Формальные языки

алфавит Σ – конечное непустое мн-во

слово, цепочка, строка $\Sigma^* = \bigcup_{k=0}^{\infty} \Sigma^k$

формальный язык $L \subset \Sigma^*$

(Ⓝ) мн-во формальных языков (далее: языков) 2^{Σ^*} несчетно

„Описание" языка – слово конечной длины

Пр.:

$\exists\ \Sigma = \{0, 1\}$

$L =$ „слова, содерж. четное число единиц"

$0101 \in L$
$0111 \notin L$

(Ⓝ) всего „описаний" счетное мн-во

Перечисление слов $\{01, 011, 10, 1010\}$

- метод распознования ($w \in L$?)
- метод порождения (описание слова L)

Описание языка через метод порождение:

ε – п.с.п

A, B – п.с.п $\Rightarrow$ AB – п.с.п

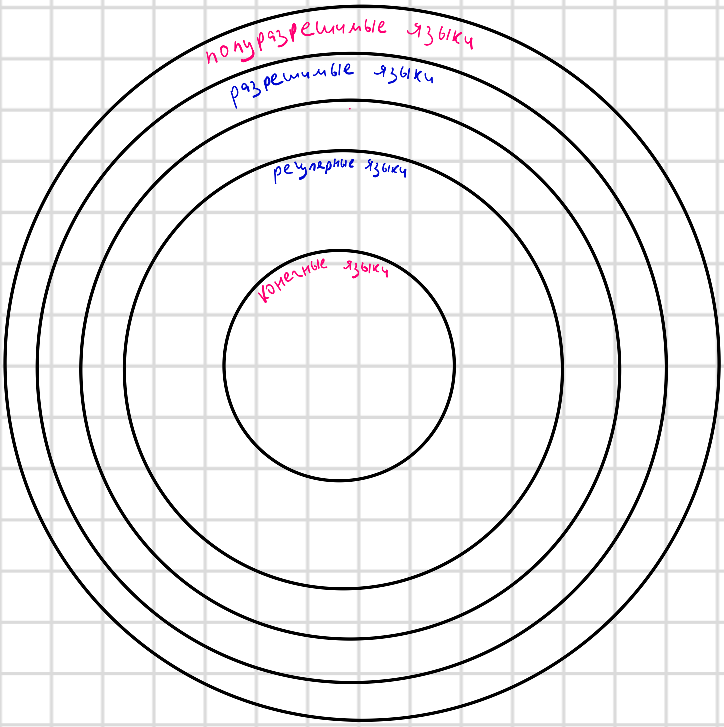
A – п.с.п $\Rightarrow$ (A) – п.с.п.

$\exists$ C++, программа P

$L = \{ w \mid p(w) = 1 \}$

Регулярные/автоматные языки

Конкатенация: $\alpha, \beta \in \Sigma^*$ $\quad \alpha\beta$



1) $\alpha \in \Sigma^k, \ \beta \in \Sigma^l \quad \alpha\beta \in \Sigma^{k+l}$

2) $\gamma = \alpha\beta \quad \gamma_i = \begin{cases} i \le k \Rightarrow \alpha_i \\ i > k \Rightarrow \beta_i \end{cases}$

3) $(\alpha\beta)\gamma = \alpha(\beta\gamma), \quad \alpha\varepsilon = \varepsilon\alpha = \alpha$

$$AB = \{ x \mid x = yz, \ y \in A, \ z \in B \}$$

Пр.:

$A = \{0, 01\}, \ B = \{0, 10\}$

$AB = \{00, 010, 0110\}$

Операции:

1) $A \cup B$

2) AB

$\Sigma^k \quad A^k = \underbrace{AA \ldots A}_{k}$

$A^0 = \{\varepsilon\}$

3) Замыкание Клини $A^* = \bigcup_{k=0}^{\infty} A^k$

$Reg_0 = \{ \varnothing, \{\varepsilon\}, \{c\} \text{ for } c \text{ in } \Sigma \}$

