

图灵机实现 $x+y$ 的两种方案及相关思考

方案一：

符号集：0，1，0'，1',*

状态集：start,noncarry,store1,store2,halt

规则：

(start,*)	(noncarry,*,L)
(noncarry,0/0')	(noncarry,0/0',L)
(noncarry,1')	(store1,0',L)
(noncarry,*)	(halt,*, -)
(store1,0')	(store1,0',L)
(store1,1')	(store2,0',L)
(store1,0)	(noncarry,1,L)
(store1,1)	(store1,0,L)
(store1,*)	(halt,1, -)
(store2,0/1)	(store1,0/1,L)

示例：11+7

10'01'11'11'

说明:

- 1.在不带上标的数 x 各位间每隔一位从左往右插入带上标的 y 的一位, 示例中 x 即为 1011, y 即为 111 (最左一位 0 略去, 仅用于对位).
- 2.系统从右往左开始运行, 初始状态为 start.
- 3.最后读取的结果为不带上标的数.

方案二：

符号集：0，1，*

状态集：start，nony（noncarry y），nonx（noncarry x），
storey1,storex1,storex2,halt

规则：

(start,*)	(nony,*,L)
(nony,0)	(nonx,0,L)
(nony,1)	(storex1,0,L)
(nony,*)	(halt,*, -)
(nonx,0/1)	(nony,0/1,L)
(storex1,0)	(nony,1,L)
(storex1,1)	(storey1,0,L)
(storey1,0)	(storex1,0,L)
(storey1,1)	(storex2,0,L)
(storey1,*)	(halt,1, -)
(storex2,0/1)	(storey1,0/1,L)

示例：11+7

10011111

说明:

- 1.从左往右, x 按奇数位输入, y 按偶数位输入, 多余的 0 用于对位.示例中 x 即为 1011, y 即为 111
- 2.此条同方案一.
- 3.若最左边*未变, 按奇数位读取结果, 若*变为 1, 则原*所在位置的 1 为第一位, 余下按原 x 所在位置读取余位.

一些思考与探索：

- 1.能否进一步简化？（删掉 start 和最右侧的*，不影响计算结果）
- 2.y 的输入问题：对 y 的输入采用了与 x 对位隔放的形式，如果将 x 单独输入，y 单独输入，中间仅用分隔符隔开会怎么样？
- 3.x, y 的存储：对于 x, y 各位信息的存储，我们用了两种方式，一种是增加符号数，一种是增加状态数，符号数增多则状态数减少，反之亦然.