Оптимизация алгоритмов синтаксического анализа, основанных на матричных операциях

Автор: Сусанина Юлия Алексеевна, 444
Научный руководитель: доцент, к.ф.-м.н. Григорьев С.В.
Рецензент: научный координатор Центра Компьютерных Наук TUCS Бараш М.Л.

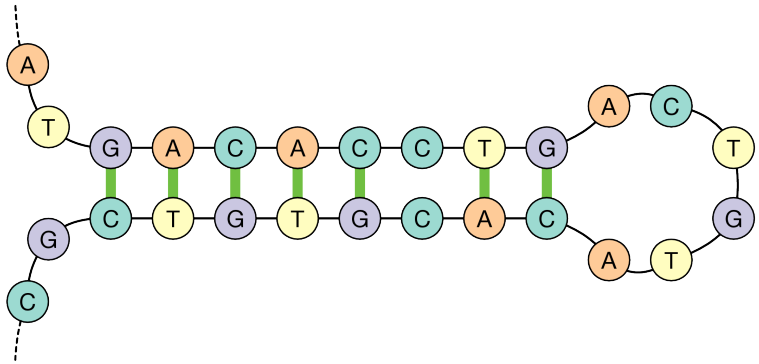
Санкт-Петербургский государственный университет
Кафедра системного программирования

24 мая 2019г.

Введение

- **Синтаксический анализ**
- **Область применения:** биоинформатика
  - Первичная структура

  - Вторичная структура



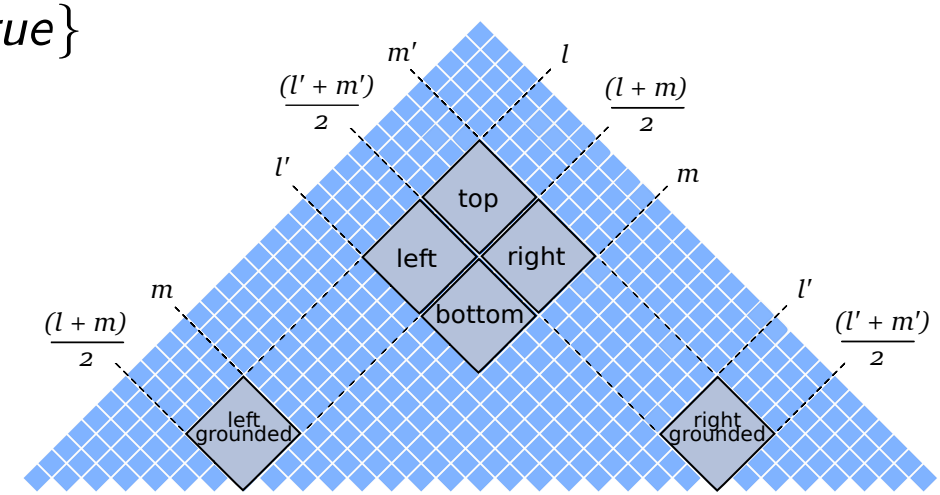
  - Особенности вторичной структуры существенны в задачах классификации и распознавания
  - Вид вторичной структуры можно задать с помощью контекстно-свободной грамматики
- **Задача:** поиск подстрок генетических цепочек, которые сворачиваются во вторичную структуру определенного вида

