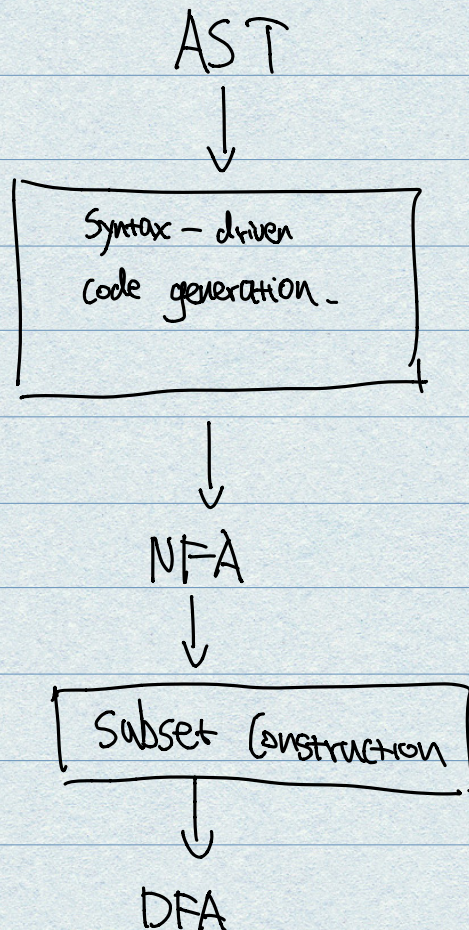


# Lecture #4. Query Language.

Call Chaining for AST Construction

Program Optimization via AST Rewrite.

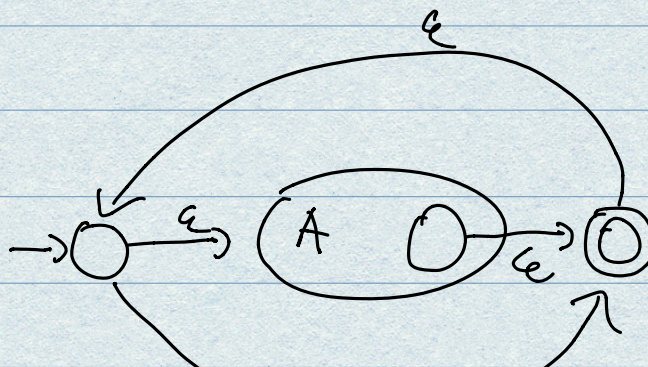


=> Regular Expressions to NFA

- Construct NFA for its constituents
- Notation: NFA for regexp  $M$ .

→  $M @$

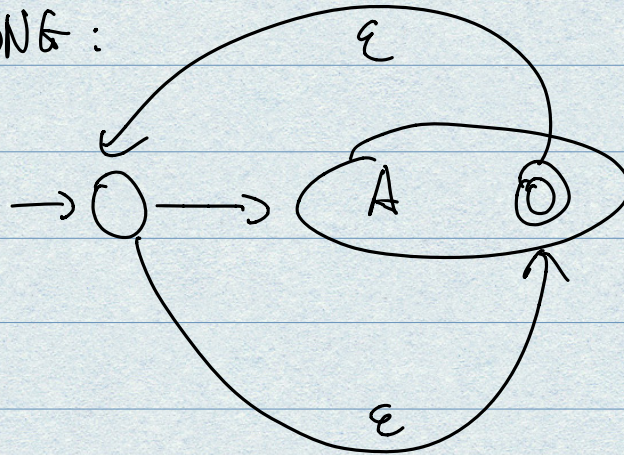
$A^*$ :



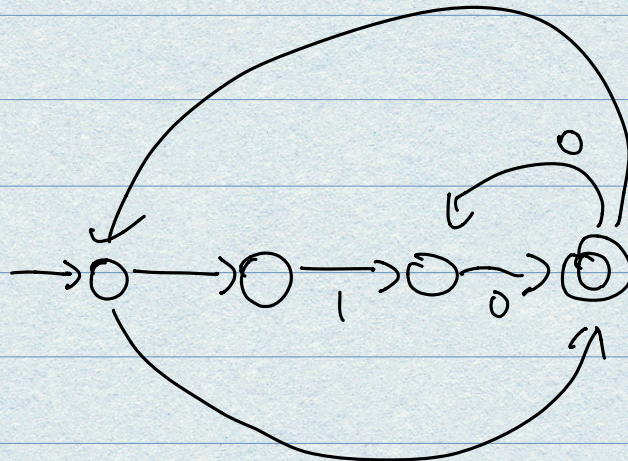


$\epsilon$

WRONG :



why :  $(10(00)^*)^*$



00 is accepted! X.

# Query Language.

var data = [...];

var query = Q.filter( $X \Rightarrow \dots$ );

var out = query.run(data);

$\Rightarrow$  Deep embedding  $\rightarrow$  optimization.



↳ program is represented as data (AST)

⇒ shallow embedding

`data.filter(λ);`

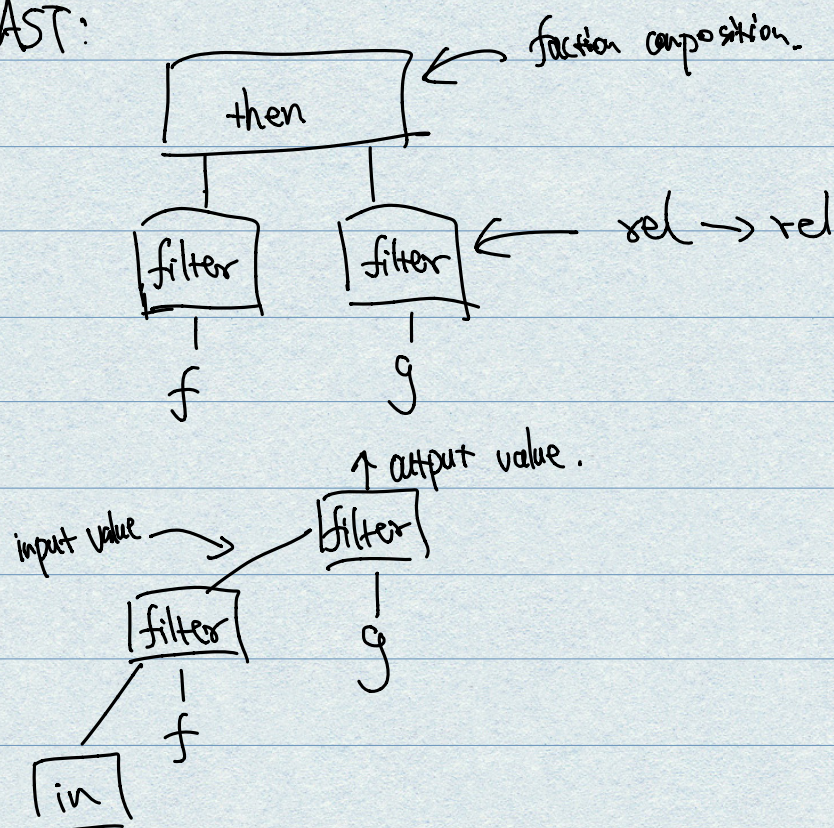
Differences:

1. we can optimize AST in deep version.
2. deferred execution. (run later)

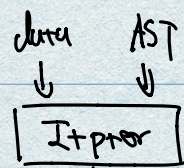
Q. `filter(x ⇒ x.stars >= 2)`

`filter(x ⇒ x.stars <= 8);`

AST:



# The Interpreter.





↓  
output.

bottom up or topdown?

