

## 获得的答案

**Recursion theorem:** Let  $T$  be a Turing machine that computes a function  $t: \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$ . There is a Turing machine  $R$  that computes a function  $r: \Sigma^* \rightarrow \Sigma^*$ , where for every  $w$ ,

$$r(w) = t(\langle R \rangle, w)$$

- The recursion theorem produces a new machine  $R$ , which operates exactly as  $T$  does.

- It has connection to the theory of self-reproducing system.

$SELF$  = "On any input:

1. Obtain, via the recursion theorem, own description  $\langle SELF \rangle$ .

2. Print  $\langle SELF \rangle$ "

- In the real programming Language, LISP plays the same role as a recursion theorem.

- The following program in LISP is an example in the spirit of recursion theorem.

```
(define self (lambda (x)
  (list x (list (quote quote) x))
))
(quote (lambda (x)
  (list x (quote quote) x)
))
```

(quote x) represents in Lisp as , (lambda(x) (list x(quote quote) x)) is printing of word x is initialization. Whole program represent self printing.