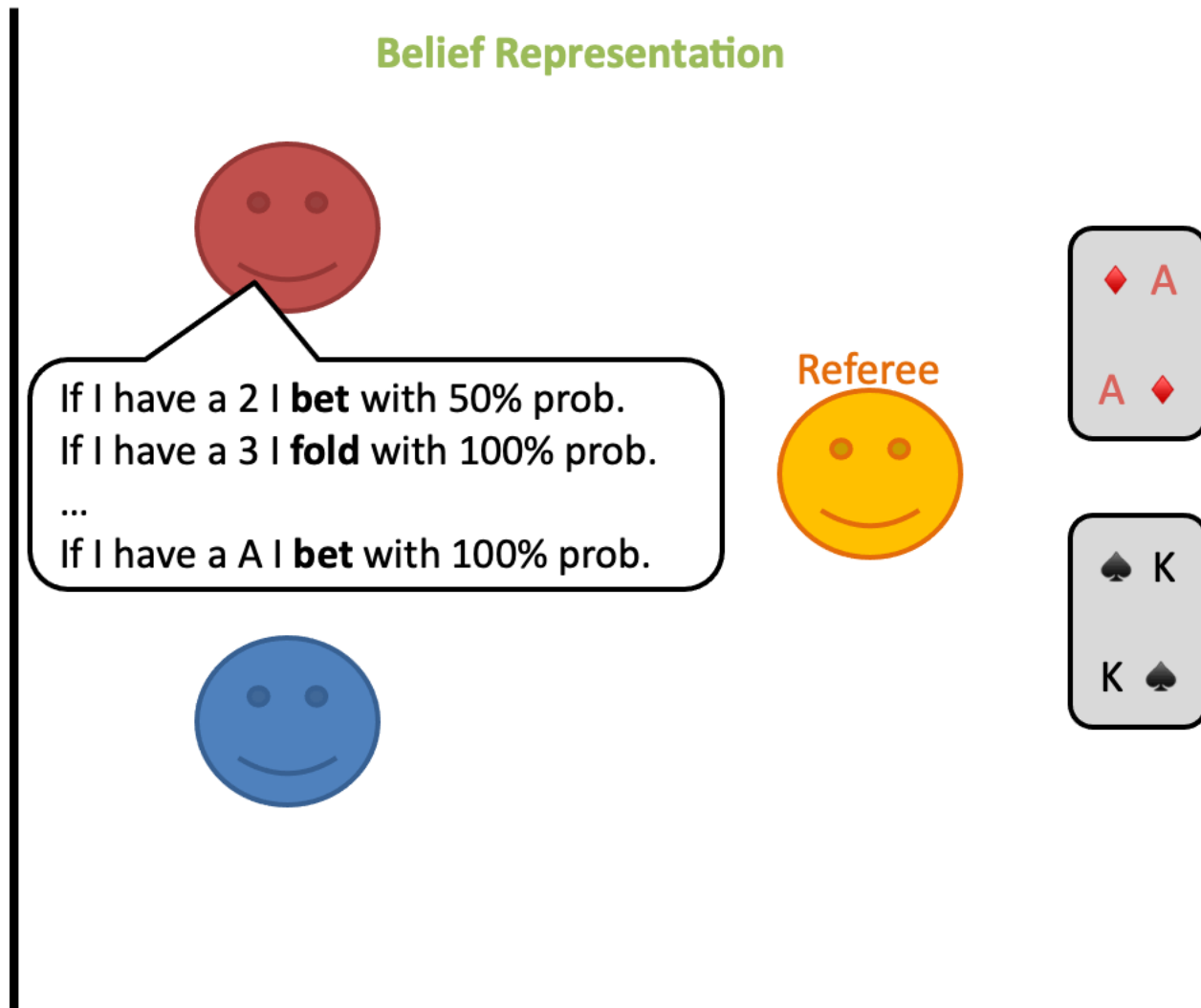


$v(\text{Rock})$ 은 잘 정의되지 않는다.
반면, $v([0.8 \text{ Rock}, 0.1 \text{ Paper}, 0.1 \text{ Scissors}]) = -0.6$

Discrete Representation



Belief Representation





I **bet** with my 2.
I **fold** with my 3.
...
I **bet** with my A.

$w(2) = \frac{1}{13}$ $w(3) = \frac{1}{13}$ $w(A) = \frac{1}{13}$

♠ 2
2 ♠

♠ 3
3 ♠

...