Алгоритмы синтаксического анализа

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

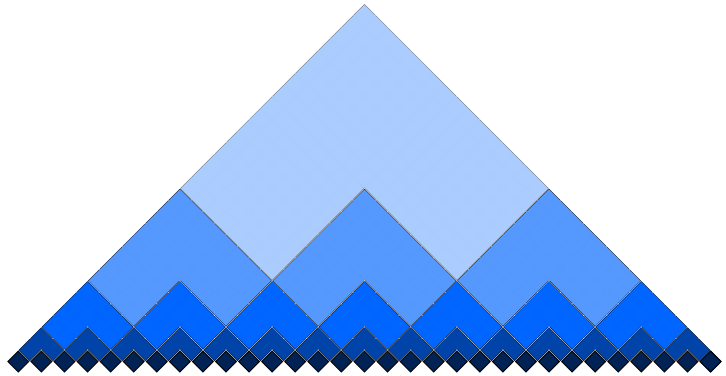
Алгоритм Валианта

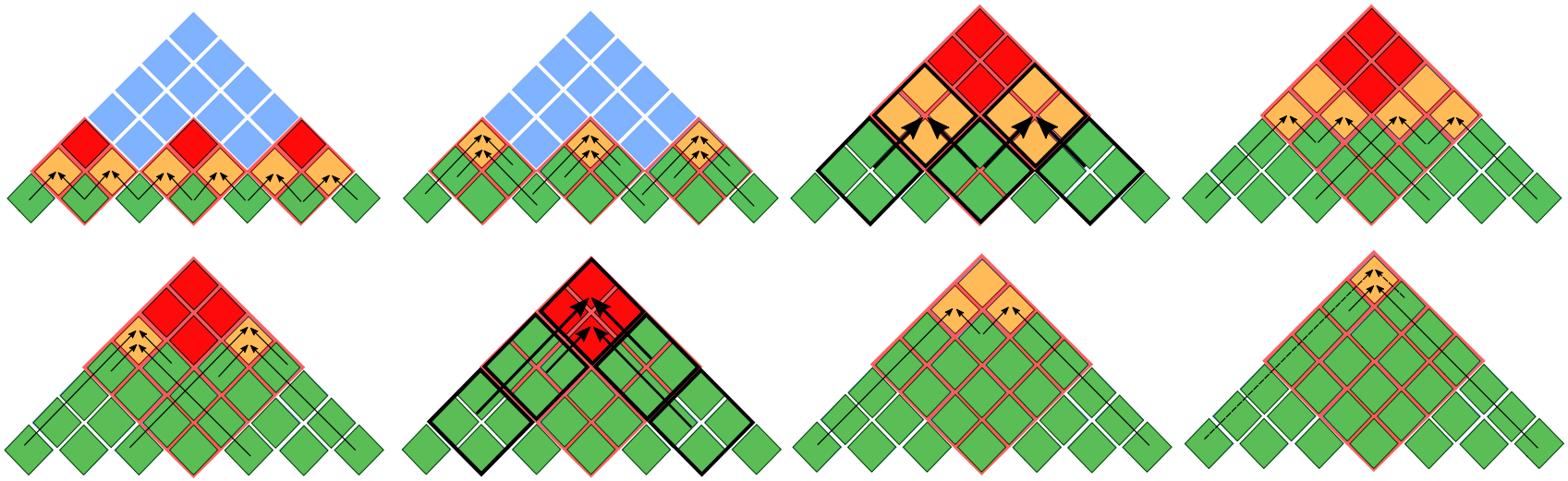
- Разбиение исходной матрицы и перемножение подматриц меньшего размера
- Возможность расширения для других классов грамматик (конъюнктивных, булевых)
- Вычислительная сложность — $\mathcal{O}(|G|BMM(n)\log n)$
  - Матрица — нетерминал
  - $T[i,j] = \{A \in N | T_A[i,j] = true\}$



Алгоритм Явейн

- Реогранизация вычислений
- Возможность разбиения на слои подматриц
- Использование параллелизма на уровне перемножения подматриц слоя





Постановка задачи

Цель: исследование алгоритма Явейн, являющегося модификацией алгоритма Валианта.

Для достижения данной цели были поставлены следующие задачи:

- Доказать корректность алгоритма Явейн и дать оценку сложности
- Проанализировать эффективность применения этого алгоритма и алгоритма Валианта к задаче поиска подстрок
- Реализовать последовательную и параллельную версии алгоритмов Валианта и Явейн
- Выполнить экспериментальное исследование алгоритма Явейн

Корректность алгоритма

Теорема (корректность алгоритма)

Алгоритм Явейн корректно заполняет $T_{i,j}$ для всех i и j, и входная строка $a = a_1 a_2 \dots a_n \in L_G(S)$ тогда и только тогда, когда $S \in T_{0,n}$.

