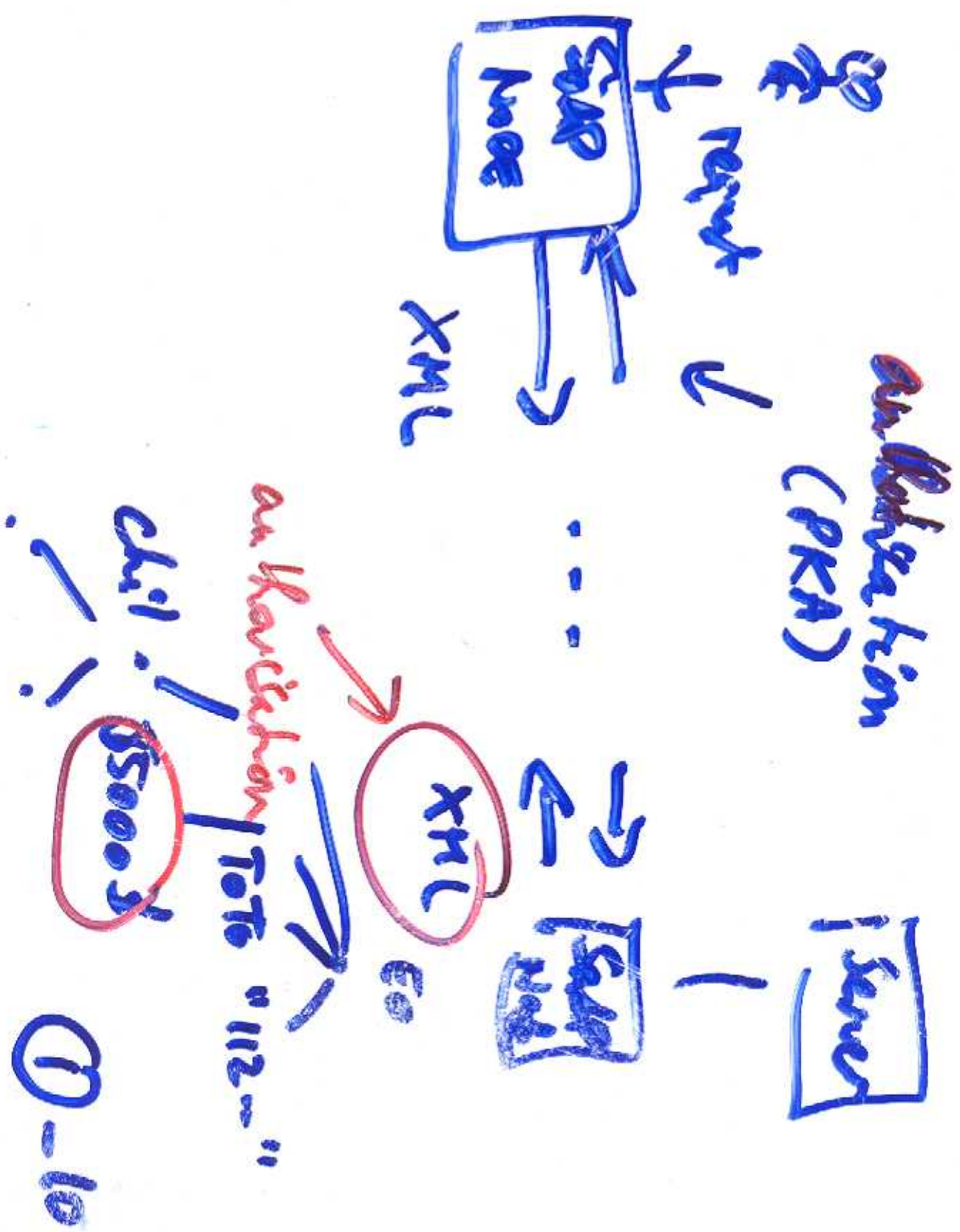


# SECURITY PART of SOAP



Access control  $\rightarrow$  allow to restrict  
access/update to information  
to authorized users only

Discretionary Access Control (DAC)

$U$  = users

AM = access mode

$O$  = objects

eg.  $r$  (read)  
 $u$  (update)

$x$  (execute)

each request checked againsts

permissions

$u_1, \{ r, w, x, \dots \}$

②-10

relation "employee"

