

SYNTAX DIAGRAMS

Contents

---

Appendix E SYNTAX DIAGRAMS ..... E-1

KAREL syntax diagrams use the following symbols:

- Rectangle



A rectangle encloses elements that are defined in another syntax diagram or in accompanying text.

- Oval



An oval encloses KAREL reserved words that are entered exactly as shown.

- Circle



A circle encloses special characters that are entered exactly as shown.

- Dot



A dot indicates a mandatory line-end (; or ENTER key) before the next syntax element.

- Caret



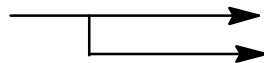
A caret indicates an optional line-end.

- Arrows



Arrows indicate allowed paths and the correct sequence in a diagram.

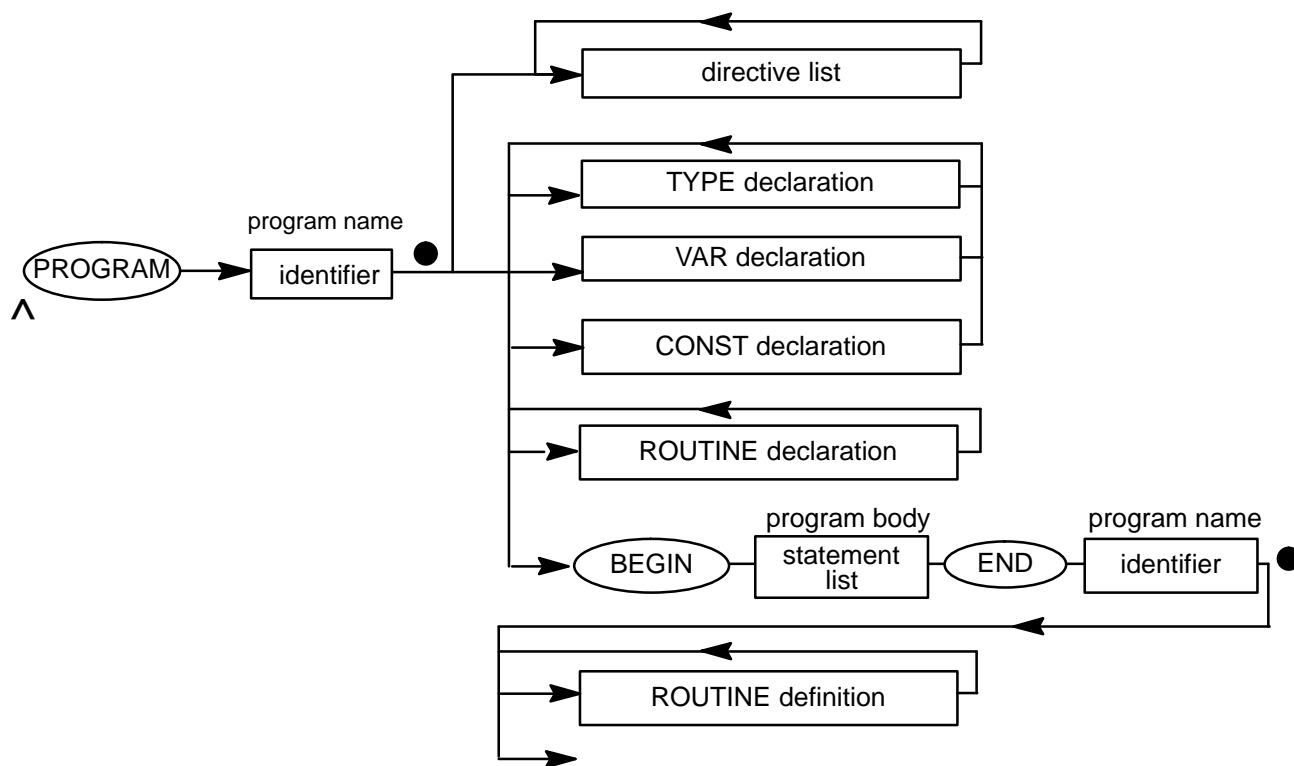
- Branch



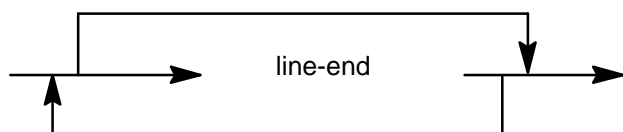
Branches indicate optional paths or sequences.

Figure E-1.

PROGRAM—module definition



^ -- 0 or more line-ends



● --newline

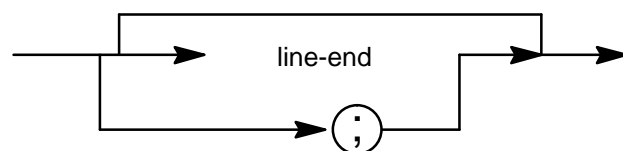


Figure E-2.

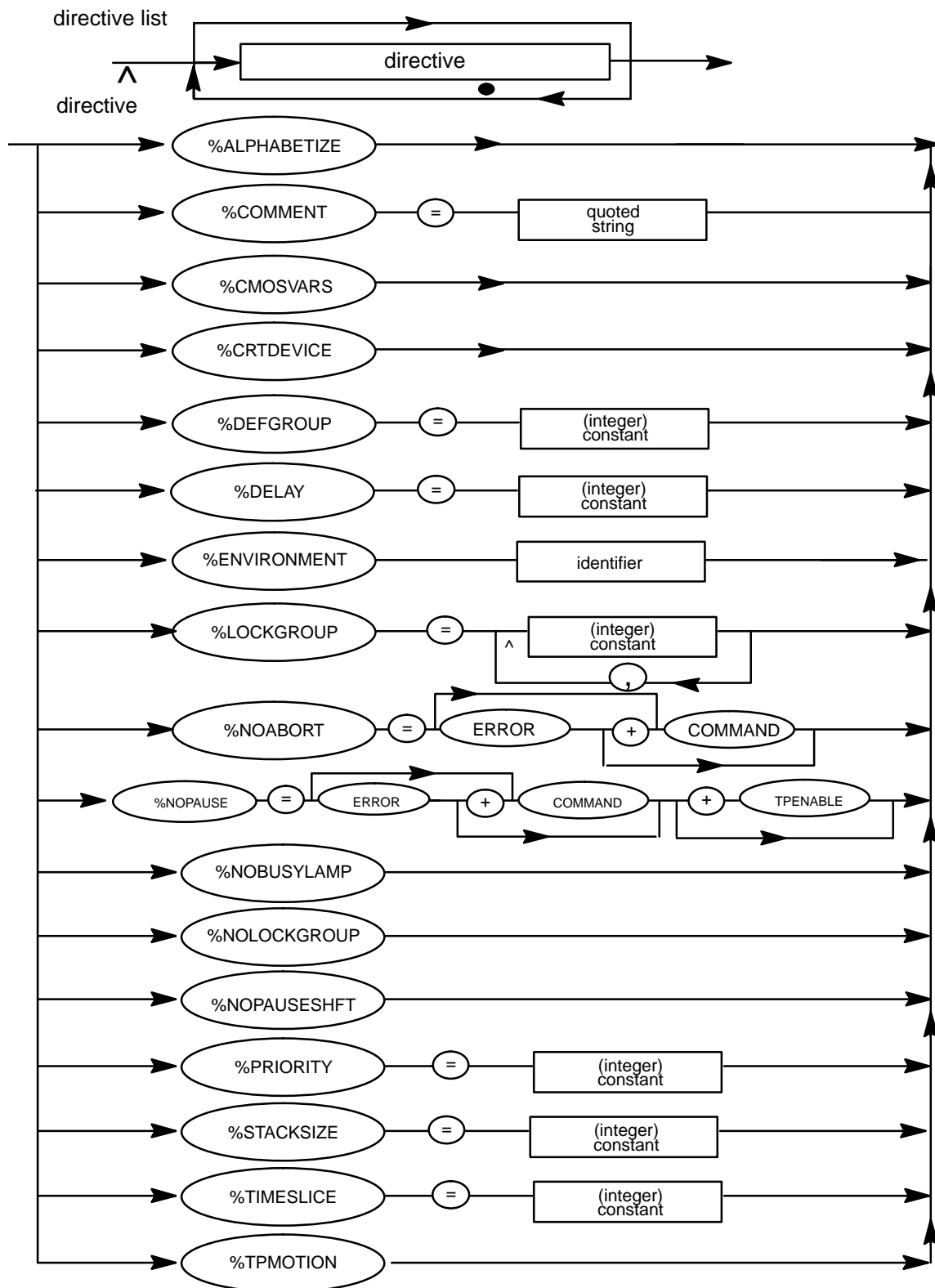
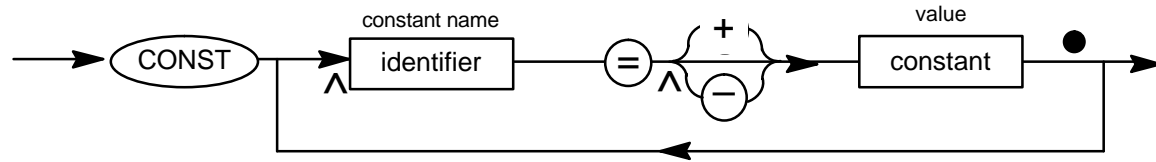
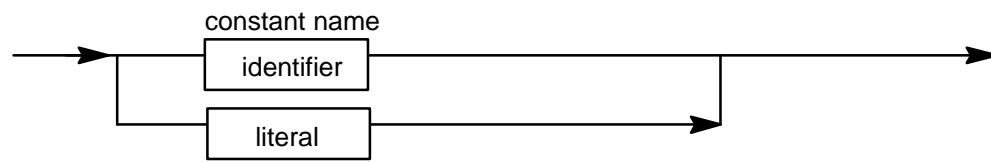


Figure E-3.

