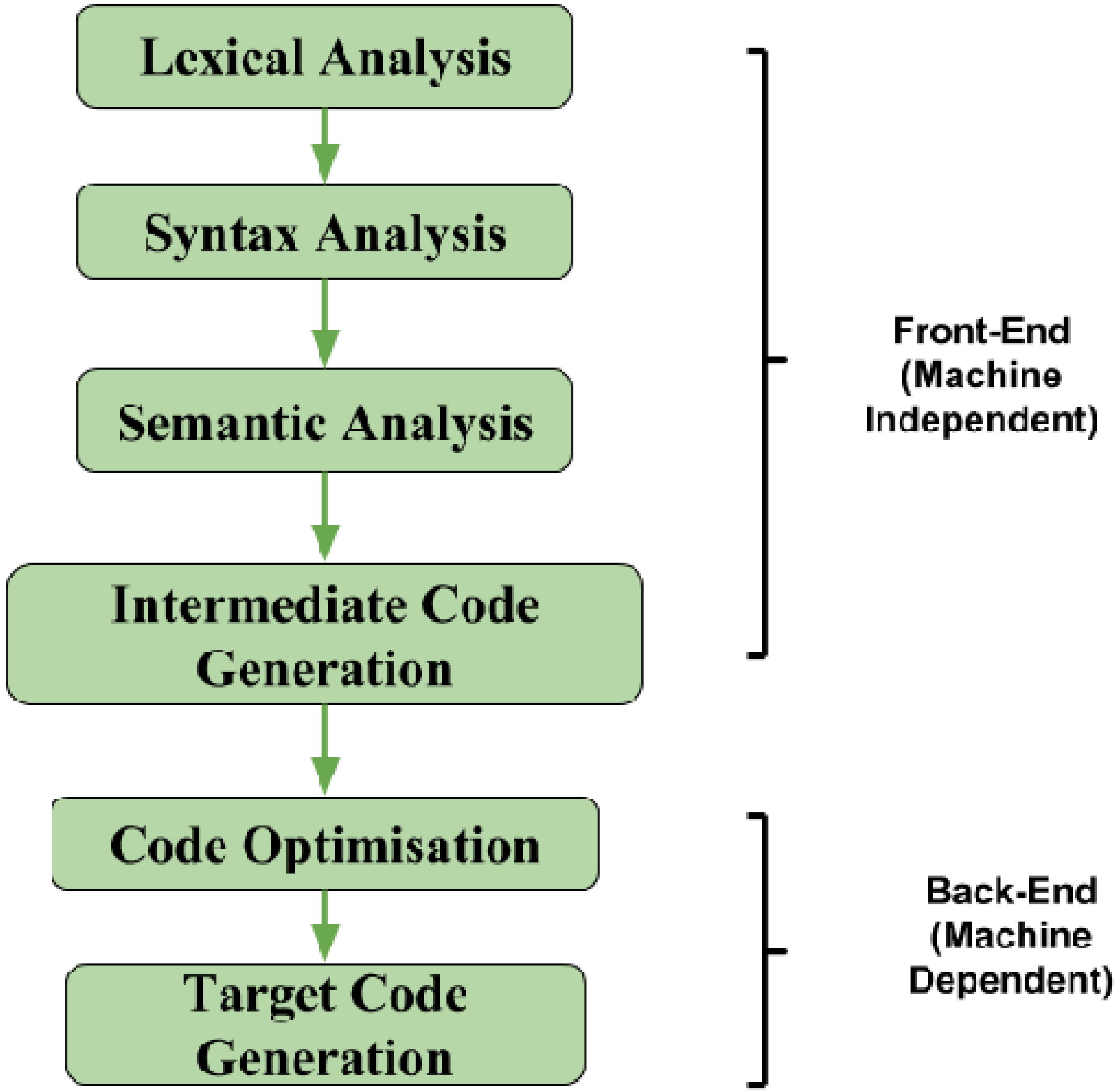
Syntax Directed Translation



Lexical Analysis

Syntax Analysis

Semantic Analysis

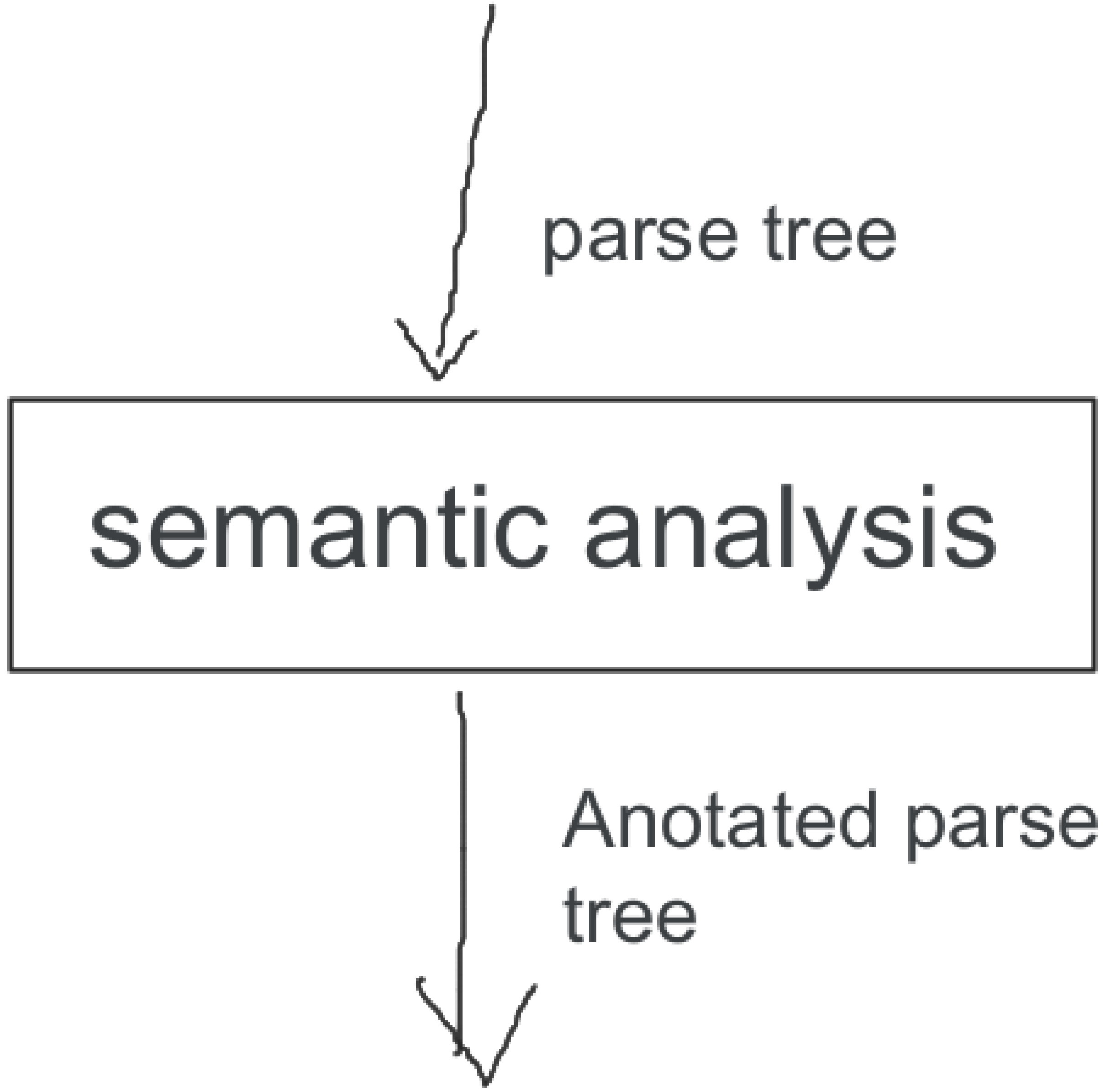
Intermediate Code Generation

Code Optimisation

Target Code Generation

Front-End (Machine Independent)

Back-End (Machine Dependent)



parse tree

semantic analysis

Anotated parse tree

SDT is used for Executing Arithmetic Expression. 5#2&1*2=?

In the conversion from infix to postfix expression.

In the conversion from infix to prefix expression.

It is also used for Binary to decimal conversion.

In counting number of Reduction.

In creating a Syntax tree.

SDT is used to generate intermediate code.

In storing information into symbol table.

SDT is commonly used for type checking also.

