

DESCRIPTION OF PREDATION MODEL  
PREDATOR-PREY FUNCTIONAL RESPONSE

C. S. Holling

September 1973

Research Memoranda are informal publications relating to ongoing or projected areas of research at IIASA. The views expressed are those of the author, and do not necessarily reflect those of IIASA.



Description of Predation Model

Predator-Prey Functional Response

(Determination of Rate of Attack Density)

C.S. Holling\*

I. Equations Used in Model

1. Hunger as a Function of Time of Food Deprivation

Experimentally it has been found that hunger varies with the time T of food deprivation according to a relation of the form

$$dH/dT = AD*(HK - H) \quad , \quad (1)$$

where AD is a constant, the rate of food digestion (gm/hr), and HK is maximum gut capacity in grams.

The hunger H is defined here as the weight of food necessary to return the animal to a condition of complete satiation, since this weight is a measure of the emptiness of the digestive tract.

Integrating (1) we have

$$(H - HK) = K*EXP(-AD*T) \quad ,$$

where K is a constant of integration. For an initial condition we define H = HO at T = 0. Then

$$H = HK + (HO - HK)*EXP(-AD*T) \quad . \quad (2)$$

This equation will be used throughout the program to update hunger levels. Suppose that H = H1 at time T1 and we wish to find the hunger level after an additional time T2 has passed. With HO replaced by H1 and T1 by T2, equation (2) yields the

---

\*Institute of Animal Resource Ecology, University of British Columbia, Canada. International Institute for Applied Systems Analysis, Laxenburg, Austria, 1973-74.

new hunger level. In other words,  $H_0$  represents the hunger level at the start of the time interval  $T$ .

## 2. Hunger Thresholds

The threshold hunger level for pursuit  $H_{TP}$  is obtained from the following equation relating  $H_{TP}$  and prey length  $L$

$$\begin{aligned}d(H_{TP})/dL &= (L - L_{OPT})/L * MYOP \\d(H_{TP}) &= MYOP * dL - L_{OPT} * MYOP * dL/L \quad (3) \\H_{TP} &= MYOP * L - L_{OPT} * MYOP * A \log(L) + K .\end{aligned}$$

Now for  $L = L_{OPT}$ ,  $H_{TP}$  has a minimum value, which we will call  $H_{OPT}$ . This allows evaluation of the constant  $K$  and we find

$$H_{TP} = MYOP * L_{OPT} * [L/L_{OPT} + A \log(L_{OPT}/L) - 1] + H_{OPT} . \quad (4)$$

The other hunger thresholds are expressed in terms of  $H_{TP}$

$$H_{TC} = CAPT * H_{TP} \quad (5)$$

$$H_{TE} = EAT * H_{TC} . \quad (6)$$

## 3. Digestive Pause Time

Using equation (2) we can represent the time required for hunger to rise from  $H_0$  to  $H$  as

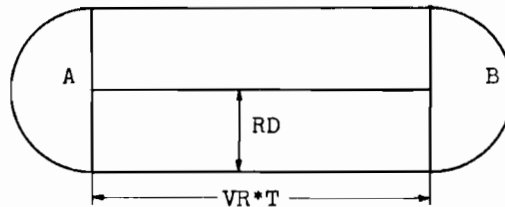
$$T = 1/AD * A \log[(HK - H_0)/(HK - H)] . \quad (7)$$

The digestive pause represents the time taken for the hunger to rise from  $H_0$  to  $H_{TE}$

$$\begin{aligned}TD &= 1/AD * A \log[(HK - H_0)/(HK - H_{TE})] \quad \text{for } H_0 < H_{TE} \\TD &= 0 \quad \text{for } H_0 \geq H_{TE} .\end{aligned} \quad (8)$$

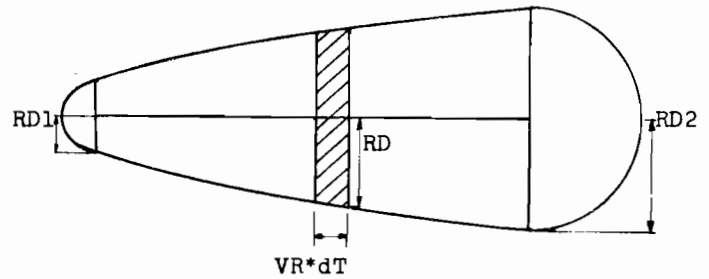
4. Search Time

To derive a general expression for the search time  $T_S$ , consider a predator moving from A to B searching for prey. If we assume a constant hunger level, the area swept out by the search will be of this form:



$$AREA = KA * HONP * (L + EYE) * (H - HTP) \quad . \quad (14)$$

Thus in the general case, AREA, and hence also RD in our original equation (11), change with hunger. The search area swept out by the predator will have a form more like this:



The above equation for  $T_S$  must be changed to reflect the change in RD

$$G = 2 * VR * \int_0^{T_S} RD * dT + .5 * PI * [(RD1)**2 + (RD2)**2] \quad . \quad (15)$$

Consider first the integral, and substitute for RD an average distance of reaction P, defined as

$$P^{**2} = KA/PI * HONP * (L + EYE) * (H - HTP) \quad (16)$$

$$P = \text{SQRT}(KA * PO^{**2} / PI) \quad .$$

Substituting equation (13) for PO and (2) for hunger, we have

$$P = \text{SQRT}\{KA/PI * HONP * (L + EYE) * [HK - HTP - (HO - HK) * \text{EXP}(-AD * T)]\} \quad . \quad (17)$$

Let

$$QA = \text{SQRT} [KA/PI * HONP * (L + EYE)]$$

$$B = HK - HTP$$

$$C = HO - HK \quad .$$

Then

$$P = QA * \text{SQRT} [B + C * \text{EXP}(-AD * T)] \quad . \quad (18)$$

Now to evaluate the integral in equation (15).

$$\int_0^{TS} RD * dT = \int_0^{TS} P * dT = \int_0^{TS} QA * \text{SQRT} [B + C * \text{EXP}(-AD * T)] * dT$$

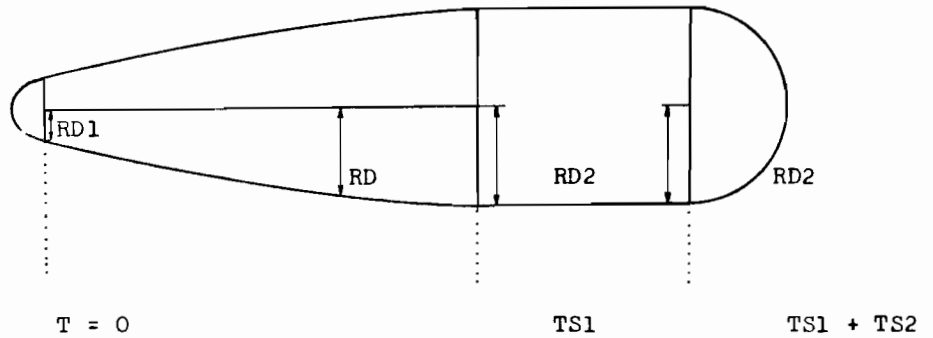
$$= - \frac{QA}{AD} \left\{ 2 \sqrt{B + C e^{-AD \cdot T}} + \sqrt{B} \ln \left| \frac{\sqrt{B + C e^{-AD \cdot T}} - \sqrt{B}}{\sqrt{B + C e^{-AD \cdot T}} + \sqrt{B}} \right| \right\} \Bigg|_{T=0}^{T=TS} \quad .$$

Let  $W = \text{EXP}(-AD * TS)$ . Then

$$\int_0^{TS} P * dT = - \frac{QA}{AD} \left\{ 2 \sqrt{B + C \cdot W} + \sqrt{B} \ln \left| \frac{\sqrt{B + C \cdot W} - \sqrt{B}}{\sqrt{B + C \cdot W} + \sqrt{B}} \right| - 2 \sqrt{B + C} - \sqrt{B} \ln \left| \frac{\sqrt{B + C} - \sqrt{B}}{\sqrt{B + C} + \sqrt{B}} \right| \right\} \quad . \quad (19)$$

In some cases of low prey density the search time may be large enough that  $C * W$  in this expression is negligible; equation (19) will then contain  $\text{ALOG}(0)$  which is meaningless and which will cause the computer to choke up.

To get around this problem we will first suppose that the predator's reactive distance increases with hunger as described, but only until such time as his gut is empty; thereafter his reactive distance will remain constant since his hunger, by our definition of hunger, has reached its maximum value.



Now let  $T$  represent the total search time, which is made up of the time  $TS1$  during which the predator's reactive distance is still increasing with hunger, and the time  $TS2$  during which the reactive distance is constant at its maximum value. We will consider the maximum hunger level to have a value of  $.99*HK$ , i.e. for which the gut contains only one percent of its capacity.

$TS1$  can be calculated as the time required for the hunger level to go from its present value to  $.99*HK$ :

$$TS1 = A \log \left[ \frac{HK - H_0}{.01*HK} \right] / AD \quad (20)$$

and  $TS2$  is simply  $(T - TS1)$ .

