

Реализация алгоритма Кока-Янгера-Касами (СҮК)



АВТОР: СУСАНИНА ЮЛИЯ АЛЕКСЕЕВНА, 143 ГРУППА

РУКОВОДИТЕЛЬ: ст.пр. ГРИГОРЬЕВ С. В.

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
20.05.2016

Введение



Алгоритм Кока-Янгера-Касами, или алгоритм СҮК – это алгоритм, позволяющий определить возможен ли вывод строки в заданной грамматике в нормальной форме Хомского. Другими словами, это алгоритм синтаксического анализа строки.

Алгоритм реализует синтаксический анализ снизу-вверх и основывается на методе динамического программирования.

Цели и задачи



Цель работы: реализовать алгоритм СУК (простая последовательная и параллельная реализации)

Задачи:

- Изучение алгоритма СУК и принципа его работы
- Реализация алгоритма на F#
- Параллельная реализация алгоритма на F#
- Создание тестов и проверка корректности работы данного алгоритма

Алгоритм Кока-Янгера-Касами



Данный алгоритм работает с контекстно-свободными грамматиками, имеющими нормальную форму Хомского.

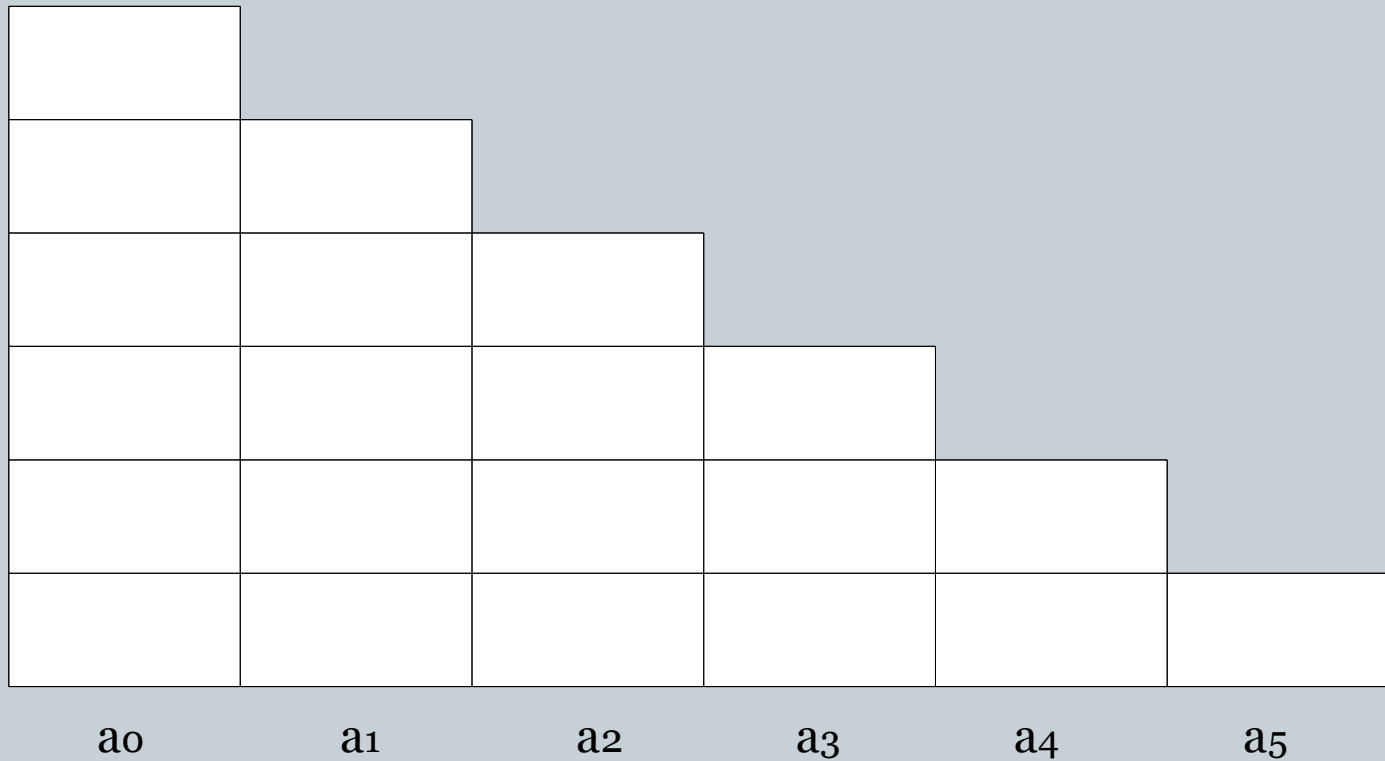
Нормальная форма Хомского требует от грамматики, чтобы каждая ее продукция выглядела одним из трех способов :

1. $S \rightarrow AB$, где S – стартовый нетерминальный символ,
2. $A \rightarrow a$ A, B – нетерминальные символы,
3. $S \rightarrow \lambda$ a – терминал, λ – пустая строка

