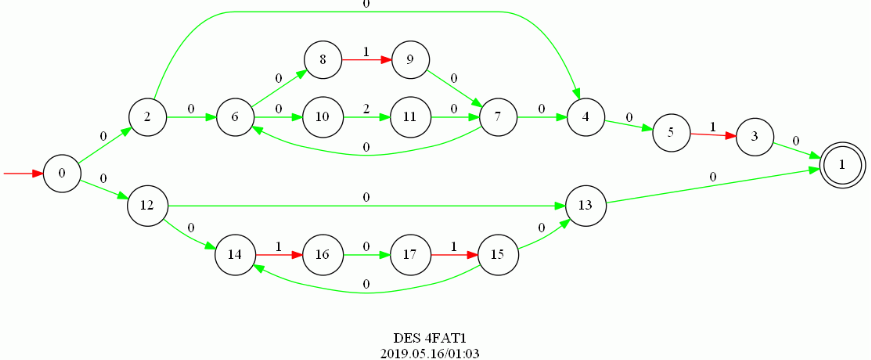
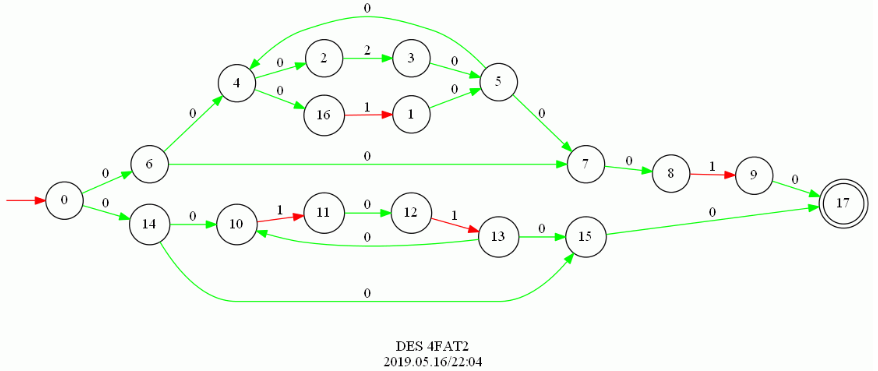
**问题1：判断是否同构功能有问题**

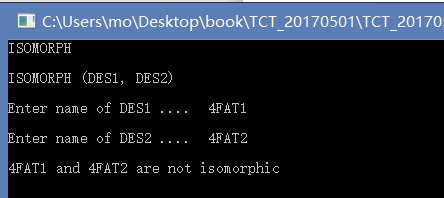
问题在于构成NFA，而非DFA。同构问题的前提必须是DFA,才会正确



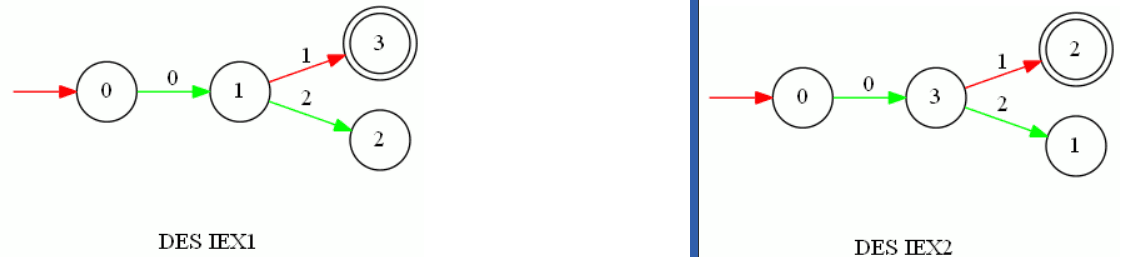
0状态存在相同字母(‘0’)的输入，成为一个NFA。应该是不同字母的输入，即DFA，才会正确。12，6，7等等状态类似。



上述两图分别是对同一个正则表达式通过Thompson算法（两种类型的构造函数）构造的FA，很明显从图可以看出两个FA是同构的，只是状态数字的命名不同，但是通过软件验证同构的结果是不同构的。



