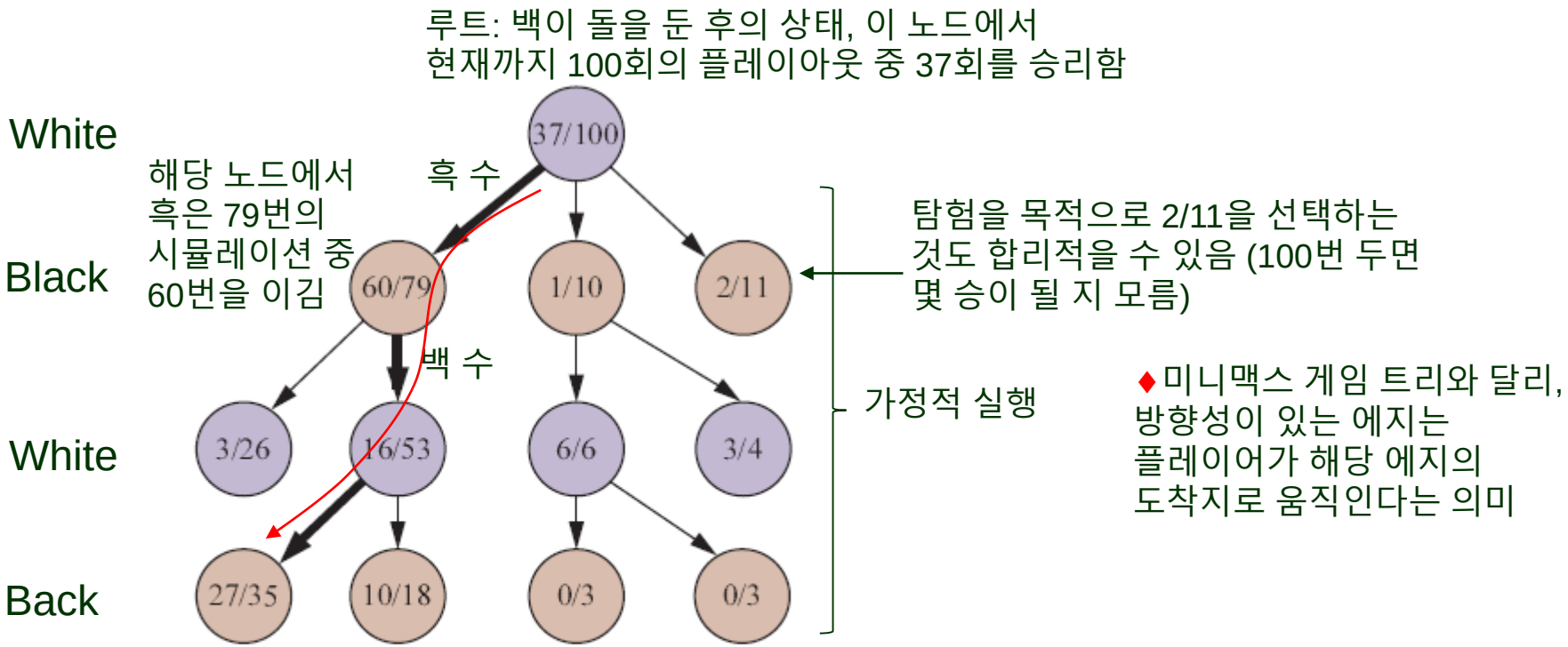
몬테 칼로 트리 탐색(MCTS)

Casino de Monte-Carlo, Monaco

- 온라인 계획 알고리즘
- 탐험(exploration)과 이용(exploitation)을 균형있게 조절하는 지능형 트리 탐색
- 시뮬레이션(플레이아웃) 방식의 무작위 샘플링
- 미래의 선택을 개선하기 위해 행동 통계를 저장
- 신경망과 결합하여 알파고와 같은 게임, 일정 관리, 계획 및 최적화와 같은 주요 AI 응용 프로그램의 많은 성공에 기여