♠ A
A ♠

$w(2) = \frac{1}{13}$ $w(3) = \frac{1}{13}$ $w(A) = \frac{1}{13}$


♠ 2
2 ♠

♠ 3
3 ♠

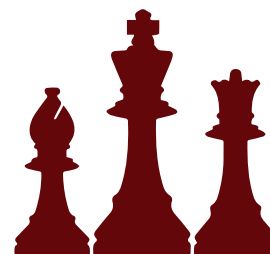
...


♠ A
A ♠

Referee



$$P(\text{fold}) = 0.08 = \frac{\sum_s P(\text{fold}|s)w(s)}{\sum_s w(s)}$$

$$P(\text{bet}) = 0.92 = \frac{\sum_s P(\text{bet}|s)w(s)}{\sum_s w(s)}$$




I **bet** with my 2.
I **fold** with my 3.
...
I **bet** with my A.



$w(2) = \frac{1}{13}$ $w(3) = \frac{1}{13}$ $w(A) = \frac{1}{13}$

♠ 2
2 ♠

♠ 3
3 ♠

...

♠ A
A ♠

$w(2) = \frac{1}{13}$ $w(3) = \frac{1}{13}$ $w(A) = \frac{1}{13}$

♠ 2
2 ♠


♠ 3
3 ♠

...

♠ A
A ♠

Player 1 bet

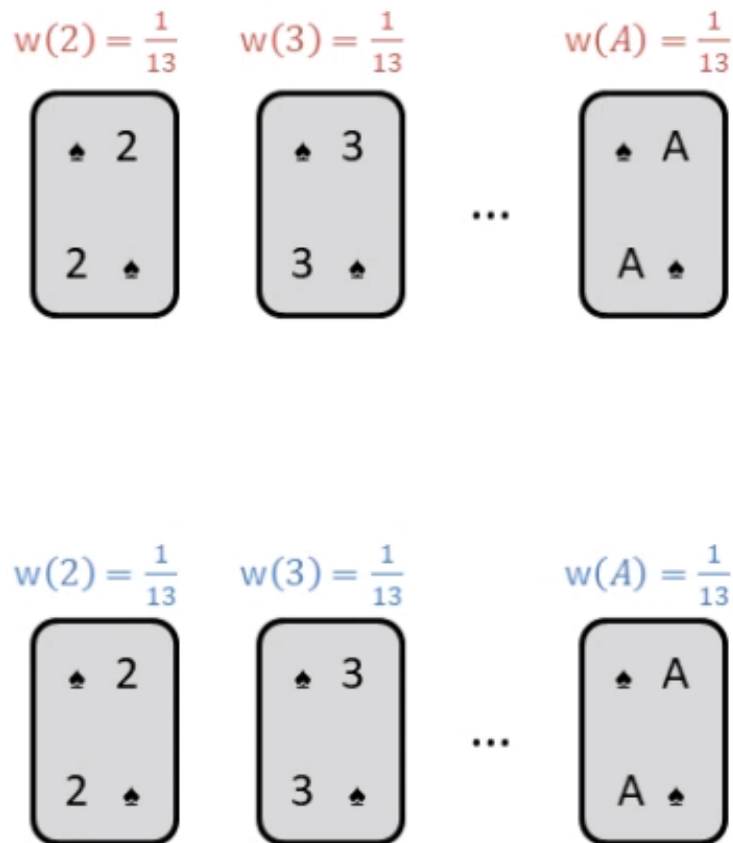
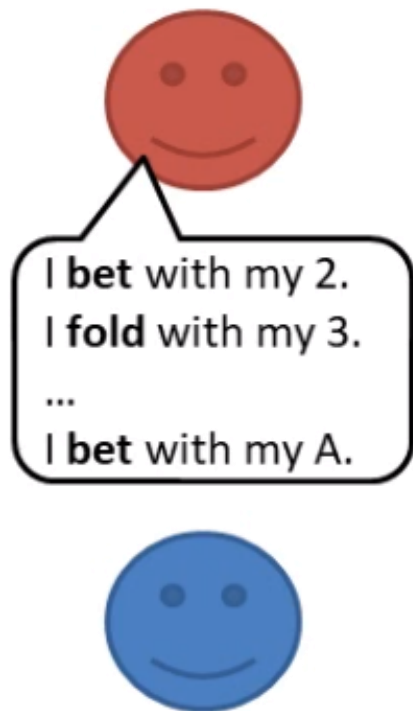
Referee



