An additivity theorem for plain Kolmogorov complexity *

Bruno Bauwens[†] Alexander Shen[‡]

February 16, 2012

Abstract

We prove the formula $C(a,b) = K(a|C(a,b)) + C(b|a,C(a,b)) + O(1)$ that expresses the plain complexity of a pair in terms of prefix-free and plain conditional complexities of its components.

The well known formula from Shannon information theory states that $H(\xi,\eta) = H(\xi) + H(\eta|\xi)$. Here ξ,η are random variables and H stands for the Shannon entropy. A similar formula for algorithmic information theory was proven by Kolmogorov and Levin [5] and says that

$$C(a,b) = C(a) + C(b|a) + O(\log n),$$

where a and b are binary strings of length at most n and C stands for Kolmogorov complexity (as defined initially by Kolmogorov [4]; now this version is usually called *plain* Kolmogorov complexity). Informally, $C(u)$ is the minimal length of a program that produces u, and $C(u|v)$ is the minimal length of a program that transforms v to u; the complexity $C(u,v)$ of a pair (u,v) is defined as the complexity of some standard encoding of this pair.

This formula implies that $I(a:b) = I(b:a) + O(\log n)$ where $I(u:v)$ is the amount of information in u about v defined as $C(v) - C(v|u)$; this property is often called "symmetry of information". The term $O(\log n)$, as was noted in [5], cannot be replaced by $O(1)$. Later Levin found an $O(1)$-exact version of this formula that uses the so-called *prefix-free* version of complexity:

$$K(a,b) = K(a) + K(b|a,K(a)) + O(1);$$

this version, reported in [2], was also discovered by Chaitin [1]. In the definition of prefix-free complexity we restrict ourselves to self-delimiting programs: reading a program from left to right, the interpreter determines where it ends. See, e.g., [7] for the definitions and proofs of these results.

In this note we provide a somewhat similar formula for plain complexity (also with $O(1)$-precision):

Theorem 1.

$$C(a,b) = K(a|C(a,b)) + C(b|a,C(a,b)) + O(1).$$

Proof. The proof is not difficult after the formula is presented. The $\leq$-inequality is a generalization of the inequality $C(x,y) \leq K(x) + C(y)$ and can be proven in the same way. Assume that p is a self-delimiting program that maps $C(a,b)$ to a, and q is a (not necessarily self-delimiting) program that maps a and $C(a,b)$

*B. Bauwens is supported by Fundaçao para a Ciência e a Tecnologia by grant (SFRH/BPD/75129/2010), and is also partially supported by project CSI^2 (PTDC/EIAC/099951/2008). A. Shen is supported in part by projects NAFIT ANR-08-EMER-008-01 grant and RFBR 09-01-00709a. Author are grateful to their colleagues for interesting discussions and to the anonymous referees for useful comments.

†Instituto de Telecomunicações Faculdade de Ciência at Porto University, R.Campo Alegre, 1021/1055, 4169-007 Porto - Portugal.

‡Laboratoire d'Informatique, de Robotique et de Microlectronique de Montpellier, UMR 5506 - CC 477, 161 rue Ada, 34095 Montpellier Cedex 5 France.

to b. The natural idea is to concatenate p and q; since p is self-delimiting, given pq one may find where p ends and q starts, and then use p to get a and q to get b. However, this idea needs some refinement: in both cases we need to know $C(a,b)$ in advance; one may use the length of pq as a replacement for it, but since we have not yet proven the equality, we have no right to do so.

So more caution is needed. Assume that the $\leq$-inequality is not true and $C(a,b)$ exceeds $K(a|C(a,b)) + C(b|a,C(a,b))$ by some d. Then we can concatenate prefix-free description $\bar{d}$ of d (that has length $O(\log d)$), then p and then q. Now we have enough information: first we find d, then $C(a,b) = |p| + |q| + d$, then a, and finally b. Therefore $C(a,b)$ does not exceed $O(\log d) + |p| + |q| + O(1)$, therefore $d \leq O(\log d) + O(1)$ and $d = O(1)$. The $\leq$-inequality is proven.

Let us prove the reverse inequality. In this proof we use the interpretation of prefix-free complexity as the logarithm of a priori probability (see, e.g., [7] for details). If $n = C(a,b)$ is given, one can start enumerating all pairs (x,y) such that $C(x,y) \leq n$; there are at most 2^{n+1} of them and the pair (a,b) is among them. For fixed x, for each pair (x,y) in this enumeration we add 2^{-n-1} to the probability of x; in this way we approximate (from below) the semimeasure $P(x|n) = N_x 2^{-n-1}$. Therefore, we get an upper bound for $K(a|n)$:

$$K(a|n) \leq -\log P(a|n) + O(1) = n - \log_2 N_a + O(1),$$

where N_a is the number of y's such that $C(a,y) \leq n$. On the other hand, given a and n, we can enumerate all these y, and b is among them, so b can be described by its ordinal number in this enumeration, therefore

$$C(b|a,n) \leq \log_2 N_a + O(1).$$

Summing these two inequalities, we get the desired result. □

We can now get several known $O(1)$-equalities for complexities as corollaries of Theorem 1.

