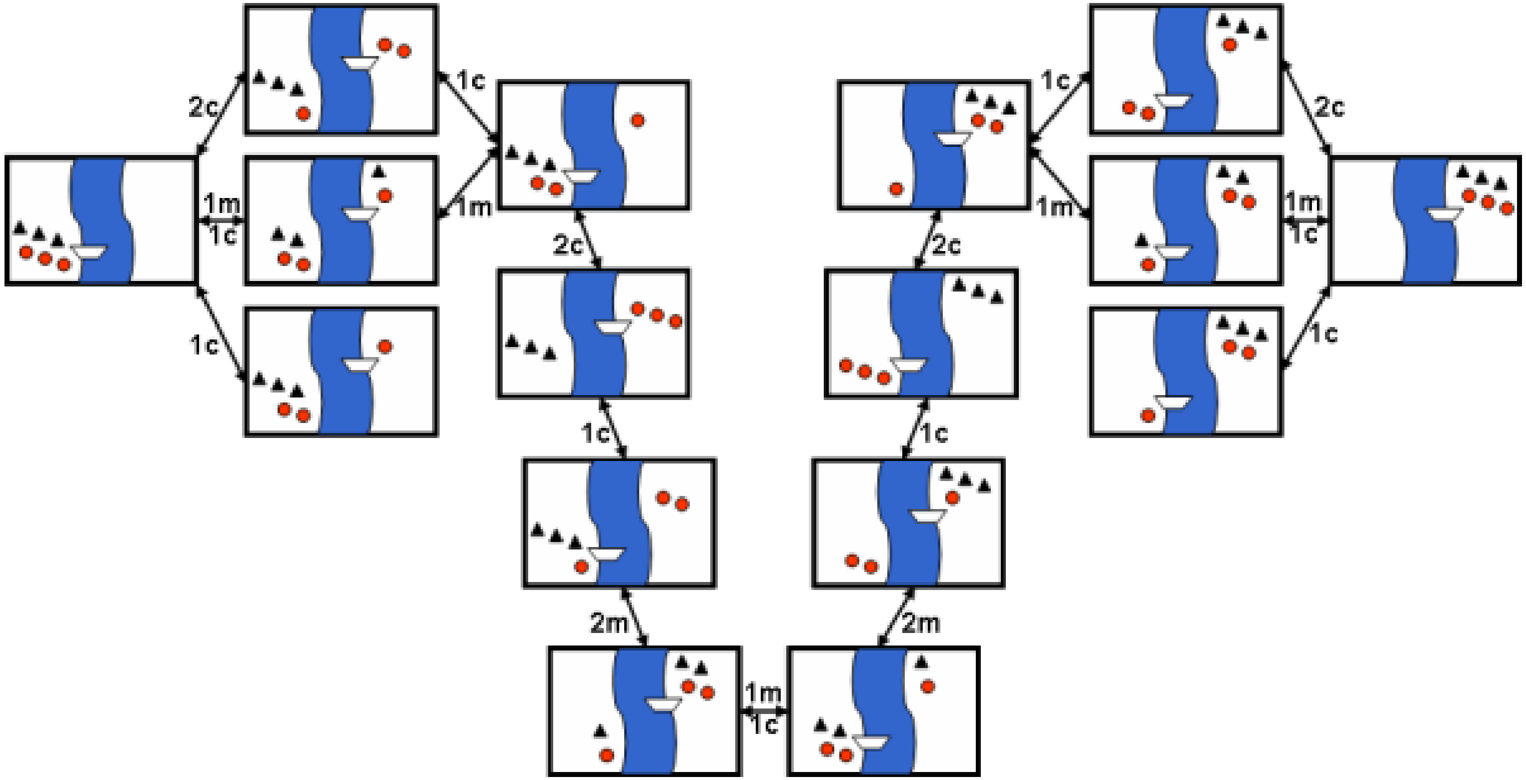
# 2장 문제해결 기법: 우리 주변의 문제를 잘 해결하기




출처: http://www.aiai.ed.ac.uk/~gwickler/missionaries.html

# 2장 문제해결 기법: 우리 주변의 문제를 잘 해결하기

- **학습 목표**
  - 체험해 봅시다: MC 문제 -> Ex6_MC 문제.xlsm

    ( https://github.com/jpub/LearnAIwithExcel )
  - 모델화
  - 상태 전이
  - 문제 해결의 구체적 예

# ⦿ MC 문제 체험

- **MC 문제**
  - http://www.novelgames.com/ko/missionaries/
  - http://kr.game-game.com/18394/