$$P(fold) = 0.08 = \frac{\sum_s P(fold|s)w(s)}{\sum_s w(s)}$$

$$P(bet) = 0.92 = \frac{\sum_s P(bet|s)w(s)}{\sum_s w(s)}$$


베이지안 룰을 통해 업데이트



Player 1 bet

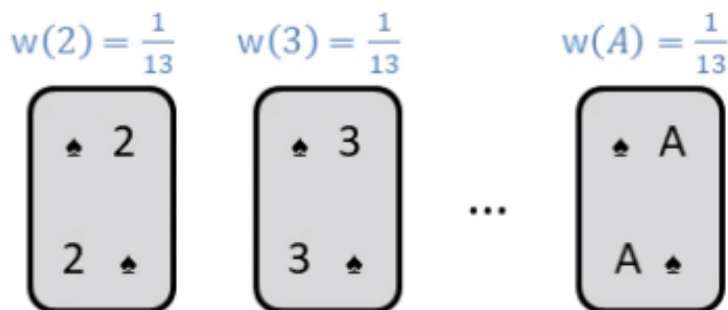
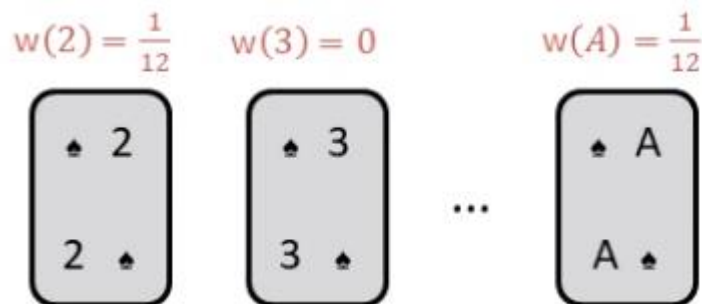
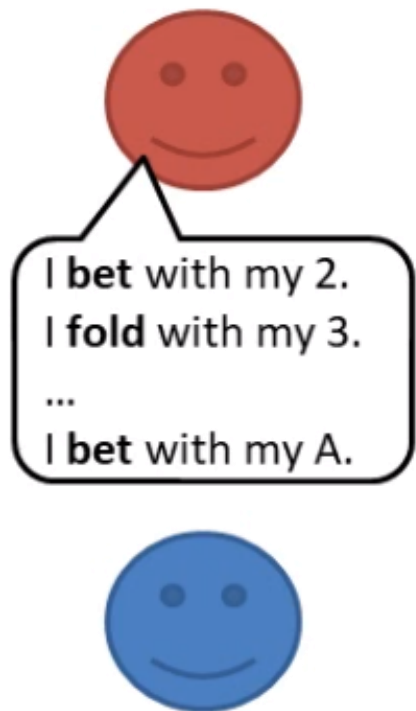
Referee

$$P(fold) = 0.08 = \frac{\sum_s P(fold|s)w(s)}{\sum_s w(s)}$$

$$P(bet) = 0.92 = \frac{\sum_s P(bet|s)w(s)}{\sum_s w(s)}$$



베이지안 룰을 통해 업데이트(완)



Player 1 bet

Referee

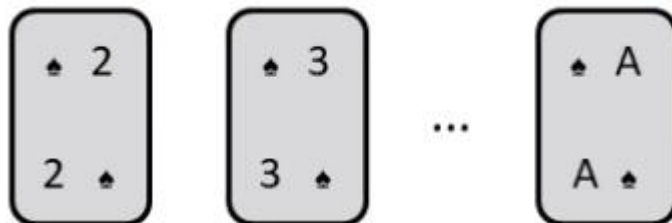
$$P(\text{fold}) = 0.08 = \frac{\sum_s P(\text{fold}|s)w(s)}{\sum_s w(s)}$$

$$P(\text{bet}) = 0.92 = \frac{\sum_s P(\text{bet}|s)w(s)}{\sum_s w(s)}$$

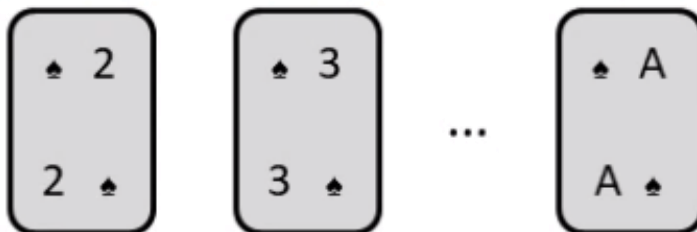


베이지안 룰을 통해 업데이트(완)

