Practice Activity: Counting Game States

One of the recurring themes in this class is the mathematical and algorithmic analysis of strategy games. One common counting problem in this domain is determining the number of positions associated with a particular game. This analysis helps guide our understanding of how "hard" a particular game is. For example, Tic-Tac-Toe has a relatively small number of board positions and is generally viewed as an easy game to play and solve. On the other, chess has many, many more possible positions and is viewed as a much harder game to play and solve. In this practice activity, we will cover the mathematics necessary to estimate the number of board positions in the games.

Counting the numbers of possible sequences with a specified number of repetitions

In the "Enumerations" lecture, we counted the number of possible sequences of length n where each item in the sequence was chosen from a set of m possible outcomes. As noted in the lecture, the total possible number of sequences was m^n. In this activity, we'll consider a version of this problem in which only sequences with a specified number of instances of each outcome are allowed. If there are m outcomes, we are only consider sequences where there are k_1 instances of the first outcome, k_2 instance of the second outcome, and so on. Observe that that the sum of k_i should be exactly n, the total length of the sequence.

As a concrete example, let's count the number of sequences of length 4 with outcomes in the set $\{1, 2\}$ where the sequence contains $k_1 = 2$ instances of 1 and $k_2 = 2$ instances of 2. We can list all the six sequences that meet this criteria: $\{(1, 1, 2, 2), (1, 2, 1, 2), (1, 2, 2, 1), (2, 1, 1, 2), (2, 1, 2, 1), (2, 2, 1, 1)\}$. Remarkably, there is a formula involving factorials for computing the number of possible sequences with a specified number of repetitions of each element. This formula (based on multinomial coefficients) has the form:

$$\frac{n!}{k_1!k_2!...k_m!}$$

According to this formula, the number of sequences for our example is $\frac{4!}{2!\times 2!} = \frac{24}{2\times 2} = 6$. As another example, let's count the number of sequences with outcomes in the set $\{1, 2, 3\}$ with the restriction that the sequences contain 2 instances of 1, 1 instance of 2, and 1 instance of 3. Applying the formula, $k_1 = 2$, $k_2 = 1$, and $k_3 = 1$. Therefore, the number of sequences is $\frac{4!}{2!\times 1!\times 1!} = \frac{24}{2\times 1\times 1} = 12$. These twelve outcomes are:

$$\{(1,1,2,3),(1,2,1,3),(1,2,3,1),(2,1,1,3),(2,1,3,1),(2,3,1,1),. \\ (1,1,3,2),(1,3,1,2),(1,3,2,1),(3,1,1,2),(3,1,2,1),(3,2,1,1)\}$$

Tic-Tic-Toe

With this formula, we can now begin our analysis of Tic-Tac-Toe. A Tic-Tac-Toe board contains nine squares (3×3 grid) that can contain either an X, an O, or can be empty (denoted by E). So, our first estimate of the number of possible board positions in Tic-Tac-Toe is to simply enumerate all possible choices from the set $\{X, O, E\}$ for each of the nine squares on the board. In this case, we can enumerate $3^9 = 19683$ possible boards.

However, this estimate is much too high in practice. Note that the number of X's and O's can differ by at most one for any valid Tic-Tac-Toe board. In particular, it's impossible to have a board that consists of nine X's. So, let's try our hand at performing a more accurate analysis of the number of possible positions.

If X always go first, then any possible board will have either an equal number of X's and O's or one more X than O. The number of boards corresponding to each of these possibilities can be computed using the formula above. For example, the number of boards with four X's, three O's and two E's is exactly:

$\frac{9!}{(4!)(3!)(2!)} = \frac{362880}{24 \times 6 \times 2} = 1260$ boards.

This program computes that there at total 5290 boards where the number of X's is either equal to or one more than the number of O's.

This analysis is still not particularly precise since it counts boards that are clearly not possible in a legal game of Tic-Tac-Toe. (For example, a game with three X's in a row and three O's in a row.) However, at this point, further analysis is not really that important in understanding how "hard" Tic-Tac-Toe is as a game. Modern computers can easily analyze all of the legal positions in a game of Tic-Tac-Toe and plan out an optimal strategy. In a future week, we will learn an algorithm that accomplishes exactly this task.

Chess

As a second example, we will compute a simple estimate on the number of positions in a game of chess. This estimate will only consider those positions in which all of the original pieces remain on the board and will include some illegal positions such as those where pawns are on the first rank. However, in reality, this number is actually much lower than the true number of positions since it does not include positions where pieces have been captured or where pawns have been promoted.

To compute this estimate, we will use the multinomial formula from above. A chess board is 8×8 grid with 64 spaces. Both White and Black have 8 pawns, 2 rooks, 2 knights, 2 bishops, 1 queen and 1 king, respectively. With these 32 pieces on the board, there are 32 empty spaces remaining. Using the multinomial formula from above, the number of positions containing all of these pieces is

$$\frac{64!}{(32!)\times(8!)^2\times(2!)^2\times(2!)^2\times(2!)^2\times(1!)^2\times(1!)^2}.$$

Note that exponent 2 in the various factorials in the denominator correspond to the fact both White and Black have that number of pieces. Evaluating this formula in Python (which is a good practice exercise) yields approximately 4.6×10^{42} positions.

The size of this number gives a rough estimate of how difficult the problem of analyzing all chess positions is. To appreciate the size of this number, let's assume that we can analyze a trillion (10^{12}) positions per second using our fastest computer. At that speed, it would take roughly 10^{31} seconds to analyze every possible position which is approximately 10 trillion times the estimated age of the universe.

In practice, modern chess playing programs don't take this approach of trying to analyze every position. Instead, they use more sophisticated search methods which we will discuss in more detail in a future week.

✓ Terminer