- Recall that $C(a, C(a)) = C(a)$, and $K(a, K(a)) = K(a)$ (the $O(1)$-additive terms are omitted here and below), since the shortest program for a also describes its own length.

- For empty b we get $C(a) = K(a|C(a))$, see also [3, 6].

- For empty a we get $C(b) = C(b|C(b))$, see also [3, 6].

- The last two equalities imply that $C(u|C(u)) = K(u|C(u))$.

  The direct proof for last three statements is also easy. To show that $C(a) \leq C(a|C(a))$, assume that some program p maps $C(a)$ to a and is d bits shorter than $C(a)$. Then we add a prefix $\bar{d}$ of length $O(\log d)$ that describes d in a self-delimiting way, and note that $\bar{d}p$ determines first $C(a)$ and then a, so $d \leq O(\log d) + O(1)$ and $d = O(1)$. To show that $K(a|C(a)) \leq C(a|C(a))$ we note that in the presence of $C(a)$ every program of length $C(a)$ can be considered as a self-delimiting one, since its length is known.

  Levin also pointed out that $C(a)$ can be defined in terms of prefix-free complexity as a minimal i such that $K(a|i) \leq i$. (Indeed, for $i = C(a)$ both sides differ by $O(1)$, and changing right hand side by d, we change left hand side by $O(\log d)$, so the intersection point is unique up to $O(1)$-precision. In other terms, $K(a|i) = i + O(1)$ implies $C(a) = i + O(1)$.)

- More generally, we may let a be some fixed computable function of b: if $a = f(b)$, we get $C(b) = K(f(b)|C(b)) + C(b|f(b), C(b))$.

One can also see that Theorem 1 can be formally derived from Levin's results mentioned above. To show that

$$C(b|a, C(a,b)) = C(a,b) - K(a|C(a,b))$$

we need to show that the right hand side $i = C(a,b) - K(a|C(a,b))$ satisfies the equality $K(b|a, C(a,b), i) = i$ with $O(1)$-precision, which implies $C(b|a, C(a,b)) = i$. (We omit all $O(1)$-terms, as usual.) In the condition of the last inequality we may replace i by $K(a|C(a,b))$ since $C(a,b)$ is already in the condition. Therefore, we need to show that

$$K(b|a, C(a,b), K(a|C(a,b))) = C(a,b) - K(a|C(a,b))$$

or

$$K(b|a, C(a,b), K(a|C(a,b))) + K(a|C(a,b)) = C(a,b).$$

But the sum in the left hand side equals $K(a,b|C(a,b))$ due to the formula for prefix complexity of a pair (a,b) relativized to the condition $C(a,b)$, and it remains to note that $K(a,b|C(a,b)) = C(a,b)$. (This alternative proof was suggested by Peter Gacs.)

We can obtain a different version of Theorem 1:

Proposition 1.

$$C(a,b) = K(a|C(a,b)) + C(b|a, K(a|C(a,b))) + O(1).$$

Proof. Indeed, the $\leq$-inequality can be shown in the same way as the $\leq$-inequality in the proof of Theorem 1, hence it remains to show the $\geq$-inequality. Let p be a program of length $C(b|a, C(a,b))$ that computes b given a and $C(a,b)$. (The program p is not assumed to be self-delimiting.) Knowing p, we can also compute b given a and $K(a|C(a,b))$. First, we compute $|p| + K(a|C(a,b))$, and this sum equals $C(a,b)$ (Theorem 1). Then, using a again, we compute b. Hence $C(b|a, C(a,b)) \geq C(b|a, K(a|C(a,b)))$. □

One may complain that Theorem 1 is a bit strange since it uses prefix-free complexity in one term and plain complexity in the second (conditional) part. As we have already noted, one cannot use C in both parts: $C(a,b)$ can exceed even $C(a) + C(b)$ by a logarithmic term. One may then ask whether it is possible to exchange plain and prefix-free complexity in the two terms we have and prove that $C(a,b)$ equals something like

$$C(a|C(a,b)) + K(b|a, C(a,b)).$$

It turns out that it is not possible: even the inequality $C(a,b) \leq C(a) + K(b|a) + O(1)$ is not true. At first it seems that one could concatenate a self-delimiting program q that produces b given a and a (plain) program p that produces a, in the hope that the endpoint of q can be reconstructed, and then the rest is p. However, this idea does not work: the program q is self-delimiting only when a is known; to know a we need to have p, and to know p we need to know where q ends, so there is a vicious circle here.

Let us show that the problem is unavoidable and that for infinitely many pairs (x,y) we have

$$C(x,y) \geq C(x) + K(y|x) + \log n - 2\log\log n - O(1),$$

where $n = |x| + |y|$ is the total length of both strings. To construct such a pair, let $n = 2^k$ for some k, and choose a string r of length n and a natural number $i < n$ such that $C(r,i) \geq n + \log n$. (For every n, there are $n2^n$ pairs (r,i), so one of them has high complexity.)