Оценка сложности алгоритма

Лемма

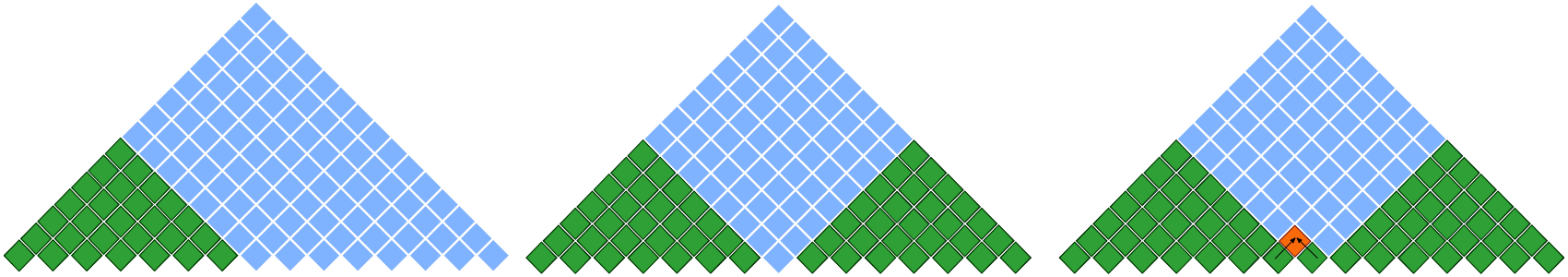
Для всех $i \in \{1, .., p-1\}$ *матрицы размера* $2^{p-i} \times 2^{p-i}$ *перемножаются ровно* $2^{2i-1} - 2^i$ *раз, где* $n = 2^p - 1$ *— длина входной строки.*

Теорема (оценка сложность алгоритма)

Пусть $|G|$ *— длина описания грамматики и* n *— длина входной строки. Тогда алгоритм Явейн заполняет матрицу* T *за* $\mathcal{O}(|G|BMM(n)\log n)$, *где* $BMM(n)$ *— время, необходимое для перемножения двух булевых матриц размера* $n \times n$.

Применимость к задаче поиска подстрок

- **Задача:** для входной строки размера $n = 2^p - 1$ найти все подстроки размера s, которые принадлежат языку, заданному грамматикой
- **Алгоритм Валианта:** необходимо полностью вычислить, как минимум, две треугольные подматрицы размера $\frac{n}{2}$
  $\mathcal{O}(|G|BMM(2^{p-1})(p-2))$



- **Алгоритм Явейн:** посчитать слои подматриц, размер которых не превышает 2^r, где $2^{r-2} < s \leq 2^{r-1}$
  $\mathcal{O}(|G|2^{2(p-r)-1}BMM(2^r)(r-1))$

Реализация

- Реализация алгоритмов Валианта и Явейн
- Последовательная версия:
  - Библиотека M4RI — метод "четырех русских"
- Параллельная версия:
  - CUDA C