## ⊙ 체험해 봅시다: MC 문제

- MC 문제는 **'선교사(Missionary)와 식인종(Cannibal)이 같은 인원수이고, 2인승 배가 한 척 있을 때, 전원이 왼쪽 강변에서 오른쪽 강변으로 강을 건너려면 어떻게 하면 될까?'**라는 문제.
- 단, 선교사 수가 식인종 수보다 적으면 잡혀 먹히기 때문에 양쪽 강변에서는 선교사 수가 항상 같거나 더 많아야 함.
- 또 반대쪽으로 배를 옮기기 위해서는 반드시 한 사람 이상은 배에 타야 함.
- 왼쪽 강변에서도 오른쪽 강변에서도 선교사 수가 식인종보다 많아야 한다는 것은 얼핏 보기에는 무리일 것 같지만 상태를 순서에 따라가 보면 해결될 수 있다는 것을 이해할 수 있음.

# ⊙ 체험해 봅시다: MC 문제

- M(선교사)과 C(식인종)의 인원수 및 배의 정원을 변경하여 다양한 조합에서 상태 전이가 이루어지는 모습을 보면, 계산 문제와는 다르게 문제의 규모에 따라 복잡성이 증가하는 것은 아니라는 것을 알 수 있음.
- 상태 전이를 머릿속에서만 생각하면 혼란스럽지만 하나씩 상태를 조사해 나가면 해가 있는 경우에는 반드시 성공하므로 이 개념은 복잡해 보이는 문제에 유용.

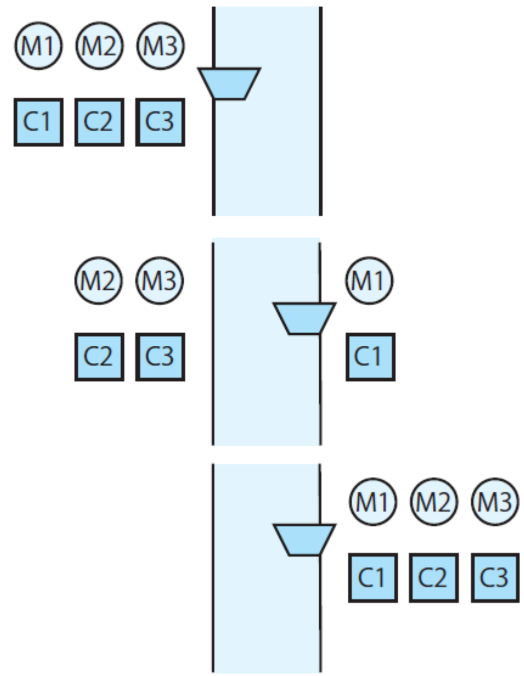



그림 5-1 MC 문제

## 2.1 모델화

- 문제 해결은 일반적으로 다음의 2단계로 실행.
  ① 문제의 모델화를 수행.
  ② 상태 전이 시뮬레이션을 수행.
- 모델화는 문제를 정리하여 컴퓨터로 처리하도록 하는 것임. 이를 위한 방법으로는 여러 가지가 있으나 문제 정리에는 **KJ 법**과 **마인드맵**이 자주 사용.
- 문제 정리는 해결에 필요한 요인들을 추출하고 그것들이 어떤 관계인지를 명확히 하는 것뿐만 아니라, 그것을 상태로 해서 정의하고 시간적 요인에 대하여 어떻게 변화하는지를 공식화하면 문제의 모델화가 가능.
- 그렇게 하면 컴퓨터상에서 파라미터를 변화시켜 시뮬레이션을 수행하고 문제 해결을 꾀할 수 있음.

