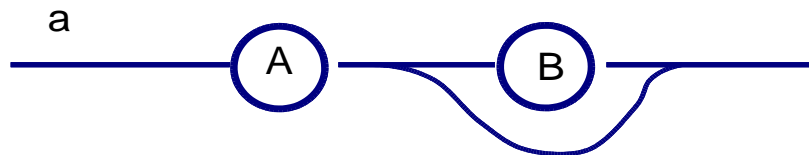# Syntax Diagrams

They are also sometimes called "Railroad Tracks" and are a graphical representation of EBNF rules. They are seldom seen anymore as EBNF is much more compact.

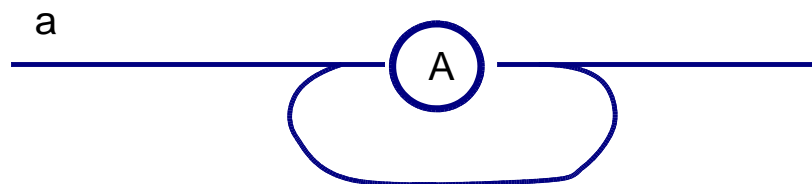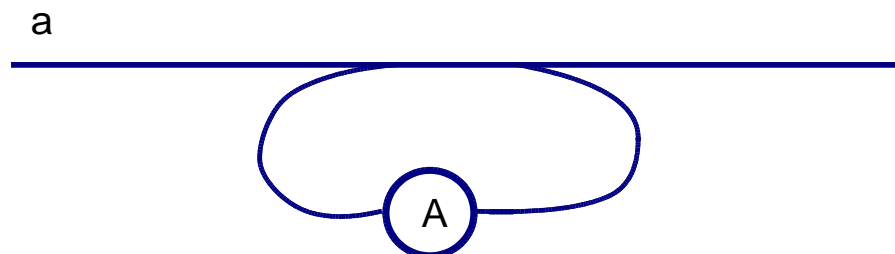The diagrams corresponding to the various EBNF rules above are:

1.   <a> ::= A [B]

a

A ⎯ B

2.   <snum> ::= [-] <num>

snum

- ⎯ num

3.   <a> ::= A { A }

a

A

4.   <a> ::= { A }

a

A

5.   <block> ::= begin <cmd> { ; <cmd> } end

block

begin → cmd → end

;
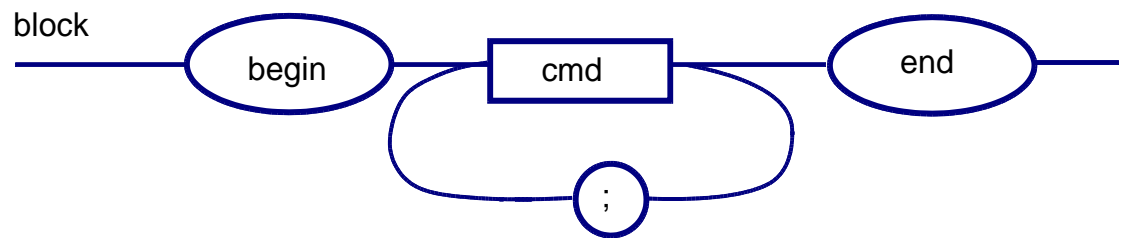
6.   <snum> ::= [(+ | -)] <num>

snum

+

-

num
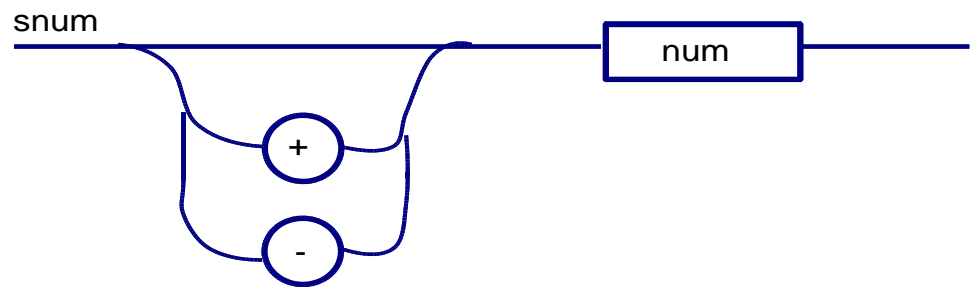
It can be helpful to imagine train tracks, to help in drawing them correctly:

- Control structures (curves and switches) should be very clear:
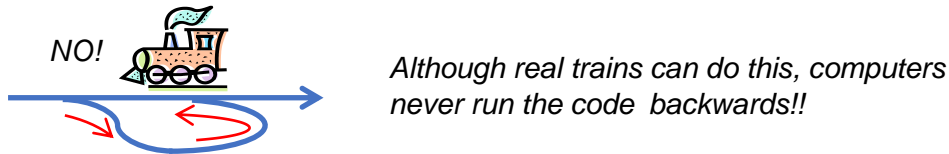

OK	*OK*	*NOT LIKE THIS!* *(intention must be clear)*
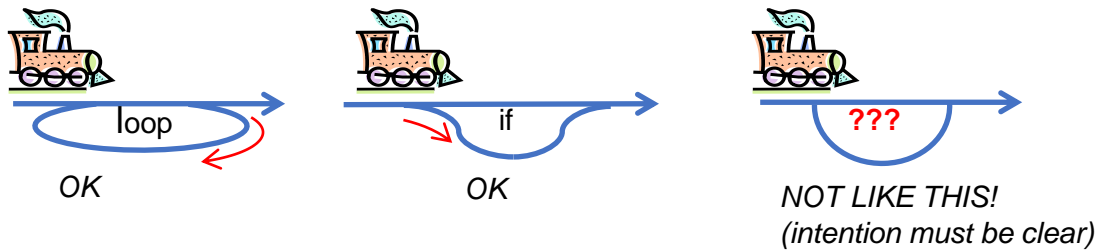
- The train must *never* "reverse directions":

Although real trains can do this, computers never run the code backwards!!

There are some common structures in programming languages.
Here is the correct way to draw them in BNF, EBNF, and Syntax Diagrams:

	BNF	EBNF	Syntax Diagram
A is optional	M ::= xxAxx | xxxx	M : : = xx[A]xx	
A is required	M ::= xxAxx	M ::= xxAxx	
1 or more of A	M ::= MA | A	M ::= A { A }	
0 or more of A	M ::= MA | ϵ	M ::= { A }	
1 or more of A with separators	M ::= M ; A | A	M ::= A { ; A }	
1 or more of A with terminators	M ::= MA; | A;	M ::= A ; { A ; }	
0 or more of A with separators	M ::= H | ϵ H ::= H ; A | A	M ::= [A { ; A }]	
0 or more of A with terminators	M ::= MA; | ϵ	M ::= { A ; }	