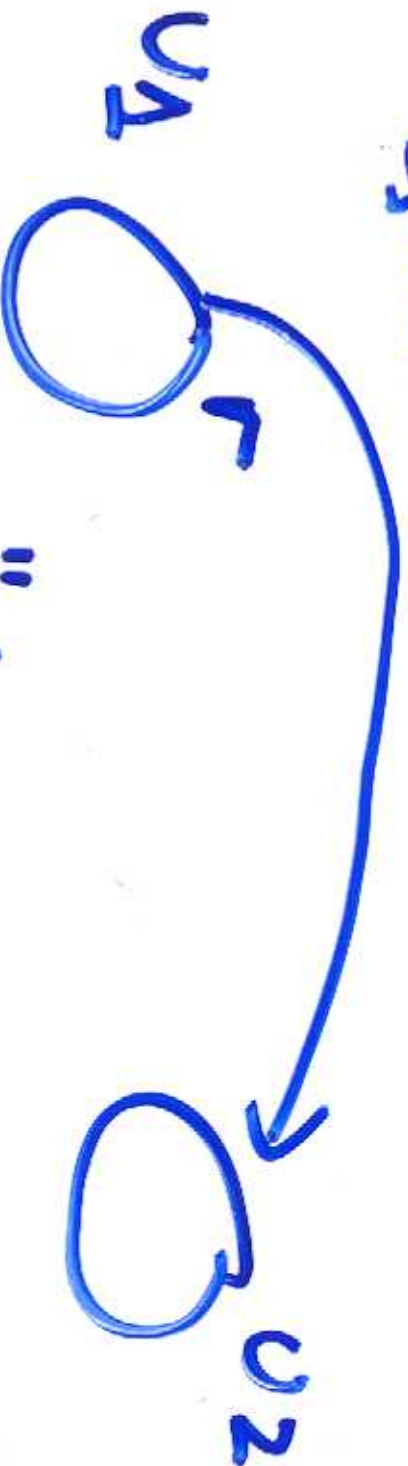
| employee | name | salary |
|----------|------|--------|
| e1       | Tom  | 50K    |
| e2       | Jim  | 70K    |
| :        | :    |        |

↳ multi-relation

|    |     |     |
|----|-----|-----|
| e1 | Tom | 50K |
| e2 | Jim | 70K |
| :  | :   |     |

DAC provides flexibility

DAC has problem of "~~the~~ non-secure" information from " "



DAC does not impose any restriction on the use of information (4-10)

- action when a user sees it

## Mandatory Access Control (MAC)

↳ Stop flow of high-~~est~~ data  
to low-sensitive data

• Security on "objects" = sensitivity  
of information contained within

the objects

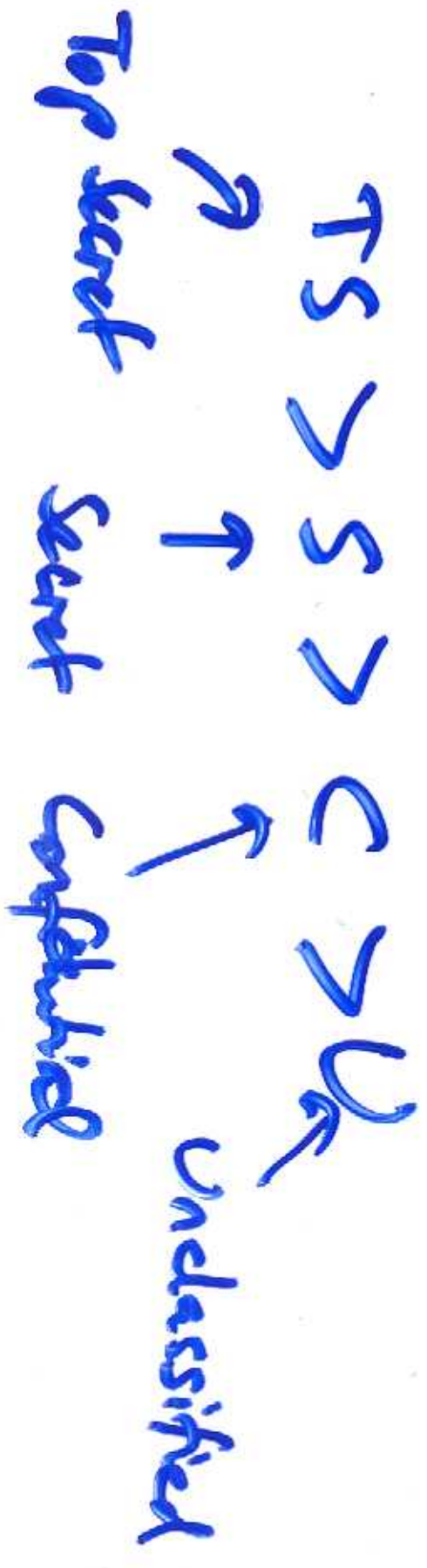
(represents the possible damage

that could be resulted from

non-authorized disclosure)

- Security on "subjects" = reflect ~~trust~~ trustworthiness not to disclose sensitive data

- ordered security labels



Security policies based on the following principles:

1. Bell and Lapin

~~clearance~~ CS

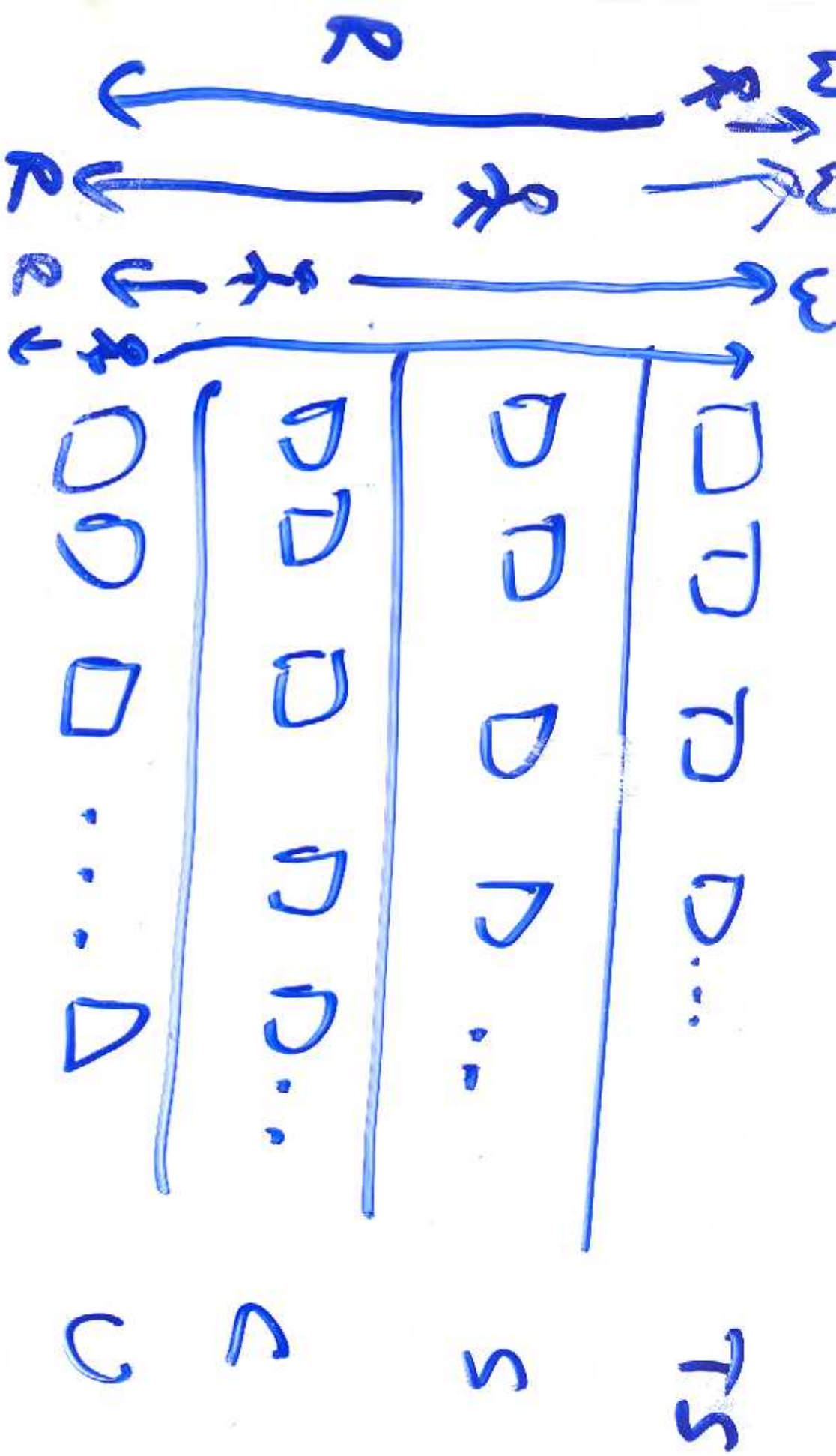
S: subject O: object

clearance (S)  $\geq$  classification (O)

2- write up

classification (O)  $\geq$  clearance (S)