1 단계: 선택

루트에서 흑 돌이 둘 수 있는 수를 결정



(a) Selection

2 & 3 단계: 확장 & 시뮬레이션

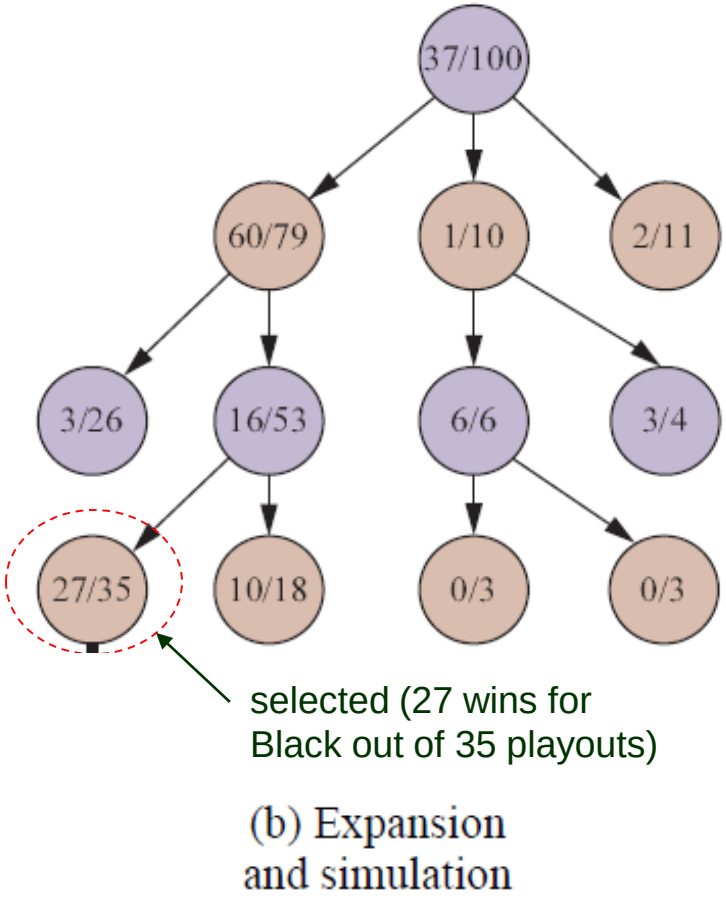
White

Black

White

Black

White



(b) Expansion and simulation

- 선택한 노드에서 하나 이상의 자식 노드 생성
  - ♦ 가능한 모든 행동을 실행하여 자식 노드 생성
  - ♦ 가능한 행동의 일부만 실행하여 자식 노드 생성 가능
  - ♦ 여러 개의 실행 가능한 행동이 있는 경우 여러가지 자식 노드 생성 시도

- 예를 들어, 자식 노드 중 하나인 n이 (무작위로) 선택됨

- 노드 n에서 플레이아웃 수행

4 단계: 역전파

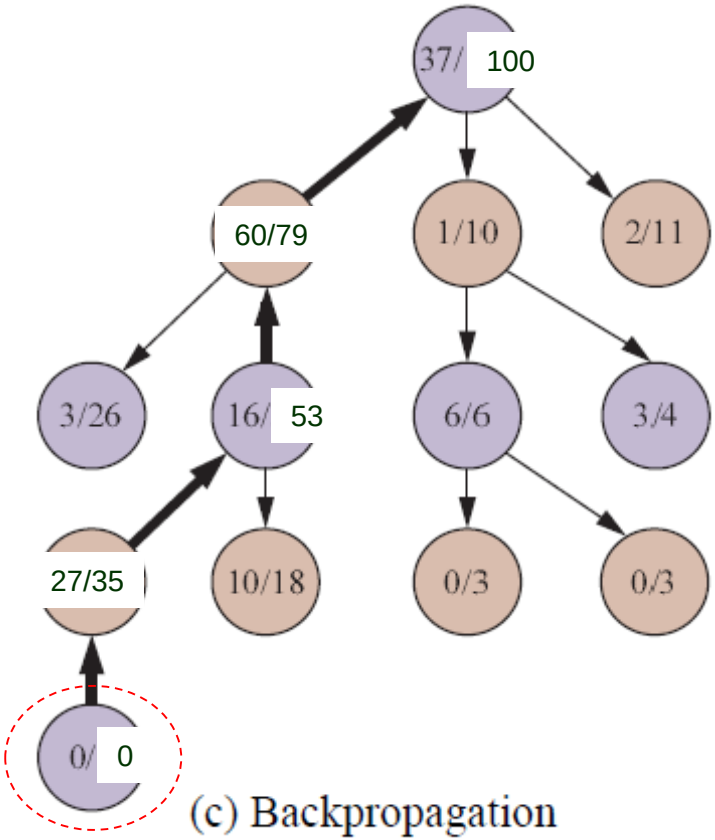
White

Black

White

Black

White



(c) Backpropagation

- 루트까지의 경로를 따라 모든 노드를 윗 방향으로 갱신

이 플레이아웃에서는 흑 승:

- 백 노드에서는 플레이아웃 횟수만 증가
- 흑 노드에서는 승리 횟수와 플레이아웃 횟수 증가

종료 조건

- MCTS는 선택, 확장, 시뮬레이션, 역전파 네 단계를 반복하여 진행
  - 일정한 횟수 N만큼 반복이 수행되었거나,
  - 지정된 시간이 만료된 경우
- 가장 많은 플레이아웃을 가진 액션을 반환

