图灵机实验报告

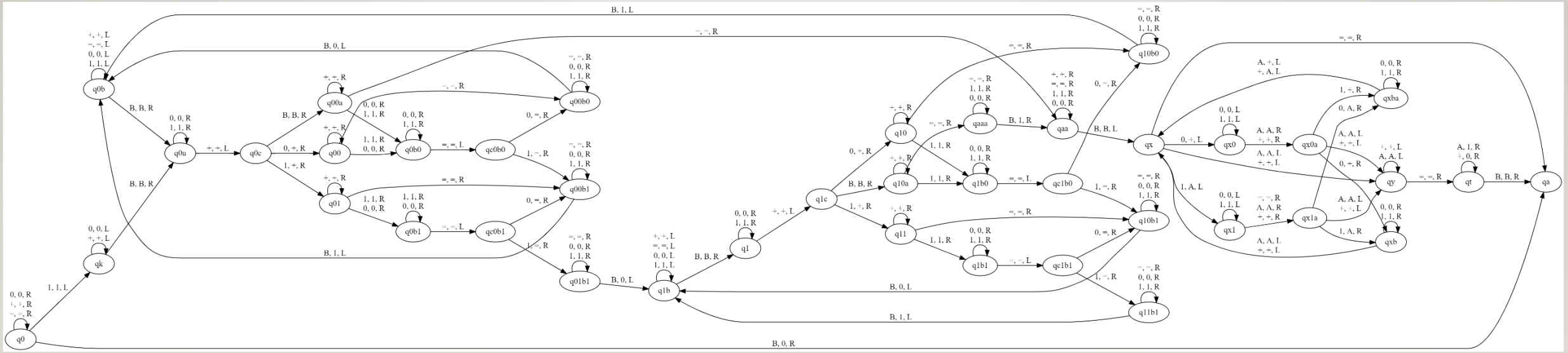
汇报人：第六组

吴子芃、舒景鸿、孙维铭、唐嘉良、余俊达

大致的思路（1）

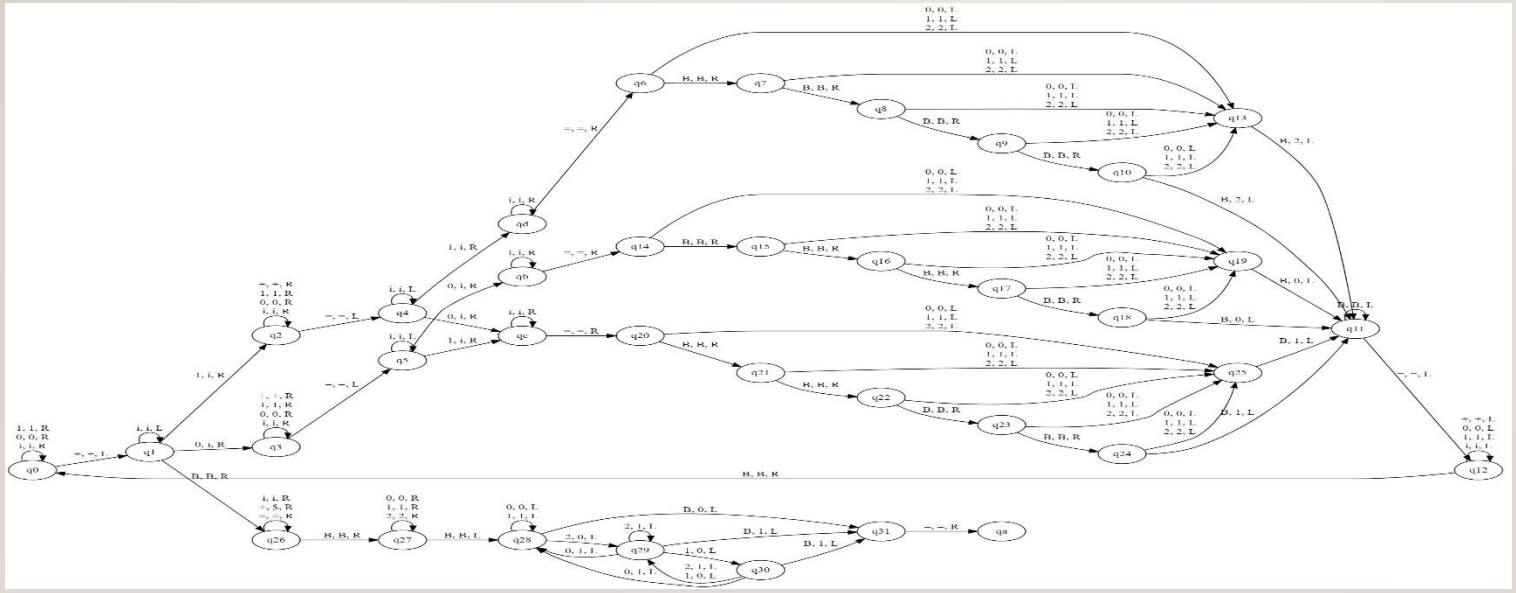
1、将进位通过返回的状态qback带回初始状态，完成进位。

2、将两个数字逐位相加，在等号右侧倒叙排列，然后再将数字反序。