$$w(2) = \frac{1}{12} \quad w(3) = 0 \quad w(A) = \frac{1}{12}$$



$$w(2) = \frac{1}{13} \quad w(3) = \frac{1}{13} \quad w(A) = \frac{1}{13}$$



Player 1 bet

Referee

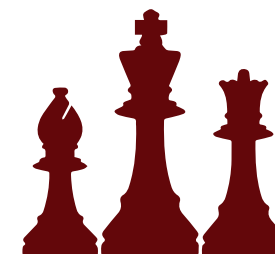
$$P(fold) = 0.08 = \frac{\sum_s P(fold|s)w(s)}{\sum_s w(s)}$$

$$P(bet) = 0.92 = \frac{\sum_s P(bet|s)w(s)}{\sum_s w(s)}$$

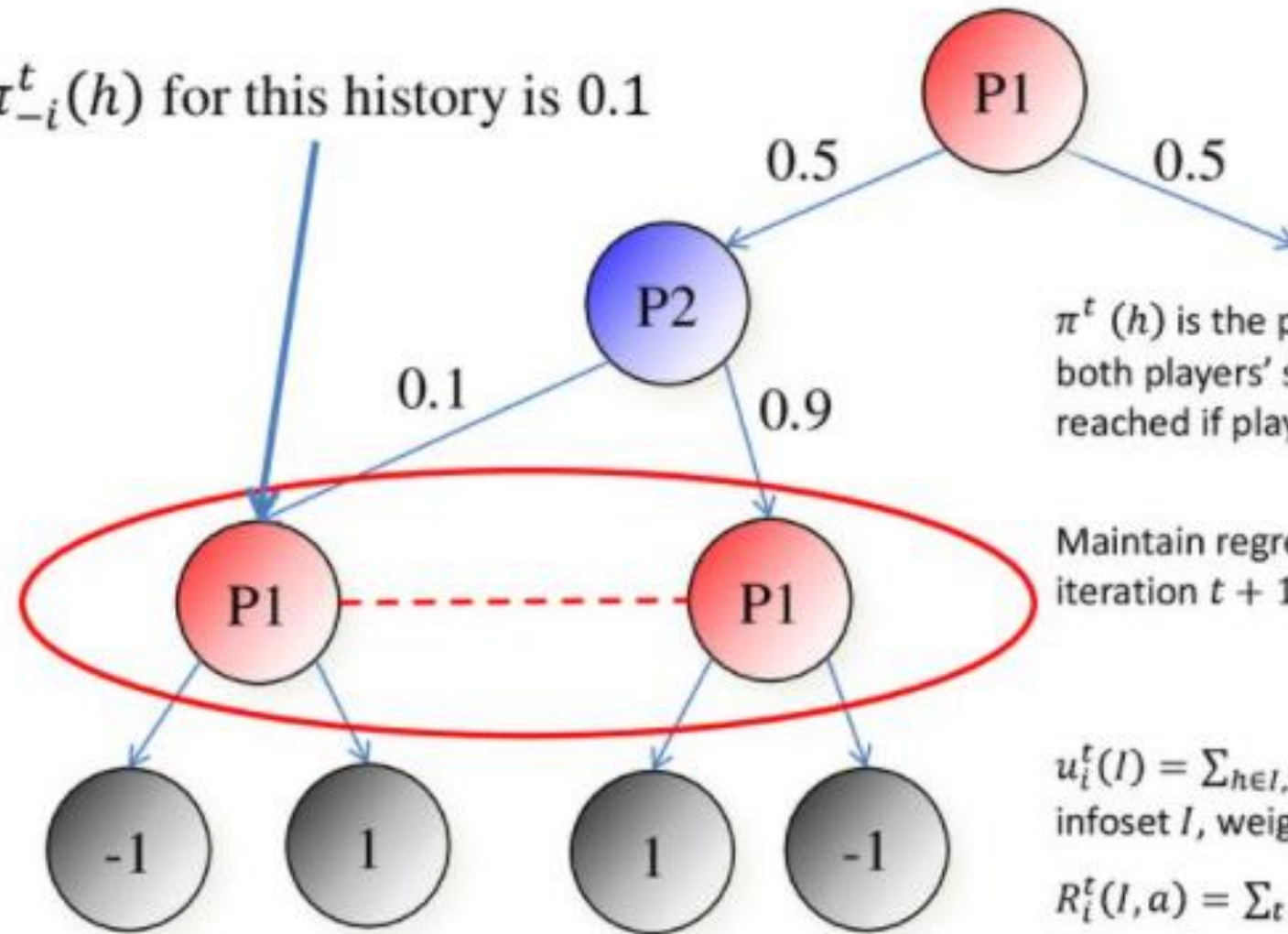
I bet with my 2.
I fold with my 3.
...
I bet with my A.

