

加法器

基本思路：考虑模拟竖式加法，用特殊符号标识已经计算的

符号集：0 1 2, * # 以及空格。

状态集：

状态集由若干二元组 (**key**, **val**) 构成，分别表示当前的任务和存储的值：

Reset	Read	Trans	Inc	Write	Halt
0	-1	-1	0		
1	0	0	1		
	1	1			
		2			

具体含义如下：

- **Reset** 表示当前正在进行复位操作：
 - 0 表示现在处在加法阶段的左移。
 - 1 表示现在进行进位阶段的复位。
- **Read** 表示当且正在读取 x 或 y 的数值，且已经读取的数值是：
 - -1 表示读取了某个数的前导零。
 - 0, 1 为具体数值。
- **Trans** 表示正在运送刚才读入的数值（准备写入或还未开始读入下一位），**val** 的含义同理。
- **Inc** 为进位器，**val** 的表示当前进位。
- **Write** 的功能较为简单，表示已经写完的状态，向后补终止符。
- **Halt** 表示停止。

	0	1	2	-	*	#	,	空
(Reset,0)	((Reset,0), ,L)	((Reset,0), ,L)	((Reset,0), ,L)	((Read,0), ,L)	((Reset,0), ,L)	((Reset,0), ,L)	((Read,-1), ,R)	
(Reset,1)	((Reset,1), ,L)	((Reset,1), ,L)	((Reset,1), ,L)			((Inc,0), ,R)		
(Read,-1)	((Trans,0),--,R)	((Trans,1),--,R)		((Read,-1), ,R)	((Trans,-1),--,R)	((Trans,-1),--,R)		
(Read,0)	((Trans,0),--,R)	((Trans,1),--,R)		((Read,0), ,R)	((Trans,0),--,R)	((Trans,0),--,R)		
(Read,1)	((Trans,1),--,R)	((Trans,2),--,R)		((Read,1), ,R)	((Trans,1),--,R)	((Trans,1),--,R)		
(Trans,-1)	((Trans,-1), ,R)	((Trans,-1), ,R)	((Trans,-1), ,R)		((Read,-1), ,R)	((Trans,-1), ,R)	((Reset,1), ,L)	
(Trans,0)	((Trans,0), ,R)	((Trans,0), ,R)	((Trans,0), ,R)		((Read,0), ,R)	((Trans,0), ,R)	(Write,0,R)	
(Trans,1)	((Trans,1), ,R)	((Trans,1), ,R)	((Trans,1), ,R)		((Read,1), ,R)	((Trans,1), ,R)	(Write,1,R)	
(Trans,2)	((Trans,2), ,R)	((Trans,2), ,R)	((Trans,2), ,R)		((Read,2), ,R)	((Trans,2), ,R)	(Write,2,R)	
(Inc,0)	((Inc,0),0,R)	((Inc,0),1,R)	((Inc,1),0,R)				Halt	
(Inc,1)	((Inc,0),1,R)	((Inc,1),0,R)	((Inc,1),1,R)				(Halt,1,R)	
Write								((Reset,0),,,L)

上述空格表示不会遇到这种情况。

起始位置：**(Reset,0)**，任意非空位置和非最右位置。