PUC

Series: Monographs in Computer Science
and Computer Applications

Nº 7/69

A SLIP APPLICATION: THE CONSTRUCTION OF TURING MACHINES

by

ROBERTO LINS DE CARVALHO

Computer Science Department - Rio Datacenter

CENTRO TÉCNICO CIENTÍFICO
Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 209 — ZC-20
Rio de Janeiro — Brasil

Simão Sirineo Toscani - 1969

A SLIP APPLICATION: THE CONSTRUCTION OF TURING MACHINES

ROBERTO LINS DE CARVALHO
COMPUTER SCIENCE DEPARTMENT
PUC - RIO DE JANEIRO

A SLIP Application: the Construction of Turing Machines

1. Introduction

In the first place we will try to define, in a concise way, what is meant by a turing machine [1], and we will also presented one of the possible formulations. A turing machine is a device composed of a read and write head and a tape unit, of arbitrary lenght. Fig. 1 below shows how it looks like, for the purpose of this paper.

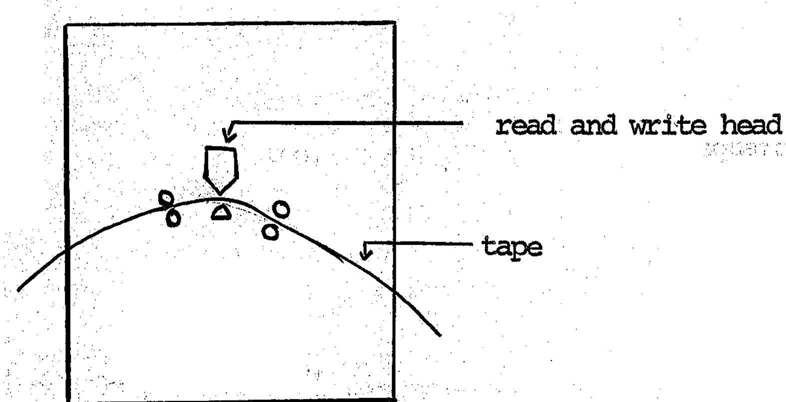

Fig. 1

We imagine the tape subdivided into squares, in such a way that each square can contain only one symbol.

The following operations are admitted on turing machines:

a) Write a symbol on the tape;

b) Right displacement relative to tape;

c) Left displacement relative to tape.

The following is one of the formulations of a turing machine:

Definition:

A turing machine is a set of quadruples of the form:

(i) $q_i\ s_j\ s_k\ q_1$

(ii) $q_i\ s_j\ R\ q_1$

(iii) $q_i\ s_j\ L\ q_1$

Remark:

$q_i\ s_j\ s_k\ q_1$ - means that the machine is at the state q_i, the read and write head reads the symbol S_j, write the symbol S_k, and the machine turns to state q_1.

$q_i\ s_j\ \underset{L}{R}\ q_1$ - is similar to the above, with the difference that instead of writing a symbol the head moves to the right (left), or the tape moves to the left (right).

2. Numerical Computation

Let $x=(x_1,x_2,\ldots,x_n)$ be a n-pla, where each component represents a certain address relative to a tape, i.e., each is a delimiter of a certain area of the tape; also let \$ (dollar sign) be a pointer to the beginning of the tape. Each integer number is represented either by a sequence of digits "1" or by a sequence of "|": for example, the number 5 is represented by |||||. The tape has the following appearence:

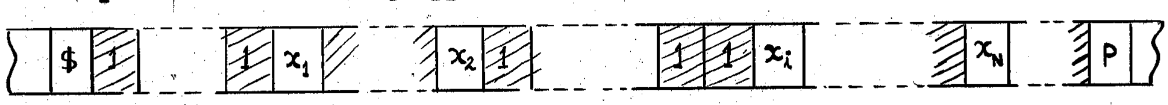

Fig. 2

3. A turing machine that determines the successor of a component

Let us suppose that during the computation of a function $f: N^n \rightarrow N$ the machine is at the state q_m and that the read and write head us positioned in any part of the magnetic tape. The set of quadruples, below, simulates the operation $x_i + 1 \rightarrow x_i$.

