Context-Free Grammar

Parsing Process

> **Info**
>
> Build the parse tree from the stream of tokens
>
> Parse tree show the syntactic structure of source
>
> Add information about identifier in symbols table
>
> Report Error

Context-free Grammar

> **Info**
>
> write in quintuple (V,T,P,S)
> V : finite set of Non-terminals, containning S
> T : finite set of Terminals
> P : set of production rules in form $\alpha \longrightarrow \beta$
> $\alpha \subset$ V and $\beta \subset$ (V$\cup$T)* = sentential form
> S : start symbol

Backus-Naur Form(BNF)

Info

Nonterminal are in < >.
Terminal are any other symbol not in < >.
::= means $\longrightarrow$ and | means or

Derivation

Info

A sequence of replacement of a substring in a sentential form

for CFG G one-step : $\Rightarrow_G$ multi-step : $\Rightarrow_G^*$

Left-most : each step replace left-most V
Right-most : each step replace right-most V

A language generated by G, denoted by L(G), is a set of strings derived from G : L(G) = {w|S$\Rightarrow_G^*$w}

Left-Recursive :
if can generate deriavation in the form A$\Rightarrow^*$AX
Right-Recursive :
if can generate deriavation in the form A$\Rightarrow^*$XA

Parse Tree

Info

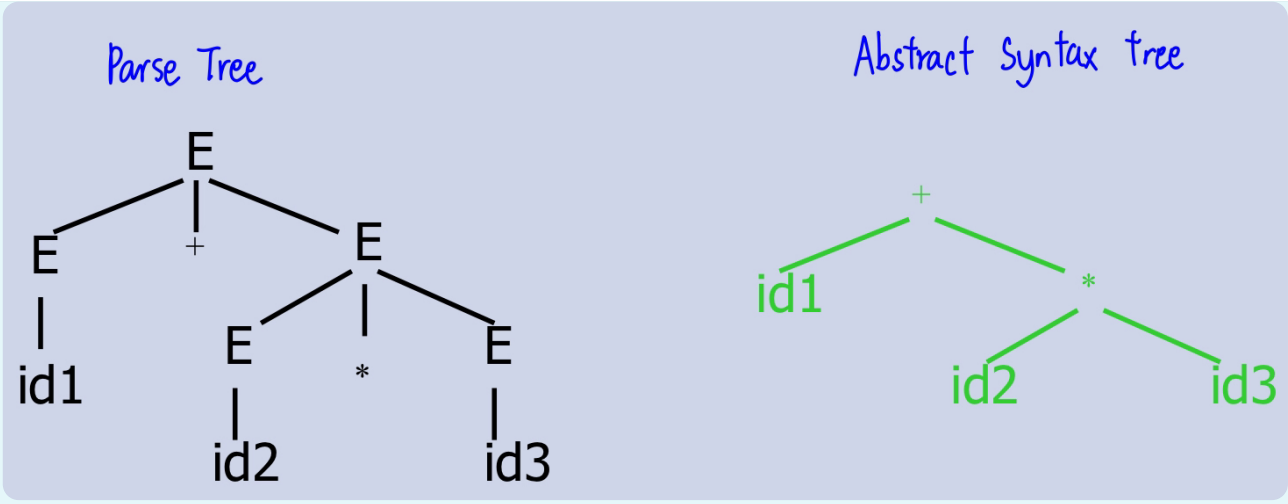
labeled tree in which
1.interior node : V non-terminals
2.leaf node : T terminals
3.child node associated to deriavation

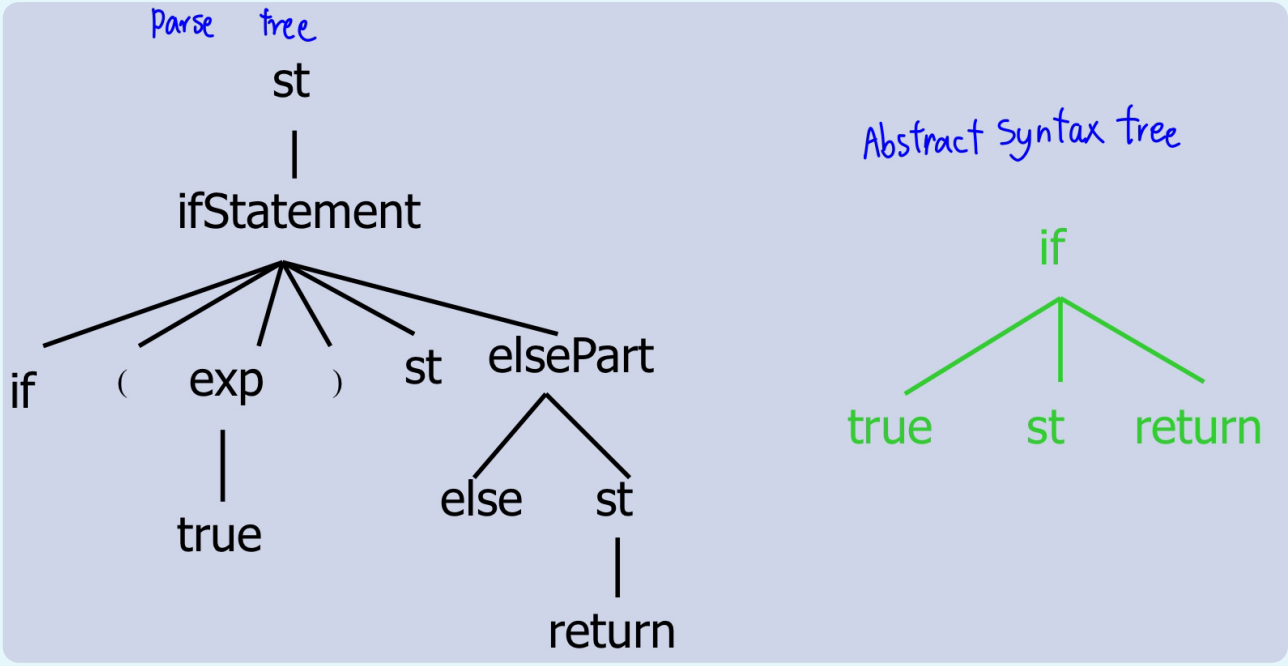
Preorder-numbering:i'th number is i'step,only $\subset$ V
Reverse Postorder numbering:postorder reverse it

