**Декартово произведение двух автоматов**

**Определение 1.**  Пусть *L* – язык над алфавитом Σ . На множестве всех слов Σ\* относительно языка *L* определим бинарное отношение *RL:*

***x*** *RL***y** выполнимо, если *v****x****w* ∍ *L* тогда и только тогда, когда *v****y****w* ∍ *L* для каждой пары слов

*v,w* ∍ Σ\*.

Это отношение является **эквивалентностью**. Непосредственно проверяется, что выполняются три свойства: **рефлексивность**, **коммутативность** и **транзитивность**.

Оно устойчиво относительно операции «**конкатенации**» слов на множестве Σ\* и называется «**конгруентностью**».

Если слово *v****x****w* ∍ *L* и ***x*** конгруентно ***y*** , тогда *v****y****w* ∍ *L*. Если же слово *v****x****w* ∉ *L*, тогда и

*v****x****w* ∉ *L*.

Если язык *L* – регулярный, тогда число классов эквивалентности конечно. Регулярный язык *L* и его дополнение Σ\*\ *L* представимы как конечное объединение классов конгруентных слов.

Множество символов алфавита, очевидно, разбивается по отношению эквивалентности *RL*

на классы эквивалентных символов.

***LId***  **- класс лексем «идентификаторов» языка Си**

**Регулярное выражение**:

**e*Id* = “[\_A-Za-z][ \_A-Za-z0-9]\*”**

***LId*** = ***L***(**e*Id***)

**Разбиение входного алфавита**:

**[\_A-Za-z0-9]** – множество **допустимых** символов;

**[^\_A-Za-z0-9]** – множество **недопустимых** символов.

Правильное разбиение:

**[\_A-Za-z], [0-9], [^\_A-Za-z0-9].**

Новый **базис** из **представителей** **классов** **символов** исходного алфавита **Σ**:

***BId* = {a, 9, ?}.**

**Регулярное выражение в базисе** ***BId*:**

**“a(a|9)\*”.**

***LOct***  **- класс лексем «целых констант с основанием 8» языка Си**

**Регулярное выражение**:

**e*Oct* = “0[ 0-7]\*((l|L|λ)(u|U|λ)|( u|U)(l|L))”**

***LOct*** = ***L***(**e*Oct***)

**Разбиение входного алфавита**:

**[0-7lLuU]** – множество **допустимых** символов;

**[^0-7lLuU]** – множество недопустимых символов.

Правильное разбиение:

**[0], [1-7], [lL], [uU], [^0-7lLuU]**

Новый **базис** из **представителей** **классов** **символов** исходного алфавита **Σ:**

***BOct* = {0, 7, l, u, ?}.**

**Регулярное выражение в базисе *BOct*:**

**“0(0|7)\* ((l|λ)(u|λ)|ul)”.**

**Диаграммы (НКА-недетерминированные конечные автоматы)**

*3*

*2*

*1*

*0*

a|9

**λ**

a

**λ**

**Рисунок 1. Диаграмма *D1 =D*(“a(a|9)\*”)**

*6*

*5*

*4*

u|λ

l|λ

l

u

*3*

*2*

*1*

*0*

0|7

**λ**

0

**λ**

**Рисунок 2. Диаграмма *D2 = D*(“0(0|7)\* ((l|λ)(u|λ)|ul)”)**

**Приведение к общему базису *B***

Базис ***BId*** = **{a, 9, ?}** выбран из разбиения **Π*Id* = { [\_A-Za-z], [0-9], [^\_A-Za-z0-9] },**

базис ***BOct*** = **{0, 7, l, u, ?}** − из разбиения **Π*Oct* = { [0], [1-7], [lL], [uU], [^0-7lLuU] }.**

Чтобы найти общий базис, необходимо найти пересечение разбиений − **наибольшую нижнюю границу разбиений:**

**Π** = **Π*Id*** ∩ **Π*Oct*** = ***Сij*****,** ***Сij*** = ***Сi* ∩ *Сj* ≠ ∅** , ***Сi*** ∈ **Π*Id*** **,** ***Сj*** ∈ **Π*Oct*** **.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Π*Id***  **Π*Oct*** | **[\_A-Za-z]** | **[0-9]** | **[^\_A-Za-z0-9]** |
| **[0]** |  | **[0]** |  |
| **[1-7]** |  | **[1-7]** |  |
| **[lL]** | **[lL]** |  |  |
| **[uU]** | **[uU]** |  |  |
| **[^0-7lLuU]** | **[\_A-Za-z-[lLuU]]** | **[89]** | **[^\_A-Za-z0-9]** |

