**Пересечение разбиений**

Языки*Li* индуцируют разбиения Π*i*алфавита *A***,** *i*  [1,3]**:**Π1 **=** {[\_a-zA-Z], [0-9], [^0-9\_a-zA-Z]};   
Π2 **=** {[0], [1-7], [lL], [uU], [^01-7lLuU]},  
Π3 **=** {[\x20\x09], [^\x20\x09]}.  
Абстрактные алфавиты, соответствующие разбиениям, следующие:  
*B*1 = {*a*, 9, ? }, *B*2 = {0, 7, l, u, ? }, *B*3 = {*s*, ? }, Регулярные выражения *еi* представляют языки *Li* в соответствующих абстрактных алфавитах *Bi* , *i*  [1,3] :*e*1 **=** а(а | 9)\*,*e***2 =** 0(0 | 7)\* (λ|u|l|ul|lu), *e*3 **=** *ss* **\***.

Вычислить Π1,2 = Π1 ∩ Π2 и выразить классы разбиений Π1  и Π2 через классы разбиения Π1,2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Π1  \ Π2 | [0] | [1-7] | [lL] | [uU] | [^01-7lLuU] |
| [\_a-zA-Z] |  |  | [lL] | [uU] | [\_a-km-tv-zA-KM-TV-Z] |
| [0-9] | [0] | [1-7] |  |  | **[89]** |
| [^0-9\_a-zA-Z] |  |  |  |  | [^0-9\_a-zA-Z] |

Π1,2= {[0], [1-7], [lL], [uU], **[89],** [\_a-km-tv-zA-KM-TV-Z], [^0-9\_a-zA-Z]}

Π1,2,3  = Π1,2 ∩ Π3 = {[0], [1-7], [lL], [uU], **[89],** [\_a-km-tv-zA-KM-TV-Z], [\x20\x09],[^0-9\_a-zA-Z\x20\x09]}

B = {0, 7, 9, l, u, a, s, ?} − aбстрактный алфавит, соответствующий разбиению Π1,2,3 .

Π1 через Π1,2,3

[\_a-zA-Z] = [\_a-km-tv-zA-KM-TV-Z] | [lL] | [uU]

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

[^0-9\_a-zA-Z] = [\x20\x09] | [^0-9\_a-zA-Z\x20\x09]

B1 через B

a = a|l|u

9=0|7|9

?=s|?

Π2 через Π1,2,3 -?

B2 через B -?

Π3 через Π1,2,3 -?

B3 через B -?

Декартово произведениедвух детерминированных конечных автоматов **−** aвтомат *M*1,2 = *M*1× *M*2, допускающий объединение исходных языков:

*L*1 ** *L*2.*M*1,2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вектор | *q\a* | 0 | 7 | 9 | a | l | u | *s* | ? |
| (0,0) | 0 | 1 | ∅ | ∅ | 3 | 3 | 3 | ∅ | ∅ |
| (2,1) 2 | 1 | 4 | 4 | ∅ | ∅ | 5 | 6 | ∅ | ∅ |
| (2,2)= ∅ | 2 | ∅ | ∅ | ∅ | ∅ | ∅ | ∅ | ∅ | ∅ |
| (1,2) 1 | 3 | 7 | 7 | 7 | 7 | 7 | 7 | ∅ | ∅ |
| (2,3) 2 | 4 | 4 | 4 | ∅ | ∅ | 5 | 6 | ∅ | ∅ |
| (2,4) 2 | 5 | ∅ | ∅ | ∅ | ∅ | ∅ | 8 | ∅ | ∅ |
| (2,5) 2 | 6 | ∅ | ∅ | ∅ | ∅ | 8 | ∅ | ∅ | ∅ |
| (3,2) 1 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | ∅ | ∅ |
| (2,6) 2 | 8 | ∅ | ∅ | ∅ | ∅ | ∅ | ∅ | ∅ | ∅ |

**Классификация состояний и переходов.**   
По построению все состояния достижимы из начального состояния *q*0.

**Классификация состояний**

Множество «тупиковых» или состояний, сигнализирующих о лексической ошибке, − **Error**  = {*q*| *g*\*(*q*, *x*) ∉*F*}.