Abstract Syntax Tree

Info

Representation of actual source tokens
Interior Node : operators
Leaf Node : operands
Comparation between Parse&Abstract Syntax Tree





Ambiguous Grammar

Info

A grammar is ambiguous if it **can be generate two different parse tree for one string.**

Ambiguous grammar can cause inconsistency in parsing

Which operation is to be done first?

higher precedence done before lower precedence

hp logically further from start symbols

let done time before plus so make it further

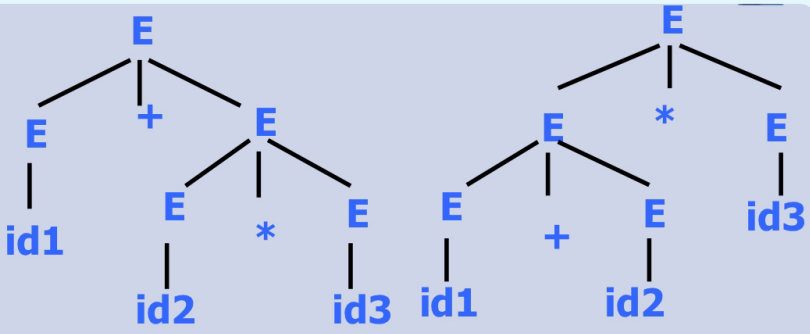
$E \rightarrow E + E$

$E \rightarrow E - E$

$E \rightarrow E * E$

$E \rightarrow E / E$

$E \rightarrow id$



ปรับเปลี่ยน

$E \rightarrow E + E$ ให้ตามลำดับ ดูณ/หาร > บวก/ลบ

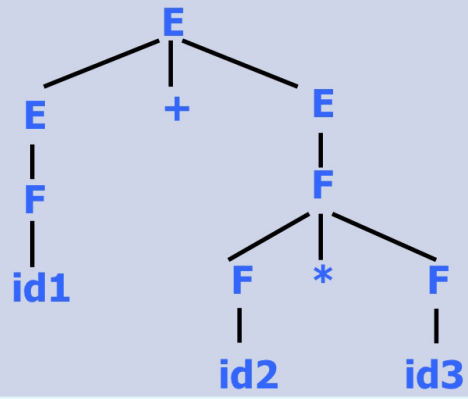
$E \rightarrow E - E$

$E \rightarrow F$: ให้ดูณ/หารไปใกล้กว่า +,-

$F \rightarrow F * F$

$F \rightarrow F / F$

$F \rightarrow id$



To avoid Ambiguous we use associativity to help

Right-associated : 2 operators do right first

W + (X + (Y + Z))

Left-associated : 2 operators do left first

((W + X) + Y) + Z

Associativity

- Left-associative operators

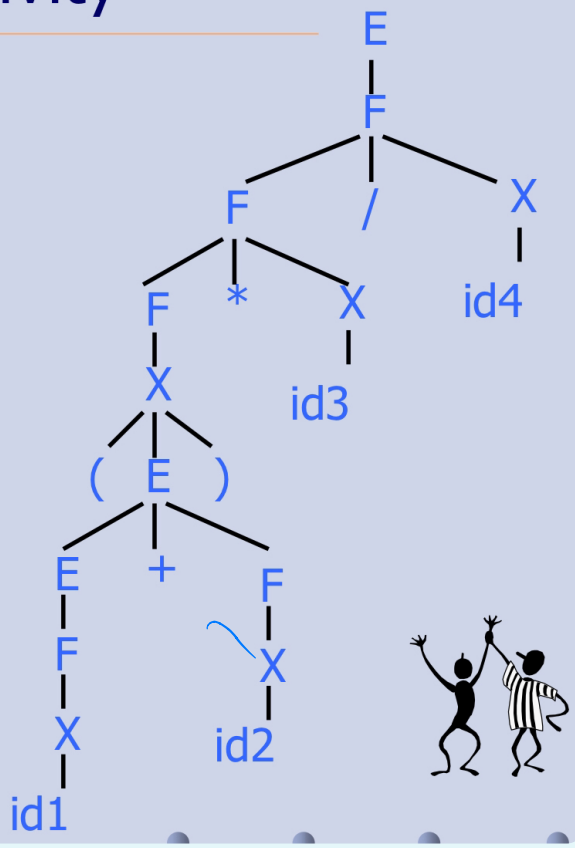
E → E + F | E - F | F

F → F * X | F / X | X

X → (E) | id

(id1 + id2) * id3 / id4

= (((id1 + id2) * id3) / id4)



Extended Backus-Naur Form

Info

Kleene's star/closure

St{; St} = St St*

{St;}St = St*St

Optional Part inside [-] have not is ok

IfSt ::= if(E)St[else St] , มี else หรือไม่ก็ได้

```
E ::= F[+E]|F[-E]
```

- Graphical representation of EBNF rules
  - nonterminals: IfSt
  - terminals: id
  - sequences and choices:
- Examples
  - X ::= (E) | id
  - Seq ::= {St ;} St

st;st;

  - E ::= F [+ E]