**CMPS 455, Handout No.1**

**ALPHABETS and LANGUAGES**

**Definition.** An ***Alphabet*** is a finite set of symbols denoted by Greek letter (Sigma)Σ.

**Example**

1. English alphabets: Σ ={a,b,c,…..,z,A,B,C,….,Z}
2. Binary alphabets: Σ ={0,1}

**Definition**. Given Σ, a ***word*** over Σ is a string of symbols to create w

**Example**

1. If Σ={a,b}, then w1=abb is a word over Σ. w2 =baba is a word over Σ
2. Σ={ a, ba }, then w1=ba is a word over Σ, but word ab is not a word over Σ

**Definition**. Give Σ, and word w over Σ. the ***length*** of w or |w| is the number of symbols used to form w

**Example**

1. Σ ={a,b}, then the length of word w=aba or |w|= 3 ( symbols a, b, and a )
2. Σ ={a, ba}, then for word w=aba, |w|= 2 ( symbols a and ba)

**Definition**. A word of length zero or ***null word*** is denoted by Greek letter λ (lambda). Therefore |λ|=0. If λ is part of a word, you can ignore it. For example, w=aλbλb= abb.

**Definition.** Given word w, then w0=λ, w1=w, w2=ww, wn=www…w. Hence, wn means ***concatenation of*** w to itself n times.

**Example**.

1. Σ={ a, b}. Then for w=ab, w3=(ab)3 = ababab. Notice that (ab)3 a3b3=aaabbb
2. Σ ={ a, b, c}. Then (abc)2 = abcabc and a2b3=aabbb

**Words factoring**. Given w1=ab and w2=a. Then w1w2=aba (***concatenation of w1 and w2***)is ONE word, but w1+w2 =ab +a reads as: ab or a is the ***union of w1 and w2(TWO words)***. Hence w1+w1 = ab+ab=ab

1. Let w1+w2 = ab + a, since word ab begins with a, and also word a begins with a, therefore both have a in common on the left-hand-side, factor them by a: ab+ a = a( b + λ ). To check your work, a(b+λ) = ab + aλ = ab + a
2. Let w1=ab and w2=bb. Both words end up at b, so we can factor b on the right-hand-side to get ab+bb = (a+b)b
3. Let w1=abc, w2=adc. Both w1 and w2 begins with a, so we factor with a on the left-hand-side to get abc+adc =a(bc + dc). The words in parenthesis have c in common on their right-hand-side, factor by c to get abc + adc = a (bc + dc )= a(b+d)c

**Definition.** Given Σ. The set of all words over Σ including λ is denoted by **Σ\***.

**Example**

Given Σ = { a, b }, to not to miss any word, we list them by their length

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| words | λ | a,b | aa,ab,ba,bb | aaa,aab,aba,abb,baa,bab,bba,bbb | ……… |
| length | Length 0 | Length 1 | Length 2 | Length 3 | ………… |

Therefore Σ\*={ λ, a,b,aa,ab,ba,bb, …….. }

**Definition**. Given Σ, any subset of Σ\* is called a ***Language***

**Example**

Let Σ={a,b}, then Σ\*={ λ, a,b,a2,ab,ba,b2, …….. }.

|  |
| --- |
| 1. Language L1 = {all powers of a including 0 power }= { λ, a, aa, aaa,aaaa, ….}= {a0,a1,a2,a3,…} we name this language a\* or a language whose words are all powers of a including the zero power, so {a0,a1,a2,a3,…} =a\* |
| 1. Language L2 ={ all powers of a excluding 0 power }={ a, aa, aaa,aaaa, ….}= { a1,a2,a3,…}   We can factor by a on the left-hand-side: =a{ λ, a1,a2,a3,…} =aa\*, or  We can factor by a on the right-hand-side: { λ, a1,a2,a3,…} a =a\*a  Therefore aa\*=a\*a = a+, thus { a1,a2,a3,…} = aa\* =a\*a = a+ |
| 1. L3 = { λ, ab, abab, ababab, ……. } = { (ab)0, (ab)1, (ab)2, …. } = (ab)\* |
| 1. L4={ λ, b, bb, bbb, ……. } = { b0, b1, b2, b3, …. } = b\* |

**Examples.** True or false?

|  |  |  |
| --- | --- | --- |
| a3 ϵ a\*, true  a\*= { λ,a,a2,a3,…..} | a2bϵ a\*b\*, true  a\*b\* = {λ,a,a2,..}{λ,b,b2,b3,..}  ={λ,a,b,ab,a2b,… } | a2bϵ a\*+ b\*, false  a\* + b\* ={λ,a,a2,… }U{λ,b,b2,… }  ={λ,a,b,a2,b2,a3,b3……..}  a2b is not in a\*+b\* |
| a\*b\* = a\*+ b\*, false  ab ϵ a\*b\* but not in a\*+ b\* | b3 ϵ a\*b\*, true  a\*b\* = λb\* = b\*={λ,b,b2,b3,..} | a2b3a2 ϵ a\*b\*a\*, true  a2b3a2 |

