Computer Analysis of Sprouts

David Applegate, Guy Jacobson, and Daniel Sleator

Sprouts is a popular (at least in academic circles) two-person pen-and-paper game. It was invented in Cambridge in 1967 by Michael Patterson, then a graduate student, and John Horton Conway, a professor of mathematics. Most people (including us) learned about this game from Martin Gardner's "Mathematical Games" column in the July 1967 issue of *Scientific American.*

The initial position of the game consists of a number of points called *spots*. Players alternate connecting the spots by drawing curves between them, adding a new spot on each curve drawn. Each curve must be drawn on the paper without touching itself or any other curve or spot (except at end points). A single existing spot may serve as both endpoints of a curve. Furthermore, a spot may have a maximum of three parts of curves connecting to it. A player who cannot make a legal move loses. Shown below is a sample game of two-spot Sprouts, with the first player winning. Since draws are not possible, either the first player or the second player can always force a win, regardless of the opponent's strategy. Which of the players has this winning strategy depends on the number of initial spots.

Sprouts is an *impartial* game: The same set of moves is available to both players, and the last player to make a legal move wins. Impartial games have variants where the condition of victory is inverted: The winner is the player who cannot make a legal move. This is called the losing or *misère* version of the game.

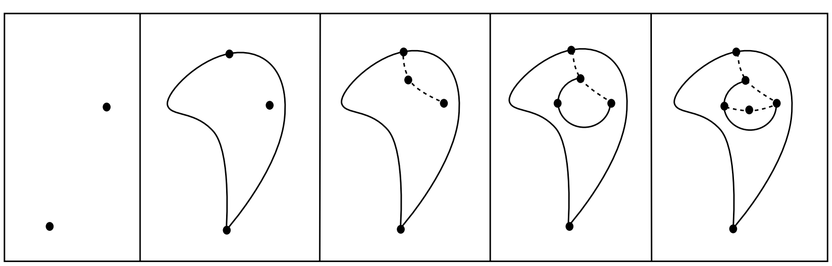

Figure 1. A sample game of two-spot Sprouts.

While Sprouts has very simple rules, positions can become fantastically complicated as the number of spots, n, grows. Each additional spot adds between two and three turns to the length of the game, and also increases the number of moves available at each turn significantly. Games with small numbers of spots can be (and have been) completely solved by hand, but as the number of spots increases, the complexity of the problem overwhelms human powers of analysis. The first proof that the first player loses in a six-spot game, performed by Denis Mollison (to win a 10-shilling bet!), ran to 47 pages.

Conway said that the analysis of seven-spot Sprouts would require a sophisticated computer program, and that the analysis of eight-spot Sprouts was far beyond the reach of present-day (1967!) computers. Of course, computers have come a long way since then. We have written a program that determines which player has a winning strategy in games of up to eleven spots, and in misère games of up to nine spots.

Our Sprouts Program

As far as we know, our program is the only successful automated Sprouts searcher in existence. After many hours of late-night hacking and experimenting with bad ideas, we managed to achieve sufficient time- and space-efficiency to solve the larger games. Our program is successful for several reasons:

- We developed a very terse representation for Sprouts positions. Our representation strives to keep only enough information for move generation. Many seemingly different Sprouts positions are really equivalent. The combination of this low-information representation and hashing (whereby the results of previous searches are cached) proved to be extremely powerful.
- Many sprouts positions that occur during the search are the *sum* of two or more non-interacting games. Sometimes it is possible to infer the value of the sum of two games given the values of the subgames. Our program makes use of these sum identities when evaluating normal Sprouts. These ideas are not nearly as useful in analyzing misère Sprouts. This is the principal reason that we are able to extend the analysis of the normal game further than the misère game.
- We used standard techniques to speed adversary search, such as cutting off the search as soon as the value is known, caching the results of previous searches in a hash table, and searching the successors of a position in order from lowest degree to highest.

- The size of the hash table turned out to be a major limitation of the program, and we devised and implemented two methods to save space without losing too much time efficiency. We discovered that saving only the losing positions reduced the space requirement by a large factor. To reduce the space still further we used a data compression technique.

Our Results

Here's what our program found:

Number of Spots	1	2	3	4	5	6	7	8	9	10	11
normal play	2	2	1	1	1	2	2*	2*	1*	1*	1*
misère play	1	2	2	2	1*	1*	2*	2*	2*		

A "1" means the first player to move has a winning strategy, a "2" means the second player has a winning strategy, and an asterisk indicates a new result obtained by our program.

The n-spot Sprouts positions evaluated so far fall into a remarkably simple pattern, characterized by the following conjecture:

Sprouts conjecture. *The first player has a winning strategy in n-spot Sprouts if and only if n is 3, 4, or 5 modulo 6.*

The data for misère Sprouts fit a similar pattern.

Misère sprouts conjecture. *The first player has a winning strategy in n-spot misère Sprouts if and only if n is 0 or 1 modulo 5.*

We are still left with the nagging problem of resolving a bet between two of the authors. Sleator believes in the Sprouts Conjecture and the Misère Sprouts Conjecture. Applegate doesn't believe in these conjectures, and he bet Sleator a six-pack of beer of the winner's choice that one of them would fail on some game up to 10 spots. The only remaining case required to resolve the bet is the 10-spot misère game. This problem seems to lie just beyond our program, our computational resources, and our ingenuity.[1]

[1] For a paper describing these results in more detail, go to http://www.cs.cmu.edu/~sleator