**Таблица 1.** **Вычисление** **наибольшей нижней границы** **двух разбиений:** **Π** = **Π*Id*** ∩ **Π*Oct***

**Π** = **Π*Id*** ∩ **Π*Oct*** = {**[0]**, **[1-7]**, **[lL]**, **[uU], [89]**, **[\_A-Za-z-[lLuU]]**, **[^\_A-Za-z0-9]**}

**Общий базис** может быть ***B*={0, 7, l, u, 9, a, ?}.** Представители соответствующих классов разбиения **Π** могут быть выбраны по-другому, но **идентификация классов остается неизменной.**

Классы исходных разбиений **Π*Id*** и **Π*Oct*** однозначно выражаются через разделенные объединения классов разбиения **Π**. **На языке регулярных выражений элементы базисов *BId* и *BOct* выражаются через базис *B* посредством элементарных аддитивных выражений:**

**а) элементы базиса** ***BId***:

**[\_A-Za-z] = [\_A-Za-z-[lLuU]] | [lL] | [uU]**

**[0-9] = [0] | [1-7] | [89]**

**[^\_A-Za-z0-9] = [^\_A-Za-z0-9]**

**a ::= l|u|a,**

**9::= 0|7|9,**

**?::= ?;**

**b) элементы базиса** ***BOct***:

**[0] = [0]**

**[1-7] = [1-7]**

**[lL] = [lL]**

**[uU] = [uU]**

**[^0-7lLuU] = [\_A-Za-z-[lLuU]] | [89] | [^\_A-Za-z0-9]**

**0::=0,**

**7::=7,**

**l::=l,**

**u::=u,**

**?::=9|a|?.**

Справа от знака определения **«::=» аддитивное выражение в базисе *B***. Cимволы слева и справа **представляют классы различных разбиений** одного и того же **входного алфавита Σ,** несмотря на совпадение знаков.

Символы **0, 7, l, u, ?** представляют одни и те же классы как слева так и справа, но

символы **9, а** представляют различные классы.

**Конструирование ДКА(детерминированных конечных автоматов)**

Возможны **две технологии конструирования ДКА:**

**a)**преобразовать диаграммы в базисах ***BId*** и ***BOct*** в диаграммы относительно общего базиса ***B*.** Затем **конструировть ДКА в общем базисе *B*** и конструировать их декартово произведение;

**b) конструировть ДКА в частных базисах *BId*** и ***BOct*** для соответствующих диаграмм.

Затем преобразовать ДКА в ДКА над общим базисом ***B*** и конструировать их декартово произведение.

**В случае a)** преобразуются **a**-дуги **(i, a, j), a ≠ λ,** **диаграмм в частных базисах**.

**В случае b)**преобразуются **a**-дуги **(i, a, j)** **функции переходов автоматов в частных базисах,** где **j = g(i,a).**

Пусть **символ** **a** в частном базисе выражается посредством **элементарного аддитивного**

**выражения b*1*| …| b*k* в общем базисе: a ::= b*1*| …| b*k*.**

Тогда **a**-дуга **(i, a, j) заменяется k кратными**(параллельными) дугами **(i, b*1*| …| b*k*, j).** Ясно, что исходные функции переходов **g*1*: Q*1* × *BId* → Q*1* и g*2*: Q*2* × *BOct* → Q*2*** просто **заменяются** соответствующими функциями переходов **g*1*: Q*1*x *B*→ Q*1* и g*2*: Q*2*x *B* → Q*2* над общим алфавитом *B*.** Причем **состояния,** включая **начальное и финальные состояния,** остаются **неизменными.**

Применим **метод б).**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Замыкание** | **q*i*\ a** | **a** | **9** | **?** |
| **{0}=[0]** | **0** | **1** |  |  |
| **{1,2,3}=[1] \*** | **1** | **3** | **3** |  |
| **{}=[{}]** | **2** |  |  |  |
| **{2,3}=[2] \*** | **3** | **3** | **3** |  |

**Таблица 2.** ДКА ***A1 =A*(“a(a|9)\*”)** над частным алфавитом ***BId***

**q*j* = g(q*i* , a) ::= [μ( q*i* , a)]**

Для **0**-состояния:

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

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

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

Для **1**-состояния:

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

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

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

Для **2**-состояния:

**g(2, a|9|?)** = **g({}, a|9|?) = [μ(** {}**, a|9|?)]** = [{}] = {}

Для **3**-состояния:

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

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

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

a|9|?

