# Исследование применимости специализации алгоритма Витерби скрытой марковской моделью

**Автор**: Иван Владимирович Тюляндин, 19.M07-мм
**Руководитель**: к.ф.-м.н., доцент С.В. Григорьев
**Консультант**: к.ф.-м.н., ст. преп. СПбГУ Д.А. Березун
**Рецензент**: ст. преп. СПбПУ М.Х. Ахин

Санкт-Петербургский Государственный Университет
Кафедра системного программирования

29 апреля 2021

# Введение

- Алгоритмы методами линейной алгебры (ЛА) на больших данных
- Любые улучшения критичны
  - оборудование
  - изменение алгоритма
- Часть данных может быть зафиксирована
- Применение специализации

# Специализация

Два типа параметров

- статичные — т.е. зафиксированные
- динамические — все остальные

**Специализация**

Техника преобразования программ для оптимизации использования статических данных с целью уменьшить количество вычислений

Применение специализации к алгоритма методами ЛА не изучено

# Алгоритм Витерби

- применяется во многих областях
- выражается методами ЛА
- два параметра: скрытая марковская модель (СММ) и последовательность наблюдений
- на практике СММ зафиксирована, меняется только последовательность

# Цель и задачи

## Цель работы

Исследовать применимость специализации к алгоритму Витерби, который описан методами ЛА, при условии, что СММ является статическим параметром

- сделать обзор предметной области
  - рассмотреть алгоритм Витерби и его существующие реализации
  - описать технику специализации
- разработать специализированный алгоритм Витерби, реализовать и протестировать корректность реализации
- провести эксперименты по сравнению производительности специализированного алгоритма с неспециализированной версией и существующими реализациями

# Скрытая марковская модель

## Скрытая марковская модель

Вероятностный детерминированный автомат, каждое состояние которого создает наблюдение

- $S_{1..N}$ — $N$ состояний
- $O_{1..K}$ — $K$ возможных наблюдений
- $B_{1..N}$ — вероятности для состояний $S_{1..N}$ быть стартовыми
- $T_{1..N,1..N}$ — матрица переходов, $T_{i,j}$ — это вероятность перехода из $S_i$ в $S_j$
- $E_{1..N,1..K}$ — матрица наблюдений, $E_{i,j}$ — это вероятность создать наблюдение $O_j$ в состоянии $S_i$

Алгоритм Витерби, выраженный с помощью полукольца *Min-plus*

# Полукольцо *Min-plus*

- Переопределяем '+' как *min*, '*' как *plus*, $\infty$ и 0 как нейтральные элементы

  Пример матричного умножения

$$\begin{pmatrix} 0 & 1 \\ +\infty & 2 \end{pmatrix} \times \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} min(0+3, 1+4) \\ min(+\infty+3, 2+4) \end{pmatrix} = \begin{pmatrix} 3 \\ 6 \end{pmatrix}$$

- Для всех вероятностей $p$ из CMM:

$$t(p) = \begin{cases} p > 0: & -1 * log_2(p)) \\ p = 0: & +\infty \end{cases}$$

# Алгоритм Витерби методами ЛА

Для всех наблюдений $o$:

$$P(o) = \begin{pmatrix} t(E[1,o]) & \dots & +\infty \\ \vdots & \ddots & \vdots \\ +\infty & \dots & t(E[N,o]) \end{pmatrix}$$

## Алгоритм Витерби методами ЛА

Обработка первого наблюдения из последовательности $Obs$:

$$Probs_1 = P(Obs[1]) \times B$$

Оставшаяся часть $Obs$:

$$Probs_t = P(Obs[t]) \times T^\top \times Probs_{t-1}$$

# Алгоритм специализации

Как использовать данные CMM?

Для всех наблюдений $o$ можно вычислить:

$$P(o) \times B, \text{ далее как } PB(o)$$

$$P(o) \times T^{\top}, \text{ далее как } PT(o)$$

## Алгоритм Витерби первого уровня специализации

$$Probs_1 = PB(Obs[1])$$

$$Probs_t = PT(Obs[t]) \times Probs_{t-1}$$

# Алгоритм специализации

Матричное умножение ассоциативно

Обработка $o_1$ и $o_2$, если столбец $Probs_0$ известен?

$$\begin{aligned} Probs_2 &= PT(o_2) \times Probs_1 \\ &= PT(o_2) \times (PT(o_1) \times Probs_0) \\ &= (PT(o_2) \times PT(o_1)) \times Probs_0 \end{aligned} \tag{1}$$

Можно вычислить $PT(o_2) \times PT(o_1)$. Два наблюдения одним умножением, т.е. второй уровень!