Ⓢ-11



8-10

# Role-based Access Control (RBAC)

RBAC<sub>0</sub> → RBAC<sub>1</sub> → RBAC<sub>2</sub>

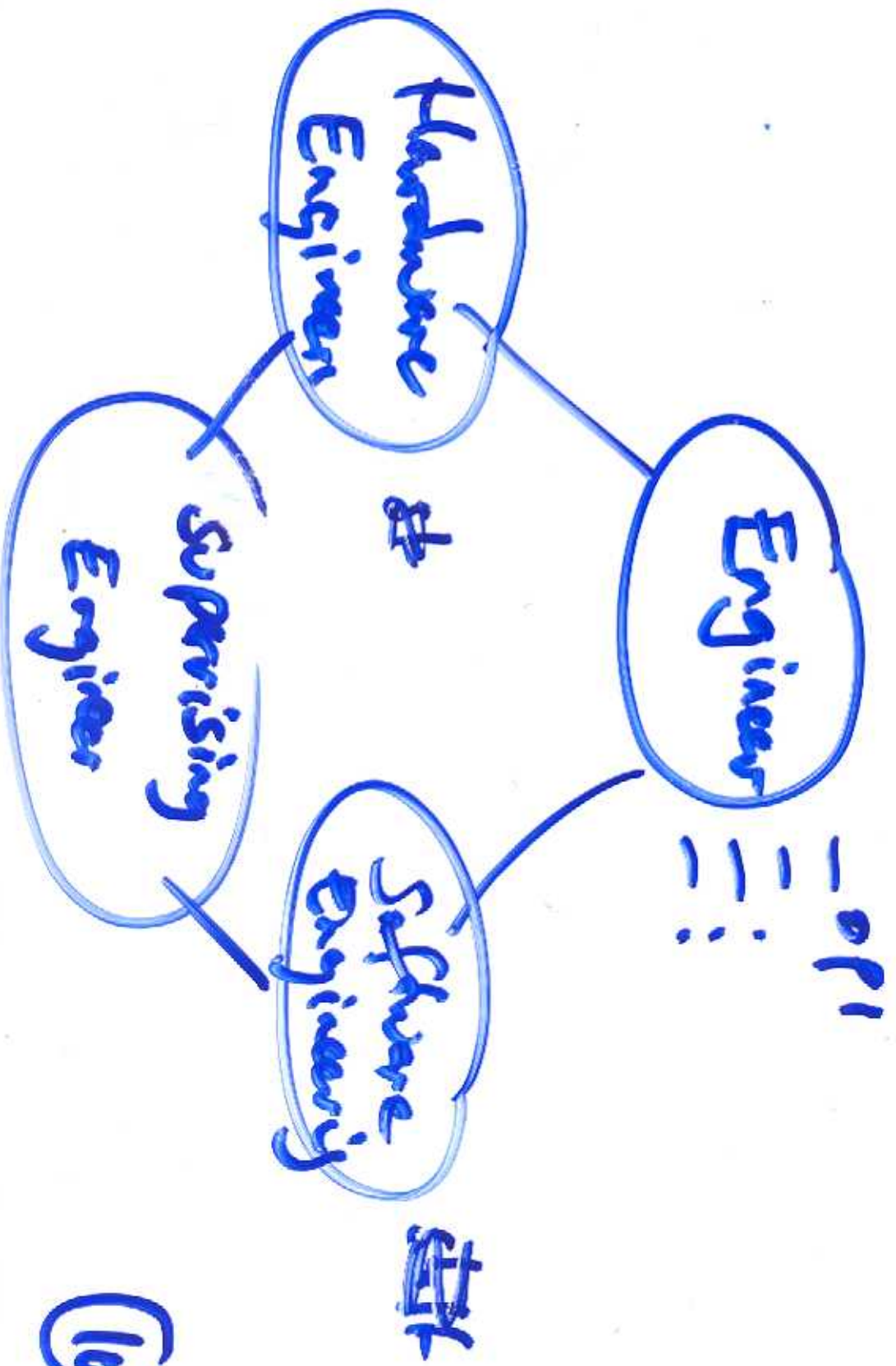
<sup>q</sup> basic                      <sup>q</sup> hierarchical                      <sup>q</sup> constraints

• DAC-style : assign authorizations to users/groups

• MAC-style : ~~for~~ info specific restrictions  
C information flows  
(control)

RBAC regulates user's access to information on the basis of activities the users execute in the system

ROLE = {actions + responsibilities}  
for certain activities



(10-15)

## SEPARATION OF DUTIES

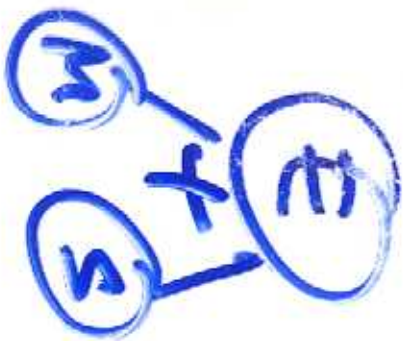
"no user should be given enough privileges to mis-use the system on their own."

eg a person authorising pay sheet

should not be the one

that prepares them.