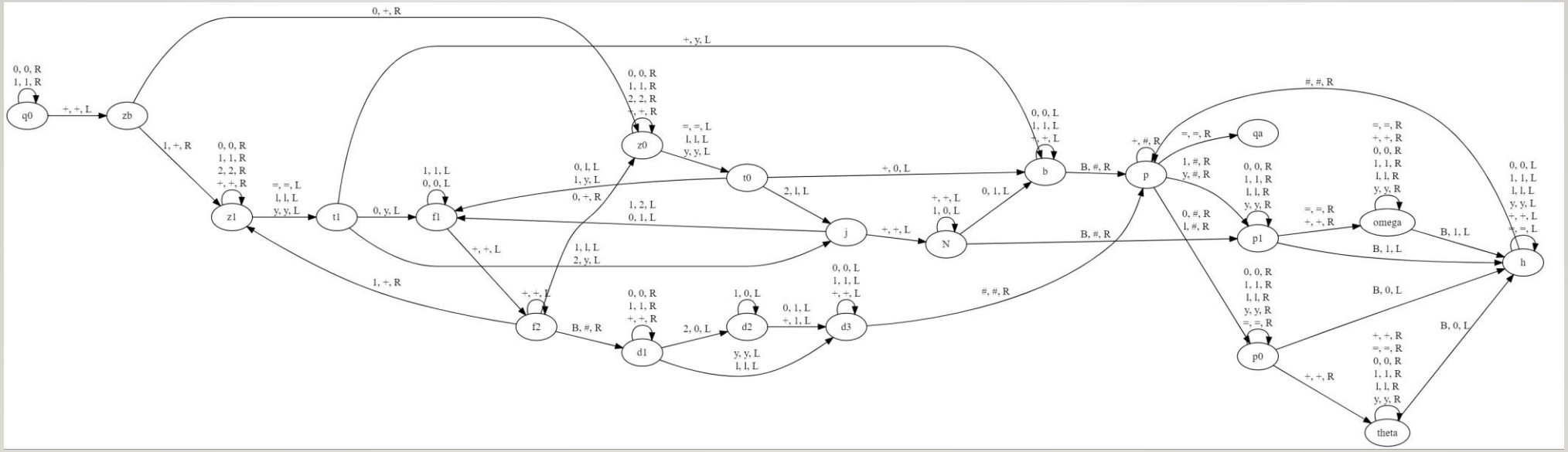


大致的思路（2）

- 1.第一题代码平凡，不去赘述
- 2.第二题中利用二进制的数字限制，将进位表达为2而非0与1，先行数值运算，再在最后一个循环中统一进位

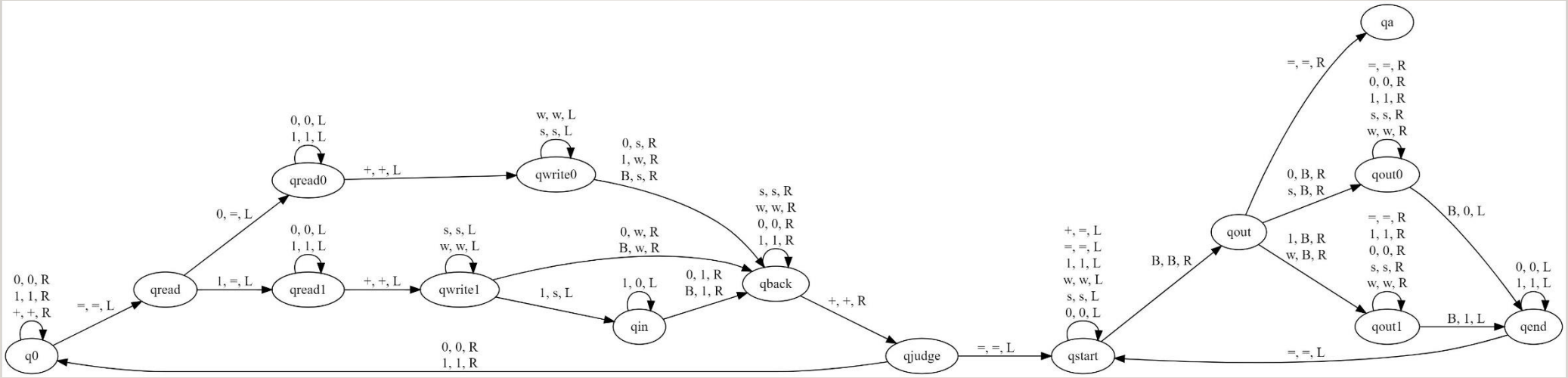