Этот подход можно применить для повышения уровня

# Анализ операций

Пусть длина *Obs* обозначается как *lo*, количество состояний CMM как *N*, количество возможных наблюдений CMM как *K*

## Неспециализированная версия

Матричных умножений:

$$1 + 2 * (lo - 1)$$

## Специализированная версия уровня *M*

Матричных умножений:

$$(lo - 1)/M + (lo - 1) \ mod \ M$$

Дополнительной памяти: $K^M$ матриц, каждая $N \times N$

# Реализация

Выбор библиотек

- Часто СММ разреженные
- Нужно и для CPU, и для GPGPU

Библиотеки SuiteSparse:GraphBLAS и CUSP

Проведено тестирование

- на корректность реализации алгоритма Витерби
- на сохранение семантики
- отсутствие утечек памяти

# Эксперименты

- Ubuntu 20.04, Intel Core i7-4790 3.60 GHz, NVIDIA GeForce GTX 1030, 32 Gb RAM
- Сравнение специализированной версии против неспециализированной и CUDAMPF
- 24 СММ из репозитория CUDAMPF, три набора данных
- Медиана из 100 запусков

# Эксперименты

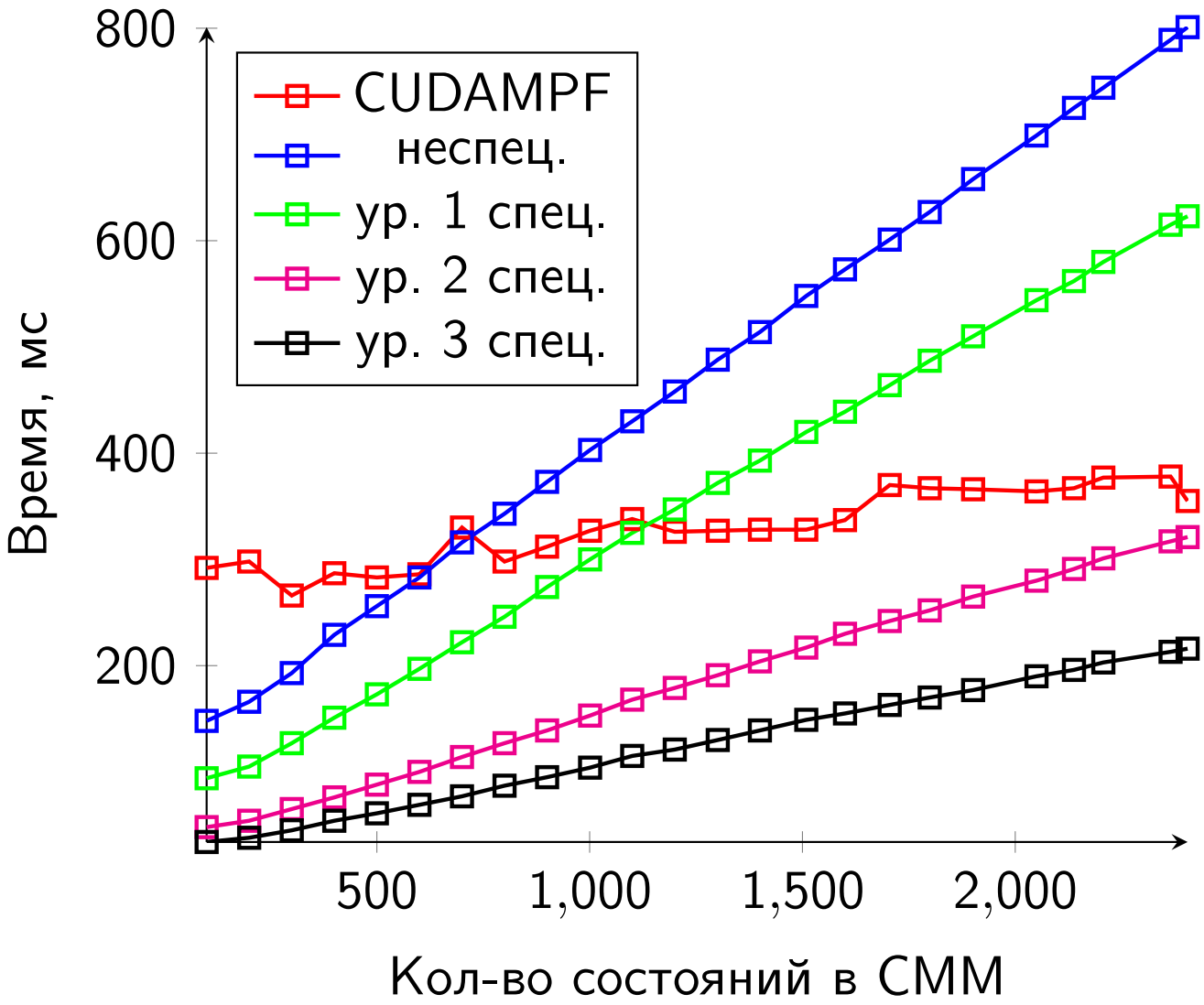



Рис. 1: 3 x 3500 наблюдений, меньше — лучше

# Эксперименты

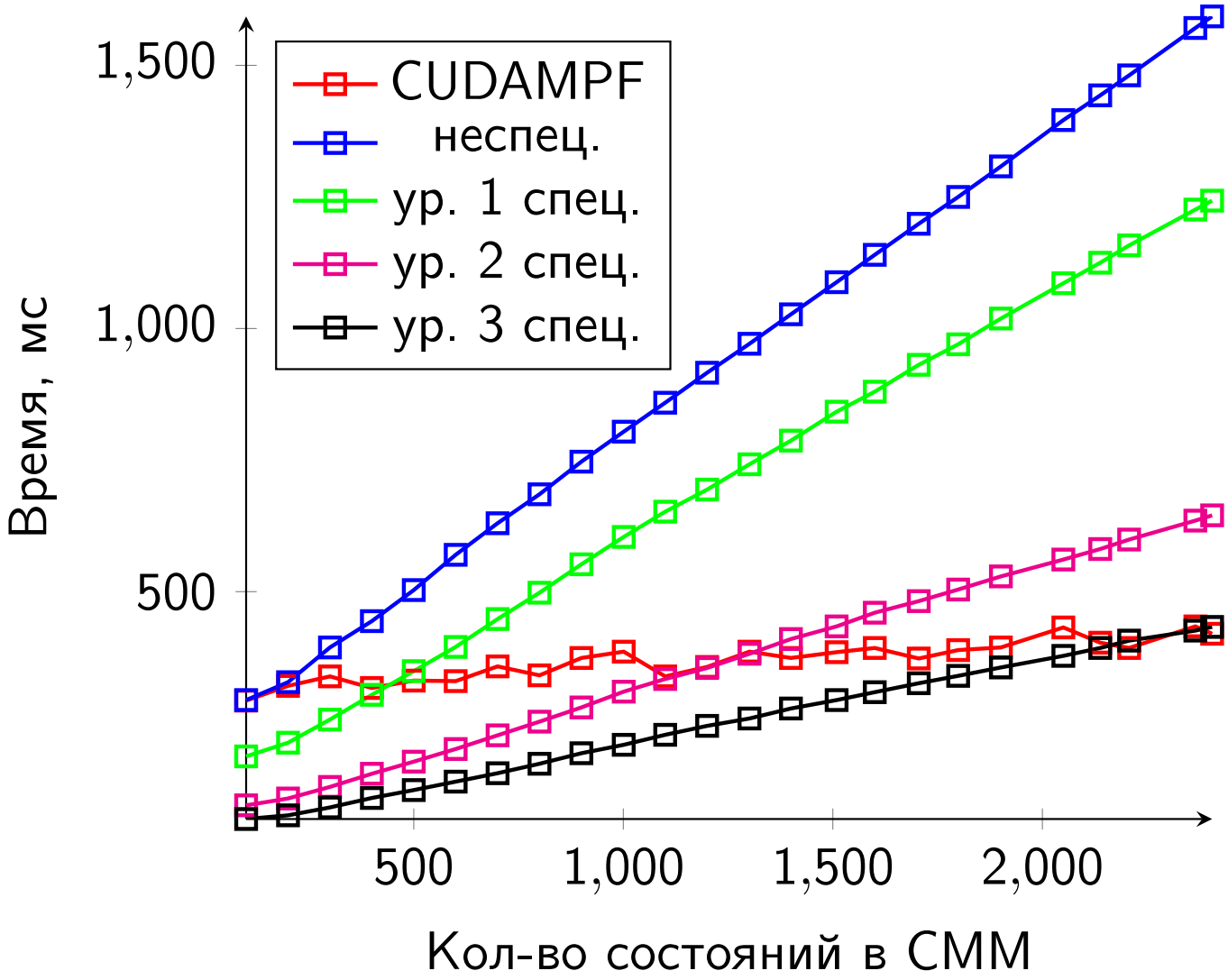



Рис. 2: 3 x 7000 наблюдений, меньше — лучше

# Эксперименты

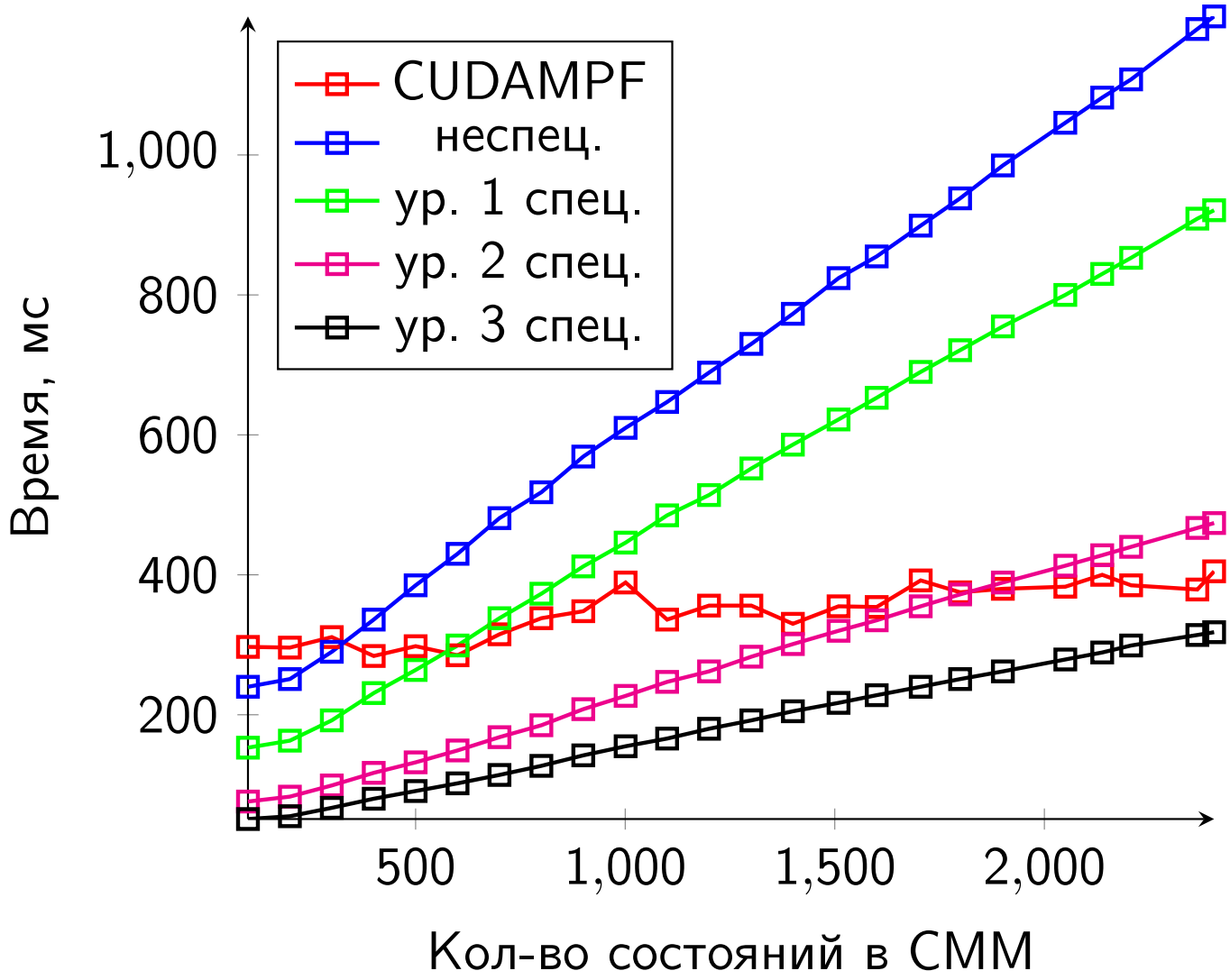



Рис. 3: 16 последовательностей из БД PFAM, меньше — лучше

# Эксперименты

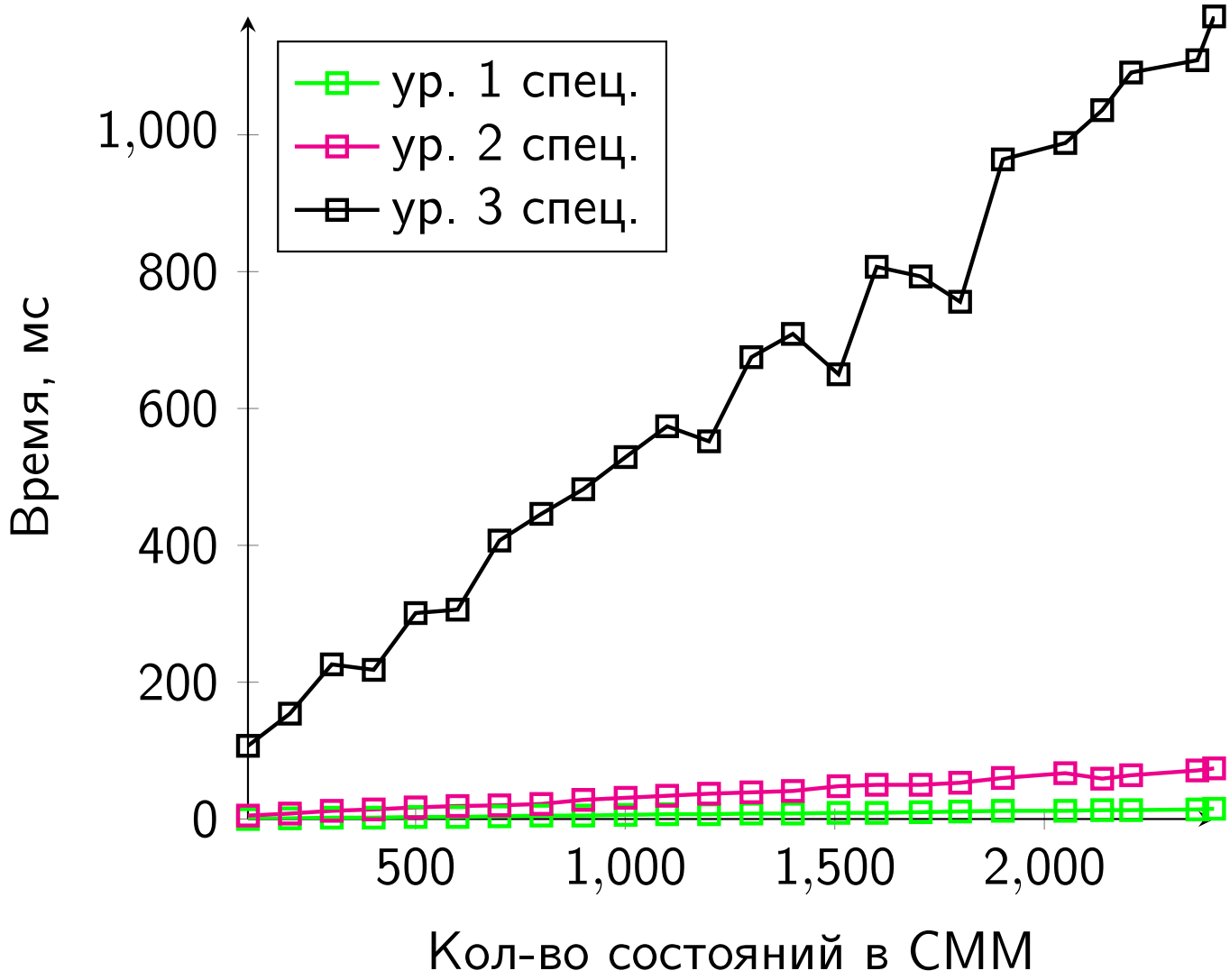



Рис. 4: Время, затраченное на специализацию, меньше — лучше

# Текущие результаты

- Выполнен обзор предметной области
- Реализованы и протестированы две реализации специализированного алгоритма Витерби
  - SuiteSparse:GraphBLAS для выполнения на CPU
  - CUSP для выполнения на GPGPU
- Проведены эксперименты на данных из репозитория CUDAMPF. Специализированная версия с использованием SuiteSparse:GraphBLAS в 1,5 раза производительнее CUDAMPF.

