

More Applications of the Pumping Lemma

The Pumping Lemma:

- Given a infinite regular language L
- there exists an integer m
- for any string $w \in L$ with length $|w| \geq m$
- we can write $w = x y z$
- with $|x y| \leq m$ and $|y| \geq 1$
- such that: $x y^i z \in L \quad i = 0, 1, 2, \dots$

Non-regular languages

$$L = \{vv^R : v \in \Sigma^*\}$$



Regular languages

Theorem: The language

$$L = \{vv^R : v \in \Sigma^*\} \quad \Sigma = \{a, b\}$$

is not regular

Proof: Use the Pumping Lemma

$$L = \{vv^R : v \in \Sigma^*\}$$

Assume for contradiction
that L is a regular language

Since L is infinite
we can apply the Pumping Lemma

$$L = \{vv^R : v \in \Sigma^*\}$$

Let m be the integer in the Pumping Lemma

Pick a string w such that: $w \in L$ and

$$\text{length } |w| \geq m$$

We pick $w = a^m b^m b^m a^m$

Write $a^m b^m b^m a^m = x y z$

From the Pumping Lemma

it must be that length $|x y| \leq m, |y| \geq 1$

$$xyz = \overbrace{a \dots a}^m \overbrace{a \dots a}^m \overbrace{a \dots a}^m \overbrace{a \dots a}^m$$
$$\underbrace{a \dots a}_{x} \underbrace{a \dots a}_{y} \underbrace{a \dots a \dots a}_{z}$$

Thus: $y = a^k, k \geq 1$

$$x y z = a^m b^m b^m a^m$$

$$y = a^k, \quad k \geq 1$$

From the Pumping Lemma: $x y^i z \in L$
 $i = 0, 1, 2, \dots$

Thus: $x y^2 z \in L$

$$x y z = a^m b^m b^m a^m \qquad y = a^k, \quad k \geq 1$$

From the Pumping Lemma: $x y^2 z \in L$

$$xy^2z = \overbrace{a \dots a}^{m+k} \overbrace{a \dots a}^m \overbrace{a \dots a}^m \overbrace{a \dots a}^m \in L$$

$\underbrace{\hspace{1.5cm}}_x$
 $\underbrace{\hspace{1.5cm}}_y$
 $\underbrace{\hspace{1.5cm}}_y$

$\underbrace{\hspace{4.5cm}}_z$

Thus: $a^{m+k} b^m b^m a^m \in L$

$$a^{m+k}b^mb^ma^m \in L \quad k \geq 1$$

BUT: $L = \{vv^R : v \in \Sigma^*\}$



$$a^{m+k}b^mb^ma^m \notin L$$

CONTRADICTION!!!

Therefore: Our assumption that L
is a regular language is not true

Conclusion: L is not a regular language

Note: Can we use a string $w=a^m a^m$?

Non-regular languages

$$L = \{a^n b^l c^{n+l} : n, l \geq 0\}$$



Regular languages

Theorem: The language

$$L = \{a^n b^l c^{n+l} : n, l \geq 0\}$$

is not regular

Proof: Use the Pumping Lemma

$$L = \{a^n b^l c^{n+l} : n, l \geq 0\}$$

Assume for contradiction
that L is a regular language

Since L is infinite
we can apply the Pumping Lemma

$$L = \{a^n b^l c^{n+l} : n, l \geq 0\}$$

Let m be the integer in the Pumping Lemma

Pick a string w such that: $w \in L$ and

$$\text{length } |w| \geq m$$

We pick $w = a^m b^m c^{2m}$

Write $a^m b^m c^{2m} = x y z$

From the Pumping Lemma

it must be that length $|x y| \leq m, |y| \geq 1$

$$xyz = \overbrace{a \dots a}^m \overbrace{a \dots a}^m \overbrace{ab \dots bc \dots cc \dots c}^{2m}$$
$$\underbrace{\hspace{1.5cm}}_x \underbrace{\hspace{1.5cm}}_y \underbrace{\hspace{4cm}}_z$$