## 2.1.1 모델화의 개념

- 일반적인 모델화 방법은 파라미터 조합의 최적화를 생각한다는 의미로 조합 최적화 문제라고 생각할 수도 있지만, 여기서는 동적인 문제를 처리하므로 다음과 같이 생각.
  - **상태는 시간 축 상에 직전 상태로부터 결정. 단, 반드시 하나로 결정되는 것은 아님. 이것이 상태 전이.**
  - **상태 전이는 일반적으로 여러 상태 후보 중에서 가장 좋다고 생각되는 상태를 선택.**
- 조합 최적화 문제에 국한되지 않고, 시간 축이 더해져 더욱 복잡해지고, 게다가 되돌리는 것이 불가능한 문제 해결에는 이러한 모델화와 시뮬레이션이라는 개념이 유효.
- 시뮬레이션은 상태 전이를 컴퓨터상에서 재현하는 것.

## 2.1.2 전략

- **상태 전이**가 유일하게 정해지지 않으면 어떠한 판단 기준이 필요하게 됨.
- 즉, 상태 전이의 선택 경로에 대하여 일정한 **평가 기준을 근거**로 가장 좋다고 생각되는 선택 경로를 취하게 되고 이것을 전략이라고 함.
- 전략이 명확하면 상태 전이에서 헤매지 않게 되지만 전략 자체에도 여러 방법이 있으므로 이 방법에 따라서는 상태 전이가 전혀 달라질 가능성도 있음.
- 이 전략 방법이 **탐색법**으로 이어짐.

## 2.2.2 탐색 트리

- 상태 전이를 효율적으로 수행하기 위해서는 전략, 즉 각 상태에서 주어지는 **연산자를 어떻게 적용할 것인지**에 대한 판단 기준이 필요.
- 이것은 여러 개의 가능한 상태 전이가 있을 경우에 어느 것을 선택하느냐의 문제로, **문제에 주어진 조건을 고려하면서 진행**할 필요가 있음.
- 상태에는 **금지 상태**도 있으므로 이것을 피해가며 진행함.
- 도중에 더 진행하지 못하게 되었을 때, 즉 가능한 상태 전이가 없어졌을 때에는 **직전의 상태로 돌아가 다른 상태 전이 후보를 조사**.

## 2.2.2 탐색 트리

- 이렇게 시간 축에 따라 전개되는 상태 전이 모습은 초기 상태를 루트로 하는 **트리 구조**로 표현할 수 있음.
- 이것을 **탐색 트리(Search Tree)**라고 함.
- 문제 해결은 탐색 트리를 루트부터 말단 노드까지 **가장 효율적으로 쫓아가는 것을 의미**.

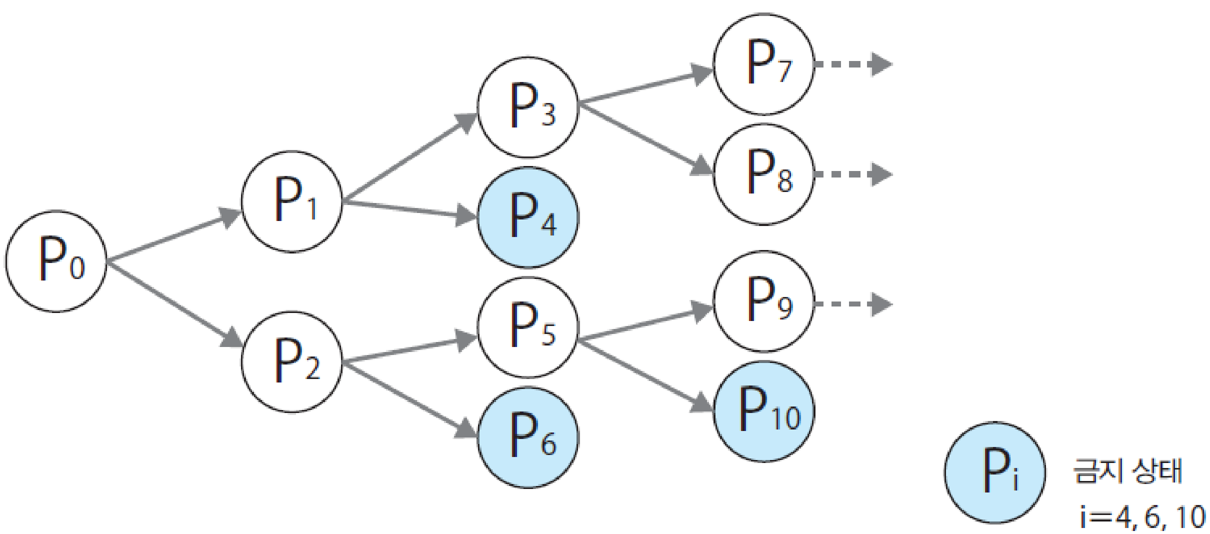
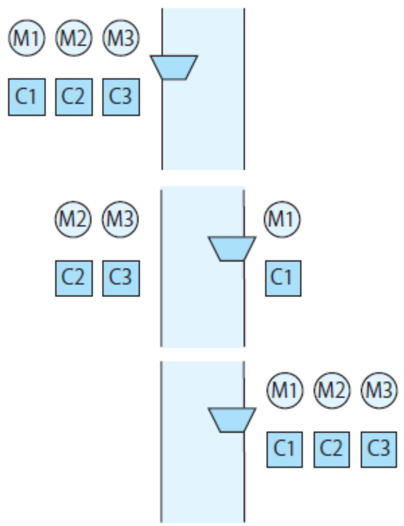



