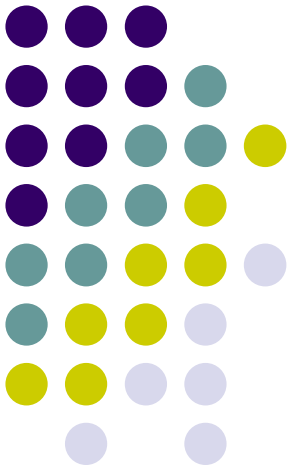
CS 460

Programming Languages

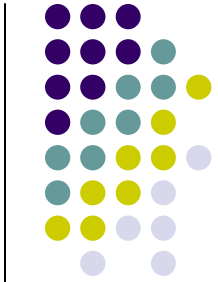
Fall 2008

Dr. Watts

(Week 4)



yourlastname.mail



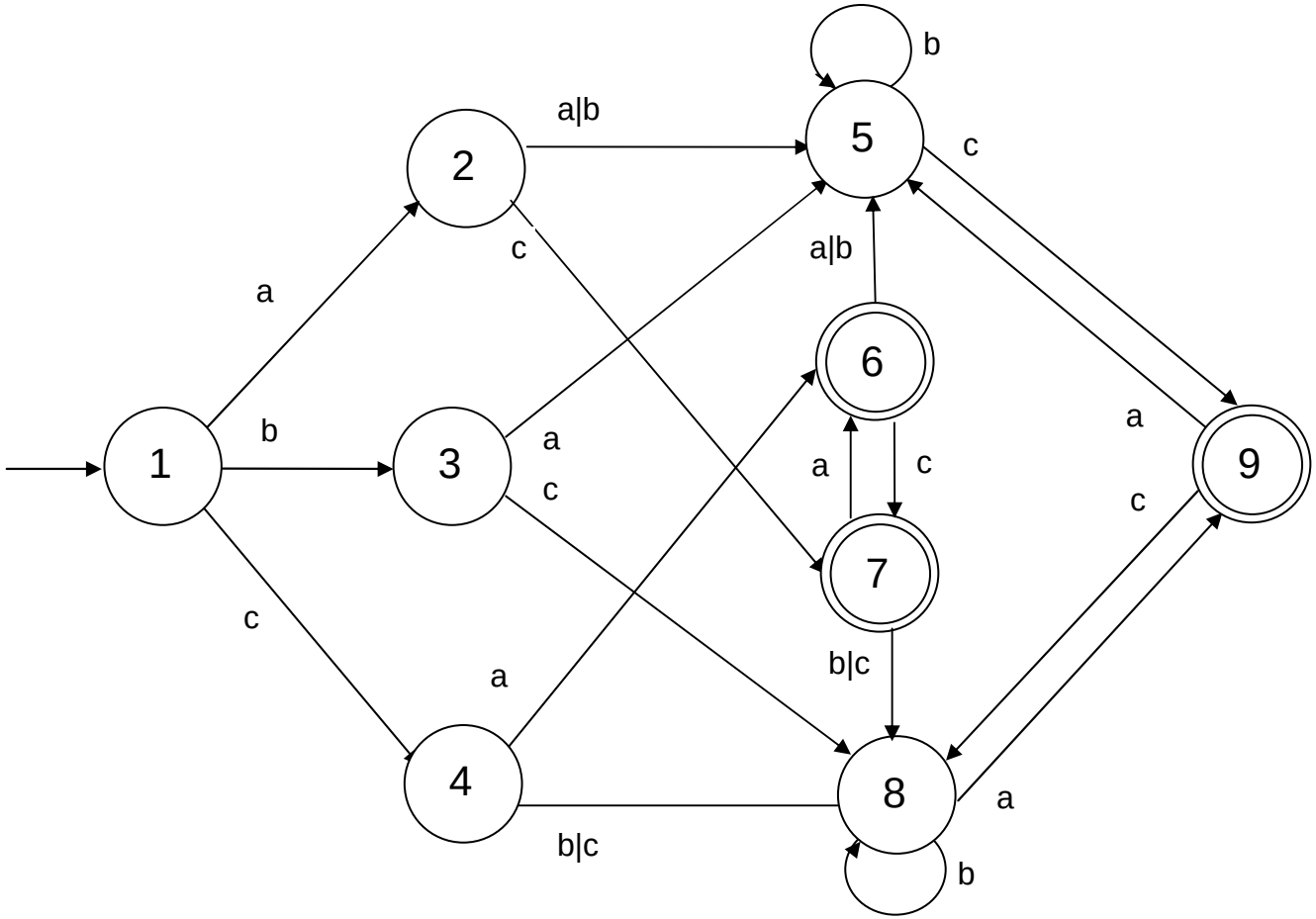
- Please create a new file called `yourlastname.mail`
- It should contain one line:

  `you@favorite.isp # First Last`
- New line (return) at end of line
- Copy to `~tiawatts/cs460drop`

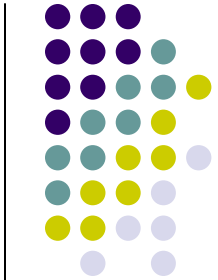
```
chmod 644 yourlastname.mail
umask 033
cp yourlastname.mail ~tiawatts/cs460drop/.
umask 077
```

- Thank you

$(a|b|c|\lambda)((ab^*c)|(cb^*a))^+$

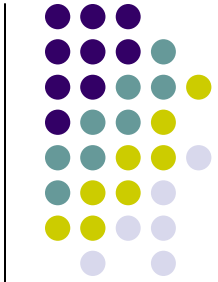


$(a|b|c|\lambda)((ab^*c)|(cb^*a))^+$



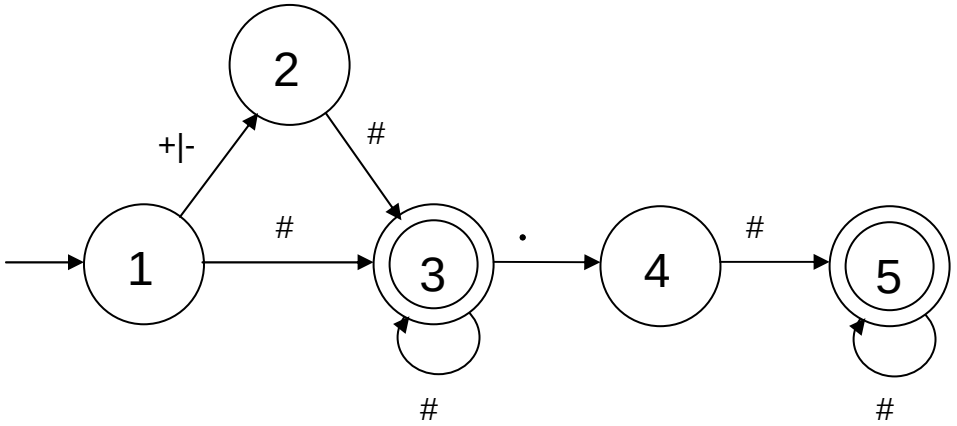
State	a	b	c
1	2	3	4
2	5	5	7
3	5		8
4	6	8	8
5		5	9
6*	5	5	7
7*	6	8	8
8	9	8	
9*	5		8

Regular Expression for Numeric Literals



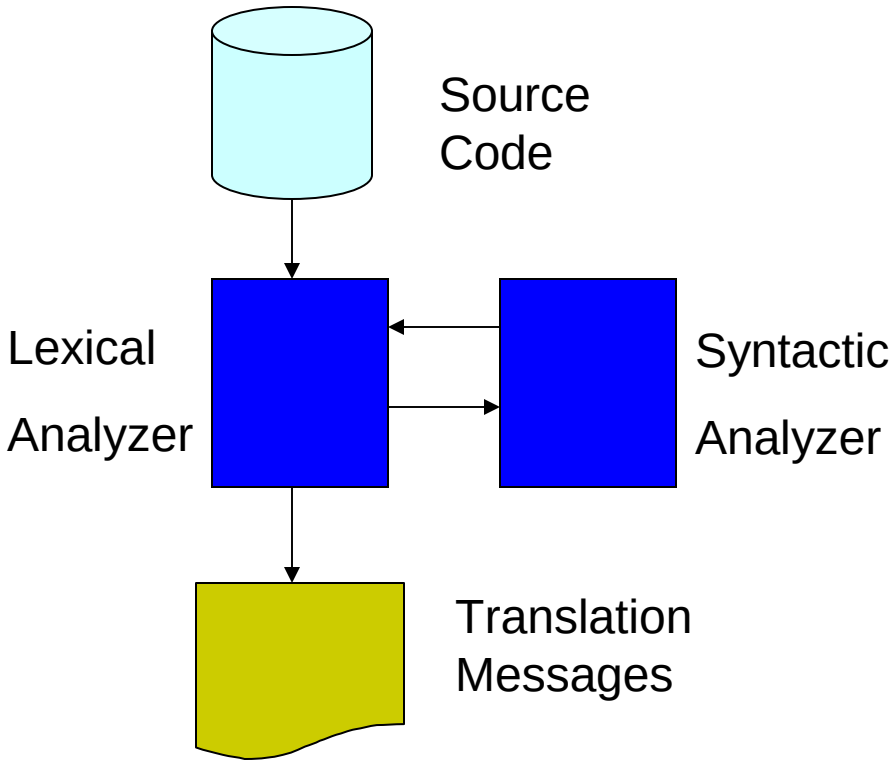
- Regular expression for general class of numeric literals
  - (+|-|λ)(0|1|2|3|4|5|6|7|8|9)$^{+}$(. (0|1|2|3|4|5|6|7|8|9)$^{+}$|λ)
- DFA?
- How do you recognize the end of a numeric literal?