Thus: $y = a^k, k \geq 1$

$$x y z = a^m b^m c^{2m}$$

$$y = a^k, \quad k \geq 1$$

From the Pumping Lemma: $x y^i z \in L$
 $i = 0, 1, 2, \dots$

Thus: $x y^0 z = xz \in L$

$$x y z = a^m b^m c^{2m} \qquad y = a^k, \quad k \geq 1$$

From the Pumping Lemma: $xz \in L$

$$xz = \overbrace{a \dots a}^{m-k} \overbrace{a \dots a}^m \overbrace{b \dots b}^m \overbrace{c \dots c}^{2m} \in L$$

$$\underbrace{\hspace{1.5cm}}_x \underbrace{\hspace{4.5cm}}_z$$

Thus: $a^{m-k} b^m c^{2m} \in L$

$$a^{m-k} b^m c^{2m} \in L \quad k \geq 1$$

BUT: $L = \{a^n b^l c^{n+l} : n, l \geq 0\}$



$$a^{m-k} b^m c^{2m} \notin L$$

CONTRADICTION!!!

Therefore: Our assumption that L
is a regular language is not true

Conclusion: L is not a regular language

Non-regular languages

$$L = \{a^{n!} : n \geq 0\}$$



Regular languages

Theorem: The language $L = \{a^{n!} : n \geq 0\}$
is not regular

$$n! = 1 \cdot 2 \cdots (n-1) \cdot n$$

Proof: Use the Pumping Lemma

$$L = \{a^{n!} : n \geq 0\}$$

Assume for contradiction
that L is a regular language

Since L is infinite
we can apply the Pumping Lemma

$$L = \{a^{n!} : n \geq 0\}$$

Let m be the integer in the Pumping Lemma

Pick a string w such that: $w \in L$

$$\text{length } |w| \geq m$$

We pick $w = a^{m!}$

Write $a^{m!} = x y z$

From the Pumping Lemma

it must be that length $|x y| \leq m, |y| \geq 1$

$$xyz = a^{m!} = \overbrace{a \dots a}^m \overbrace{a \dots a}^{m!-m}$$
$$\underbrace{\hspace{1.5cm}}_x \underbrace{\hspace{1cm}}_y \underbrace{\hspace{4cm}}_z$$

Thus: $y = a^k, 1 \leq k \leq m$

$$x y z = a^{m!}$$

$$y = a^k, \quad 1 \leq k \leq m$$

From the Pumping Lemma: $x y^i z \in L$
 $i = 0, 1, 2, \dots$

Thus: $x y^2 z \in L$

$$x y z = a^{m!}$$

$$y = a^k, \quad 1 \leq k \leq m$$

From the Pumping Lemma: $x y^2 z \in L$

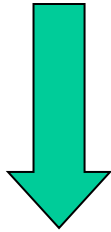
$$xy^2z = \overbrace{a \dots a}^{m+k} \overbrace{a \dots a}^{m!-m} \in L$$

$\underbrace{\hspace{1.5cm}}_x \underbrace{\hspace{1.5cm}}_y \underbrace{\hspace{1.5cm}}_y \underbrace{\hspace{4cm}}_z$

Thus: $a^{m!+k} \in L$

$$a^{m!+k} \in L \qquad 1 \leq k \leq m$$

Since: $L = \{a^{n!} : n \geq 0\}$



There must exist p such that:

$$m!+k = p!$$

However: $m!+k \leq m!+m$ for $m > 1$

$$\leq m!+m!$$

$$< m!m + m!$$

$$= m!(m+1)$$

$$= (m+1)!$$



$$m!+k < (m+1)!$$



$$m!+k \neq p! \quad \text{for any } p$$

$$a^{m!+k} \in L \qquad 1 \leq k \leq m$$

BUT: $L = \{a^{n!} : n \geq 0\}$



$$a^{m!+k} \notin L$$

CONTRADICTION!!!