базовые регулярные языки

$Reg_{i+1} = Reg_i \cup \{ A \cup B, AB, A^* \mid A, B \in Reg_i \}$

$\varnothing^* = \{\varepsilon\}$

$Reg_1 = \{\varnothing, \{\varepsilon\}, \{a\}, \{b\}, \dots, \{a, b\}, \{a, \varepsilon\}, \dots, \{ab\}, \{aa\}, \{\varepsilon, a, a, a, \dots\}, \{\varepsilon, b, b, \dots, b\} \dots\}$

Опр.: Семейство регулярных языков: $Reg = \bigcup\limits_{k=0}^{\infty} Reg_k$

Лемма (замкнутость регулярных яз-ов)

$\text{Пусть } A, B \in Reg \Rightarrow$
- $A \cup B \in Reg$
- $AB \in Reg$
- $A^* \in Reg$

▷ $A \in Reg_i$, $B \in Reg_j$

$A \cup B, AB, A^* \in Reg_{max(i,j)+1} \subset Reg$ ◀

$X \in Good$ $\quad X \subset 2^{\Sigma^*}$ Хорошее семейство языков

$X = set<lang>$

$Good : set<set<lang>>$

1) $Reg_0 \subset X$
2) X замкнуто относ. $AB, A \cup B, A^*$ $(A, B \in X \Rightarrow A \cup B, AB, A^* \in X)$

Теорема („Хорошесть" регулярных)

$$Reg = \bigcap_{U \in Good} U$$

▷ потом ◀

Обозначение:

1) $\varnothing$ – пустое мн-во
2) ε – пуст. строка
3) c – символ

4) $\exists$ язык A описание α
B β

- $AB \to \alpha\beta$ (mid)
- $A \vee B \to \alpha | \beta$ (min)
- $A^* \to \alpha^*$ (макс priority)

Пр.:

$\Sigma = \{0, 1\}$

$(0|11)^*$

1) $0 = \{0\}$

$1 = \{1\}$

2) $11 = \{11\}$

3) $0|11 = \{0, 11\}$

$(\ ^* \ | A \cdot B)^*$ ← академические регулярные выражения

$\alpha^+ = \alpha \cdot \alpha^*$

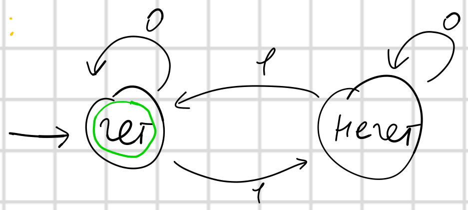
$\alpha^k = \underbrace{\alpha\alpha \dots \alpha}_{k}$

Опишем „слова, содерж. четное число единиц"

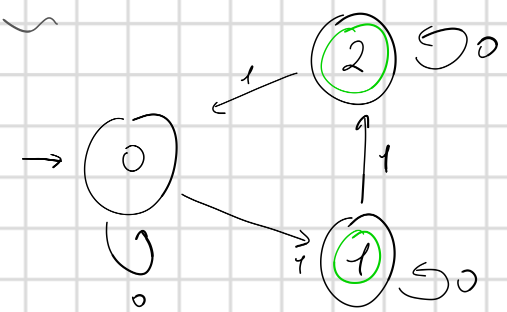
$0^* | (0^*10^*10^*)^*$

Детерминированный конечный автомат

Ч:



четн. кол-во единиц



число не делится на 3

$$A = \langle \Sigma, Q, S \in Q, T \subset Q, \delta : Q \times \Sigma \to Q \rangle$$

Σ — алфавит; Q — конечное мн-во состояний; S — начальное состояние; T — допускающее состояние; δ — функция переходов

$$i \xrightarrow{c} \delta(i, c)$$

$\exists X$ (строка) $\quad |X| = n$

$S_{nap} = Q \times \Sigma^*$ (мн-во всех мгновенных описаний автомата)

„$\vdash$": $\langle q, \alpha \rangle \vdash \langle r, \beta \rangle$ (отношение „переходит за один шаг")