***2 Error-состояние***

3\*

***1\****

***0***

a|9

a

a|9

**Рисунок 3.** Диаграмма ДКА ***A1  = A*(“a(a|9)\*”)** над **частным** алфавитом ***BId***

Элементы базиса ***BId*** в общем базисе ***B*:**

**a ::= l|u|a,**

**9::= 0|7|9,**

**?::= ?;**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***BId*** |  | **a ::=** | | | **9::=** | | | **?::=** |
| **Замыкание** | **q*i*\ a** | **a** | **l** | **u** | **0** | **7** | **9** | **?** |
| **{0}=[0]** | **0** | **1** | **1** | **1** |  |  |  |  |
| **{1,2,3}=[1] \*** | **1** | **3** | **3** | **3** | **3** | **3** | **3** |  |
| **{}=[{}]** | **2** |  |  |  |  |  |  |  |
| **{2,3}=[2] \*** | **3** | **3** | **3** | **3** | **3** | **3** | **3** |  |

**Таблица 3.** ДКА ***A1 =A*(“(l|u|a )( l|u|a|0|7|9)\*”)** над **общим** алфавитом ***B*,** допускающий

класс лексем **идентификаторов**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Замыкание** | **q*i*\ a** | **0** | **7** | **l** | **u** | **?** |
| **{0}=[0]** | **0** | **1** |  |  |  |  |
| **{1,2,3,4,6}=[1] \*** | **1** | **3** | **3** | **4** | **5** |  |
| **{}=[{}]** | **2** |  |  |  |  |  |
| **{2,3,4,6}=[2] \*** | **3** | **3** | **3** | **4** | **5** |  |
| **{4,6}=[4] \*** | **4** |  |  |  | **6** |  |
| **{5,6}=[5] \*** | **5** |  |  | **6** |  |  |
| **{6}=[6] \*** | **6** |  |  |  |  |  |

**Таблица 4.** ДКА ***A2 = A***(**“0(0|7)\* ((l|λ)(u|λ)|ul)”**)над **частным** алфавитом ***BOct***

l|u|0|7|?

*2 Error-состояние*

l

*4\**

u

0|7

*6\**

*5\**

u

l

l

u

*3\**

*1\**

*0*

0|7

0

**Рисунок 3.** Диаграмма ДКА ***A2 = A*(“0(0|7)\* ((l|λ)(u|λ)|ul)”)** над **частным** алфавитом ***BOct***

Элементы **базиса** ***BOct*** в **общем** адфавите ***B***:

**0::=0,**

**7::=7,**

**l::=l,**

**u::=u,**

**?::=9|a|?**

**Классы допустимых символов в данном случае в точности совпадают. Однако**

**класс недопустимых символов разлагается на три подкласса, и такое представление необходимо при определении ДКА над общим алфавитом** ***B*, чтобы можно было построить декартово произведение.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***BOct*** |  | **?::=** | **l::=** | **u::=** | **0::=** | **7::=** | **?::=** | |
| **Замыкание** | **q*i*\ a** | **a** | **l** | **u** | **0** | **7** | **9** | **?** |
| **{0}=[0]** | **0** |  |  |  | **1** |  |  |  |
| **{1,2,3,4,6}=[1] \*** | **1** |  | **4** | **5** | **3** | **3** |  |  |
| **{}=[{}]** | **2** |  |  |  |  |  |  |  |
| **{2,3,4,6}=[2] \*** | **3** |  | **4** | **5** | **3** | **3** |  |  |
| **{4,6}=[4] \*** | **4** |  |  | **6** |  |  |  |  |
| **{5,6}=[5] \*** | **5** |  | **6** |  |  |  |  |  |
| **{6}=[6] \*** | **6** |  |  |  |  |  |  |  |

**Таблица 5.** ДКА ***A2 =A*(“0(0|7)\* ((l|λ)(u|λ)|ul)”)**над **общим** алфавитом ***B*,** допускающий

класс лексем **целых констант с основанием 8**

Диаграмма ДКА ***A2 =*** ***A*(“0(0|7)\* ((l|λ)(u|λ)|ul)”)** над **общим** алфавитом ***B*** отличается от диаграммы ДКА над **частным** алфавитом ***BOct*** только тем, что дуги **(i, ?, j)** заменяются на

кратные дуги **(i, 9|a|?, j)**.

