

$f(g()) \times$

$f(\underbrace{(x+1)}_{\text{atom}}, \underbrace{(y+2)}_{\text{atom}})$

How to fix?

Read Parsing  
chapter in  
Textbook.

How does calls & indexing fit into  
the grammar?

$\underbrace{- \text{fact}(20)}_{\text{fact}(20)}$

Highest precedence, unary negation  
↓  
lowest

( )	[ ]
*	/
+	-

↳ Pointer dereference

\*f(x)

Is it dereferencing the returned pointer  
or the function pointer?

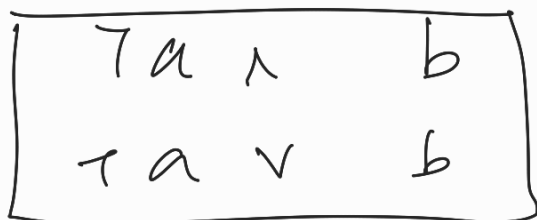
\*a + \*b

a && b || c → You can disallow this  
in the grammar.

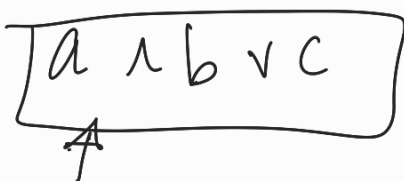
Partial ordered set  
of operators  
(Poset)

$\neg$   $\wedge$   $\vee$  (Boolean ops)

natural



confusing!



Syntax error

Force the programmer to write

$(a \wedge b) \vee c$

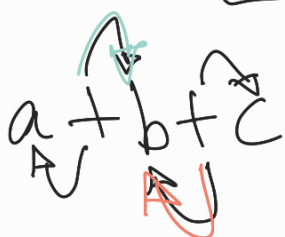
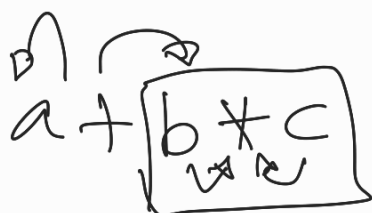
or

$a \wedge (b \vee c)$

Electronic circuit design

$\wedge \rightarrow \cdot$   
 $\vee \rightarrow +$

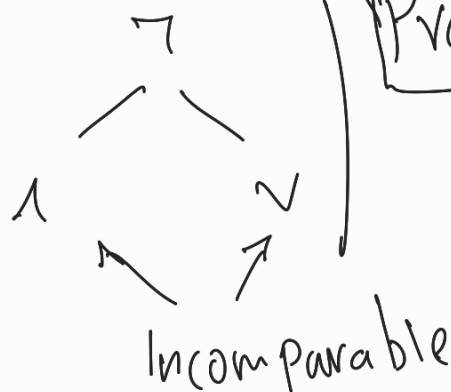
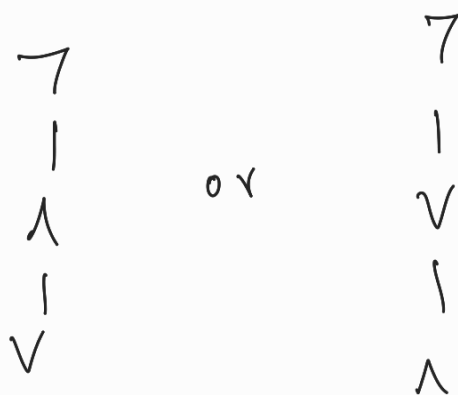
$ab + c$



