

Bitboards and Eliminates

Wees

August 2025

1 Introduction

An eliminate is a problem that consists of the same Start and Finish as another problem, with an arbitrary number of Middle and Foot holds removed, and no extra holds added. This article will focus on techniques to efficiently detect eliminates in a problem set.

2 Problems as Bitboards

Besides None, there are four roles a hold can occupy: Foot, Start, Middle, or Finish. Hence, for a board with N holds, we could represent each problem as an N digit long base-5 number, where each hold corresponds to a unique place value, and each role corresponds to a unique digit. For instance, a problem with a Start at hold zero, a Foot at hold one, and None at hold two could be assigned the number

problem : 012

However, most computers are built to handle base-2 (binary) numbers. Hence, rather than represent each problem as a single base-5 number, we will store each problem as four binary numbers, one for each role, which we will call a bitboard. For instance, the problem from before would be represented as

feet : 010
start : 001
middle : 000
finish : 000

3 Finding Eliminates

Now that we have our problems represented as bitboards, we will define an algorithm to determine whether one set of bitboards represents an eliminate of another set of bitboards.

Given two problems, Problem A and Problem B, we first check if they have the same start and finish. To do so, we assert that the Start number for Problem A equals the Start number for Problem B. We repeat the same process for the Finish numbers. If either of the two pairs of numbers are not equal, then we conclude neither problem is an eliminate of the other and exit our test.

Next, we find the Foot and Middle holds that Problem A and Problem B have in common. To do so, we compute a bitwise AND ('&' in most programming languages) between the Middle and Foot numbers of Problem A and those of Problem B, respectively. This leaves us with two numbers, one for Middle and one for Finish, that represent where Problem A and Problem B have holds in common. For instance, if Problem A has the Foot number 011 and Problem B has the Foot number 101, then the operation

$$011 \& 101 = 001$$

tells us that the problems have a Foot in common at hold zero.

We then check if Problem A is an eliminate of Problem B. If Problem A is an eliminate of Problem B, then Problem A only consists of holds that the two problems have in common. Hence, we assert that the Middle (Mid) and Foot numbers of Problem A are equal to those that both problems share in common, respectively. If both of these pairs of numbers are equal, then Problem A is an eliminate of Problem B. If either of the two pairs are unequal, Problem A is not an eliminate of Problem B. In most programming languages, the assertion of equality is represented by '=='. Hence, the above test could be written as

```
if (Foot_A & Foot_B == Foot_A and Mid_A & Mid_B == Mid_A)
{
    # A is an eliminate of B
}
```

We repeat the same process for Problem B. Only this time, an assertion of true indicates that Problem B is an eliminate of Problem A.

Note, if Problem A is an eliminate of Problem B, and Problem B is an eliminate of Problem A, then Problem A is identical to Problem B.

4 Conclusion

Bitboards supply a simple algorithm to detect eliminates in a problem set by leveraging bitwise AND operations on computers.