그림 5-3 탐색 트리

# 2.3 문제해결의 구체적 예

- **MC 문제**
  - 수식으로는 잘 표현하지 못하는 동적인 문제의 대표적 예로서 시뮬레이션으로 체험한 MC 문제를 자세히 살펴봄.
  - 이 문제는 재귀적으로 풀 수 없음.
  - 문제의 규모를 M, C 인원수는 3명, 배의 정원은 2명으로 하여 다음과 같이 모델화.




**MC 문제**

선교사와 식인종이 셋씩 있고, 2인승 배로 왼쪽 강변에서 오른쪽 강변으로 모두 건너고 싶지만, 선교사 수가 식인종 수보다 적으면 잡혀 먹히고 만다. 잡혀 먹히지 않고 전원이 건너려면?

**MC 문제의 모델화**

- 상태: (왼쪽 강변의 M, C 인원수), 배의 위치; (왼쪽일 때는 1, 오른쪽일 때는 −1)

  $P_i=(m_i, c_i, t_i)$ ; $0 \leqq m_i \leqq 3,\ 0 \leqq c_i \leqq 3,\ t_i = 1/-1,\ i \geq 0$

  $m_i \geq c_i$ (왼쪽의 인원수 관계)

  $3-m_i \geq 3-c_i$ (오른쪽의 인원수 관계)
- 초기 상태: $P_0=(3,3,1)$
- 목표 상태: $P_n=(0,0,-1)$
- 금지 상태: $Pj=(mj, cj, tj)$ ; $m_j<c_j$ ($m_i=0$제외) 또는 $3-m_i<3-c_i$ ($m_i=3$ 제외)
- 연산자: 배의 탑승 인원수 만큼만 m, c를 증감, 언제나 $t_{i+1}=-t_i$(왼쪽, 오른쪽 교대)

그림 5-4 **MC 문제 모델화**

# 2.3.1 MC 문제

- 상태를 **{왼쪽 강변의 M, C 인원수, 배의 위치(1/-1)}**로 나타내고 수식화하면 다음과 같음.

$P_i = (m_i, c_i, t_i)$

$0 \leqq m_i \leqq 3,\ 0 \leqq c_i \leqq 3,\ t_i = 1$(왼쪽) 또는 $-1$(오른쪽)

$m_i \geqq c_i$ (왼쪽 강변에서의 사람 수 제약)

$3 - m_i \geqq 3 - c_i$ (오른쪽 강변에서의 사람 수 제약)

- 초기 상태 $P_0$ = (3, 3, 1), 목표 상태 $P_n$ = (0, 0, -1)
- 연산자는 배의 탑승 인원수 **{M의 인원수, C의 인원수}**만으로 상태(왼쪽)의 인원수가 증감.

## 2.3.1 MC 문제

- 앞의 내용을 수식화하면 다음과 같음.