Of course if  $TS1 > T$  as calculated by equation (20) then  $TS1$  is replaced by  $T$  and represents the total search time,  $TS2$  being zero. Now equation (19) will be integrated only between  $T = 0$  and  $T = TS1$ . Equation (15) becomes

$$G = 2*VR* \int_0^{TS1} P*dT + 2*VR*(T - TS1)*RD1 + .5*PI*[(RD1)**2 + (RD2)**2] ; \quad (21)$$

the last term represents the area of the semi circular parts of the reactive field. The radius  $RD1$  of the first one is

given by (16) calculated for  $T = 0$

$$\begin{aligned} RD1 &= \text{SQRT}[KA/PI * HONP * (L + EYE) * (HO - HTP)] \\ &= QA * \text{SQRT}(HO - HTP) . \end{aligned} \quad (22)$$

RD1 is obtained from (16) by considering the term  $(HO - HK) * \text{EXP}(-AD * T)$  as negligibly small:

$$RD2 = QA * \text{SQRT}(HK - HTP) ; \quad (23)$$

thus the total area of the two semicircles is

$$.5 * PI * QA ** 2 * [(HO - HTP) + (HK - HTP)] . \quad (24)$$

We now have expressions for each of the components of equation (15) which we can write in the form

$$\begin{aligned} 2 * VR * \int_0^{TS1} [P * dT + (T - TS1) * RD2] + .5 * PI * [(RD1) ** 2 \\ + (RD2) ** 2] - G = 0 . \end{aligned} \quad (25)$$

#### 5. Pursuit Time

The velocity of pursuit VP is assumed constant and unaffected by hunger. Now the general expression for the area of the reactive field of any predator is given by  $A = KA * PO ** 2$ . Let us assume that the average distance RM to the outer contour of the reactive field is equal to the radius of a circle with the same area.

$$PI * RM ** 2 = KA * PO ** 2$$

$$RM = PO * \text{SQRT}(KA/PI) .$$

Let DS be the average strike distance and DP the average pursuit distance:



$$DP = RM - DS$$

$$DP = \text{SQRT}[KA/PI * HONP * (L + EYE) * (H1 - HTP) - DS]$$

for  $H1 > HTP$

$$DP = 0 \quad \text{for } H1 \leq HTP \quad ,$$

where  $H1$  is the updated hunger level after the search routine.

The pursuit time is simply  $DP/VP$ :

$$TP = \{\text{SQRT}[KA/PI * HONP * (L + EYE) * (H1 - HTP)] - DS\} / VP$$

for  $H1 > HTP$  (26)

$$TP = 0 \quad \text{for } H1 \leq HTP \quad .$$

#### 6. Eating Time

If  $W$  is the number of grams of prey weight and  $AKE$  is the time required by the predator to eat one gram weight of prey, then

$$TE = AKE * W \quad \text{when } H1 \geq W$$
$$TE = AKE * H1 \quad \text{when } H1 < W \quad .$$

(27)

The uneaten waste is  $WD = W - H1$ .

#### 7. Pursuit Success

$TSP$  was defined as the proportion of prey not moving away during pursuit. Assuming that the durations of prey movements are randomly distributed, the possibility of a prey not moving is given by  $EXP(-AM * T)$  where  $AM$  is a constant and  $T$  is time (zero frequency class of a poisson distribution). Then the number of periods of immobility lasting  $T$  units of time is  $N * EXP(-AM * T)$  where  $N$  is the total number of periods of immobility. Then

$$TSP = (1/N) * N * EXP(-AM * TP) = EXP(-AM * TP) \quad , \quad (28)$$

where  $T$  has been replaced by  $TP$ .

## II. Structure of the Program

### 1. FNEW(t)

FNEW is a function subroutine called by GETTS to evaluate the integral equation (25) for search time. The equation is expressed in the following way in the program:

$$FNEW = F1 + F2 + F3 - G ,$$

where

- a) F1 represents the two semicircular search areas. If the predator's hunger reaches a maximum during the search F1 is given by equation (24). If not, the second semi-circular area is calculated using the updated value of hunger level after each iteration. Thus, if HTEMP is the updated hunger level, then

$$F1 = \begin{cases} .5*PI*QA**2*((HC - HTP) + (HTEMP - HTP)) & HO > HTP \\ .5*PI*QA**2*(HO - HTP) & HO \leq HTP \end{cases} ;$$

- b) F2 is the rectangular area whose constant width is the reactive distance for maximum hunger and whose length is the distance searched by the predator in time TS2. If TS2 = 0, F2 is zero. Otherwise

$$F2 = 2*VR*(T - TS1)*QA*SQRT(HK - HTP) ;$$

- c) F3 is the integral part of the equation, the area swept out while the reactive distance is increasing with hunger. It is given by equation (19) with W replaced by EXP(-AD\*TS1).

### 2. GETTS

This subroutine calculates a value of TS for given values of hunger and prey density.

A. If FR is run without prey avoidance, PVULN is used only as a switch and is set equal to -1. Otherwise it represents the proportion of prey whose reactive distance is less than that of the predator.

B.  $G$  is calculated according to the value of  $PVULN$ , and  $AREA$  according to the value of  $(H1 - HTP)$ . A number of values used repeatedly in  $FNEW$  are calculated.

C. If  $AREA \geq G$  there is already a prey within the reactive field of the predator. In this case  $T = 0$  and  $GETTS$  returns this value to  $FR$ .

D. If  $IFLD < 0$ ,  $TS$  is calculated directly from equation (11)  $IFLD \leq 0$  indicates that reactive distance does not change with hunger and  $GETTS$  returns to  $FR$ .

E. If  $AREA < G$  or  $IFLD > 0$ ,  $TS$  is calculated by iteration. Since the method used converges very quickly the initial estimate of  $TS$  is simply made equal to 1.

F. This initial value is substituted into equation (22). (Calculation of the function is done by the function subroutine  $FNEW(T)$ .) Depending on whether the resulting value  $FNEW(TTMAX)$  is i) positive or ii) negative (positive means that  $TTMAX$  is greater than the real value of  $TS$ , negative means  $TTMAX$  is less than  $TS$ ),  $TTMAX$  is assigned to  $TMAX$  or  $TMIN$ , which represent respectively maximum or minimum bounding values for  $TS$ .

In case (i)  $TMAX$  is divided by 10. If the resulting value is less than a specified small value  $EPS$ ,  $TS$  is set equal to zero, and  $GETTS$  returns to  $FR$ . As long as  $FNEW(TMAX) > 0$ ,  $TMAX$  is repeatedly replaced by  $TMAX/10$ . As soon as  $FNEW(T)$  becomes less than zero,  $TMIN$  is set equal to this value of  $T$ , while  $TMAX$  retains its previous value. (If  $FNEW(T) = 0$ , then  $TS = T$ , and  $GETTS$  returns to  $FR$ .) We now have upper and lower bounds,  $TMAX$  and  $TMIN$ , on the value of  $TS$ .

In case (ii)  $TTMAX$  is multiplied by 10. As long as  $FNEW(TTMAX) < 0$ ,  $TMIN$  is replaced by  $TTMAX$  giving us again upper and lower bounds on  $TS$ .

G. Now that two bounding values have been found, intermediate values of  $TS$  are chosen and substituted into equation (22). The resulting values of  $FNEW$  will approach closer and closer to zero until either the desired accuracy has been obtained or until 100 iterations have been made. At this point  $GETTS$  returns the current value of  $TS$  to the calling program  $FR$ .

### 3. FR

The subroutine  $FR$  calculates the number of attacks per day per predator on prey present in a given density. This is done by simulating individual attack cycles and calculating the number of cycles generated. There are four components to

each attack simulation--the predator's digestive pause, the search for prey, the pursuit of a found prey, and the eating of it.

The digestive pause TD is the time taken for the predator's hunger to reach the eat threshold level HTE, measured from the end of the last eating period. Once the hunger level has reached HTE, the predator becomes responsive to the presence of prey. If his reactive distance is a function of hunger, his required search time will shorten as he gets hungrier. The time spent searching, TS, is measured from the moment the predator's hunger level reaches HTE until a prey is located and recognized. Then it takes the predator a time TP to pursue the prey and a time TE to eat it.

The total time taken for the attack cycle, T1, is

$$T1 = TD + TS + TP + TE .$$

Each of these four elements can be calculated as functions of the predator's hunger level from the equations previously described.

An arbitrary hunger level is chosen for the first cycle, and thereafter the level at the end of each cycle becomes that used for the start of the next. Complete attack cycles are repeatedly generated until the cycle time T1 has been stabilized; i.e. until two consecutive values of T1 differ by less than one percent. The detailed structure of the program is as follows:

#### A. Digestive Pause

- A1. The initial hunger level HO is compared with HTE. If  $HO > HTE$ , the predator is ready to start searching again. The digestive pause TD is zero. HO is stored.
- A2. If  $HO < HTE$ , the predator waits until his hunger rises to HTE before searching. This wait time TD is calculated from equation (8). The initial hunger level HO is stored and the increased to HTE, its value after time TD has passed.

#### B. Preliminary Estimate of TP

This procedure is necessary to obtain an initial value of pursuit success TSP, which is needed by the subroutine GETTS in the calculation of G (equation (9)). Since TSP is a function of pursuit time (equation (25)), an initial estimate of TP must

be found before the pursuit part of the attack is actually simulated.

- B1. If  $HO > HTP$ , TP is calculated from equation (26) and TSP from equation (28).
- B2. If  $HO \leq HTP$  or if TP as calculated in B1 is not greater than zero, then TP is set equal to zero and TSP = 1.