create different/conflicting roles



Global Access Control

Global role

Global RBAC

- global role
- global hierarchy

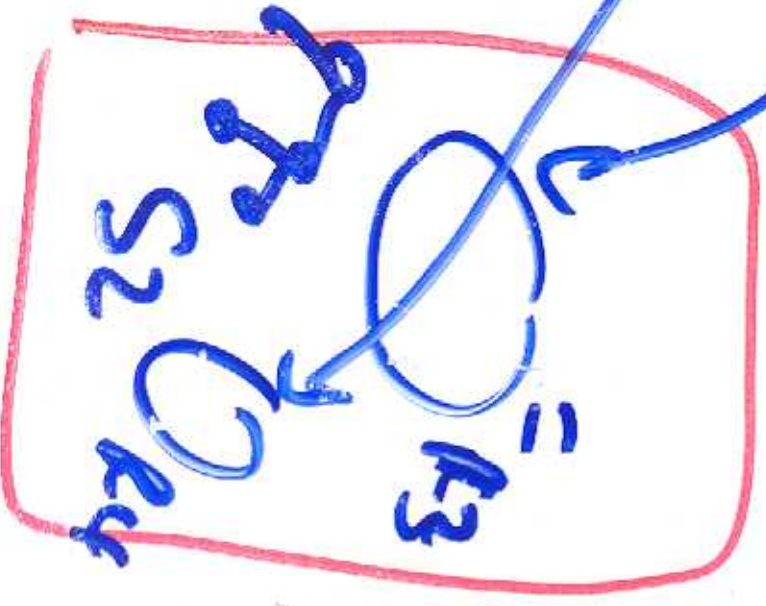
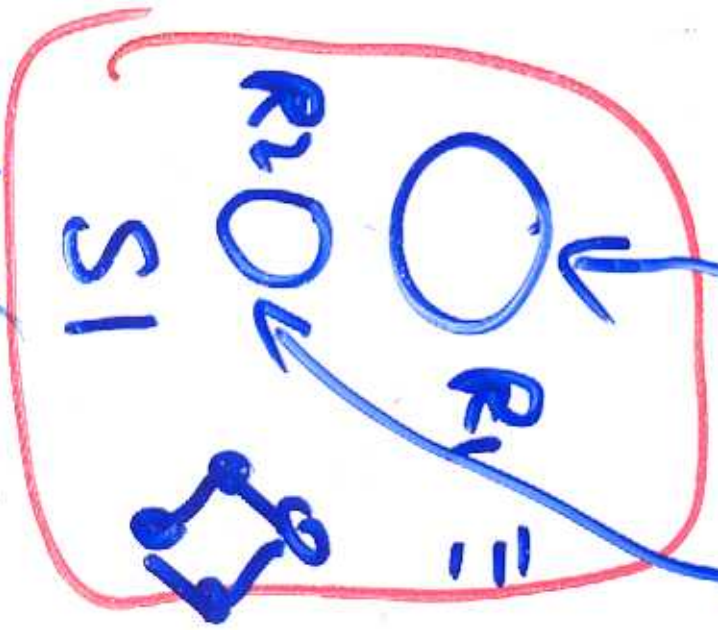
S1 {R1, R2}  
 S2 {R3, R4}

