Анализ символьных последовательности различной языковой природы

Мирошниченко Любовь Александровна
Институт математики СО РАН
luba@math.nsc.ru

Объект исследования: символьные последовательности различной языковой природы.

Σ – непустое конечное множество символов (алфавит);

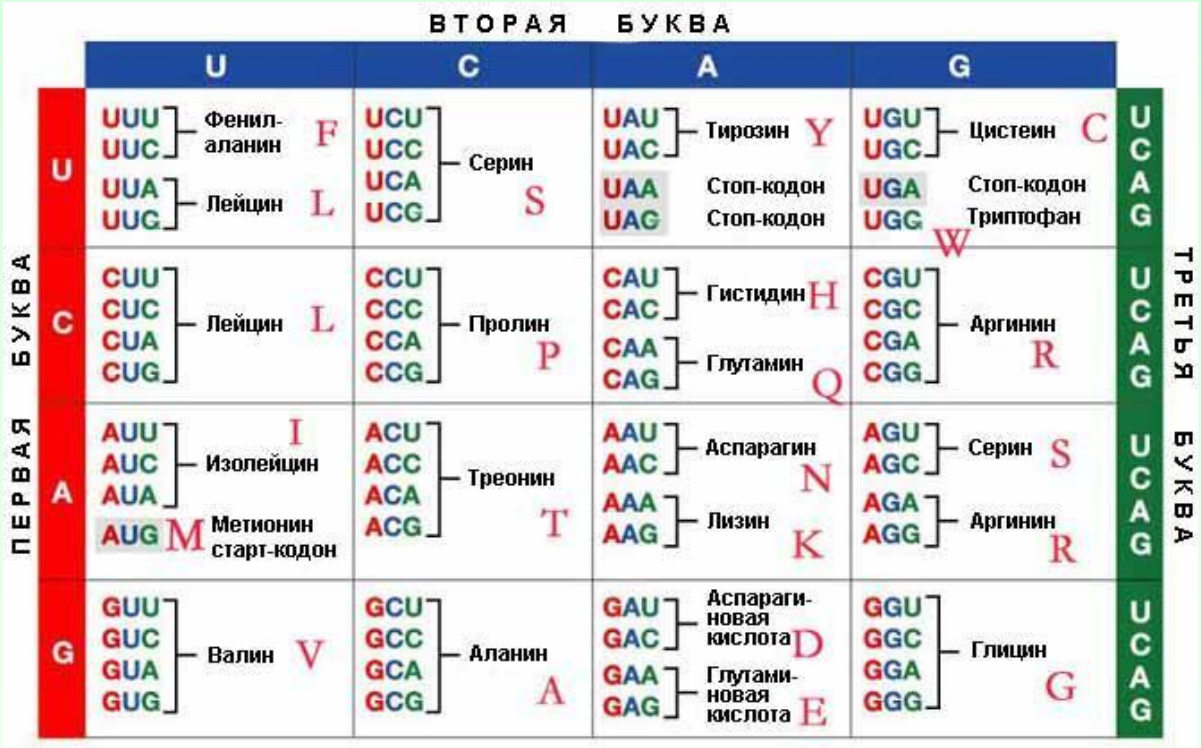
$\boldsymbol{T} = t_1 t_2 \ldots t_N$ $(t_i \in \Sigma,\ 1 \le i \le N)$ – последовательность символов, цепочка символов, текст, строка, слово.

