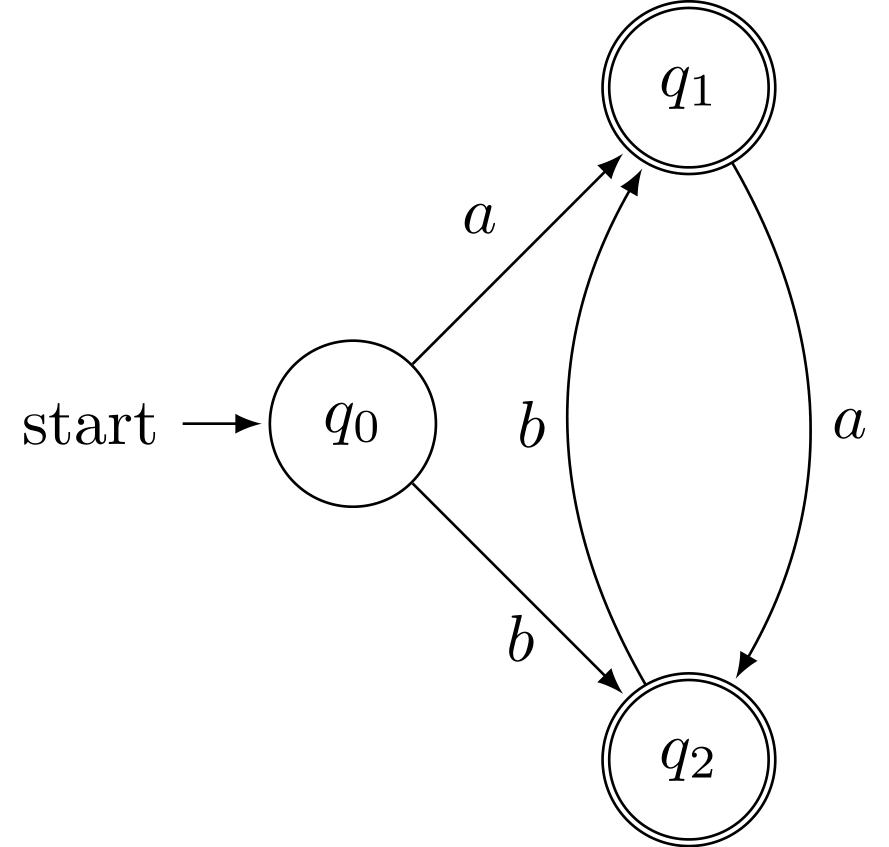
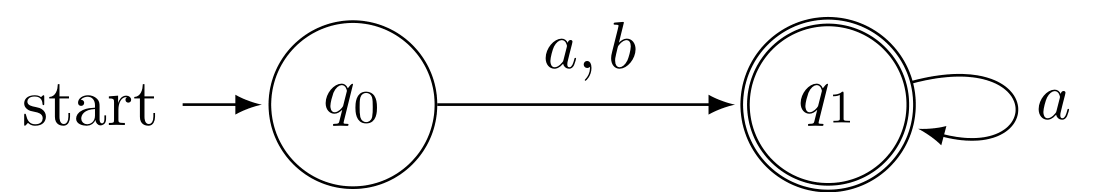
对于这样一个自动机



显然它接受的语言是，显然该自动机已经是一个最小确定性有限状态自动机(DFA)，但是算法 DFA:: min\_Hopcroft() 的输出结果为



根据Hopcroft算法的思想，在对这个自动机进行最小化操作的时候，把自动机分为两个等价类，接受状态和非接受状态，在此例中，分为以下两个等价类

等价类 内仅有一个状态，不可再分。对于等价类 ， 仅接受字符“a”， 仅接受字符“b”，所以该等价类应该再细分为两个等价类，所以该自动机最后被细分为三个等价类

下面是 DFA::min\_Hopcroft() 算法的运行分析过程