SDT Basics

Production Rules/ Grammer:	Semantic Rules:
E --> E + T	E.val: = E. val + T.val
E --> T	E.val: = T.val
T --> T * F	T.val: = T.val + F.val
T --> F	T.val: = F.val
F --> num	F.val: = num.lexval

in SDT every non terminal can have 0 or more attributes
A semantic rule contains values that can be string, numbers or memory location

Grammer + Semantic Rules = SDT

Example: 01

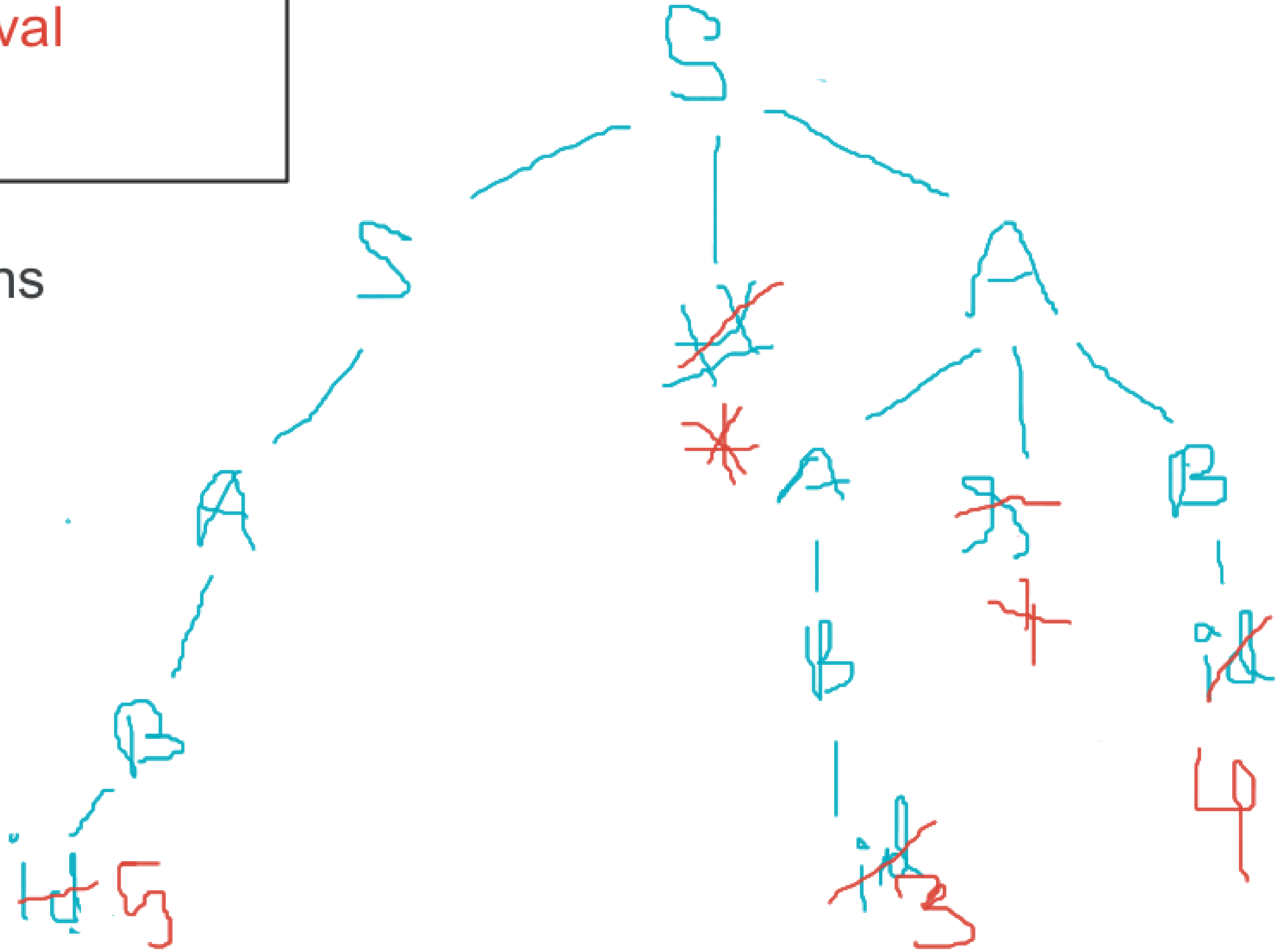
Grammer	Rules
S --> S # A / A	S.val:= S.val * A.val
A --> A & B / B	A.val:= A.val + B.val
B --> id	B.val:= id.lexval