Примеры:

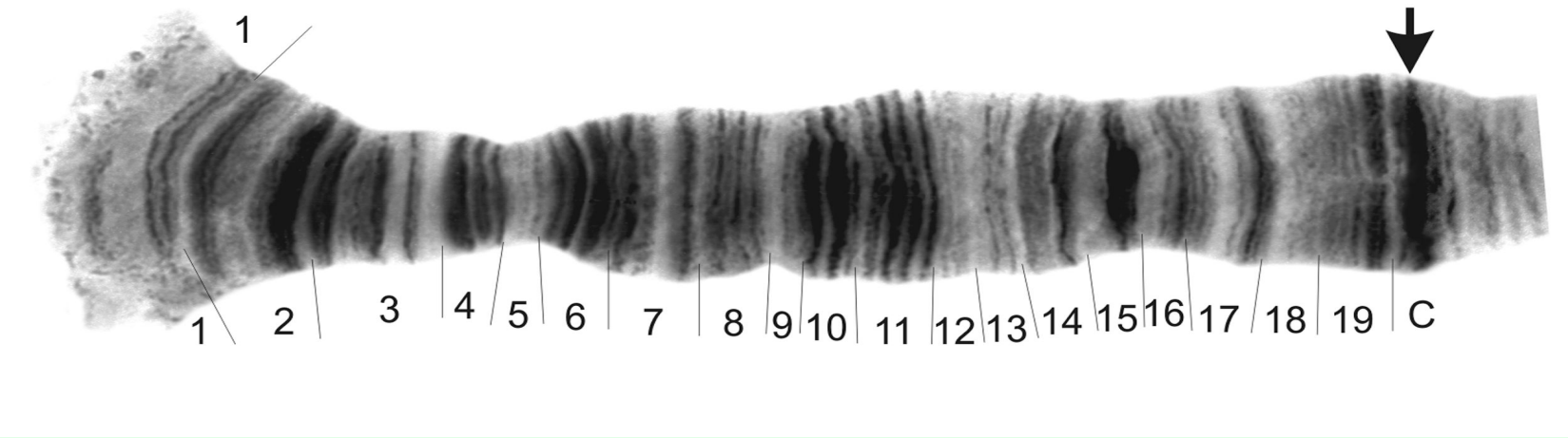
- слова, предложения,…, тексты естественного языка;
- музыкальные тексты (песенные мелодии);
- древнерусские церковные песнопения;
- тексты программ;
- ДНК, РНК ($|\Sigma| = 4$); аминокислотные послед. ($|\Sigma| = 20$);
- порядки генов; порядки дисков политенных хромосом;
- последовательность действий;
- двоичные последовательности;
- формальные последовательности.

ДНК и аминокислотные последовательности

- ДНК: $\Sigma = \{A, C, G, T\}$, РНК: $\Sigma = \{A, C, G, U\}$;
- Белки практически всех живых организмов построены из аминокислот всего 20 видов.