Public Belief Action

Public Belief State



$\pi_{-i}^t(h)$ for this history is 0.1



$\pi^t(h)$ is the probability node h is reached according to both players' strategies. $\pi_{-i}^t(h)$ is the probability h is reached if player i tried to get there.

Maintain regret vector for actions in each info set I . On iteration $t + 1$, actions chosen with probability

$$p^{t+1}(I, a) = \frac{\max\{0, R^t(I, a)\}}{\sum_{a'} \max\{0, R^t(I, a')\}}$$

$u_i^t(I) = \sum_{h \in I, z \in Z} \pi_{-i}^t(h) \pi^t(h, z) u_i(z)$ is the reward for info set I , weighted by π_{-i}^t

$R_i^t(I, a) = \sum_{\tau} (u_i^t(I \rightarrow a) - u_i^t(I))$ is regret for action a in info set I

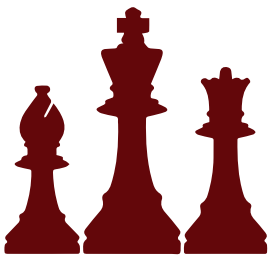
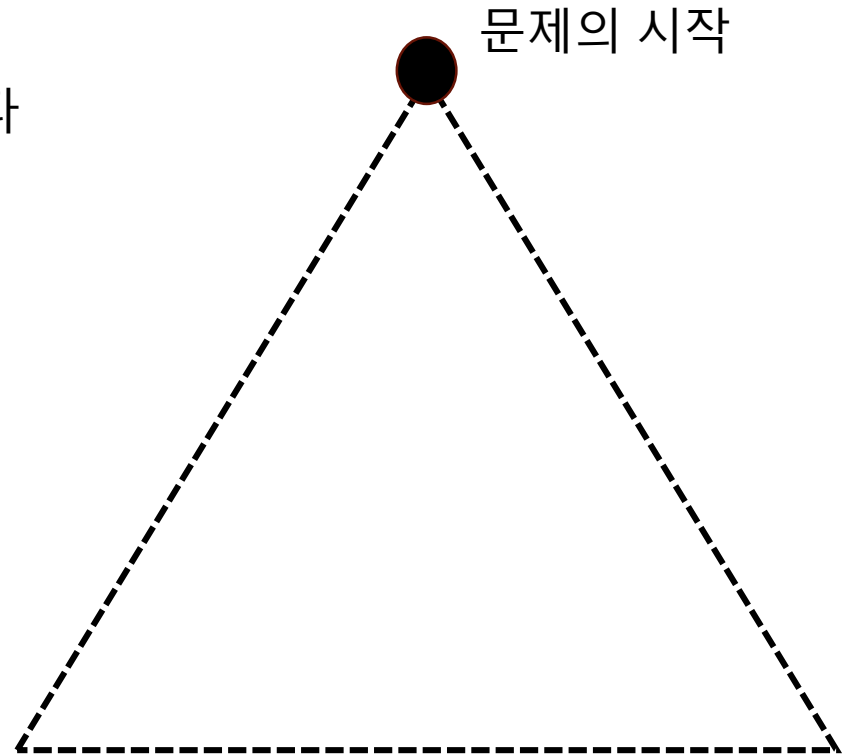
Theorem: regret for the whole game is bounded by regret summed over all info sets. $R_i^t \leq \sum_I R_i^t(I)$