3.第三题则采用直接进位，先在等号左边完成运算，再将数值搬运到等号右边

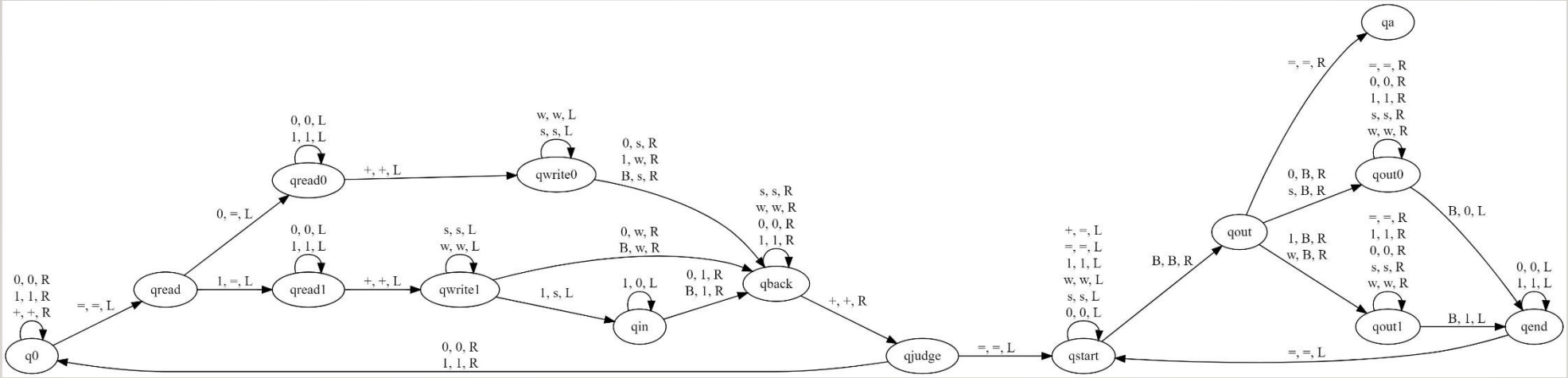


大致的思路（3）

- 第三题：每次将加号右边最后一位数加到加号左边，并将这一位改为等号。用两个特殊字符代替0、1表示加号左边当前计算的位置。最后再将加号左边的数搬运到等号右边（运算过程中进位）