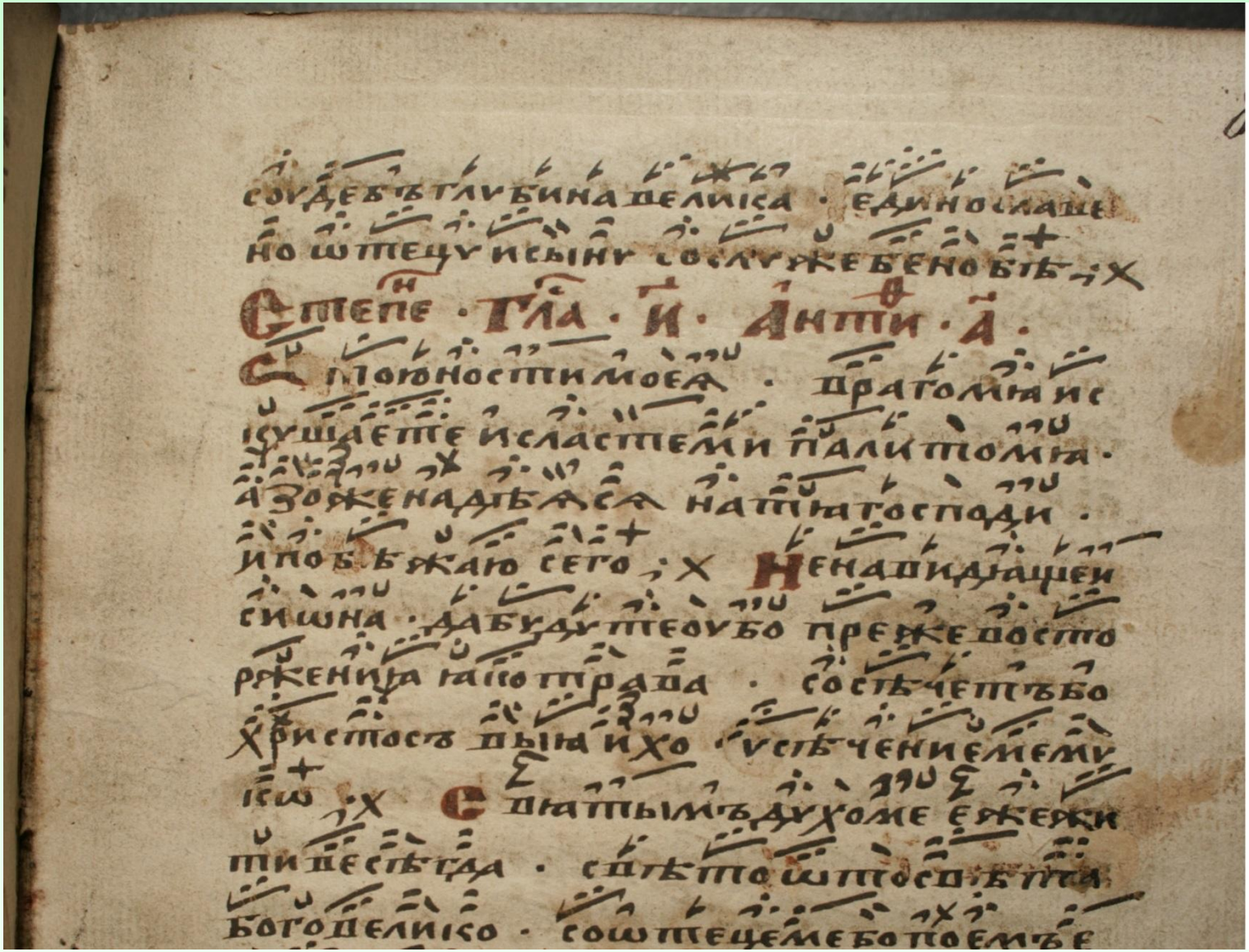


Polytene chromosomes



Cytophotomap of arm A of the species *C. piger*

Пример древнерусской церковной рукописи



Знамена (крюки)

Примеры начертаний:

– ***крюк:*** *e*2; – ***палка;*** – ***чаша:*** *d*4*c*4

– ***стрела простая:*** *f*1 (*e*1…)

– ***стрела поводная с облачком и оттяжкой*:** *d*4*e*4*f*2.*d*4

– ***голубчик борзый*:** *c*4*d*4 (*d*4*e*4, *e*4*f*4 …)

