MICROSOFT®

Microsoft® Pascal Compiler

Quick Reference Guide

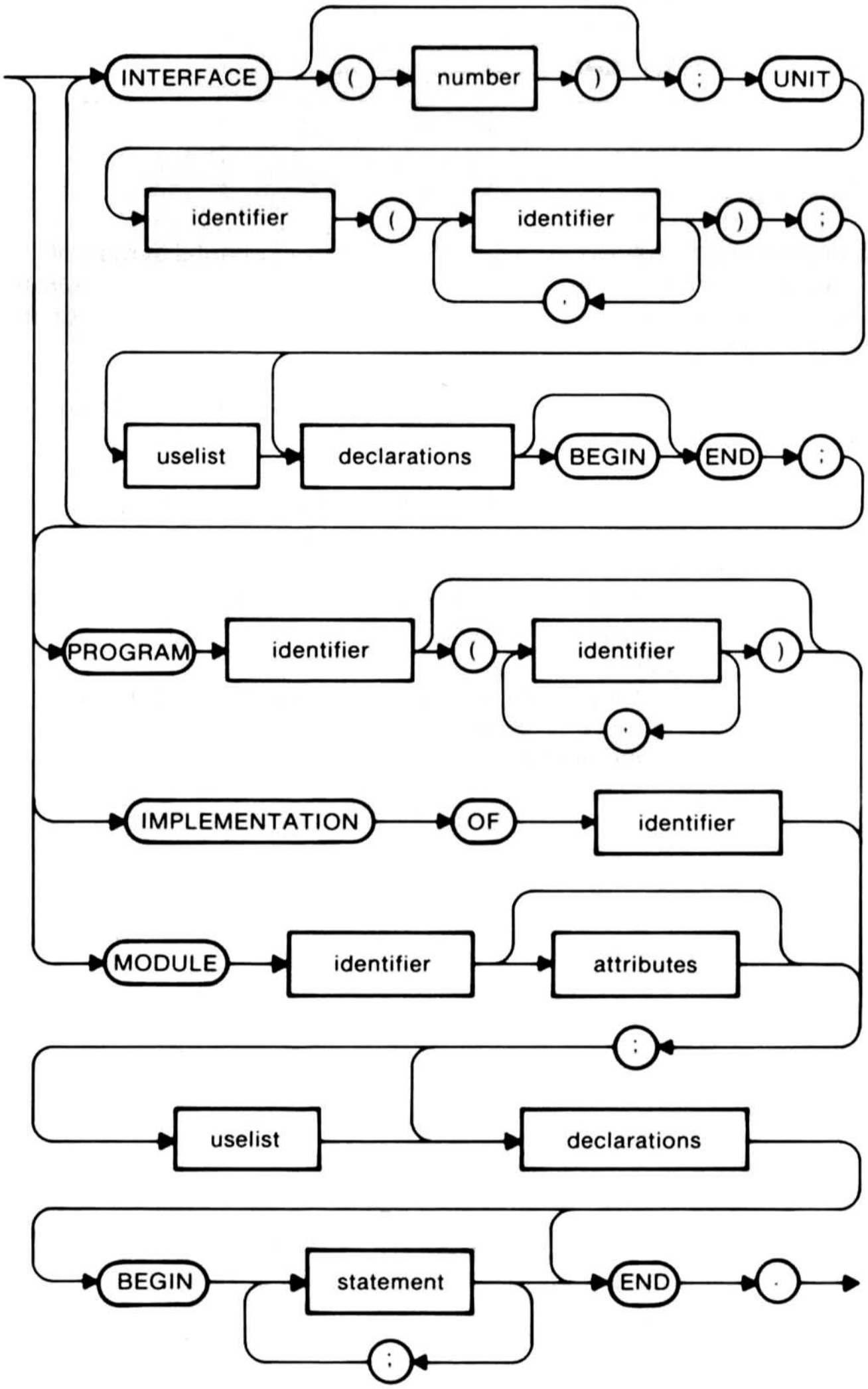
MICROSOFT®
Pascal Compiler
Quick Reference Guide

Syntax Diagrams

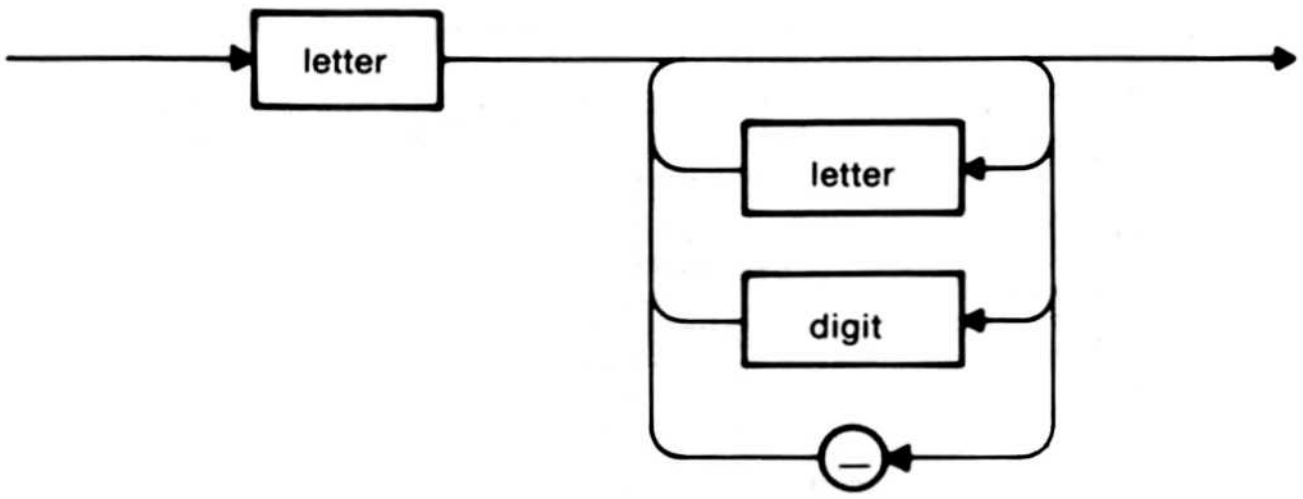
The diagrams on the following pages show the fundamental syntax of the MS™-Pascal language. They are arranged in the order that you would be likely to use the elements while writing a program. The meaning of the differently shaped outlines is as follows:

OVALS	Indicate reserved words or symbols of the MS-Pascal language. These must be typed as shown.
BOXES	Indicate higher-level constructions that usually have syntax diagrams of their own.
CIRCLES	Indicate punctuation that is required and must be typed as shown.
ARROWS	Help to show the path through the diagram, including any possible looping (i.e., repetition of syntax elements).

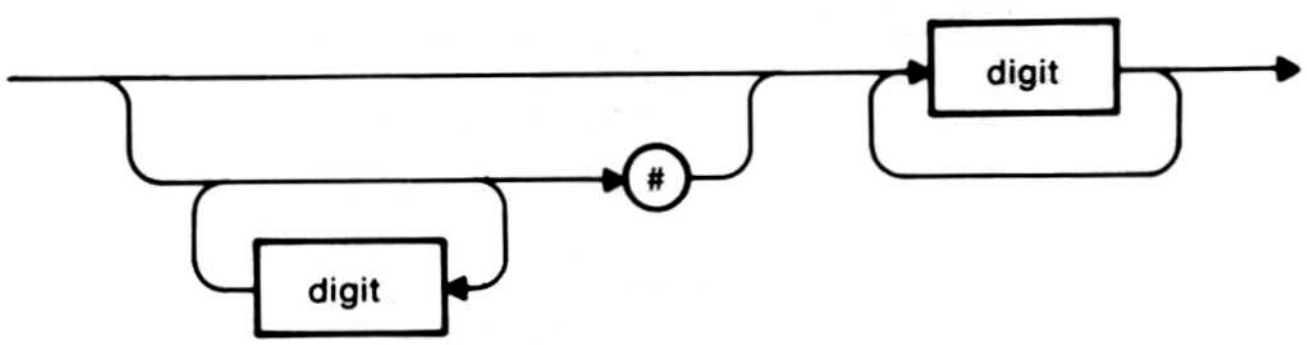
Source File


INTERFACE
(
number
)
;
UNIT
identifier
(
identifier
)
;
,
uselist
declarations
BEGIN
END
;
PROGRAM
identifier
(
identifier
)
,
IMPLEMENTATION
OF
identifier
MODULE
identifier
attributes
:
uselist
declarations
BEGIN
statement
END
.
;