DFA for Numeric Literals
(+|-|λ)(0|1|2|3|4|5|6|7|8|9)+(.(0|1|2|3|4|5|6|7|8|9)+| λ)

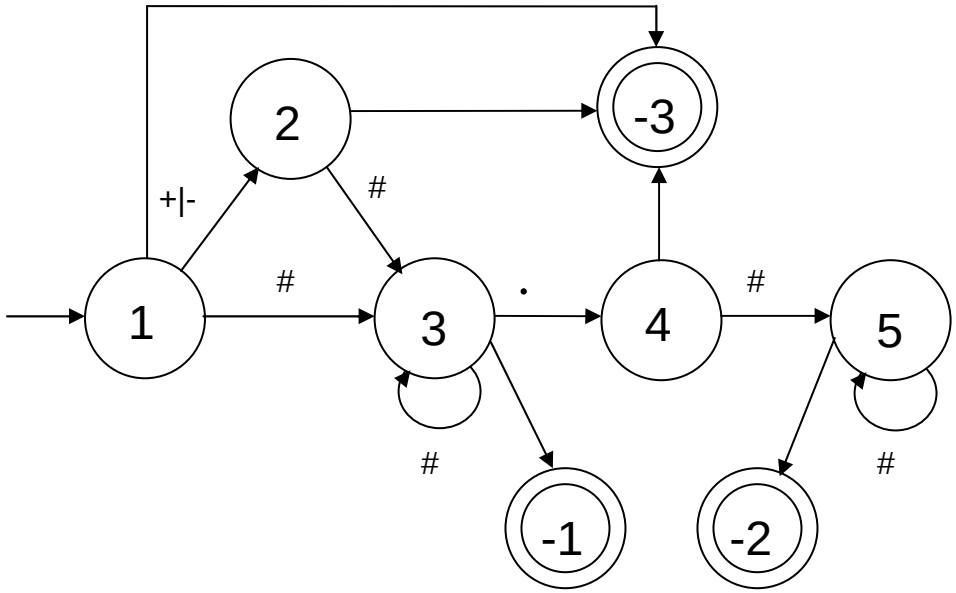


Lexical Analyzer

- Recognize lexemes within character stream of source program
- Generates tokens
- Performs input and error output
- Under control of the syntactic analyzer

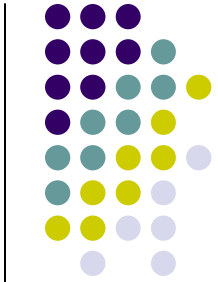


DFA for Numeric Literals – with terminating states (+|-|λ)(0|1|2|3|4|5|6|7|8|9)+(.(0|1|2|3|4|5|6|7|8|9)+| λ)



State table for

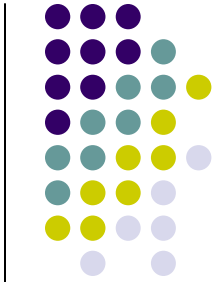
(+|-|λ)(0|1|2|3|4|5|6|7|8|9)+(.(0|1|2|3|4|5|6|7|8|9)+| λ)