– ***стрела*:** *H*4*H*4*A*2

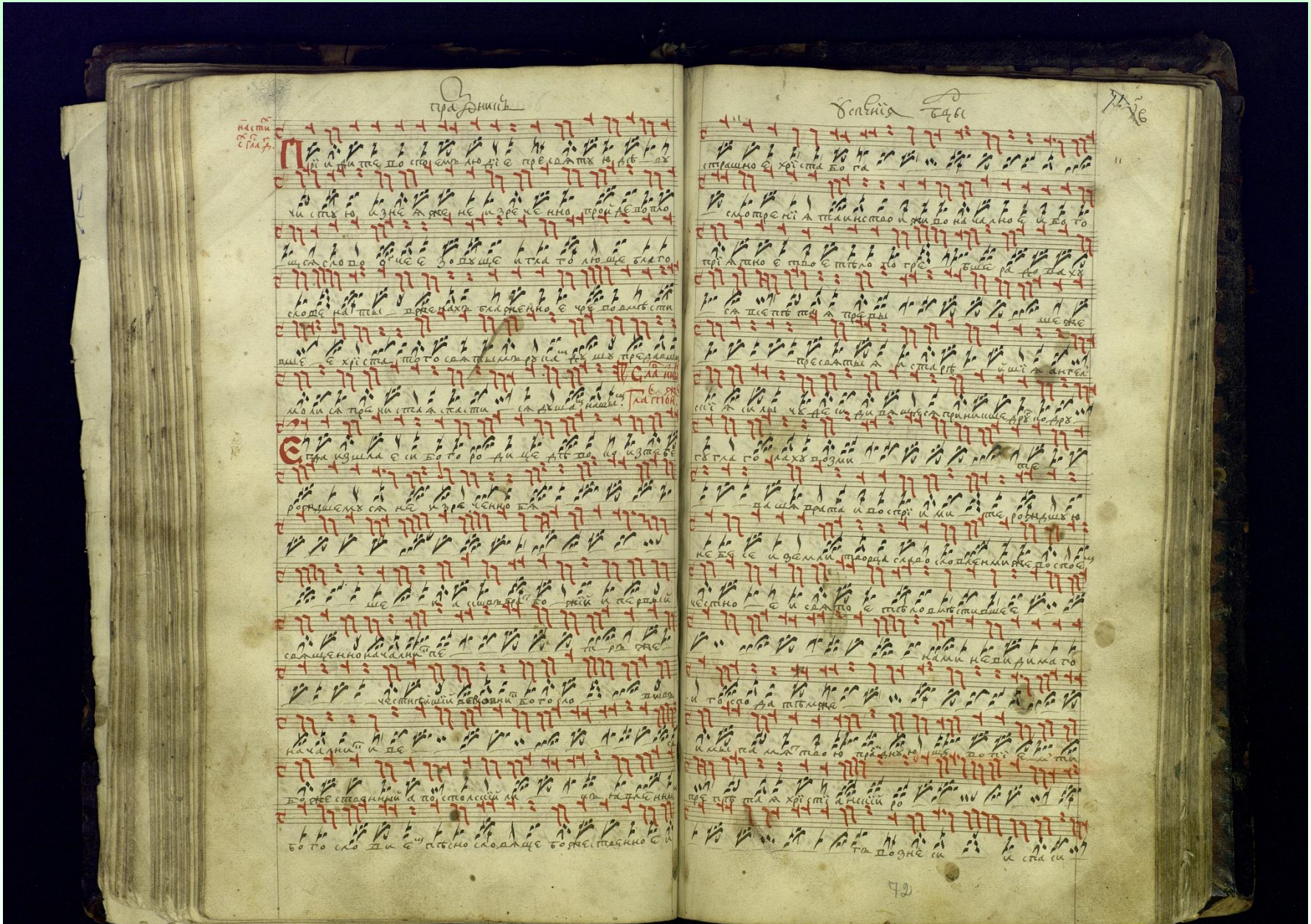
– ***скамейца со статьей*:** *d*4*e*4*d*4.*c*8*H*4

Примеры толкований:

– ***стопица с очком****:* назад *отшибнуть* гортанью, *вскочить* и *опуститься* на голубчик или на скамейцу: *e*4*d*4 (*d*4*c*4…)