C. Search

- C1. The reactive distance is calculated from equation (22). GETTS is called and returns a value of the search time TS.
- C2. If  $TS \leq 0$ , the rest of the search routine and the pursuit routine are bypassed; TS is set equal to zero and TP has the value determined in B. The current hunger level (denoted by H1 in the search and pursuit routines) is updated to its value after time TP, using equation (2):

$$H1 = HK + (HO - HK) * \text{EXP}(-AD * TP) \quad .$$

- C3. If  $TS > 0$ , the current hunger level H1 is updated to the end of the search routine (equation (2)):

$$H1 = HK + (HO - HK) * \text{EXP}(-AD * TS) \quad .$$

D. Pursuit

- D1. If  $H1 > HTP$ , TP1 is calculated from equation (26) and TSP from equation (28).
- D2. If TP1 as calculated in D1 is less than or equal to zero, then TP1 = 0 and SP1 = 1.
- D3. If  $AD * TS \geq 13.5$  (a rare occurrence) it is supposed that TS is large enough compared to TP1 that any improvement on the latter by iteration is negligible. Step D4 is thus bypassed. If the reactive field is independent of hunger ( $IPLD \leq 0$ ), step D3 is bypassed.
- D4. This step compares the current value of pursuit success SP1 with the previous one TSP. If

their difference is greater than 2 percent, TSP is replaced with SP1 and the search and pursuit routines are repeated. Otherwise the program passes to D5.

D5. The current hunger level H1 is updated as in C2.

E. Eating

E1. The time TE to eat a prey is calculated from equation (27). The new value of HO is calculated from equation (2)

$$HO = [HK - (1/(AD*AKE))] + \{H1 - HK + [1/(AD*AKE)]\} * EXP(-AD*TE)$$

with HK replaced by  $\{HK - [1/(AD*AKE)]\}$  to account for the fact that food was being ingested as well as digested during time TE. Since it has been assumed that the whole prey has been eaten in this calculation, it is possible for HO to become negative.

E2. If  $HO < 0$ , a version of equation (8) is used to recalculate TE as the time for the hunger to go from H1 down to zero. HK is again replaced by  $\{HK - [1/(AK*AKE)]\}$  and  $HO = 0$ . The negative sign arises because in this case H1 represents the starting hunger level and  $HO = 0$  the final one. The weight eaten is then simply TE divided by the eating rate AKE, (grams per unit time), and the weight not eaten is  $WD = W1 - WE$ . HO is made zero.

E3. If  $NWST > 0$ , the predator hoards his prey until he is hungry enough to continue eating it. TE is added into an accumulator TEL. The time TD1 required for the hunger HO to rise from zero to HTE is calculated from equation (8) and added to TD. With  $W1 = WD$  instead of  $W1 = W$ , and  $H1 = HTE$  the program cycles back to E1, repeating the loop until the entire prey has been eaten. At this point HO will become non-negative and the program will branch to E4 instead of E5.

E4. If  $HO \geq 0$ , the weight eaten is the entire prey weight and the weight discarded is zero. The time taken for eating is the sum of TE calculated in E1 and any increments TEL calculated in E3.

If  $NWST \geq 0$ , the predator does not hoard the prey and discards WD uneaten.

- E5. The total cycle time  $T1 = TD + TS + TP + TE$  is calculated. A is determined from  $A = 24/T1$  and is compared with the previous value. If the two differ by more than one percent the program loops back to A1 to generate another attack cycle. Once A has reached a stable value, PR returns to the calling program.

#### 4. STARVE

This subroutine modifies the predator density if there is insufficient food. The amount of food in the predator's gut  $GUT = HK - HO$  is compared with the malnutrition threshold HDIE. If  $GUT \geq HDIE$  a return is made.

Otherwise, the time integrated food deficiency  $T*(HDIE - GUT)$  is accumulated in DEBT. If  $DEBT \geq DEBTMX$  all the predators die. When  $DEBT < DEBTMX$  only the fraction  $DEBT/DEBTMX$  of the predator population starves.

#### 5. PREP

The subroutine PREP calculates the prey progeny/adult prey according to the following relation

$$CHILD = PRATE*(TANO - XMIN)**(OPT)*EXP(XMIN - TANO) ,$$

where XMIN is the minimum prey density for reproduction

$$OPT = OPTNO - XMIN$$

OPTNO is the prey density for which maximum recruitment, MAXCH, occurs

$$PRATE = MAXCH*EXP(OPT)/(OPT**OPT).$$

If  $REPRO \leq 0$ , then the number of progeny per adult is given a constant value PRATE. Otherwise it is calculated from the above equation.

If DIAP, the proportion of progeny entering diapause in the current year, is zero, then the new prey density is calculated as

$$TANO = TANO*CHILD$$

and the subroutine returns to the calling program.

If  $DIAP > 0$ , the total number of progeny produced in one year is

$$TOTCH = TANO * CHILD$$

and the number of progeny entering diapause in the current year is

$$DICHT = DIAP * TOTCH .$$

DICHL is the number of progeny which entered diapause in the previous year. Thus the new prey density is

$$TANO = DICHL + TOTCH - DICHT .$$

#### Competition Model for Predators or Parasites

Competition affects predation in two ways: .

1. When the attacking predators or parasites compete for the prey or host they are searching for.
2. When the progeny compete for the energy provided by their attacking parents.

The first case is dealt with in the model by the subroutine ADCOM (adult competition), and the second by CHCOM (child competition).

#### I. Equations Used

##### 1. The Effects of Discrimination

To describe the elements of the first process mentioned above, we consider a parasite-host system. It has been found experimentally that even at high densities (of parasite) there is no significant effect of interference on parasite oviposition, and that the density of attacks, or of eggs laid (ANA) during time T may be expressed as a linear function of the density of attacking parasites P

$$ANA = A * T * P , \quad (29)$$



where A is the number of attacks per predator per unit time. A given host may bear more than one parasite egg--i.e. the parasite, unlike many predators, will attack a host that has already been attacked. There is then no discrimination between previously attacked and unattacked hosts.

The negative binomial was found to describe the distribution of eggs among hosts. Now the proportion of hosts in the zero category of the negative binomial distribution is given by'

$$P(0) = (g)^{-AK} ,$$

where  $g = 1 + ANA/(TANO*AK)$ . AK is the dispersion coefficient of the negative binomial. The proportion of hosts attacked at least once is simply one minus this zero category. Therefore the number of different hosts attacked per unit area (TANHA) is

$$TANHA = TANO*[1 - P(0)]$$

$$TANHA = TANO*[1 - [1 + ANA/(TANO*AK)]^{-AK}] \quad (30)$$

This model can simulate both the case of perfect discrimination and that of no discrimination. For the former, it is necessary to generate attacks in a series of steps, allowing only enough time in each step so that the resulting reduction in unattacked prey does not significantly affect the attack rate. We have chosen to use time intervals small enough that the density of attacked prey is no more than one percent of the original prey density--i.e. T should be such that  $TANHA \leq .01*TANOX$ , where TANOX represents the original prey density. If we consider for the small interval T that  $ANA = A*T*P$ , where A represents the rate of attack per unit time that is instantaneous with respect to P, then this condition means

$$.01*TANOX \geq TANO - TANO * \left( \frac{TANO*AK}{TANO*AK + A*T*P} \right)$$

$$T \leq \frac{TANO*AK}{A*P} \left[ \left( \frac{TANO}{TANO - .01*TANOX} \right)^{1/AK} - 1 \right] \quad (31)$$

This gives the amount of time to be allowed for the first increment of the attack period. Since T is short enough that only a minute portion of the attacks occurring can be multiple ones, then the number of different hosts attacked, TANHA, is simply ANA.

The next increment of the attack period is started by subtracting the density of attacked hosts TANHA from the starting host density, since with perfect discrimination these hosts cannot be attacked again. With this new value of density of hosts available for attack, new values for A, T, ANA, and TANHA are computed. The iterations continue, with each new TANHA and T added to the previously computed ones until all the time available for attack (TAG) has elapsed.

For the case where there is no discrimination there is no need to break down the attack period; the total density of attacks is given by equation (29) and the density of different hosts attacked, by equation (30).

## 2. The Effect of Competition Among Progeny

The parent parasite distributes its eggs among individual host units, but since any one host unit might contain insufficient energy to provide fully for all the parasite progeny on or within it, the progeny may have to compete for this energy. There may be either contest competition, in which one individual or a number of individuals sequester that part of the available resources that they require and the remainder die, or scramble competition, in which all die if there are not enough resources for all.

The progeny submodel used in this model is based on the following assumption:

1. The universe within which competition occurs is divided into discrete blocks.
2. The number of attacks per block (eggs per host) is approximated by a negative binomial distribution.
3. The capacity of each block is constant regardless of egg density.
4. The progeny survival rate SPR is constant.

Then the density of parasite adults (PROG) emerging is

$$\text{PROG} = \text{SPR} \cdot \text{TANO} \cdot \left[ \sum_{X=1}^B X \cdot P(X) + (B - \text{ALPH}) \cdot \sum_{X=B+1}^{\infty} P(X) \right], \quad (32)$$

where B is the capacity of each host in yielding adults, P(X) is the probability of a host having X parasite eggs, as predicted by a negative binomial with mean ANA/TANO,

$$P(X) = P(X - 1) * (AK + X - 1) * ANA / [X * (TANO * AK + ANA)] , \quad (33)$$

$$P(0) = 1 + [ANA / TANO * AK] ** (-AK) , \quad (34)$$

and

ALPH is the degree of scramble  $0 \leq ALPH \leq B$  .

