From Math Hysteria,
by Ian Stewart, Oxford
University Press

Yucky Choccy and Chomp are chocaholic games with very similar rules, but there the resemblance ends. The first is a 'dream game' with an easy winning strategy. The second is a 'nightmare game' – we know that the first player should always win, if they make the best moves, but we don't know what those moves are. Between them, these games teach us a lot about winning strategies and how to find them. Or not.

Just because a game has simple rules, that doesn't imply that there must be a simple strategy for winning it. Sometimes there is – noughts-and-crosses (tic-tac-toe) is a good example. But sometimes there isn't – another childhood game, Boxes, in which players take turns to fill in edges on a grid of dots and capture any square they complete, is a case in point (but see Chapter 16). I call the first kind 'dream games' and the others 'nightmare games', for fairly obvious reasons. Games with very similar rules can be surprisingly different when it comes to their dream or nightmare status. And, of course, the nightmare games are often the most interesting, because you can play them without knowing in advance who ought to win – or, in some cases, knowing who ought to win but not knowing how they can do it.

As an illustration of these surprising facts, I'm going to discuss two games based around chocolate bars. One, 'yucky choccy', is a dream game. The other, 'chomp', has very similar rules, but it's a nightmare game – with the startling extra ingredient that with optimal play the first player should always win, but nobody knows how.

I have no idea who invented yucky choccy: it was explained to me by Keith Austin, a British mathematician at Sheffield University. It takes place on an idealized chocolate bar, a rectangle divided into smaller squares. Players – I'll name them 'Wun' and 'Too' after the order in which they play – take turns to break off a lump of chocolate, which they must eat. Call this action a *move* in the game. The break must be a single straight line cutting all the way across the rectangle along the lines between the squares. The square in one corner contains

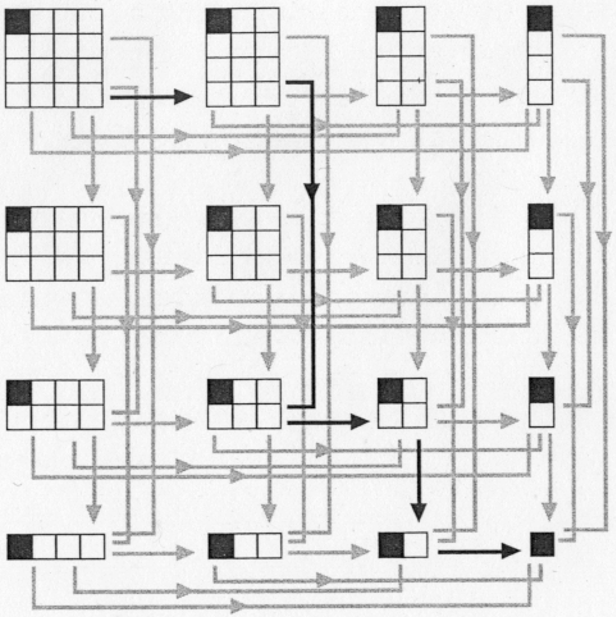


Figure 65
Game tree for 4 × 4 yucky choccy. Arrows indicate legal moves: the piece removed is eaten. The square shown in black is the soapy one. Solid arrows indicate an actual game, shaded arrows alternative moves that could have been made instead.

a lump of soap, and the player who has to eat this square loses. The solid arrows in Figure 65 shows the moves in the game played with a 4 × 4 bar, and the shaded arrows show all the other moves that could have been made instead. This entire diagram constitutes the *game tree* for 4 × 4 yucky choccy. As we'll shortly see, Too made a bad mistake and lost a game that should have been won.

A *winning strategy* is a sequence of moves that forces a win, no matter what moves the opponent makes. The concept of a strategy involves not just one game, but all possible games. When you play chess, most of your planning centres on 'what if' questions. 'If I advance my pawn, what could his queen do then?' Tactics and strategy centre around what moves you or your opponent *could* make in future, not just the moves that they *do* make.

There is a neat theory of strategies for 'finite' games – ones that can't continue forever and in which draws are impossible. It relies on two simple principles:

(1) A position is a winning one if you can make *some* move that places your opponent in a losing position.

