Special Restrictions

None.

Related SNAs

None.

FUNCTION

A FUNCTION Command defines the rules for a table lookup.

There are several types of Function Entities. Each has its own rules pertaining to the table lookup. For each, the lookup table is specified in one or more Function Follower Statements. Type C Functions are a special case. They use a table lookup, followed by a linear interpolation.

The use of Function Commands to define probability distributions has been largely supplanted by the built-in distributions in the Procedure Library. This is discussed in Chapter 8. The old Function Types are still supported by GPSS World.

NAME FUNCTION A,B

Label / Operands

NAME - Entity Label this entity. Required. The field must be Name.

A - Function argument. Required. The operand must be Name, PosInteger, String, ParenthesizedExpression, SNA, or SNA*Parameter.

B - Function type (one letter) followed immediately by the number of data pairs in the Function Follower Statements. Required.

Action

The FUNCTION Command is a Queued Command. When the Simulation Object receives one, it places it at the end of the simulation's Command Queue.

A FUNCTION Command together with one or more Function Follower Statements defines a GPSS Function Entity. Later references to an SNA of the class FN can evaluate the Function and return the result. Operand A of the FUNCTION Command is evaluated numerically. There are several types of Functions which must be considered individually. The type is specified in Operand B of the FUNCTION Command.

Each FUNCTION Command must be followed immediately by a list of data pairs, separated by slashes, which define a table. The text lines that contain the list are called

Function Follower Statements. Each data pair has an X value and a Y value (or SNA) separated by a comma. Function Follower Statements create tables in the simulation which allow a mathematical function of one variable to be evaluated. When the Simulation Object encounters a reference to an FNEntnum SNA, it evaluates the Function Entity and returns the result. The tables and the manner in which they are referenced depends on the Function type.

Function Modifiers

FN class SNAs used in Operand B of GENERATE or ADVANCE Statements are called Function Modifiers. When a Transaction enters a GENERATE or ADVANCE Block with a Function Modifier, the result of the Function is multiplied by Operand A of the Block and used as the time increment.

Operand C of an ASSIGN Statements is also called a Function Modifier, although it is specified differently. In this case, only the entity specifier, not the FN class SNA, is used in the ASSIGN Statement. When a Transaction enters an ASSIGN Block with a Function Modifier, Operand C is used to determine the Function Entity number. Then the result of evaluating that Function is multiplied by Operand B of the ASSIGN Block and the result is used as the ASSIGN value.

Function Types

There are 5 different type of Function Entities:

Type C Functions

Type C - "Continuous" valued Function. Given an X value, after a linear interpolation, the Function returns a Y value. A random argument is a special case.

In Function Follower Statements of C type Functions, the X and Y Values must be Integer, or Real.

In a C type Function, without a random argument, the data pairs in the Function Follower Statements define a piecewise linear Function of the argument. The first data pair defines the left end point and the last data pair defines the rightmost end point. The X and Y values are stored as double precision floating point numbers.

The Function evaluation begins with the evaluation of the argument. The result is used to identify the line segment of the Function. The argument is then used in a double precision linear interpolation to arrive at the double precision result of the Function. If the argument falls outside the end points of the Function definition, the value at the nearest end point is returned.

When Operand A of the FUNCTION Command is an RN class SNA, the Function is said to have a random argument. A type C Function with a random argument is used to define a "continuous" probability distribution. This is a special case. The Function is specified as the cumulative distribution function (CDF) with 0 as the value of the left end point and 1 the value of the right end point. As before, the CDF is specified as a piece wise linear function. A random number between 0 and .999999, inclusively, is taken from the random number stream and is used in a linear interpolation to arrive at the double precision value of the Function.

Example

```
Output FUNCTION V$Input,C3
1.1,10.1/20.5,98.7/33.3,889.2
```

This example defines a piece wise linear function with two line segments. When the

Function Entity is evaluated for a FN$Output SNA, first the Function argument V$Input is evaluated. If the argument result is outside the defined range, 1.1 to 33.3, the nearest endpoint is returned. For example, if V$Input returns 1 or less, FN$Output returns 10.1.

If the argument result falls within a defined line segment, a linear interpolation is performed. For example, if V$Input returns 25, then FN$Output returns the result of the following calculation:

98.7 + (889.2-98.7) # (25-20.5)/(33.3-20.5)

Therefore, FN$Output returns 376.6101563.

Example

```
Xpdis FUNCTION RN200,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/.75,1.38
.8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2
.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
```

In this example we define the Function Entity named Xpdis. The FUNCTION Command names the Function, used random number generator 200, and tells GPSS World that it is to be a type C Function Entity with 24 data pairs to follow in one or more Function Follower Statements.

This is an example of an approximation to a negative exponential distribution with mean of 1. You can use a built in probability distribution from the Procedure Library, which is slightly more accurate, but Function Entities are generally more efficient.