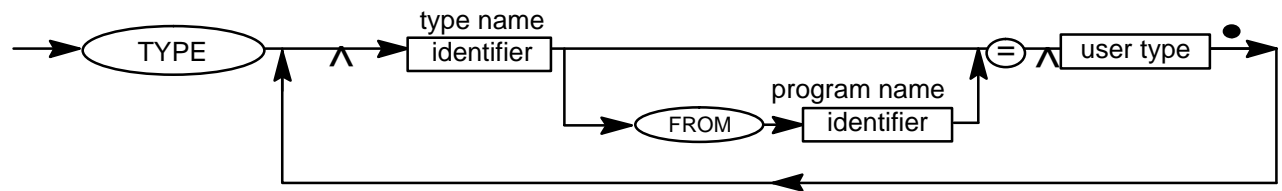
CONST --constant declaration



constant

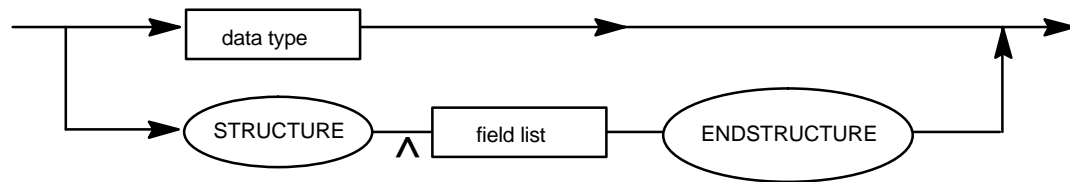


TYPE -- type declaration

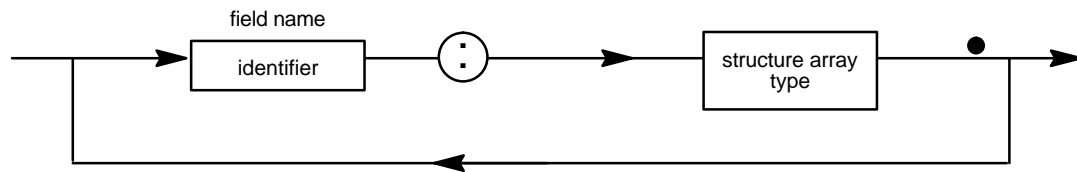


**Figure E-4.**

user type



field list



structure array type

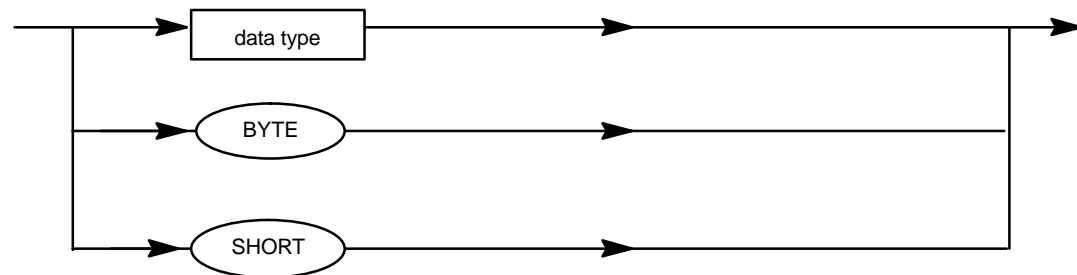
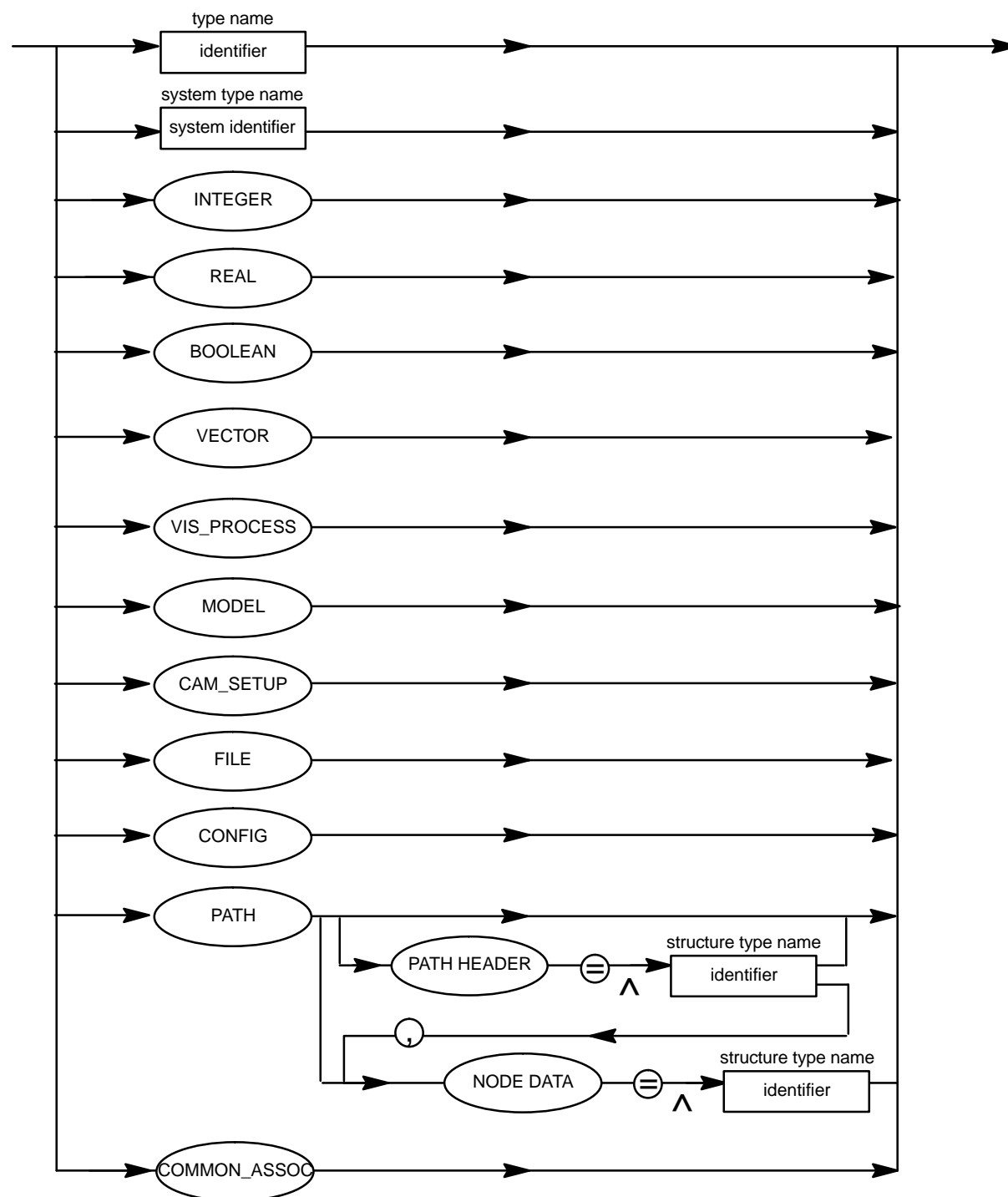


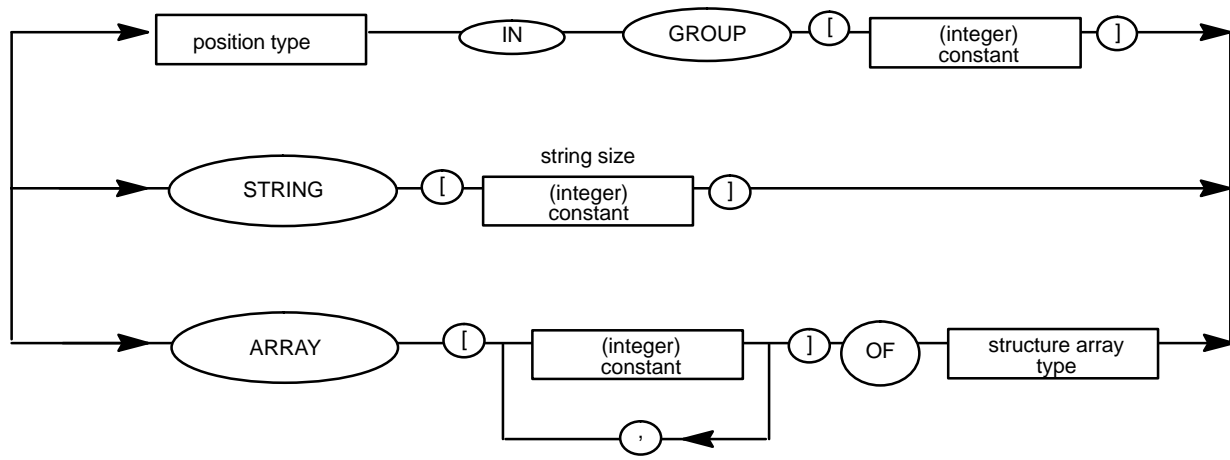
Figure E-5.

