

# **Экспериментальный анализ производительности CFRQ- алгоритмов**

Выполнил: Павел Кеворкянц

Дата: 1 октября 2020

Информация о машине:

- Ubuntu 20.4
- Intel Core i5 9300
- 8 gb RAM

Информация о замерах:

Были произведены замеры алгоритмов вычисления контекстно-свободных запросов: алгоритм Хеллингса, матричное произведение и тензорный алгоритм. Первым двум алгоритмам на вход подавались граф и грамматика в ОНФХ. Тензорному алгоритму подавались пары как с грамматикой в ОНФХ, так и не в ОНФХ. Графы и грамматики были взяты из папки DataForFLCourse. Все замеры проводились для грамматик с названием 'g1', по 5 раз, затем бралось среднее значение.

Точность – 5 знаков после запятой. Прочерки в таблицах означает, что вычисление заняло более 1000 секунд.

### FullGraph:

graph	tensor	tensor+crf	hellings	MxM
fullgraph_10	0,00914	0,01612	0.02202	0.00055
fullgraph_50	0,08453	0,1345	0.33906	0.00252
fullgraph_100	0,5235	0,8238	4.76155	0.00672
fullgraph_200	2,10448	5,93212	65.13312	0.02606
fullgraph_500	11,35296	23,30663	967,35563	0.14489

### Memory Aliases

graph	tensor	tensor+crf	hellings	MxM
bzip2.txt	0.00343	0,00462	0.002573	0.00021
gzip.txt	0,14675	0,22523	19.76711	0.00026
ls.txt	0,00745	0,00912	6.09662	0.00022
pr.txt	0,00325	0,0463	2.7581	0.00021
wc.txt	0,00412	0,01434	0.67755	0.00027

# SparseGraph

graph	tensor	tensor+crf	hellings	MxM
G10k-0.001	0,56796			0,02568
G10k-0.01	0,53623	10,58362		0,04361
G10k-0.1	0,23943	2,64958	244,83325	0,00634
G20k-0.001	0,39123			0,01941
G40k-0.001	1,14124			0,16334
G5k-0.001	0,02346	1,3463	20,25235	0,00035
G80k-0.001	4,25023			0,52352

### WorstCase

graph	tensor	tensor+crf	hellings	MxM
worstcase_4	0,00324	0,00635	0,00457	0,00084
worstcase_8	0,00853	0,01934	0,016455	0,00078
worstcase_16	0,04667	0,02557	0,03522	0,00953
worstcase_32	0,53667	2,55895	0,23566	0,03312
worstcase_64	6,77352	14,79934	0,12562	0,8456
worstcase_128	70,34637	250,74742	10,93466	0,95563
worstcase_256			276,25235	2,25303
worstcase_512				70,35631
worstcase_1024				803,34632
worstcase_2048				

### Выводы:

На первом месте по производительности оказался алгоритм с матричным произведением. Алгоритм Хеллингса оказался самым последним по производительности, кроме тестов с худшими вариантами, где он оказался вторым. В среднем он уступил тензорным алгоритмам. Интересен тот факт, что работа тензорного алгоритма на основе грамматики в ОНФХ оказалось заметно медленнее тензорного алгоритма на основе грамматики не в ОНФХ.

Существенные различия во времени работы алгоритмов тензорного произведения, скорее всего, обусловлены различным количеством состояний рекурсивного автомата, построенного по грамматике.