

---

# 그룹핑에 기반한 바둑 프로그램에서의 사석검출 알고리즘

김동준\* · 김윤희\*

\*안동대학교

## A Detection Algorithm of Dead Stone for the Go program based on the Grouping

Dong-june Kim\* · Yun-ho Kim\*

\*Andong National University

E-mail : timecx@nate.com, unokim@andong.ac.kr

### 요 약

본 논문에서는 바둑 프로그램에서 사석을 제거하기 위해 그룹핑에 기반하여 사석을 검출하는 알고리즘을 제안하고자 한다. 그룹은 같은 색의 돌이 상하좌우로 연결되는 경우에 같은 그룹으로 그룹핑 되도록 정의한다. 사석제거 알고리즘은 이와 같은 그룹을 기반으로 하여 사석을 제거하는 방법으로서 같은 그룹의 모든 돌이 같은 색의 돌과 상관없이 상하좌우가 막혀 있는 Case에 만족 할 경우에 해당되는 그룹을 검출하여 사석으로 처리한다.

### ABSTRACT

In this paper, A Detection Algorithm of Dead Stone for the Go program based on the Grouping is proposed. The group of the same color as the stone is connected to the left, right, up and down side in the same group is defined to be grouping. A Detection Algorithm of Dead Stone based on these groups to remove stone and all the stones of the same color as the stones of the same group, regardless of the Case for TILT blocked, if satisfied that the detection of the group is treated as a dead stone.

### 키워드

그룹화, 사석제거 알고리즘

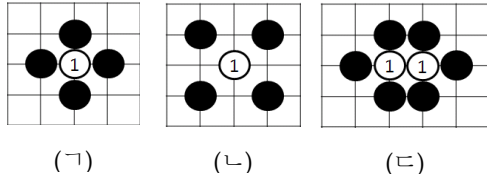
## I. 서 론

현대로 바둑이 이어지면서 바쁜 일상 속에 기원을 찾거나 바둑을 둘 대상을 직접 만나지 않고 바둑을 둘 수 있는 온라인 바둑이나 바둑 소프트웨어가 각광을 받게 되었다. 그에 따라서 은별바둑, 천하수담, 골리앗 등의 다양한 종류의 바둑 소프트웨어들이 만들어 졌고 바둑의 룰을 적용하기 위한 다양한 알고리즘이 고안되었다. 이 논문에서는 바둑 소프트웨어에서 가장 중요하고 핵심적인 기능인 사석 검출 알고리즘을 그룹핑에 기반하여 제안 하고자 한다.

## II. 그룹핑과 사석검출

바둑의 기본룰 중에는 들어냄의 규칙(사석제거)이 존재한다. 바둑에서 가장 기본규칙 중의 하나로 그림 1-(ㄱ)와 같이 백 돌의 활로(상하좌우)를 흑 돌이 완전히 막으며 포위했을 때 백 돌을 반상에서 들어내 백 돌을 들어낸 자리를 흑 돌의 영역으로 만드는 것이다.[2] (그림 1-(ㄴ)와 같이 완전히 봉쇄하지 못했을 때는 백 돌을 따내지 못한다.)

(ㄷ)와 같이 여러 개의 백 돌이 같이 있는 경우 역시 모든 흑 돌에 의해 활로가 막히면 사석으로 여기고 반상에서 들어낸다.



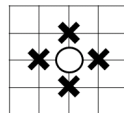
[그림 1] 사석 여부에 대한 예

그림 1의 (가)와 같이 하나의 돌을 사석으로 검사하고 제거하는 알고리즘을 처리하기 위해서는 하나의 돌에 대해 활로의 막힘 여부를 검사하여 제거하면 되지만 (다)와 같이 두 개 이상의 돌이 연결된 경우에 대해 사석여부를 검사하기 위해서는 둘 이상의 돌에 대한 사석여부를 검사하는 알고리즘을 필요로 한다.

이 논문에서는 두 개 이상의 돌이 연결된 경우에 대한 사석제거 알고리즘을 구현하기 위해서 각각의 돌을 그룹화 하는 것을 제안하고 그 그룹화에 기반하여 사석을 제거하는 방법에 대해 제안한다.

2.1 그룹화

돌들의 연결이 상하좌우로 이루어져 있으며 돌의 색이 같은 경우에 그 돌들을 한 그룹에 속해 있다고 여기고 그룹핑 해준다. 예로, 그림 2에서 X로 표시된 부분 중에 가운데 놓인 돌과 같은 색인 흰 돌이 놓여 진다면 새로 놓여진 돌은 기존에 가운데 놓여있는 흰 돌과 같은 그룹번호를 가지게 된다.