data type



**Figure E-6.**

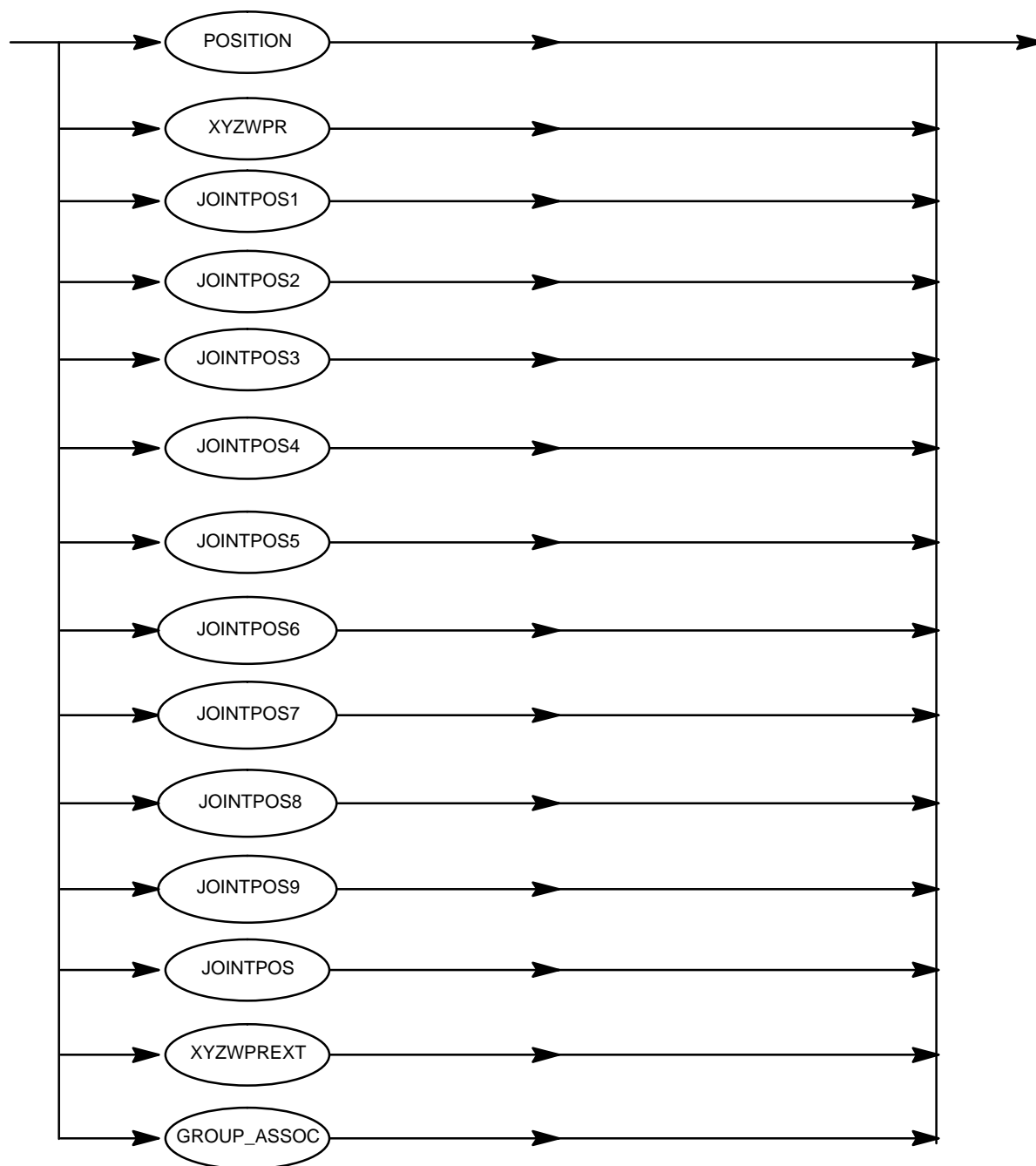
data type continued





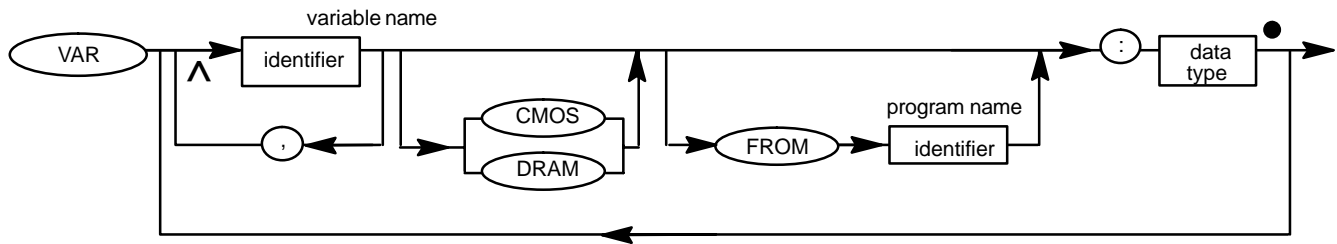
**Figure E-7.**

position type



**Figure E-8.**

VAR -- variable declaration



ROUTINE -- routine declaration

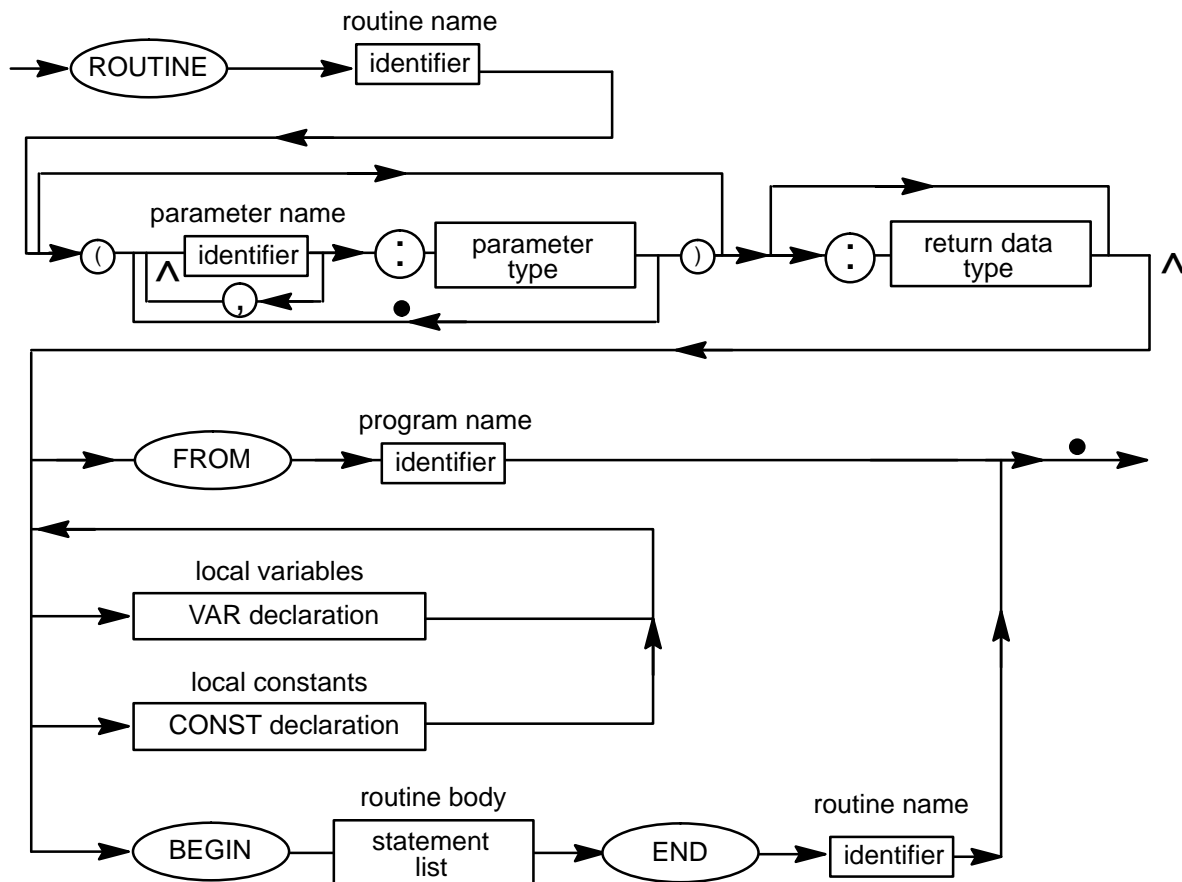


Figure E-9.

return data type

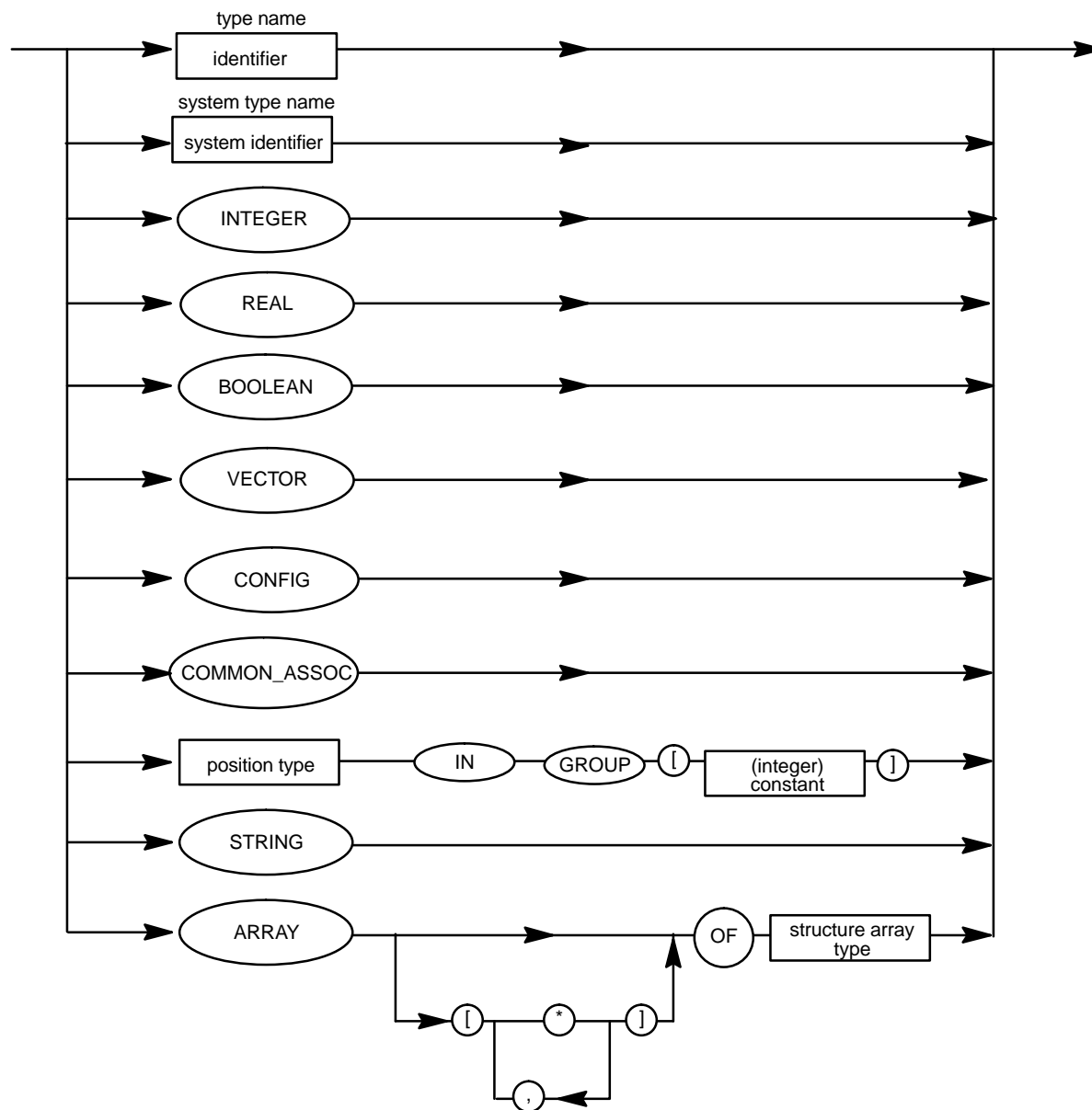
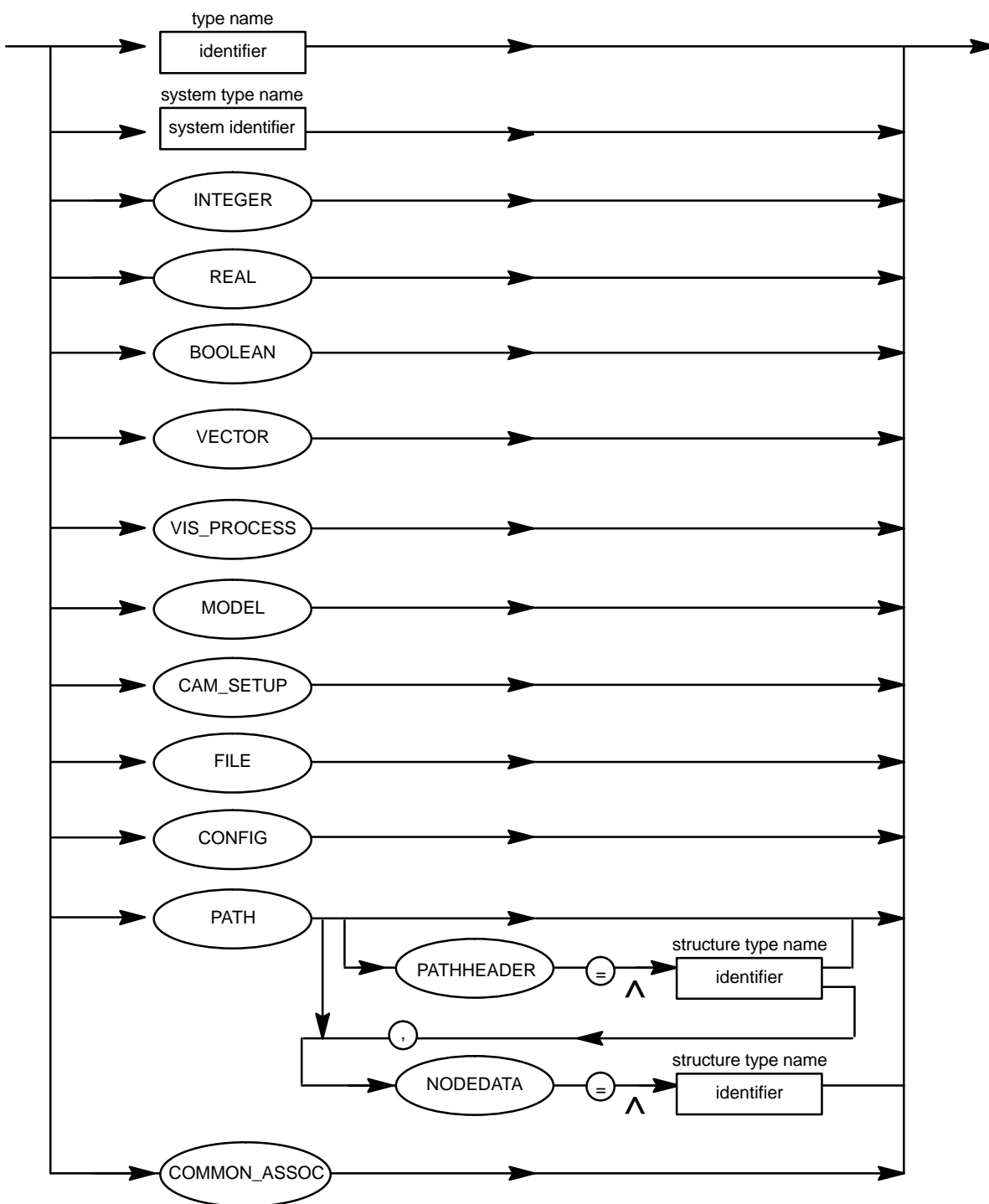


Figure E-10.

parameter type



**Figure E-11.**

parameter type continued

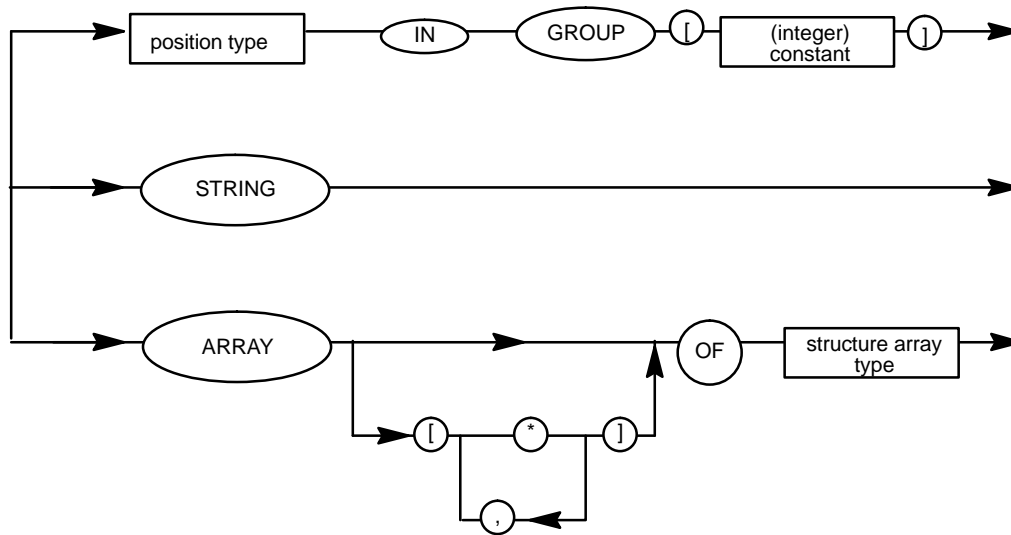


Figure E-12.