**$\alpha$ : M이 한 명만 이동** $m_{i+1}=m_i-t_i,\ t_{i+1}=-t_i$

$t_i=1$일 때는 $1\leqq m_i\leqq 3$, $t_i=-1$일 때는 $0\leqq m_i\leqq 2$

**$\beta$ : C가 한 명만 이동** $c_{i+1}=c_i-t_i,\ t_{i+1}=-t_i$

$t_i=1$일 때는 $1\leqq c_i\leqq 3$, $t_i=-1$일 때는 $0\leqq c_i\leqq 2$

**$\gamma$ : M이 두 명 이동** $m_{i+1}=m_i-2t_i,\ t_{i+1}=-t_i$

$t_i=1$일 때는 $2\leqq m_i\leqq 3$, $t_i=-1$일 때는 $0\leqq m_i\leqq 1$

**$\delta$ : C가 두 명 이동** $c_{i+1}=c_i-2t_i,\ t_{i+1}=-t_i$

$t_i=1$일 때는 $2\leqq c_i\leqq 3$, $t_i=-1$일 때는 $0\leqq c_i\leqq 1$

**$\psi$ : M, C가 한 명씩 이동** $m_{i+1}=m_i-t_i,\ c_{i+1}=c_i-t_i,\ t_{i+1}=-t_i$

$t_i=1$일 때는 $1\leqq m_i\leqq 3$ & $1\leqq c_i\leqq 3$

$t_i=-1$일 때는 $0\leqq m_i\leqq 2$ & $0\leqq c_i\leqq 2$

## 2.3.1 MC 문제

- 번거로운 정의지만 요약하면, 한 번 배를 이동하면 **{M}, {C}, {MM}, {CC}, {MC} 중의 어느 한 가지 패턴**으로 인원수가 증감.
- 단, 배가 있는 강변의 인원수 이하만 탈 수 있고, 최소 한 사람은 타야 함. 따라서 왼쪽 강변의 인원수에 주목하면 앞 페이지의 5가지 연산자가 가능.
- 상태 정의는 얼핏 보기에 왼쪽 강변에서 m≧c이면 오른쪽 강변에서 반대로 3-m≦3-c가 될 것 같으므로 양쪽 강변에서 선교사 쪽이 많다는 것이 불가사의로 여겨지겠지만, 다음과 같은 금지 상태를 생각하면 납득할 수 있음.
  - **금지 상태 $P_j = (m_j, c_j, t_j)$**
  - **$m_j < c_j$ ( 단, $m_j = 0$ 제외 ) 또는 $3-m_j < 3-c_j$ ( 단, $m_j = 3$ 제외 )**

## 2.3.2 MC 문제의 탐색 트리

- 이 예에 대한 탐색 트리는 금지 상태를 제외하고 이전으로 돌아가는 중복 상태도 제거하면, 적용 가능한 연산자가 거의 한 가지로 결정되므로 가지가 넓게 퍼지지 않고 쉽게 구축할 수 있음.
- 그림은 트리 형태가 되진 않았지만 왼쪽 끝을 루트로 해서 왼쪽에서 오른쪽 방향으로 가지가 펼쳐지는 것으로 생각하기 바람.
- 이 경우에는 금지 상태를 제외하면 항상 한 가지이므로 이 상태에 관한 트리를 전개함.

| i=0 L | 1 R | 2 L | 3 R | 4 L | 5 R | 6 L | 7 R | 8 L | 9 R | 10 L | 11 R |
|---|---|---|---|---|---|---|---|---|---|---|---|
| (3,3,1) | (3,2,−1) | (3,3,1) | (3,1,−1) | (3,2,1) | (3,0,−1) | (3,1,1) | (2,1,−1) | (2,2,1) | (0,2,−1) | (2,1,1) | (0,1,−1) |
| | (3,1,−1) | (3,2,1) | (3,0,−1) | (3,1,1) | (2,1,−1) | (2,2,1) | (2,0,−1) | (1,3,1) | (0,1,−1) | (1,1,1) | (0,0,−1) |
| | (2,3,−1) | (2,3,1) | (2,2,−1) | | (2,0,−1) | (2,1,1) | (1,2,−1) | (1,2,1) | | (0,3,1) | |
| | (2,2,−1) | | (2,1,−1) | | (1,1,−1) | (1,3,1) | (1,1,−1) | (0,3,1) | | (0,2,1) | |
| | (1,3,−1) | | (1,2,−1) | | | (1,2,1) | (0,2,−1) | | | (12,1) | |

※ 배는 교대로 왼쪽(L)이나 오른쪽(R)에 있다. 가 상태 전이, 는 금지 상태, 는 중복(이전으로 돌아감)

**그림 5-5** MC 문제의 탐색 트리

# 2.3.2 MC 문제의 탐색 트리

- 앞 표에서의 상태 전이에 따른 이동 모습은 다음 그림과 같음.