**Error = ?**

Множество активных состояний − **Active**  = *Q*\ **Error**.

**Active = ?**

**Классификация переходов.**  
 Множество активных переходов −  
**ActiveTransition** = {(*q*, *a*)| *q* ∈**Active** и *g*(*q*, *a*) ∈**Active,** *a* ∈ *B* }.

*f*(*q*,*a*) = *p*ActiveTransition, если **(*q*,*a*)∈ActiveTransition**;

*M*1,2[*q*,*a*] = *M*1,2[*q*,*a*] ***+p*ActiveTransition**

Множество переходов, определяющих условие распознавания лексемы класса *Li* , − **EndL*i*** = {(*q*, *a*)| *q* ∈***Fi*** и *g*(*q*, *a*) ∈**Error,** *a* ∈ *B* }.

*f*(*q*,*a*) = *p*EndL*i*, если(*q*,*a*) ∈ **EndL***i* , *i* **∈** [1, *n*];

*M*1,2[*q*,*a*] =  *M*1,2[*q*,*a*] *+* ***p*EndL*i***

Символ *a* такой, что (*q*, *a*) ∈ **EndL*i*** определяет правый контекст лексемы *x* ∈ *Li*.Слово *x* переводит автомат из начального состояния в финальное состояние*q* ∈ *Fi*, причем *q* **=** *g*\*(*q*0, *x*)**.**

Переходы множества **ErrorL** = {(*q*, *a*)| *q* ∈**Active** \ *F* и *g*(*q*, *a*) ∈**Error,** *a* ∈ *B* } определяют условия обнаружения лексической ошибки.

*f*(*q*,*a*) = *p*ErrorL, если(*q*,*a*) ∈ **ErrorL**;

*M*1,2[*q*,*a*] =  *M*1,2[*q*,*a*] ***+ p*ErrorL**

Если (*q*, *a*) ∈**ErrorL**, тогда слово *x* такое, что *q*  **=** *g*\*(*q*0, *x*) переводит автомат из начального состояния вактивное не финальное состояние *q*, определяет допустимый собственный префикс некоторой лексемы, но

слово *xa* – не допустимый префикс, так как не существует слова *y* ∈ *B***\***, что *xaу* ∈ *L*.

Очевидно**, ActiveTransition**, **EndL**1 , **EndL**2, **ErrorL**, **−** попарно не пересекающиеся множества переходов.

**Семантические процедуры**

Определить (классифицировать) множество *P* «действий» (*семантических* *процедур*)значитопределить функцию выхода *f : Q*×*B′ → P*,где*B*′= *B* ∪ **{**#}.Если *f*(*q*,*a*)=*p* ∈ *P***,** тогда*p* – действие, выполняемое лексическим анализатором в состоянии *q* ∈ *Q*привходном символе*a* ∈ *B*′. **Формально символ # ∉ *B* служит признаком конца входного потока.**  
 Фактически результатом проектирования является разработка модели детерминированного конечного преобразователя*M* = (*Q*, *B*′, *P*, *f, g*)простого типа.

Пусть *P* = { *p*ActiveTransition, *p*EndL1, … , *p*EndL*n*, *p*ErrorL}. Определим очевидным образом функцию выхода:

*f*(*q*,*a*) = *p*ActiveTransition, если **(*q*,*a*) ∈ ActiveTransition**;

если(*q*,*a*) ∈ **EndL***i* , то (*q*,*#*) ∈ **EndL***i* , и   
*f*(*q*,*a*) = *p*EndL*i*, если(*q*,*a*) ∈ **EndL***i* , *i* **∈** [1, *n*];

если(*q*,*a*) ∈ **ErrorL**, то(*q*,*#*) ∈ **ErrorL**, и

*f*(*q*,*a*) = *p*ErrorL, если(*q*,*a*) ∈ **ErrorL**.

Тогда программная реализация (интерпретация) преобразователя *M* =(*Q*, *B*′, *P*, *f, g*) определяет основные действия (функции) лексического анализатора при сканировании входного потока символов алфавита *B*′.