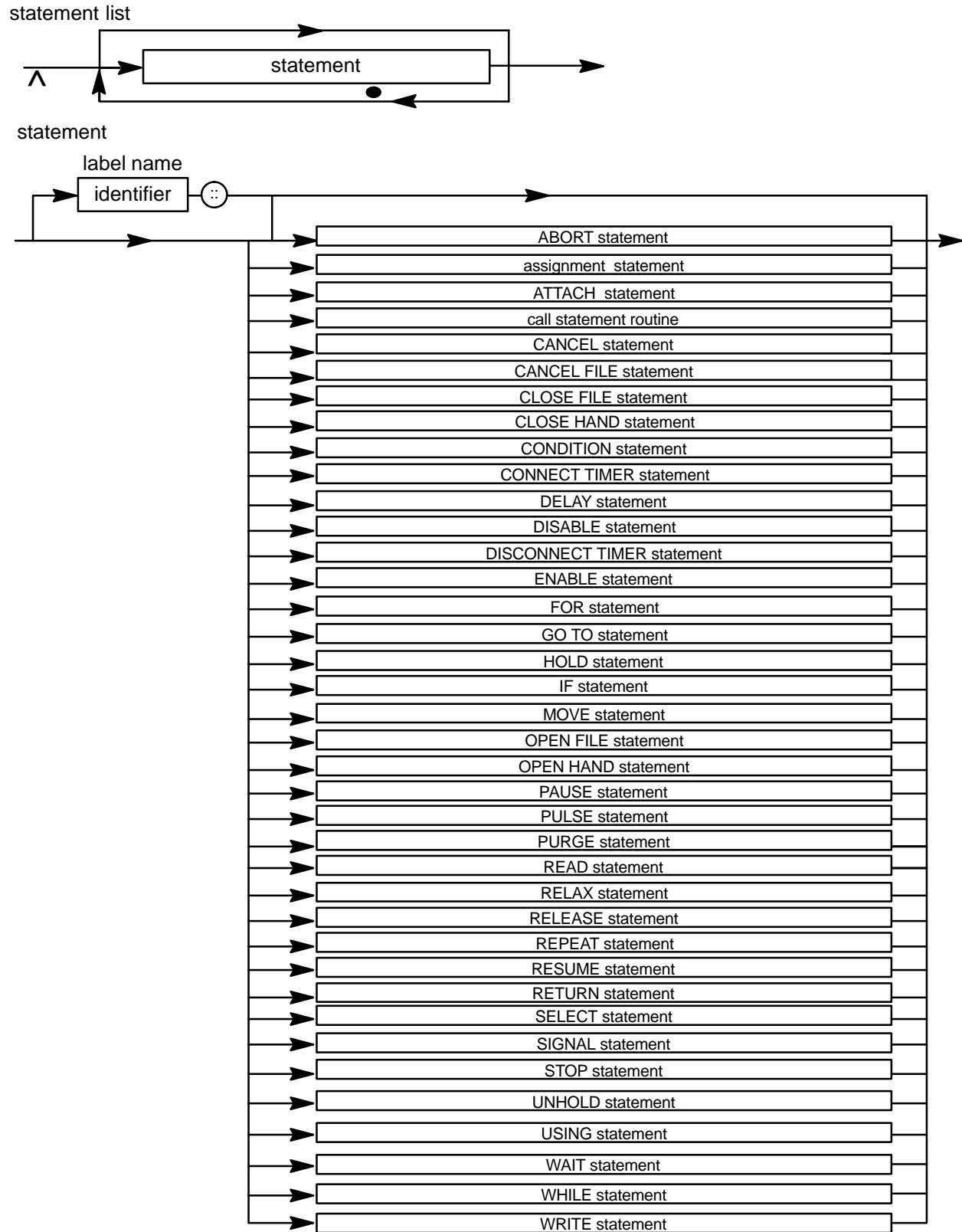


Figure E-13.

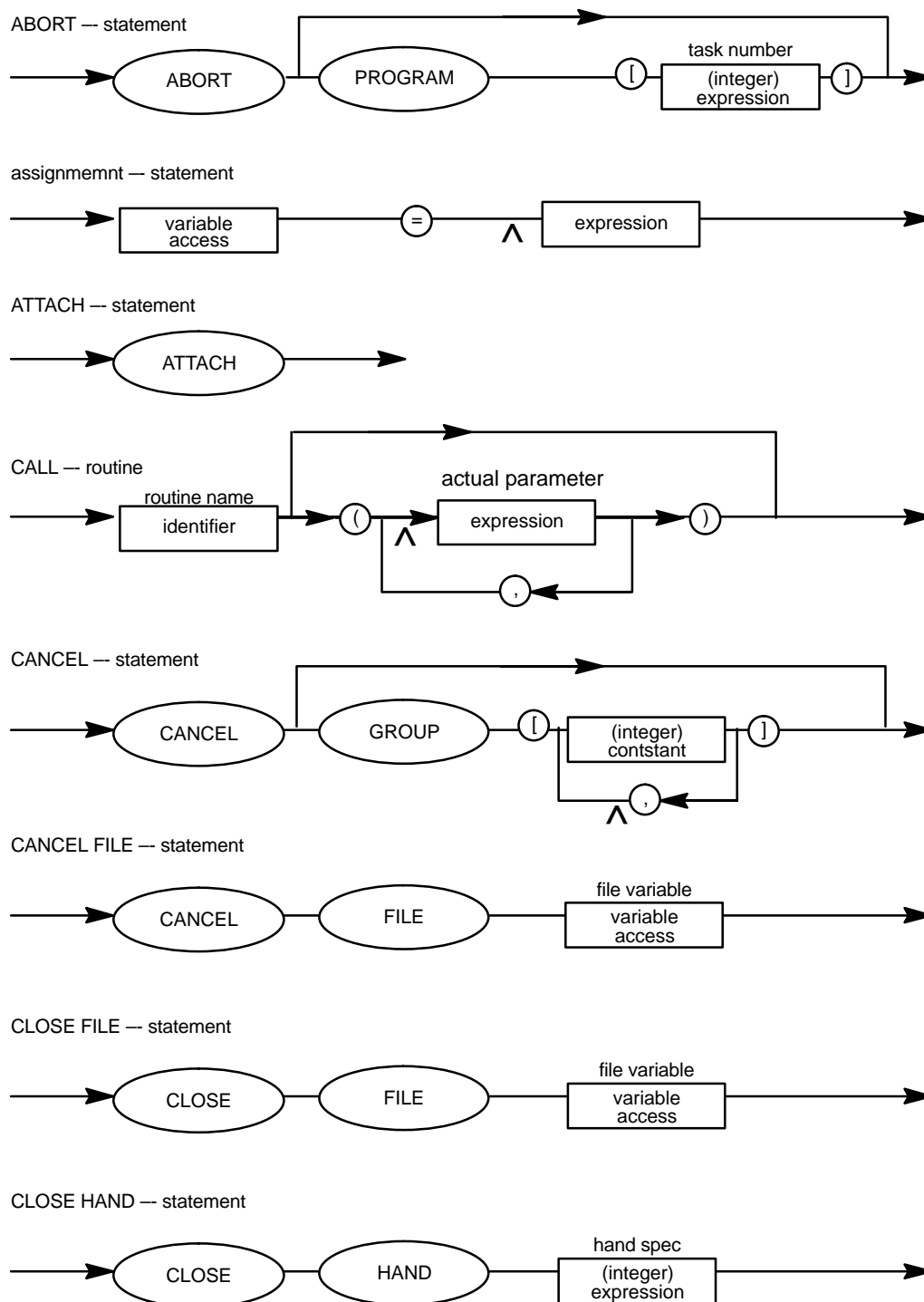
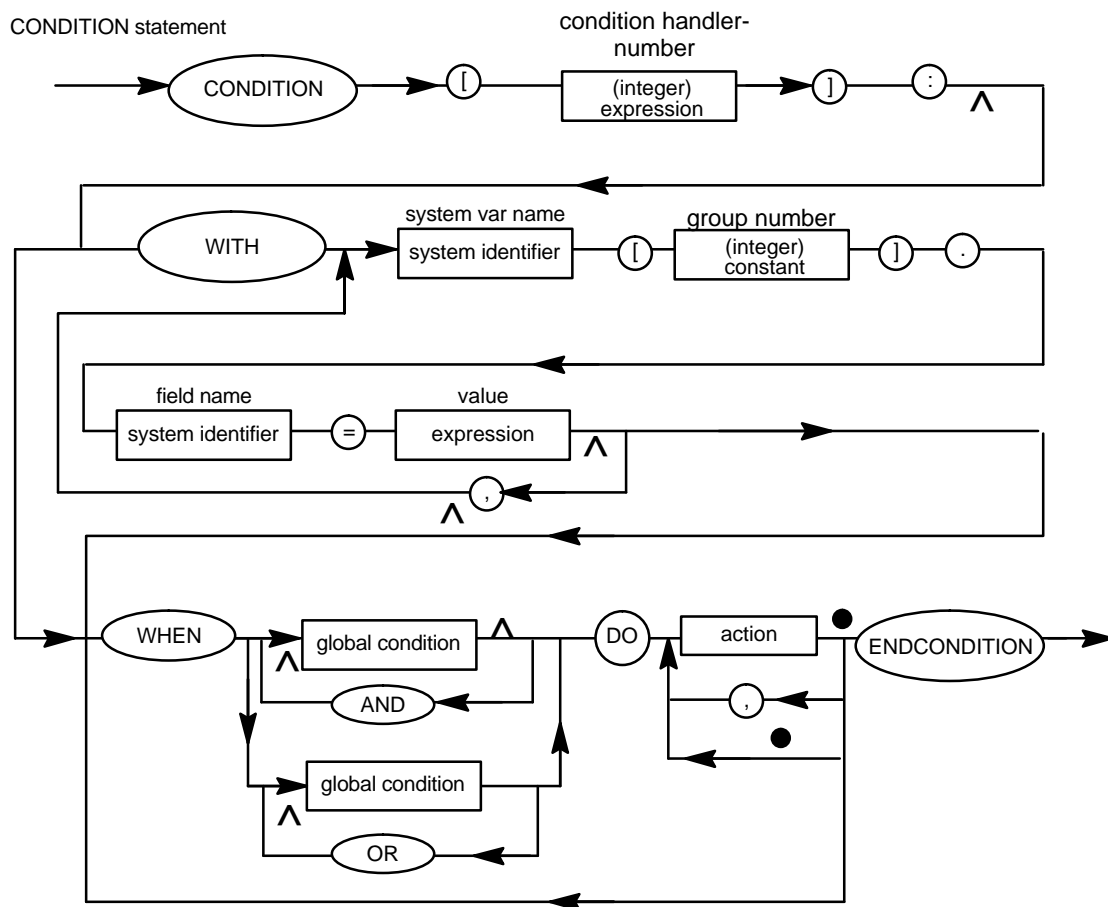
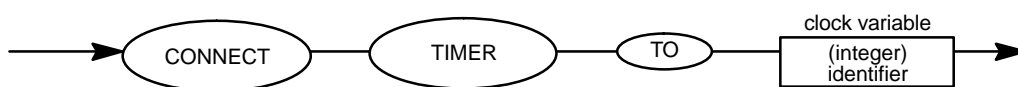


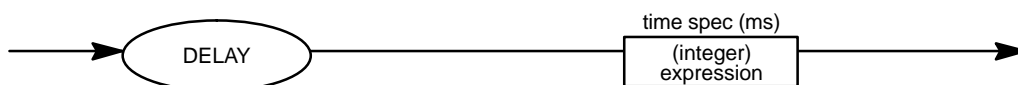
Figure E-14.