**Example.** Expand (a+b)\* to show the words of the language

(a+b)\* = { (a+b)0, (a+b)1, (a+b)2 , ………………….. }

={ λ , a,b, (a+b)(a+b),…………….}

={ λ , a, b, aa,ab,ba,bb,…………..}

={ λ , a,b, a2,ab,ba,b2, ………………..},

which is the set of all powers of a, all powers of b, and any combinations of a’s and b’s. All of the following are true:

a3 ϵ (a+b)\*, b5 ϵ (a+b)\*, a3b2 ϵ (a+b)\*,a2b3a3 ϵ (a+b)\*, abb ϵ (a+b)\*, λ ϵ (a+b)\*,

**FINITE AUTOMATA (FA)**

Suppose we want to design a vending machine to accept 5, 10, and 25 cents(₵ )coins and return items which worth a total of 10, 20, or 25 cents. To make it a simple machine, assume the machine does not return any change and you can get something back from the machine if the exact chain of coins your drop in the machine is exactly the price of the item you want to receive.

Initial state Final state Final state Final state

start 5₵ 5₵ 5₵ 5₵ 5₵

10₵ 10₵

10₵ 10₵

25₵

This is an example of a Finite Automata ( or machine with finite number of states) . Finite Automata( FA) consist of the following components :

1. **Set of states** ={ q0, q1, q2,q3, q4, q5 }, with only ONE initial state (q0), one or more final states (q2,q3, q5, states you can get items back from the machine), and zero or more NULL states (q1,q4, states are not final nor initial state )
2. Set of inputs= **alphabets:** Σ={5₵, 10₵, 25₵ }, these are the only type of coins the machine accepts.
3. **Machine** **language: L**= set of chain of coins you can insert in the machine to get an item back from the machine=set of outcomes={ we list them by the number of coins in each chain}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| String of coins (words) | (25) | (5)(5), (10)(5), (5)(10) | (5)(5)(5),(5)(10)(10).. |  | (5)(5)(5)(5)(5) |
| No. of coins | 1 | 2 | 3 | …….. | 5 |

L={ (25), (5)(5), (10)(5), (5)(10),……., (5)(5)(5)(5)(5)}

NOTE, pay attention to (10)(5) and (5)(10) chains. They are two different form of dropping coins in the machine and that’s why we have to consider them as two different chains of inputs.

1. **Set of rules:** Rules to go from one state to another state. There are two methods to provide these rules:

|  |  |
| --- | --- |
| **Transition table** | **Grammar** |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | q0 | q1 | q2 | q3 | q4 | q5 | | q0 |  | 5 | 10 |  |  | 25 | | q1 |  |  | 5 | 10 |  |  | | q2 |  |  |  | 5 | 10 |  | | q3 |  |  |  |  | 5 | 10 | | q4 |  |  |  |  |  | 5 | | q5 |  |  |  |  |  |  |   The highlighted 10, means from q1 coin 10₵ will take us to state q3 | <q0>🡪 5<q1>  Means from state q0, if we drop 5₵ we will enter state q1  <q1>🡪10<q3>  Means from q1, if we drop 10₵ we will enter state q3 |

**Notations:**

1. If you use a lower case letters to name a state, then in grammar rules we enclosed the name of the state within < and >. If you use upper case letters the < and > are not required anymore.

**Example**

|  |  |
| --- | --- |
| using lowercase letters to name states:q0,q1 | using uppercase letters to name states: A,B |
| 5₵  q0 q1 , <q0>🡪5<q1> | 5₵  A B , A🡪5B |

1. To identify a final state in grammar, we use this notation

|  |  |
| --- | --- |
| More grammars of the above FA | |
| <q0>🡪5<q1>  <q0>🡪10<<q2>  <q1>🡪5<q2>  <q1>🡪10<q3>  <q2>🡪5<q3>  <q2>🡪10<q4>  <q2>🡪λ, q2 is a final state | <q3>🡪5<q4>  <q3>🡪10<q5>  <q3>🡪 λ, q3 is final state  <q4>🡪5<q5>  <q5>🡪λ ,q3 and q5 are final states |

1. Instead of labeling “Initial state” and “Final state”, we will use the following notations:

|  |  |  |
| --- | --- | --- |
| Name | symbol | Preferred symbol |
| Initial state | Start | * , minus sign |
| Final State |  | **+ ,** plus sign |
| Initial and final. | Start | **± ,**plus minus |
| Null state |  |  |

