# 상태 공간과 맹목적 탐색

## 상태 공간

### 정의

문제의 해결 과정에서 도달할 수 있는 모든 집합들의 상태로, 문제의 해가 될 가능성이 있는 모든 상태들의 집합이다.

### 용어

* 상태 : 특정 시점에 문제의 세계가 처해 있는 모습
* 세계 : 문제에 포함된 대상들과 이를 수 있는 상황을 포괄적으로 지칭
* 목표 상태 : 문제에서 원하는 최종 형태
* 상태 공간 그래프 : 상태 공간에서 각 행동에 따른 상태 변화를 나타낸 그래프으면 백트래킹

## 맹목적 탐색

### 정의

정해진 순서에 따라 상태공간 그래프를 점차 생성해 가며 해를 탐색하는 방법

### 종류

* DFS(Depth first search)
  + 초기 노드에서 시작하여 깊이 방향으로 탐색
  + 목표 노드에 도달하면 종료
  + 더 이상 진행 할 수 없으면 백트래킹
  + 방문한 노드는 재방문 하지 않음
* BFS(Breadth first search)
  + 초기 노드에서 시작하여 모든 자식 노드로 확장하여 탐색
  + 목표 노드가 없으면 단말 노드에서 다시 자식노드 확장
* 반복적 깊이 심화 탐색(Iterative-deeping search)
  + 깊이 한계가 있는 깊이 우선 탐색을 반복 실행
  + 깊이 한계에 도달할 경우 다음으로 넘어감

### 장단점

* DFS(Depth first search)
  + 메모리 공간 사용 효율적
  + 최단 경로 해 탐색 보장 불가
* BFS(Breadth first search)
  + 최단 경로 해 탐색 보장
  + 메모리 공간 사용 비 효율
* 반복적 깊이 심화 탐색(Iterative-deeping search)
  + 최단 경로 해 보장
  + 메모리 공간 사용 효율
  + **실제 비용이 크게 늘지 않음**

# 정보 탐색과 게임 탐색

## 정보 탐색(휴리스틱 탐색)

시간이나 정보가 불충분하여 합리적인 판단을 할 수 없거나, 굳이 체계적이고 판단을 할 필요가 없는 상황에서 **신속하게 어림짐작을 하는 방법**

### 종류

* 언덕 오르기 방법
  + 지역 탐색, 휴리스틱 탐색
  + 현재 노드에서 휴리스틱에 의한 평가 값이 **가장 좋은 이웃 노드** 하나를 확장해 탐색하는 방식
  + 국소 최적화에 빠질 가능성이 있다.
* 최상 우선 탐색
  + 확장중인 노드들 중 목표 노드까지 **남은 거리가 가장 짧은 노드**를 확장하여 탐색
  + 남은 거리를 정확히 알 수 없으므로 휴리스틱 사용
* 빔 탐색
  + 휴리스틱 평가에 의한 평가값이 가장 우수한 **일정 개수의 확장 가능한 노드**만을 메모리에 관리하며 최상 우선 탐색을 적용
* A\* 알고리즘{\displaystyle {\hat {f}}(n)=g(n)+{\hat {h}}(n)}
* 추정한 전체비용 f(n)을 최소로 하는 노드를 확장해 가는 방법
* {\displaystyle f(n)}f(n): 노드 n을 경유하는 전체 비용
* {\displaystyle g(n)}g(n): 투입된 비용, {\displaystyle h(n)}h(n) : 남은 비용 일 때 f(n)은 그 둘의 합
* {\displaystyle h(n)}h(n) 값을 정확히 예측 할 수 없기 때문에 휴리스틱 함수를 사용하여 {\displaystyle {\hat {h}}(n)}h`(n)을 구한다.
* {\displaystyle {\hat {f}}(n)=g(n)+{\hat {h}}(n)}f`(n) = g(n) + h`(n)

## 게임 탐색

### 종류

* mini-max 알고리즘
  + MAX 노드 : 자신에게 해당하는 노드로 자기에게 유리한 최대 값 선택
  + MIN 노드 : 상대방에 해당하는 노드로 최소값 선택
  + 단말 노드로 부터 위로 올라가며 최소-최대 연산을 반복하며 자신이 선택할 수 있는 것 중 가장 좋은 값을 결정
* {\displaystyle \alpha -\beta } α- β가지치기(prunning)
  + 검토해 볼 필요가 없는 부분을 탐색하지 않도록 하는 기법
  + 깊이 우선 탐색으로 제한 깊이 까지 탐색을 하며 MAX 노드와 MIN 노드 값을 결정
  + {\displaystyle \alpha }α -자르기 : MIN 노드의 현재값이 부모 노드의 현재 값 보다 작거나 같으면 나머지 자식 노드 탐색 중지
  + {\displaystyle \beta } β-자르기 : MAX 노드의 현재값이 부모 노드의 현재 값 보다 같거나 크면, 나머지 자식 노드 탐색 중지
* 몬테 카를로 시뮬레이션
  + 형세 판단 함수 대신 특정 확률 분포로 부터 무작위 표본을 생성
  + 표본에 따라 행동하는 과정을 반복 후 결과 확인
  + 결과 확인 과정 반복 후 최종 결정