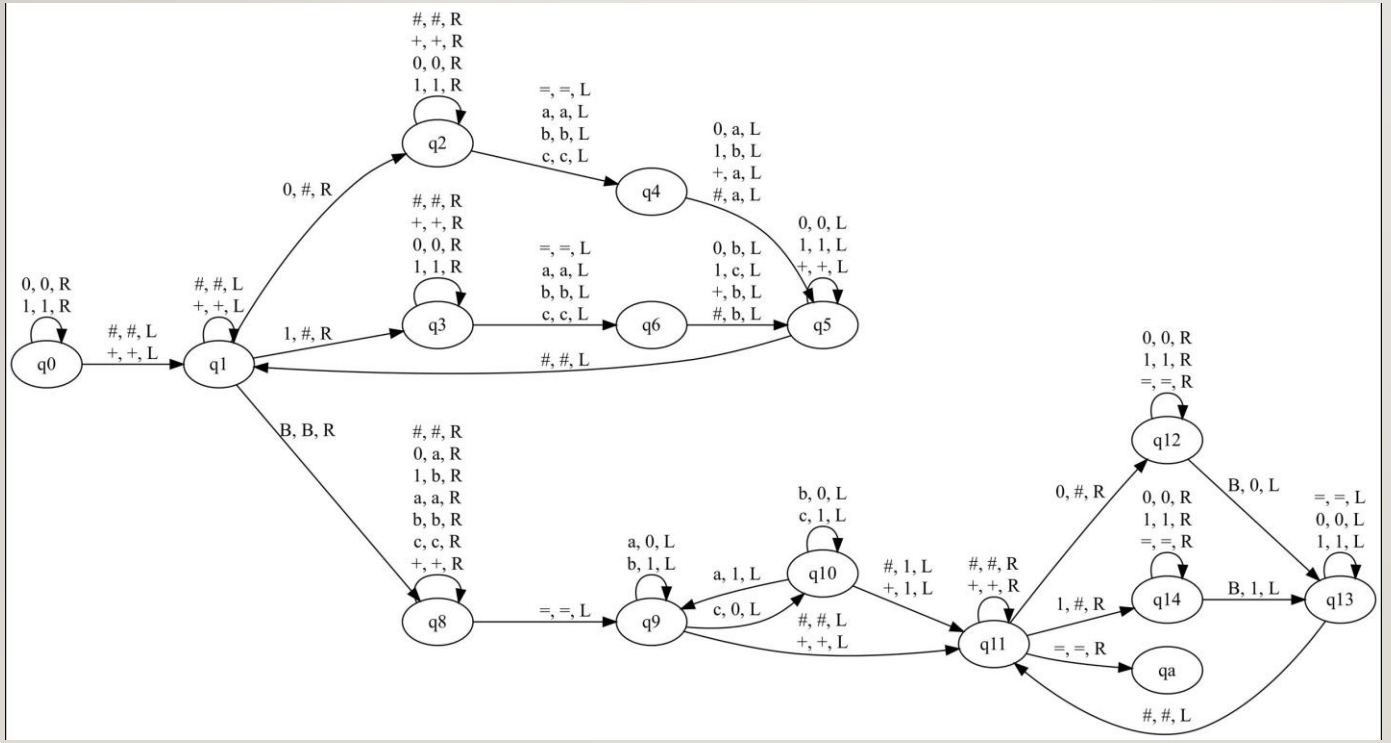


- 第二题：先在式子左边加0，再直接用第三题代码

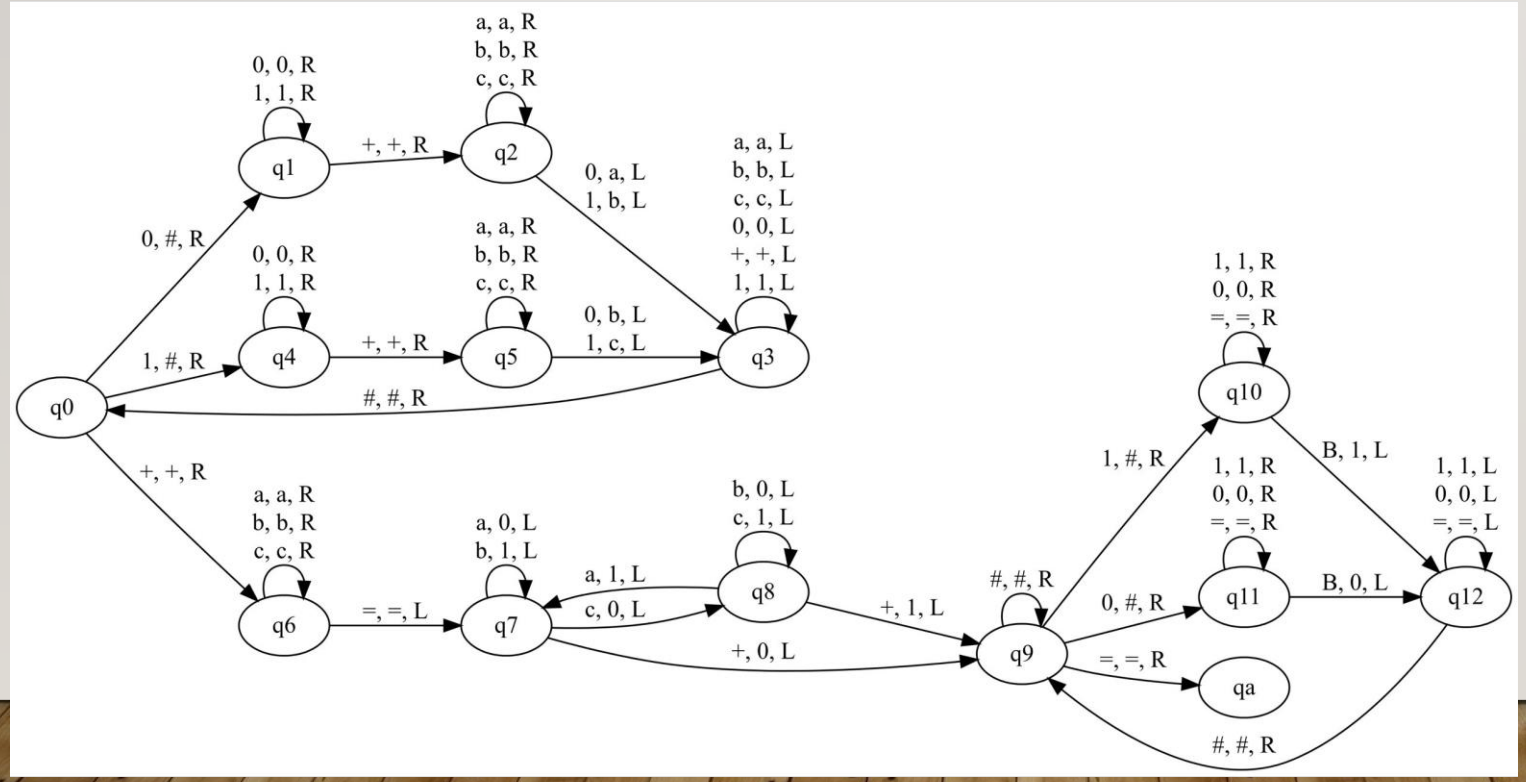


大致的思路（4）

- 第三题：和思路（3）类似，但是运算过程是把加号左边的数加到加号右边去



- 第二题：同第三题的思路，运算完之后如果结果只有四位，把+改成0搬到等号右边就变成了五位数。



遇到的困难

- 1、一开始想直接让数字在等号右侧按照顺序排列，但是难以找到末尾，于是先将其倒叙排列，再反序。
- 2.繁杂的代码使得一些难以被直接发现的bug诞生，只能通过大量数据测试来验明代码的可靠性
- 3.设计算法时可能会忽略某些特殊状态，导致极端情况下原算法不再适配
- 4.代码中的状态量过多会导致实际编写时搞混某些状态量