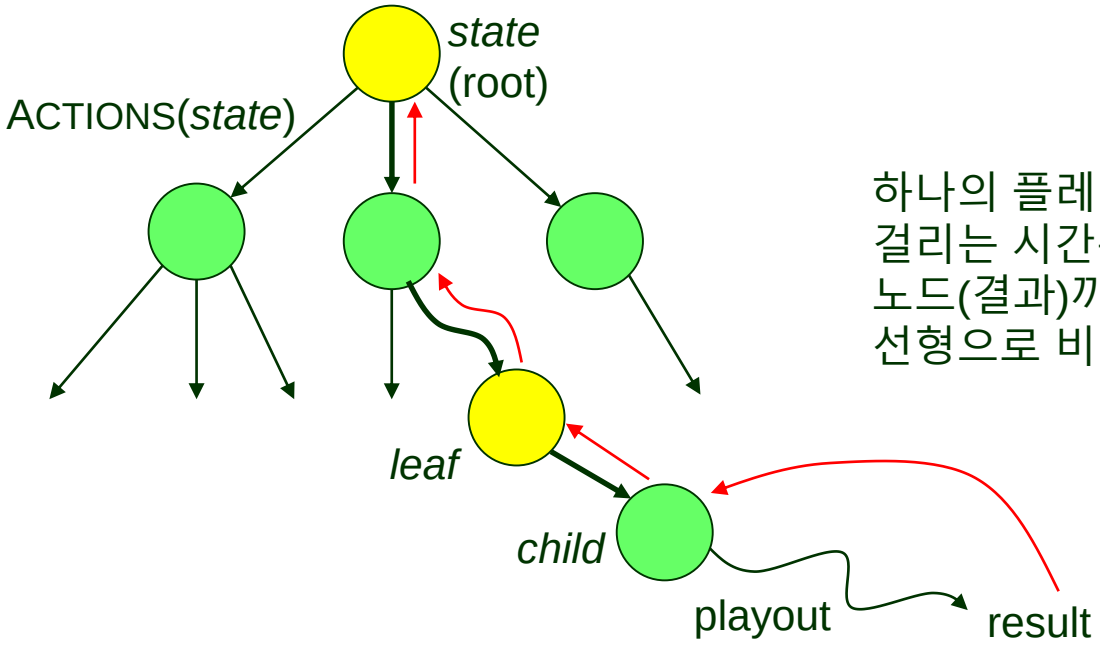
왜 가장 높은 비율의 액션이 아니고?

반복이 종료되면 플레이아웃 횟수가 가장 많은 움직임이 반환됨. 평균 유틸리티가 가장 높은 노드를 반환하는 것이 더 나을 것 같지만, 100회 중 65회 승리한 노드가 3회 중 2회 승리한 노드보다 더 나은 선택임. 왜냐하면 후자는 불확실성이 많기 때문

- 더 나은 움직임이 더 자주 선택되므로, 가장 유망한 움직임이 가장 많은 플레이아웃을 가지고 있을 것으로 예상
- 100번 중 65번 승리한 노드는 3번 중 2번 승리한 노드보다 더 나은 것임(불확실성 때문에)

몬테 칼로 트리 탐색 알고리즘

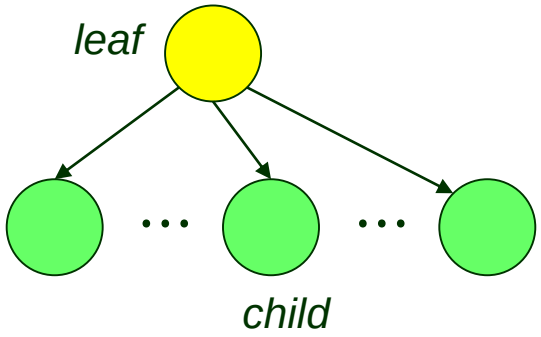
function MONTE-CARLO-TREE-SEARCH(*state*) **returns** *an action* // 현 상태에서 액션 선택
  tree ← NODE(*state*) // 현 상태를 루트로 놓고 트리 초기화
  while IS-TIME-REMAINING() **do** // *N* 회 반복, 반복할 때 마다 한 노드씩 확장
    leaf ← SELECT(*tree*) // 확장할 노드는 리프노드
    child ← EXPAND(*leaf*) // 리프노드의 자식노드 확장
    result ← SIMULATE(*child*) // 플레이아웃: 트리에 기록되지 않는 게임 플레이
    BACK-PROPAGATE(*result*, *child*) // 윗 방향으로 루트까지 모든 노드 갱신
  return the move in ACTIONS(*state*) whose node has highest number of playouts