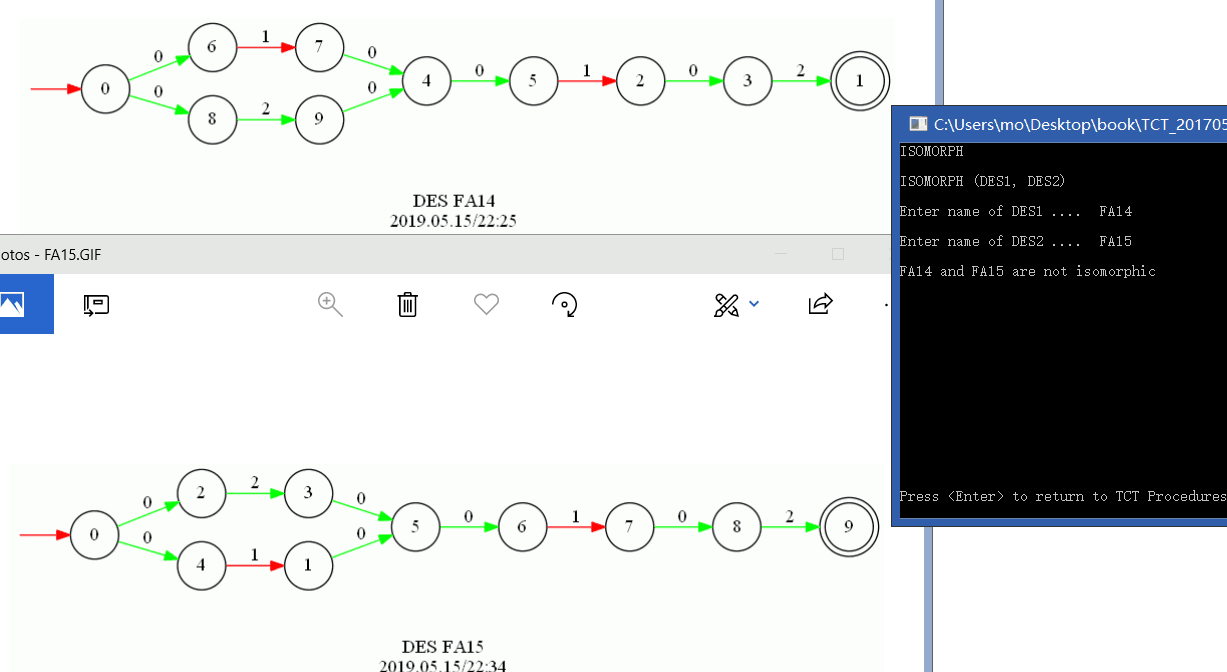
1. 为此我首先输入一个极其简单的例子

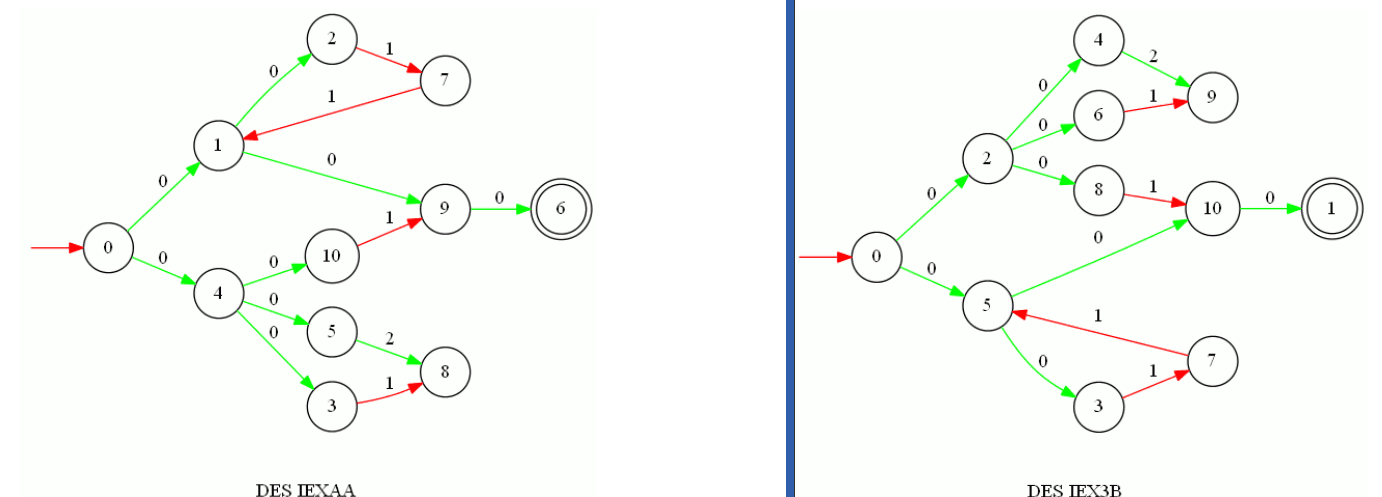


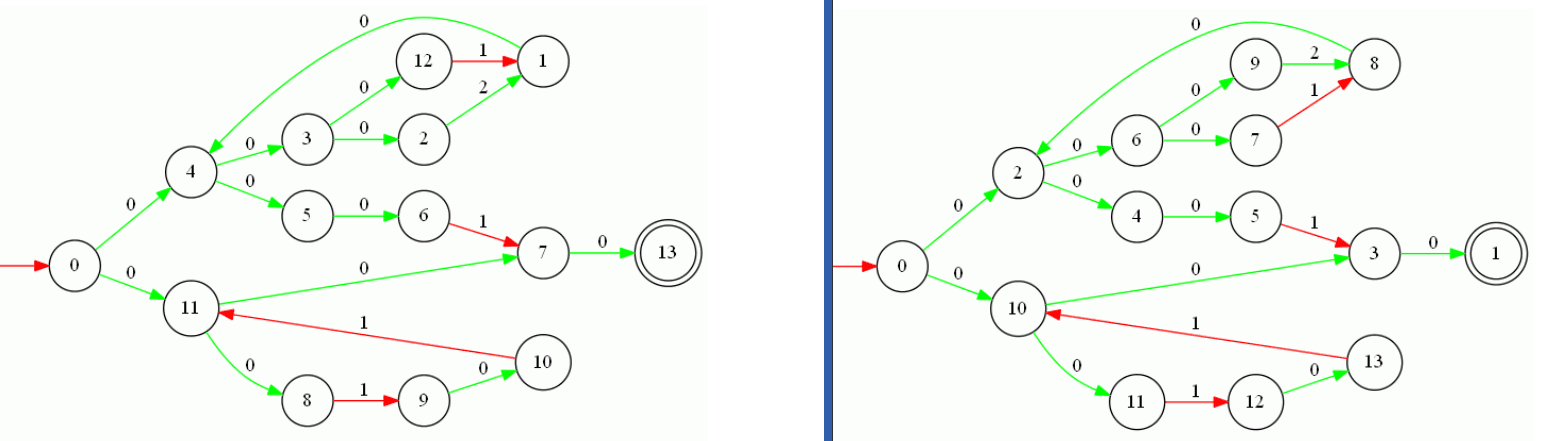
正常DFA，因此，结果正确。

这两个的结果是同构的没有问题；

2）然后我输入三组稍微复杂点的例子



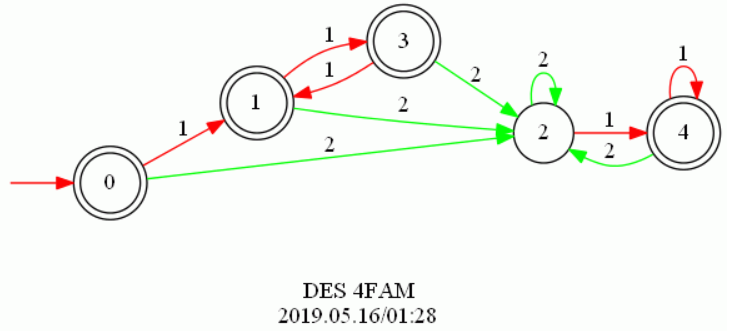




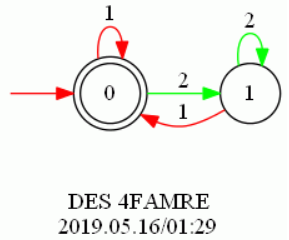
这三个的结果都是不同构的，但很明显从图上可以看出来这也只是改变了状态数字的命名而已，所有结构和转移条件都是一样的。

所以TCT的验证同构功能好像是没办法用的，也就没办法通过这个验证FA是否等价。

