Распараллеливание алгоритмов синтаксического анализа, основанных на матричных операциях

Автор: Сусанина Юлия Алексеевна, 344 группа
Научный руководитель: к.ф.-м.н., доцент Григорьев С.В.

Санкт-Петербургский государственный университет
Кафедра системного программирования

21 мая 2018г.

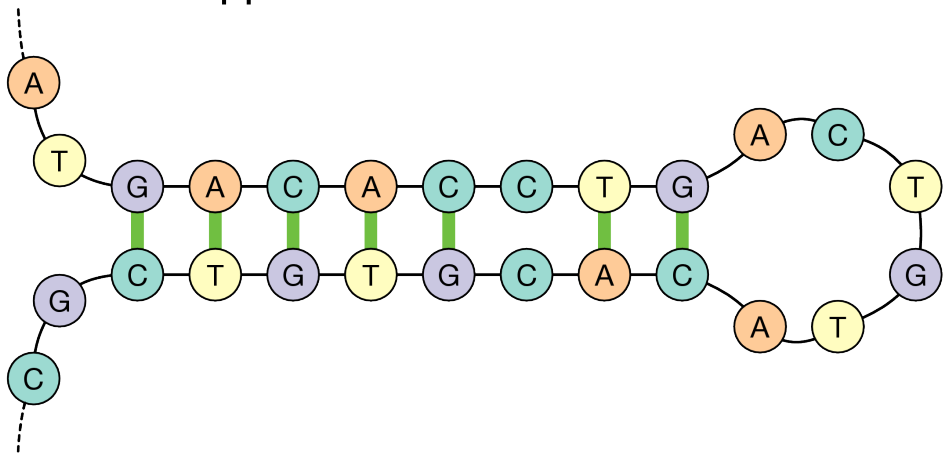
Предметная область

- **Синтаксический анализ** — процесс определения принадлежности некоторой последовательности лексем языку, порождаемому грамматикой
- **Область применения:** биоинформатика

Биоинформатика

- Первичная структура

- Вторичная структура
  - Определение принадлежности организма семейству по вторичной структуре тРНК, 16s рРНК
  - Вид вторичной структуры можно задать с помощью контекстно-свободной грамматики
- **Задача:** поиск подстроки РНК, которая сворачивается во вторичную структуру определенного вида



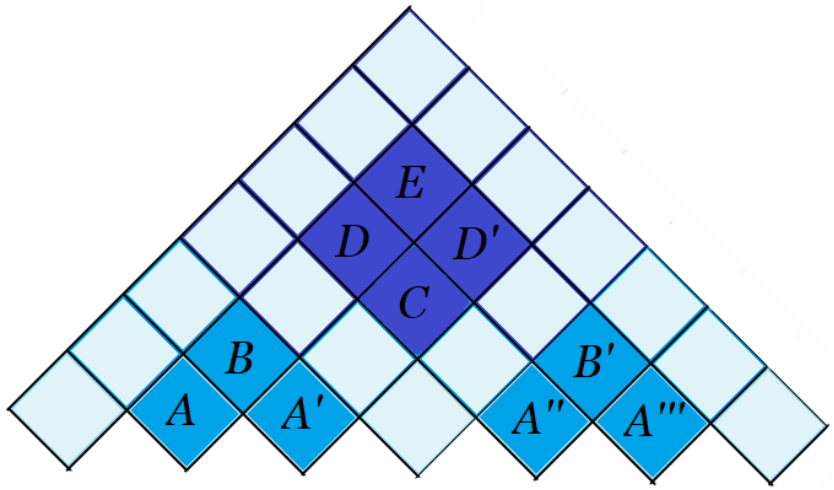
Алгоритмы синтаксического анализа

- **Вход**:
  - $a_1 ... a_n$ — строка
  - G — КС-грамматика в нормальной форме Хомского
- **Результат**: матрица разбора, элементы которой отвечают за выводимость конкретной подстроки из стартового нетерминала S ($a_{i+1} ... a_j \in L_G(S) \Leftrightarrow S \in T[i, j]$)

Алгоритм А.С.Охотина

- Okhotin A. Parsing by matrix multiplication generalized to Boolean grammars
  - Разбиение исходной матрицы и перемножение подматриц меньшего размера