CONNECT TIMER — statement



DELAY — statement



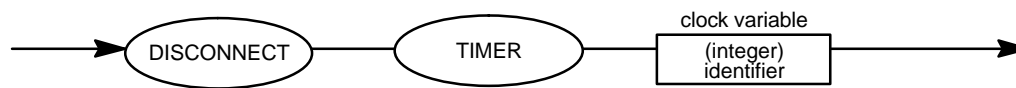
DISABLE — statement





**Figure E-15.**

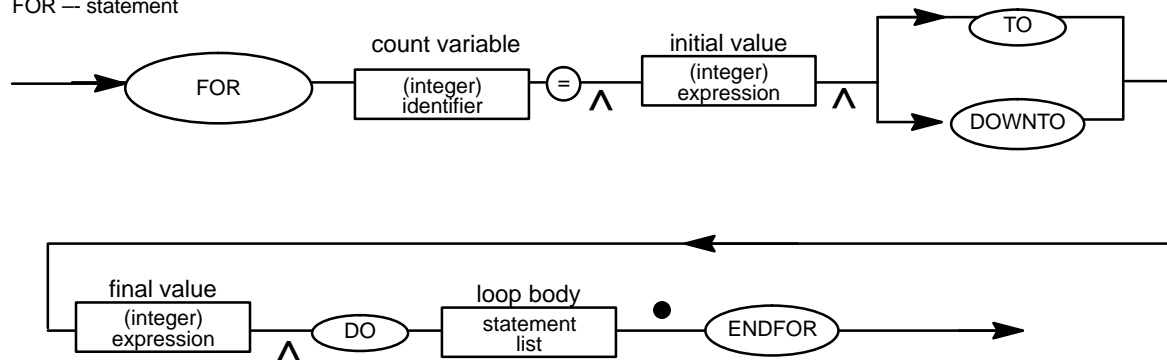
DISCONNECT TIMER — statement



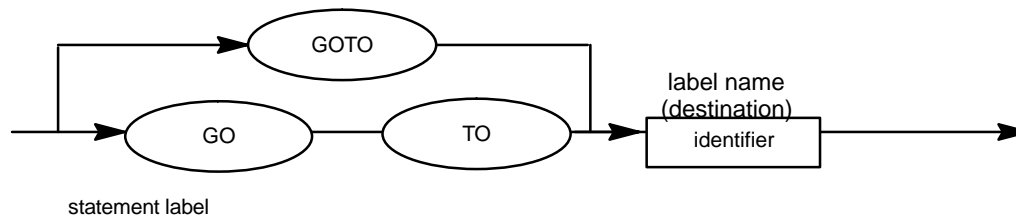
ENABLE — statement



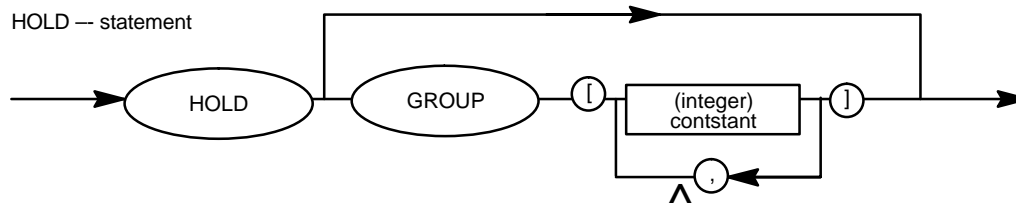
FOR — statement



GO TO — statement



HOLD — statement



IF THEN — statement

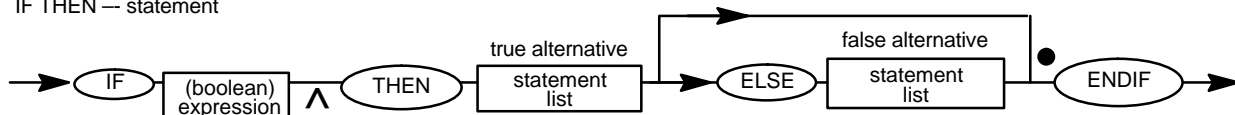


Figure E-16.

MOVE — statement

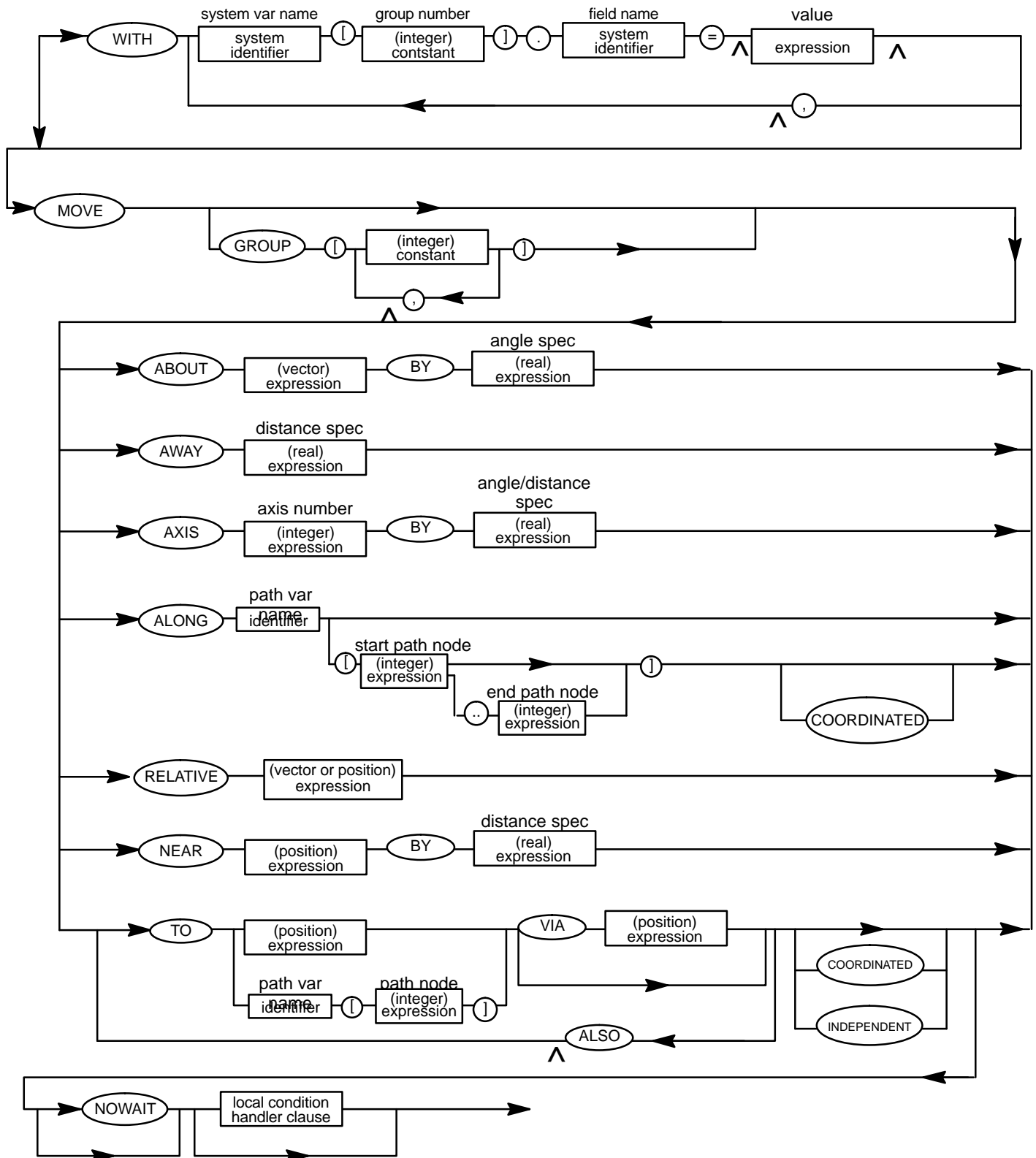
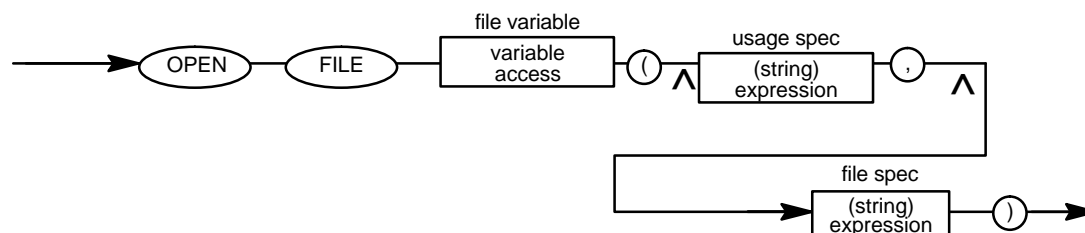
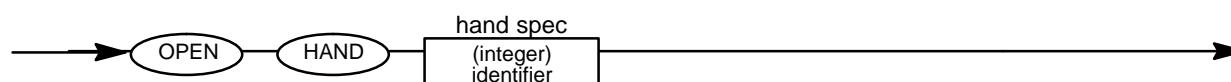


Figure E-17.