1) $\alpha = c\beta, \; c \in \Sigma$

2) $r = \delta(q, c)$

Пр.:

$$\langle 4, 0101 \rangle \vdash \langle 4, 101 \rangle \vdash \langle 4, 01 \rangle \vdash \langle 4, 1 \rangle \vdash \langle 4, \varepsilon \rangle$$

$$L(A) = \{ w \mid \langle S, w \rangle \vdash^* \langle t, \varepsilon \rangle, \; t \in T \}$$ ← язык автомата A

Теорема (Клини)

$Reg = Aut$

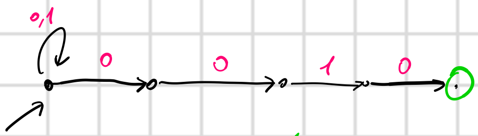
$Aut = \{ X \mid \exists$ ДКА $A : X = L(A) \}$

План док-ва:

для L $\exists$ рег. выражение $\Leftrightarrow$ для L $\exists$ DKA

рег. выражение $\xrightarrow{\text{рекурсивно}}$ ε-НКА $\xrightarrow{\varepsilon\text{-замык.}}$ НКА $\xrightarrow{\text{алг. Томпсона}}$ DKA

DKA $\xrightarrow{\text{О.П.}}$ рег. выр.



Недетерминированный КА

0 0 1 0 ✓

0 0 0 0 0 ✗

▷ $\boxed{\Longrightarrow}$ Reg ⊂ Aut

НКА $= \langle \Sigma, Q, S \in Q, T \subset Q, \delta: Q \times \Sigma \to 2^Q$ (set<int>)

ДКА $= \langle , , , , , \delta: Q \times \Sigma \to Q$

$L_A = \{ w \mid w$ - можно принять $A \}$

$w \in L_A$: явно показать путь

$w \notin L_A$: ни один путь не приводит в терм. сост.

Эмулируем ДКА

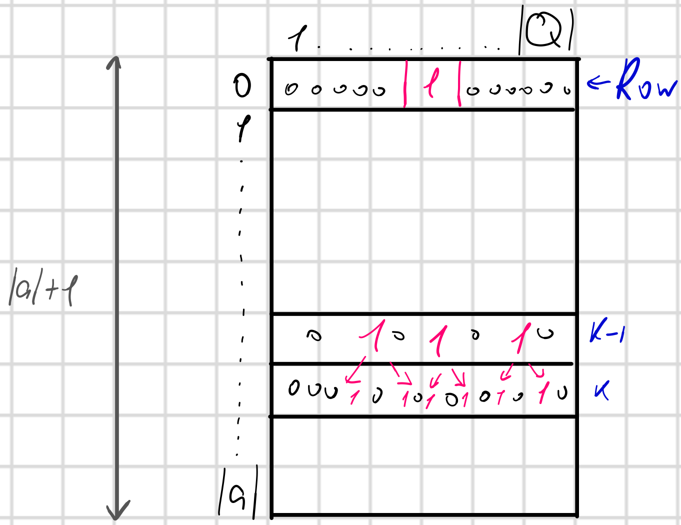
S: int ↔ →•

T: array<bool>

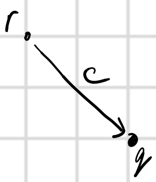
$T[q] = true$ ↔ (q)

$\delta[q][c] = r$ $\quad q \xrightarrow{c} r$

```
bool check(a):
    int cur = S
    for c in a:
        cur = δ[cur][c]
    return T[cur]
```



$can[k][q]$ - можно прийти в q, считав k символов



$can[k][q] = 1$ if $can[k-1][r] = 1$ && $\delta[r][c] \ni q$

```
Row next(Row, X, c):      // X — вектор
    Row ans;
    for r in Q:
        if X[r] = 1
            for q in δ[r, c]:
                ans[q] = 1
```

```
Check(a):
    Row cur = {s}
    for c in a:
        cur = next(cur, c)
    return (cur ∩ T) ≠ ∅
```

ДКА

$Q_{ДКА} = 2^{Q_{НКА}}$ $\qquad 2^{A}$ = мн-во подмн-в (мн-во)

