

Отчет

Работу выполнил:

Мирзазянов Глеб, 371 группа, матмех СПбГУ

Основные результаты:

<https://github.com/Glebuska/formal-language2020>

Железо:

ОС: Ubuntu 20.04

CPU: Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz

RAM: 8Gb DDR3

Датасет:

<https://drive.google.com/file/d/1BkiAFd1rYzPA0uoHo7TQvITvp-j8QVFM/view?usp=ssharing>

Цель эксперимента

Основная цель данного эксперимента: сравнение производительности CFPQ-запросов через следующие реализации:

- 1) Алгоритм Хеллингса(hellings)
- 2) Через произведение булевых матриц(multiplication)
- 3) Через тензорное произведение булевых матриц, на входе данная грамматика(tensor)
- 4) Через тензорное произведение булевых матриц, на входе ОНФХ-грамматика(tensor weak cnf или tensor_wcnf)

Эксперименты проводились 5 раз для каждого метода и бралось среднее арифметическое.

Результаты эксперимента

Результаты замеров изложены в таблице ниже. Использовалась часть данных, так как на более ресурсоемкие графы не хватило оперативной памяти. К тому же, по таблице заметно, что я не проверял тензорное произведение для графов из папки MemoryAliases, опять же из-за нехватки

RAM. Еще один интересный момент, что на графах из этой же папки алгоритм Хеллингса медленнее перемножения больше чем в 100 раз.

Graph	Grammar	Algotitm	Time
fullgraph_100	g3	hellings	12.417
		mult	0.002
		tensor	0.655
		tensor_wcnf	1.550
	g2	hellings	41.418
		mult	0.004
		tensor	1.507
		tensor_wcnf	2.911
	g1	hellings	22.317
		mult	0.002
		tensor	0.709
		tensor_wcnf	2.138
	g4	hellings	56.624
		mult	0.004
		tensor	0.856
		tensor_wcnf	2.341
fullgraph_50	g3	hellings	1.677
		mult	0.001
		tensor	0.163
		tensor_wcnf	0.388
	g2	hellings	5.287
		mult	0.002
		tensor	0.396
		tensor_wcnf	0.651
	g1	hellings	3.129
		mult	0.001
		tensor	0.164
		tensor_wcnf	0.569
	g4	hellings	7.521
		mult	0.002
		tensor	0.217
		tensor_wcnf	0.650

fullgraph_10	g3	hellings	0.022
		mult	0.000
		tensor	0.008
		tensor_wcnf	0.018
	g2	hellings	0.073
		mult	0.001
		tensor	0.018
		tensor_wcnf	0.030
	g1	hellings	0.056
		mult	0.001
		tensor	0.010
		tensor_wcnf	0.027
	g4	hellings	0.093
		mult	0.001
		tensor	0.010
		tensor_wcnf	0.028
worstcase_4	g1	hellings	0.006
		mult	0.003
		tensor	0.001
		tensor_wcnf	0.003
worstcase_16	g1	hellings	0.061
		mult	0.034
		tensor	0.001
		tensor_wcnf	0.006
worstcase_128		hellings	3.349
		mult	5.591
		tensor	0.003
		tensor_wcnf	0.033
worstcase_64		hellings	0.833
		mult	0.739
		tensor	0.002
		tensor_wcnf	0.017
worstcase_32		hellings	0.218
		mult	0.138
		tensor	0.001
		tensor_wcnf	0.010
worstcase_256		hellings	13.846

worstcase_8		mult	71.013
		tensor	0.009
		tensor_wcnf	0.073
		hellings	0.019
		mult	0.009
		tensor	0.001
		tensor_wcnf	0.004
wc.txt	g2	hellings	5.181
		mult	0.662
	g1	hellings	8.649
		mult	0.136
bzip2.txt	g2	hellings	145.252
		mult	0.199
	g1	hellings	253.119
		mult	0.358

Выводы

Алгоритм произведения булевых матриц работал быстрее на всех данных. Алгоритм тензорного произведения через булевы матрицы с исходной грамматикой работает ощутимо быстрее, чем с преобразованием в ОНФХ-грамматику. Алгоритм Хеллингса оказался самым медленным из всех 4-х вариантов реализации.