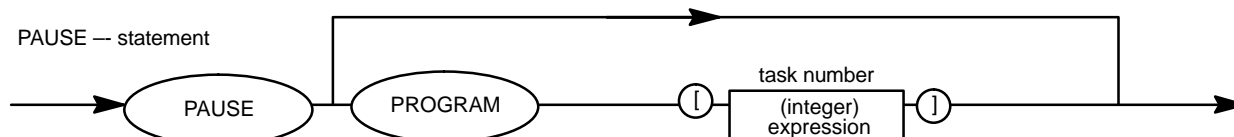
OPEN FILE — statement



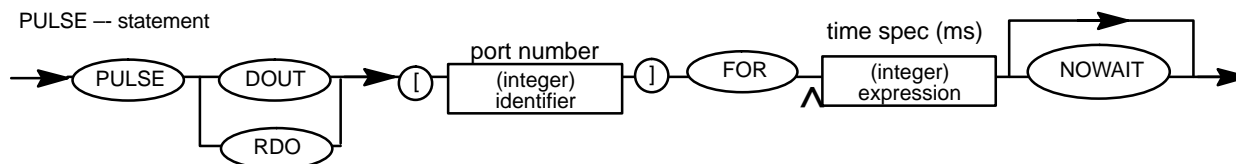
OPEN HAND — statement



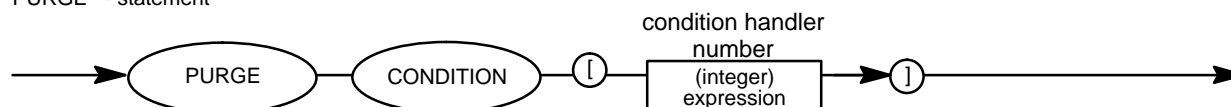
PAUSE — statement



PULSE — statement

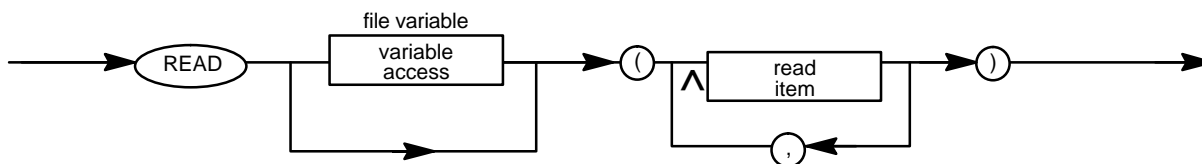


PURGE — statement

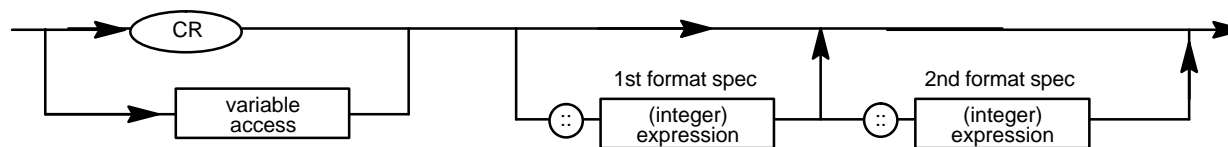


**Figure E-18.**

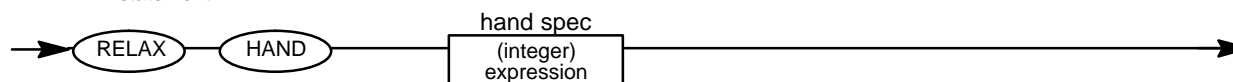
READ — statement



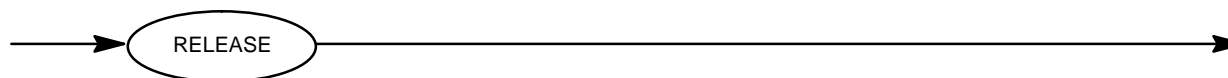
read item — statement



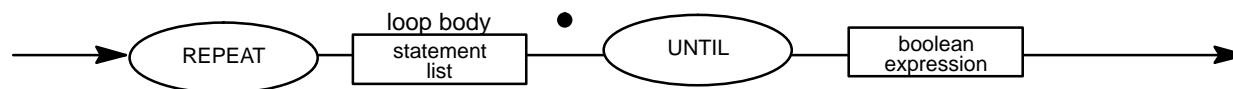
RELAX — statement



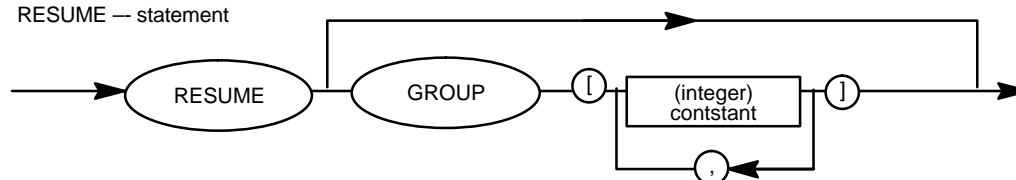
RELEASE — statement



REPEAT — statement

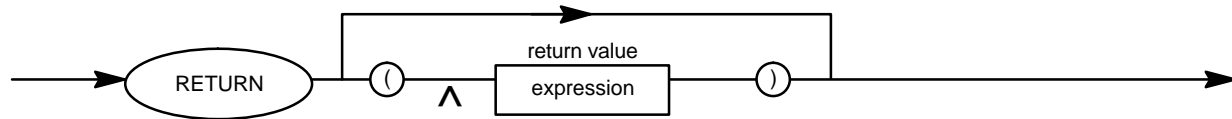


RESUME — statement

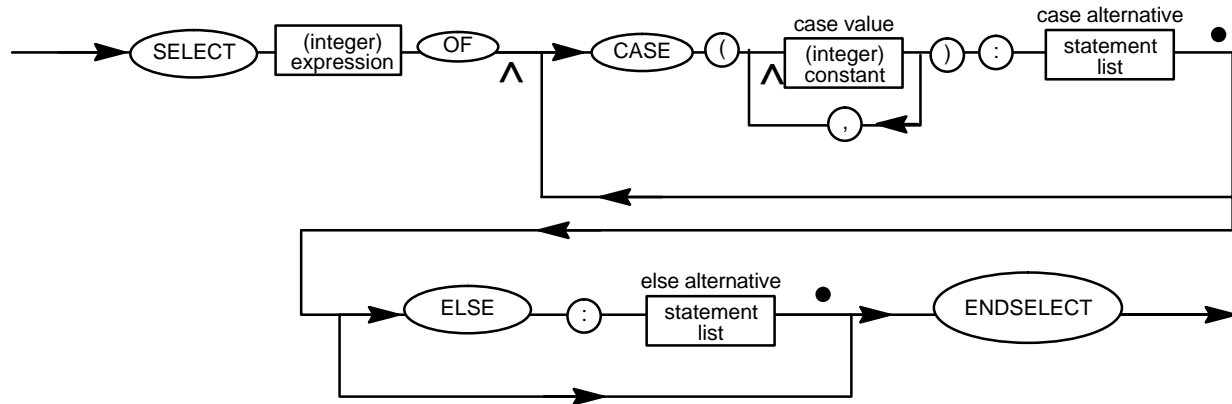


**Figure E-19.**

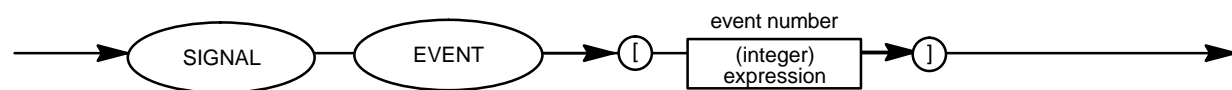