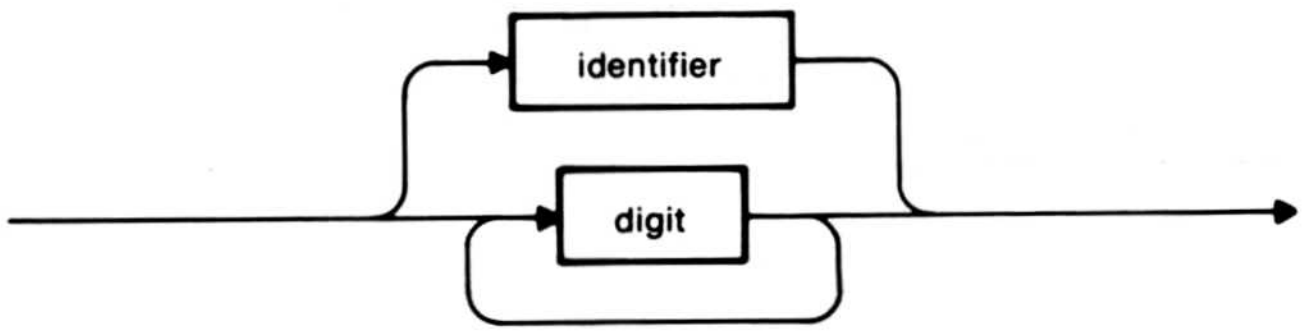
Identifier



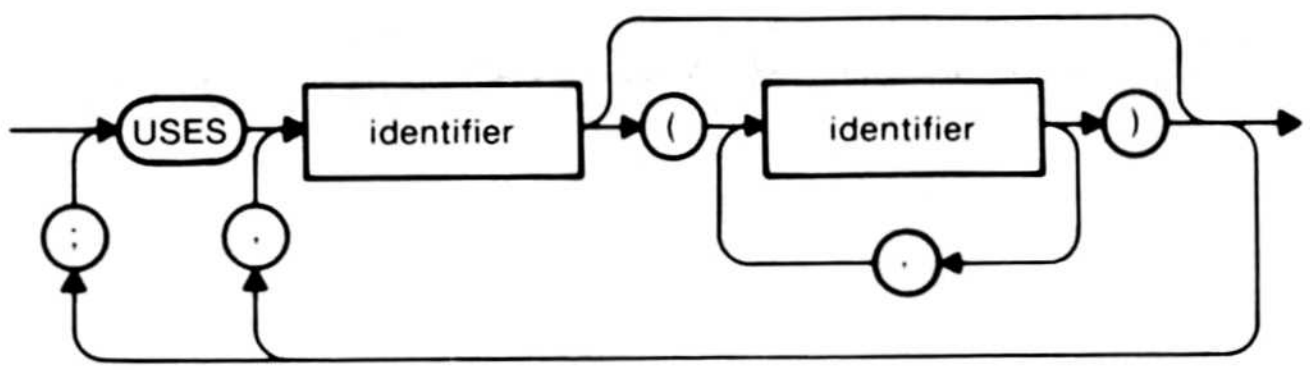
Number



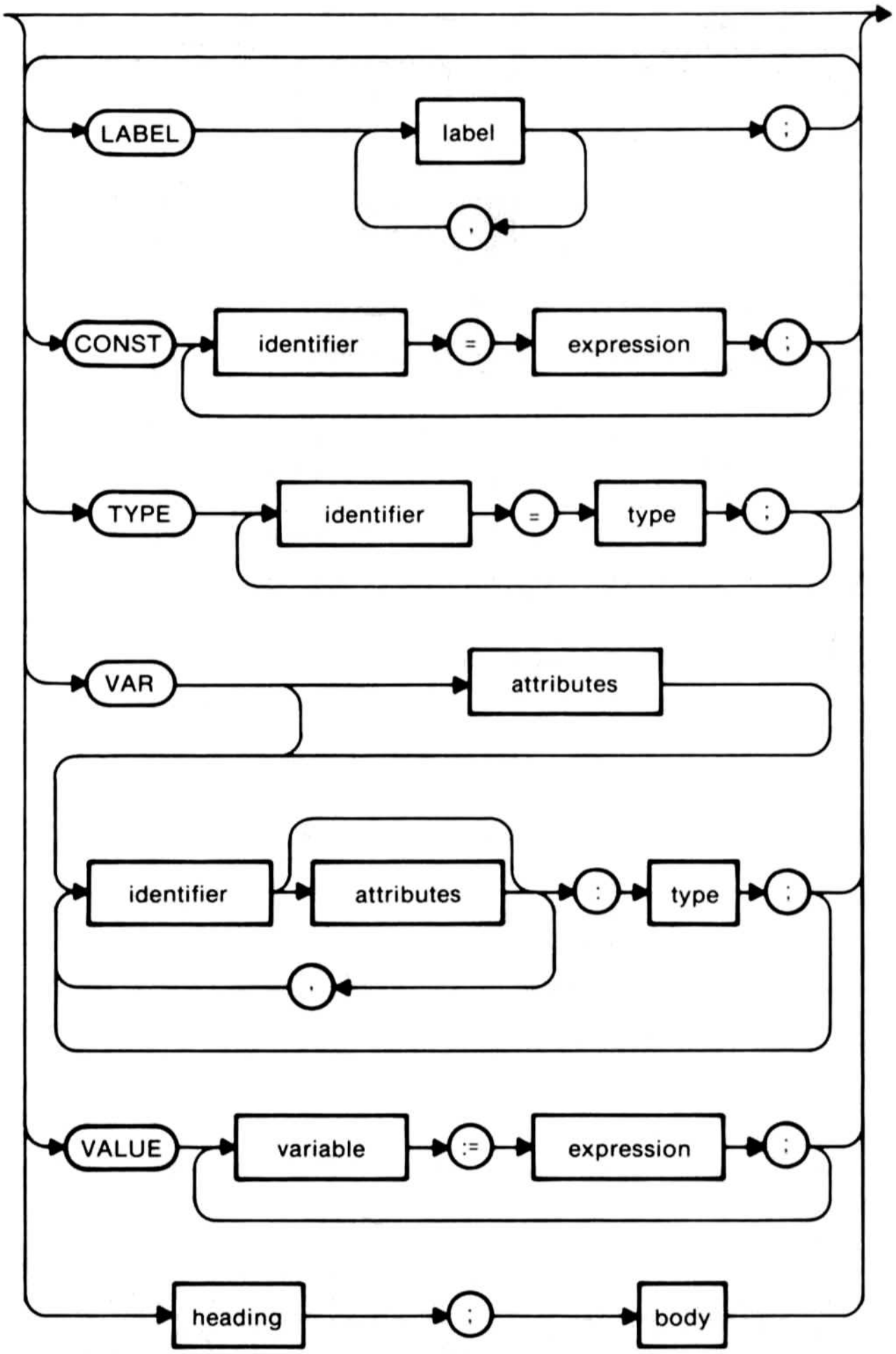
Label



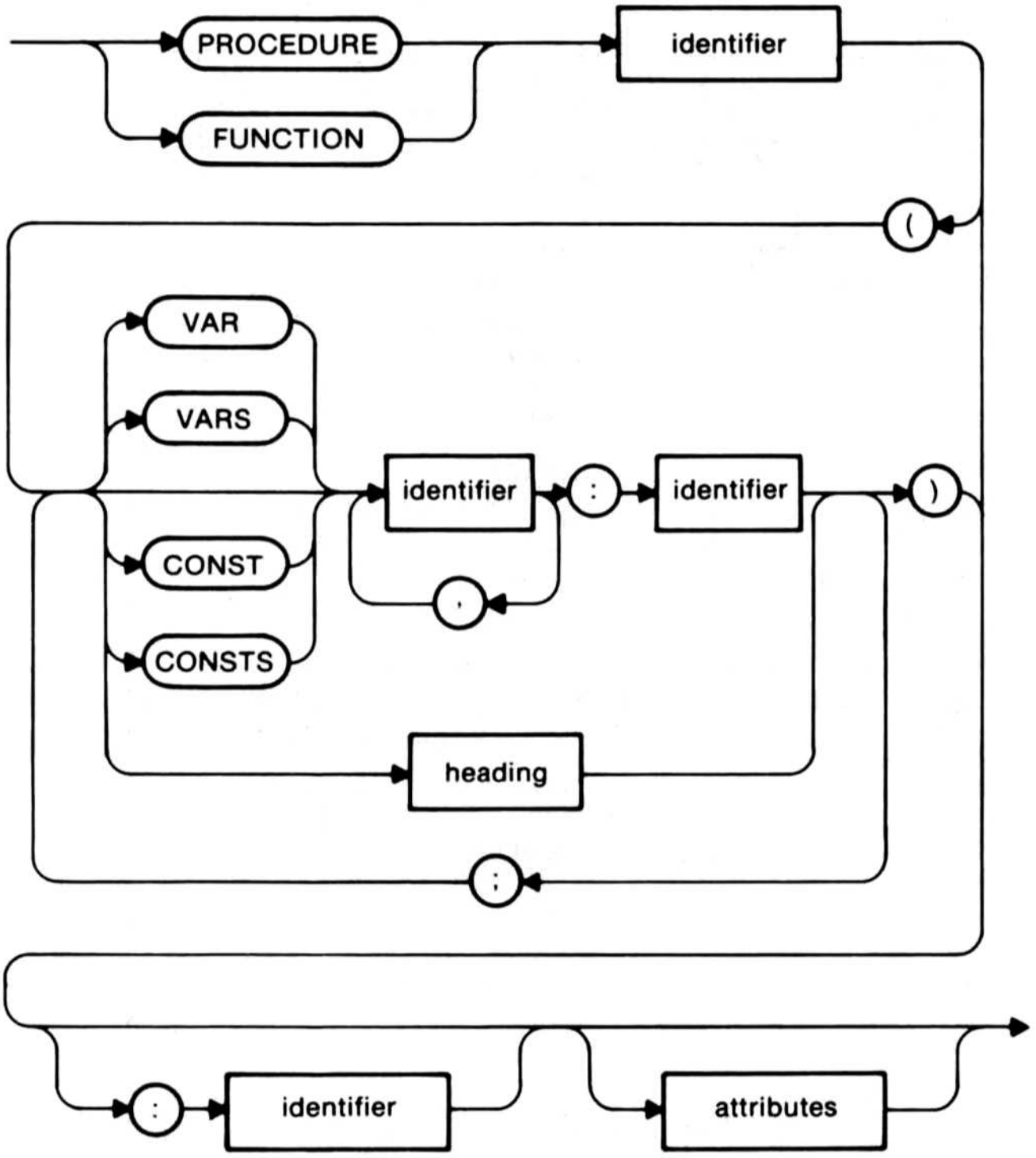
Uselist



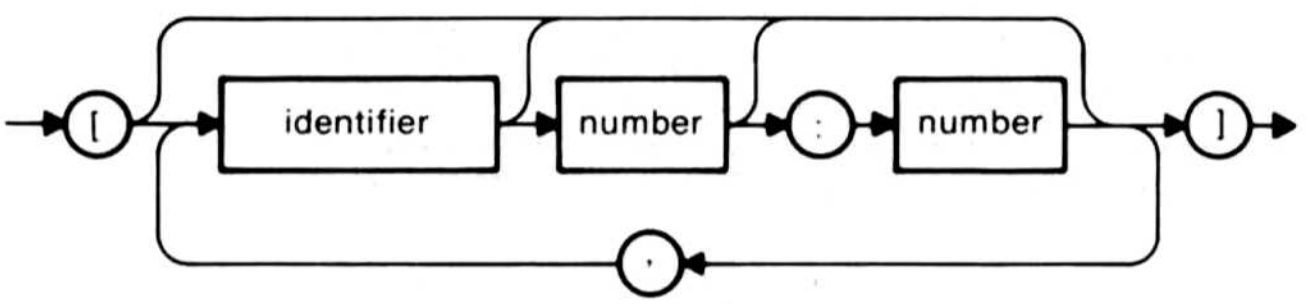