This makes it possible to represent different kinds of competition, from pure contest to pure scramble.

If there is perfect discrimination (i.e. one egg per host) this can be reduced to

$$PROG = SPR * TANHA . \quad (35)$$

## II. Subroutines Used

### 1. ADCOM

The predator-prey model which includes the effects of competition uses all the subroutines described under "Functional Response", the only difference being that there is an additional subroutine ADCOM, which fits in at the head of the chain. It calls FR, which in turn activates the other subroutines as previously described, in order to obtain values for the instantaneous rate of attack A corresponding to different prey densities. DISC is a variable which enables the user to choose between an attack system with no discrimination and one with perfect discrimination.

A. The hunger thresholds are given initial values which depend on whether or not there is predator learning. The starting prey density is stored and FR is called. It returns a value of A.

B. If DISC > 0 we are dealing with perfect discrimination.

B1. As previously explained, we generate the attacks in a series of steps, updating the prey density at the end of each attack and accumulating TANHA and the elapsed time. Before the attack step iterations are begun the prey density is initialized and the time and attack accumulators are zeroed.

- B2. The interval of time  $T$  allowed for the first attack step is calculated from equation (31).  $SUMTR$  represents the total time elapsed up to the start of the current attack step, and  $SUMT$  the total time elapsed up to the end of the current step.
- B3. If  $SUMT \geq TAG$ , then the  $T$  calculated in B2 is greater than the time actually left ( $TAG$  is the total time available for attack). So  $T$  is recalculated as the difference between  $TAG$  and  $SUMTR$ .
- B4. The density of attacks  $ANA$  occurring in time  $T$  is calculated from equation (29), and in case any multiple attacks occur, the density of different hosts or prey attacked ( $BUC$ ) is determined from equation (30).
- B5. If  $BUC$  as calculated in B4 is not greater than zero, it is given the value  $ANA$ . If the value of  $TANHA + BUC$ , the density of attacks up to the latest attack cycle, exceeds the prey density then  $ANA$  is made equal to  $TANHA$ .
- B6. The value of  $BUC$  for this step is added into  $TANHA$  which accumulates the total density of prey attacked for the whole attack time  $TAG$ . The density of attacked hosts is subtracted from the starting host density.
- B7. If there is predator learning, the hunger thresholds are recalculated. If  $SUMT < TAG$ , all the available time has not yet been used up. Subroutine  $STARVE$  is called to take into account any possible predator starvation.  $FR$  is again called to supply a value of  $A$  for the new prey density.
- B8. As long as the prey density is greater than 1% of the original density, the program goes back to B2 to generate another attack step. If not, it is assumed that after one more attack step there will be effectively no more prey left. So the program goes to B3, and sets  $T$  equal to whatever is remaining of the available attack time  $TAG$ .
- B9. When  $SUMT = TAG$ , the available time has been used up.
- B10. The output values, various functions of the predator, prey, and attack densities are calculated.

C.  $DISC < 0$  means attack with no discrimination. The attack density is calculated in one step.

C1. The total density of attacks ANA occurring in time TAG is calculated from equation (30) and the density of hosts TANHA attacked from equation (31). If TANHA should be less than zero, it is given the value ANA.

## 2. CHCOM

To help reconcile the FORTRAN equations with those previously described, the following list relates the two sets of variables:

CAP ----- B  
ALPH ----- ALPE  
FX ----- P(X)  
FO ----- P(0)

A. The capacity of hosts in yielding adults, and also the degree of scramble are calculated as functions of the weight of host eaten.

B. If  $DISC = 1$ , then each attacked host contains only one parasite, and the density of progeny is given by equation (35).

C. If  $DISC < 0$ , the more general equation must be used. The various accumulators are first initialized. FO is calculated as the proportion of unattacked prey to original prey, and its value is stored.

D. If  $FO \geq 1$ , no prey are attacked and  $FX = 0$ .

E.  $FO < 1$ , FX is calculated from equation (33) with  $P(X - 1)$  replaced by FXO.

F. The sums of FX's and  $X*FX$ 's for X ranging from 1 to CAP are determined. Each time the value of  $P(X - 1)$  appearing in equation (30) for FX is replaced by the value of FX from the previous iteration.

G. The terms computed in F are assembled to calculate the progeny density as given by (32).

Dictionary of Variable Names Used in Predation Model

AD	rate of food digestion (gm/hr)
AK	dispersion coefficient
AM	constant relating success to duration of pursuit (input)
AKE	rate of eating (hr/gm)
A	overall attack rate (number of attacks/predator/day)
AREA	area of reaction of predator (sq.m.)
AREA1	= (1/PI)*(area of a circle with radius equal to predator's reactive distance at start of search) (sq.m.)
AREA2	(1/PI)*(area of a circle whose radius is the reactive distance at end of search) (sq.mi.)
CHILD	number of offspring per parent prey
CAPT	ratio of HTC to HTP
CAPA	maximum host carrying capacity
CAP	actual capacity of each host in yielding adults
DS	strike distance (m)
DIAP	proportion of progeny entering diapause this year
DICHL	density of progeny which entered diapause the previous year
DICHT	density of progeny which will enter diapause this year (no./sq.m.)
EAT	ratio of HTE to HTP
EYE	distance between predator's eyes (m.)
EPSIL	convergence interval for attack rate A
EZERO	term containing $t = 0$ in the integral equation for TS
FOOD	= -1* (amount of food which would still be undigested in predator's gut after time T)
FRNHA	number of attacks per predator

FRNA number of attacks per predator

F2X  $(1/TS2) * (\text{area of constant reactive distance, swept out by predator in time } TS2)$

G area containing a catchable prey (sq.m.)

HO hunger level at beginning of search cycle

H1 updated hunger level during rest of attack cycle

IFLD indicates whether or not reactive distance is a function of hunger. REACT varies with hunger if  $IFLD > 0$

HK maximum gut capacity of predator (gm)

HONP constant of proportionality, determined experimentally, used in calculation of predator's distance of reaction

HOPT value of HTP at minimum of HTP vs. L curve, i.e. value for which  $L = LOPT$

KA constant used in calculating area of reaction of a predator; the value of KA depends on the shape of this area.

L prey length (m)

LOPT prey length for which HTP has a minimum (m)

MAXCH maximum curve of progeny/adult vs. adult density

MYOP a constant (input) related to the geometry of the predator's vision; used in the equation for HTP

NWST indicates whether predator hoards his prey or not

OPTNO prey density for which maximum recruitment (MAXCH) occurs

P predator density (no./sq.m.)

PRNHA percent prey attacked

QA hunger-independent factor in expression for predator reactive distance

REACT predator reactive distance (m.)

REPRO an indicator for type of relation between progeny and adult prey. If  $REPRO = 1.$ , the progeny produced per adult prey is a constant

SPR	progeny survival rate
SR	recognition success of predator for prey
SCRA	degree of scramble competition among predator progeny
SS	predator strike success
TAG	total time allowed for attacks (days)
TANO	total prey density (no./sq.m.)
TANHA	total density of attacks (no./sq.m.)
TOTCH	total number of prey progeny produced in one year
TS	search time (hr.)
TSP	overall proportion of successful pursuits
TS1	that part of TS during which the predator's hunger is a maximum, and his reactive distance remains constant (hr.)
TP	pursuit time (hr.)
TE	eating time (hr.)
TD	digestion time (hr.)
TX	time required for predator hunger to increase to 99% of its maximum possible value HK (hr.)
VP	speed of predator (m./hr.)
VR	relative speed of predator to prey
W	weight of prey (gm.)



Instructions for Using Predation Model

1. The only MTS instruction needed to run the predation model is

RUN CRAWLING .

CRAWLING is a file containing the FORTRAN G--compiled routines of the predation model, link-edited with the libraries RHIL:LIBR and \*LIBRARY. This command will cause the program to begin execution, after which all manipulations and data entries required will either be guided by specific instructions from the program or be initiated independently by the user with the aid of the command language.

2. The user will be asked to choose from five program options:

1 ATTACK RATE/PREDATOR: A = F(ANO)  
2 DENSITY OF ATTACKS: ANHA = F(ANO)  
3 DENSITY OF ATTACKS: ANHA = F(P)  
4 PREDATOR PROGENY: PROG = F(P)  
5 PHASE SPACE .

He will then be asked to specify whatever parameter values are needed and whether or not there is to be predator learning.

3. Program options 1-4 generate functional relationships which will be plotted on SYN6. Once such a curve has been displayed the user has the choice of terminating the program, restarting with another program option, or repeating the same option with the choice of continuing the same curve over a smaller range of independent variable, adding another curve (i.e. generated by changing some parameter) to the same graph, or beginning a new graph. In the case of generating a family of curves on the same graph it is important that the curve with the highest y-values be generated first because the graph will be scaled according to these values.
4. Whenever the program is told to start a new graph or new curve, it automatically goes into command mode before continuing, thus enabling the user to specify any parameter changes. Command mode may also be entered during any iter-

ation of options 2-5, simply by giving an attention interrupt. In this case, upon exiting from command mode the user will be asked if he wants to continue the current simulation or not. Thus he can stop a particular simulation during its execution and go on to something else.

5. With the option to explore phase space, the user will be asked to supply bounding values for the space and to indicate whether or not the exploration plot is to be stored in a file as well as displayed on SYN6.