Given String: 5 # 3 & 4

numerical tokens

5 * 3 + 4 = 19

1. Draw a syntax tree
2. Apply the rules on the tree
3. Evaluate the leaf nodes (left to right)



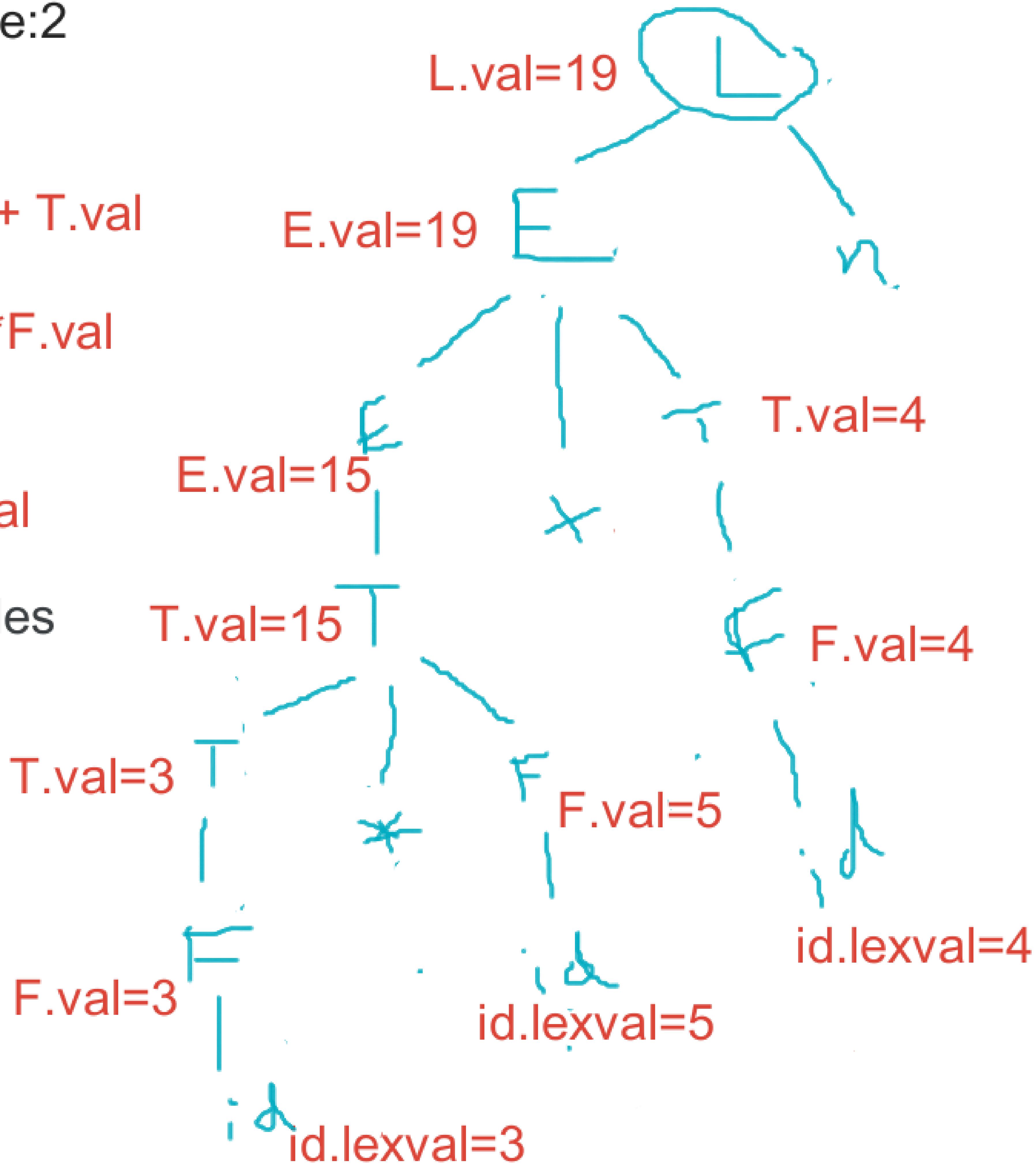
Example:2

Grammer	Semantic Rules
L--> E n	L.val = E.val
E--> E + T	E.val = E.val+ T.val
E --> T	E.val= T.val
T--> T * F	T.val = T.val*F.val
T --> F	T.val=F.val
F--> (E)	F.val=E.val
F --> id	F.val=id.lexval