**Декартово произведение автоматов** ***A1* × *A2***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **2-вектор** | **q*i*\ a** | **a** | **l** | **u** | **0** | **7** | **9** | **?** |
| **0:0** | **0** | **1** | **1** | **1** | **2** |  |  |  |
| **1:2 \*Id** | **1** | **4** | **4** | **4** | **4** | **4** | **4** |  |
| **2:1 \*\*Oct** | **2** |  | **5** | **6** | **7** | **7** |  |  |
| **2:2 Error** | **3** |  |  |  |  |  |  |  |
| **3:2 \*Id** | **4** | **4** | **4** | **4** | **4** | **4** | **4** |  |
| **2:4 \*\*Oct** | **5** |  |  | **8** |  |  |  |  |
| **2:5 \*\*Oct** | **6** |  | **8** |  |  |  |  |  |
| **2:3 \*\*Oct** | **7** |  | **5** | **6** | **7** | **7** |  |  |
| **2:6 \*\*Oct** | **8** |  |  |  |  |  |  |  |

**Таблица 6. Декартово произведение** автоматов ***A1* × *A2*** над **общим**

алфавитом ***B* = {a, l, u, 0, 7, 9, ?}**

Для 0-состояния

g(0,**a|l|u**) = g(0:0, **a|l|u**) = 1:2 = 1

g(0,**0**) = g(0:0, **0**) = 2:1 = 2

g(0,**7|9|?**) = g(0:0, **7|9|?**) = 2:2 = 3

Для 1-состояния

g(1,**a|l|u**| **0|7|9**) = g(1:2, **a|l|u**| **0|7|9**) = 3:2 = 4; g(1,**?** )= g(1:2, **?**) = 2:2 = 3

Для 2-состояния

g(2,**a**) = g(2:1, **a**) = 2:2 = 3

g(2,**l**) = g(2:1, **l** ) = 2:4 = 5

g(2,**u** ) = g(2:1, **u**) = 2:5 = 6

g(2, **0|7**) = g(2:1, **0|7**) =2:3 = 7

g(2,**9|?**) = g(2:1, **9|?**) = 2:2 = 3

Для 3-состояния

g(3,**a|l|u**| **0|7|9|?**) = g(2:2, **a|l|u**| **0|7|9|?**) = 2:2 = 3

Для 4-состояния

g(4,**a|l|u**| **0|7|9**) = g(3:2, **a|l|u**| **0|7|9**) = 3:2 = 4; g(4,**?**) = g(3:2, **?**) = 2:2 = 3

Для 5-состояния

g(5,**a|l| 0|7|9|?**) = g(2:4, **a|l| 0|7|9|?**) = 2:2 = 3 ; g(5,**u**) = g(2:4, **u**) = 2:6 = 8

Для 6-состояния

g(6,**a|u| 0|7|9|?**) = g(2:5, **a|u| 0|7|9|?**) = 2:2 = 3; g(6,**l**) = g(2:5, **l**) = 2:6 = 8

Для 7-состояния

g(7,**a**) = g(2:3, **a**) = 2:2 = 3

g(7,**l**) = g(2:3, **l**) = 2:4 = 5

g(7,**u**) = g(2:3, **u**) = 2:5 = 6

g(7,**0|7**) = g(2:3, **0|7**) = 2:3 = 7

g(7,**9|?**) = g(2:3, **9**|**?**) = 2:2 = 3

Для 8-состояния: g(8,**a|l|u**| **0|7|9|?**) = g(2:6, **a|l|u**| **0|7|9|?**) = 2:2 = 3

***4\****

***1\****

a|l|u|0|7|9

a|l|u

a|l|u|0|7|9

a|l|u|0|7|9|?