ReBeL

Agent가 action을 할 때마다 서브게임을 만들고 이를 풉니다

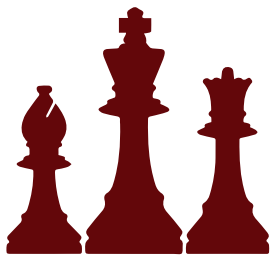
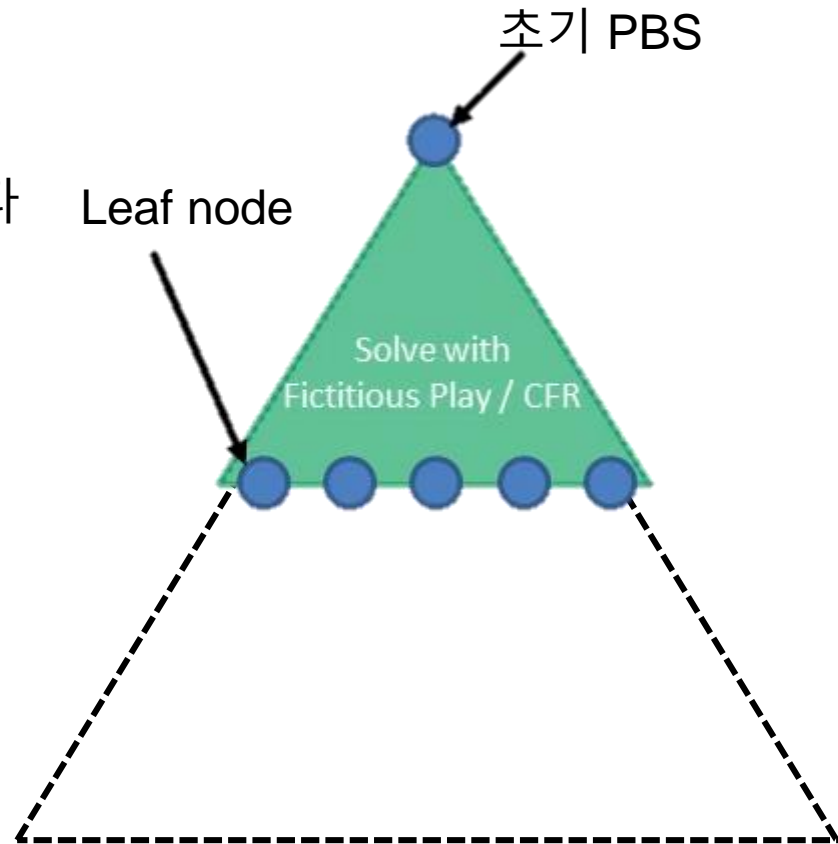
- 문제를 Fictitious Play나 CFR을 통해 풀어줍니다.
- 다음 Action을 합니다.



ReBeL

Agent가 action을 할 때마다 서브게임을 만들고 이를 풉니다

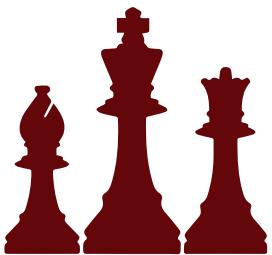
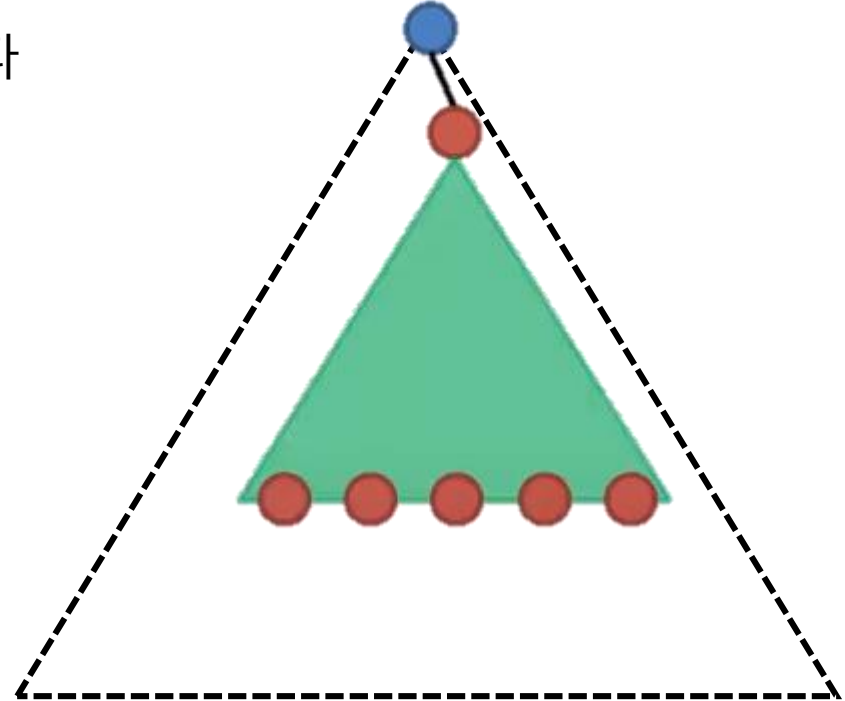
- 문제를 Fictitious Play나 CFR을 통해 풀어줍니다.
- 다음 Action을 합니다.



