Context-free recognition via shortest paths computation

Григорий Волков, 371

Задача

$\omega = a_1 a_2 \ldots a_n$ - некоторое слово

G - контекстно-свободная грамматика

$\omega \overset{?}{\in} L(G)$

Решетчатый граф (Lattice graph)

$$\mathrm{L} = (\mathrm{V}, \mathrm{E}, \mathrm{s})$$

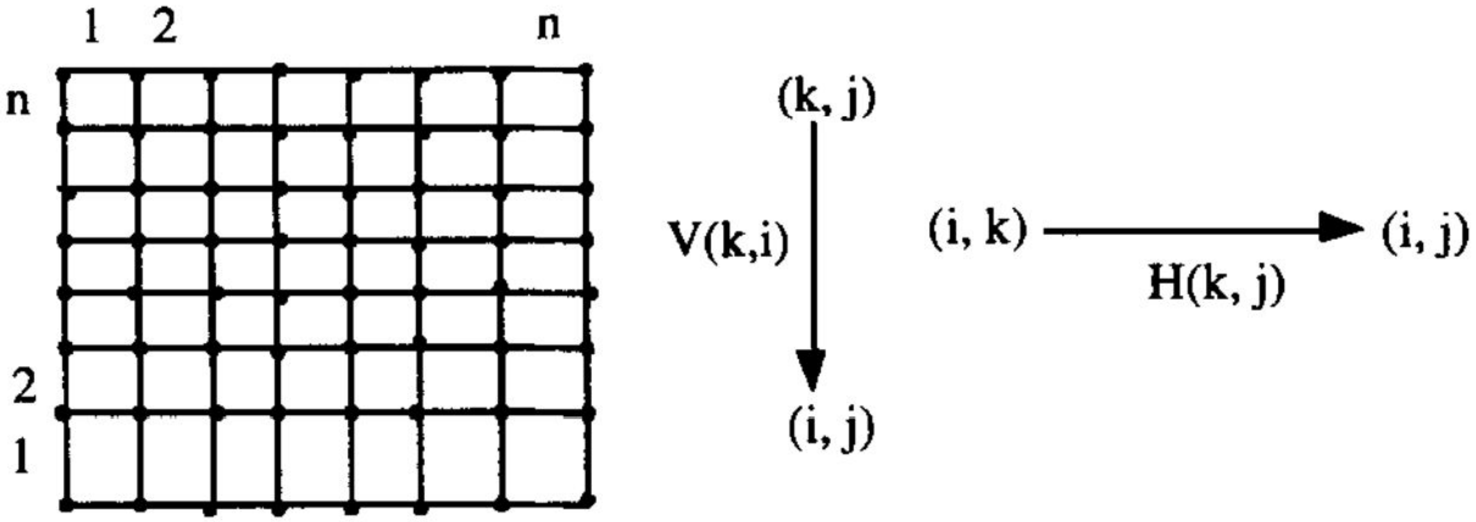
$$\mathrm{V} = \{(\mathrm{i}, \mathrm{j})\colon 1 \leq i, j \leq n\} \;\; \mathrm{s} \in V$$

$$\mathrm{E}_H = \{(i, j) \to (i, j+k) : 1 \leq i, j, j+k \leq n, k > 0\}$$

$$\mathrm{E}_V = \{(i, j) \to (i-k, j) : 1 \leq i, j, i-k \leq n, k > 0\}$$

$$\mathrm{E} = \mathrm{E}_H \cup E_V$$

Решетчатый граф (Lattice graph)



Strongly congruent ребра

Ребра π_1 и π_2 - strongly congruent

$$\Updownarrow$$

Оба имеют одинаковую длину и
1. Либо оба горизонтальные и имеют начало в одном столбце
2. Либо оба вертикальные и имеют начало в одной строке

Например, ребра $(4,7) \to (4,15)$ и $(8,7) \to (8,15)$ - strongly congruent
Еще один пример: ребра $(3,5) \to (10,5)$ и $(3,8) \to (10,8)$