Алгоритм Кока-Янгера-Касами



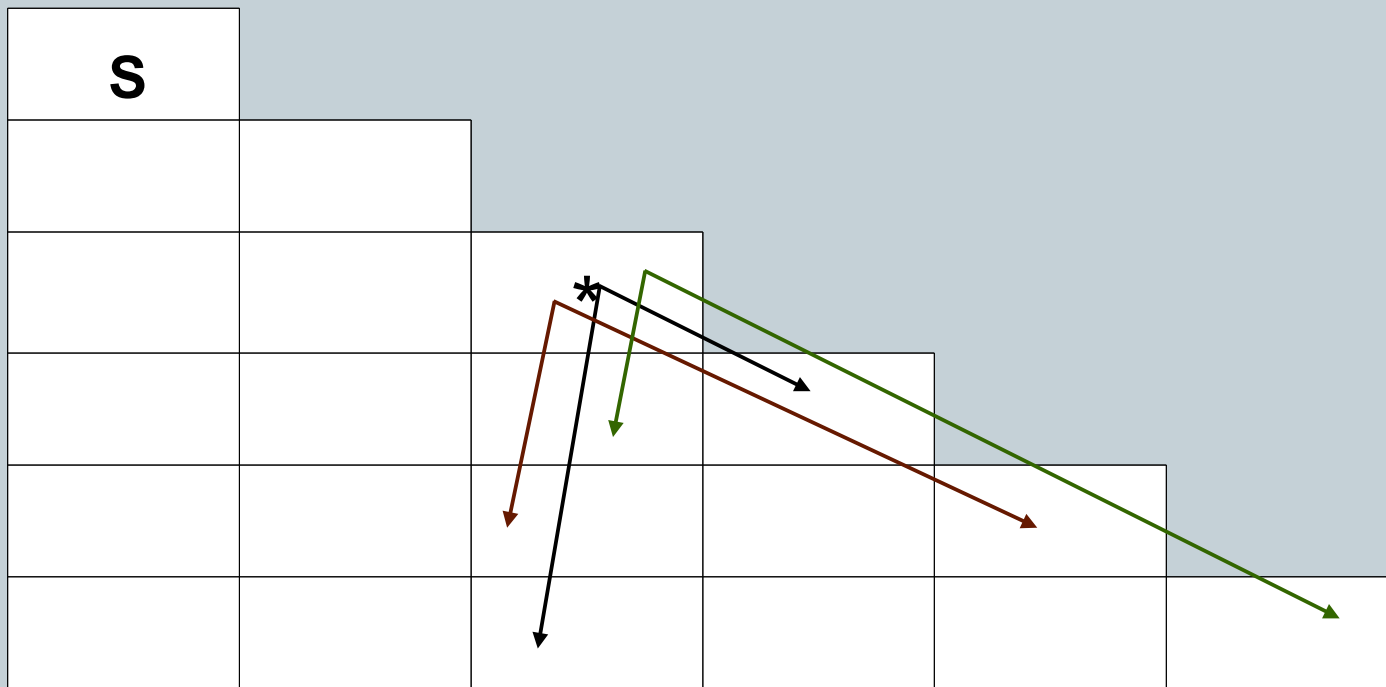
Для применения алгоритма к входной строке (a_1, \dots, a_n) строится треугольная матрица размера $n \times n$, где n – длина входной строки.



Алгоритм Кока-Янгера-Касами



Если после заполнения всех элементов матрицы, если в верхнем левом углу стоит стартовый нетерминал, то строка выводится из данной грамматики.



Реализация алгоритма



На языке программирования F#:

- Простая последовательная реализация
- Параллельная реализация (с использованием класса Parallel)

```
[| [| [| |""; "A"; "S"|]; [|""|]; [|""|]; [|""|]; [|""|] |];  
 [| [|""; "A"; "S"|]; [|""; "A"; "C"|]; [|""|]; [|""|]; [|""|] |];  
 [| [|""; "A"|]; [|""; "C"|]; [|""; "S"|]; [|""|]; [|""|] |];  
 [| [|""; "C"; "S"|]; [|""; "C"|]; [|""; "S"|]; [|""; "B"; "C"|]; [|""|] |];  
 [| [| "A"; "C"|]; [| "B"|]; [| "A"|]; [| "B"|]; [| "A"; "C"|] |] |]  
The string can be generated.
```

Пример выполнения программы для строки “cbabc” и грамматики

$(S \rightarrow AB; A \rightarrow CC; B \rightarrow BC; C \rightarrow CB; C \rightarrow BA; B \rightarrow b; A \rightarrow a; A \rightarrow c; C \rightarrow c)$

Тестирование



Создание тестов для проверки корректности работы алгоритма:

- Для строк, выводимых из заданной грамматики
- Для строк, невыводимых из заданной грамматики
- Случаев, когда пользователь вводит пустую строку

Результаты



- Изучен алгоритм СУК и принцип его работы
- Реализован алгоритм на F#
- Реализована параллельная версия алгоритма на F#
- Созданы тесты и проведена проверка корректности работы данного алгоритма