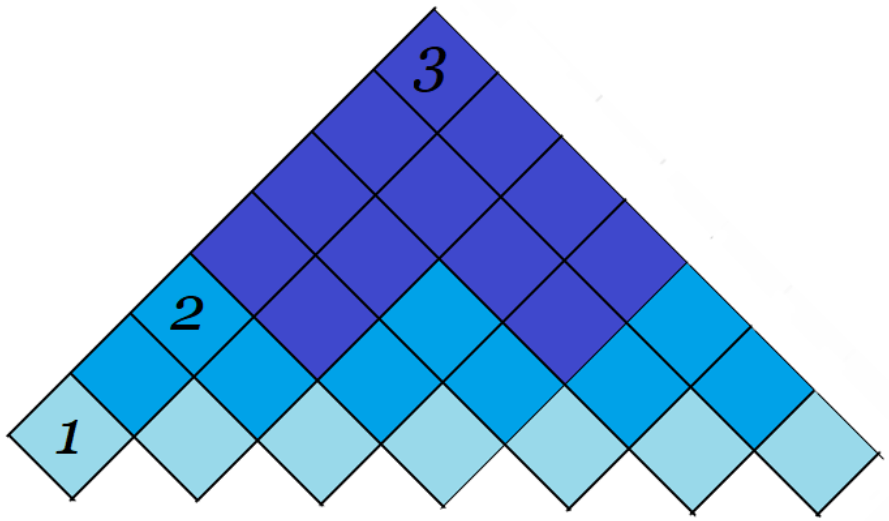
$$\mathcal{D} = \mathcal{D} + \mathcal{B} \times \mathcal{C}$$
$$\mathcal{D}' = \mathcal{D}' + \mathcal{B} \times \mathcal{C}$$
$$\mathcal{E} = \mathcal{E} + \mathcal{B} \times \mathcal{D}'$$
$$\mathcal{E} = \mathcal{E} + \mathcal{D} \times \mathcal{B}'$$



- **Основной недостаток**: сложность разделения на независимые потоки

Модификация

- Явейн А. Разработка алгоритма синтаксического анализа через перемножение матриц
  - Реорганизация вычислений
  - Возможность разбиения на слои подматриц
  - Использование параллелизма на уровне перемножения подматриц слоя
- Реализация отсутствует



Постановка задачи

Цель: исследование и реализация модифицированного алгоритма А.С.Охотина.

Для достижения данной цели были поставлены следующие задачи:

- Реализовать модифицированный алгоритм, а также исходный алгоритм А.С.Охотина
- Дать теоретическую оценку эффективности использования параллельных вычислений в модифицированном алгоритме
- Провести экспериментальное исследование модифицированного алгоритма

Детали практической реализации

- Представление матрицы разбора в виде нескольких булевых матриц
  - Матрица — соответствующий нетерминал
  - $T[i,j] = \{A \in N | \, T_A[i,j] = true\}$
  - Умножения независимы

Используемые технологии

- Реализованы в рамках исследовательского проекта YaccConstructor
- .NET, F#
- The NVIDIA cuBLAS (basic linear algebra subroutines) library

Оценка сложности модифицированного алгоритма

Теорема (оценка сложности последовательной версии)

Модифицированный алгоритм строит таблицу разбора для грамматики G и строки длины n за время $O(|G| \cdot BMM(n) \cdot log(n))$.

Теорема (оценка сложности параллельной версии)

Параллельная версия модифицированного алгоритма строит таблицу разбора для грамматики G и строки длины n за время $O(|G| \cdot BMM(n))$.

- BMM(n) — количество операций необходимых для перемножения двух булевых матриц размера n $\times$ n