The user is then asked to specify a starting point in phase space, and a trajectory is generated over the number of generations given by the user. At this point the user must choose between generating another trajectory, restarting with a different program option, storing the boundary of the stable region, or stopping the program entirely.

Command mode is automatically entered each time a new trajectory is started, and may be entered at any time during a trajectory by giving an attention interrupt. In the latter case the user will be asked if he wishes to discontinue that particular trajectory.

6. Upon termination of execution of the predation program, any graphic output which the user requested to have stored will all be contained in a scratch file -PLOT. Before signing off, it is necessary to transfer this information to a permanent file(s).

If there is only one graph stored in -PLOT then it can be simply copied to a permanent file (preferably with a name coded to identify the graph). If there are several graphs stored in -PLOT and you wish to have each stored in a separate file, the program WEBB:EDPLOT will separate out the graphs in the scratch file and copy each to a file which will be created with a filename specified by the user.

```
RUN WEBB:EDPLOT
INPUT -PLOT
OUTPUT filename integer
```

where "integer" specifies which graph (i.e. 1st, 2nd, etc.) in the file -PLOT is to be copied into the file called filename.

If a range of integers is given instead of a single integer, that range of graphs will be stored together in the file filename. The OUTPUT command may be repeated as many times as necessary.

7. For each graph stored in -PLOT a list of parameter values identifying the graph is written into the file -PLTPAR. At the end of each run, these parameter lists should be copied into short permanent files whose names should contain the code numbers used to identify the corresponding graphs.

8. To replot on SYN6 a graph stored in a file:

```
RUN WEBB:EDPLOT
INPUT plotfile
PLOT n
```

where n refers to the nth graph in the file 'plotfile'.

To replot a graph on the /360 CALCOMP

```
$RUN *PLOTQ PAR=plotfile
papertype plottersize
```

where papertype is either BLANK (or B) or LINED (or L) and plottersize is either SMALL (or S) or LARGE (or L).

#### Command Mode

Whenever the prefix < appears at the start of a line on the terminal, the program is in command mode and is ready to receive commands from the user. The basic set of these commands is as follows:

DISPLAY VNAME will cause the current value of the variable with the FORTRAN name VNAME to be displayed.

SET VNAME = value causes the variable VNAME to be assigned the numerical value entered. There must be a space before and after the equal sign.

EDIT causes control to be passed to the file editor \*ed. The user will be asked to enter the name of the file to be edited only at the time of the first call to the editor. To leave the editor and re-enter command mode, enter STOP.

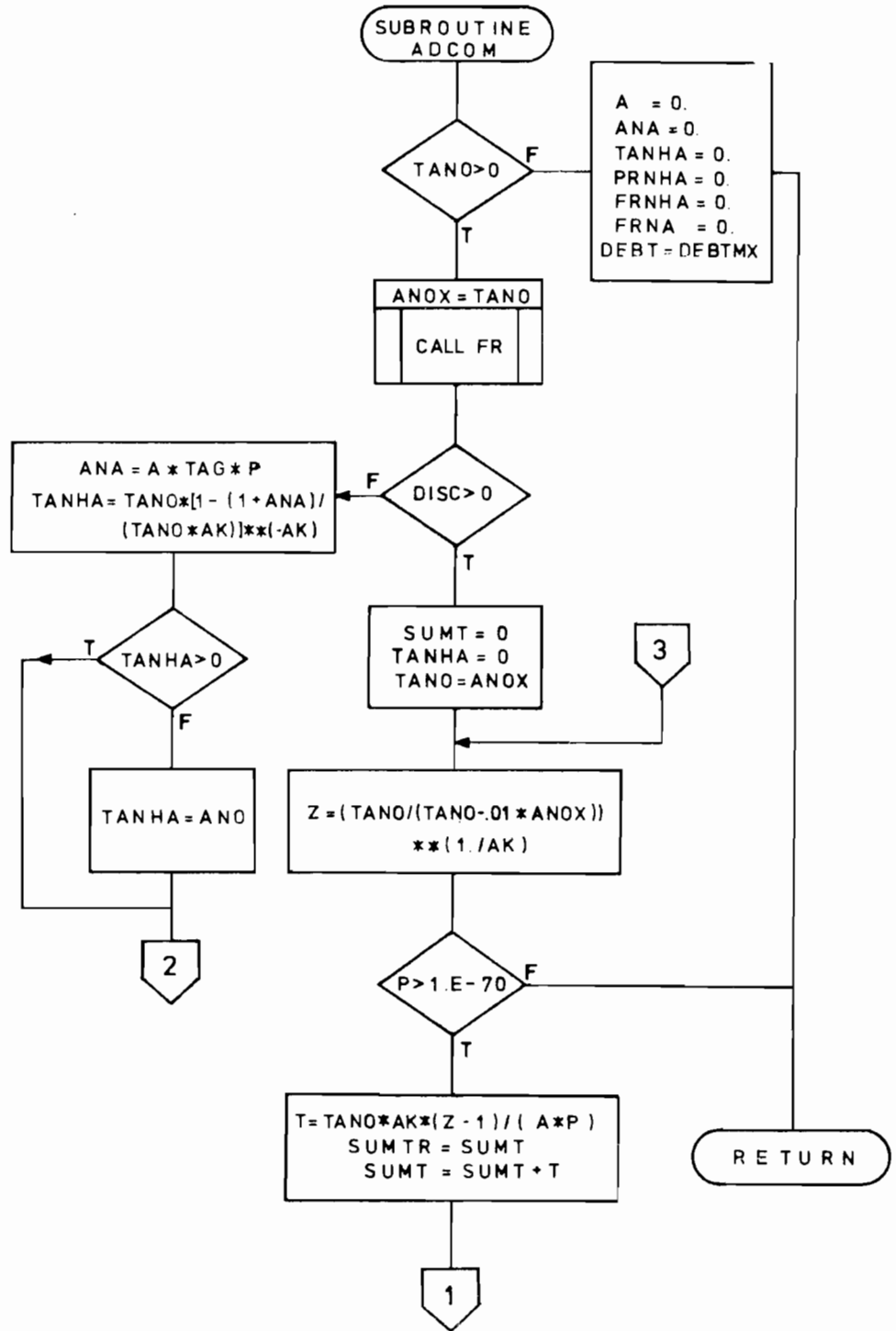
MTS gives control to the MTS command level. The user may issue any MTS commands which do not cause loading of another program (e.g. LIST, CREATE, COPY are permissible but RUN is not). Entering RESTART will generate a return to command mode.

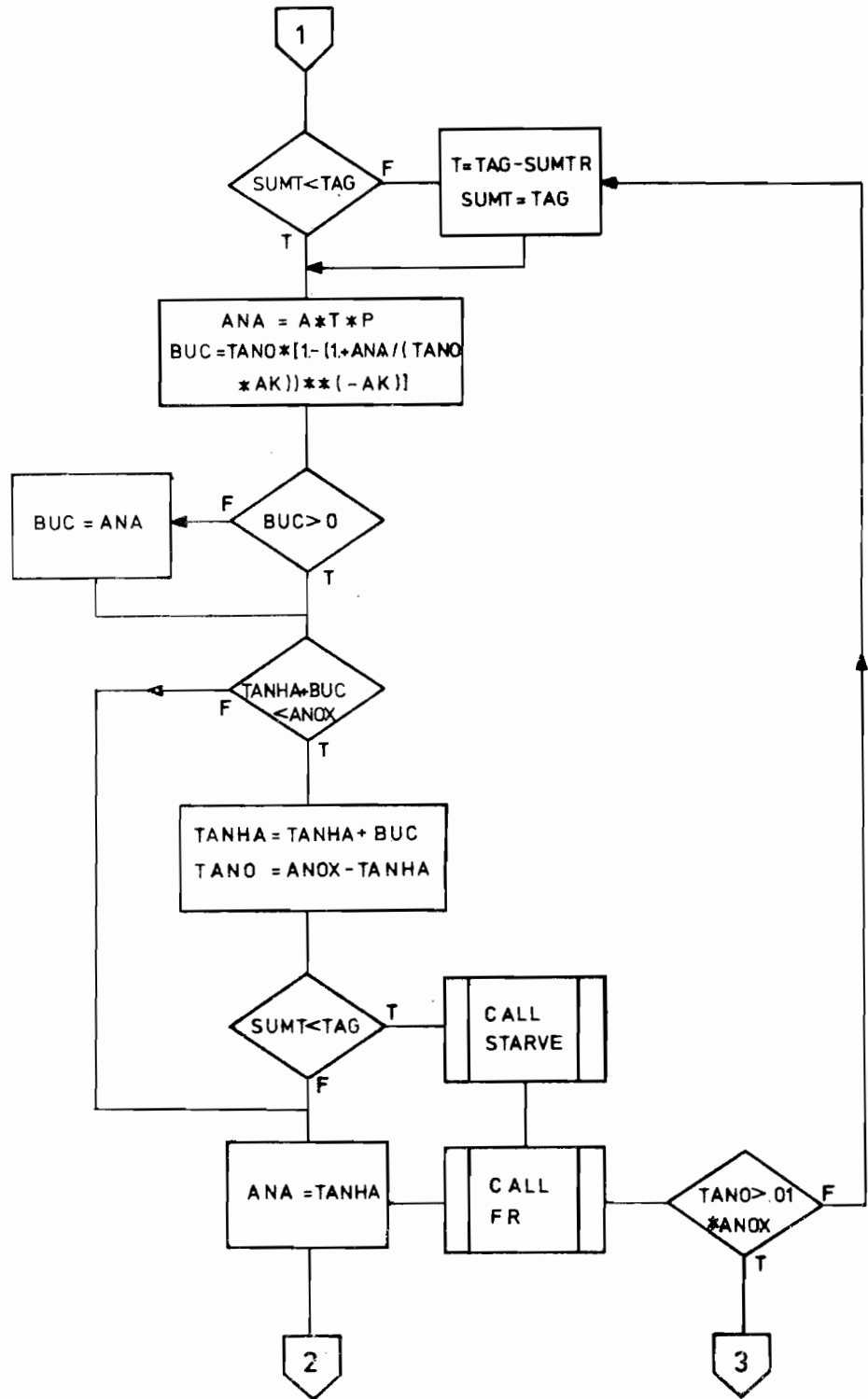
STOP causes execution of the program to terminate.