– ***сложитие****: покудрить* гортанью: *f*8*e*8*f*4, *g*4*f*4…

Двознаменник

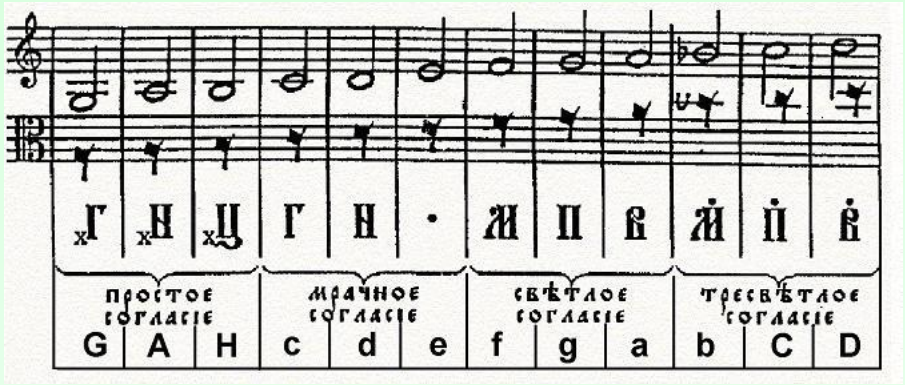


Кодировка песнопений из двознаменника

Первый и шестой символ кода – степенные и указательные пометы

- **Степенные –** указывают высоту распева знамен.
- **Указательные пометы (** – тихая, – **борзая**…**)** определяют характер исполнения распева знамен.

Знамена кодируются четырехсимвольным кодом.



Длительности звуков: – 1 (целая), – 2, – 4, – 8

Н4 – четвертная нота «си» малой октавы

Пример кодировки песнопения из двознаменника

```
(m0401-c2Во)(v0121-e2нми)(r0121-e2зе)(r0111-e2мле)
(r0211-e4d4и)(r1941-c4d4e2не)(p1011-d1бо)(v0901-c4e4и)
(p0302-d4c4вну)/
(-0501Td2e2ши)
(*1021-f1 )(-0511-d4e4гла)(#0141-f2го)(-1601Ld4e4лы)
(-0901-d4c4мо)(-1002-d1я)(-1001-c1 )(m0211-c4H4воз)
(-0511-c4d4гла)/
(v0121-e2го)(r0121-e2лю)(r0211-e4d4бо)(-0511-c4d4на)
(v0301-e2зе)(p1001-d1мли)(v0905Td2e2бо)(p0111-d2жи)
(p1861-c2d1я)(p0201-d2чю)(m0301-c2де)(-2801-H1са.)/@
```

Основные задачи анализа текста

- поиск образцов;
- восстановление структуры текста: выявление повторов (периодичностей, симметрий …);
- сравнение последовательностей: разные определения расстояний и мер близости;
- сложность текста
- сегментация, фрагментация, выделение структурных единиц…

Формальные языки и грамматики

Σ – алфавит;

$\boldsymbol{T} = t_1 t_2 \ldots t_N$ $(t_i \in \Sigma,\ 1 \le i \le N)$ – строка (слово, текст) ;

$N = |\ \boldsymbol{T}\ |$ – длина строки T;

$T[1 : p] = t_1 t_2 \ldots t_p$ – префикс слова $(1 \le p \le N)$,

$T[k : N] = t_k t_{k+1} \ldots t_N$ – суффикс $(1 \le k \le N)$,

$T[k : p] = t_k t_{k+1} \ldots t_p$ – подслово $(1 \le k \le p \le N)$;

e – пустая строка ($|\ e\ | = 0$);

Σ* – множество всех слов (строк) в алфавите Σ, включая e.

Язык $\boldsymbol{L}$ над Σ – произвольное множество слов в Σ ($L \subseteq \Sigma^*$).

Конкатенация языков L_1 и L_2 есть $L_1 L_2 = \{\ \alpha\beta : \alpha \in L_1,\ \beta \in L_2\}$.