Матрицы весов

$$H(k,j) = weight((i,k) \to (i,j))$$
$$V(k,i) = weight((k,j) \to (i,j))$$

Заметим, что количество ребер в графе: $O(n^3)$,
в то время, как его представление в виде (H, V, s)
имеет размер $O(n^2)$

Полукольцо весов

Определим на множестве весов ребер полукольцо с двумя операциями операциями: $\otimes$ и $+$

Прим. пустой путь - нулевой элемент полукольца

Shortest path

$$DIST(x) = \sum weight(\mu),$$

где суммируются все непустые пути μ из s в x

Single edge extend

$$\Psi_v(i,j) = \sum_k \Psi(k,j) \otimes weight((k,j) \to (i,j)) = \sum_k \Psi(k,j) \otimes V(k,i),$$

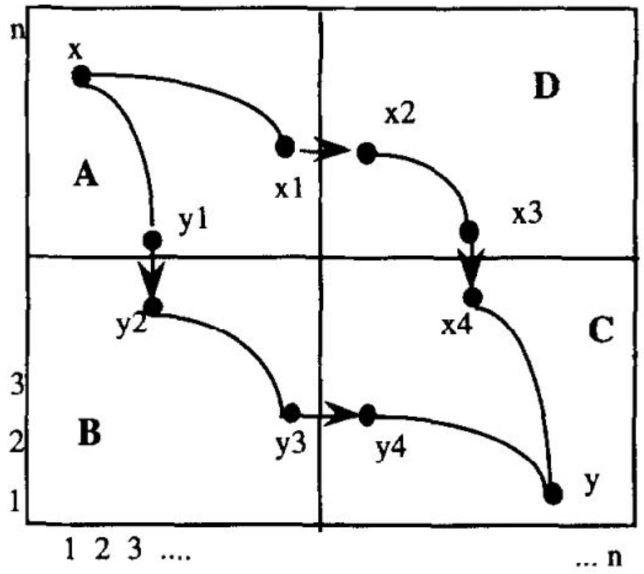
$$\Psi_h(i,j) = \sum_k \Psi(i,k) \otimes weight((i,k) \to (i,j)) = \sum_k \Psi(i,k) \otimes H(k,j).$$

The total change of Ψ corresponds to the following operation on matrices:

$$\Psi := \Psi + \Psi_v + \Psi_h := \Psi + (\Psi^{\mathrm{T}} \otimes V)^{\mathrm{T}} + \Psi \otimes H,$$

Shortest path алгоритм

procedure *ShortestPaths*(L);
  initially $DIST[v] = \emptyset$ for each node v;
if size(L) is small **then**
  compute *ShortestPaths*(L) in a constant time **else**
begin
  1: *ShortestPaths*(A);
  2: *SingleEdgeExtend*($DIST$);
  3: *ShortestPaths*(B); *ShortestPaths*(D);
  4: *SingleEdgeExtend*($DIST$);
  5: *ShortestPaths*(C);
end;



Main algorithm

1. Будем рассматривать CFG в нормальной форме Хомского
2. Назовем элементы множества $\{(A, i, j) : A \in V_N, 1 \leq i \leq j \leq n\}$ $items$
3. Некоторый item (A, i, j) считается валидным $\Leftrightarrow$ подстрока $a_i a_{i+1} ... a_j$ выводится из нетерминала A