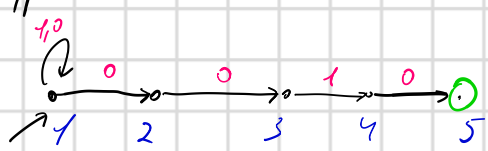
$S_{ДКА} = \{S_{НКА}\}$ Row

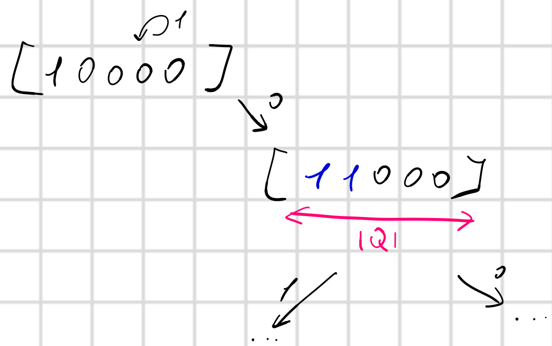
$\delta(q, c) = next(q, c)$

$q \in T_{ДКА} : q \cap T_{НКА} \neq \emptyset$

} алгоритм Томпсона

Пр.:

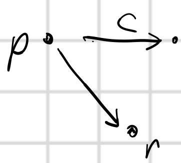




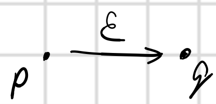
ε − переход:

ε − НКА

было:



можно:



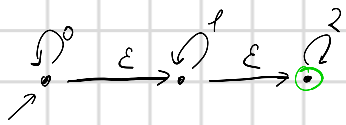
$p \rightsquigarrow q$, не читая символ

НКА $\overset{?}{\Longleftrightarrow}$ ε − НКА

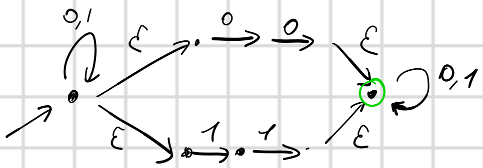
ε − замыкание

Пр.:

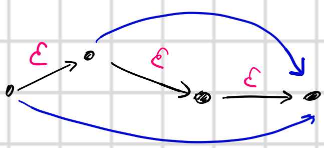
1) $0^* 1^* 2^*$

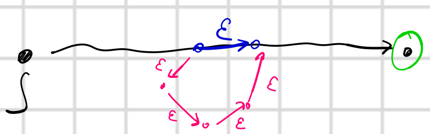


2) есть 00 или 11

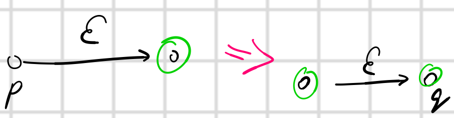


- Транзитивно замкнуть граф ε − пер.





- если слово можно допустить, то его можно допустить не делая 2 ε − пер. подряд

- 

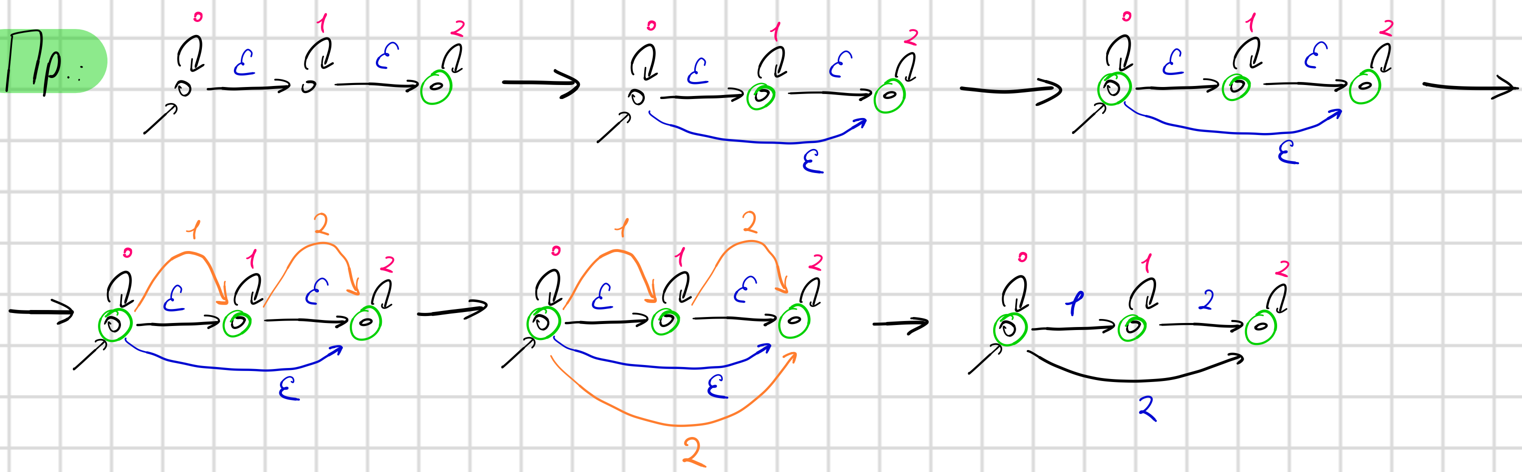
если слово можно допустить, то его можно принять и