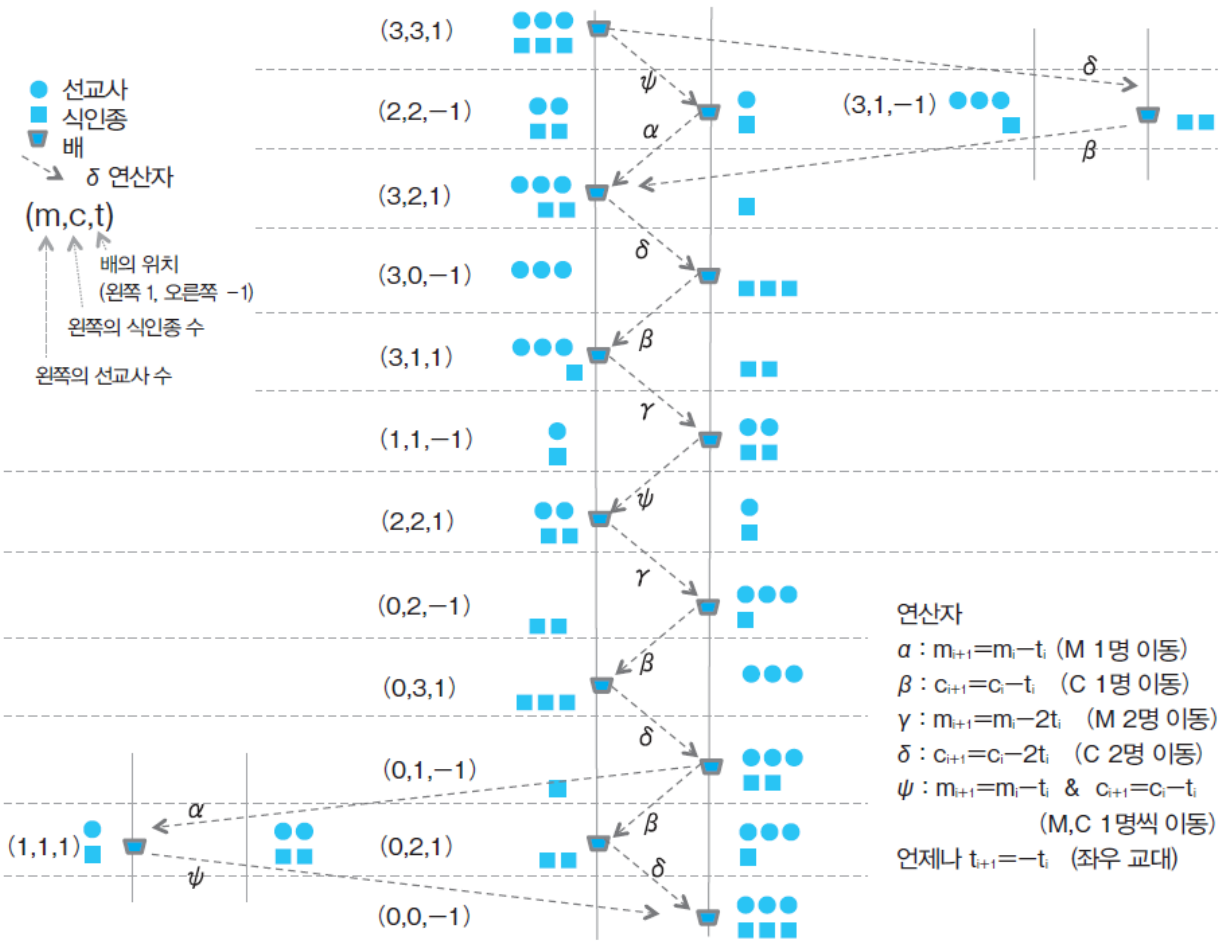



그림 5-6 MC 문제의 해답

# 2.3.2 유사 문제 체험

- **다리 건너기 문제**
  - http://www.novelgames.com/ko/bridge/
  - http://melodybox.tistory.com/250