类型：添加 ，StatePool.h

Line 58

// set how many state can be in the statepool

inline int set\_domain(const int r);

line 92

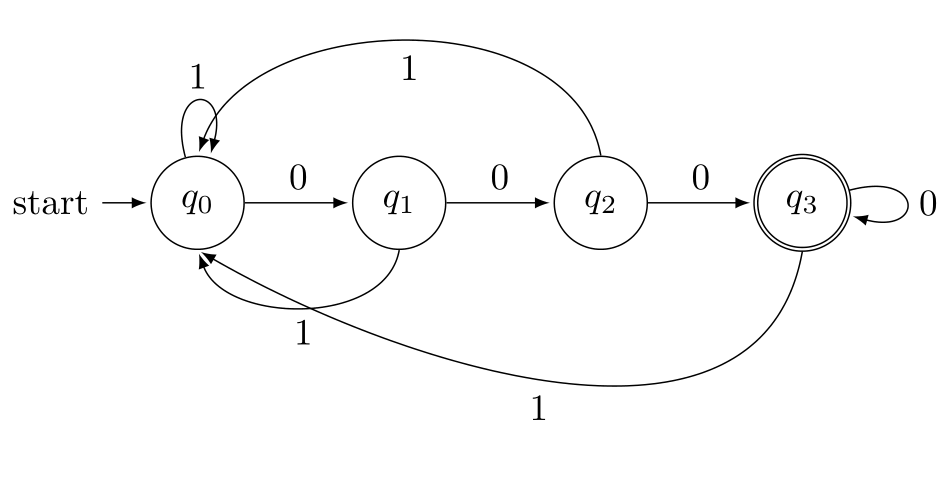
inline int StatePool::set\_domain(const int r)

{

return (next = r);

}

初始化这样一个DFA



其语言为

转换函数为

|  |  |  |  |
| --- | --- | --- | --- |
| 状态说明 | 状态 | 输入字符 | |
| 0 | 1 |
| start | q0 | q1 | q0 |
|  | q1 | q2 | q0 |
|  | q2 | q3 | q0 |
| final | q3 | q4 | q0 |

DFATest.cpp

|  |
| --- |
| void DFATest1()  {  DFA\_components dfa\_com1;  // StateSet S 开始状态集  dfa\_com1.S.set\_domain(10);  dfa\_com1.S.add(0);  // StateSet F 结束状态集  dfa\_com1.F.set\_domain(10);  dfa\_com1.F.add(3);    dfa\_com1.Q.set\_domain(10);  dfa\_com1.T.set\_domain(10);  dfa\_com1.T.add\_transition(0, '0', 1);  dfa\_com1.T.add\_transition(1, '0', 2);  dfa\_com1.T.add\_transition(2, '0', 3);  dfa\_com1.T.add\_transition(3, '0', 3);  dfa\_com1.T.add\_transition(0, '1', 0);  dfa\_com1.T.add\_transition(1, '1', 0);  dfa\_com1.T.add\_transition(2, '1', 0);  dfa\_com1.T.add\_transition(3, '1', 0);  dfa\_com1.T.add\_transition(1, '1', 4);  dfa\_com1.T.add\_transition(2, '0', 5);  dfa\_com1.T.set\_domain(10);  DFA dfa1(dfa\_com1);  cout << dfa1 << endl;  dfa1.min\_Brzozowski(); //首次最小化  cout << dfa1 << endl;  DFA dfa2(dfa1);  cout << dfa2 << endl;  dfa2.min\_Brzozowski();  cout << dfa2 << endl;  } |

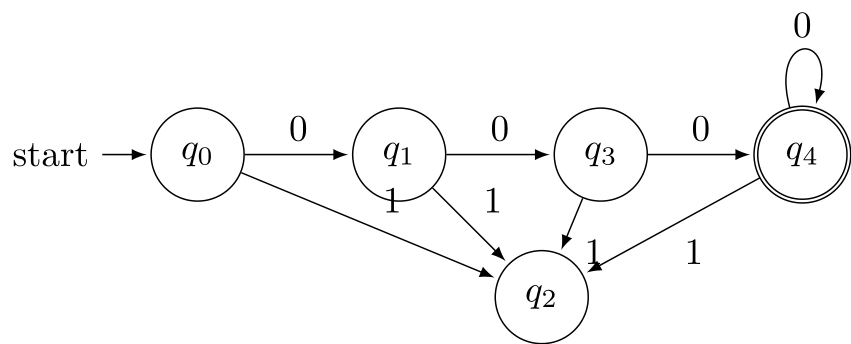
表现：与预期不符

经过第一次DFA::min\_Brzozowski()之后，转换函数成为

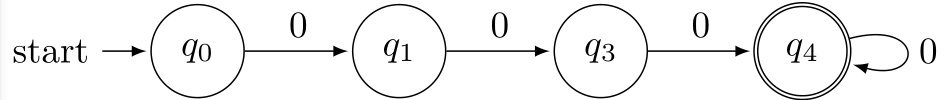
|  |  |  |  |
| --- | --- | --- | --- |
| 状态说明 | 状态 | 输入字符 | |
| 0 | 1 |
| start | q0 | q1 | q2 |
|  | q1 | q3 | q2 |
|  | q2 |  |  |
|  | q3 | q4 | q2 |
| final | q4 | q4 | q2 |

此DFA的语言为

关系图如下：



再次对其进行最小化，关系图变成



第二个DFA与第三个DFA等价，第一个DFA与他们都不等价。

结论：DFA::min\_brzozowski()最小化函数可能存在错误。