Local

RBAC

- local roles
- local hierarchy

RBAC



RBAC

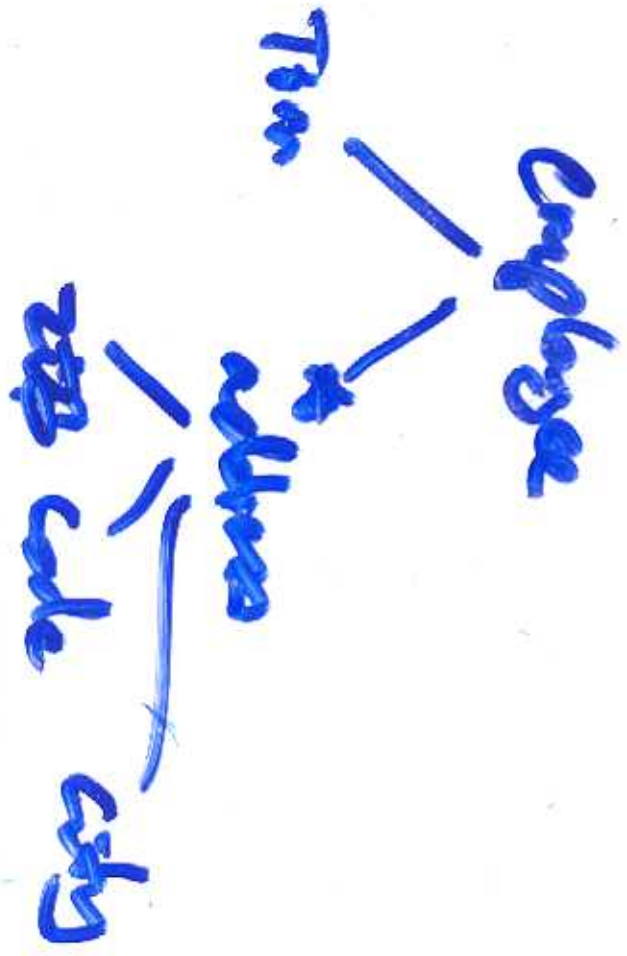
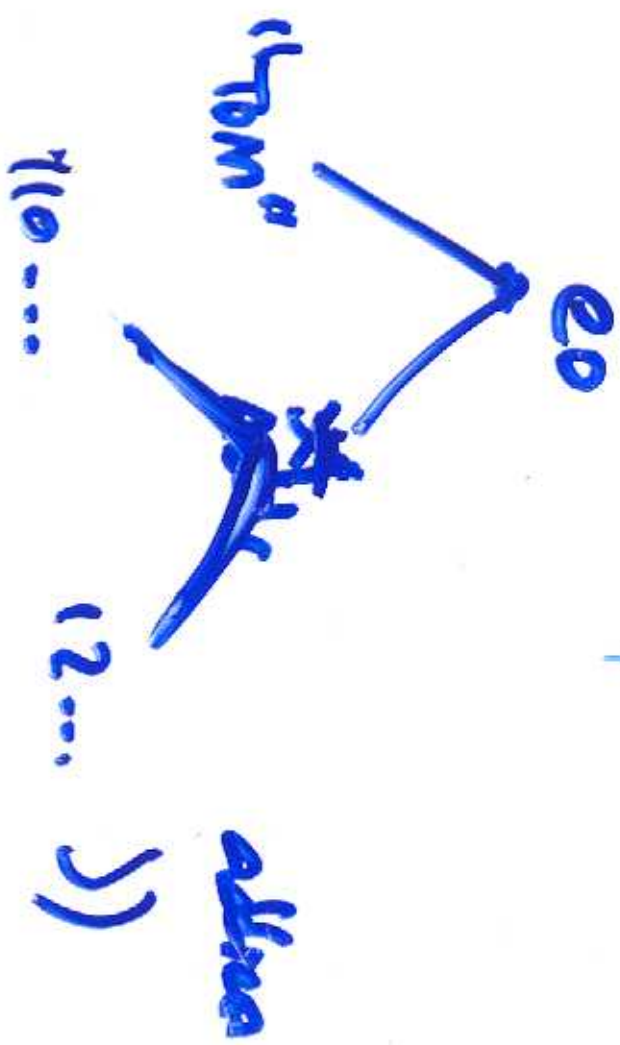
12-6

