More Applications

of

the Pumping Lemma

The Pumping Lemma:

- Given a infinite regular language L
- there exists an integer m จำนวน state
- 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: $x\, y^i\, z \in L \qquad i = 0, 1, 2, \ldots$

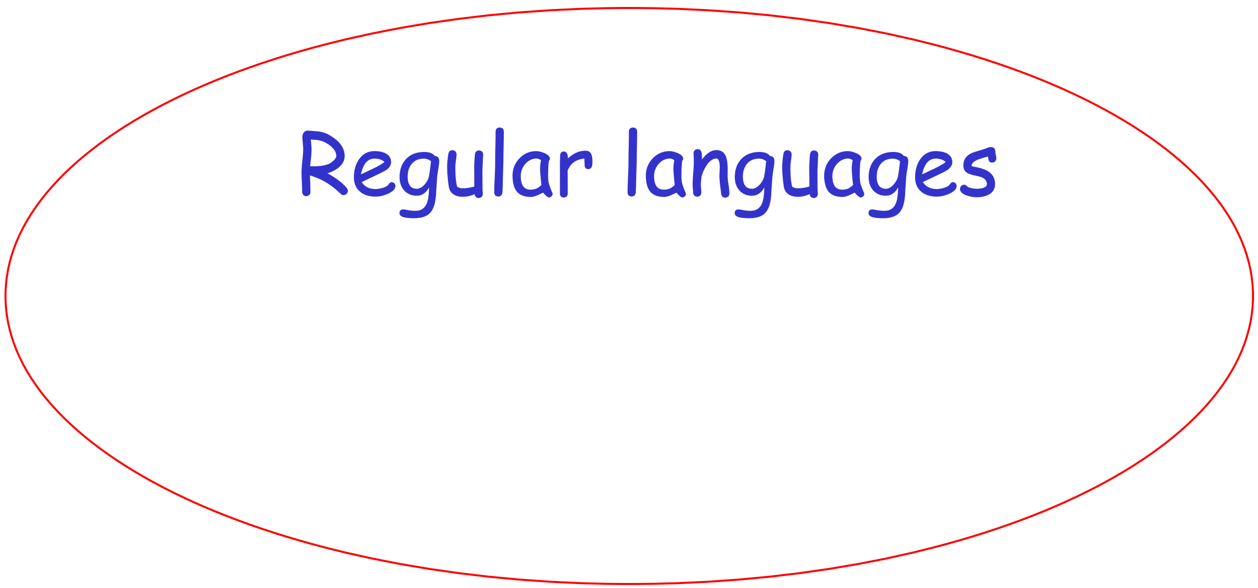
$\Sigma = \{a, b\}$

ความยาว ของ เป็นเลขคู่

Non-regular languages

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

aa , abba



Theorem: The language

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

is not regular

Proof: Use the Pumping Lemma

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

สมมติ

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

length $|w| \geq m$

$a^m a^m \rightarrow$ เลือก str ที่ ปั๊มไม่หลุด

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

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

From the Pumping Lemma

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

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

$$\overbrace{a \ldots aa \ldots a \ldots a}^{m}$$

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

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

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

$$i = 0, 1, 2, \ldots$$

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

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

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

$$xy^2z = \overbrace{\underbrace{a\ldots a}_{x}\underbrace{a\ldots a}_{y}\underbrace{a\ldots a}_{y}a\ldots a}^{m+k}\overbrace{b\ldots b}^{m}\overbrace{b\ldots b}^{m}\overbrace{a\ldots a}^{m} \in L$$

with z the part from the last $a\ldots a$ of the first block to the end.

y เกินมา 1 ส่วน

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

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

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

$$\Downarrow$$

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

CONTRADICTION!!!

