**Регулярные выражения**

Пусть **Σ** = {a*i* | 0 ≤ *i* < *n*} – входной алфавит. Определим и обозначим через ***E*** метаязык регулярных выражений над алфавитом **Σ**, через *L* – функцию, сопоставляющую каждому выражению *e* ∈ ***E*** язык *L*(e) ⊆ Σ\*. Функция *L* называется стандартной интерпретацией выражений, определяет значение произвольного регулярного выражения.

Определим и обозначим *D* – функцию, сопоставляющую каждому выражению *e* ∈ ***E***  
*D*(е) – диаграмму, графическое представление *недетерминированного конечного автомата* (НКА) как распознавателя языка *L*(e). Функция *D*  называется не стандартной интерпретацией выражений, определяет другое значение произвольного регулярного выражения.

*Базис*.  
1) Если a ∈ Σ, тогда слово а – элементарное регулярное выражение и *L*(a) = {a} – элементарный язык;

2) Символ ∅ обозначает элементарное регулярное выражение и *L*(∅) = {} – элементарный пустой язык;

3) Символ λ обозначает элементарное регулярное выражение и *L*(λ) = {λ} – элементарный язык, состоящий из единственного пустого слова λ;

Элементарные диаграммы *D*(a), *D*(∅) и *D*(λ) имеют следующие представления:

1

0

a

1

0

1

0

λ

*D*(a):

*D*(∅):

*D*(λ):

*Индуктивный переход.*

Пусть e, e1, e2 – произвольные регулярные выражения, которые обозначают соответствующие им языки *L*(e), *L*(e1), *L*(e2) и диаграммы *D*(e), *D*(e1), *D*(e2).  
Тогда  
1) слово (e) – регулярное выражение и *L*((e)) = *L*(e);

2) слово (e1 | e2) – регулярное выражение и *L*((e1 | e2)) = *L*(e1) ∪ *L*(e2);

3) слово (e1e2) – регулярное выражение и *L*((e1e2)) = *L*(e1)*L*(e2);

4) слово e\* – регулярное выражение и *L*(e\*) = *L*\*(e);

Выразим диаграмму *D*(e) схемой:

n

0

*D*(e):

e

Тогда

1)

n

0

*D*((e)):

e

2)

n1

0

e1

*D*(e1):

n2

0

e2

*D*(e2):

n1+n2-1 111

-1

0

e1

e2

*D*(e1|e2):

Modify *D*2 : i+n1 – 1, i in [1-n2]; **0~0** and **n1 ~ n1+n2 – 1**, **n1+n2 – 1**

3)

n1

0

e1

*D*(e1e2):

n2 + n1

e2

Modify *D*2 : i+n1 , i in [1-n2]; n1~0 and **n1 + n2**

4)

0

n + 1

n

λ

λ

ee

*D*(e\*):

Modify *D*: 0~n, add **(0,λ,n)** and  **(n,λ,n+1)**