RETURN — statement



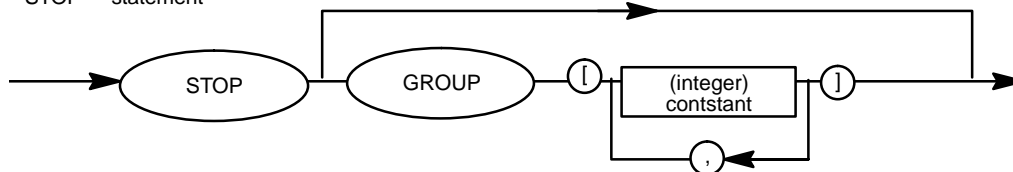
SELECT — statement



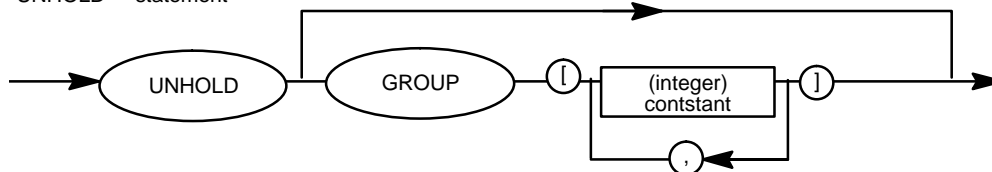
SIGNAL — statement



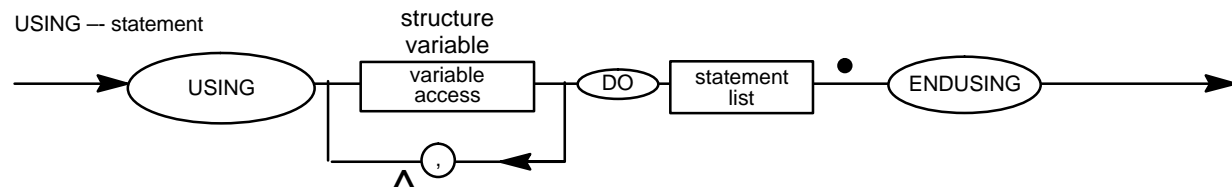
STOP — statement



UNHOLD — statement

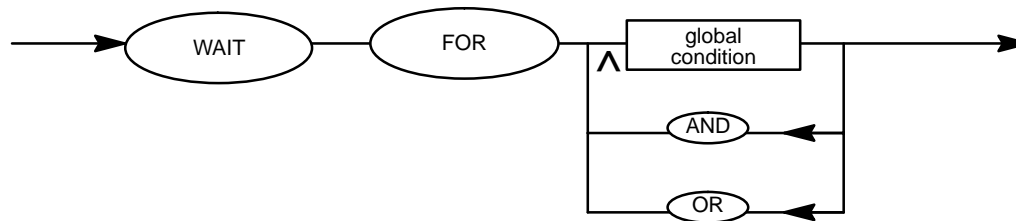


USING — statement



**Figure E-20.**

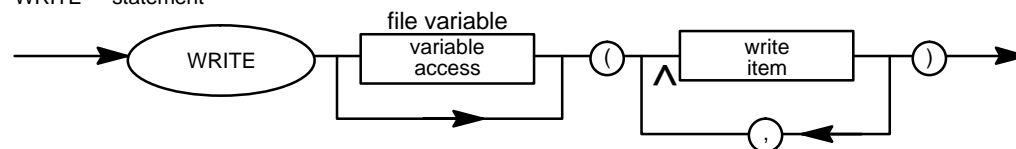
## WAIT -- statement



## WHILE — statement



WRITE — statement



write item

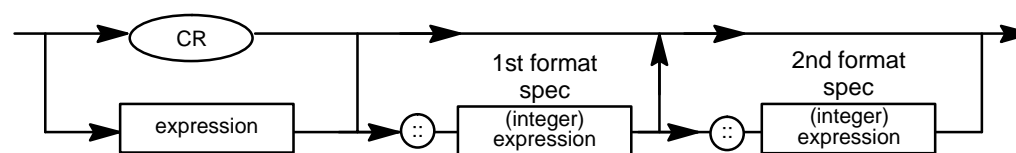
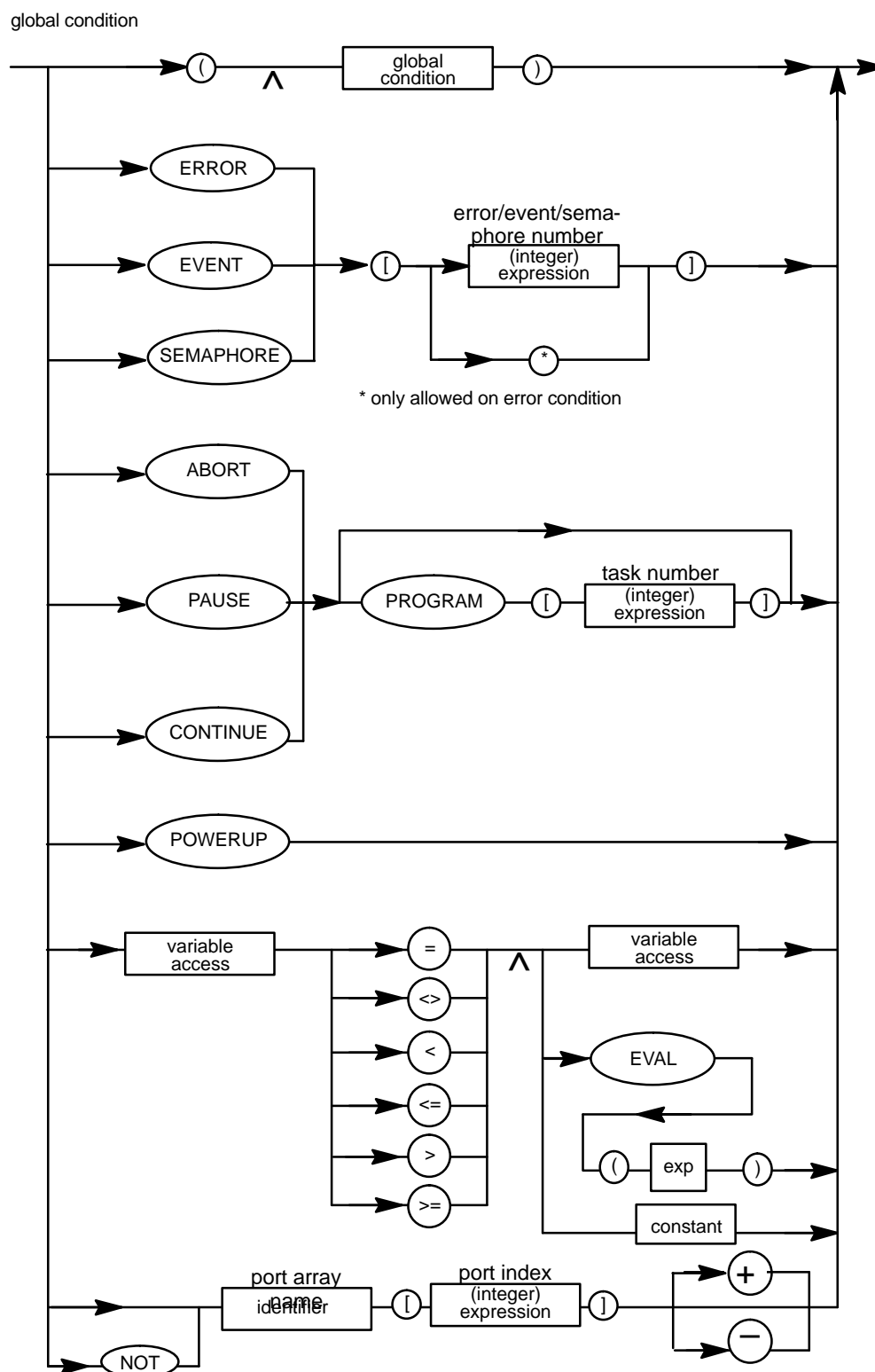
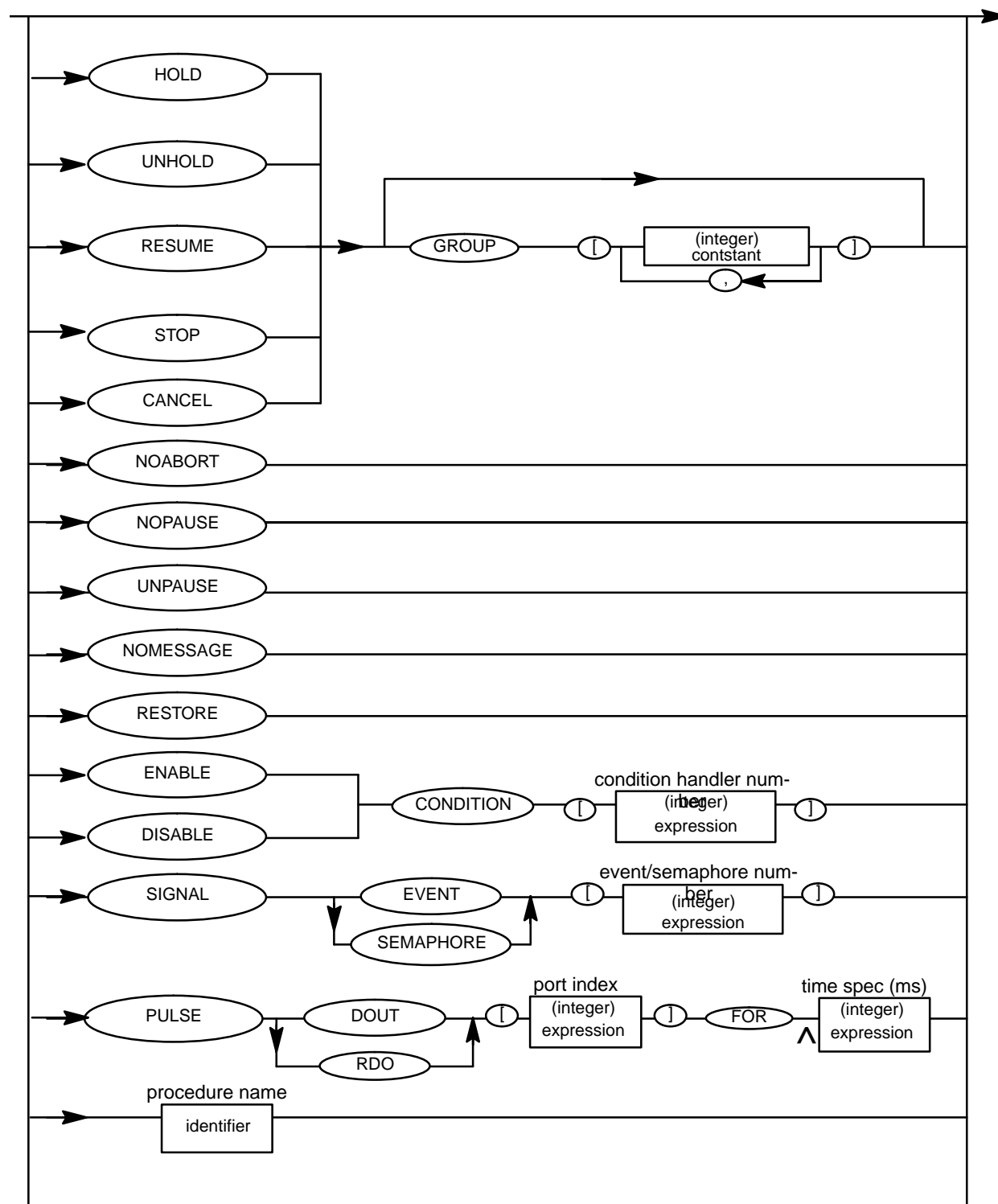


Figure E-21.



**Figure E-22.**

condition handler action





**Figure E-23.**

condition handler action continued

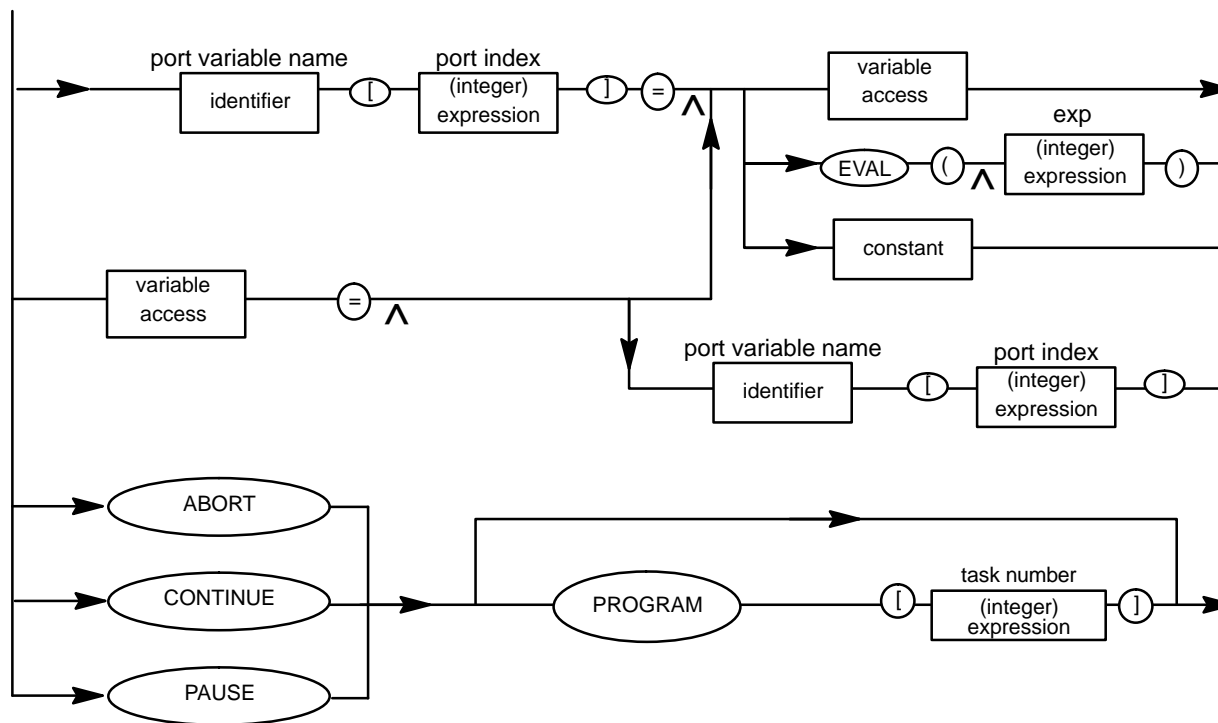
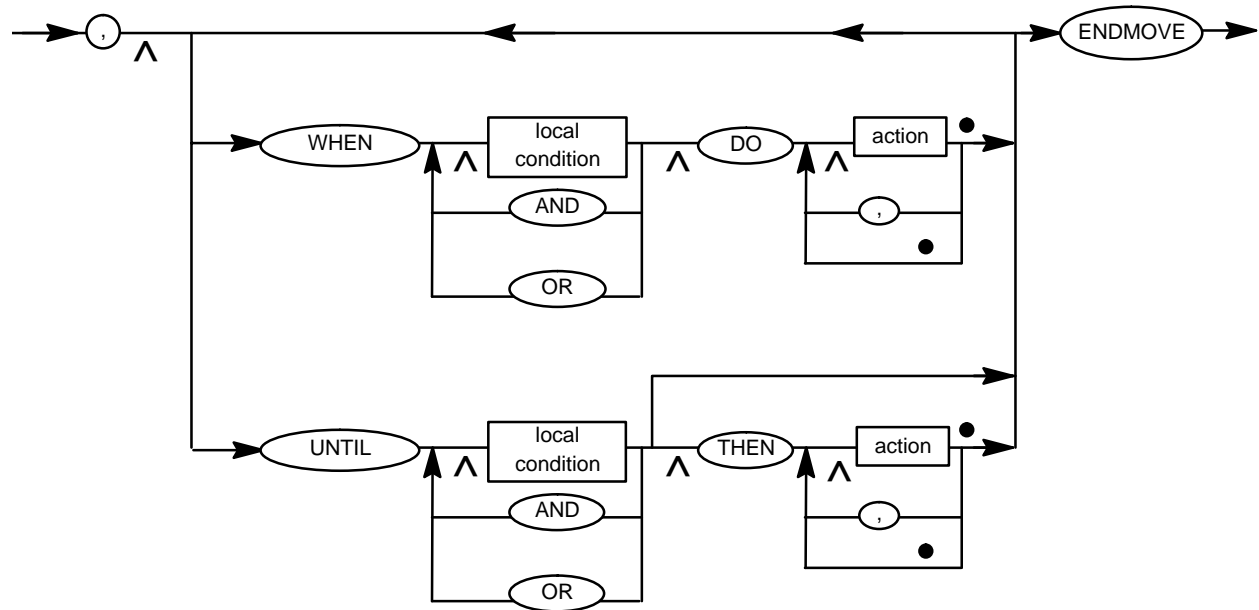


Figure E-24.