**问题2：最小化功能有问题**

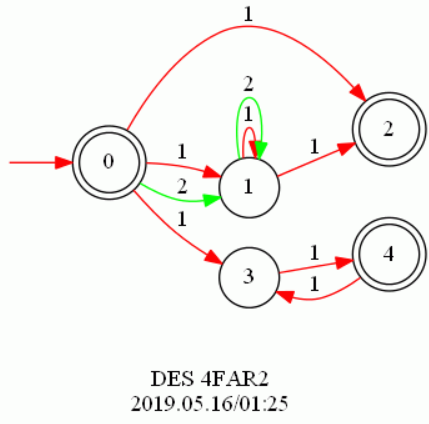


上图是对与问题一相同的正则表达式通过MYG算法（因为BS算法和ASU算法构造出来的DFA是一模一样的，所以只用一个图代替）构造的FA。但是对他进行最小化（选项9的MINSTATE），结果是



正确的最小化，与RE等价。

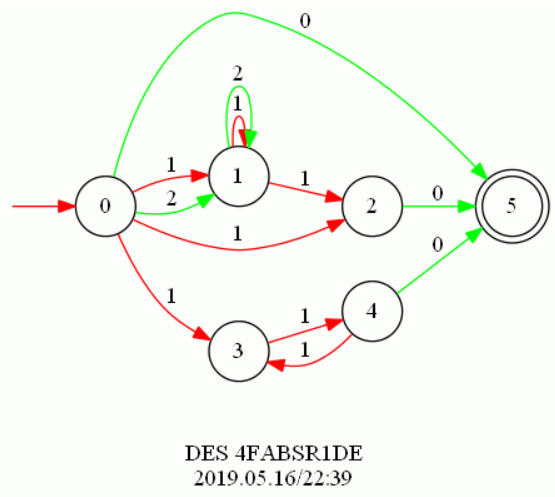
这明显是跟RE不符合的FA，说明其最小化的结果是有错误的。



是一个NFA，而不是DFA，不会正确。

上图是对同一个RE通过BS算法RFA函数构造出来的RFA，对该RFA最小化后的结果，明显与MYG算法最小化后的结果不等价。

下图是对同一个RE通过BS算法的RBFA构造函数构造出来的RBFA，对该RBFA最小化后的结果



是一个NFA，而不是DFA，不会正确。

从图也明显可以得到最小化的结果是不等价的FA。

以下问题下周一见面时讨论。

**问题3：比如：同一个RE构造出来的RFA和DFA肯定是不一样的（RFA可以有多个开始状态且可以有转移，而DFA只有一个开始状态且每个转移都是确定的而不能有空转移）。我们可以通过对应同一个正则表达式直接判断他们等价吗？**

**除了问题外，通过TCT验证可以得出（把BS算法的RFA构造当做单独的一个算法的话）**

**①Thompson算法两个不同构造函数得出来的结果是同构且等价的；**

**②RFA算法和BS算法得出来的RFA和LBFA和RBFA是等价的；**

**③所有算法得出来的DFA都是相同的所以没有比较是否等价的必要。**