



链滴

# CODE | 用 C# 实现 DFA

作者: [wwbaby](#)

原文链接: <https://ld246.com/article/1543473739612>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## DFA的定义

dfa是面向装配的设计(Design for assembly)的英文简称，是指在产品设计阶段设计产品使得产品具有良好的可装配性，确保装配工序简单、装配效率高、装配质量高、装配不良率低和装配成本低。面向配的设计通过一系列有利于装配的设计指南例如简化产品设计、减少零件数量等，并同装配工程师合作，简化产品结构，使其便于装配，为提高产品质量、缩短产品开发周期和降低产品成本奠定基础。

在DFA中，有初态、终态和转换函数。用一位数组存储前两个，用二维数组存储转换函数。

## 在项目中添加一个DFAobject类

```
class DFAobject
{
    //非终结符、终结符、转换函数
    public char[] n;
    public char[] t;
    public char[,] f;
    //初始化自动机，输入状态和转换函数
    public void Initialize()
    {
        Console.WriteLine("输入非终结符集的长度：");
        int x = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("输入终结符集的长度：");
        int y = Convert.ToInt32(Console.ReadLine());
        n = new char[x];
        t = new char[y];
        f = new char[x,y];
        Console.WriteLine("输入非终结符，默认第一个为初态最后一个为终态：");
        for (int i = 0; i < n.Length; i++)
        {
            n[i] = char.Parse(Console.ReadLine());
        }
    }
}
```

```

Console.WriteLine("输入终结符: ");
for (int i = 0; i < t.Length; i++)
{
    t[i] = char.Parse(Console.ReadLine());
}
Console.WriteLine("输入转换函数后的状态, 用空格分隔开每个字符, 按行输入: ");
for (int i = 0; i < x; i++)
{
    string str = Console.ReadLine();//表示一位数组的一行数据
    string[] s = str.Split(" ").ToArray();//通过split方法以空格为分隔符分隔输入的字符串
    for (int j = 0; j < y; j++)
    {
        f[i, j] = Convert.ToChar(s[j]);//将分隔后的字符传给二维数组的每个元素
    }
}
}
//判断行的转换函数
public int IfX(char c)
{
    for (int i = 0; i < n.Length; i++)
    {
        if (n[i] == c)
        {
            return i;
        }
    }
    return -1;
}
//判断列的转换函数
public int IfY(char c)
{
    for (int i = 0; i < t.Length; i++)
    {
        if (t[i] == c)
            return i;
    }
    return -1;
}
}
}

```

在这里用了矩阵去存储转换函数, 终结符和非终结符的数组索引代表了矩阵的行数和列数。所以IfX() IfY()函数的作用就是得到矩阵的行和列对应元素。

在识别的过程中, 当前的非终结符会不断更新, 能到达终态即为改语言的串。所以在主函数中实现了前终结符的更新和最后是否到达终态的判别。

主函数代码如下

```

static void Main(string[] args)
{
    DFAobject dfa = new DFAobject();
    dfa.Initialize();
    char c, x = dfa.n[0];
    while(true)
    {

```

```
Console.WriteLine("请输入要识别的串: ");
string str = Console.ReadLine();
Console.WriteLine("过程: ");
for (int i = 0; i < str.Length; i++)
{
    c = str[i];
    x = dfa.f[dfa.IfX(x), dfa.IfY(c)];
    Console.WriteLine(x);

}
Console.WriteLine("是否是该语言的串: ");
if (x == dfa.n[dfa.n.Length - 1])
    Console.WriteLine("yes");
else
    Console.WriteLine("no");
x = dfa.n[0]; //回到初态
}
}
}
```