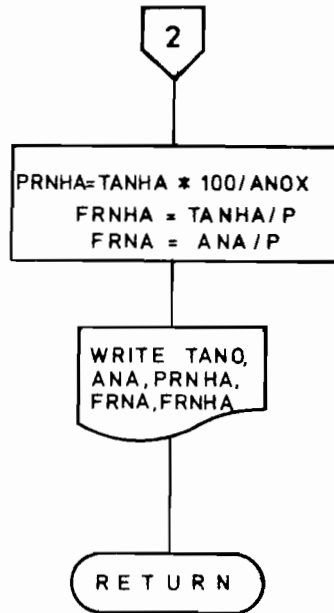
CONTINUE causes an exit from command mode and restarts execution of the simulation.

Any variable in the common block may be referenced by DISPLAY and SET. If the variable is subscripted (up to four subscripts are allowed) it must be referenced in one of the following three ways:

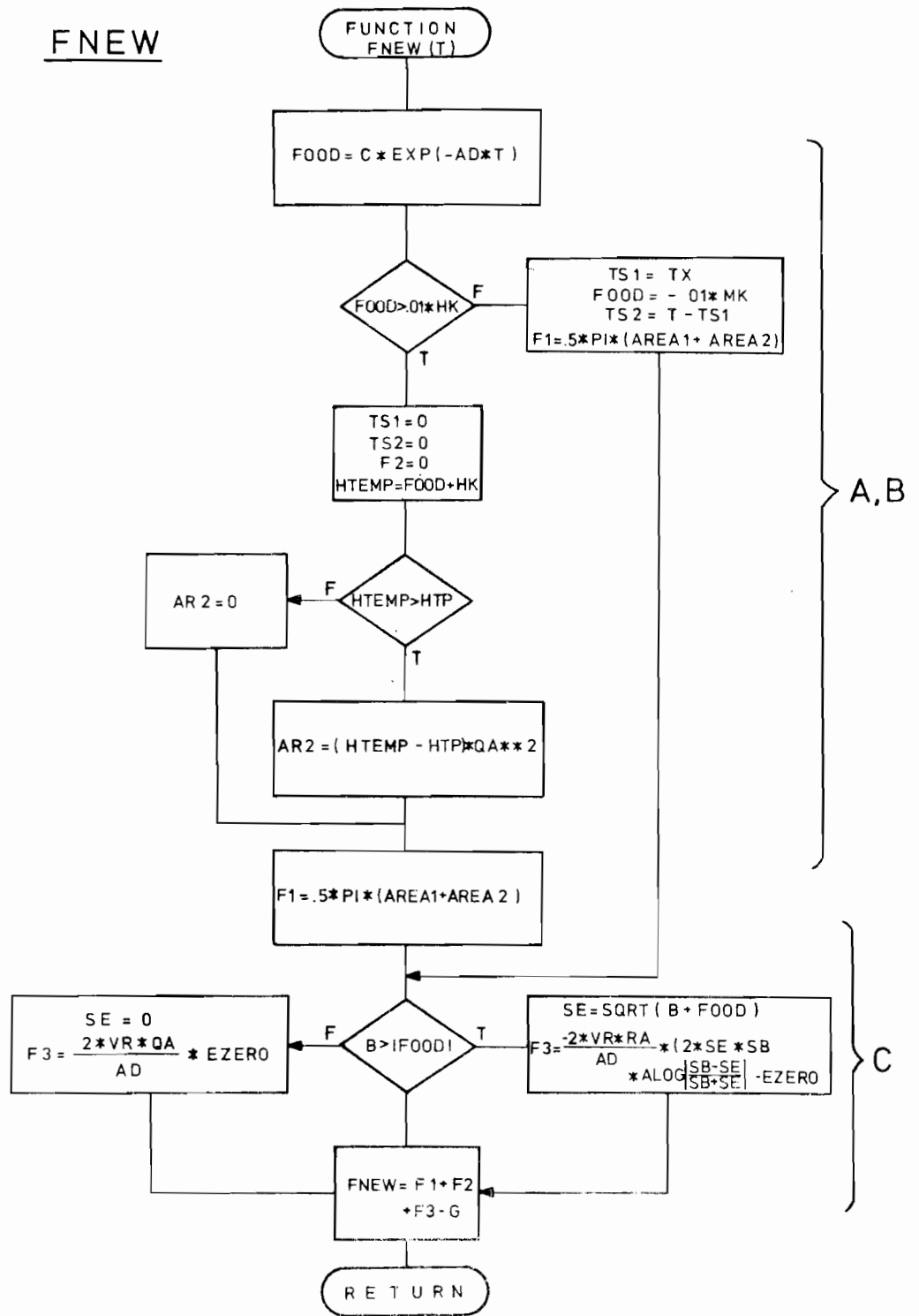
1. A(I) references the Ith element of the array A.
2. A(ALL) references all values for which A is dimensioned.
3. A(I...J) references A(I) through A(J).