(2) A position is a losing one if *every* move that you can make places your opponent in a winning position.

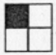
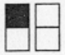
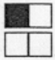
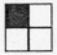
The logic here may seem circular, but it's not: it's recursive. The difference is that with recursive reasoning you have a place to start. To see how, I'll use the above two principles to find a winning strategy for 4 × 4 yucky choccy. The trick is to start from the end and work backwards, a process called 'pruning the game tree'.

The single soapy piece ■ is a losing position. I'll symbolize that fact by the diagram

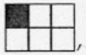
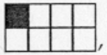
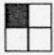
L * * *
* * * *
* * * *
* * * *

whose entries refer not to a chocolate bar, but to the various positions marked in Figure 65. Here 'L' means 'losing position', * means 'don't know yet', and 'W' will mean 'winning position' once I've found some. In fact, ■□ ■□□ ■□□□ are all winning positions, because you can break off all the white squares in one move to leave your opponent with the single-piece losing position. Equivalently, there are arrows in the game tree that lead from those positions directly to ■, and by principle (1) all such positions are winners. For similar reasons the same positions rotated through a right angle are also winners, so now we have pruned away all branches of the game tree that lead in one step to the single soapy square, which tells us the status of those positions:

L W W W
W * * *
W * * *
W * * *

What about ? Well, the only moves you can make are  or , and when you remove the all-white piece you leave a winning position for your opponent. Principle (2) now tells us that  is a loser, so we can prune one more branch to get

L	W	W	W
W	L	*	*
W	*	*	*
W	*	*	*

This in turn implies that , , and so on are winners (break off a chunk to leave ) leading to

L	W	W	W
W	L	W	W
W	W	*	*
W	W	*	*

Working backwards in this manner you can eventually deduce the win/lose status of any position. The logic runs not in circles, but in interlocking spirals, climbing down the game tree from leaf to twig, from twig to branch, from branch to limb ... Hence the 'pruning' image. We have to start from the *end*, though, which is a nuisance. What we really want to do, though, is chop down the entire game tree in one blow, George Washington fashion, to find the status of the opening position – and if it's a winner, to find what move to play. For games with a small tree there's no difficulty: repeated pruning yields the status of all positions. In Figure 65 we can carry this out, to get

L	W	W	W
W	L	W	W
W	W	L	W
W	W	W	L

So the 4 × 4 position, for instance, is a loser.

If you try larger bars of chocolate, square or rectangular, you'll quickly find that the same pattern emerges: losers live along the diagonal line, all other bars are winners. Now the bars on that diagonal are the square ones: 1 × 1, 2 × 2, 3 × 3, 4 × 4. This suggests a simple strategy that should apply to bars of any size: squares are losers, rectangles are winners. Having noticed this apparent pattern, we can check its validity *without* working through the entire game tree by verifying properties (1) and (2). Here's the reasoning. Clearly any rectangle (winner) can always be converted to a square (loser) in one move. In contrast, whatever move you make starting with a square (loser), you cannot avoid leaving your opponent a rectangle (winner). Moreover, is square, and we know it is a losing position. All this is consistent with principles (1) and (2), so working backwards we deduce (recursively) that *every* square is a loser and *every* rectangle a winner. We now see that Too's first move in Figure 65 was a mistake. And we see that yucky choccy is a dream game no matter what size the bar is.

In principle the same procedure applies to any finite game. The opening position is the 'root' of the game tree. At the other extreme are the tips of the outermost twigs, which terminate at positions where one or other player has won. Since we know the win/lose status of these terminal positions, we can work backwards along the branches of the game tree using principles (1) and (2), labelling positions 'win' or 'lose' as we proceed. The first time, we determine the status of all positions that are one move away from the end of the game. The next time, we determine the status of all positions that are two moves away from the end of the game, and so on. Since, by assumption, the game tree is finite, eventually we reach the root of the tree – the opening position. If this gets the label 'win' then Wun has a winning strategy; if not, Too has.