Therefore: Our assumption that L
is a regular language is not true

Conclusion: L is not a regular language

Lex

Lex: a lexical analyzer

- A Lex program recognizes strings
- For each kind of string found the lex program takes an action

Input

```
Var = 12 + 9;  
if (test > 20)  
    temp = 0;  
else  
    while (a < 20)  
        temp++;
```

*Lex
program*

Output

```
Identifier: Var  
Operand: =  
Integer: 12  
Operand: +  
Integer: 9  
Semicolon: ;  
Keyword: if  
Parenthesis: (  
Identifier: test  
....
```

In Lex strings are described
with regular expressions

Lex program

Regular expressions

"+"

"_"

"=

/* operators */

"if"

"then"

/* keywords */

Lex program

Regular expressions

`(0|1|2|3|4|5|6|7|8|9)+ /* integers */`

`(a|b|..|z|A|B|...|Z)+ /* identifiers */`

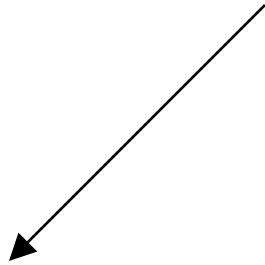
integers



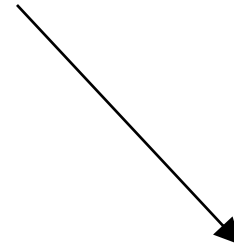
$(0|1|2|3|4|5|6|7|8|9)^+$

$[0-9]^+$

identifiers



$(a|b|..|z|A|B|...|Z)^+$



$[a-zA-Z]^+$

Each regular expression
has an associated action (in C code)

Examples:

Regular expression	Action
<code>\n</code>	<code>linenum++;</code>
<code>[0-9]+</code>	<code>printf("integer");</code>
<code>[a-zA-Z]+</code>	<code>printf("identifier");</code>

Default action: ECHO;



Prints the string identified
to the output

A small lex program

%%

[\t\n] ; /*skip spaces*/

[0-9]+ printf("Integer\n");

[a-zA-Z]+ printf("Identifier\n");

Input

1234 test

var 566 78

9800

Output

Integer

Identifier

Identifier

Integer

Integer

Integer

Another program

```
%{  
int linenum = 1;  
%}
```

```
%%
```

```
[ \t]          ; /*skip spaces*/  
\n             linenum++;
```

```
[0-9]+         printf("Integer\n");
```

```
[a-zA-Z]+     printf("Identifier\n");
```

```
.             printf("Error in line: %d\n",  
                    linenum);
```

Input

1234 test

var 566 78

9800 +

temp

Output

Integer

Identifier

Identifier

Integer

Integer

Integer

Error in line: 3

Identifier

Lex matches the longest input string

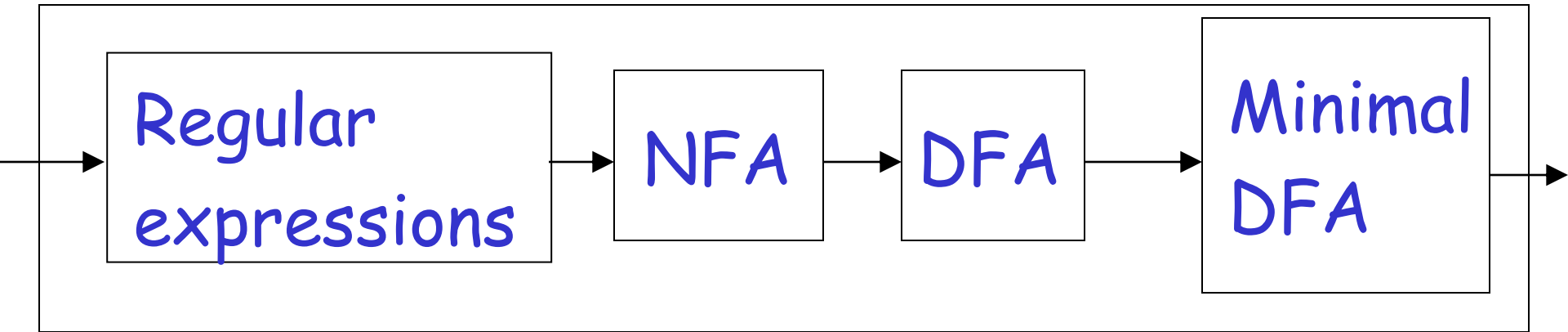
Example: Regular Expressions "if"
 "ifend"

Input: ifend if

Matches: "ifend" "if"

Internal Structure of Lex

Lex



The final states of the DFA are associated with actions