Declarations



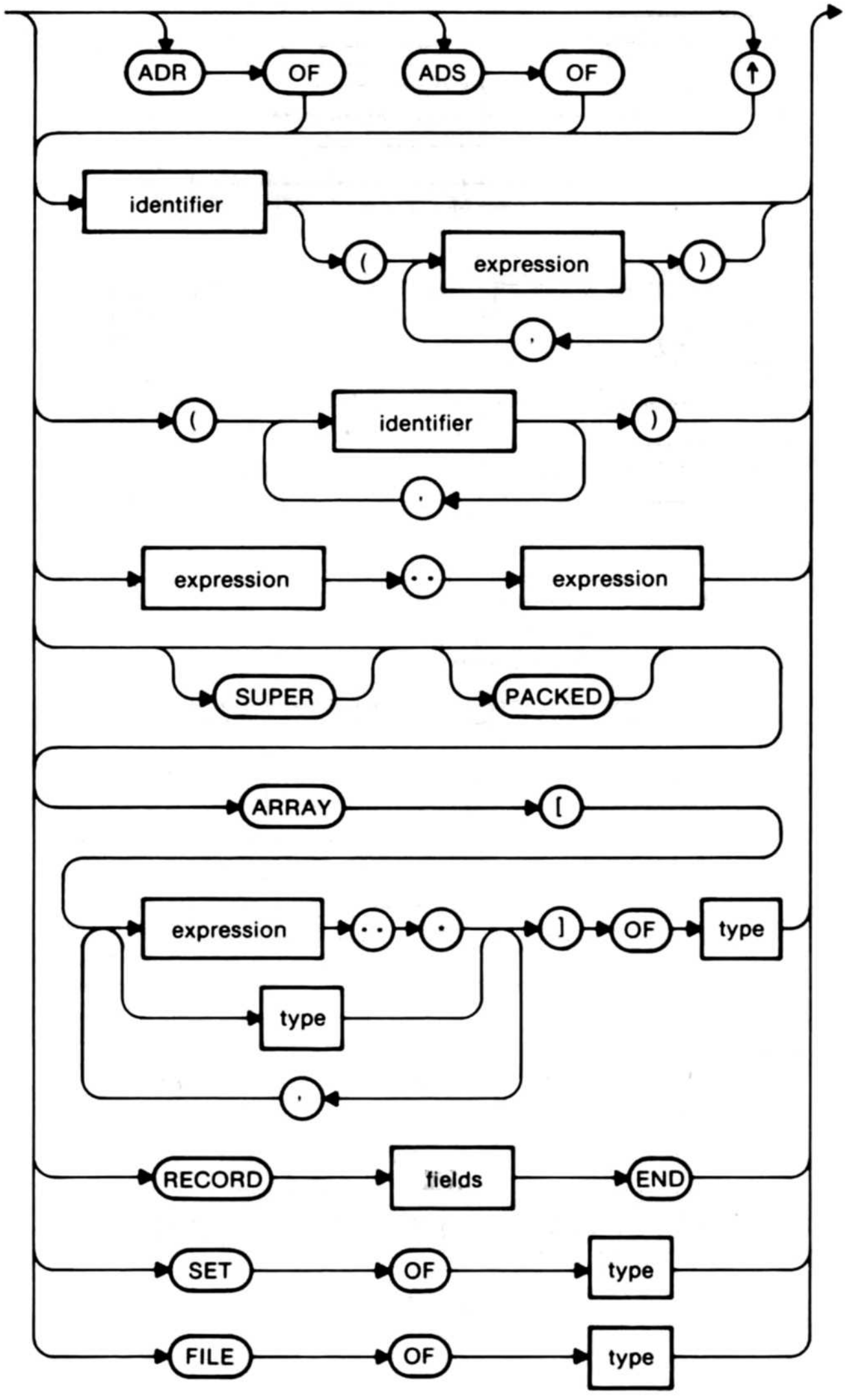
Heading



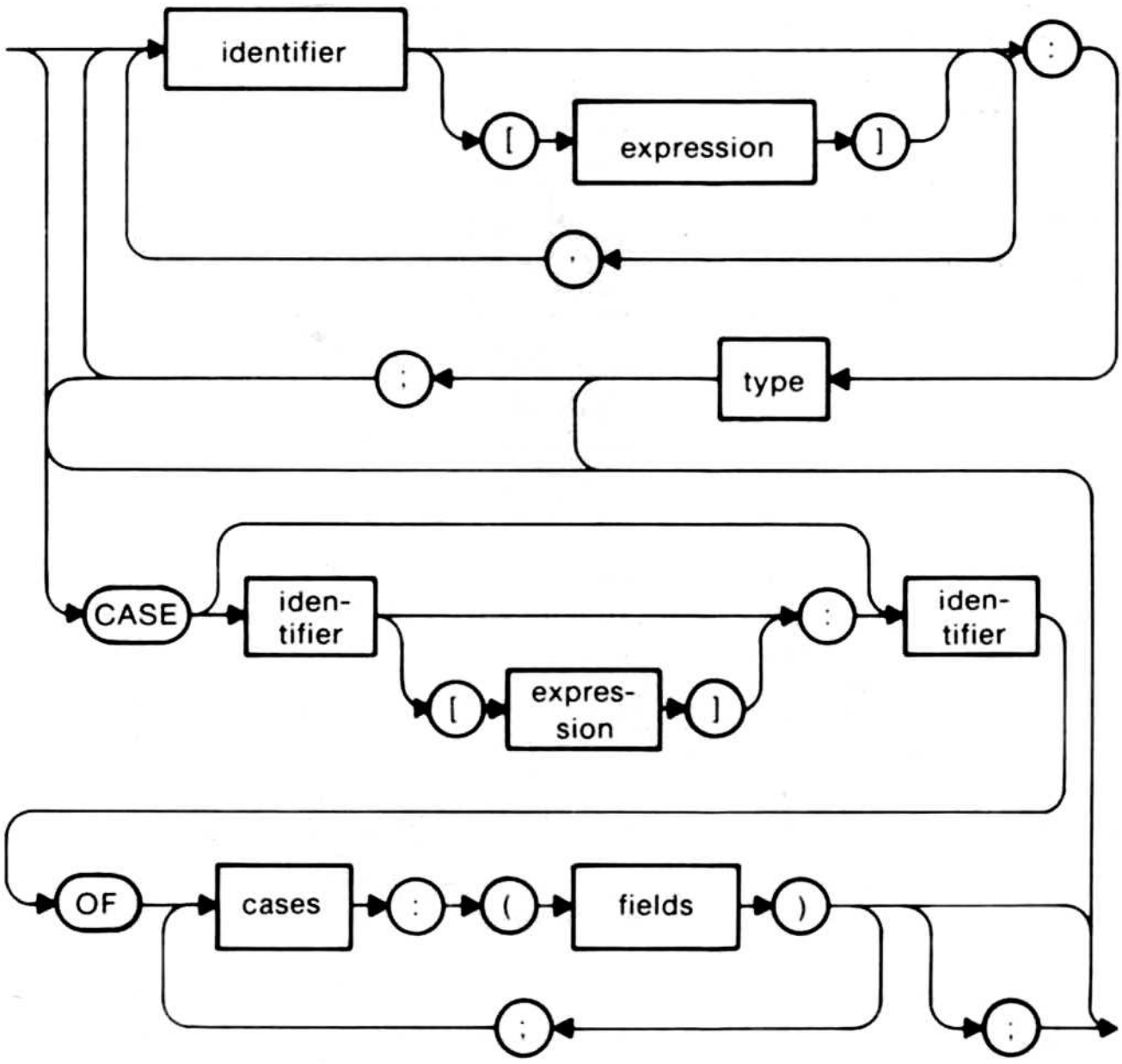
Attributes



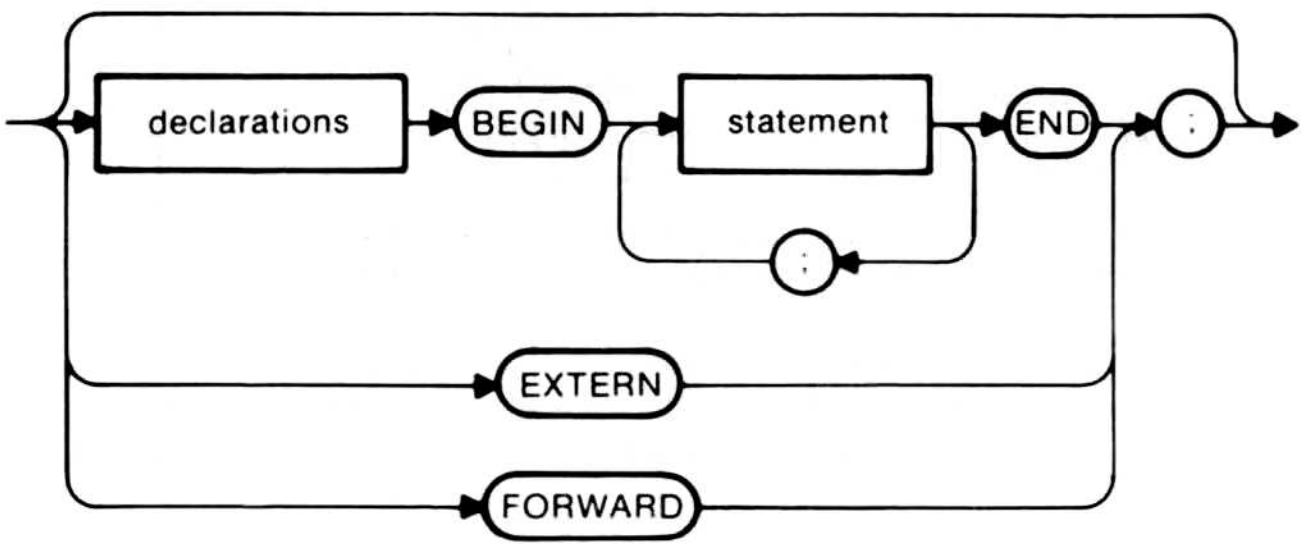
Type


ADR
OF
ADS
OF
↑
identifier
(
expression
)
,
(
identifier
)
,
expression
..
expression
SUPER