We can even say, again in principle, what the winning strategy is. If the opening position is 'win' then Wun should always move to a position labelled 'lose' – which Too will then face. Because this is a losing position, any move Too makes presents Wun with a 'win' position. So Wun can repeat the same strategy until the game ends. Similarly, if the opening position is labelled 'lose', then Too has a winning strategy – with the same description. So in finite, drawless games, working backwards through the game tree in principle decides the status of all positions, including the opening one. I say 'in principle' because the calculations become intractable if the game tree is large. And even simple games can have huge game trees, because the game tree involves all possible positions and all possible lines of play. This opens the door to nightmare games.

We now contrast yucky choccy with a game whose rules are almost the same, but where pruning the game tree rapidly becomes impossible – and where pruning *is* possible, it does not reveal any pattern that could lead to a simple strategic recipe. That game is chomp, invented many years ago by David Gale (University of California at Berkeley) and described in his marvellous book on recreational mathematics, *Tracking the Automatic Ant* (Further Reading). Gale describes chomp using a rectangular array of cookies or biscuits, but I'll stick to chocolate. (It is best played with an array of buttons or the like.) Chomp is just like yucky choccy, with the sole difference that a legitimate move consists of removing a rectangular chunk of chocolate, as in Figure 66. Specifically, a player chooses a component square and then removes all squares in that row and column, together with all squares to the right of and above these.

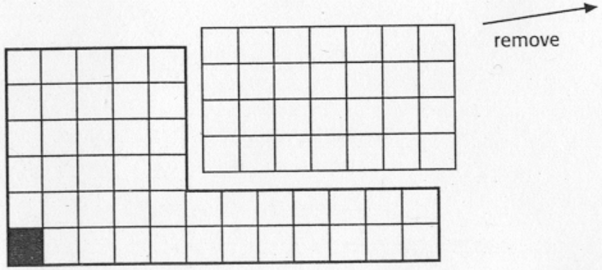


Figure 66
Typical move in chomp.

There is a neat proof that for any size of bar (Figure 67a) other than 1 × 1, chomp is a win for Wun. Suppose, to the contrary, that Too has a winning strategy. Wun then proceeds by removing the upper right square (Figure 67b). This cannot leave Too facing a losing position, since we are assuming the opening position is a loser for Wun. So Too can play a winning move, something like Figure 67c, to leave Wun facing a loser. But then Wun could have played Figure 67d, leaving Too facing the same loser. This contradicts the assumption that Too has a winning strategy, so that assumption must be false. Therefore Wun has a winning strategy.

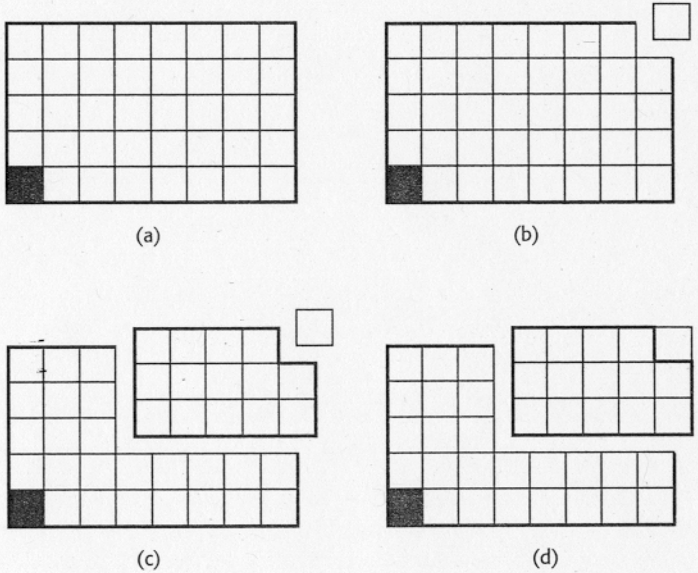


Figure 67

(a) Chomp bar ready for strategy stealing.

(b) If Wun does this ...

