"Turing's paper… contains, in essence, the invention of the modern computer and some of the programming techniques that accompanied it." (Minsky (1967)

Chapter 54 – Backus-Naur form

In-text questions

Q1: Peter and Anna may or may not be married to each other

There may be one or several salesmen involved

Either "Close one eye and look at that dog" or "Look at that dog who has only one eye"

Q2: <positive integer> ::= <non-zero digit> | <positive integer><digit>

< digit> ::= <non-zero digit>|<0>

<non-zero digit> ::= 1|2|3|4|5|6|7|8|9

Q3: <real number> ::= <integer> <point> <integer>

< integer> ::= <digit> | digit> <integer>

<digit> ::= 1|2|3|4|5|6|7|8|9|0

<point> := .

Q4: <expression> ::= <term>|<expression> + <term>|<expression> - <term>
<term> ::= <variable>|<term> * <variable>|<term> / <variable>
<variable> ::= a|b|c|d

(a) a – b

a is a <variable> and is therefore a <term>, and hence an <expression>

b is a <variable> and is therefore a <term>

a - b is <expression> - <term> and therefore an <expression>.

(b) a + b * c

b is a <variable> and is therefore a <term>

c is a <variable>

b * c is <term> * <variable> and therefore a <term>

a is <variable> and is therefore a <term>, and hence an <expression>

a + b * c is <expression> + <term> and therefore an <expression>

(c) As above, a * b is a term, c is a <variable>, therefore

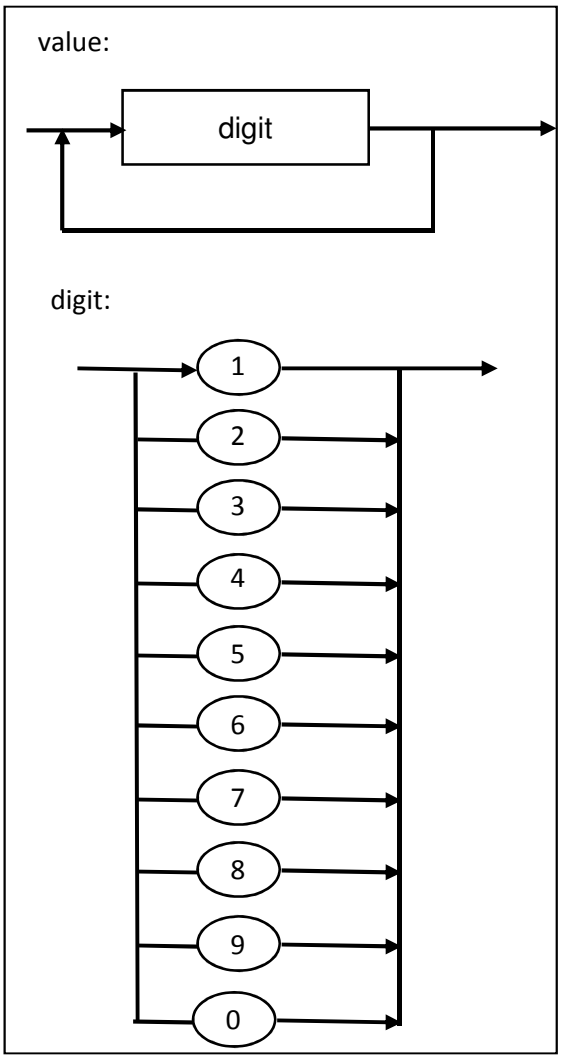
a * b * c is< term> * <variable> which is a <term> and therefore an <expression>.

Exercises

1. (a) 4686: 4 is a <digit>, therefore a <value>.
Applying the recursive rule for <value>., 4686 is a <value>.
7 + 8: <digit> + <digit>, therefore 7 + 8 is a <sum>.
05 + 170: 0 is a <digit>, therefore a <value>
Applying the recursive rule for <value>, 05 is a <value>
Applying the same rules, 170 is a <value>, therefore 05 + 170 is a <sum>

(b) <hex> ::= <digit>|<letter>|<hex><digit>|<hex><letter>
<digit> ::= 1|2|3|4|5|6|7|8|9|0
<letter> ::= A|B|C|D|E|F

(c)



2. (a) aea is valid, since e is a vowel, and aea is a<vowel>a therefore a vowel string.

uuu is valid, since u is a vowel, and uuu is u<vowel>u therefore a vowel string.

A and E are undefined therefore AEA is not a valid vowel string

aeae is not a valid vowel string as it contains four vowels and only three are allowed

oooaeio is invalid since although a,e,i,o are all in <vowel>, ooaeio is not a valid vowel-string as it contains more than three vowels.

(b) <vowel-string> ::= <vowel>| double-vowel |a<vowel-string>a |e< vowel-string >e | i<vowel-string>i | o<vowel-string >o | u< vowel-string >u

vowel ::= a|e|i|o|u

<double-vowel> ::= aa|ee|ii|oo|uu

3. **AC 234** is valid, since

AC is a valid code,

234 is <pos digit><digit><digit> and therefore a valid number,

AC 234 is <code><' '><number> therefore a valid reg-no.

AB 13 is not valid since

AB is not a valid code therefore this is not a valid reg-no.

AX 099 is not valid since 099 is not a valid number (does not start with pos digit.)

BB 2345 is valid since

BB is a valid code

2340 is <pos digit><digit><digit><digit> and therefore a valid number,

BB 2340 is <code><' '><number> therefore a valid reg-no.

AX 6 is not valid since 6 is not a valid number.