после пер. на ε

если слово можно допустить, то его можно допустить без ε-пер.

$\Rightarrow$ ε-НКА → НКА

Пр.:

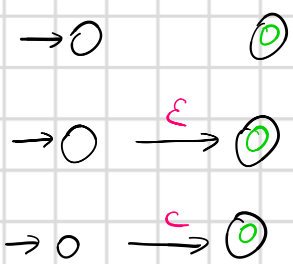


← автомат, - терм, - вход

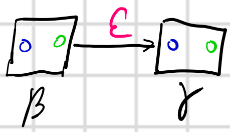
∅

{ε}

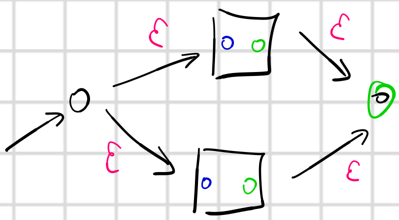
{c}



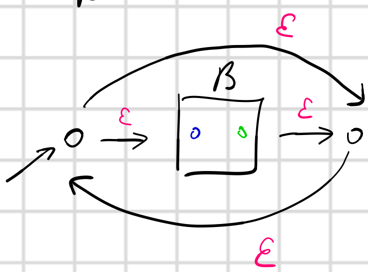
$\alpha = \beta\gamma$ $w = ab$



$\alpha = \beta | \gamma$



$\alpha = \beta^*$



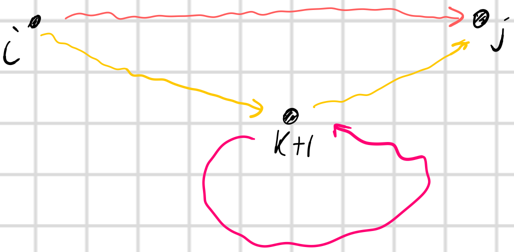
Aut с Reg

ДКА → рег. выр.

$\mathcal{G}_{ijk}$ — рег. выр. для слов. $\leftrightarrow$ $i \underbrace{\rightsquigarrow}_{\leq k} j$

$$\mathcal{G}_{ii0} = \varepsilon \mid \underbrace{c_1 \mid c_2}_{\text{символы на петлях}} \ldots$$

$$\mathcal{G}_{ij\,k+1} = \mathcal{G}_{ijk} \mid \mathcal{G}_{i,k+1,k} \left(\mathcal{G}_{k+1,k+1,k} \right)^* \mathcal{G}_{k+1,j,k}$$



Один раз: $\mathcal{G}_{i,k+1,k}$ $\mathcal{G}_{k+1,j,k}$

Два раза: $\mathcal{G}_{i,k+1,k}$ $\mathcal{G}_{k+1,k+1,k}$ $\mathcal{G}_{k+1,j,k}$

Ответ: $\mathcal{G}_{st_1n} \mid \mathcal{G}_{st_2n} \mid \mathcal{G}_{st_3n} \ldots, \; t_i \in T$ ◀