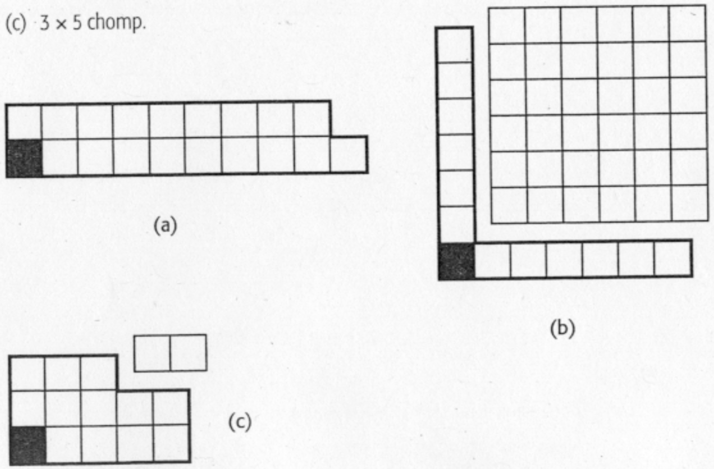
(c) ... and Too makes a supposed winning move ...

(d) ... then Wun could have played Too's move in the first place.

Proofs of this kind are called 'strategy stealing'. If Wun can make a 'dummy' move, pretend to be the second player, and win by following what ought to be a winning strategy for Too, then Too could not have had such a strategy to begin with – implying that Wun must have a winning strategy. The irony of this method of proof, when it works, is that it offers no clue to what Wun's winning strategy should be!

For chomp, detailed winning strategies are unknown, except in a few simple cases. In the 2 × *n* (or *n* × 2) case, Wun can always ensure that Too faces a position that is a rectangle minus a single corner square (Figure 68a). In the *n* × *n* case, Wun removes everything except an *L*-shaped edge (Figure 68b), and after that copies whatever move Too makes, but reflected in the diagonal. A few other small cases are known: for example in 3 × 5 chomp the sole winning move for Wun is Figure 68c. 'The' winning move need not be unique: in the 6 × 13 game there are two different winning moves.

Figure 68
Winning moves in
(a) 2 × *n* chomp.
(b) n × *n* chomp.
(c) 3 × 5 chomp.



Other information about chomp positions can be found in *Winning Ways* by Berlekamp, Conway, and Guy (Further Reading). Chomp can also be played with an infinite chocolate bar – in which case, paradoxically, it remains a finite game because after finitely many moves only a finite portion of bar remains. But there is a change: Too can sometimes win. This happens, for example, with the 2 × ∞ bar. Figure 69 shows that whatever Wun does, Too can choose a reply that leads to Figure 68a, which we already know is a loser. Strictly speaking, I should be more careful here. By '∞' I really mean the set of positive integers in their usual order, which set theorists symbolize as ω ('omega') and refer to as 'the first infinite ordinal'. There are many other infinite ordinals, but their properties are too technical to describe here: see Gale's book for further details. Chomp can be played on doubly infinite arrays of ordinals, or in three or more dimensions: on the whole, little is known about winning strategies for these generalizations.

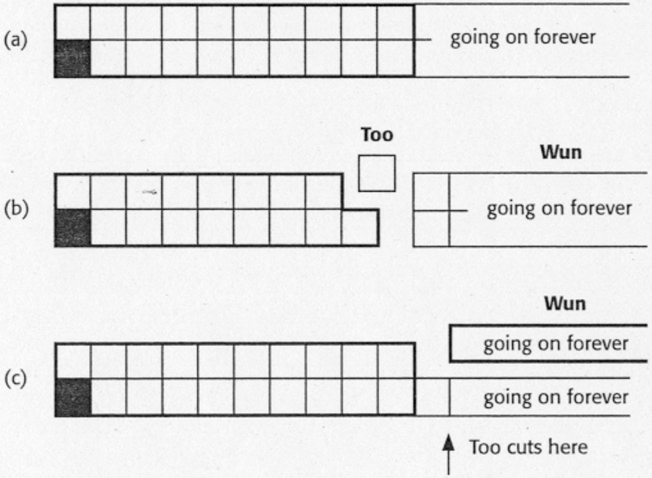


Figure 69 How Too wins 2 × ∞ chomp.
(a) Start.
(b) One type of possible play for Wun and its reply.
(c) The other type of possible play for Wun and its reply.