1. $q_m\ \bar{\alpha}\ L\ q_m$
2. $q_m\ \bar{\alpha}\ x_i\ q_{m1}$

Looks for the beginning of the tape, replaces α for x_i (pointer to the address of the i.th component), and displaces α a square left, passing to

3. q_{m1} x_i L q_{m1} 4. q_{m1} B α q_{m1} 5. q_{m1} α R q_{m2}	the state q_{m2}.
6. q_{m2} $\overline{x}_i$ R q_{m2} 7. q_{m2} x_i R q_{m3}	Looks for the first pointer x_i and then passes to state q_{m3}.
8. $q_{m\_}$ $\overline{x}_i$ R q_{m3} 9. q_{m3} x_i L q_{m4}	Looks for the second pointer x_i, and then passes to state q_{m4}.
10. q_{m4} B L q_{m4} 11. q_{m4} x R q_{m5} 12. q_{m5} $\overline{x}_i$ R q_{m5} 13. q_{m5} x_i B q_{m6}	If a digit "|" is not formed in the area corres̲ponding to x_i, eliminates the pointer and passes to state q_{m6}.
14. q_{m6} $\overline{x}_i$ L q_{m6} 15. q_{m6} x_i 1 q_{m7} 16. q_{m7} 1 L q_{m7} 17. q_{m7} B x_i q_{m+1}	Looks for the beginning, put another digit "|" at the-new address for x_i, displaces one square left and finally passes to state q_{m+1}
18. q_{m4} 1 B q_{m8} 19. q_{m8} $\overline{x}_i$ L q_{m8} 20. q_{m8} x_i L q_{m9} 21. q_{m9} 1 L q_{m9} 22. q_{m9} α 1 q_{m10} 23. q_{m10} 1 L q_{m10} 24. q_{m10} B α q_{m2}	If a "|" is found in the corresponding area for x_i, erase this |, returns to the beginning of the tape, put another "|" at the new position for x_i, displaces one square left and set back to state q_{m2}.

4. Application of the SLIP languages

In the following we will try to obtain a program that simulates the machine described above; some changes will name to be done with respect to the tape organization the principal difference is that the delimiter will be placed at the beginning of the reserved area. On the other side we will eliminate the wquadruples concerning the displacement of α, because in SLIP the user need not be concerned with the displacements relative to the insertion of nodes. We will use the letters of the alphabet as area delimiters and as the name of the list, in place of α, the symbol $. The tape organization is showing below:

($ (A |||...|) (B ||...|)...(I ||...|)...(N ||..|))

In the program that follows we introduced enough comments to help the under standing of the machine performance.