L* : итерация языка L :

$L^0 = \{\varepsilon\}$,

$L^n = LL^{n-1}$ для $n \ge 1$,

$$L^* = \bigcup_{n \ge 0} L^n$$

Порождающей грамматикой называется четверка

$G = (\Sigma, N, P, S)$, где

Σ – алфавит терминальных символов, из которых составляются «слова» языка ($L(G) \subseteq \Sigma^*$);

N – алфавит нетерминальных символов (или переменных);

$\Sigma \cap N = \varnothing$;

P – конечное множество правили вывода вида $\alpha \to \beta$, где $\alpha \in (N \cup \Sigma)^* N (N \cup \Sigma)^*$, $\beta \in (N \cup \Sigma)^*$;

S – выделенный символ из N, называемый начальным (или исходным).

Формальная грамматика позволяет получить все цепочки данного языка и только их. Формальные грамматики были введены Хомским (1956г). Им же определена классификация грамматик в зависимости вида применяемых правил вывода (иерархия Хомского).

Иерархия Хомского

Пусть $G = (\Sigma, N, P, S)$ – грамматика. G называется:

- ***праволинейной***, если каждое правило из P имеет вид

  $A \to \alpha B$, где $A, B \in N$ и $\alpha \in \Sigma^*$;

- *праволинейная* грамматика называется ***регулярной*** (или автоматной), если все ее правила имеют вид

  $A \to aB$ или $A \to a$, где $A,B \in N$ и $a \in \Sigma$

- ***контекстно- свободной***, если каждое правило из P имеет вид

  $A \to \alpha$, где $A \in N$ и $\alpha \in (N \cup \Sigma)^*$

- ***контекстно- зависимой*** (или неукорачивающейся), если каждое правило из P имеет вид $\alpha \to \beta$, где $\alpha, \beta \in (N \cup \Sigma)^*$ и $|\alpha| \leq |\beta|$.

- Грамматика, на которую не накладывается ни одно из указанных ограничений, называется ***грамматикой составляющих***.

Языки называются праволинейными, КС или КЗ в зависимости от того, какой грамматикой он порожден.

Пример формальной грамматики

Пусть $G = (\{a,b,c\}, \{A,B,S\}, P, S)$,

где правила вывода P имеют вид:

$S \to AB$, $A \to a$, $A \to ac$, $B \to b$, $B \to cb$.

Данная грамматика позволяет получить всего 4 вывода терминальных строк:

(1) $S \to AB \to aB \to ab$

(2) $S \to AB \to aB \to acb$

(3) $S \to AB \to acB \to acb$

(4) $S \to AB \to acB \to accb$

$L(G) = \{ab, acb, accb\}$.

Для строки acb имеются два разных вывода.

Пример. Арифметические выражения

- Σ = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, – , * , / , (,) }
- N = {ФОРМУЛА, ЗНАК, ЧИСЛО, ЦИФРА };
- S = ФОРМУЛА
- Правила:
  1. ФОРМУЛА → ФОРМУЛА ЗНАК ФОРМУЛА
  2. ФОРМУЛА → ЧИСЛО
  3. ФОРМУЛА → (ФОРМУЛА)
  4. ЗНАК → + | – | * | /
  5. ЧИСЛО → ЦИФРА
  6. ЧИСЛО → ЧИСЛО ЦИФРА
  7. ЦИФРА → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Пример вывода (12 + 5) * 3

ФОРМУЛА $\Rightarrow_1$ ФОРМУЛА ЗНАК ФОРМУЛА $\Rightarrow_4$ ФОРМУЛА * ФОРМУЛА $\Rightarrow_2$ ФОРМУЛА * ЧИСЛО $\Rightarrow_5$ ФОРМУЛА * ЦИФРА $\Rightarrow_7$ ФОРМУЛА * 3 $\Rightarrow_3$ (ФОРМУЛА) * 3 $\Rightarrow_1$ (ФОРМУЛА ЗНАК ФОРМУЛА) * 3 $\Rightarrow_4$ (ФОРМУЛА + ФОРМУЛА) * 3 $\Rightarrow_2$ (ФОРМУЛА + ЧИСЛО) * 3 $\Rightarrow_5$ (ФОРМУЛА + ЦИФРА) * 3 $\Rightarrow_7$ (ФОРМУЛА + 5) * 3 $\Rightarrow_2$ (ЧИСЛО + 5) * 3 $\Rightarrow_6$ (ЧИСЛО ЦИФРА + 5) * 3 $\Rightarrow_5^2$ (ЦИФРА ЦИФРА + 5) * 3 $\Rightarrow_7$ (1 ЦИФРА + 5) * 3 $\Rightarrow_7$ (1 2 + 5) * 3

