



# Синтаксический анализ для поиска в метагеномных сборках

**Автор:** Семён Григорьев

Лаборатория языковых инструментов JetBrains  
Санкт-Петербургский государственный университет  
Математико-механический факультет

11 мая 2016г.

- Исследования в области лексического и синтаксического анализа
- Открытый исходный код
  - ▶ <https://github.com/YaccConstructor>
- Основной язык разработки — F#

- REAGO — HMM
- Xander — HMM,
- EMIRGE — HMM
- Infernal — CM, линейный вход

# Вторичная структура РНК

- Несёт полезную информацию для идентификации подпоследовательностей
- Может быть задана с помощью грамматики

`stem<s> :`

```
    A stem<s> U | U stem<s> A
  | G stem<s> C | C stem<s> G
  | s
```

`folded: stem<A*[3..7]>`

# Грамматики для описания вторичной структуры tRNA

- Уточнение шаблона
  - ▶ Возможности языков описания грамматики: метаправила, повторения
  - ▶ Кыход за пределы КС-грамматик
- Далее на грамматику можно “навесить” вероятности и т.д.

# Синтаксический анализ метагеномной сборки

- Регулярное множество ( $R$ ) = конечный автомат ( $M$ ) = граф конечного автомата или просто граф ( $H$ )

# Синтаксический анализ метагеномной сборки

- Регулярное множество ( $R$ ) = конечный автомат ( $M$ ) = граф конечного автомата или просто граф ( $H$ )
- Грамматика  $G$  задаёт для цепочек свойство выводимости:  
 $S \Rightarrow_G^* \omega$
- Задача: найти все цепочки  $\omega \in R : S \Rightarrow_G^* \omega$
- Метагеномная сборка представима в виде графа, который можно рассматривать как КА, в котором все вершины стартовые

# Синтаксический анализ регулярных множеств

- Состояние синтаксического анализатора однозначно задаётся конечным набором данных — дескриптором
  - ▶  $s \rightarrow Ab \cdot c$  — “ситуация”, “слот”
  - ▶ Позиция во входе
  - ▶ ...
- Имея дескриптор можно продолжить разбор с описанного им места
- Дескрипторы переиспользуются: дальнейший разбор не зависит от истории дескриптора (контекстно-свободная грамматика)



# Синтаксический анализ регулярных множеств: детали

- Процесс анализа конечен
- Внутренние структуры аналогичны структурам, используемым в обобщённом синтаксическом анализе (GLR, GLL)
  - ▶ Структурированный в виде графа стек (GSS)
  - ▶ Сжатое представление леса разбора (SPPF)
  - ▶ Управляющие таблицы
- Полиномиальная сложность (?)

- $F^*$  (<https://www.fstar-lang.org/tutorial/>)
- Coq
- Agda
- ...

# Текущие результаты

- $F\# + F^* = F\#^*$
- Парсер для  $F\#^*$
- Транслятор из AST  $F\#$  в AST  $F^*$
- Разбивается на подзадачи (до 3 человек)
- Диплом (вся задача), публикации

- Поддержать в модели проекта, редакторе, отладчике
  - ▶ Создание файлов, шаблоны
  - ▶ Подсветка синтаксиса
  - ▶ Сообщения об ошибках, подсветка ошибок
  - ▶ ....
- Разбивается на подзадачи (до 4 человек)
- Диплом (вся задача), публикации

- Почта: `rsdpisuy@gmail.com`
- Исходный код YaccConstructor:  
`https://github.com/YaccConstructor`
- Google+ сообщество: `https://goo.gl/DuPWkM`