PACKED
ARRAY
[
expression
..
.
]
OF
type
type
,
RECORD
fields
END
SET
OF
type
FILE
OF
type

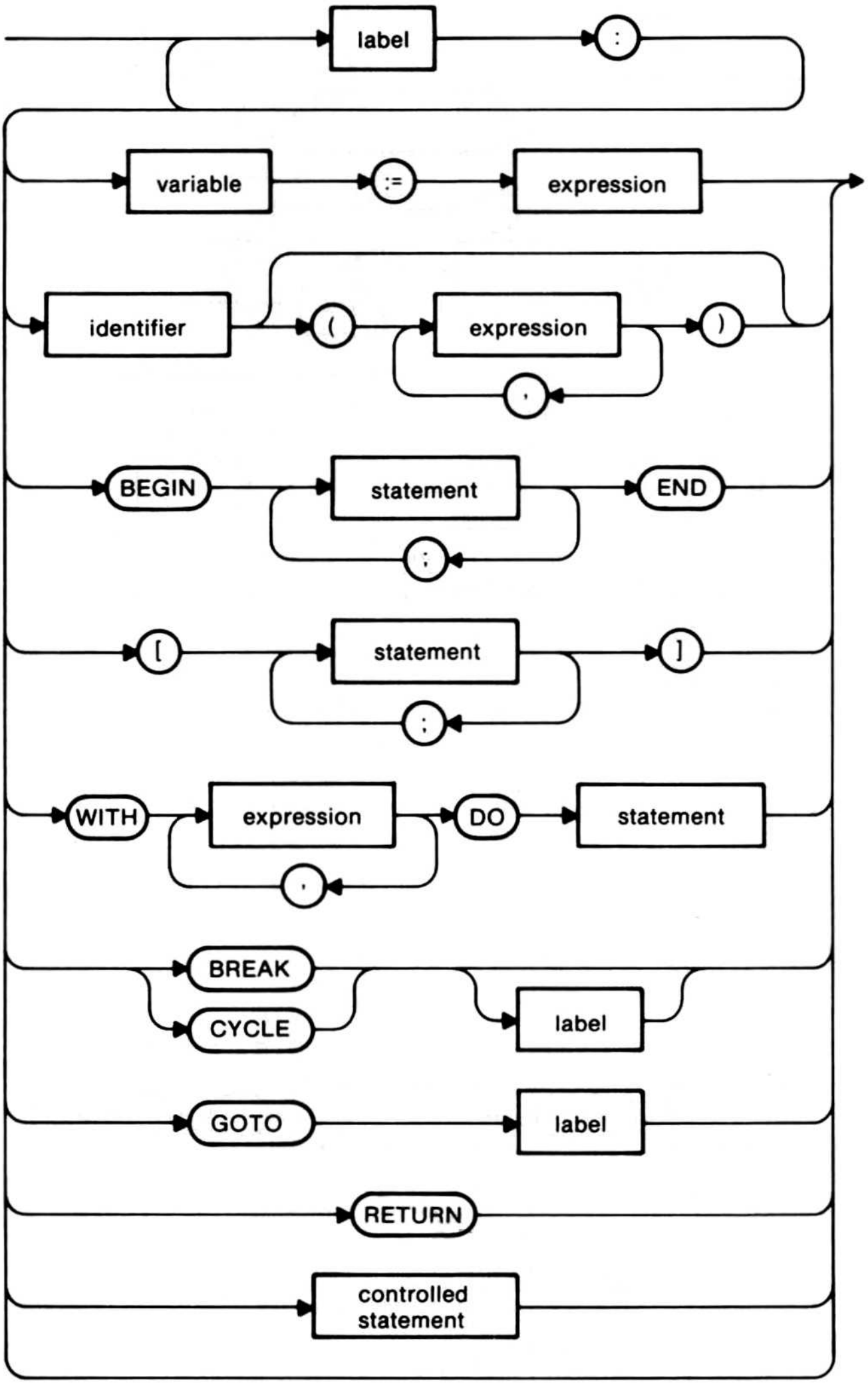
Fields


identifier
[
expression
]
:
,
:
type
CASE
iden-
tifier
[
expres-
sion
]
:
iden-
tifier
OF
cases
:
(
fields
)
;
;