Main algorithm

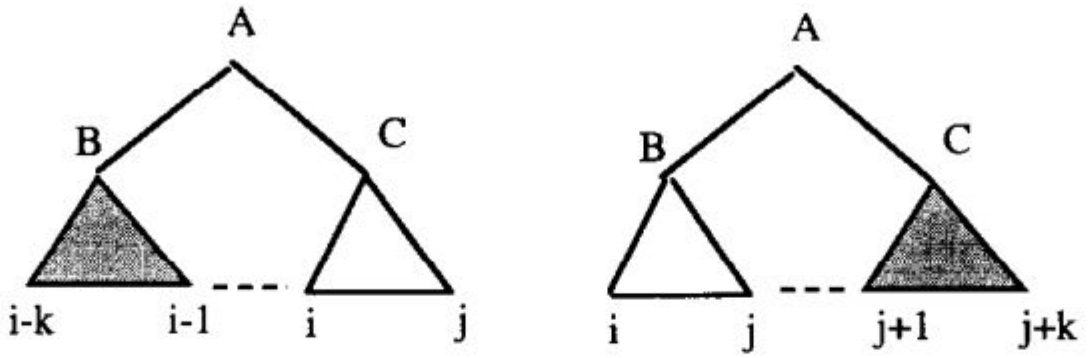
Обозначим $VALID(k, l)$ - множество валидных $items$ таких, что $k \leq i \leq j \leq l$

$\omega \in L(G) \iff (S, 1, n) \in VALID(1, n)$

Main algorithm

Предположим, что множество
$\Pi = VALID(1, n/2) \cup VALID(n/2+1, n)$ уже вычислено

Определим на множестве *items* импликацию:
$(a)\,(C, i, j) \Rightarrow (A, i-k, j) \iff$ существует правило $A \to BC$ и $(B, i-k, i-1) \in \Pi$
$(b)\,(B, i, j) \Rightarrow (A, i, j+k) \iff$ существует правило $A \to BC$ и $(C, j+1, j+k) \in \Pi$



(a) (b)

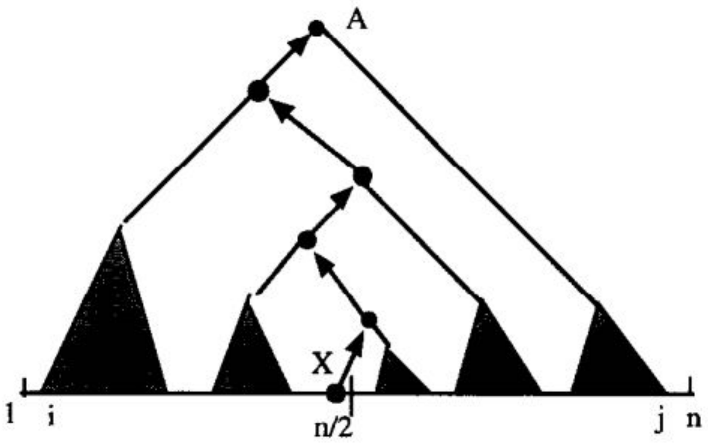
Main algorithm

$\Rightarrow^*$ - транзитивное замыкание $\Rightarrow$

$\omega = a_1 a_2 ... a_n$ - входное слово

Определим множество

$IMPLIED = \{(A, i, j) : (X, n/2, n/2) \Rightarrow^* (A, i, j) \; and \; X \rightarrow a_{n/2}\}$



Main algorithm

Вес ребра $(i, j) \to (i', j')$ - бинарное отношение R:

$(X, A) \in R \iff (X, i, j) \Rightarrow (A, i', j')$

$DIST$ - матрица, весов путей из $s \in V$ в любую другую вершину

В качестве вершины s возьмем $(n/2, n/2)$

Main algorithm

function $VALID(i, j)$;
$m := j - i + 1$ {m is a power of two}
if m is small **then**
compute $VALID(i, j)$ in a constant time **else**

begin
$\Pi := VALID(i, m/2) \cup VALID(m/2 + 1, j)$;
construct the lattice graph L;
compute the table $DIST$ by the *shortest paths algorithm*;
compute the set $IMPLIED$ {by Lemma 2(b)}
return $VALID(i, j) = \Pi \cup IMPLIED$.
end;