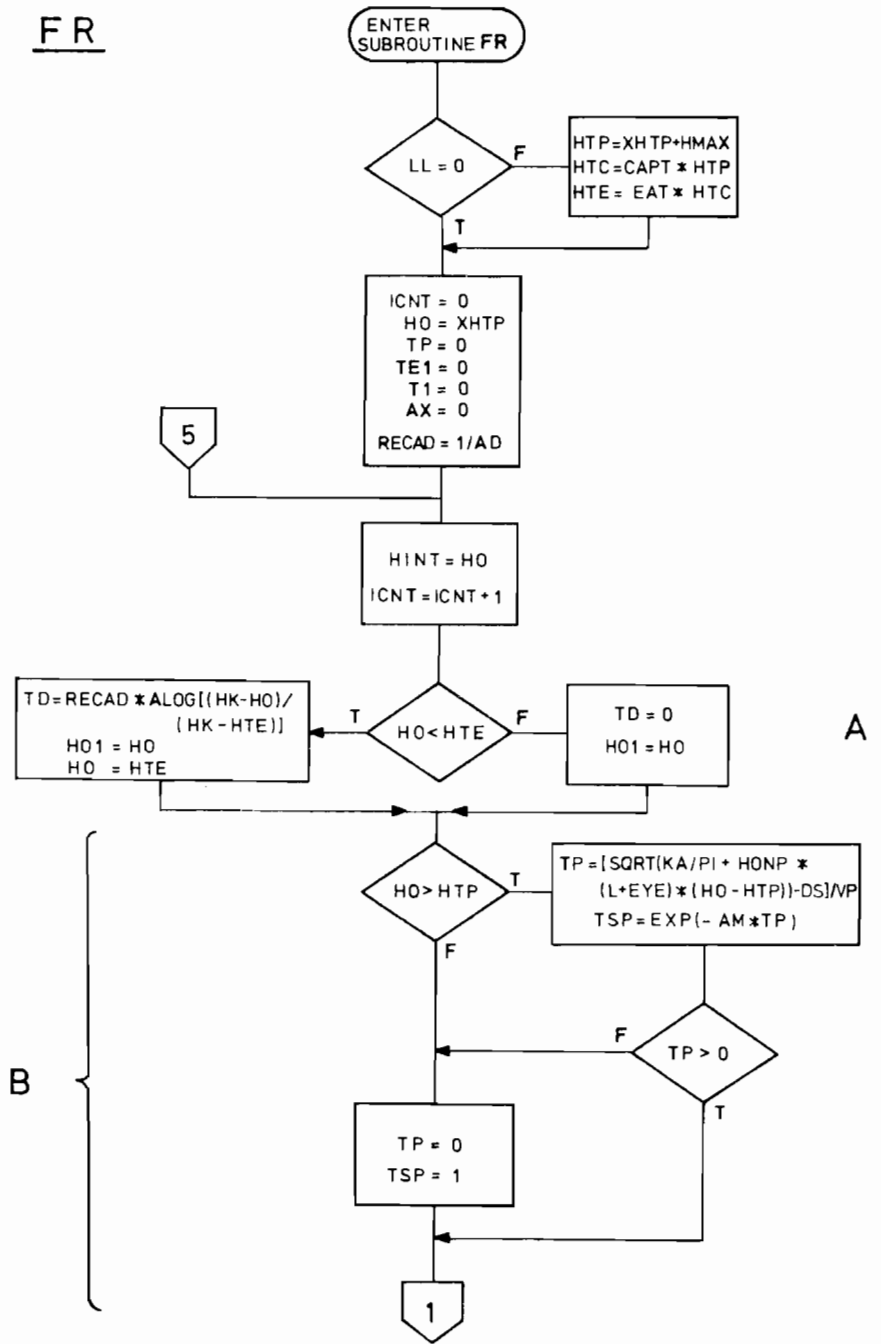


FNEW



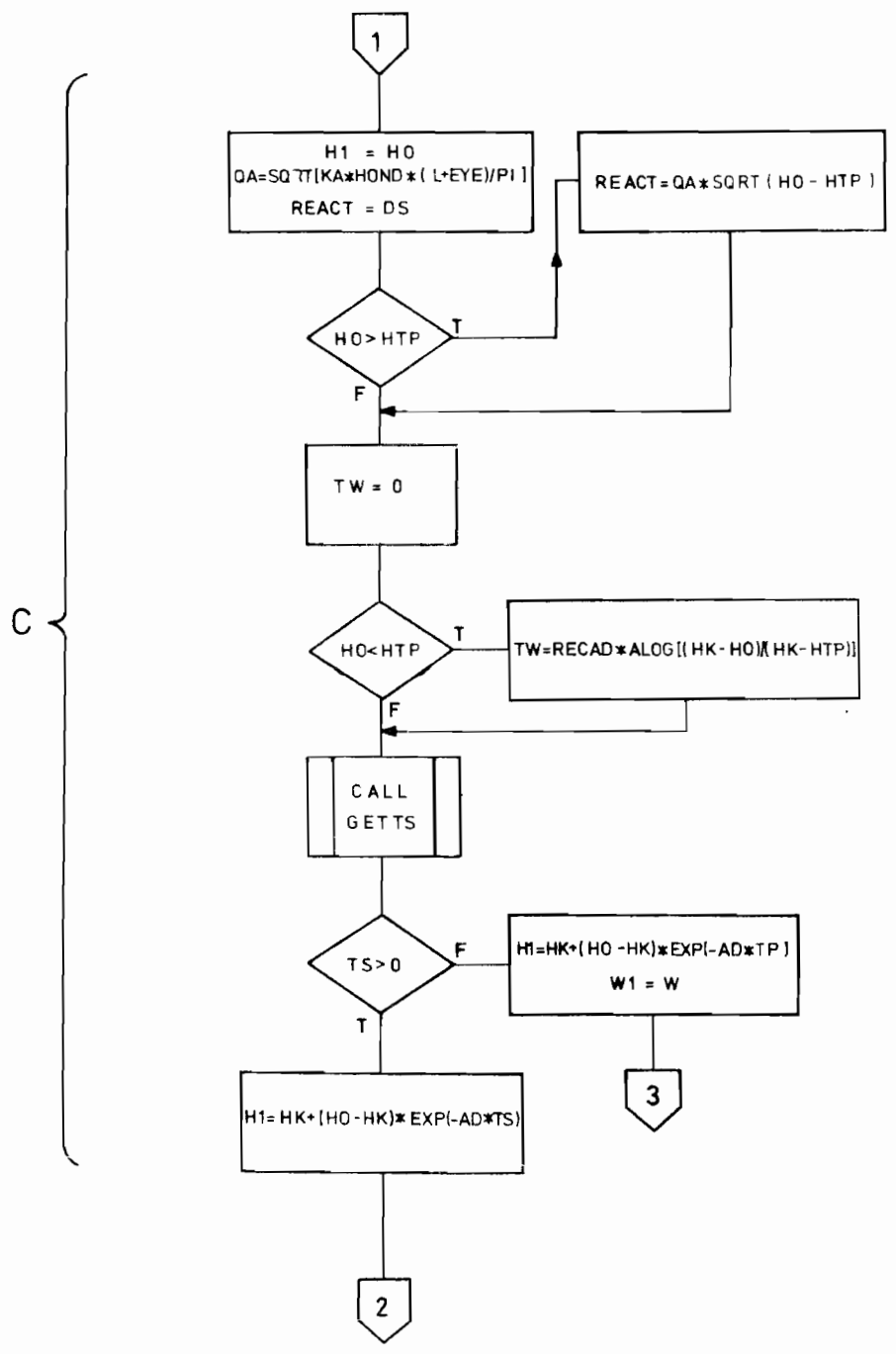


FR

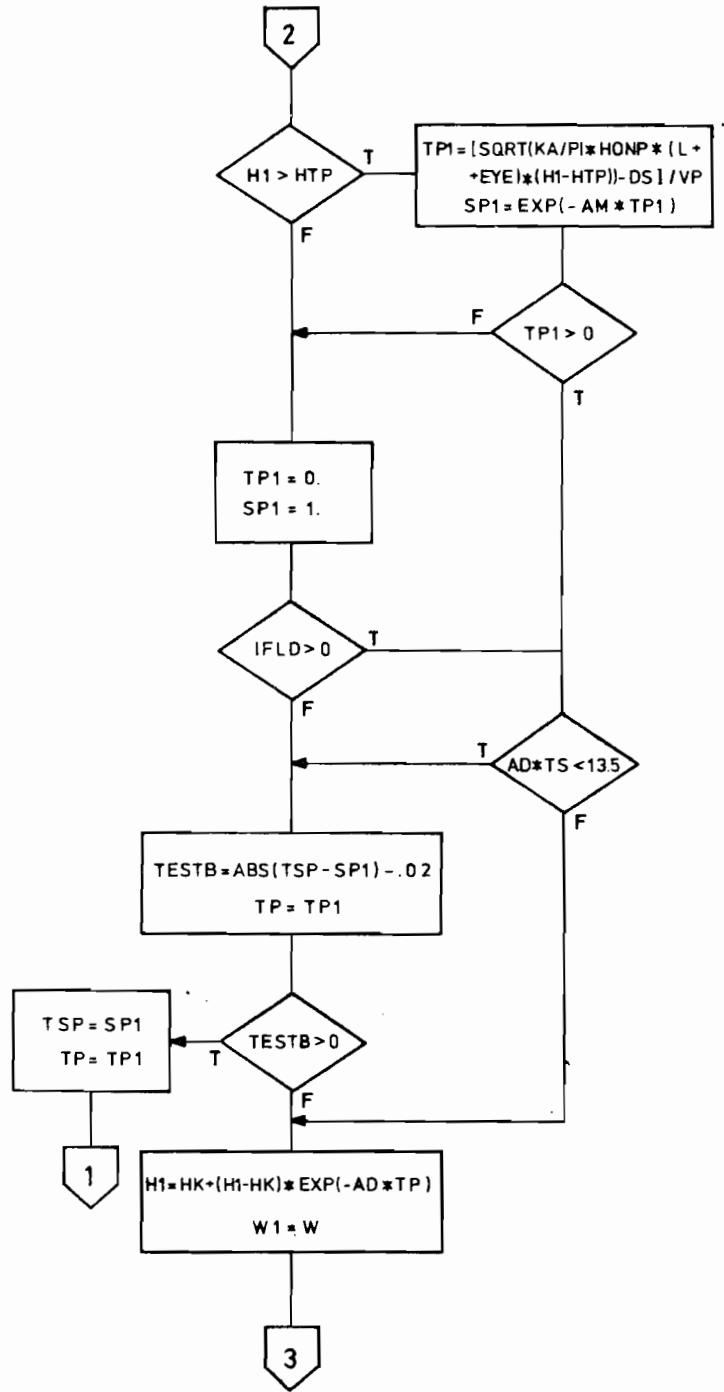


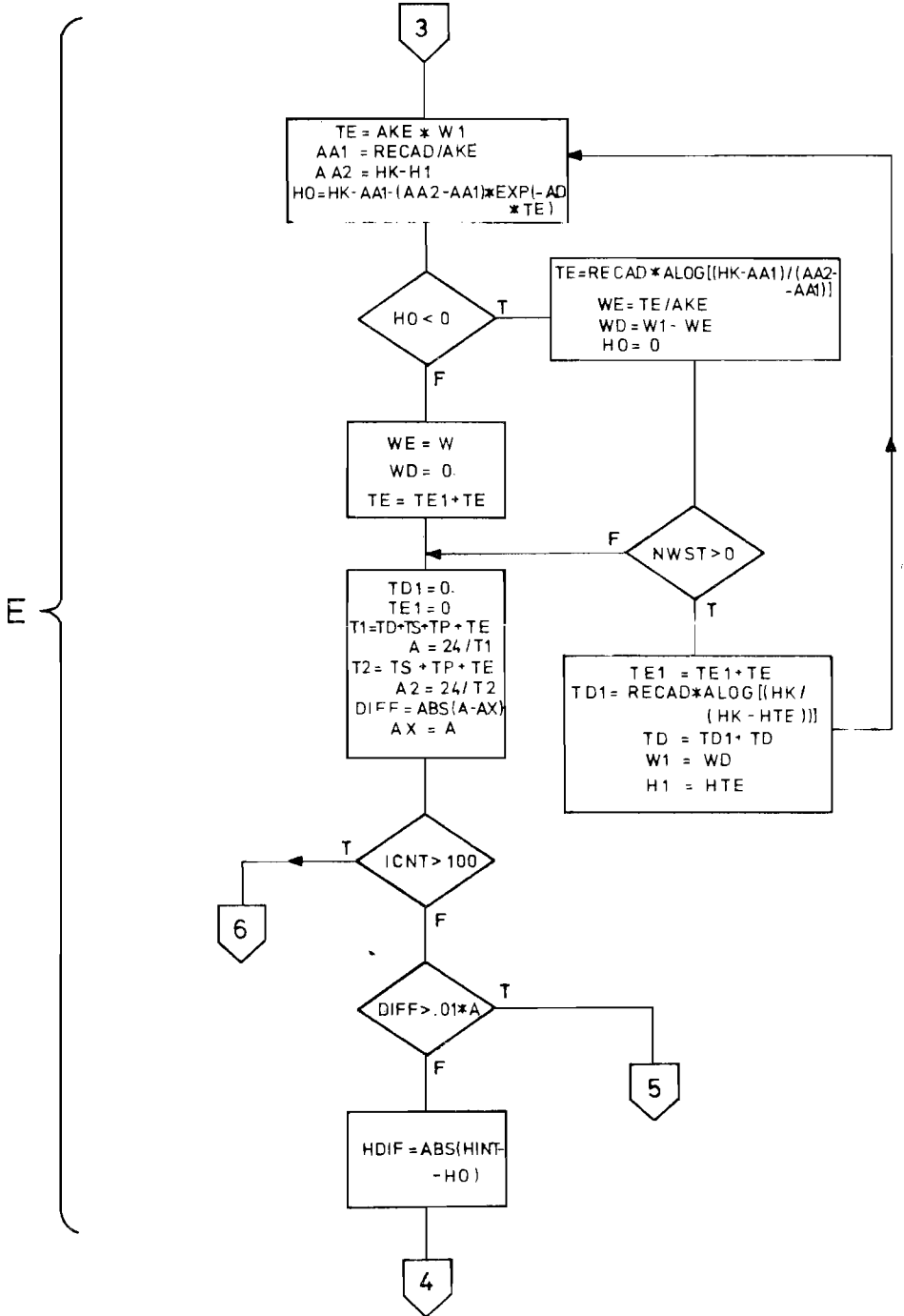