**Definition**. Given ∑ ={a,b}, then ∑\* ={ (a,b)0,(a,b)1,(a,b)2……} ={ λ, a,b, a2, ab,ba, b2, ……..}. Any subset of ∑\* is called a ***language*** . Method to find all words of length 3

aaa=a3

a

a aab=a2b

aba

b

abb=ab2

baa=ba2

a

b bab

bba=b2a

b

bbb=b3

Note. (a,b)2 = (a,b)(a,b) = aa+ab+ba+bb = a2 +ab+ba+b2

(a,b)3 =(a,b)(a,b)(a,b) = (a,b)(a,b)2 =

=(a,b)(a2+ab+ba+b2)

=a3+a2b+aba+ab2+ba2+bab+b2a+b3

**Examples:**

L1=set of all power of a including the zero power={a0, a1, a2, a3, ….}

={λ a1, a2, a3, ….}=a\*, so a\* is a language whose words are all powers of a including zero power

L2= set of powers of a except zero power={ a, a1, a2, a3, ….}= a{λ, a1, a2, a3, ….}=aa\*

Also L2=={ a, a1, a2, a3, ….}=factor a from right-hand-side= {λ, a1, a2, a3, ….}a=a\*a.

Therefore aa\* = a\*a

L3= set of all powers of (ab)= { (ab)0, (ab)1, (ab)2, …..}={ λ, (ab), (ab)2, (ab)3, …}=(ab)\*

L4= set of all words begin with a and followed by powers of b =ab\*

**FAS , LANGUAGES AND GRAMMARS**

Examples to understand the concepts of FAs , grammars and languages

Example . Find the language and grammar of the following FAs

|  |  |  |
| --- | --- | --- |
| FA | Language | Grammar |
| a   * +   A B | ∑={ a}, only a takes us from  Initial state to final state  Language=L={a} | A🡪aB, from A input a goes B  A🡪λ , A is a finial state |
| a b   * +   A B | ∑={a, b}  L={a, ab, abb, abbb, …. }  ={ a, ab, ab2, ab3, ……}  =a{λ,b, b2, b3, …..}=ab\* | A🡪aB  B🡪bB, at B input b goes back to B  B🡪λ , B is final state |
| a  ±  A | ∑={a}  L={**λ** ,a, a2, a3 , …..}=a\*  Note. When initial state is also final state, then λ is a word in the language | A🡪aA  A🡪λ |
| a b  ±  X | ∑={a,b}  L={λ, a, b, aa,ab,ba,bb,…}  ={(a+b)0, (a+b)1, (a+b)2,.}  =(a+b)\*  Means all combinations of a’s and b’s in any order | X🡪aX  X🡪bX  X🡪λ |
| a b  b a   * + +   A B C | ∑={a,b}  FA has 2 final states, means the language has 2 parts. One ends up at B (L=a\*b) and the other ends up at C (L=a\*bab\*). So all together L=a\*b + a\*bab\* =a\*b( λ +ab\* | A🡪aA  A🡪bB  B🡪λ  B🡪aC  C🡪bC  C🡪λ |
| B b  a +  -  X b  +  A c | ∑={a,b,c}  L= ab\* + bc\* | X🡪aB  X🡪bA  B🡪bB  B🡪λ  A🡪cA  A🡪λ |
| b a  + ± + b  A B C | 3 final states, language has 3 parts: stop at A,L=b. stop at B, L=λ, stop at C,L=ab\*. So  L=b + λ + ab\* | B🡪aC, B🡪bA, B🡪λ  C🡪bC, C🡪λ  A🡪λ |

So far we did some example to find the language and grammar when FA is given. Now let’s look at some examples in which the grammar is given and we want to find the FA and the language of that grammar

|  |  |  |
| --- | --- | --- |
| **Given: Grammar** | **Find: FA** | **Find: Language** |
| A🡪aA, A🡪bB  B🡪bB, B🡪aC, B🡪λ  C🡪cC, C🡪λ  3 states: A,B,C | a b c  a a  A B C | ∑={a,b,c}  2 final states:  L1=a\*b, L2=a\*bb\*ac\*  L=L1+L2  =a\*b + a\*bb\*ac\*  =a\*b( λ +b\*ac\*) |
| X🡪bB  B🡪bB, B🡪λ   1. states: X and B | b  b +  X B | ∑={a.b}  L1=λ ,X is final  L2=bb\* ,B is final  L=L1+L2  =λ + bb\* |
| X🡪aA, X🡪bY  A🡪aA, A🡪bY, A🡪λ  Y🡪aY, Y🡪bY, Y🡪λ  3 states: X, A, Y | A a    a b a   * b + * b   X Y | ∑={a,b }  1 final but 2 ways to get to it  L1=b(a+b)\*  L2=aa\*b (a+b)\*  L=L1+L2  =aa\*b(a+b)\* + b(a+b)\* |
| **Given: Language** | **Find : FA** | **Find: Grammar** |
| L=ab\*c(a+b)\*  ∑={a,b,c} | b a,b  - a c +    a b a,b  Note: same as | A🡪aB  B🡪bB, B🡪cX  X🡪aX, X🡪bX, X🡪λ |
| L= a(a+b)\* + b(a+b)\*  ∑={a,b} | a b  + - +  a b a b  X A Y | A🡪aX, A🡪bY  X🡪aX, X🡪bX, X🡪λ  Y🡪aY , Y🡪bY, Y🡪λ |
| L= aa\* + b(a+b)\*  ∑={a,b} | a A X a  + +  b  a b   * - B | B🡪aA, B🡪bX  A🡪aA, A🡪λ  X🡪aX, X🡪bX, X🡪λ |