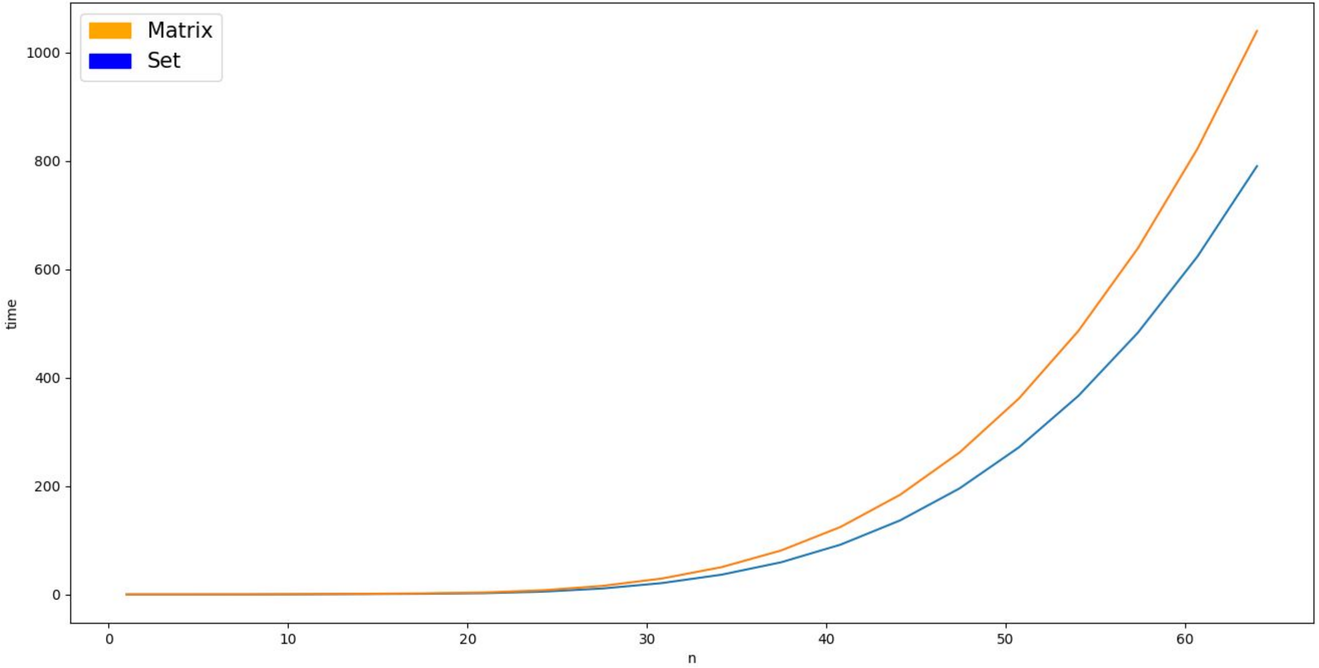
Практические результаты

Предметом изучения была скорость работы алгоритма

Тестирование проводилось для грамматики языка $a^n b^n$:

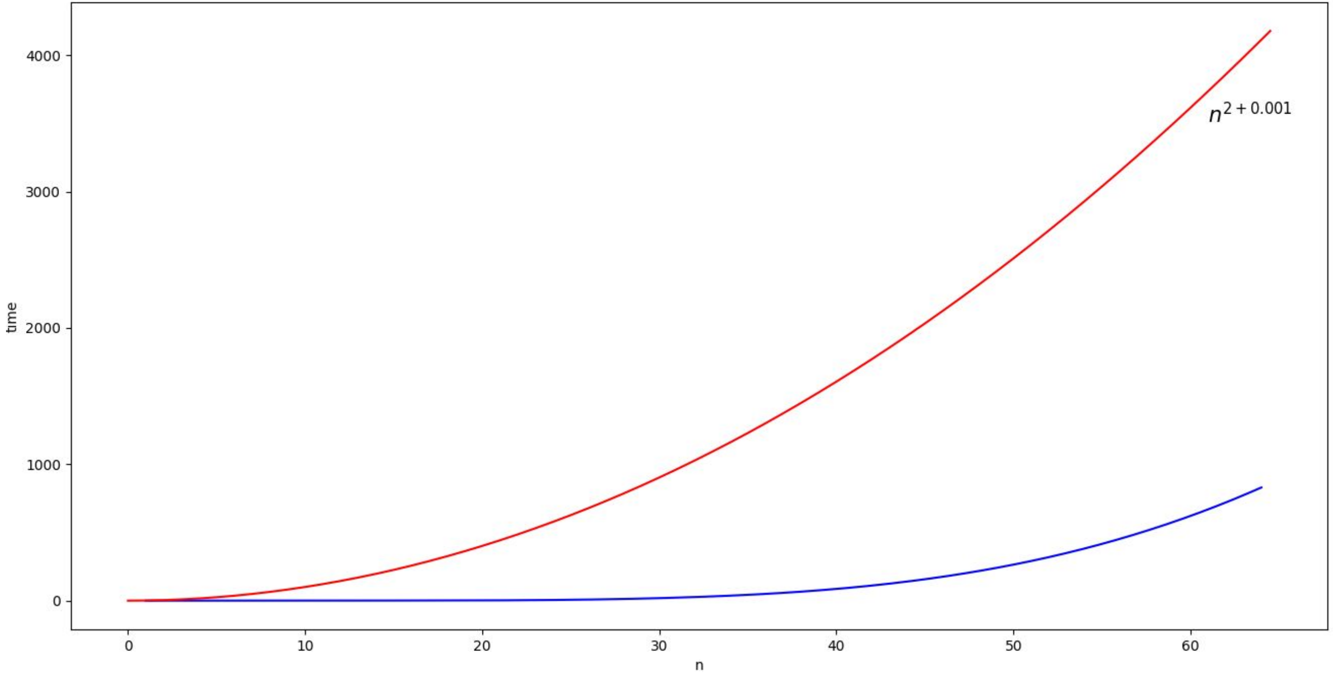
- $S \to AM$
- $S \to AB$
- $M \to SB$
- $A \to a$
- $B \to b$

Практические результаты



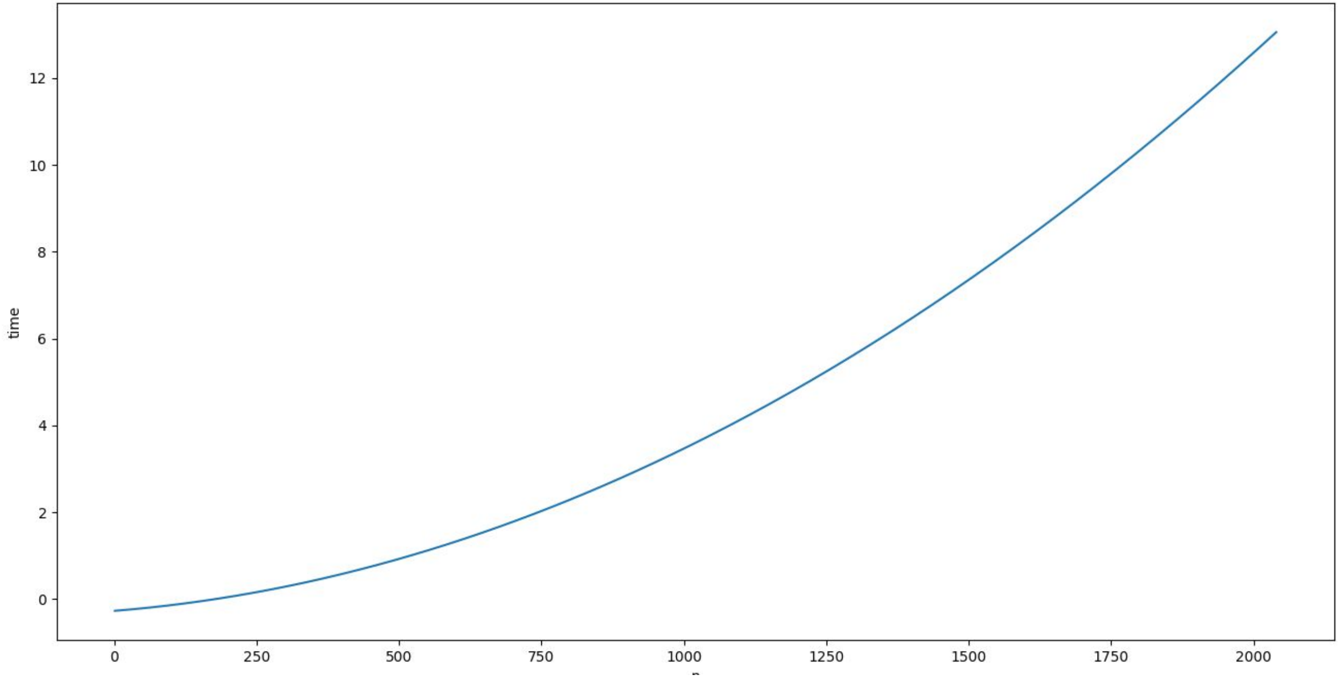
Зависимость времени выполнения от различных способов хранения отношения

