# CS323 Compilers

## Homework #1

**Site Fan**
**fanst2021@mail.sustech.edu.cn**

# Exercise 1

Java programs may contain lexical errors. Please give at least two types of possible lexical errors in Java programs and provide code snippets as examples when possible.

## Solution

1. Invalid identifier: starts with a digit

```
int 1stVariable = 10;
```

2. Unmatched double quotes

```
String greeting = "Hello World;
```

# Exercise 2

Given a string $s$, can you find a string $x$ that is both a prefix and a suffix of $s$? Can you further find a string $y$ that is both a proper prefix and a proper suffix of $s$? If yes, please provide an example. Otherwise, please explain the reason.

## Solution

1. The string $x = s$ is both the prefix and suffix of $s$ itself, so we can always find a string $x$ that is both a prefix and a suffix of $s$.

2. The string $y$ that is both a proper prefix and a proper suffix of $s$ may not exist, e.g., $s =$"CS323″. We can use the `Next` array of KMP algorithm as the reference.

   Assume that the length of $s$ is $n$, after calculating the `Next` array, if `Next[n]!=0`, then there exists a string $y$ of length `Next[n]` that is both a proper prefix and a proper suffix of $s$.

```
void calculate_next(int l)
{
    int i = 0, j = -1;
    Next[0] = -1;
    while (i < l)
    {
        if (j == -1 || s[i] == s[j])
        {
            i++, j++;
            Next[i] = j;
        }
        else
            j = Next[j];
    }
}
```

# Exercise 3

In a string of length n (n > 0), how many of the following are there? For simplicity, we assume that the string contains n different characters. Besides giving the final answers, please also explain how you derive the answers.

1. Substrings of length m (0 < m ≤ n)

2. Subsequences

### Solution

1. There are $n - m + 1$ letters that can be the starting of a substrinig of length $m$. So there are $(n - m + 1)$ different substrings of length $m$.
2. For every letter, it has 2 states: contained by the subsequence and not contained. Hence there are $2^n$ different subsequences for $s$.

## Exercise 4

Write a regular definition as well as a regular expression to represent all strings of valid telephone numbers in Shenzhen. A valid telephone number contains the country code (86), a hyphen, the area code 755, another hyphen, and eight digits where the first one cannot be zero (e.g., 86-755-88015159).

### Solution

^86-755-[1-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]$

## Exercise 5

Given an alphabet Σ = {0, 1}, are the following two regular languages equivalent? Besides saying yes or no, please also prove your answer.

1. L1 = L((0*1*)*)
2. L2 = L((0|1)*)

### Solution

We can prove the two regular languages are equivalent by induction.

**Base Case**

Both L1 and L2 generate empty string $\varepsilon$, whose length is 0.

**Induction Step**

Assume that for any 01-string $s_k$ of length $k$ ($k = 0, 1, 2...$), it is contained by both L1 and L2.

Then for any string $s_{k+1}$ of length $k + 1$, it can be generated from a string $s_k$ of length $k$ in the following ways:

1. $s_{k+1} \in$ L($s_k$(0*1*))

Since $s_k \in$ L((0*1*)*), $s_{k+1} \in$ L((0*1*)*(0*1*)) = L((0*1*)*) = L1.

2. $s_{k+1} \in$ L($s_k$(0|1))

Since $s_k \in$ L((0|1)*), $s_{k+1} \in$ L((0|1)*(0|1)) = L((0|1)*) = L2.

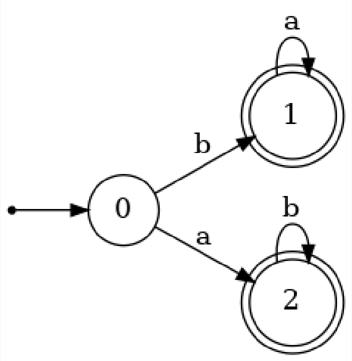
Both concatenate a 0 or 1 to the tail of $s_k$ to generate $s_{k+1}$, and any 01-string of length $k + 1$ is contained by both L1 and L2.

By mathematical induction, we conclude that L1 and L2 are equivalent, both contain all 01-strings of non-negative length $0, 1, 2...$.

## Exercise 6

Consider the regular expression ba*|ab*. Please provide a state transition diagram that can recognize the strings in the corresponding regular language. Can the transition diagram recognize the string baab? If yes, please give the sequence of state transitions. Otherwise, please explain the reason.

According to the precedence of regex operators, ba*|ab* = (ba*)|(ab*).



The string baab fails to match the regex since it's ending with "b" after entering state 1.