State	+	-	.	0	1	2	3	4	5	6	7	8	9	Other
1	2	2	-3	3	3	3	3	3	3	3	3	3	3	-3
2	-3	-3	-3	3	3	3	3	3	3	3	3	3	3	-3
3*	-1	-1	4	3	3	3	3	3	3	3	3	3	3	-1
4	-3	-3	-3	5	5	5	5	5	5	5	5	5	5	-3
5*	-2	-2	-2	5	5	5	5	5	5	5	5	5	5	-2

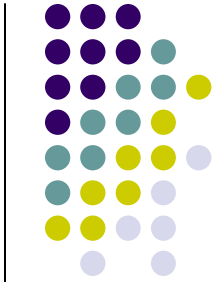
- -1 : Integer literal found
- -2 : Real literal found
- -3 : Not a numeric literal

Sample Input and Output



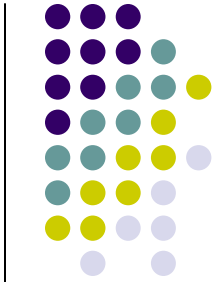
- `123 -123 123.45 -123.45 123.45.7`
  - `Integer literal found (123)`
  - `Integer literal found (-123)`
  - `Real literal found (123.45)`
  - `Real literal found (-123.45)`
  - `Real literal found (123.45)`
  - `Not a numeric literal (.)`
  - `Integer literal found (7)`

Driver function



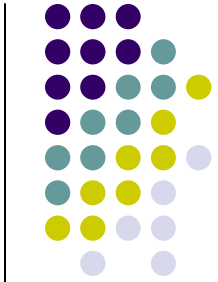
```
int main (int argc, char * argv[])
{
    ifstream input (argv[1]);
    int result;
    do
    {
        result = nextone(input);
        switch (result)
        {
            case 1: cout << "Integer literal found\n"; break;
            case 2: cout << "Real literal found\n"; break;
            case 3: cout << "Not a numeric literal\n"; break;
        }
    } while (result);
    input.close();
    return 0;
}
```

int nextone (istream & ins);



```
int nextone (istream & ins)
{
    char alphabet[] = "+-.0123456789";
    int table[][14] = {{0,0,0,0,0,0,0,0,0,0,0,0,0,0}, {2,2,-3,3,3,3,3,3,3,3,3,3,3,-3},
              {-3,-3,-3,3,3,3,3,3,3,3,3,3,3,-3}, {-1,-1,4,3,3,3,3,3,3,3,3,3,3,-1},
              {-3,-3,-3,5,5,5,5,5,5,5,5,5,5,-3}, {-2,-2,-2,5,5,5,5,5,5,5,5,5,5,-2}};
    static string text = " ";
    static int pos = 0;
    while (isspace (text[pos]))
    {
        pos++;
        if (pos >= text.length())
        {
            getline (ins,text);
            if (ins.fail())
                return 0;
            text += " ";
            pos = 0;
        }
    }
    int state = 1;
    while (state > 0)
    {
        int a;
        for (a = 0; alphabet[a] && alphabet[a] != text[pos]; a++);
        state = table [state][a];
        pos++;
    }
    if (state != -3)
        pos--;
    return -state;
}
```

Project 1

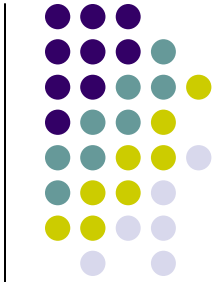


- List of lexemes
- Enumerated type for token

```
enum token {PLUS, MINUS,…};
```

- Bypass whitespace
- Numeric literals unsigned
- Produce listing
- What is a lexical error?

Sample input and output



```
identifier 123 456.78 0 +
++ +=; $ another
123 456.78 0+ ++ += ;
```

```
[1] identifier 123 456.78 0 +
        IDENT       identifier
        NUMLIT      123
        NUMLIT      456.78
        NUMLIT      0
        PLUS        +
[2] ++ +=; $ another
        PLUSPLUS    ++
        PLUSEQ      +=
        SEMI        ;
invalid character found near 2:9
        ERROR       $
        IDENT       another
[3] 123 456.78 0+ ++ += ;
        NUMLIT      123
        NUMLIT      456.78
        NUMLIT      0
        PLUS        +
        PLUSPLUS    ++
        PLUSEQ      +=
        SEMI        ;
```