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

$$\overbrace{aaa\ldots aa}^{m}\ \overbrace{aaaa\ldots}^{m}, \quad y = aa$$

$$\underbrace{aaa\ldots}_{x}\underbrace{aa}_{y}\ \underbrace{aaaa\ldots}_{z}$$

$i=0 \quad a^{m-2}\, a^{m} = a^{m-1}\, a^{m-1}$

$i=2 \quad a^{m-2} \cdot a^{4}\, a^{m} = a^{m+2}\, a^{m} = a^{m+1}\, a^{m+1}$

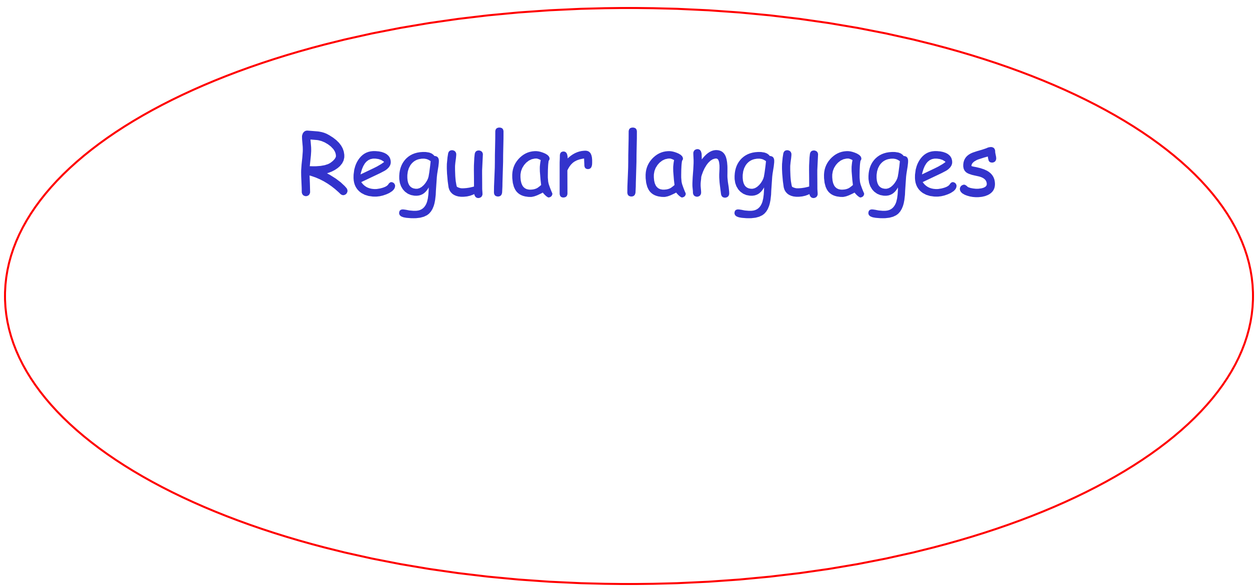
Conclusion: L is not a regular language

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

Non-regular languages

ไม่รู้ว่า a กี่ตัว b กี่ตัว
นับตัวใช้ความจำเยอะเกิน

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



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

length $|w| \geq m$

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

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

From the Pumping Lemma
it must be that length $\quad |x\ y| \leq m, \quad |y| \geq 1$

$$xyz = \underbrace{\overbrace{a\ldots a}^{}}_{x}\ldots$$

$$xyz = \overbrace{\underbrace{a\ldots a}_{x}\underbrace{a\ldots a}_{y}a\ldots a}^{m}\overbrace{b\ldots b}^{m}\overbrace{c\ldots cc\ldots c}^{2m}$$

$$xyz = \underbrace{a\ldots a}_{x}\underbrace{a\ldots a}_{y}\underbrace{a\ldots ab\ldots bc\ldots cc\ldots c}_{z}$$

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

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

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

$$i = 0, 1, 2, \ldots$$

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

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

From the Pumping Lemma: $xz \in L$

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

$$xz = \underbrace{a...a}_{x}\underbrace{a...ab...bc...cc...c}_{z} \in L$$

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

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

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

$\Downarrow$

เพราะ: ต้องเกิดจาก

$n+l$

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

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: $w \in L$

length $|w| \geq m$

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

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

From the Pumping Lemma

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

$$xyz = a^{m!} = \overbrace{a\ldots a\,a\ldots a\,a\ldots a}^{m}\,\overbrace{a\ldots a\,a\ldots a}^{m!-m}$$

$$= \underbrace{a\ldots a}_{x}\,\underbrace{a\ldots a}_{y}\,\underbrace{a\ldots a\,a\ldots a\,a\ldots a}_{z}$$

$m + m! - m$
$= m!$ มันฟิกให้ว่า
$a!$ ดังนั้น
จำนวน str
ต้องเท่ากัน

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

$$x\,y\,z = a^{m!} \qquad\qquad y = a^k, \quad 1 \le k \le m$$

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

$$i = 0, 1, 2, \ldots$$

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

$$x\ y\ z = a^{m!} \qquad y = a^k, \quad 1 \le k \le m$$

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

$$xy^2z = \overbrace{\underbrace{a\ldots a}_{x}\underbrace{a\ldots a}_{y}\underbrace{a\ldots a}_{y}a\ldots a}^{m+k}\overbrace{a\ldots aa\ldots a}^{m!-m} \in L$$

(with z spanning $a\ldots aa\ldots aa\ldots a$ after the second y)

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

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

ความยาวของ y

ซึ่ง $|xy| < m$

y ก็ต้องน้อยกว่า m

ไม่มีทางเกิน

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

There must exist p such that:

$$m! + k = p!$$

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

$\leq m!+m!$

เอา m คูณเข้าแล้วจะทำให้เครื่องหมายมากกว่าเลย

$< m!m + m!$

ดึงตัวร่วม

$< m!(m+1)$ นิยามของ

$< (m+1)!$

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

จาก

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

$$m!+k \neq p!$$ **for any** p

ไม่สามารถเขียนให้อยู่ในรูป ที่มี ! ได้เลย

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

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

$$\Downarrow$$

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

CONTRADICTION!!!

