

Экспериментальный анализ реализации

Автор: Суханова Анжела

Задание: 6

1 ноября 2020

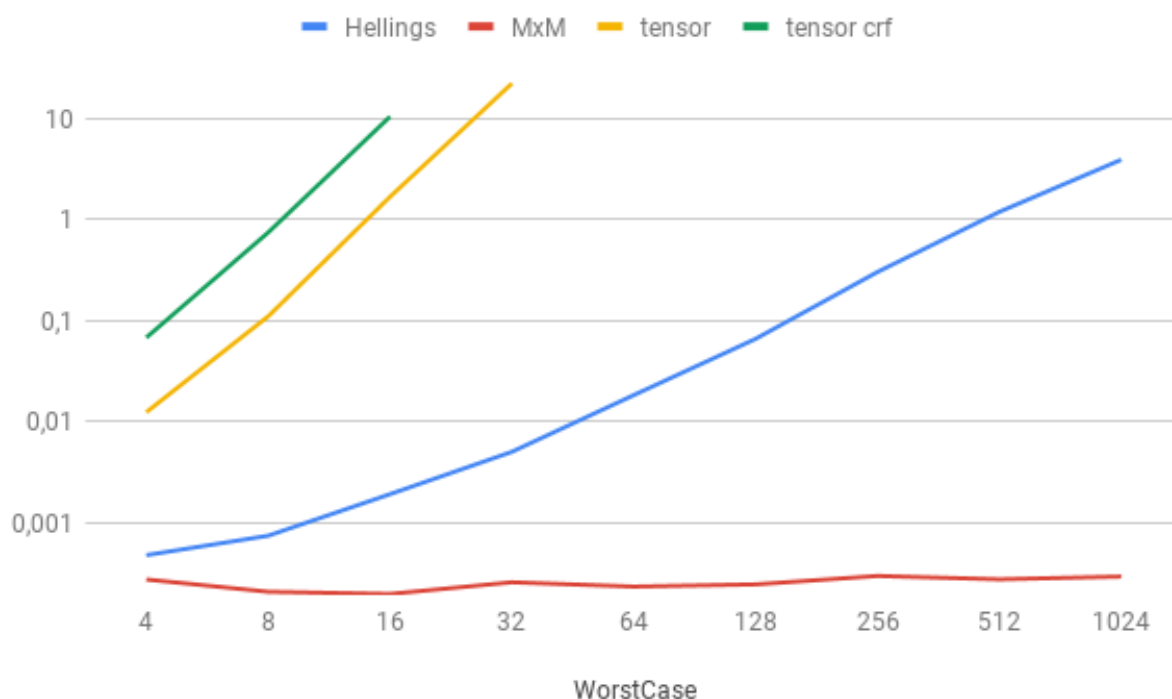
Комментарии к работе

Замеры проводились на ноутбуке с Ubuntu 20.04, Intel Core i5-7300HQ CPU, 2.50GHz, DDR4 8Gb RAM.

На графиках 1, 2 и 3 представлены средние значения данных (в секундах), полученных по 5-ти запускам для каждого теста. Все данные для замеров взяты из архива DataForFLCourse. Измерения были произведены по грамматике g1. Результаты замеров округлены до 6-ого знака после запятой.

1. Сравнение производительности алгоритмов КС-достижимости

Рис. 1: WorstCase, сек



Измерения на графах из SparseGraph представлены только для алгоритма, использующего умножение булевых матриц, так как реализации остальных алгоритмов работают значительно дольше.