Type D Functions

Type D - Discrete valued function. Each argument value or probability mass is assigned an numeric value. A random argument is a special case.

In Function Follower Statements of D type Functions, the X Values must be Expression, and the Y Values must be Integer, Real, or Name.

In a D type Function, without a random argument, the data pairs in the Function Follower Statements define a set of argument values which are associated with specific Function values. The X values in the Function Follower Statements must be non decreasing. They are stored internally in double precision. When the Function is evaluated, the X values from the Function Follower Statements are searched from lowest to highest value. When an X value is found which is greater than or equal to the argument value, the corresponding Y value is returned. If there is no such X value, the Y value or named value associated with the largest X value is returned.

Example

```
Dlis1 FUNCTION X$A2,D5
1.1,6.9/2.1,7/6.33,9.4/7,10/9.9,12.01
```

A D type Function with a random argument is used to define a discrete probability distribution. The Function is specified as a cumulative distribution function (CDF) with 0 as the value of the left end point and 1 the value of the right end point. When such a Function is evaluated, a random number between 0 and .999999 is taken from the random number stream and is used. The smallest X value in the Function Follower Statement that is greater than or equal to the random number is selected. The associated Y value is returned as the value of the Function.

Example

```
Ran1 FUNCTION RN1,D5
0,0/.2,7.2/.4,6.667/.8,9.92/1.0,10
```

Type E Functions

Type E - Discrete, "attribute valued" function. Each argument value or probability mass is assigned an SNA to be evaluated. A random argument is a special case.

In Function Follower Statements of E type Functions, the X Values must be Expression, and the Y Values must be Integer, Real, Name, SNA, or ParenthesizedExpression.

A type E Function is evaluated in the same way as a type D Function, except that a type E Function requires one more step. After the appropriate X value is chosen, the associated SNA (Y value) is evaluated and returned as the result of the Function.

Example

```
Edisc FUNCTION X$QRA,E5
1,S$Stor1/3,S$Stor2/5,S$Stor3/9,S$Stor5/10,S$Stor6
```

Type L Functions

Type L - List valued function. The argument value is used to determine the list position of the value to be returned.

In Function Follower Statements of L type Functions, the X Values must be Integer, and the Y Values must be Integer, Real, or Name.

The Function Follower Statements define a list of values from which the result is chosen. When the Function is evaluated, the argument is evaluated and used as the ordinal number (X value) of the list member. The Y value with that position within the list is returned as the result. If the argument is too large or less than 1, an Error Stop occurs. X values must begin with 1 and be incremented by 1 for each successive data pair. X values may not be omitted in Function Follower Statements.

Example

```
Listtype FUNCTION Q$Barber,L5
1,PAR1/2,PAR2/3,PAR3/4,PAR4/5,PAR5
```

Type M Functions

Type M - Attribute list valued function. The argument value is used to determine the list position of the SNA to be evaluated and returned as the result.

In Function Follower Statements of M type Functions, the X Values must be Expression, and the Y Values must be Integer, Real, Name, SNA, or ParenthesizedExpression.

A type M Function is evaluated in the same way as a type L Function, except that a type M Function requires one more step. After the list position is chosen, the associated SNA is evaluated and returned as the Y value.

Example

```
Mlist FUNCTION X$Name1,M5
1,Q$Nnam1/2,Q$NamX/3,Q$Nam4/4,Q$Nam6/5,F$Tan1
```

Rules For Functions

You must obey several rules when you create a Function. They apply to both FUNCTION

Commands and/or Function Follower Statements.

The X values of Function Follower Statements must be non decreasing.

Function Follower Statements are NEVER line numbered.

A Function which has a random argument must describe a valid cumulative probability distribution in the Function Follower Statements.

All fields in a Function Command are required.

All X values and Y values in Function Follower Statements are required.

The number of data pairs stated in the B Operand of the FUNCTION Commands must correspond to pairs separated by slashes, [/], in the Function Follower Statements.

· X1, the first probability value specifying a random continuous Function must be 0. CDF values must be nonnegative, nondecreasing, and may not exceed 1.

Function Follower Statements have no comments field.

In a Function Follower Statement, an X value is followed by [,], [,] is followed by a Y value, a Y value is followed by [/] or [CR], and [/] is followed by an X value.

CDFs must be nondecreasing between 0 and 1, inclusively. Any missing probability in C type Functions is given to the rightmost interval, otherwise missing probability is an error.

C,D, and L type Functions cannot have SNAs for Y values.

E and M type Functions must have SNAs, or parenthesized Expressions, for Y values.

L and M type Functions may not have random arguments.

The Function Follower Statements of L and M type Functions must have sequential X values starting with 1.

The special rules applied when running in GPSS/PC Compatibility Mode are discussed in Chapter 3.