Agent가 action을 할 때마다 서브게임을 만들고 이를 풉니다

- 문제를 Fictitious Play나 CFR을 통해 풀어줍니다.
- 다음 Action을 합니다.

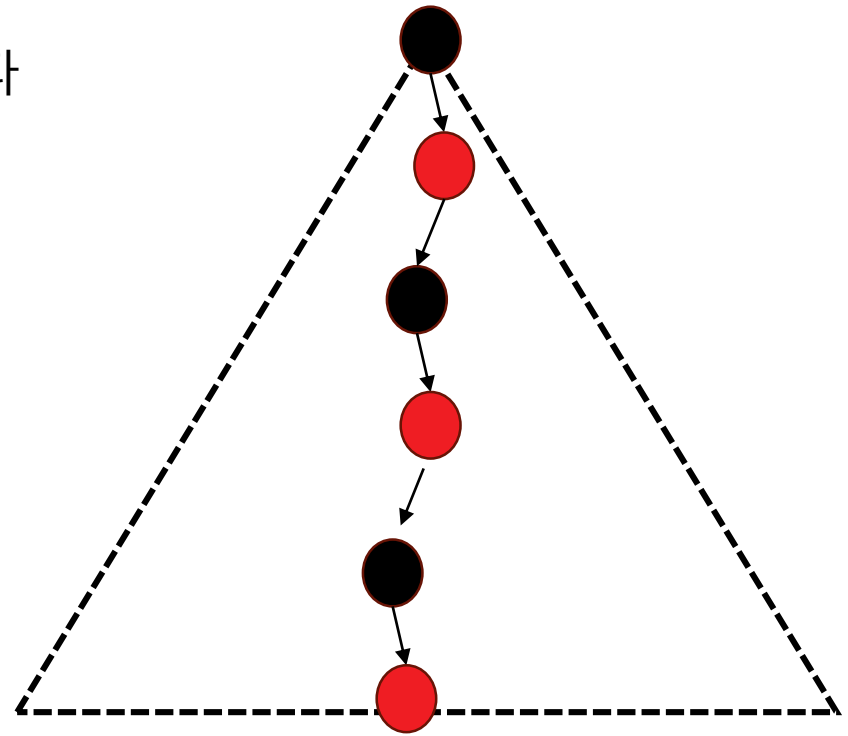
이를 끝날 때까지 반복해줍니다.



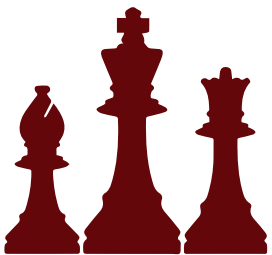
Agent가 action을 할 때마다 서브게임을 만들고 이를 풉니다

- 문제를 Fictitious Play나 CFR을 통해 풀어줍니다.
- 다음 Action을 합니다.

이를 끝날 때까지 반복해줍니다.