B

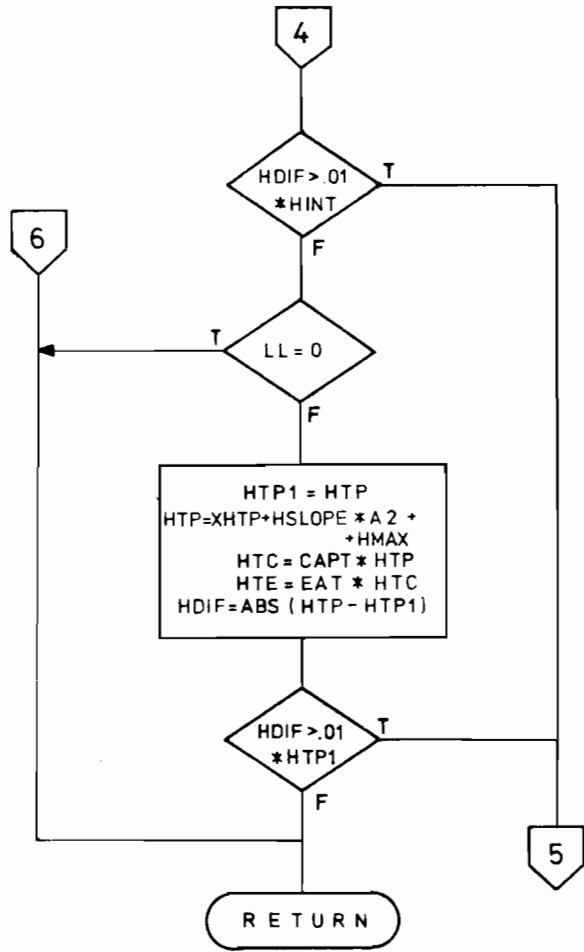
A



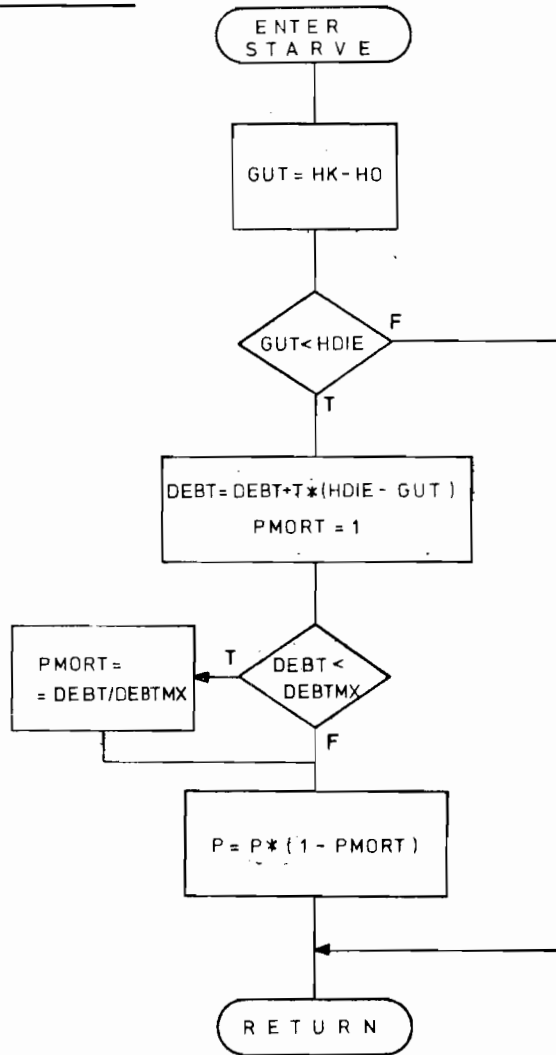
D {



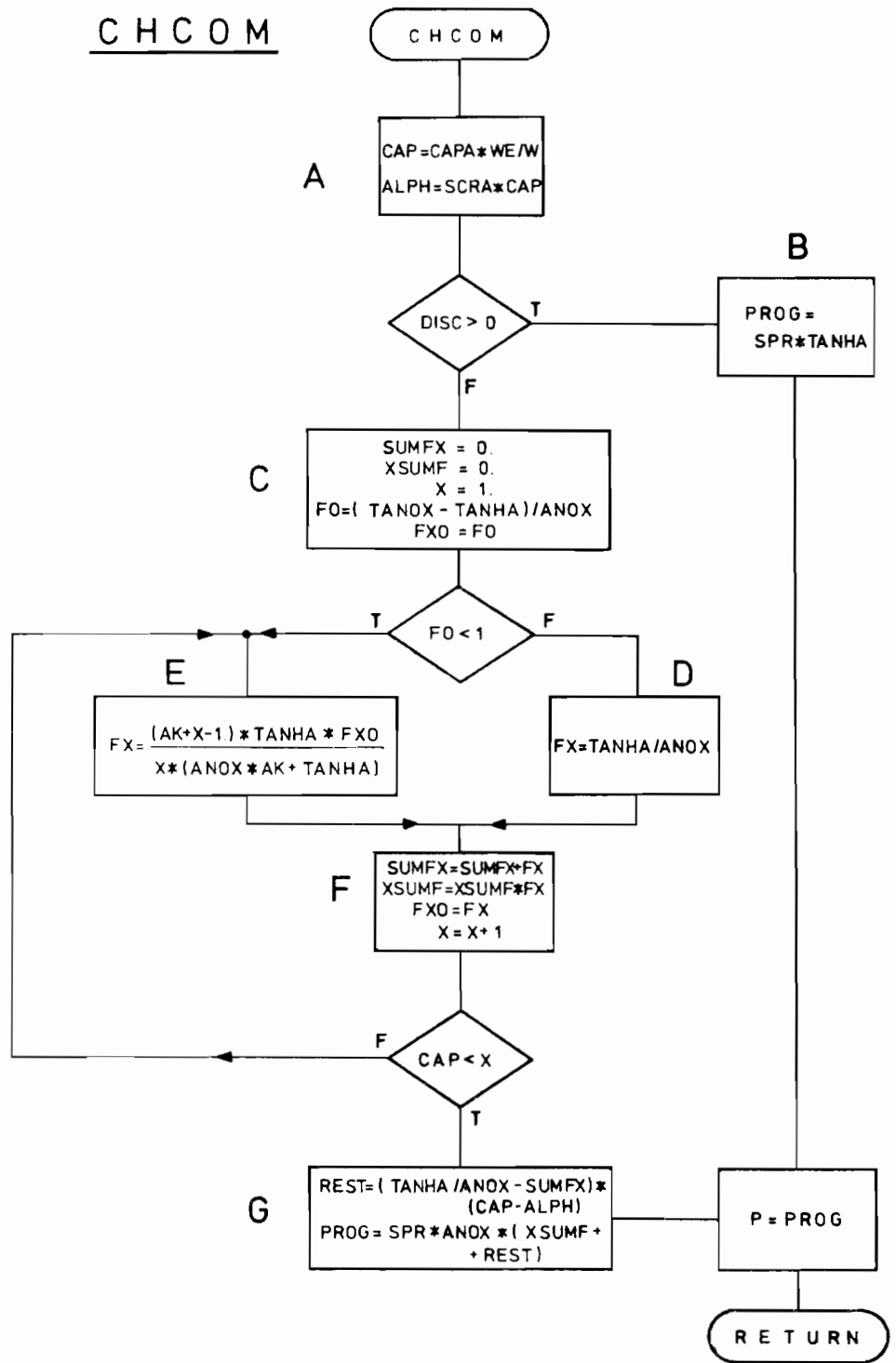




STARVE



CHCOM



PREP

