Entropy of non rectangular LEGO bricks

Jon Wilson
Ferris State University

joint work with David McClendon
Ferris State University

MOVES Conference
MoMath, New York
August 7, 2017

What is a LEGO Brick?

A *LEGO brick* is a plastic building toy which typically has studs on one side and holes on another side used for interlocking them.

Most LEGO bricks are rectangular prisms. Here is a picture of a 2×4 LEGO brick (the studs are on the top; the holes are on the bottom):

Question: Suppose you connect n LEGO bricks of the same size (and color) together. How many different buildings can you make?

Notation

Define B to be a specific type of LEGO brick (for example, a 2×4 brick).

Then let $T_B(n)$ be the number of buildings (counted up to rotations and translations) that can be constructed out of n bricks of type B.

Main Question: What kind of function is $T_B(n)$? How fast does it grow?

What is entropy?

Definition: The *entropy* of a LEGO brick of type B is the number

$$h_B = \lim_{n\to\infty} \frac{1}{n} \log T_B(n)$$

(that this limit exists needs to be proven).

Idea: The entropy of a function captures its exponential growth rate. If h_B exists and is finite, then $T_B(n) \sim 2^{h_B n}$ so T_B grows exponentially at rate h_B.

Note: we use log base 2, but the base is not important.

Remark: By "entropy", we mean information entropy, which is somewhat different than the thermodynamic entropy you learn about in chemistry.

History

In a paper published in 2014 by Durhuus and Eilers, the authors showed:

1. The entropy of any rectangular LEGO brick is finite.

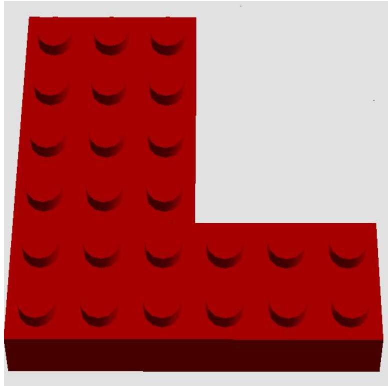
   (Reason: superadditivity of a sequence growing at the same rate as $\log T_B(n)$.)

2. $\log 78 \leq h_{2\times 4} \leq \log 192$. (The methods they use could be adapted to give bounds for any rectangular brick.)

We want to extend these results to other types of LEGO bricks.

L-shaped LEGO bricks

A brick in class $\mathcal{L}(B, W, b, w)$ is a $B \times W$ rectangular brick, with a $b \times w$ notch cut out of the upper-right corner (when the brick is rotated so that the side of length B is horizontal):



The picture above is a brick in class $\mathcal{L}(6, 6, 3, 4)$.

General results about L-shaped bricks

Lemma For any B, W, b and w,

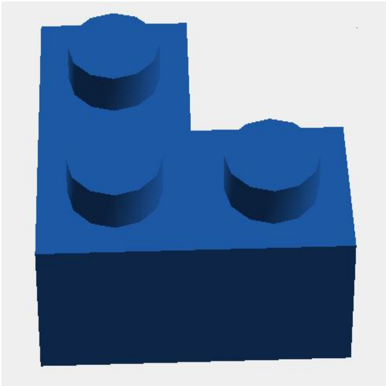
$$T_{\mathcal{L}(B,W,b,w)}(2) = 2(2B-1)(2W-1)+2(B+W-1)^2-9(B-b)(W-w).$$

Theorem 1 (McClendon-W) For any B, W, b and w, $h_{\mathcal{L}(B,W,b,w)}$ exists and is finite.

Theorem 2 (McClendon-W) $\log T_{\mathcal{L}(B,W,b,w)}(2) \leq h_{\mathcal{L}(B,W,b,w)} \leq$

$$\log\left(\frac{(2(BW-(B-b)(W-w))-1)^{BW-(B-b)(W-w)-1}(BW-(B-b)(W-w))}{(2(BW-(B-b)(W-w))-2)^{(BW-(B-b)(W-w))-2}}\right).$$

Our favorite example: $\mathcal{L}(2,2,1,1)$



From the formula on the previous slide:

$$T_{\mathcal{L}(2,2,1,1)}(2) = 27 \Rightarrow h_{\mathcal{L}(2,2,1,1)} \geq \log 27.$$

Our favorite example: $\mathcal{L}(2, 2, 1, 1)$

Using techniques involving generating functions, we have improved the lower bound to

$$h_{\mathcal{L}(2,2,1,1)}(2) \geq \log 36.$$

As an interesting aside, this bound shows that 2×2 L-shaped brick has more entropy than a 2×2 square (which has entropy at most $\log 34$ by the techniques of Durhuus and Eilers), despite having fewer studs.