5. Program Description

```
C**** EXEMPLO 2 - TURING MACHINE
C**** THIS MACHINE CALCULATES THE SUCCESSOR OF THE I-TH COMPONENT OF THE N TUPLE
           (X1,...,XN)
C*****INITIALIZATION
      COMMON AVSL,W(100)
      DIMENSION-SPACE(5000),L(20),IENDCO(20),ITAPE(1000),ICOMP(20)
      INTEGER BLK,UM
      DATA MARK,IENCO/1H$,1HA.1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1
     1HL,1HM,11HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT/,UM,BLK/1H1,1H /
      KKK=5
      IPROJ=IENDCO(KKK)
      N=10
      READ(5,1)(I),I=1,N)
    1 FORMAT(16I5)
C**** LIST STRUCTURE
      CALL INITAS(SPACE,5000)
```

```
      CALL LIST(LL)
      CALL NEWTOP(MARK,LL)
      DO 2 I=1,N
    2 CALL NEWTOP(IENDCO(I),LIST(L(I)))
      DO 3 J=1,N
      K=ICOMP(J)
      DO 3 I=1,K
    3 CALL NEWBOT(UM,L(J))
      DO 4 I=1,N
    4 CALL NEWBOT(L(I),LL)
      M=LRDROV(LL)
      DO 5 I=1,1000
      ITAPE(I)=INTIGER(ADVSER(M,F))
      IF(F)6,5,6
    5 CONTINUE
    6 II=I-1
      WRITE(6,7)(ITAPE(I),I=1,II)
    7 FORMAT(//10X,15HTAPE DE ENTRADA,/10X,120A1,//)
C**** STATE QM SEARCHS THE BEGINNING OF THE TAPE
    8 LEITOR=INTIGER(ADVSEL(M,F))
      IF(LEITOR.NE.MARK)GO TO 8
C**** STATE QM1(BEGINNING OF THE TAPE FOUND)
      LEITOR=INTIGER(ADVSEL(M,F))
      CALL POPTOP(LL)
      CALL NEWTOP(IPROJ,LIST(L(N+1)))
      CALL NEWTOP(L(N+1),LL)
      CALL NEWTOP(MARK,LL)
C**** IF IPROJ IS FOUND PASSES TO STATE QM2
    9 LEITOR=INTIGER(ADVSER(M,F))
      IF(LEITOR.NE.IPROJ)GO TO 9
C**** STATE QM2-LOOKS FOR THE SECOND IPROJ
   10 LEITOR=INTIGER(ADVSER(M,F))
      IF(F)9,11,9
   11 IF(LEITOR.NE.IPROJ)GO TO 10
```

```
C**** STATE QM3 FOUND THE SECOND IPROJ,PASSES TO STATE QM4
   12 LETTOR=INTGER(ADVSER(M,F))
      IF(LETTOR.EQ.BLK)GO TO 12
      IF(LETTOR.EQ.IENDCO(KKK+1))GO TO 15
      IF(F)15,13,15
C**** A 1 IS FOUND-SUBSTITUTES BY A BLANK AND PASSES TO STATE QM8
   13 K1=LPNTR(M)
      LETTOR=INTEGER(ADVSER(M,F))
      CALL SUBST(BLK,K1)
   46 LETTOR=INTGER(ADVSEL(M,F)
      IF(F)47,46,47
   47 LETTOR=INTGER(ADVSER(M,F))
      LETTOR=INTGER(ADVSEL(M,F))
      CALL NEWBOT(UM,L(N+1))
      IN=LRDROV(LL)
      DO 80 I=1,1000
      ITAPE(I)=INTGER(ADVSER(IN,G))
      IF(G)81,80,81
   80 CONTINUE
   81 WRITE(6,85)(ITAPE(I),I=1,II)
   85 FORMAT(//10X,16HSITUACAO DO TAPE,/10X,120A1)
      GO TO 9
   15 LETTOR=INTGER(ADVSEL(M,F))
      IF(LETTOR.NE.MARK)GO TO 15
      LETTOR=INTGER(ADVSER(M,F))
      CALL NEWBOT(UM,L(N+1))
      CALL MTLIST(L(KKK))
      M=LRDROV(LL)
      DO 16 I=1,1000
      ITAPE(I)=INTGER(ADVSER(M,F))
      IF(F)17,16,17
   16 CONTINUE
   17 II=I-1
```

```
    WRITE(6,70)(ITAPE(I),I=1,II)
 70 FORMAT(//10X,13HTAPE DE SAIDA,/10X,120A1,//)
    CALL IRARDR(M)
200 FORMAT(10X ,22HO READER APONTA PARA ,2H**,A1,2H**,/)
    CALL EXIT
    END
```

6. Conclusions

The method presented here can be used in the construction of any of turing machine, and we found that SNP is a language will adapted for such treatments. Our program is not necessarily the best, but we believe it can be a guide for work.

References

1. Nelson, R.J. "Introduction to Automata". Wiley and Sons, Inc.

2. Rosen, Saul "Programming Systems and Languages" Mc Graw Hill Book Company.