***3 Error-состояние***

l

***5\*\****

u

0|7

***8\*\****

***6\*\****

u

l

l

u

***7\*\****

***2\*\****

***0***

0|7

0

**Рисунок 4.** Диаграмма автомата ***A1*** × ***A2***

**Классификация состояний и переходов**

Обозначим через ***M =*** ***A1*** **×** ***A2*** ***=(Q, B, g, q0 , F)*** – декартово произведение

автоматов ***A1*** и ***A2***; ***L1*** – класс **идентификаторов**, ***L2*** – класс **восьмеричных констант**

**языка Си.**

Автомат ***М*** допускает язык ***L1* ∪ *L2*** и так, как ***L1* ∩  *L2* = ∅,** то множество финальных состояний ***F*** представимо как объединение двух не пересекающихся подмножеств

***F1*** и ***F2*** :

***F*** = ***F1*** ∪ ***F2***, ***F1*** ∩ ***F2*** = ∅.

Тогда автомат ***Mi = (Q, B, g, q0 , Fi)*** допускает язык ***Li*** , ***i =* 1,2**.

Расширим область определения функции ***g*** по второму аргументу до множества

всех слов ***B\****:

1. ***g\**(q, λ**) **::= q** для всех **q** ∈ ***Q***;
2. ***g\**(q, *w*a**) **::= *g*(*g\**(q, *w***)**, a**) для всех **q** ∈ ***Q***, ***w* ∈ *B\**, a ∈ *B***.

Множество состояний распадается на **два непересекающихся класса**:

***Q = Active* ∪ *Error,***

где ***Error*** = {**q|** **q** ∈ ***Q,* *g\**(q, *w***) **∉ *F*** для всех слов ***w* ∈ *B\**** } и ***Active*** = ***Q*** \ ***Error*** **.**

По построению множество ***Error*** −одноэлементное. В данном примере ***Error*** ={**3**}.

Множество переходов суть множество дуг в автоматной диаграмме :

**{ (q, a, p)| p *= g*(q, a);q , p∈ *Q,* a ∈ *B*}**

или множество пар

***Q*** × ***B*** = **{(q, a)}**.

Множество переходов ***Q*** × ***B*** в общем случае, когда ***M =*** ***A1*** **×…×** ***A****n*, распадается

на***n***+2, ***n*** ≥ 1, попарно **непересекающихся классов:**

***Q*** × ***B* = *ActiveTransition* ∪ *EndL1* ∪…∪ *EndLn* ∪ *ErrorL*,**

где

***ActiveTransition*** = **{(q, a)| p *= g*(q, a)** и **q, p ∈ *Active*** **}** и

***EndLi* = {(q, a)| p *= g*(q, a)** и **q ∈ *Fi* , a ∈ *B* , p∈ *Error* }**, ***i =*** 1,***n***;

***ErrorL*** = **{(q, a)| p *= g*(q, a)** и **q ∈ *Active* \ *F* , a ∈ *B* , p∈ *Error* }**.

Сопоставим классам переходов множество ***P*** «*действий*» (*семантических процедур*):

***P*** = { ***fActiveTransition***, ***fEndL1*** , **…** , ***fEndLn*** , ***fErrorL*** }.

Определим функцию выхода ***f*** **:** ***Q*** × ***B*** **→ *P***. Если ***f*(q, a)** = ***h*** ∈ ***P***, тогда ***h*** - «действиe»,

выполняемое лексическим анализатором в состоянии **q** при входном символе **a ∈ *B***,

более точно, − ***при любом входном символе с ∈ Σ из класса, который представляет a.***

Здесь ***P*** – выходной алфавит, символы которого суть *имена семантических процедур.*

Фактически результатом проектирования лексического анализатора является специальный автомат с выходом, называемый ***преобразователем простого типа***:

***M***  ***= (Q, B, P, g, f, q0 , F)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **2-вектор** | **q*i*\ a** | **a** | **l** | **u** | **0** | **7** | **9** | **?** |
| **0:0** | **0** | **1 */ f*** | **1 */ f*** | **1 */ f*** | **2 */ f*** | ***/ err*** | ***/ err*** | ***/ err*** |
| **1:2 \*Id** | **1** | **4 */ f*** | **4 */ f*** | **4 */ f*** | **4 */ f*** | **4 */ f*** | **4 */ f*** | ***/ f1*** |
| **2:1 \*\*Oct** | **2** | ***/ f2*** | **5 */ f*** | **6 */ f*** | **7 */ f*** | **7 */ f*** | ***/ f2*** | ***/ f2*** |
| **2:2 Error** | **3** |  |  |  |  |  |  |  |
| **3:2 \*Id** | **4** | **4 */ f*** | **4 */ f*** | **4 */ f*** | **4 */ f*** | **4 */ f*** | **4 */ f*** | ***/ f1*** |
| **2:4 \*\*Oct** | **5** | ***/ f2*** | ***/ f2*** | **8 */ f*** | ***/ f2*** | ***/ f2*** | ***/ f2*** | ***/ f2*** |
| **2:5 \*\*Oct** | **6** | ***/ f2*** | **8 */ f*** | ***/ f2*** | ***/ f2*** | ***/ f2*** | ***/ f2*** | ***/ f2*** |
| **2:3 \*\*Oct** | **7** | ***/ f2*** | **5 */ f*** | **6 */ f*** | **7 */ f*** | **7 */ f*** | ***/ f2*** | ***/ f2*** |
| **2:6 \*\*Oct** | **8** | ***/ f2*** | ***/ f2*** | ***/ f2*** | ***/ f2*** | ***/ f2*** | ***/ f2*** | ***/ f2*** |

**Таблица 7.** Автомат ***M***  ***= (Q, B, P, g, f, q0 , F)*** - преобразователь простого типа

Выходной алфавит ***P*** **= { *fActiveTransition = f*, *fEndL1 = f1*, *fEndL2 = f2*, *fErrorL = err* }**

***F = F1* ∪ *F2*, *F1* = {1,4} *F2* = {2,5,6,7,8}**

**Неформальное описание семантических процедур**

**1)**

*Предусловие*: **лексический анализатор** находится в текущем состоянии **q ∈ *Active*;**

**сканировал очередной символ** **c** ∈ Σ из входного потока и классифицировал как символ эквивалентный символу **а; p = g(q, a) ∈ *Active***

***fActiveTransition* = *f*(q, a):**

**Назначить состояние p текущим ;**

*Постусловие*: лексический анализатор находится в текущем состоянии **q =p;**

**готов сканировать** очередной символ **c** ∈ Σ из входного потока**;**

**Возвратить управление лексическому анализатору.**

**2)**

*Предусловие*: **лексический анализатор** находится в текущем состоянии **q ∈ *Active* \ *F* ;**

сканировал очередной символ **c** ∈ Σ из входного потока и классифицировал как символ эквивалентный символу **а;**  **p = g(q, a) ∈ *Error***

***fErrorL* = *f*(q, a):**

**Регистрировать Лексическую Ошибку (позицию ошибочного** символ **c** ∈ Σ **и т.д.);**

**Назначить состояние q*0* текущим;**

*Постусловие***: лексический анализатор** находится в текущем состоянии **q = q*0*;**

**готов сканировать** очередной символ **c** ∈ Σ из входного потока **для очередной лексемы;**

**Возвратить управление лексическому анализатору.**

**3)**

*Предусловие*: **лексический анализатор** находится в текущем состоянии **q ∈ *Fi* ;**

сканировал очередной символ **c** ∈ Σ из входного потока и классифицировал как символ эквивалентный символу **а**; **p = g(q, a) ∈ *Error*;**

**выделена лексема** ***w*** ∈ ***Li***

***fEndLi*** = ***f*(q, a):**

**Сохранить лексему *w* ∈ *Li* в *таблице символов***;

**Вывести токен** <***Li, w****>* в **выходной поток токенов**;

**Возвратить текущий символ** **c** ∈ Σ **обратно во входной поток**;

**Назначить состояние q*0* текущим ;**

*Постусловие***: лексический анализатор** находится в текущем состоянии **q = q*0*;**

**готов сканировать** очередной символ **c** ∈ Σ из входного потока **для очередной лексемы;**

**Возвратить управление лексическому анализатору.**

Замечание.

В состояние **2:2 (Error)** Лексический анализатор **никогда не переходит**!