$(a + b) + c$

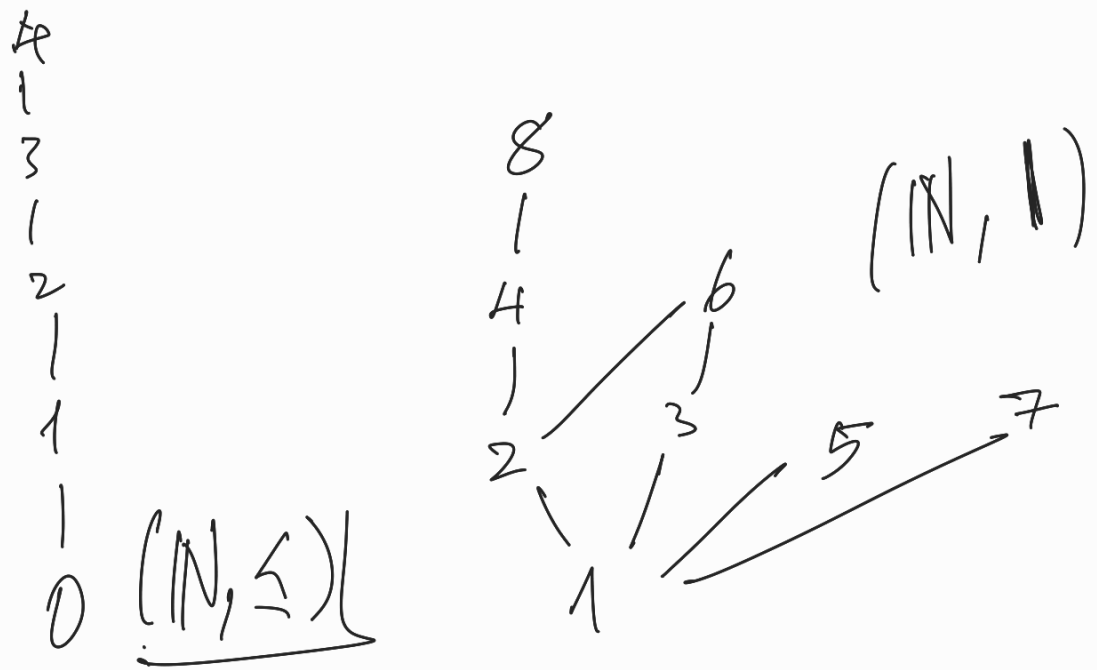
Notion of "force" for each operator.

$\text{rhs-force}(+)$   
 $> \text{lhs-force}(+)$



Pratt Parsers

$a + b * c$



Recap: DM partial and total orders

$$A, R \subseteq A \times A$$

$R$  is an ordering if the following Properties are satisfied.

(Reflexive)  $aRa$

(Transitive)  $aRb$  and  $bRc \rightarrow aRc$

(Anti-Symmetric)  $aRb$  and  $bRa \rightarrow a=b$

$R$  is usually written as  $\leq$

For a total order,

For all  $a, b$

$$a \leq b \text{ or } b \leq a$$

TopoSort  $\rightarrow$  From a Partial order create a consistent total order

# Subset ordering

atoms

$A \subseteq B$  | ~ Partial but not total.

( ) [ ]

In C.

many neg. def.

$\times, /$

$+, -$

$<, >, \geq, \leq$

$\&\& \ ||$

left assoc.

$(a + b) * c$

or

$a + (b * c)$

What is force in a partial order?

```
struct force {
```

```
enum {
```

```
LT,
```

```
EQ,
```

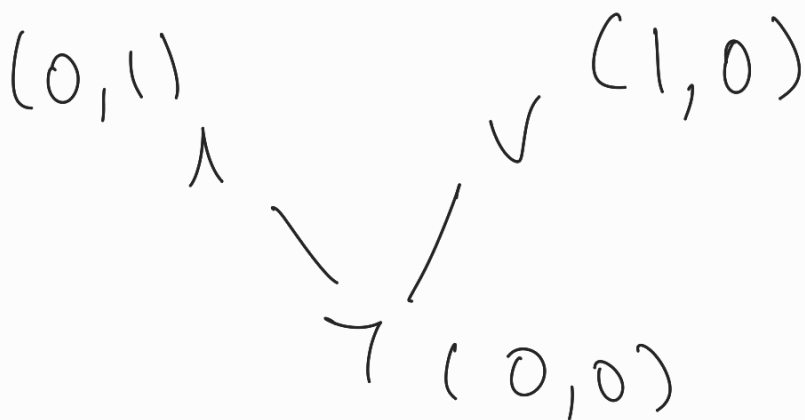
```
GT,
```

```
INCOMP
```

```
};  
comp_result;
```

```
comp_result comp (force a, force b)
```

```
comp (force (^), force (v))  
→ INCOMP
```



$(a,b) \leq (c,d)$   
iff  $a \leq c$   
and  $b \leq d$

# Function overloading

```
void f(long);  
void { f(char *);  
      ^  
      |  
      v  
f(0);  
-----  
f("hello");
```

```
def f(*args)  
    if len(args) == 0  
        :  
        :
```

Parametric Polymorphism

Bounded Parametric Polymorphism

$!a \neq b + c \parallel d \times e$  X