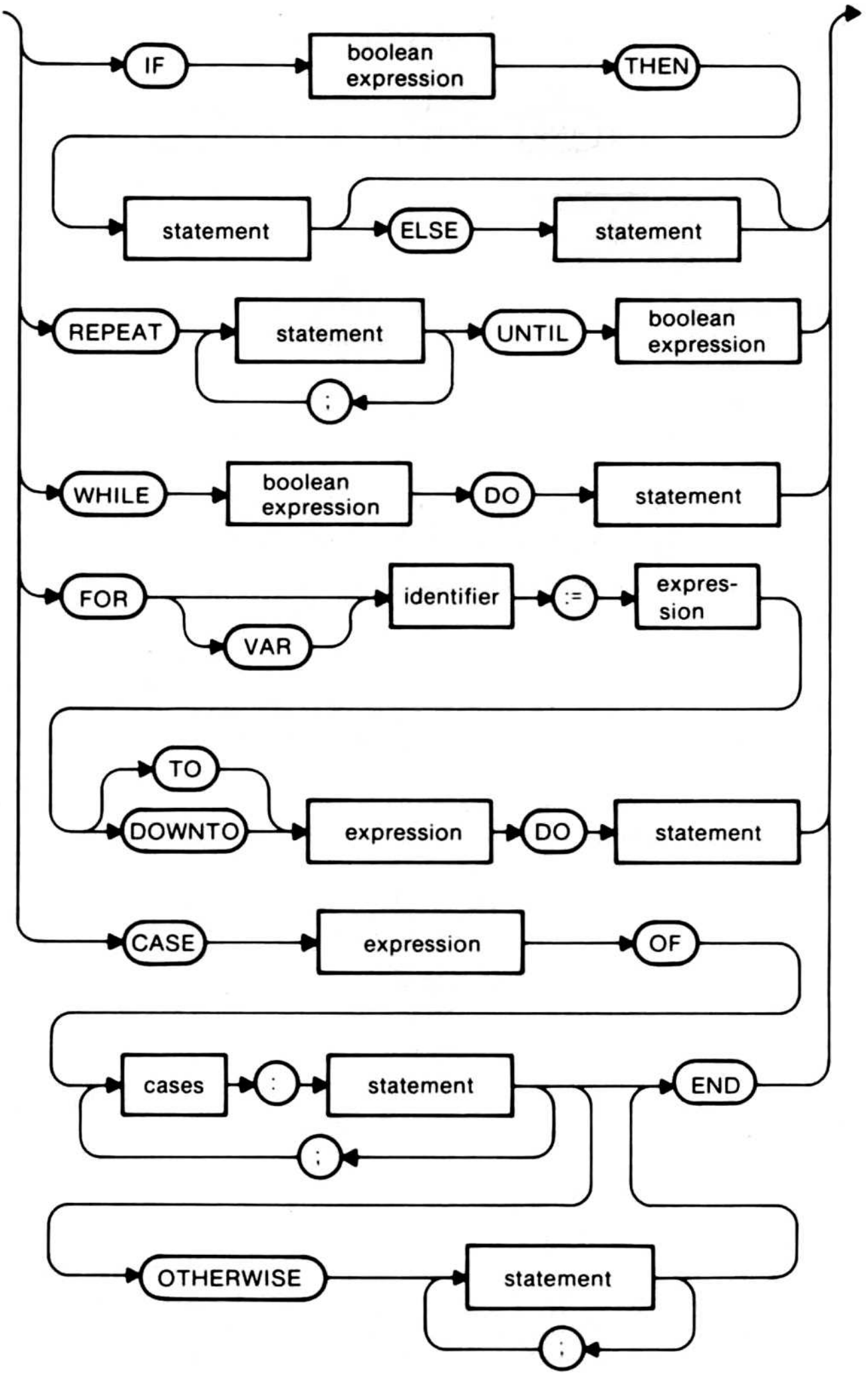
Body


declarations
BEGIN
statement
END
;
;
EXTERN
FORWARD

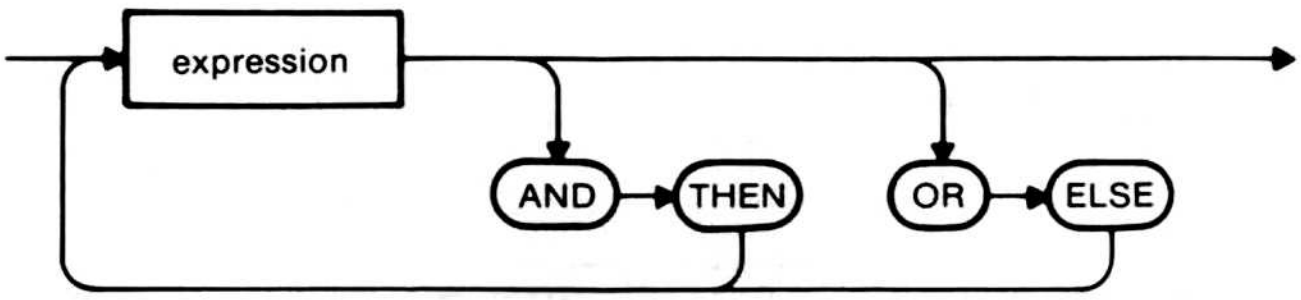
Statement


label
:
variable
:=
expression
identifier
(
expression
)
,
BEGIN
statement
END
;
[
statement
]
:
WITH
expression
DO
statement
,
BREAK
CYCLE
label
GOTO
label
RETURN
controlled
statement