Input String: 3 * 5 + 4 n

1. Draw anotated parse tree
2. Determine the value of start Symbol

⇒ 19

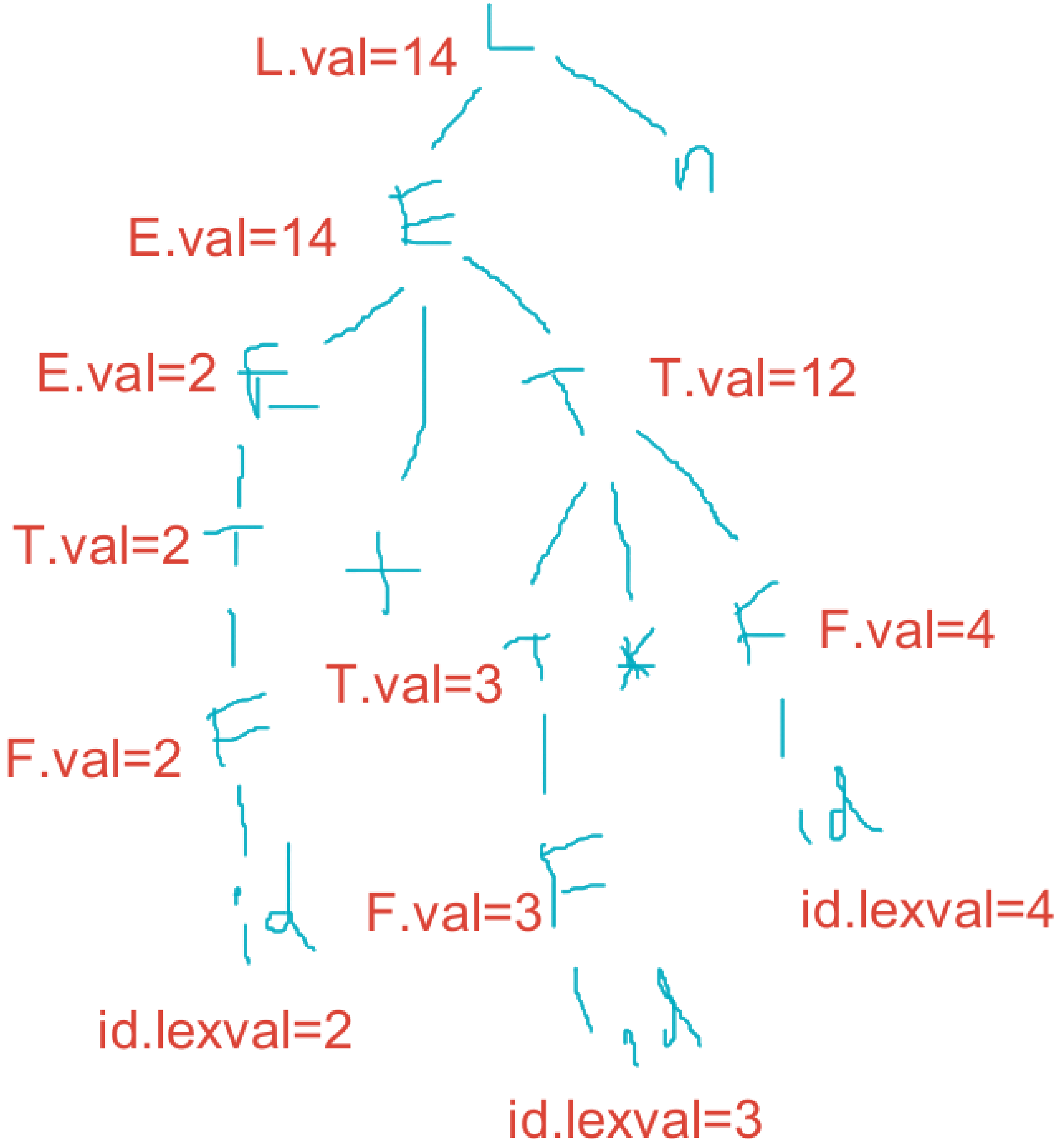


L--> E n	L.val = E.val
E--> E + T	E.val = E.val+ T.val
E --> T	E.val= T.val
T--> T * F	T.val = T.val*F.val
T --> F	T.val=F.val
F--> (E)	F.val=E.val
F --> id	F.val=id.lexval

2+3*4 n,
E.val= 14
T.val= 12

Always take max value

example: 3



L.val=14

E.val=14

E.val=2

T.val=12

T.val=2

T.val=3

F.val=4

F.val=2

F.val=3

id.lexval=4

id.lexval=2

id.lexval=3