**Проектирование ДКА, допускающего класс «идентификаторов» языка СИ**

1. Σ = ASCII – входной алфавит.
2. Спецификация – описание формы произвольного слова(идентификатора).  
   *w* ∈ *L* , *w* = a*s*, где а – любая буква, *s* – произвольное слово, составленное из букв и цифр. Под буквой подразумевается любая прописная или строчная буква английского алфавита, а также знак «подчеркивания».
3. [ \_a-zA-Z0-9] – допустимые символы,  
   [^\_a-zA-Z0-9] – недопустимые символы
4. Регулярное выражение:  
   е = [\_a-zA-Z][ \_a-zA-Z0-9]\*
5. Разбиение класса допустимых символов и всего входного алфавита:  
   [\_a-zA-Z], [0-9], [^\_a-zA-Z0-9].  
   Символы a , 9, ? – представители соответствующих классов, составляют абстрактный алфавит.
6. Система регулярных выражений, определяющих язык:  
   *е*  = a(a|9)\* ;  
   a = [\_a-zA-Z] ;  
   9 = [0-9].  
   ? = [^\_a-zA-Z0-9].
7. Построить λ- диаграмму ( графическое представление НКА) по заданному  
   главному регулярному выражению *е* в абстрактном алфавите.

а

а

9

а|9

λ

λ

а|9

λ

λ

а|9

а

0

1

2

3

λ- диаграмма ( графическое представление НКА ) для *е* = a(a|9)\* в абстрактном алфавите {а, 9, ?}.

1. Построить ДКА = (Q, Σ, *g*, q0, F).  
   Q – множество состояний;  
   Σ = {а, 9, ?} – абстрактный алфавит;  
   q0 = [{0}] = [0] – начальное состояние, q0 ∈ Q;  
   F – множество финальных (или заключительных) состояний, F ⊆ Q;  
   *g*: Q×Σ → Q – функция перехода ДКА,   
   *g*(q, a) = [*μ* (q, a)], где *μ*: Q×Σ →2X – функция перемещения ДКА,

X = {0,1,2,3} = [0-3] – множество состояний λ- диаграммы (НКА).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| База Данных: q' = [*μ* (q, a)] | q | а | 9 | ? |
| q0 = [0] = {0} | q0 | q1 |  |  |
| q1 = [1] = {1, 2, 3} | q1 | q2 | q2 |  |
| q2 = [{2,3}] = {2, 3} | q2 | q2 | q2 |  |
| q3 = [{}] = {} | q3 |  |  |  |

q0 = [0] = {0} = 0

*g*(q0, a) = [*μ* (q0, a)] = [*μ* ({0}, a)] = [*μ* (0, a)] = [{1}] = [1] = {1, 2, 3} = q1

*g*(q0, 9) = [*μ* (q0, 9)] = [*μ* ({0}, 9)] = [*μ* (0, 9)] = [{}] = {}

*g*(q0, ?) = [*μ* (q0, ?)] = [*μ* ({0}, ?)] = [*μ* (0, ?)] = [{}] = {}   
  
*g*(q1, a) = [*μ* (q1, a)] = [*μ* ({1,2,3}, a)] = [{2}] ={2, 3} = q2

*g*(q1, 9) = [*μ* (q1, 9)] = [*μ* ({1,2,3}, 9)] = [{2}] = {2, 3} = q2

*g*(q1, ?) = [*μ* (q1, ?)] = [*μ* ({1,2,3}, ?)] = [{}] = {}

*g*(q2, a) = [*μ* (q2, a)] = [*μ* ({2,3}, a)] = [{2}] ={2, 3} = q2

*g*(q2, 9) = [*μ* (q2, 9)] = [*μ* ({2,3}, 9)] = [{2}] = {2, 3} = q2

*g*(q2, ?) = [*μ* (q2, ?)] = [*μ* ({2,3}, ?)] = [{}] = {}

Q = {q0, q1, q2, q3} = [0-3]

F = {q| q ∈ Q, n ∈ X, n+1 = |X|, n ∈ q}.

n – финальное состояние λ- диаграммы (НКА).

F = {q1, q2}. Любое q ∈ F называется состоянием «допуска».

Cостояние q3={} называется состоянием «ошибки».

ДКА называется распознавателем или акцептором лексем-идентификаторов.

а|9

а|9

а

0

1\*

2\*

3

**Определение функции перемещения *μ* (S, a)**

*D* = (X,E) –диаграмма над алфавитом Σ.

Х – множество состояний,

Е ⊆ Х × Σ`× Х- множество помеченных дуг, где Σ` = Σ ⋃ {λ}.

***μ* (S, a) = {r | (s,a,r) ∊ E, s ∊ S, r ∊ X}, a ≠ λ.**

Очевидны следующие **свойства функции перемещения *μ* (S, a)**:

***μ* (S, a) = {},** если **S = {}** пусто или, если для каждого **s ∊ S не существует а**-дуги вида **(s, а ,r)**

**Замкнутые и открытые множества. Замыкание множества.**

**Оператор замыкания.**

Пусть **s ∊ Х –** состояние диаграммы.