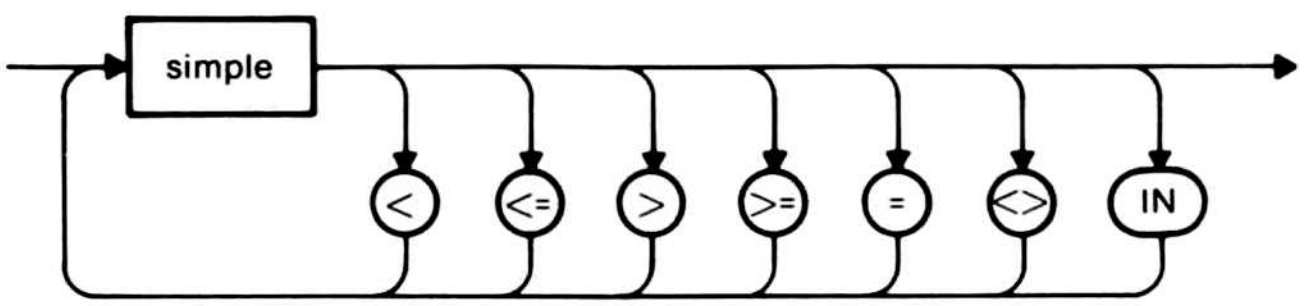
Controlled Statement



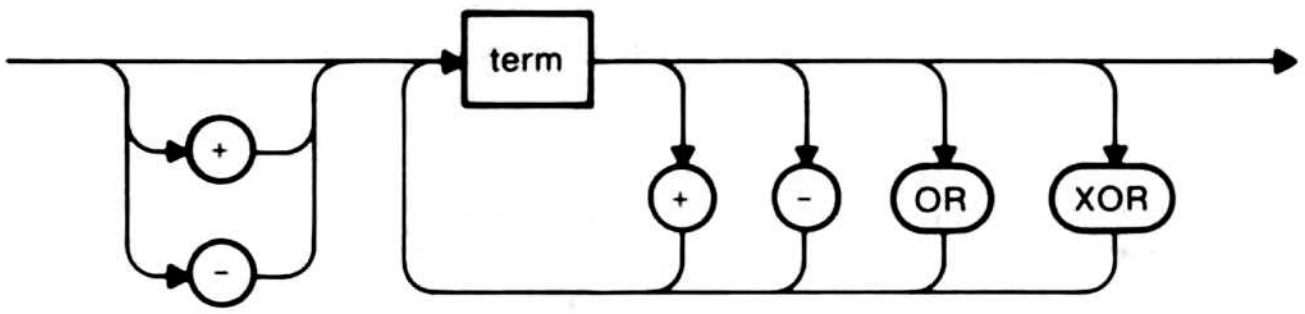
Boolean Expression



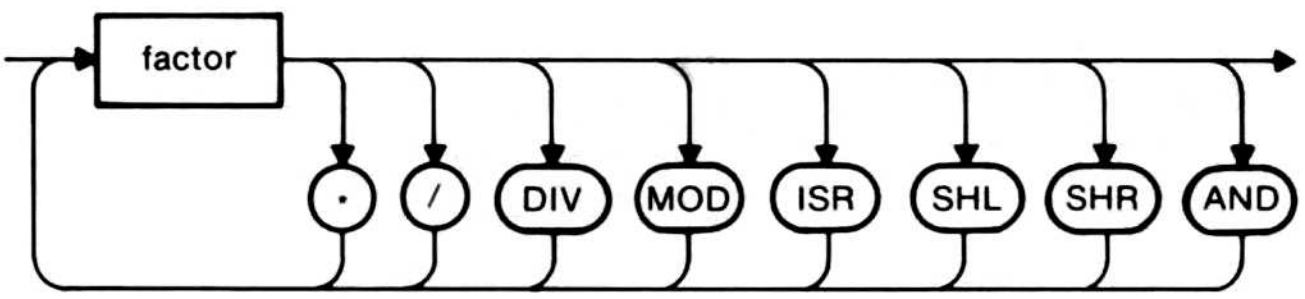
Expression



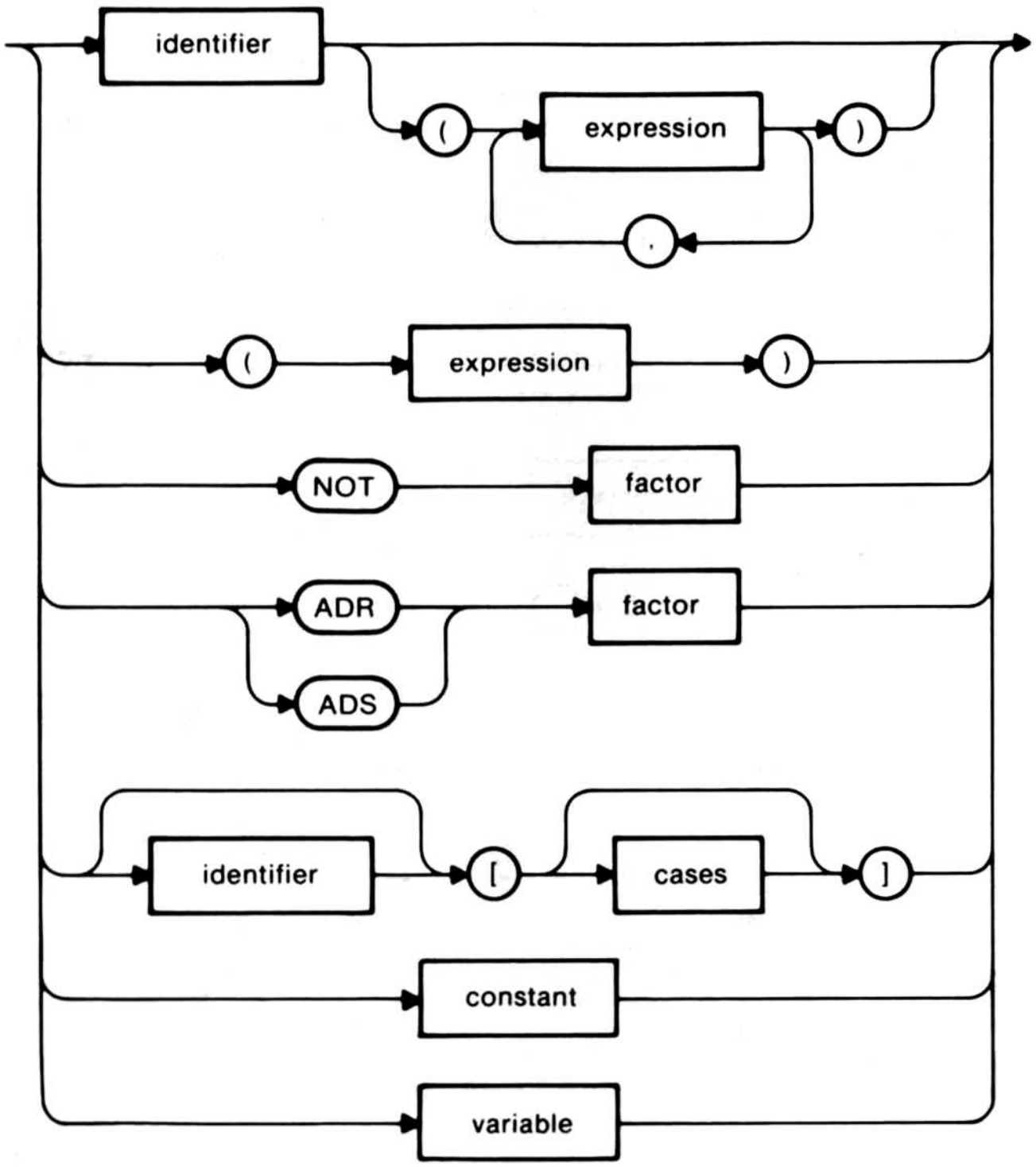
Simple



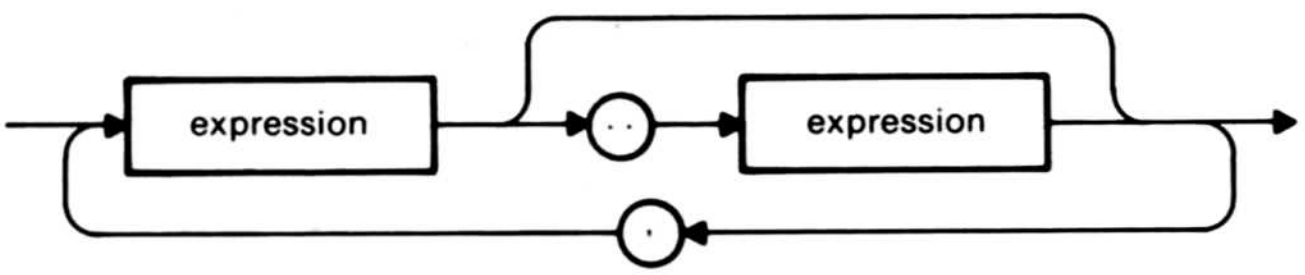
Term



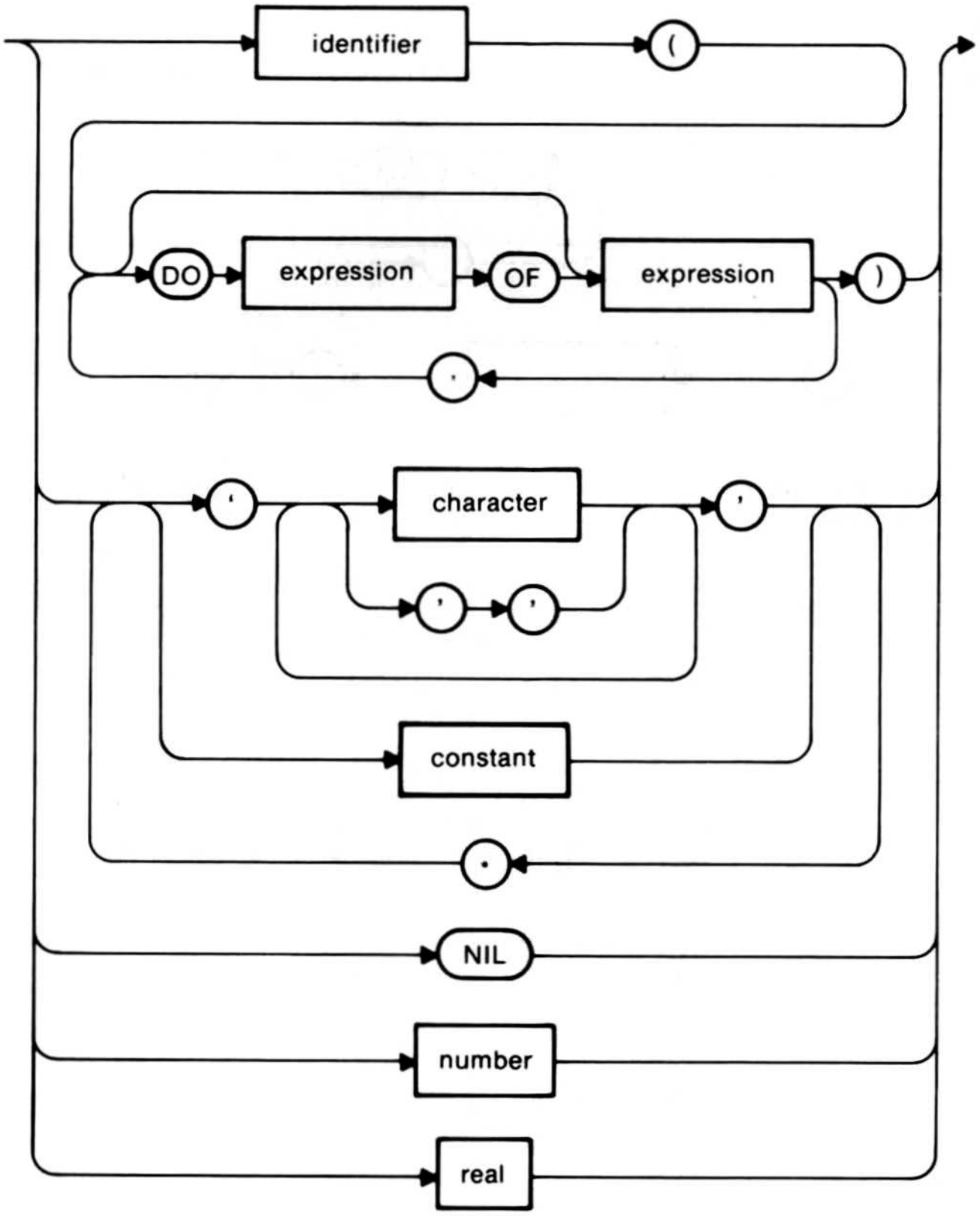
Factor



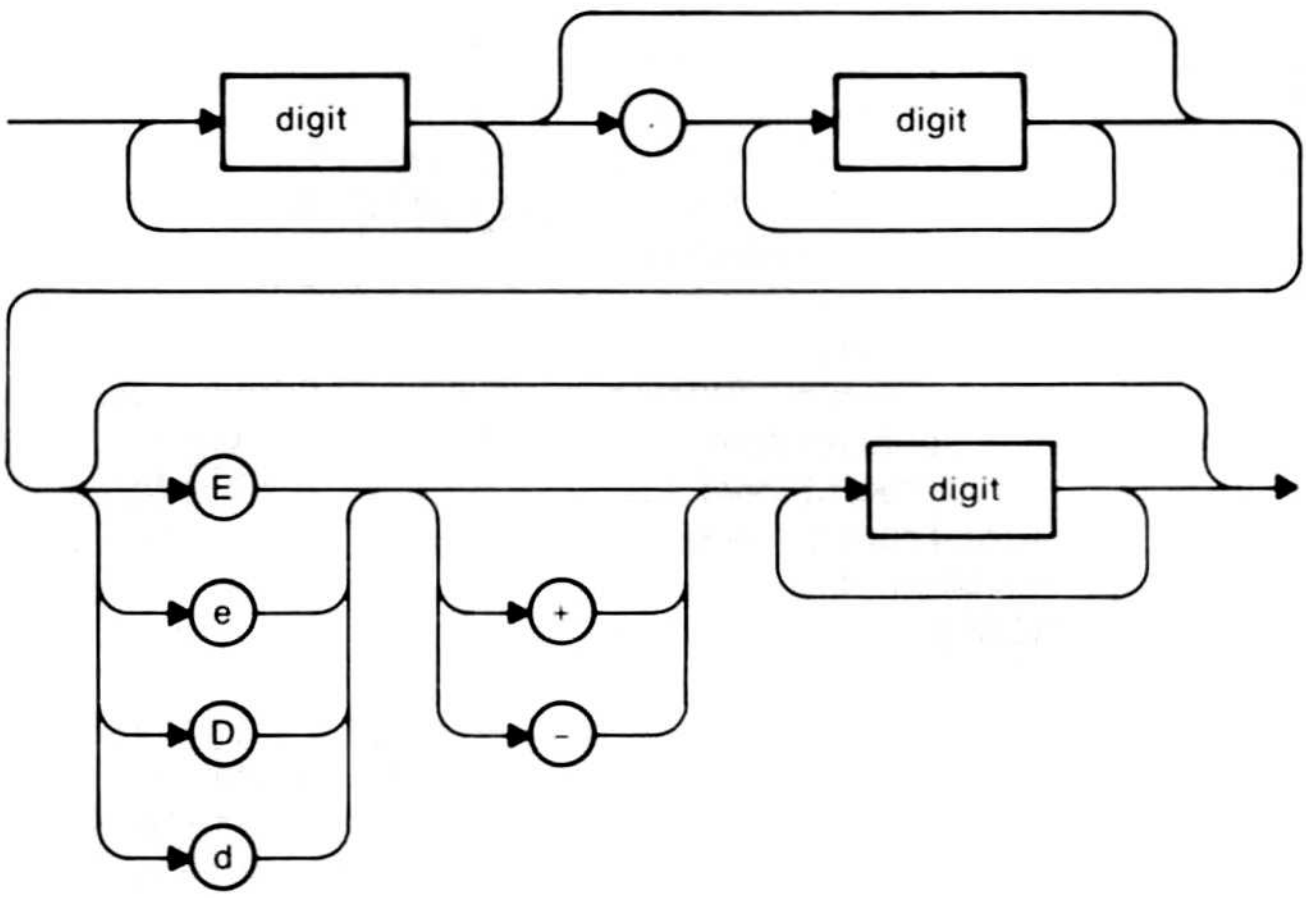
Cases



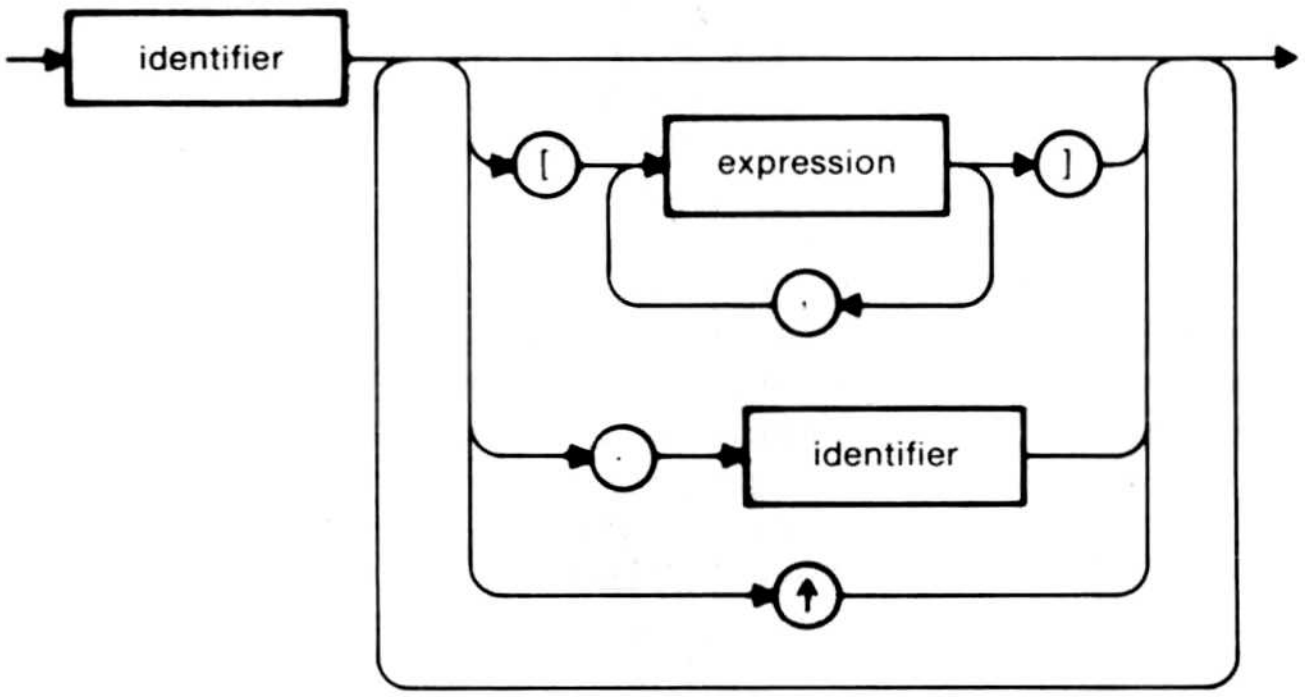
Constant


identifier
(
DO
expression
OF
expression
)
,
'
character
'
'
'
constant
.
NIL
number
real

Real



Variable



Available Procedures and Functions

Name	*Description*	*Category*
ABORT	Terminate program	Extend Level
ABS	Absolute value function	Arithmetic
ACDRQQ	REAL8 arc cosine function	Arithmetic
ACSRQQ	REAL4 arc cosine function	Arithmetic
AIDRQQ	REAL8 truncate function	Arithmetic
AISRQQ	REAL4 truncate function	Arithmetic
ALLHQQ	Allocate heap item	Library
ANDRQQ	REAL8 round toward zero	Arithmetic
ANSRQQ	REAL4 round toward zero	Arithmetic
ARCTAN	Arc tangent function	Arithmetic
ASDRQQ	REAL8 arc sine function	Arithmetic
ASSRQQ	REAL4 arc sine function	Arithmetic
ASSIGN	Assign filename	File System
ATDRQQ	REAL8 arc tangent function	Arithmetic
ATSRQQ	REAL4 arc tangent function	Arithmetic
A2DRQQ	REAL8 arc tangent (A/B)	Arithmetic
A2SRQQ	REAL4 arc tangent (A/B)	Arithmetic
BEGOQQ	Initialize user	Library
BEGXQQ	Overall initialization	Library
BYLONG	WORD or INTEGER to INTEGER4	Extend Level
BYWORD	Put bytes in word	Extend Level
CHDRQQ	REAL8 hyperbolic cosine	Arithmetic
CHR	Get ASCII char of value	Data Conversion
CHSRQQ	REAL4 hyperbolic cosine	Arithmetic
CLOSE	Close file	File System
CNDRQQ	REAL8 cosine function	Arithmetic
CNSRQQ	REAL4 cosine function	Arithmetic
CONCAT	Concatenate LSTRING	String
COPYLST	Copy to LSTRING	String
COPYSTR	Copy to STRING	String
COS	Cosine function	Arithmetic
DATE	Date function	Library
DECODE	Decode LSTRING to variable	Extend Level
DELETE	Remove portion of LSTRING	String
DISBIN	Disable interrupts	Library
DISCARD	Close and delete file	File System
DISPOSE	Dispose of heap item	Dynamic Alloc