하나의 플레이아웃을 계산하는 데 걸리는 시간은 자식부터 유틸리티 노드(결과)까지의 경로의 길이에 선형으로 비례함

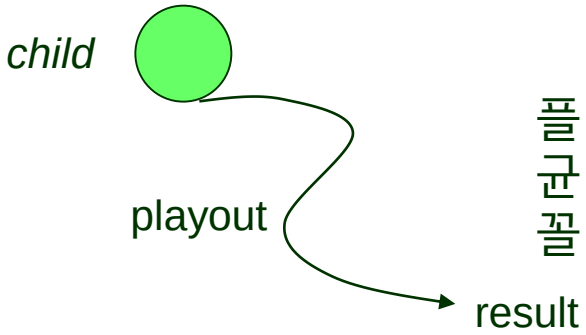
이슈

- *child* ← EXPAND(*leaf*)



여러 자식 노드를 생성하고 그 중 하나의 자식 노드만 선택하는 문제

- *result* ← SIMULATE(*child*)



플레이아웃은 게임이 결정될 때까지 균일하게 무작위로 액션을 선택하는 꼴이 될 수 있음

- IS-TIME-REMAINING()

진정한 몬테카를로 탐색은 N번의 시뮬레이션을 수행

가능한 액션의 순위 정하기

특정 상태를 루트로 둔 확장 MCTS 트리의 노드 n에서

UCB(Upper confidence bound) 공식:

해당 노드를 통해 플레이아웃을 거쳐 특정 위치(n)에서 액션을 취하는 플레이어의 승리 횟수

n의 부모를 통한 플레이아웃 수

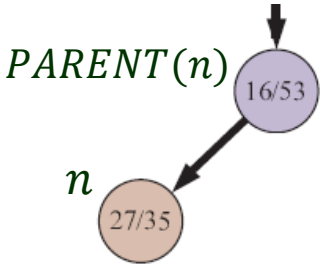
$$\mathrm{UCB}(n) = \frac{U(n)}{N(n)} + C \times \sqrt{\frac{\log N(PARENT(n))}{N(n)}}$$

이용 항:
n의 평균 효용성

해당 노드를 통해 수행된 플레이아웃 횟수

탐험과 이용간 균형 계수

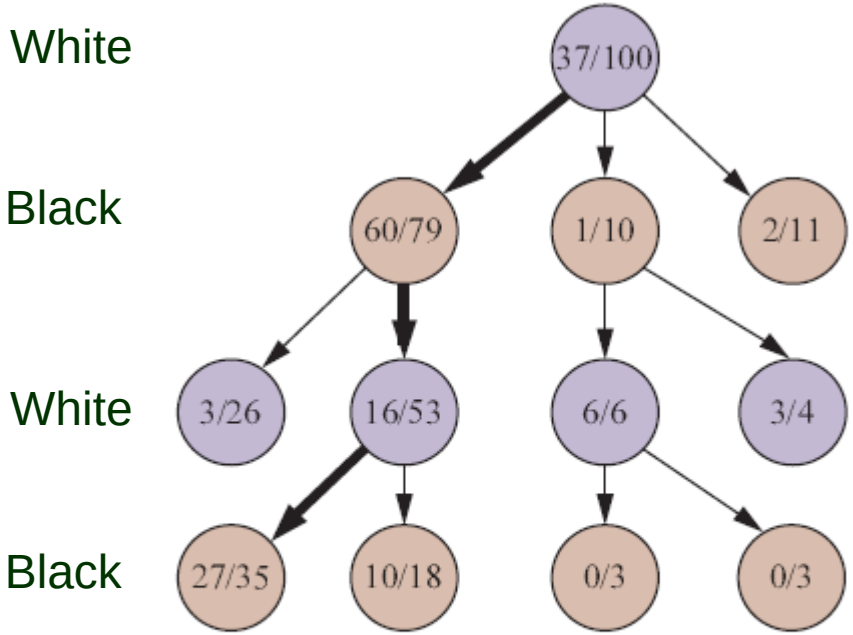
탐험 항:
n이 수차례 탐험되었을 때 그 중 높은 값



$$U(n) = 27$$
$$N(n) = 35$$
$$N(PARENT(n)) = 53$$
$$C = \sqrt{2} \quad \text{(이론적인 수치)}$$

$$\mathrm{UCB}(n) = \frac{27}{35} + \sqrt{2} \cdot \sqrt{\frac{\log 53}{35}}$$

상수 C



(a) Selection

- 탐험과 이용 간 균형 상수
- 여러 값을 시도해보고 가장 잘 수행되는 값을 선택
  - $C = 1.4$

    60/79 노드에서 가장 높은 점수
  - $C = 1.5$

    2/11 노드에서 가장 높은 점수