local condition handler clause



local condition

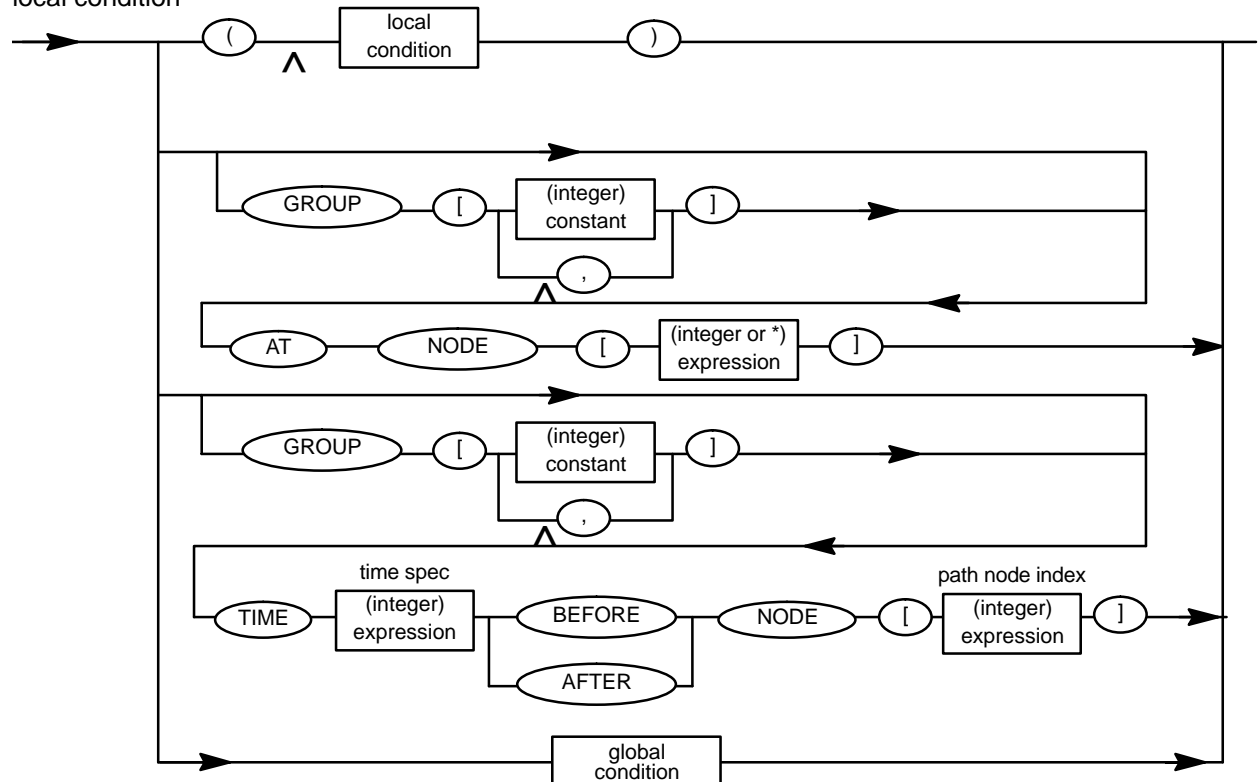
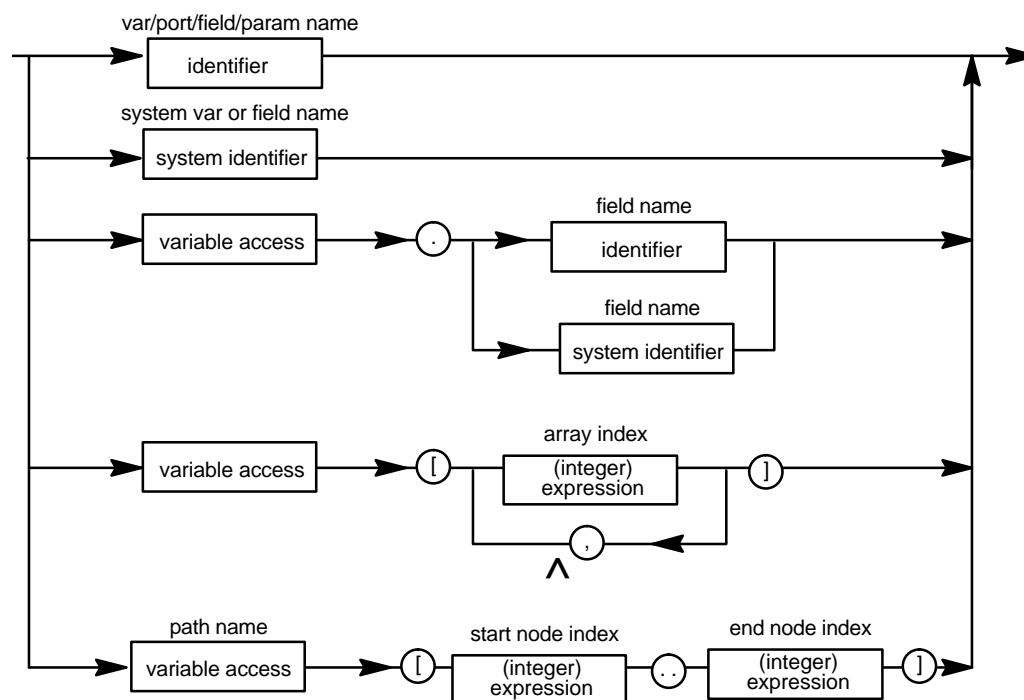
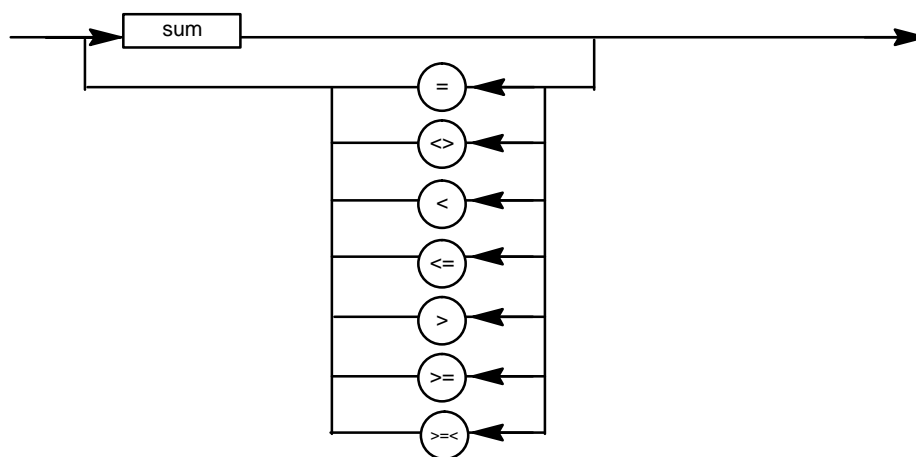


Figure E-25.

variable access

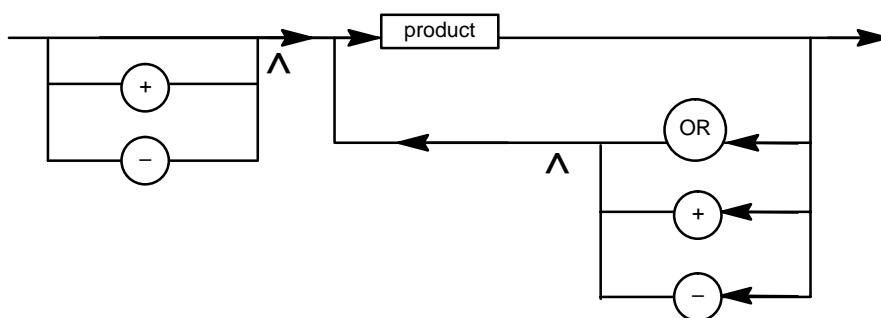


expression

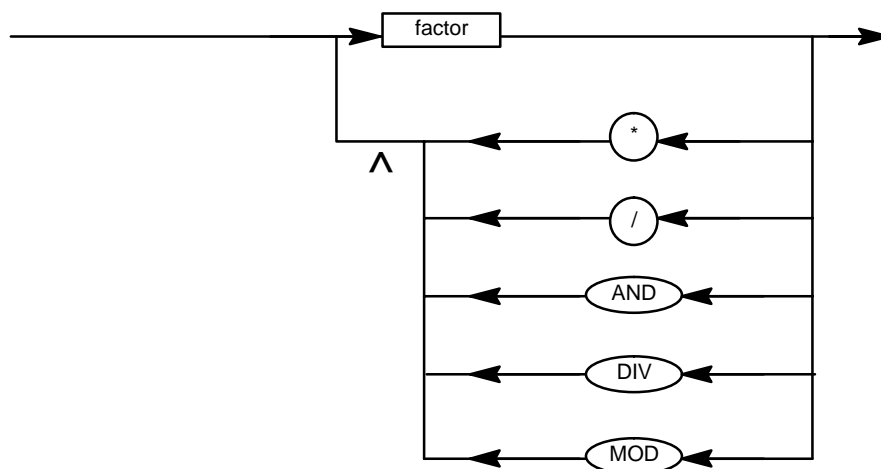


**Figure E-26.**

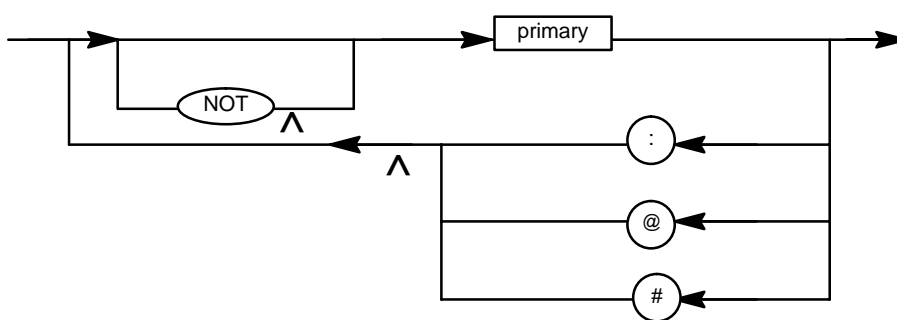
sum



product

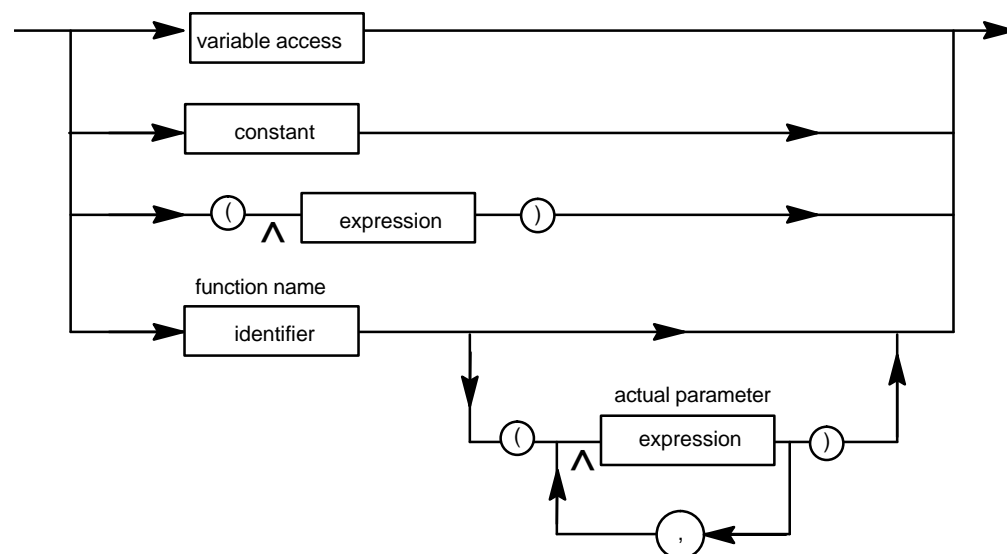


factor



**Figure E-27.**

primary



literal