ENABIN	Enable interrupts	Library
ENCODE	Encode expression to LSTRING	Extend Level
ENDOQQ	User termination	Library
ENDXQQ	Program termination	Library
EOF	Boolean end-of-file	File System

Name	*Description*	*Category*
EOLN	Boolean end-of-line	File System
EVAL	Evaluate functions	Extend Level
EXDRQQ	REAL8 exponential function	Arithmetic
EXP	Exponential function	Arithmetic
EXSRQQ	REAL4 exponential function	Arithmetic
FILLC	Fill area with C, relative	System Level
FILLSC	Fill area with C, segmented	System Level
FLOAT	Convert INTEGER to REAL	Data Conversion
FLOAT4	Convert INTEGER4 to REAL	Data Conversion
FREECT	Give count of free blocks	Library
GET	Get next file component	File System
GTYUQQ	Direct terminal input	Library
HDLUQQ	Return file handle	File System
HIBYTE	Get high BYTE	Extend Level
HIWORD	Get high WORD	Extend Level
INSERT	Insert string	String
LADDOK	32-bit signed addition check	Library
LDDRQQ	REAL8 log base ten function	Arithmetic
LDSRQQ	REAL4 log base ten function	Arithmetic
LMULOK	32-bit signed multiply check	Arithmetic
LN	Natural log function	Arithmetic
LNDRQQ	REAL8 natural log function	Arithmetic
LNSRQQ	REAL4 natural log function	Arithmetic
LOBYTE	Get low BYTE	Extend Level
LOCKED	Resource locked status	Library
LOWER	Get lower bound	Extend Level
LOWORD	Get low WORD	Extend Level
MARKAS	Mark heap bounds	Library
MDDRQQ	REAL8 modulo function	Arithmetic
MDSRQQ	REAL4 modulo function	Arithmetic
MEMAVL	Available memory	Library
MNDRQQ	REAL8 minimum function	Arithmetic
MNSRQQ	REAL4 minimum function	Arithmetic
MOVEL	Move bytes left, relative	System Level
MOVER	Move bytes right, relative	System Level
MOVESL	Move bytes left, segmented	System Level
MOVESR	Move bytes right, segmented	System Level
MXDRQQ	REAL8 maximum function	Arithmetic
MXSRQQ	REAL4 maximum function	Arithmetic
NEW	Allocate new heap item	Dynamic Alloc
ODD	Boolean odd function	Data Conversion
ORD	Get ordinal value	Data Conversion
PACK	Pack CHAR array	Data Conversion
PAGE	Write new page	File System
PIDRQQ	REAL8 to INTEGER power	Arithmetic
PISRQQ	REAL4 to INTEGER power	Arithmetic