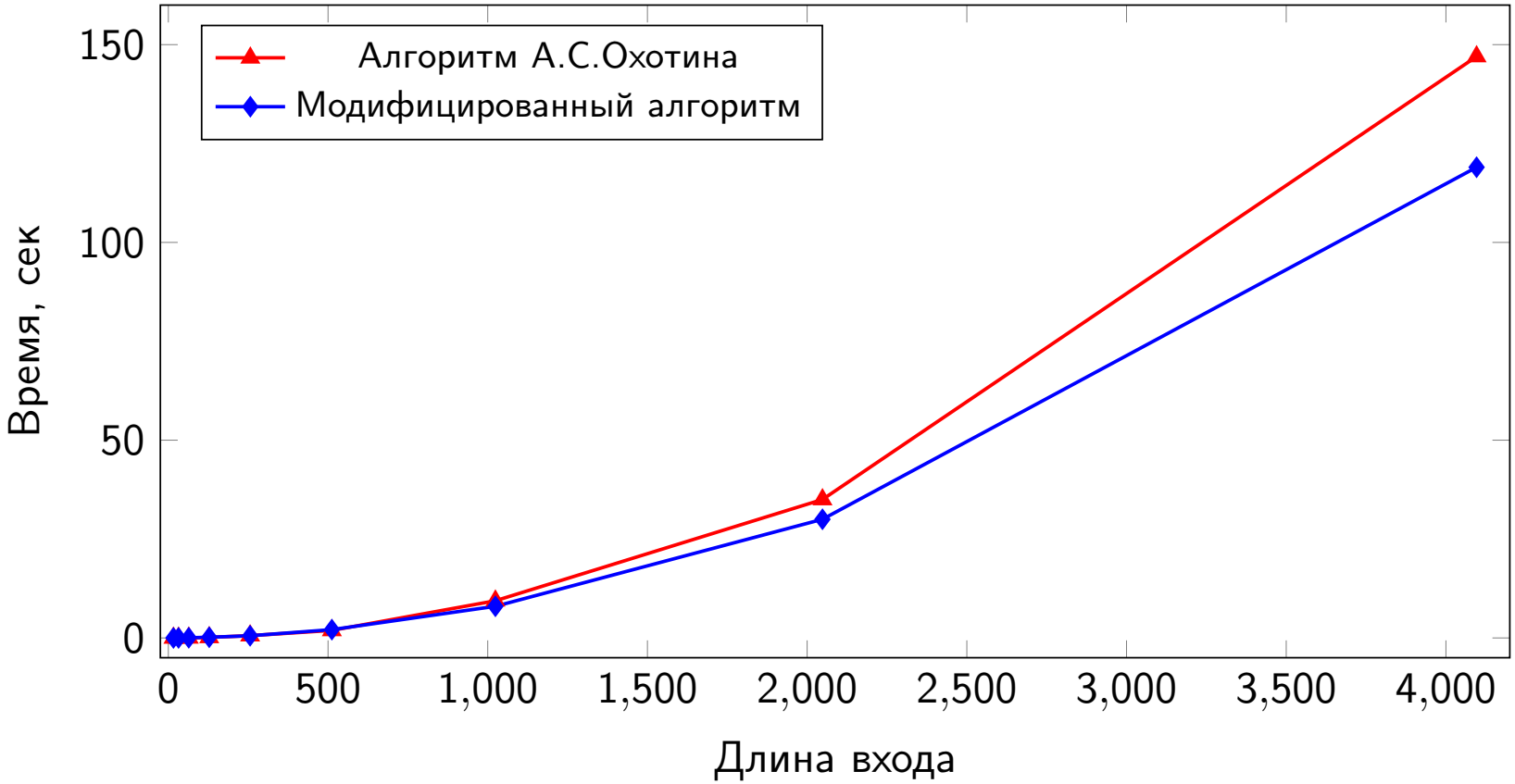
Оценка эффективности использования параллелизма

- **Количество процессоров:** $p = n - 2$
- **Критерии:**
  - Ускорение $S_p = \frac{T_0}{T_p} = log(n)$
  - Загруженность $E_p = \frac{S_p}{p} = \frac{log(n)}{p}$

Эксперименты: сравнительный анализ

- Сильно неоднозначная грамматика G1:

$$s \quad : \quad s\ s\ s \quad | \quad s\ s \quad | \quad B$$



Результаты

- Реализованы алгоритм А.С.Охотина и его модифицированная версия на языке программирования F# с использованием библиотеки для параллельных вычислений cuBLAS в рамках исследовательского проекта YaccConstructor
- Получена теоретическая оценка эффективности использования параллельных вычислений в модифицированном алгоритме: ускорение параллельной версии в сравнении с последовательной составляет $log(n)$, а загруженность — $\frac{log(n)}{n-2}$
- Проведен сравнительный анализ алгоритма А.С.Охотина и модифицированной версии, который показал что модификация показывает лучшие результаты на строках большой длины