

# Benchmarks for Combinations Calculation

## 1. Intro

The tests have been made in a laptop with a 2.40 GHz 2-core processor, without any other applications running in background.

Due to the lazy evaluation from Haskell, to force the calculation of the combinations, they are saved into text files. Although this may produce a slight inaccuracy on the results, its still possible to get some concrete conclusions.

As the minimal combinations and minimal ruled combinations produce a very small number of combinations, we can discard them, and focus testing in full ruled combinations.

Two sets of results will be presented, each of them represents a different input with its respective set of rules. The first set will be a large one (580.000 possible combinations), and the second will be a smaller one (2.400 possible combinations).

## 2. Results

### 2.1. Set 1

Total Categories: 10

Total Classes: 40

Total Possible Combinations: 580608

Rules	Full	1	2	3	4	5	6
#Combinations	580608	290304	48384	32832	2592	456192	129024
Time (seconds)	14,988	7,902	1,297	0,888	0,077	12,405	3,47

### 2.2. Set 2

Total Categories: 6

Total Classes: 24

Total Possible Combinations: 2400

Rule	Full	1	2	3	4	5
#Combinations	2400	2040	1360	1200	600	900
Time (seconds)	0,037	0,038	0,025	0,021	0,011	0,017

### **3. Conclusions**

For big combinatorial trees, the results point to one simple conclusion: the less combinations the result has, the faster it is to execute. This happens because the time it takes to calculate the semantic of the rule is much smaller than the time necessary to fully combine with remaining categories. Therefore more restricting rules (ones with more categories specified, and therefore less possible combinations) will calculate faster.

This, however, does not stand for smaller inputs. As we can see in the second set of results, some of the ruled combinations take longer to calculate than the full combination. As in this case the time to compute the combination with the remaining categories is much smaller, the time taken to compute the rule takes a bigger weight in the total processing time.