Cracking the Cracker Barrel Game

Christopher Frost
frost@virginia.edu

Michael Peck
mike@virginia.edu

The Cracker Barrel peg game, pictured in Figure 1, is a one player game played on a board of fifteen holes. The game begins with one hole empty and the other holes filled with pegs. The goal is to jump pegs over one another (removing the jumped peg) such that the board has only one remaining peg. We consider a generalization of the Cracker Barrel problem with a board of any size and some initial layout of pegs. The specific problem is:

> **CB:** Given an arbitrarily sized board with some initial configuration of pegs, is there a sequence of jumps such that the board will be left with one remaining peg?

Complexity

Our goal is to determine the complexity of the problem CB. Computer scientists examine time and memory growth as a function of input size to understand the complexity of a problem. Many complexity classes have been defined, the most important of which is nondeterministic polynomial, NP. NP is the set of problems which can be solved in polynomial time on a nondeterministic computer. Two other important complexity classes are P and NP-complete. P is the class of all decision problems that can be solved in polynomial time. NP-complete is the class of problems who are as hard as any other problem in NP.

The question "Does P=NP?" is the greatest unanswered question in computer science today; in fact, it is recognized as a Millennium Prize problem by the Clay Mathematics Institute. By showing CB to belong to the class NP-complete, we give researchers another problem to consider while pursuing the P=NP question. If a polynomial time algorithm is found which answers CB, then all problems in the class NP can be solved in polynomial time using CB, and it will have been proved that P=NP. Alternatively, if it can be shown that no polynomial time algorithm exists which answers CB, then since we have proved CB to be in the class NP, P≠NP.

Figure 1: Cracker Barrel Peg Game

Proof Method

A problem A is proved to be NP-complete by showing the following conditions are true:

1. A belongs to the class NP; an algorithm can be constructed which, given a solution, determines in polynomial time if the solution is correct.
2. Given any decision problem in the class NP, it can be transformed, in polynomial time, into an instance of A, whose answer is the same as the answer to the given decision problem.

We have already shown condition one for CB and produced such an algorithm and are progressing in our proof of condition two. We intend to prove condition two by demonstrating a reduction from a known NP-complete problem to CB. A known NP-complete problem, 3-SAT, asks if given a set of boolean variables and a set of clauses, each consisting of three boolean variables, is there an assignment of true or false to these variables so that the whole expression is true. In 1971, Cook showed 3-SAT to be in the class NP-complete.

We will show a reduction from 3-SAT to CB by:

1. Showing how any instance of 3-SAT can be transformed into an instance of CB.
2. Showing that 3-SAT's result is "yes," if and only if this instance of CB's result is also "yes."

Figure 2 shows a representation of an instance of the 3-SAT problem, $(x_1 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee x_3)$, as an instance of CB.

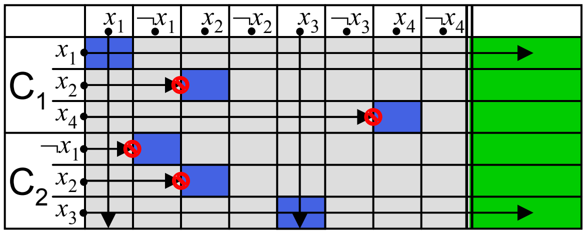

Figure 2: An instance of 3-SAT transformed into CB

The Boolean expression evaluates to true if x_1 and x_3 are true and x_2 is false. A peg is placed at each Boolean variable along the top row and is moved down the board if this variable has the value true. There is a peg at each Boolean variable instance in the clauses down the rows as well. The peg corresponding to a given term can move across the board if the corresponding Boolean term is satisfied, but will get stuck if it is not. The game will end with exactly one peg if and only if at least one peg of each clause has moved across the board. We have used this strategy to prove that a variation of CB where pegs are colored and a solution is a sequence of jumps that leaves just one peg of a particular color is NP-complete. We are progressing towards proof for the general problem CB.