[그림 2] 그룹화 허용 방향

놓여지는 모든 돌은 그룹 리스트로 관리가 되며 놓여진 하나의 돌에 대해서도 모두 그룹 번호가 매겨지고 각 색의 돌마다 각각의 그룹 리스트를 유지한다.

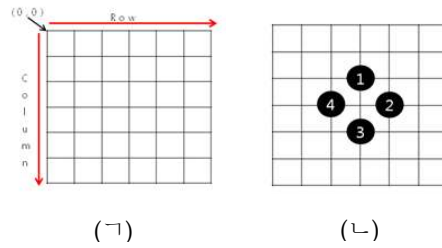


[그림 3] 각 돌별 그룹 리스트

서로 다른 두 개 이상의 그룹이 연결 되어야 할 경우, 그룹번호가 정해지는 우선순위는 Row, Column 중 Row가 우선되며, Row Column의 값 중 좌표점 0에 가까운 곳에 위치한 돌의 그룹번호로 통합 되어지는 것을 규칙으로 정의한다.

( Row, Column의 좌표 기준 정리 : 그림 4-(가) ) 그림 4-(나)로 예를 들면 4개의 돌 가운데 돌을 놓을 경우 Row의 값이 가장 낮은 4번 그룹으로

모든 돌의 그룹이 통합되게 된다.



[그림 4] 그룹번호 지정의 우선순위

2.1.1 그룹이 연결되는 위치별 형태

Case 1. 중앙 부분



Case 2. 귀 부분



Case 3. 변 부분



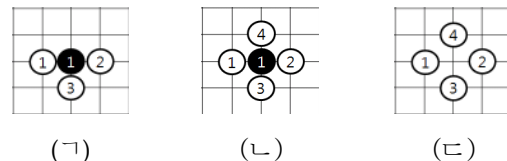
그룹화는 위의 3 개의 Case를 포함해서 총 9가지 대표적인 Case가 있다.

각 "(0,0), (0,18), (18,0), (18,18)" 각 네 귀를 포함하여 각 네 변과 상하좌우를 모두 확인해야하는 중앙 부분 이렇게 총 9가지이다.

2.2 사석제거 알고리즘

사석을 판별하기 위하여 그룹 알고리즘을 통해 그룹핑된 돌들은 그룹 단위로 돌의 사석 여부를 검사 받게 되며 (이 때, 사석제거의 우선순위를 가리기 위해, 방금 놓은 돌의 반대되는 색을 가진 돌의 그룹들만 검사를 한다.) 사석은 그룹단위로 제거가 되고 사석으로 판명된 그룹의 돌들은 모두 반상에서 들어내게 된다.

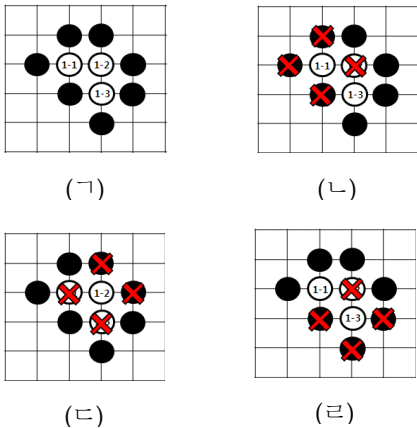
우선 하나의 돌을 제거하는 검사 방법으로 그림 5-(가)의 흑 돌은 상하좌우가 막혀있지 않지만 (나)과 같이 백 돌이 흑 돌의 상하좌우를 막게 되면 (다)과 같이 흑 돌을 죽은 돌이라고 보고 흑 돌의 1번 그룹을 반상에서 들어내게 된다.



[그림 5] 돌 하나에 대한 사석 검출 조건

둘 이상의 돌을 제거 할 경우의 검사방법은 하나의 돌을 제거하는 규칙에서 착안하여 둘 이상의 돌로 이루어져 있어도 돌이 죽는 경우를 만족하려면 상하좌우가 막혀 있어야 한다는 조건에서 처리된다.

그림 6-(㉠)의 백 돌은 1-1, 1-2, 1-3 세 개의 돌로 이루어진 1개의 그룹을 가지고 현재 모든 면이 막혀 활로가 없기 때문에 죽은 돌이며 (㉡), (㉢), (㉣)을 순서대로 살펴보면 같은 색의 돌과 상관없이 백 돌은 상하좌우가 막혀있는 특징이 있다.