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

Conclusion: L is not a regular language

ไม่ออกสอบ

Lex

free
โหลดได้

Lex: a lexical analyzer

- A Lex program recognizes strings

- For each kind of string found
  the lex program takes an action

ID OP INT OP INT Semi Colon

Input

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

→ **Lex program** →

Output แปลงให้ เป็น token

Identifier: Var

Token Operand: =

Token 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

ทำซ้ำตั้งแต่ 1 ครั้งขึ้นไป

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

เซ็ตของ ๐-๙

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

integers

(0|1|2|3|4|5|6|7|8|9)+ $\xrightarrow{\text{| สัยพาแบบนี้ได้}}$ [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
\n ขึ้นบรรทัดใหม่	internal veriable linenum++;
[0-9]+	token printf("integer");
100 [a-zA-Z]+	Token printf("identifier");

Default action: ECHO;

Str ที่ไม่ไปแมทเลย เข้า

สิ่งไปรวมมา print ตัวที่หลุดออกไป

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");
```

มีทอะไมสนใจ

Input

1234 test

var 566 78

9800

Output

Integer

Identifier

Identifier

Integer

Integer

Integer

Another program

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

ต้อง define ที่ข้างบนก่อน

```
%}

%%
```

ไม่ทำอะไร

```
[ \t]          ;    /*skip spaces*/
\n             linenum++;
[0-9]+         prinf("Integer\n");
[a-zA-Z]+      printf("Identifier\n");
.              printf("Error in line: %d\n",
                      linenum);
```

→ default

Input

```
1234    test
var  566     78
9800  +
temp
```

ไม่ตรง reg ex. มันเลย Error

Output

```
Integer
Identifier
Identifier
Integer
Integer
Integer
Error in line: 3
Identifier
```

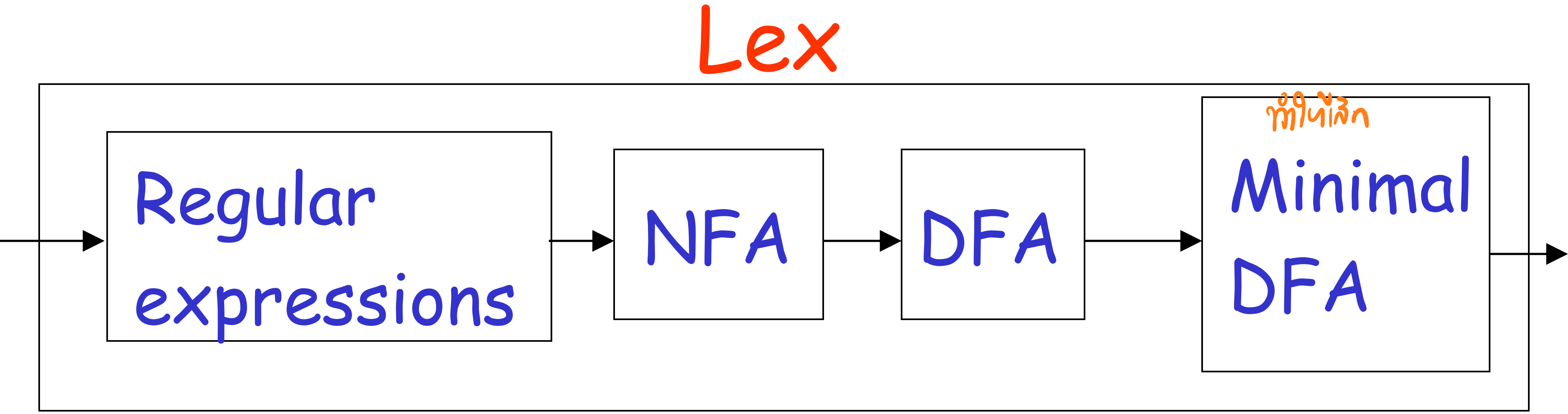
Lex matches the longest input string

Example: Regular Expressions "if"
"ifend"

หลักการ match str ที่ยาวที่สุด

Input:	ifend	if
Matches:	"ifend"	"if"

Internal Structure of Lex



The final states of the DFA are associated with actions