优化方法

- 1、减少所设变元数量。很多状态函数其实可以用一个代替。
- 2、奥卡姆剃刀原理：如无必要，勿增实体。有些情况是不可能出现的，所以没有必要探讨这些问题，从而减少状态转移函数。
- 3、充分考虑运算过程，构造更简单的运算方法。比如之前方法直接在等号右侧倒叙运算，不如现在等号左侧顺序运算，最后挪到右侧，这样大约能节省一半的步骤。

图灵机功能强大的原因

- 1.图灵机只需要有限的字符编码就可以进行一切人利用纸和笔能够进行的运算，并且无限长纸带表明它的计算能力是极强的（比现代的的计算机要更加强大）
- 2."丘奇 .图灵论题"， 断言:假设不考虑计算资源(脑力体力等)的限制，定义在自然数域，能被人通过算法计算的函数，也能被图灵机计算;反之亦然。

图灵机功能强大的原因

- 实际上，我们平时笔算乘法的思维过程，跟一台图灵机的运转非常相似：在每个时刻，我们只将注意力集中在一个地方，根据已经读到的信息移动笔尖，在纸上写下符号；而指示我们写什么怎么写的，则是早已背好的九九乘法表，以及简单的加法。如果将一个笔算乘法的人看成一台图灵机，纸带就是用于记录的纸张，读写头就是这个人和他手上的笔，读写头的状态就是大脑的精神状态，而状态转移表则是笔算乘法的规则，包括九九表、列式的方法等等。这种模式似乎也适用于更复杂的机械计算任务。如此看来，图灵机虽然看起来简单，但它足以作为机械计算的定义。
- 既然图灵机如此简单，能不能将它“升级”，赋予更多的硬件和自由度，使它变得更强大呢？比如说，让它拥有多条纸带和对应的读写头，而纸带上也不再限定两种符号，而是三种四种甚至更多种符号？的确，放宽限制之后，在某种程度上，对于相同的任务我们能设计出更快的图灵机，但从本质上来说，“升级”后的图灵机能完成的任务，原来的图灵机也能完成，虽然也许会慢些。也就是说，这种“升级”在可计算性上并没有意义，放宽限制后的机器能计算的，原来的机器也能完成。既然计算能力没有质的变化，无论采取什么样的结构，用多少种符号，都无所谓。

编程思想：模拟

- 模拟法解决问题的基本思想是对事物进行抽象，将现实世界的事物映射成计算机所能识别的**代码符号**，而将现实事物之间的**关系映射成运算或逻辑控制流**。抽象思想的一个简单例子就是整数的产生。数字取代了具体的事物用于对数量进行度量。人类从一只羊、一头狼、一条鱼、一棵树等事物中看到了共通的单位性，从感知中产生出了抽象的数字，而对这些数字进行运算的过程，则代表了这些事物的累加、增加或减少。从此，古人类对数的把握从具体的事务中解放出来。
- 模拟法是编程的基本功，没有复杂的公式和技巧，只需要读懂题目，照着逻辑，理对代码，基本都可以完成。这些题目非常有利于编程的初学者建立基本的代码感觉。需要提醒的是，有些模拟法的题目背景可能设定得错综复杂，稍微有所大意或者逻辑没有整理清楚就可能步入歧途，简单的问题也可能纠缠半天，不必要的复杂逻辑使得查错无处可循。

感谢垂听！