  SEIM 2021: статья *Viterbi Algorithm Specialization Using Linear Algebra*

  Необходимо провести эксперименты на CUSP и эксперименты на большем количестве последовательностей

# Пример специализации: возведение в степень

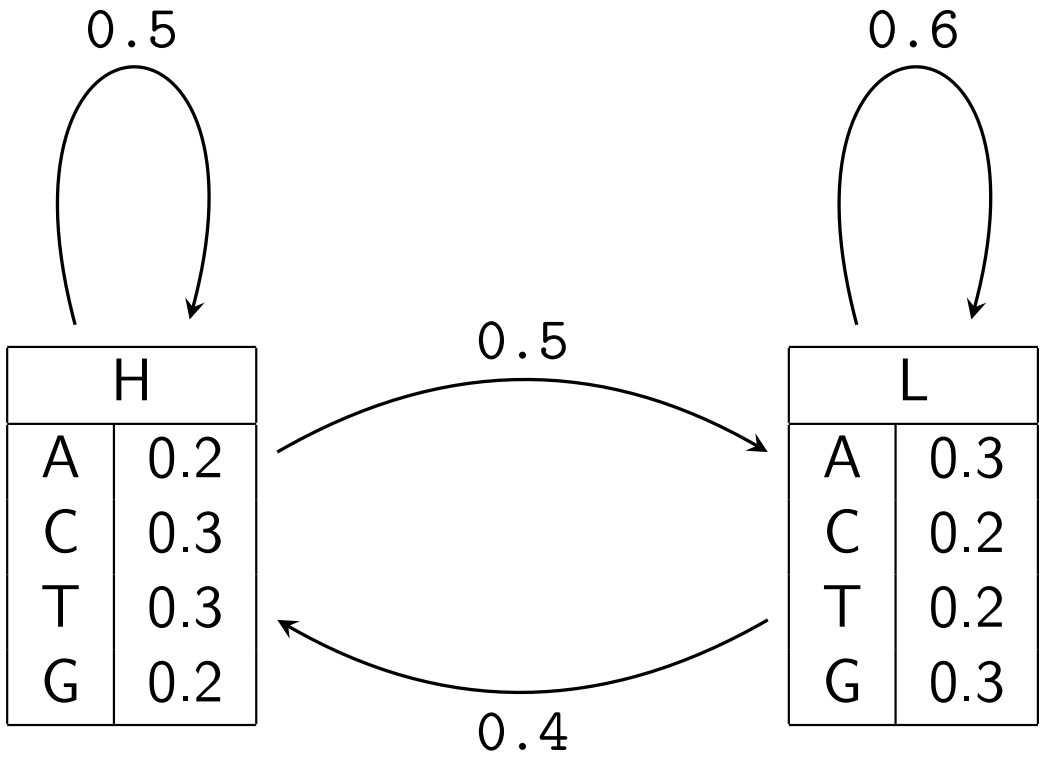
```
function f(x, n)
  if n == 0 then 1
  elif even(x) then f(x, n/2)^2
  else x * f(x, (n-1)/2)^2
```

Предположим $n = 5$, т.е. это статический параметр

```
function f_spec(x) = x * (x^2)^2
```

Специализация бесполезна, если $x$ статический параметр

# CMM: пример




Вероятности быть стартовым состоянием

| H | L |
|---|---|
| 0.5 | 0.5 |

# О повышении уровня

Обработка $o_{t-2}$, $o_{t-1}$ и $o_t$, если столбец $Probs_{t-3}$ известен?

$$Probs_t = PT(o_t) \times PT(o_{t-1}) \times PT(o_{t-2}) \times Probs_{t-3} \qquad (2)$$

Все произведения $PT(o)$ могут быть вычислены по данным из СММ

# Общее время выполнения

| | CUDAMPF | Initial | 1-level | 2-level | 3-level |
|---|---|---|---|---|---|
| 3 x 3500 | 7907 | 11365 | 8747 | **5332** | 18858 |
| 3 x 7000 | **8862** | 22641 | 17342 | 9630 | 21356 |
| Real world | 8347 | 17064 | 12956 | **7311** | 19581 |

Таблица 1: Общее время выполнения (специализация и выполнение специализированного алгоритма Витерби), мс