Bulletin of the **S**ection of **L**ogic
Volume 20/1 (1991), pp. 10–13

reedition 2005 [original edition, pp. 10–13]

Wojciech Suchoń

COUNTING FORMULAE OF PROPOSITIONAL CALCULUS

Anybody who learned syllogistic knows that there are 256 moods divided among four figures – 64 moods for every figure. In 1953 C.A. Meredith in his paper published in Dominican Studies gave the full answer to the question how many moods and figures are there for compound syllogisms of a given number of terms. So the problem of counting formulae of a logical calculus was raised up and solved if it is only for syllogistic.

We will use prefix notation. The first presupposition we take is that the functors occurring in the vocabulary of language are one-argument or two-argument only. The second assumption: the length of counted formulae is given. And the last one: we will count only substitutionally inequivalent formulae.

Two formulae are substitutionally equivalent if and only if there exist substitutions transforming one of them into the other and vice versa. An example: formulae Cpq, Crs are substitutionally equivalent and Cpq, Cpp are not.

The fourth assumption means that we will count in fact not formulae but rather their construction patterns.

At first let us confine ourselves to the analysis of the formulae built from symbols of implication C, of negation N and the sole propositional variable p only.

The value of the function $F(LNH, APP)$ will be a number of substitutionally inequivalent formulae of the length LNH and with APP appearances of the variable p.

It is obvious that in any formula a number of the appearances of the variable must not exceed $INT((LNH + 1)/2)$, so $F(LNH, APP) = 0$

always when $APP > INT((LNH+1)/2)$.

Other obvious statement is that for every $LNH : F(LNH,1) = 1$.

In order to count the value of the function F we have an arithmetic formula:

$$F(LNH, APP) = F(LNH-1, APP)+$$
$$+\sum_{K=1}^{LNH-2}\sum_{L=1}^{APP-1} F(K,L) * F(LNH-K-1, APP-L)$$

The first component gives the number of formulae obtained from the shorter one by adding negation. The second component is a sum of the numbers of possible combination of shorter formulae which can be connected by implication taking in account the requirement on total length and total number of appearances of the variable.

If we allow now use of more functors – let say $FUN1$ one-argument functors and $FUN2$ two-argument ones we receive the recursive definition of the function F:

$$F(LNH, APP) = FUN1 * F(LNH-1, APP)+$$
$$+FUN2 * \sum_{K=1}^{LNH-2}\sum_{L=1}^{APP-1} F(K,L) * F(LNH-K-1, APP-L)$$

Let us remove the limitation on number of variables occurring in the formula. Every place occupied up till now by variable p may be filled by a variable different from any other. But it is also possible that the same variable occurs in more than one place.

There are as many different ways of filling places for variables in a formula as equivalence relations in the set of those places or – in other words – as many as partitions of the mentioned set.

The number of parts is any such partition may be conceived as the number of different variables occurring in the formula.

Let the value of the function $P(APP, VAR)$ be the number of partitions dividing the set of APP elements on VAR parts.

It is rather obvious that:

$$P(APP, APP) = 1$$
$$P(APP, 1) = 1$$
$$\text{for } VAR > APP: \qquad P(APP, VAR) = 0$$

The definition of the function P is recursive:

$$P(APP, VAR) = P(APP-1, VAR) * VAR + P(APP-1, VAR-1).$$

The first component is the number of partition of the set possessing one element less than the considered set on VAR parts multiplied by number of parts. This is the number of partitions obtained by adding one element to some set being member of partition of smaller set. The second component is the number of partitions constructed from partitions having $VAR-1$ parts by appending as the new part, formed by the singleton made from the added element.

Now, let $FOR(LNH, APP, VAR)$ be the number of all formulae of the length LNH, with APP appearances of variables and VAR different variables occurring in it.

$$FOR(LNH, APP, VAR) = F(LNH, APP) * P(APP, VAR)$$

The final answer to our question about number of substitutionally inequivalent formulae of given length is:

$$FORMULAE(LNH) = \sum_{APP=1}^{INT((LNH+1)/2)} \sum_{VAR=1}^{APP} FOR(LNH, APP, VAR)$$

It is quite hopeless task to calculate values of the function $FORMULAE$ when LNH is more than 3 or 4. Only the computer is patient enough to do such things. The author wrote in BASIC a quite short program counting values of the functions F and P (KRZ002 [1]) Some results received using the above program may be quoted:

Let us consider De Morgan laws. They are built using functors A, C, K, N. Their length is 10. There are 712369 formulae of the same length. More similar formulae are those with four appearances of two different variables. There are 79380 such formulae.

It is known that the shortest unique axiom for the purely equivalential classical propositional calculus must consist of 11 symbols and must have 3 different variables. There are 3780 formulae which fulfill those requirements.

[1] This program is a part of the KRZ system. The KRZ system was written in 1989 by the author for research project RPBP III-24.

Department of Logic
Jagiellonian University
Cracow, Poland