XML / SOAP Access subtree

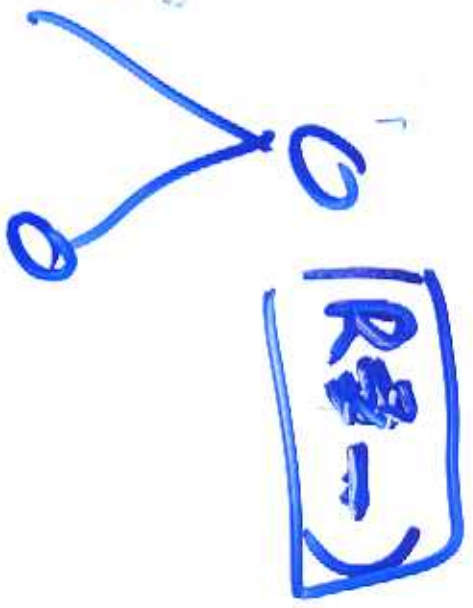
- type "do protect" tree

- DTD and

instance



LDH > L > LD



inspired



# Three hierarchies

## User Group Hierarchy (UGH)

$(U, UG, \leq_{UG})$

$U$ : set of user identifiers

$G$ : set of group names

$UG = U \cup G$

$x \leq_{UG} y$  iff  $x$  is a member of  $y$

## IP hierarchy (IPH)

$(I, IP, \leq_{IP})$

$IP$  set of IP patterns

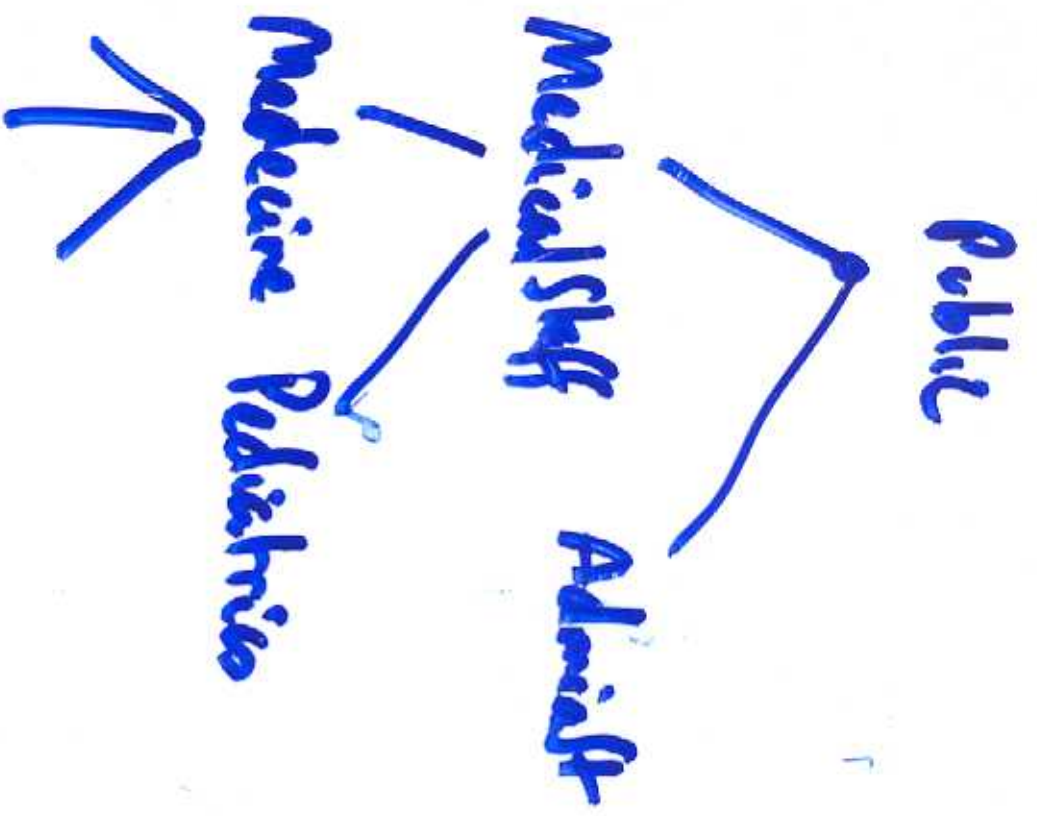
$x \leq_{IP} y$  if either each component of  $y$  is the wild card character or is equal to the corresponding component of  $x$

## Symbol Name Hierarchy (SNH)

$(S, SN, \leq_{SN})$

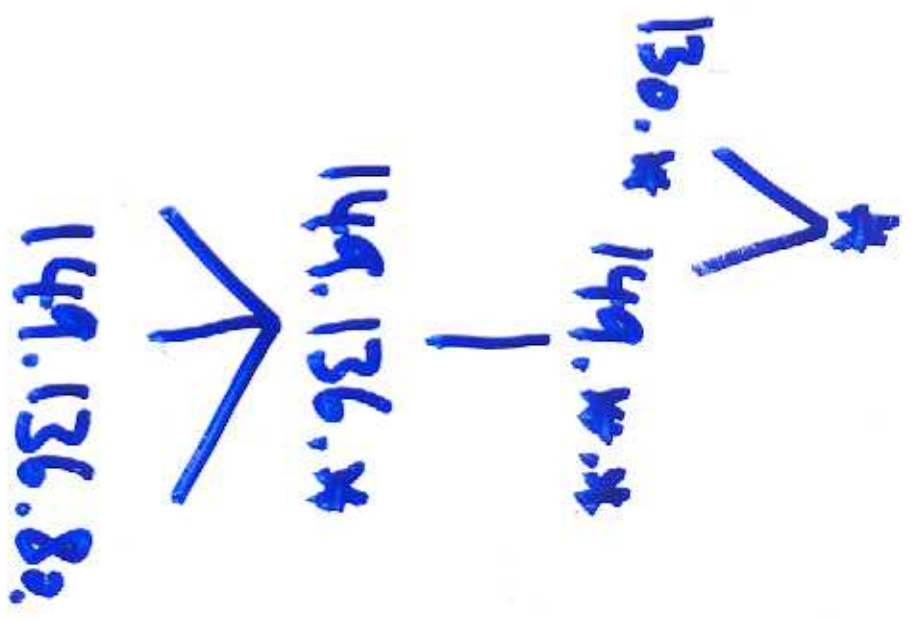
$x \leq_{SN} y$  if each comp. of  $y$  is wild character...

①



U.S. Group  
Hierarchy

2



IP Hierarchy



Symbol  
Name  
Hierarchy

# Authorisation Subject Hierarchy (ASH)

$$ASH = (R, AS, \leq_{AS})$$

$$R = (U \times I \times S)$$

$$AS = (U_G \times I_P \times S_N)$$

$$\langle U_{G_i}, I_{P_i}, S_{N_i} \rangle \leq_{AS} \langle U_{G_j}, I_{P_j}, S_{N_j} \rangle$$

$$\text{iff } U_{G_i} \leq_{U_G} U_{G_j}$$

$$I_{P_i} \leq_{I_P} I_{P_j}$$

$$S_{N_i} \leq_{S_N} S_{N_j}$$

③

# Principle

— Distinguish between DTD & Instance level of Authorisation

— "most specific takes precedence" Principle:  
DTD-level authorizations being propagated to an instance are overridden by possible authorizations specified for the instance.