Практические результаты



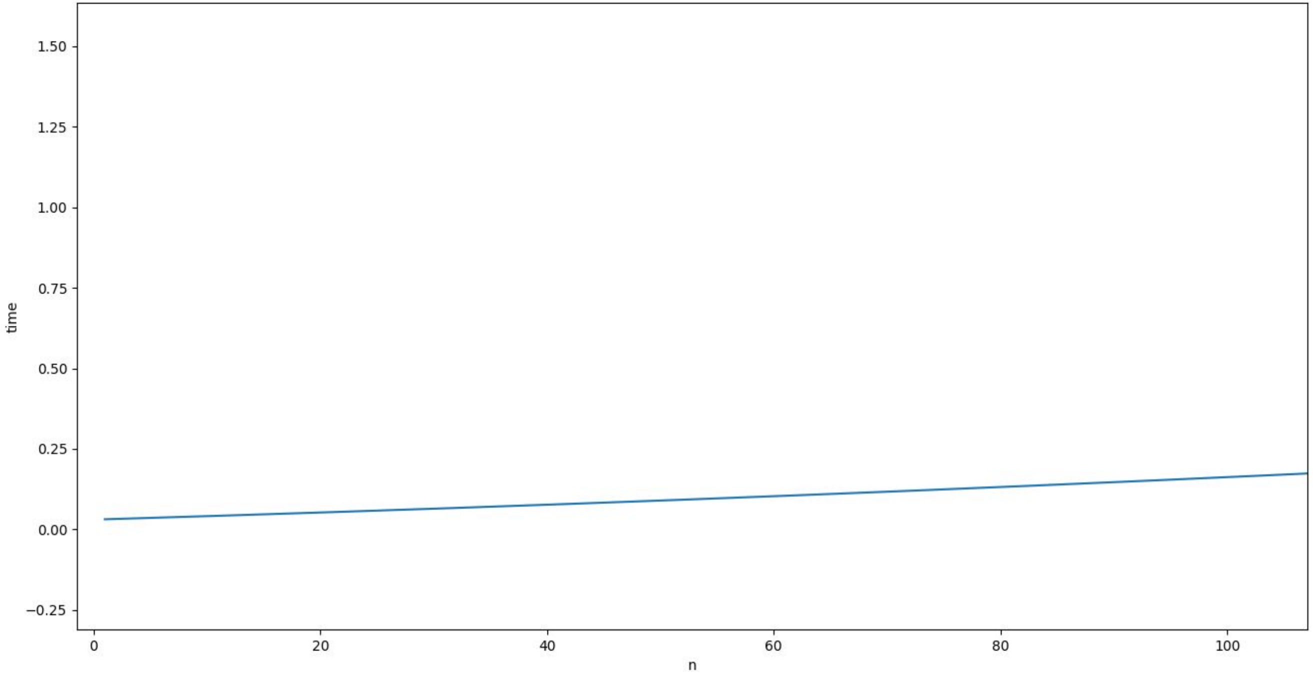
Сравнение времени выполнения с заявленной в статье сложностью

Практические результаты



Время работы метода, использующего алгоритм Флойда–Уоршелла

Практические результаты



Время работы метода, использующего алгоритм Флойда–Уоршелла