Our favorite example: $\mathcal{L}(2,2,1,1)$

The crude upper bound coming from our theorem is

$$h_{\mathcal{L}(2,2,1,1)} \leq \log 177$$

(as we will see, this can be significantly improved).

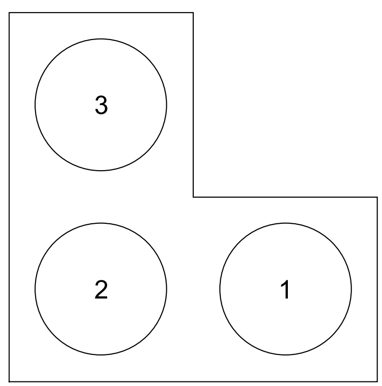
Where does this crude upper bound come from?

Finding the upper bound

Consider a finite string of $6(n-1)$ symbols taken from a "alphabet" of size 13 (we use the alphabet $\{0, 1, 2, ..., 13\}$.

Example: $0, 9, 0, 7, 0, 0, 0, 2, 0, 0, 6, 0, ...$

Start with one brick; call this "brick # 1". This brick has three studs on top, and three holes on the bottom. Number the studs and holes as follows:



Finding the upper bound

Example: $0, 9, 0, 7, 0, 0, 0, 2, 0, 0, 6, 0, \ldots$

Now look at the first three symbols. These tell you, respectively, whether or not to attach a brick to the top of stud 1, 2 and/or 3 of brick # 1 (a zero tells you not to attach a brick to that stud; any number from 1 to 12 tell you to attach a brick... each number corresponds to a different way to attach the new brick to that stud).

For the example above, you would attach one new brick on top of stud 2 of brick # 1. Call this new brick "brick # 2".

Finding the upper bound

Example: $0, 9, 0, 7, 0, 0, 0, 2, 0, 0, 6, 0, ...$

Now look at the next three symbols. These tell you, respectively, whether or not to attach a brick to the **bottom** of stud 1, 2 and/or 3 of brick # 1 (as before, a zero tells you not to attach a brick; the numbers from 1 to 12 tell you to attach a brick... each number corresponds to a different way to attach the new brick to that stud).

For the example above, you would attach one new brick beneath hole 1 of brick # 1. Call this new brick "brick # 3" (and keep numbering the new bricks in order as they are attached).

Finding the upper bound

Example: $0, 9, 0, 7, 0, 0, 0, 2, 0, 0, 6, 0, ...$

The next two groups of symbols tell you how to attach new bricks to the top and/or bottom of brick # 2, etc.

Keep going until you run through the entire (finite) sequence.

Finding the upper bound

Some of these sequences will lead to contradictions: for instance,

- you might be told to attach the wrong number of bricks (you need to end up with n bricks hooked together); or

- two bricks might be forced to occupy the same space

The sequences that do not lead to a contradiction are called *allowable.* Since every configuration of n bricks comes from at least one allowable sequence, any upper bound on the number of allowable sequences gives us an upper bound on $T_B(n)$.

Finding the upper bound

We compute an upper bound on the number of allowable sequences using methods including:

- brute-force counting of simple configurations of ≤ 4 bricks;
- computer calculations; and
- combinatorial estimates involving Stirling's formula.

This gives the formula for the upper bound appearing in Theorem 2.

Improving the upper bound

In particular, the methods shown on the preceding slides show that one can "code" a LEGO building made from n $\mathcal{L}(2,2,1,1)$ bricks by a string of $6(n-1)$ symbols taken from an alphabet of size 13. From this, we get

$$h_{\mathcal{L}(2,2,1,1)} \leq \log 177.$$

Actually, one can code these buildings much more efficiently; with a more complicated coding that uses strings of $5n-9$ symbols taken from an alphabet of size 10, we get the improved bound

$$h_{\mathcal{L}(2,2,1,1)} \leq \log 110.$$

We don't know what the most efficient coding is.