



The Stanford InfoLab

Hector Garcia-Molina

Bob Mungamuru

Stanford University

InfoLab Research

- Information:
 - how to obtain information
 - how to manage it
 - how to exploit it

Hector's Current Interests

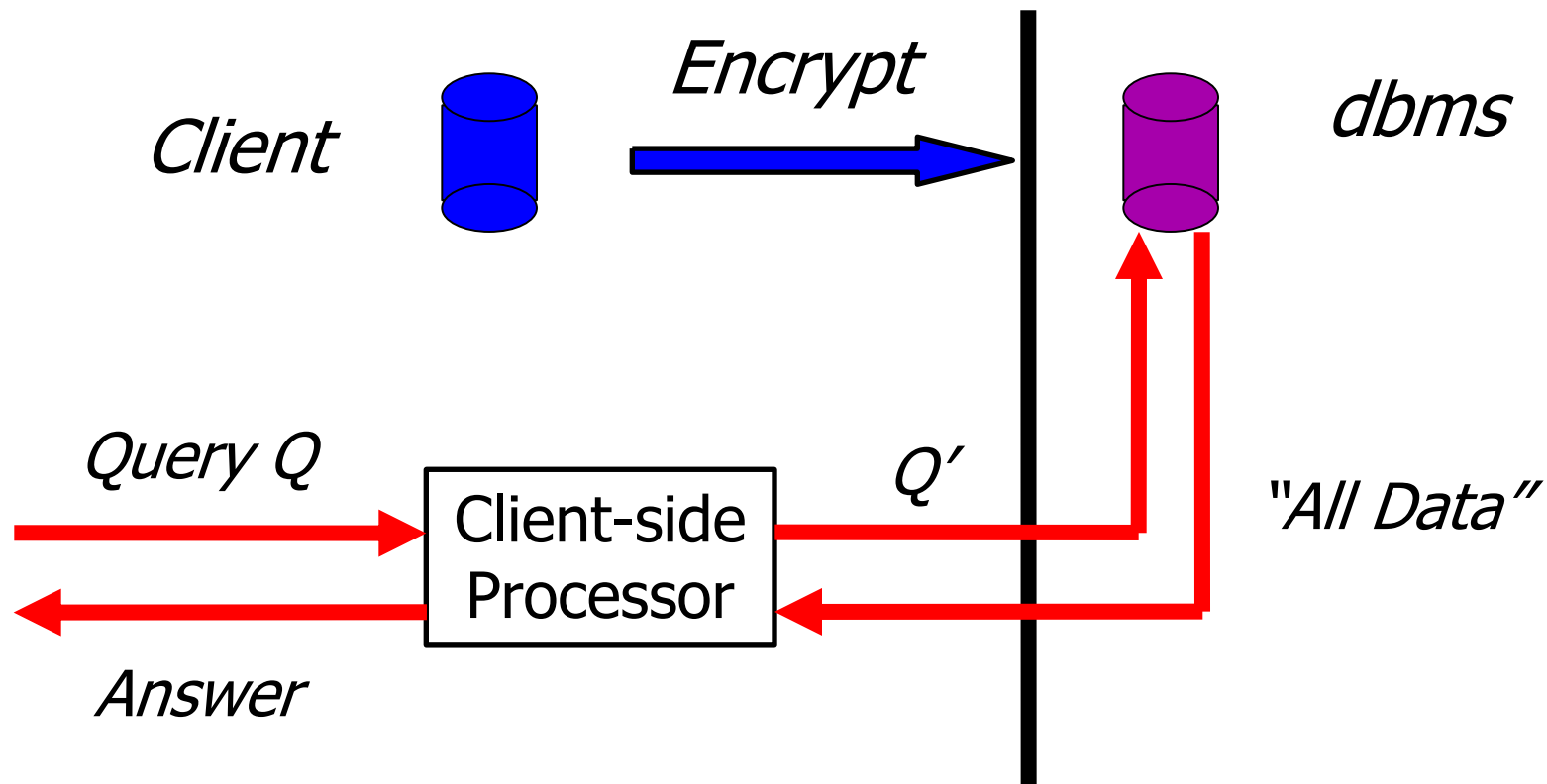
- