Особенности взаимодействия CPU и GPU при решении задачи синтаксического анализа

Автор:Митенев Алексей
мат-мех, 371 группа
Научный руководитель:
Ст. преп., магистр информационных технологий
Григорьев Семен Вячеславович

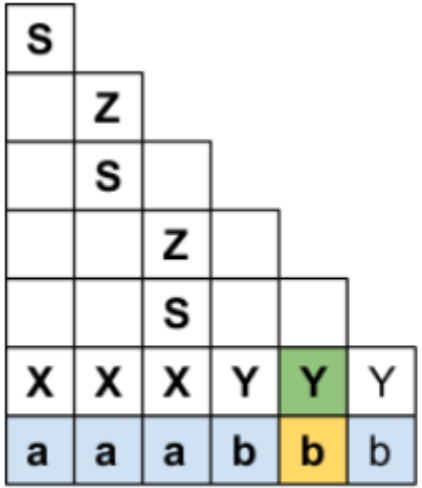
Алгоритм CYK

S -> X Y | X Z

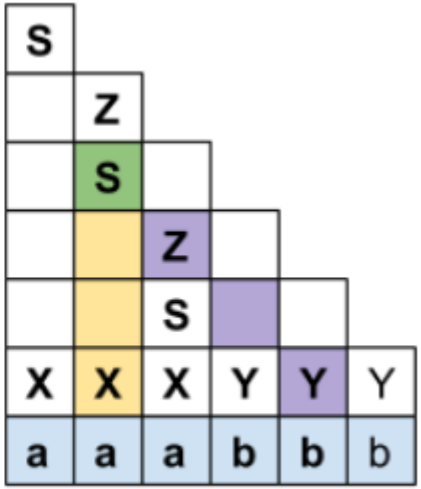
Z -> S Y

X -> a

Y -> b



(1) Y -> b

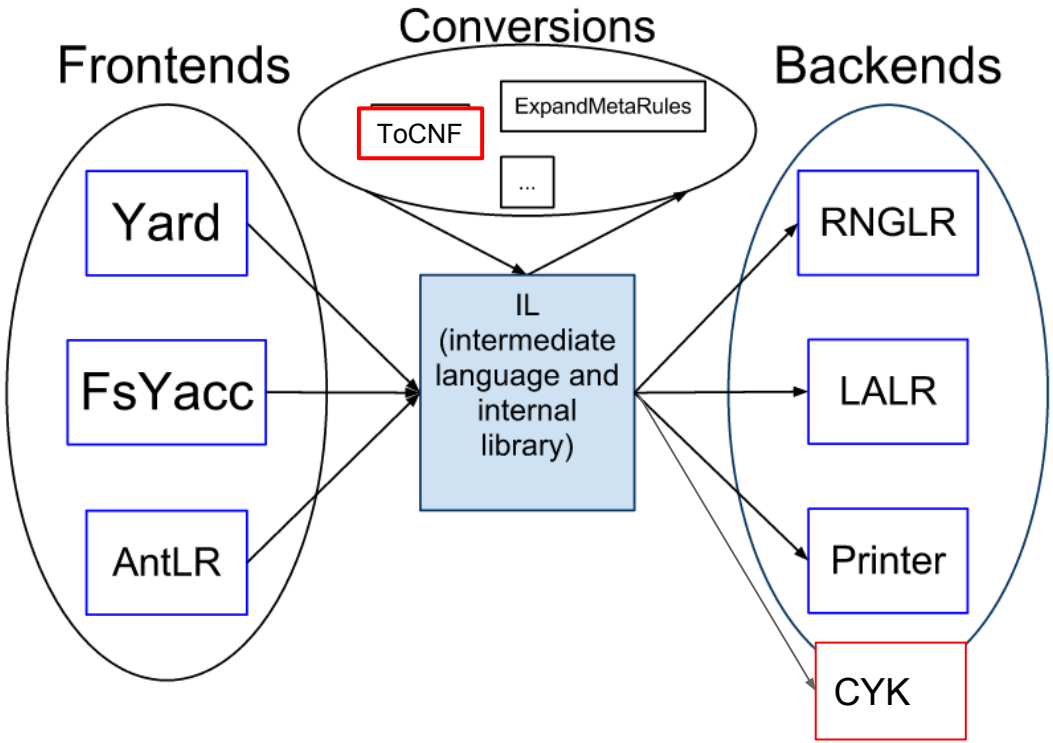


(2) S -> X Z

Постановка задачи

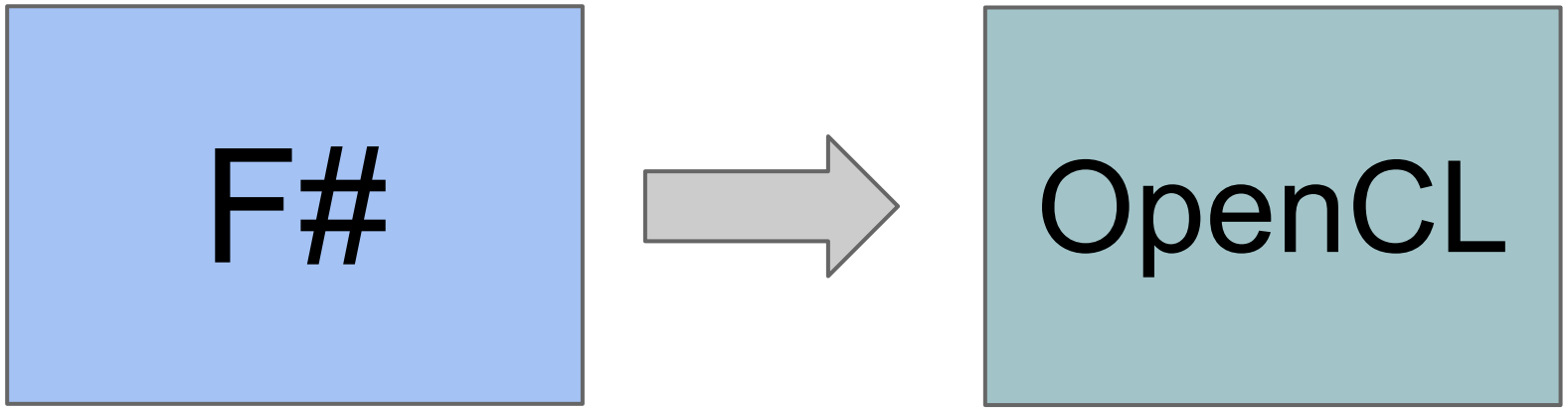
- Реализовать алгоритм CYK и оптимизировать его под выполнение на GPU
- Проанализировать реализованные оптимизации на результативность
- Протестировать алгоритм на грамматике Transact SQL

YaccConstructor



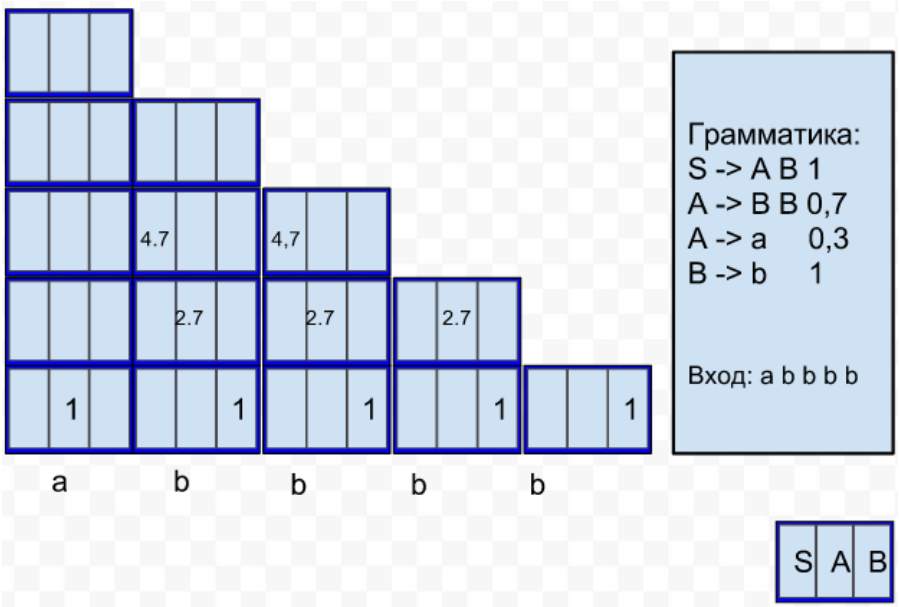
Brahma F#

Трансляция кода F# в OpenCL



Модифицированный CYK

- Можно подавать на вход взвешенную грамматику
- Поддержка диалектов



Реализация

Оптимизации

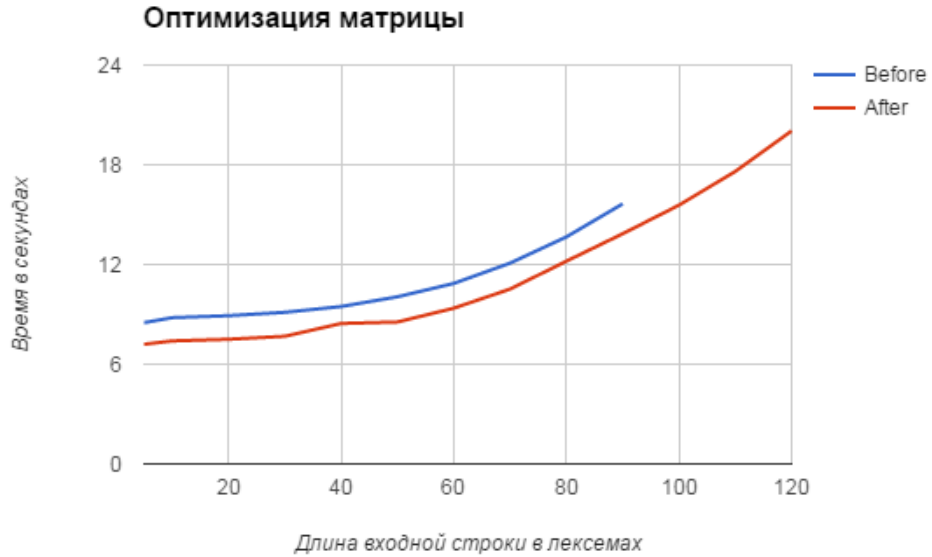
- Уменьшение размера пересылаемой матрицы
- Фильтрация грамматики
- Выбор “базы” для распараллеливания
- Использование возможностей Brahma F# для уменьшения объема кода
- Подбор LocalGroupSize

Синхронизация

- Используя атомарные операции

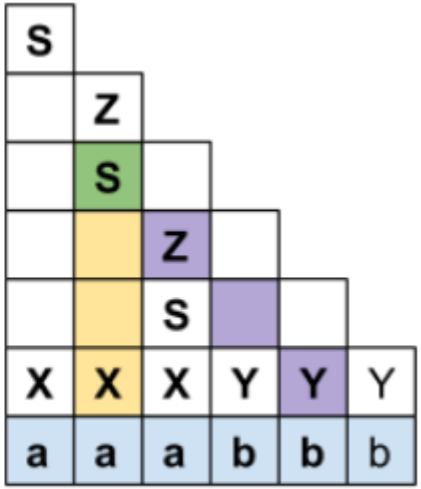
Уменьшение матрицы

- Минимизация размера ячейки
- Разворачивание треугольной матрицы в массив



Выбор базы распараллеливания

- Отдельный поток на каждое правило ✓
- Отдельный поток на обработку нетерминала



S -> X Y | X Z

Z -> S Y

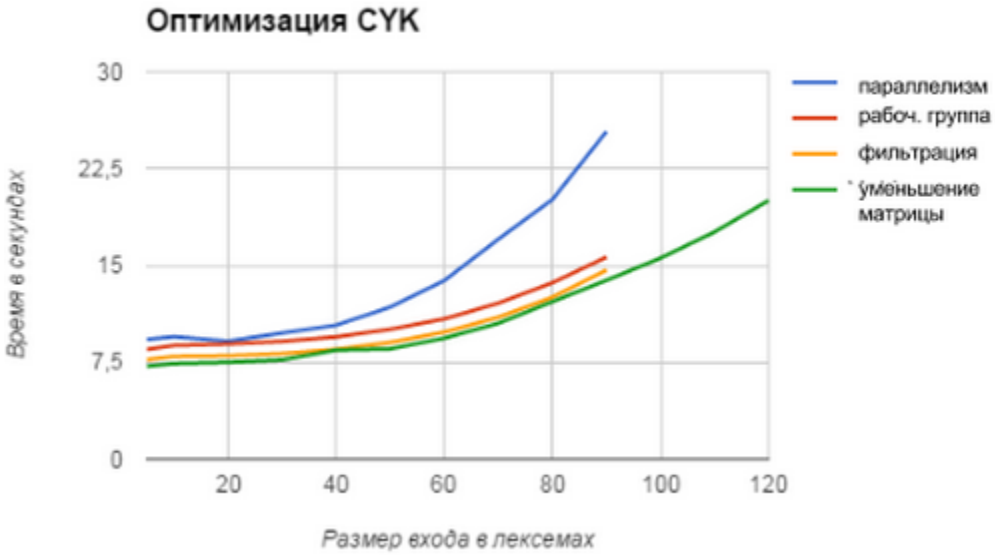
X -> a

Y -> b

Оптимизация: график

Тестирование алгоритма проходило на грамматике TransactSql.

Количество правил: 6046 (в НФХ), Количество нетерминалов: 1172



CYK: сравнение

Процессор: AMD A6-5200 APU - 4 ядра, 2 ГГц
Видеокарта: AMD Radeon HD 8400 - 128 ядер

Итоги

- Алгоритм CYK реализован и оптимизирован под выполнение на GPU
- Был выполнен набор оптимизаций и было выявлено их влияние на производительность.
- Алгоритм CYK протестирован на грамматике Transact SQL