[그림 6] 두 개 이상의 돌에 대한 사석 검출 조건

그림 6에서 본 것과 같이 같은 그룹에 속한 각각의 돌에 대해 상하좌우가 막혔는지에 대한 여부를 확인하고 해당 그룹의 모든 돌의 상하좌우가 모두 막혀있다면 해당 그룹의 돌들을 사석이라고 볼 수 있는 근거가 된다. 그러므로 모든 활로가 막혀있는 패턴의 경우를 구해서 목록을 정리하여 사석 검사 시 검사의 조건으로 사용한다. 상하좌우가 막혀있는 패턴 외에 네 귀 "(0,0), (0,18), (18,0), (18, 18)" 부분에 돌이 놓여졌을 경우 각각의 예로 방금 놓은 돌을 기준으로 "(0,0) - 우, 하", "(18,18) - 좌, 상"등과 같이 상하좌우 중 두 방향만을 확인하는 특징이 있으므로 상하좌우를 확인 하는 것 외에 예외 패턴을 구해야 하며 네 변의 경우는 "Row가 0 이거나 18, Column이 0 이거나 18"인 경우로 상하좌우 중 하나가 해당의 비교 대상에서 제외되어 세 방향만을 확인하기 때문에 이 역시 예외로 패턴을 구해야 한다.

이와 같은 패턴의 경우의 수는 다음과 같이 구한다.

2.2.1 돌의 활로가 막히는 위치별 패턴

Case 1. 귀 부분

1. 같은 색의 돌이 하나일 경우  
 $2C_1 = 2$
2. 같은 색의 돌이 두 개일 경우 : 2  
 (자리 두 개 중에 돌 두 개를 흑 돌 백돌 각각 한번씩 놓아야 하기 때문에 2이다.)

3. 하나의 귀에 대한 패턴 총 합계 : 4  
 네 개의 귀가 존재하기 때문에 모든 귀에 대한 패턴의 경우의 수를 계산하면 "4 X 4 = 16", 귀 부분에 총 16가지 패턴이 존재함을 알 수 있다.

Case 2. 변 부분

1. 같은 색의 돌이 하나일 경우  
 $3C_1 = 3$
2. 같은 색 돌이 두 개일 경우  
 $3C_2 = 3$
3. 같은 색 돌이 세 개일 경우 : 2  
 ( 자리 세 개 중에 돌 세 개를 흑 돌 백 돌 각각 한번씩 놓아야 하기 때문에 2이다)
4. 하나의 변에 대한 패턴 총 합계 : 8

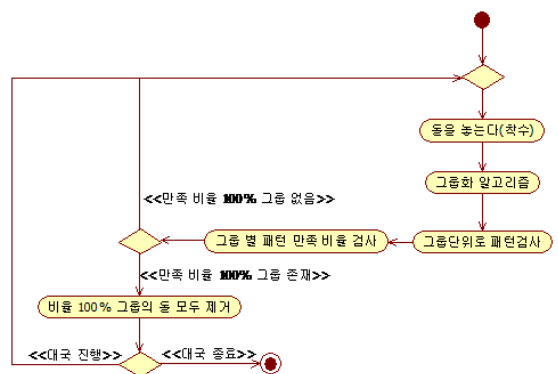
네 개의 변이 존재하기 때문에 모든 변에 대한 패턴의 경우의 수를 계산하면 "8 X 4 = 32" 변 부분에 총 32가지 패턴의 존재함을 알 수 있다.

Case 3. 중앙 부분

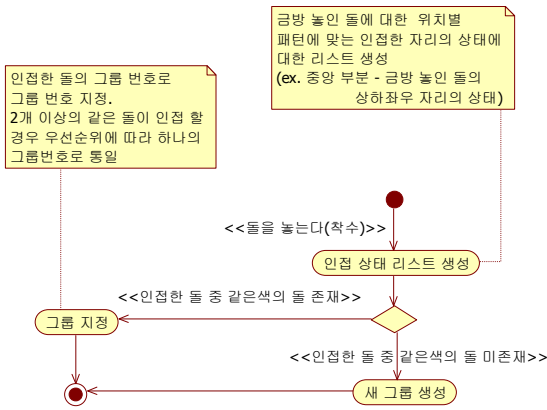
1. 같은 색 돌이 하나일 경우  
 $4C_1 = 4$
2. 같은 색 돌이 두 개일 경우  
 $4C_2 = 6$
3. 같은 색 돌이 세 개 일 경우  
 $4C_3 = 4$
4. 같은 색 돌이 네 개일 경우 : 2  
 (자리 네 개 중에 돌 네 개를 흑 돌 백 돌 각각 한번씩 놓아야 하기 때문에 2이다)
5. 네 방향에 대한 패턴 총 합계 : 16