Let $x = r_1 \ldots r_i$ and $y = r_{i+1} \ldots r_n$. Note that $C(x,y) = C(r,i) \geq n + \log n$ and that $C(x) \leq i$. Furthermore, $K(y|x) \leq K(y|x,n) + K(n)$. Here $K(y|x,n) \leq |y| = n - i$, since x and n determine $|y|$ and $K(y \mid |y|) \leq |y|$; on the other hand, $K(n) \leq 2\log\log n$.[1]

There is still some chance to get a formula for the plain complexity of a pair (x,y) that involves only plain complexities, assuming that we add some condition in the left hand side, i.e., to get some formula of the type $C(a,b|?) =?$. Unfortunately, the best result in this direction that we managed to get is the following observation:

Proposition 2. *For all x,y there exists a (unique up to $O(1)$-precision) pair (k,l) such that $C(x|l) = k$, $C(y|x,k) = l$. For such a pair we have $C(x|l) = k$, $C(y|x,k) = l$ and this implies $C(x,y|k,l) = C(x,y|k) = C(x,y|l) = l + k$ (all with $O(1)$-precision).*

Proof. The pair in question is a fixed point of $F\colon (k,l) \mapsto (C(x|l), C(y|x,k))$. It exists and is unique since F maps points at distance d into points at distance $O(\log d)$. (Here "distance" means geometric distance between points in $\mathbb{Z}^2$.)

[1] As a byproduct of this example and the discussion above we conclude that $K(x|y)$ cannot be defined as minimal prefix-free complexity of a program that maps y to x: the value $K(y|x)$ can be smaller than $\min\{K(p) : U(p,x) = y\}$, where U is the universal function. Indeed, in this case we would have the inequality $C(x,y) \leq C(x) + K(y|x)$, since the prefix-free description of a program that maps x to y and a shortest description for x can be concatenated into a description of the pair (x,y).

Using the relativized version of the statement $C(z) = C(z|C(z))$, we conclude that $C(x|k,l) = k$ and $C(y|x,k,l) = l$. Let us prove now that $C(x,y|k,l) = k+l$. Indeed, the standard proof of Kolmogorov–Levin theorem shows that for any x,y,k',l' such that

$$C(x,y|k',l') \leq k'+l'$$

we have either

$$C(x|k',l') \leq k' \quad \text{or} \quad C(y|x,k',l') \leq l'.$$

Hence if $C(x|k,l) = k$ and $C(y|x,k,l) = l$ for some k and l, we have $C(x,y|k,l) \geq k+l$ (otherwise k and l can be decreased to get a contradiction). By concatenation we obtain also that $C(x,y|k,l) \leq k+l$, so $C(x,y|k,l) = k+l$ (all equations with $O(1)$-precision).

It remains to show that $C(x,y|k,l) = k+l$ implies $C(x,y|k) = k+l$ and, similarly, $C(x,y|l) = k+l$. Indeed, a program of length $k+l$ that maps (k,l) to (x,y), can be used to map k (or l) to (x,y): knowing the length of the program and one of the values of k and l, we reconstruct the other value. □

Remark 1. *One can ask what can be said about pairs (k',l') such that $C(x|l') \leq k'$ and $C(y|x,k') \leq l'$. The pair (k,l) given by the theorem is not necessarily coordinate-wise minimal: for example, taking a large k' that contains full information about y we may let $l' = 0$. Indeed, $C(x|0) \leq k'$ (since k' is large) and $C(y|x,k') \leq 0$ (since k' determines y). However, to get some decrease in k' (compared to k) or l' (compared to l) we need to change the other parameter by an exponentially bigger quantity, since the information distance between i and i' is $O(\log|i-i'|)$. The change in the other parameter should be its increase, otherwise we could repeat the arguments exchanging k and l and get a contradiction (each of two changes could not be exponentially big compared to the other one).*

References

[1] G. Chaitin, A Theory of Program Size Formally Identical to Information Theory, *Journal of the ACM*, **22**(3):329–340 (1975).

[2] P. Gács. On the symmetry of algorithmic information. *Soviet Math. Dokl.*, **15**(5):1477–1480 (1974).

[3] P. Gács. Lecture notes on descriptional complexity and randomness. `http://www.cs.bu.edu/faculty/gacs/papers/ait-notes.pdf`, (1988-2012).

[4] A.N. Kolmogorov, Three approaches to the quantitative definition of information, *Problemy peredachi Informatsii*, vol. 1, no. 1, pp. 3–11 (1965)

[5] A.N. Kolmogorov, Logical basis for information theory and probability theory, *IEEE Transactions on Information Theory*, IT-14(5): 662–664 (1968)

[6] M. Li and P.M.B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, New York, (2008).

[7] A. Shen, *Algorithmic Information theory and Kolmogorov complexity*. Technical report TR2000-034, Uppsala University (2000).