Name	*Description*	*Category*
PLYUQQ	Direct terminal end line	Library
POSITN	Find position of substring	String
PRDRQQ	REAL8 to REAL8 power	Arithmetic
PRED	Predecessor function	Data Conversion
PRSRQQ	REAL4 to REAL4 power	Arithmetic
PTYUQQ	Direct terminal output	Library
PUT	Put value to file	File System
READ	Read file	File System
READFN	Read filename	File System
READLN	Read file to end-of-line	File System
READSET	Read set	File System
RELEAS	Release heap space	Library
RESET	Ready file for read	File System
RESULT	Return result of function	Extend Level
RETYPE	Force expression to type	System Level
REWRITE	Ready file for write	File System
ROUND	Round REAL	Data Conversion
ROUND4	Round INTEGER4	Data Conversion
SADDOK	16-bit signed addition check	Library
SCANEQ	Scan until char found	String
SCANNE	Scan until char not found	String
SEEK	Position at direct file record	File System
SHDRQQ	REAL8 hyperbolic sine	Arithmetic
SHSRQQ	REAL4 hyperbolic sine	Arithmetic
SIN	Sine function	Arithmetic
SIZEOF	Get size of structure	Extend Level
SMULOK	16-bit signed multiply check	Library
SNDRQQ	REAL8 sine function	Arithmetic
SNSRQQ	REAL4 sine function	Arithmetic
SQR	Square function	Arithmetic
SQRT	Square root function	Arithmetic
SRDRQQ	REAL8 square root function	Arithmetic
SRSRQQ	REAL4 square root function	Arithmetic
SUCC	Successor function	Data Conversion
THDRQQ	REAL8 hyperbolic tangent	Arithmetic
THSRQQ	REAL4 hyperbolic tangent	Arithmetic
TICS	Time in arbitrary units	Library
TIME	Time of day function	Library
TNDRQQ	REAL8 tangent function	Arithmetic
TNSRQQ	REAL4 tangent function	Arithmetic
TRUNC	Truncate REAL	Data Conversion
TRUNC4	Truncate INTEGER4	Data Conversion
UADDOK	Unsigned addition check	Library
UMULOK	Unsigned multiply check	Library
UNLOCK	Unlock resource	Library
UNPACK	Unpack STRING to array	Data Conversion

Name	*Description*	*Category*
UPPER	Get upper bound	Extend Level
VECTIN	Set interrupt vector	Library
WRD	Convert to WORD value	Data Conversion
WRITE	Write file	File System
WRITELN	Write line to file	File System

Metacommands

Defaults, if any, are shown following the metacommand in column one.

Metacommand	*Action*
$brave+	Sends messages to the terminal screen.
$debug-	Turns on or off all error checking (CK).
$decmath-	Directs the compiler to use the decimal math routines in the auxiliary math runtime library, DECMATH.LIB.
$entry-	Generates procedure entry and exit calls for debugger.
$errors:25	Sets number of errors allowed per page.
$extend	Adds extend level features.
$floatcalls+	Directs the compiler to make calls to the real number math routines.
$goto-	Flags GOTOs as "considered harmful."
$if *constant* $then *text1* $else *text2* $end	Allows conditional compilation of *text1* source if *constant* is greater than zero.
$include:'*file*'	Switches compilation to file named.
$inconst:*text*	Allows interactive setting of constant values at compile time.
$indexck-	Checks for array index values in range.
$initck-	Checks for use of uninitialized values.

Metacommand	*Action*
$integer:2	Sets the length of the INTEGER type.
$line-	Generates line number calls for debugger.
$linesize:79	Sets width of source listing.
$list+	Turns on or off source listing.
$mathck-	Checks for mathematical errors.
$message:'*text*'	Displays a message on terminal screen.
$nilck-	Checks for bad pointer values.
$ocode+	Turns on or off object code listing.
$page+	Skips to next page.
$page:*n*	Sets page number for next page.
$pageif:*n*	Skips to next page if less than n lines left.
$pagesize:55	Sets page length of source listing.
$pop	Restores saved value of all metacommands.
$push	Saves current value of all metacommands.
$rangeck-	Checks for subrange validity.
$real:4	Sets the length of the REAL type.
$rom	Warns on static initialization.
$runtime-	Determines context of runtime errors.
$simple	Disables global optimizations.
$size	Minimizes size of code generated.
$skip:*n*	Skips n lines or to end of page.
$speed	Minimizes execution time of code.
$stackck-	Checks for stack overflow at entry.
$standard	Enables standard level only.

Metacommand	*Action*
$subtitle:'*text*'	Sets page subtitle.
$symtab+	Sends symbol table to source listing.
$system	Adds extend and system level features.
$tagck-	Checks tag fields in variant records.
$title:'text'	Gives page title for source listing.
$warn+	Gives warning messages in source listing.

Error Code Classification

Range	*Classification*
1- 999	Front end errors
1000-1099	Unit U file system errors
1100-1199	Unit F file system errors
1200-1299	Unit V file system errors
1300-1999	Reserved
2000-2049	Heap, stack, memory
2050-2099	Ordinal and INTEGER4 arithmetic
2100-2149	REAL4 and REAL8 arithmetic
2150-2199	Structures, sets, and strings
2200-2399	Reserved
2400-2449	P-code interpreter
2450-2499	Other internal errors
2500-2999	Reserved

ASCII Character Codes

Dec	*Hex*	*CHR*	*Dec*	*Hex*	*CHR*	*Dec*	*Hex*	*CHR*
000	00H	NUL	043	2BH	+	086	56H	V
001	01H	SOH	044	2CH	,	087	57H	W
002	02H	STX	045	2DH	-	088	58H	X
003	03H	ETX	046	2EH	.	089	59H	Y
004	04H	EOT	047	2FH	/	090	5AH	Z
005	05H	ENQ	048	30H	0	091	5BH	[
006	06H	ACK	049	31H	1	092	5CH	\
007	07H	BEL	050	32H	2	093	5DH	]
008	08H	BS	051	33H	3	094	5EH	^
009	09H	HT	052	34H	4	095	5FH	_
010	0AH	LF	053	35H	5	096	60H	'
011	0BH	VT	054	36H	6	097	61H	a
012	0CH	FF	055	37H	7	098	62H	b
013	0DH	CR	056	38H	8	099	63H	c
014	0EH	SO	057	39H	9	100	64H	d
015	0FH	SI	058	3AH	:	101	65H	e
016	10H	DLE	059	3BH	;	102	66H	f
017	11H	DC1	060	3CH	<	103	67H	g
018	12H	DC2	061	3DH	=	104	68H	h
019	13H	DC3	062	3EH	>	105	69H	i
020	14H	DC4	063	3FH	?	106	6AH	j
021	15H	NAK	064	40H	@	107	6BH	k
022	16H	SYN	065	41H	A	108	6CH	l
023	17H	ETB	066	42H	B	109	6DH	m
024	18H	CAN	067	43H	C	110	6EH	n
025	19H	EM	068	44H	D	111	6FH	o
026	1AH	SUB	069	45H	E	112	70H	p
027	1BH	ESCAPE	070	46H	F	113	71H	q
028	1CH	FS	071	47H	G	114	72H	r
029	1DH	GS	072	48H	H	115	73H	s
030	1EH	RS	073	49H	I	116	74H	t
031	1FH	US	074	4AH	J	117	75H	u
032	20H	SPACE	075	4BH	K	118	76H	v
033	21H	!	076	4CH	L	119	77H	w
034	22H	"	077	4DH	M	120	78H	x
035	23H	#	078	4EH	N	121	79H	y
036	24H	$	079	4FH	O	122	7AH	z
037	25H	%	080	5OH	P	123	7BH	{
038	26H	&	081	51H	Q	124	7CH	|
039	27H	'	082	52H	R	125	7DH	}
040	28H	(	083	53H	S	126	7EH	~
041	29H	)	084	54H	T	127	7FH	DEL
042	2AH	*	085	55H	U			

Dec=Decimal, Hex=Hexadecimal (H), CHR=Character, LF=Linefeed, FF=Form Feed, CR=Carriage Return, DEL=Rubout

MICROSOFT®

Microsoft Corporation
10700 Northup Way
Box 97200
Bellevue, WA 98009

Part No. 020-000-010