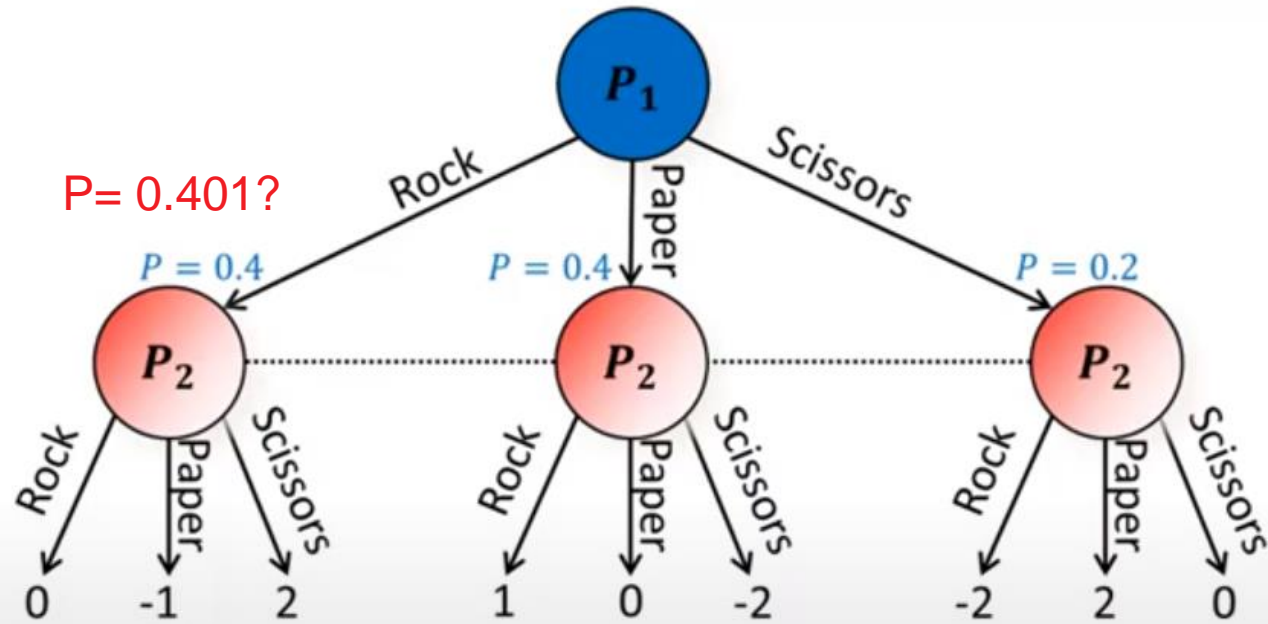


Red Win



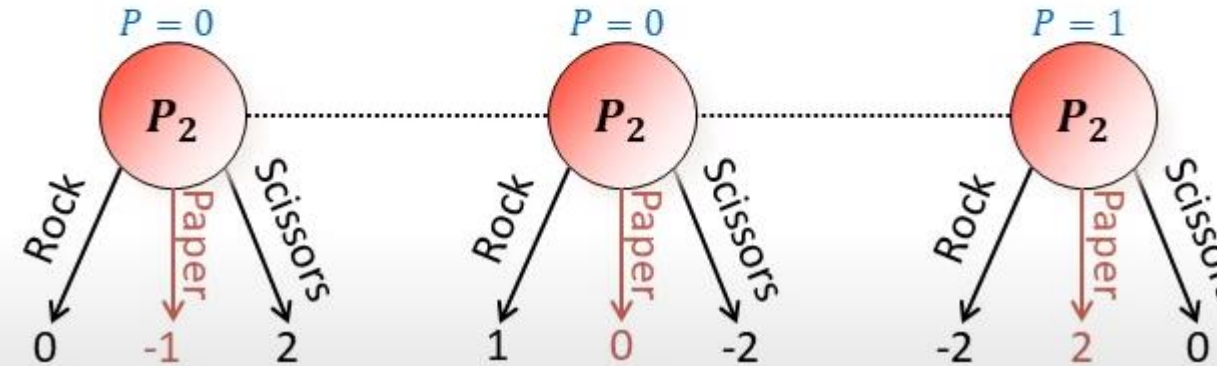
테스팅

Rock-Paper-Scissors+



$P = 0.401?$

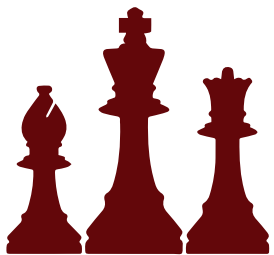
Rock-Paper-Scissors+ Subgame



Testing에서는 학습에서와 같이 Policy network를 사용합니다.

Results in Two Player No Limit Texas Holdem

	Slumbot	Baby Tartanian8	Local Best Response	Top Humans
DeepStack			383 ± 112	
Libratus		63 ± 14		147 ± 39
Modicum	11 ± 5	6 ± 3		
ReBeL	45 ± 5	9 ± 4	881 ± 94	165 ± 69



Results in Two Player Liar's Dice

	1 die, 4 faces	1 die, 5 faces	1 die, 6 faces	2 dice, 3 faces
Tabular Full-Game FP	0.012	0.024	0.039	0.057
Tabular Full-Game CFR	0.001	0.001	0.002	0.002
ReBeL with FP	0.041	0.020	0.040	0.020
ReBeL with CFR	0.017	0.015	0.024	0.017



Q: Noam Brown은 왜 이 논문을
NSC에 못냈을까요?