Рис. 2: SparseGraph, сек

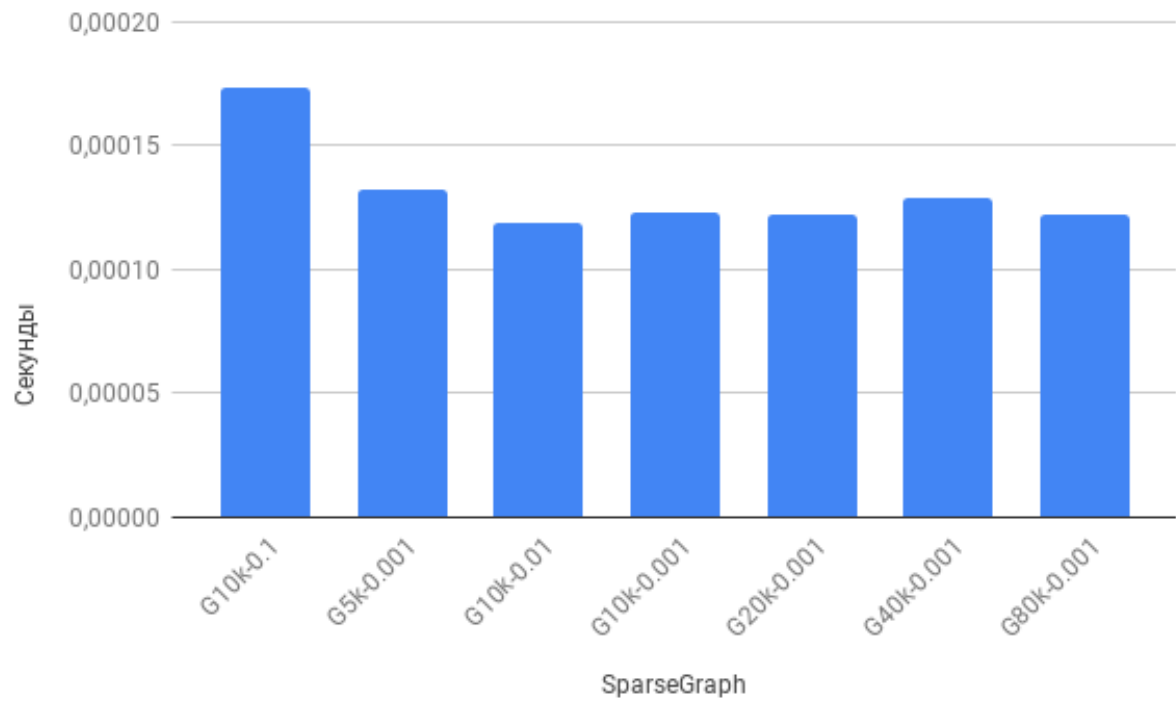
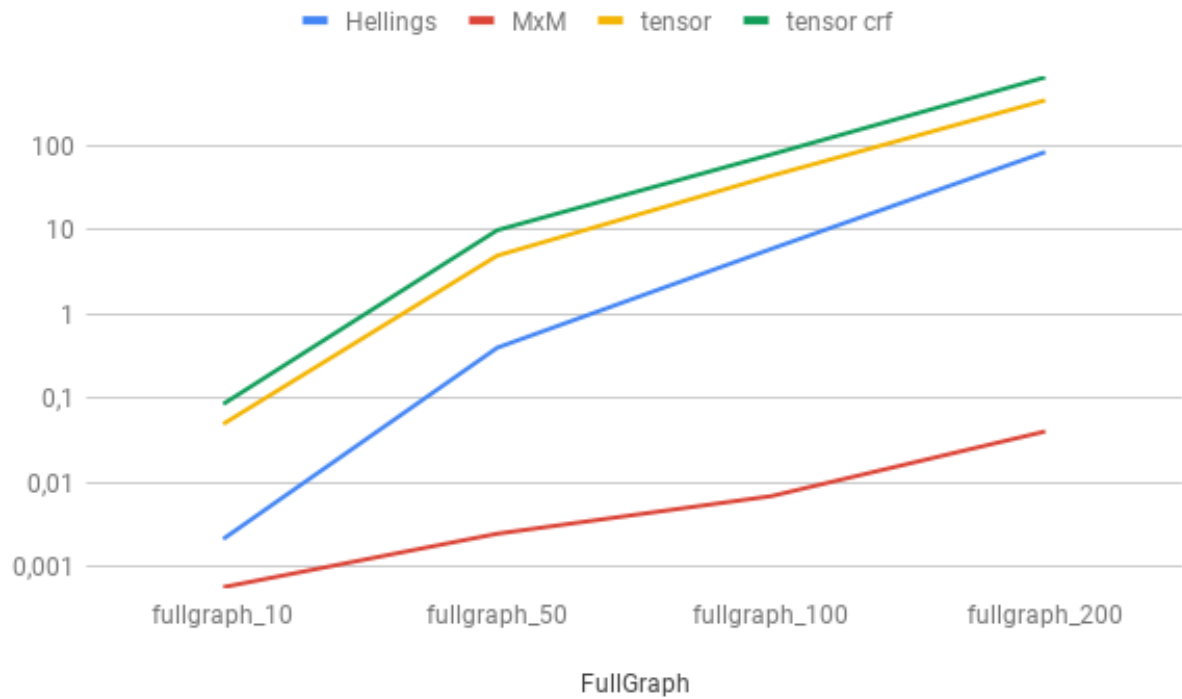


Рис. 3: FullGraph, сек



2. Выводы

Можно видеть, что для всех графов, по которым производились замеры, реализация алгоритма, основанного на умножении булевых матриц, ощутимо обгоняет остальные. Худшие результаты демонстрирует представление тензорного алгоритма, причём, оно работает дольше с грамматикой в ОНФХ, чем с непреобразованной. Последнее можно объяснить появлением дополнительных состояний в строящем рекурсивном автомате (то есть ростом соответствующего графа) и увеличением искомой матрицы, вызванных преобразованием. Алгоритм Хеллингса на всех входных данных обгоняет тензорный алгоритм, но отстаёт от матричного.