+ { "soft" apply for document unless otherwise stated at DTD  
"hard" apply DTD authorisations, without exceptions

LDH > RDH > L > R > LD > RD > LS > RS

Strong

weak

4

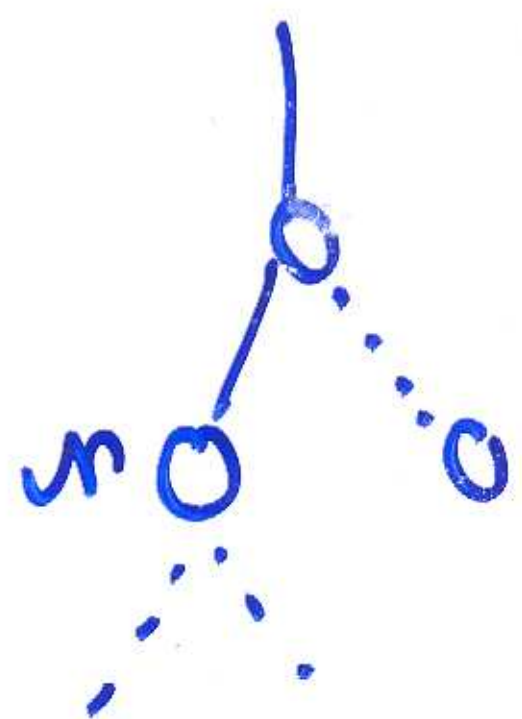
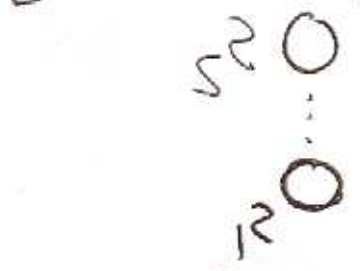
# DOCUMENT TREE LABELING

- 2 steps
- each node
- propagation of values
- returns down the tree

$$[z_1, \dots, H_{t-1}, H_t] = [H] \text{ label } m$$

$$\left. \begin{matrix} + \\ - \\ z \end{matrix} \right\} = \text{vector } \text{sign} \cdot m$$

$$\left\{ \begin{matrix} "+" \\ "-" \end{matrix} \right\} \text{ at level } 2 = \begin{matrix} z_1 \\ z_2 \end{matrix}$$



$$\{z_2\} \Rightarrow \{z_1\} \text{ and } \{z_2\} \Rightarrow \{z_1\} \text{ and } \{z_2\} \Rightarrow \{z_1\}$$

$z_1, z_2, \dots$

$$\begin{aligned} &\Leftrightarrow \phi \neq \begin{matrix} z_1 \\ z_2 \end{matrix} \text{ and } \phi = \begin{matrix} z_1 \\ z_2 \end{matrix} \text{ if } \\ &\Leftrightarrow \phi = \begin{matrix} z_1 \\ z_2 \end{matrix} \text{ and } \phi \neq \begin{matrix} z_1 \\ z_2 \end{matrix} \\ &\Leftrightarrow \phi \neq \begin{matrix} z_1 \\ z_2 \end{matrix} \text{ and } \phi \neq \begin{matrix} z_1 \\ z_2 \end{matrix} \end{aligned}$$



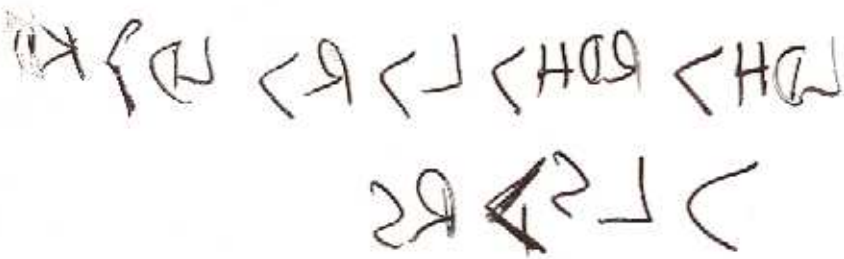
# Steps: substitution propagation down in the mechanism

(a) - propagate on  
p. vector [E] that are  
reverse



(b) - propagate according  
to "more specific subject"  
"denial to presence"

to "more specific subject"  
"denial to presence"



Keep in substitution

$$R = 9$$

$$L = 1$$

the substitution will  
be same as 9.

$$R = 9$$

$$L = 1$$