Конечные автоматы − средство распознавания

Детерминированный конечный автомат – это пятерка

M = (S, Σ, δ, s_0, F), где

S – конечное множество состояний;

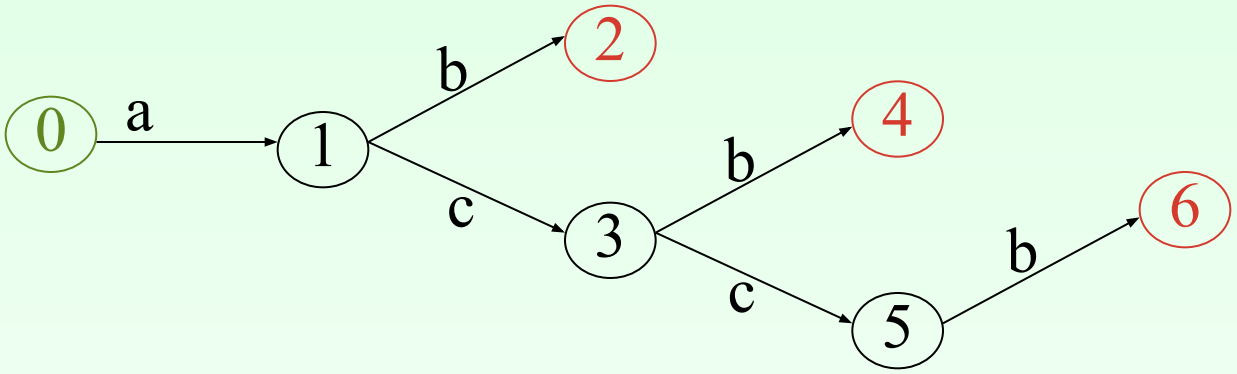
Σ – алфавит;

δ : S×Σ → S – функция переходов;

$s_0 \in$ S – выделенное начальное состояние;

$F \subseteq$ S – множество заключительных состояний;

ДКА, допускающий {ab, acb, accb}.



Формальные последовательности

Последовательность Туэ - Морса

Способы задания

1. итерации морфизмов.
   $\Sigma = \{a_1 \ldots a_q\}$ $\phi : \Sigma^* \rightarrow \Sigma^*$ – *морфизм*, если $\phi(XY) = \phi(X)\phi(Y)$ $\forall$ слов X и Y.
   $\phi = \{0 \rightarrow 01, 1 \rightarrow 10\}$.
   $X_0 = 0, X_1 = 01, X_2 = 0110, X_3 = 01101001, X_4 = 0110100110010110$ …
2. X[*i*] = 0, если число единиц в двоичной записи числа *i* чётно,
   X[*i*] = 1, в противном случае.
3. Итеративный способ: X[0] = 0, X[2*i*] = X[*i*], X[2*i*+1] = ((X[i] + 1) mod 2

Свойства последовательности Туэ-Морса:

1. Отсутствуют подслова вида VVV.
2. $X_{2n} = X_n X_n^R$: слово, полученное на чётном шаге является палиндромом.

Чи́сла Фибона́ччи — 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, …

Последовательность Фибоначчи

$X_0 = 0, X_1 = 01, X_n = X_{n-1}X_{n-2}$
$X_2 = 01.0, X_3 = 010.01, X_4 = 01001.010, X_5 = 01001010.01001$
Морфизм: $\phi = \{0 \rightarrow 01, 1 \rightarrow 0\}$