Тогда ***о*(s) – *окрестность*** *состояния* ***s*** *в диаграмме**D:*

***о*(s) = {s}** ⋃ {r| **(s, λ ,r) ∊ E}.**

Окрестность ***о*(s) ={s},** если из **s** не выходит ни одна **λ**-дуга **(s, λ, r).**

*Подмножество состояний*  **S ⊆ X** называется ***замкнутым*,** если выполнима импликация  **s ∊ S => *о*(s) ⊆ S.**

Очевидно пустое множество **{}** и все множество состояний **Х**

диаграммы ***D*** замкнуты**.**

Наименьшее по включение замкнутое множество **T,** включающеемножество **S,** называется ***замыканием* S.**

Для каждого **S ⊆ X существует и единственно**  его замыкание и обозначается **[S] .**

В силу вышеуказанного свойства квадратные скобки **[]** обозначают ***унарный оператор*,** который каждому **S ⊆ X** однозначно ставит в соответствие ***замыкание* [S], S ⊆ [S] ⊆ X.**

Выражение [s] – краткая запись замыкания [{s}] одно элементного множества.

**Очевидны следующие свойства:**

**1) [S] = S,** если **S -** замкнутое множество;

**2) множество S не является замкнутым, если**

**существует λ**-дуга **(s, λ ,r), что s ∊ S и r ∉ S.**

*Не замкнутые* множества называются *открытыми.*

3) если **S** открытое множество, тогда **S** ⊊ **[S].**

Если множество **S** открыто, тогда на основании свойства **2)** с помощью

**λ**-дуги **(s, λ ,r)** можно перейти к **S** ⋃ {r} к большему множеству и, в

конечном счете, прийти к **[S].**

Обозначим через **path(**s, r**)** - путь, проходящий по **λ-**дугам из состояния sвсостояние r**,** тогда **замыкание подмножества состояний** **S ⊆ X** вычисляется по формуле

[**S**] = **S** ⋃ {r | **path(**s, r**),**  **s ∊ S,** r **∊ S** };

[s] = {s}⋃ {r | **path(**s, r**),**  r **∊ S** }.

Множество [s] включает состояние s и все состояния достижимые по

**λ**-дугам из состояния s, т.е. множество {r | **path(**s, r**),**  r **∊ S** }.

Имеет место следующее свойство

[**S**] = .

Очевидно [S] = S, если S – **пустое** множество или, если **не пустое**, S ≠ ⊘, и

**не существует** **path(**s, r**)** - пути, проходящего по **λ-**дугам, для всех состояний **s ∊ S,** т.е. {r | **path(**s, r**),** r **∊ S** } = ⊘.

Другими словами, [S] = S, если **не существует λ**-дуги **(s, λ, r)** для всех состояний **s ∊ S.**

**Автоматная функция переходов ДКА**

**g(q, a) = [ *μ* (q, a)] ,** где **q ⊆ Х, a ∊** Σ**, q = [q], a ≠ λ.**

**Преобразование НКА (недетерминированного конечного автомата)**

**в ДКА(детерминированный конечный автомат)**

**Вход: НКА, представленный в виде λ-диаграммы над алфавитом** Σ:

***D* = (X, E),**

**Х** – множество состояний (вершин),

**Е ⊆ Х × Σ` × Х**- множество помеченных дуг, где **Σ` = Σ ⋃ λ,**

**Х** = {0, 1, … , n}, n > 0,

**0** – начальное состояние,

**n** – финальное (заключительное) состояние.

**Выход: ДКА = (Q, Σ, g, q0, F).**  
**Q** – множество состояний;  
**Σ**  – алфавит;  
**q0** = [{0}] = [0] – начальное состояние, **q0 ∈Q**;  
**F** – множество финальных (или заключительных) состояний, **F ⊆ Q,**

**F = {q| n ∊ q**, **q ∊ Q**}**;**  
**g**: Q×Σ →Q – функция переходов **ДКА**,   
**g**(q, a) = [**μ** (q, a)], где **μ**: Q×Σ →2X – функция перемещения ДКА,

X – множество состояний **λ- диаграммы** (**НКА**).

**Расширение функции переходов ДКА -- g\*: Q×Σ\* →Q**

**g\*(q, a) = g(q, a), q ∊ Q, a ∊**Σ;

**g\*(q, λ) = q;**

**g\*(q, w) = g(q, a) g\*(q, w’), w = aw’, a ∊** Σ, **w’ ∊** Σ\*.

**ДКА допускает (распознает) язык**

**L(ДКА) = {w| g\*(q0, w) ∊ F}.**

Любое слово **w ∊** **L(ДКА)** находится в биективном соответствии с путем, проложенным по **а**-дугам, переводящим автомат из начального состояние в заключительное состояние.

**L(ДКА)** называется **автоматным языком,** причем **L(ДКА) = L(*D*).**

**ДКА** называется **акцептором** или **распознавателем** языка **L(ДКА).**