Now, let’s put all these cases together and complete the following table. The shaded boxes are given, complete the table by filling out all empty boxes

|  |  |  |
| --- | --- | --- |
| **Language** | **FA** | **Grammar** |
| L=ab\* + ba\* | (i) | (ii) |
| (iii) | X  -  a b  + b a  A B | (iv) |
| (v) | (vi) | A🡪bX, A🡪aA, A🡪λ  X🡪aB, X🡪λ  B🡪bB, B🡪λ |

Solutions.

|  |  |
| --- | --- |
| 1. X   -  a b  b a  + +  B A | 1. X🡪aB, X🡪bA   B🡪bB, B🡪λ  A🡪aA, A🡪λ |
| 1. L1=a A is final   L2=aba\* + ba\*, B is final, 2 ways  to get to B  L=L1+L2= a+aba\*+ ba\* | 1. X🡪aA, X🡪bB   A🡪bB, A🡪λ  B🡪aB, B🡪λ |
| A is final: from A to A ,L1=a\*  X is final: from A to X, L2=a\*b  B is final: from A to B,L3=a\*bba\*  L=L1+L2+L3=a\*+a\*b+a\*bba\* | 1. First find FA and then use FA to write the language. FA has 3 final states: A, X, B   A ± a  b  + b + a    X B |

The following are the same. Use them when you want to simplify FAs

|  |  |
| --- | --- |
| a b | a+b |
| a    b | a+b |
| a  b c | ba\*c |
| a  b c  d | ba\*c  ba\*d |

Example. Simplify and find the language of this FA

|  |  |
| --- | --- |
| a b c  d e f g | abf\*c + def\*g  L=abf\* + def\*g |



1. Find the language of each CFG. Write your answers in the space provided
2. S🡪aA | bB

A🡪aA | λ

B🡪bB | λ

Answer ……………………

ii S🡪aS | bX | λ

X🡪aX | bX | λ

Answer ………………………………

1. **Programming assignment**

Write a program to read a postfix expression and display its numeric value. Suppose a=5,b=7,c=2,d=4

***Sample Input/Output***

Enter a postfix expression with $ at the end: ab+cd\*+$

Value = 20

CONTINUE(y/n)? y

Enter a postfix expression with $ at the end:abcd+++$

Value = 18

CONTINUE(y/n)? y

Enter a postfix expression with $ at the end:abcd\*-\*$

Value = -5

CONTINUE(y/n)? n

Directions. Include the following information at the beginning of your program

//-------------------------------------------------------------------------------------------------------------

// Group names: Smith, John and Brown, Anna

// Assignment : No.1

// Due date : September 9

// Purpose: this program reads an expression in postfix form, evaluates the expression

// and displays its value

//-----------------------------------------------------------------------------------------------------------------

Comment all function and class members. Copy and paste your program sample run to the end of your program. Answer the quiz question(s) and make all one words or PDF document and email it to: [mahmadnia@laverne.edu](mailto:mahmadnia@laverne.edu)

In the subject box of your email, please type CMPS455\_assig\_1\_lastname1\_lastneme2.

Computer science 455

Quiz No.1 (q1) Names …………………………………………………………………..

1. True or false

|  |  |  |  |
| --- | --- | --- | --- |
| 1. L=a\*ba\* +b\* | ab is a member of L  ba is a member of L  aaab is a member  λ is a member of L | T  T  T  T | F  F  F  F |
| 1. L=a\*b\* | ab is a member of L  b4 is a member of L  b2a3 is a member of L | T  T  T | F  F  F |
| 1. L=a\* + b\* | a4 is a member of L  a5b5 is a member of L | T  T | F  F |
| 1. L=(a + b)\* | a is a member of L  ba is a member of L  aba is a member of L | T  T  T | F  F  F |

1. Complete the following table

Language FA CFG

|  |  |  |
| --- | --- | --- |
| L= a\*b\*c\* |  |  |
|  | b    a  b |  |
|  |  | S 🡪 aA |bB  B🡪bB |aA  A🡪aA | bA | λ |