Постановка экспериментов

- Грамматика D2

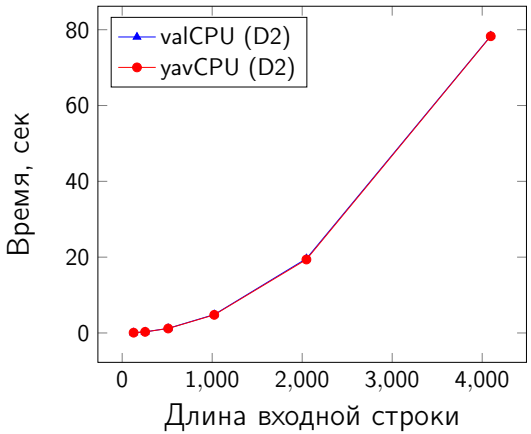
```
s: s s | ( s ) | [ s ] | ε
```

- Грамматика BIO

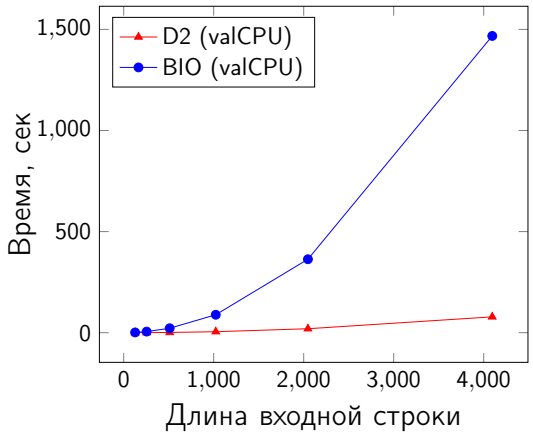
```
s1: stem<s0>
any_str: any_smb*[2..10]
s0: any_str | any_str stem<s0> s0
any_smb: A | T | C | G
stem1<s>: A s T | G s C | T s A | C s G
stem2<s>: stem1<stem1<s>>
stem<s>:
      A stem<s> T
    | T stem<s> A
    | C stem<s> G
    | G stem<s> C
    | stem1<stem2<s>>
```

Эксперименты: сравнительный анализ

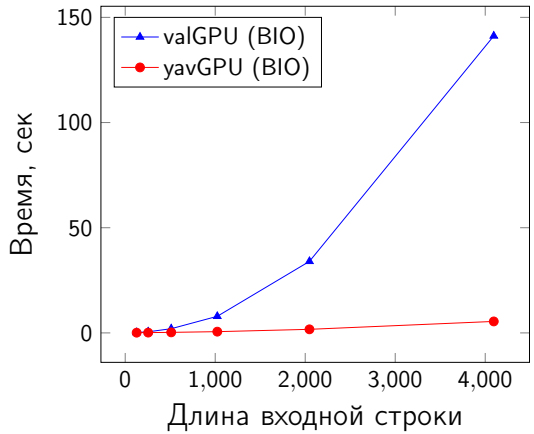
N	Время (сек)							
	Грамматика *D2*				Грамматика *BIO*			
	valCPU	yavCPU	valGPU	yavGPU	valCPU	yavCPU	valGPU	yavGPU
127	0.08	0.08	0.20	0.10	1.35	1.34	0.19	0.10
255	0.28	0.30	0.52	0.13	5.40	5.50	0.53	0.14
511	1.21	1.18	1.90	0.25	21.97	22.35	1.99	0.26
1023	4.90	4.78	7.88	0.54	88.70	90.32	7.89	0.60
2047	19.61	19.38	33.50	1.50	363.32	374.20	34.01	1.70
4095	78.36	78.28	140.47	4.45	1467.68	1480.59	141.10	5.47
8191	315.67	315.08	-	13.65	-	-	-	18.04



(a)



(b)



(c)

Эксперименты: поиск подстрок

s	N	Время (сек)			
		valCPU	yavCPU	valGPU	yavGPU
250	1023	4.90	3.00	7.88	0.24
	2047	19.61	6.65	33.50	0.26
	4095	78.36	13.83	140.47	0.32
	8191	315.67	28.90	-	0.46
510	2047	19.61	12.18	33.50	0.58
	4095	78.36	26.58	140.47	0.65
	8191	315.67	56.70	-	0.88
1020	4095	78.36	48.31	140.47	1.59
	8191	315.67	108.38	-	1.95
2040	8191	315.67	197.32	-	5.10

Результаты

- Доказана корректность алгоритма Явейн и дана оценка вычислительной сложности, которая составляет $\mathcal{O}(|G|BMM(n)log(n))$
- Проведен анализ, который показал, что алгоритм Явейн лучше применим к задаче поиска подстрок, чем оригинальная версия алгоритма
- Реализованы последовательная и параллельная версии алгоритмов Валианта и Явейн. Исходный код доступен в репозитории: https://github.com/SusaninaJulia/PBMM
- Проведено экспериментальное исследование алгоритма Явейн, показавшее его эффективность
- Результаты работы приняты к публикации в журнале «Труды ИСП РАН»