● 사석 검출 및 제거 알고리즘

패턴검사와 사석 검출 및 제거 되는 전체 알고리즘과 세부 알고리즘의 플로우 차트는 다음과 같다.

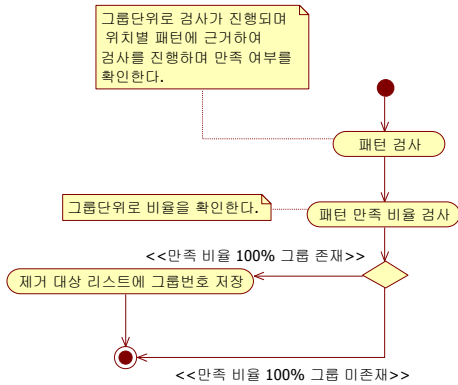


1. 그룹화는 다음과 같다.



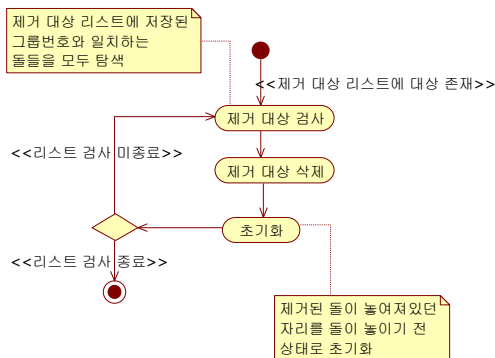
금방 놓은 돌에 위치별 Case에 맞는 패턴으로 인접 상태를 확인하여 인접 상태 리스트를 생성하여 그룹을 지정하게 된다.

2. 그룹단위 패턴 검사는 다음과 같다.



방금 놓은 돌의 반대되는 색 돌의 그룹들을 위치별 Case의 패턴에 맞는지 그룹 단위로 검사하게 되며 그룹단위 패턴 만족 비율을 확인하여 100%만족 그룹이 있을 경우 제거 대상 리스트에 해당 그룹 번호를 저장한다.

3. 사석 제거는 다음과 같다.



제거 대상 리스트에 저장된 그룹번호와 방금 놓인 돌의 반대되는 색을 만족하는 모든 돌들을 제거한다.

III. 결 론

이 논문에서는 돌들을 그룹으로 분류하여 그룹 기반으로 관리하는 것을 제안하였다. 그룹 생성 방법에 대한 정의를 알고리즘으로 정리하고 그것을 기반으로 그룹을 생성하게 되며 관리한다. 그룹 알고리즘은 같은 색의 돌이 상하좌우로 연결되는 경우에 대해서만 같은 그룹으로 구분하며 두 개 이상의 그룹이 합쳐지는 경우에 대한 우선순위를 정의하였다. 사석 제거 알고리즘은 그룹으로 이루어진 돌들의 활로 존재 여부를 확인하는 세 개의 Case를 통해 도출해낸 64가지 패턴을 기준으로 그룹 단위로 검사하여 해당 그룹에 속한 모든 돌이 패턴의 경우의 수를 만족하는지에 대한 여부를 확인, 모두 만족하는 상황에는 그 그룹에 속한 돌들은 사석으로 돌 수 있으며 반상에서 들어내는 조건이 된다.

참고문헌

[1] 한국기원 홈페이지 - [http://www.baduk.or.kr/baduk/baduk\\_history.asp](http://www.baduk.or.kr/baduk/baduk_history.asp)  
 [2] 한국기원 홈페이지 - [http://www.baduk.or.kr/baduk/baduk\\_rule.asp](http://www.baduk.or.kr/baduk/baduk_rule.asp)  
 [3] Robert Eckstein, Marc Loy, and Dave Wood, *Java Swing*(Ver 1.2), O' REILLY, 1998.  
 [4] Cay Horstmann, *Core Java™, Volumn I - Fundamentals*, 8th ed., PrenticeHall, 2007.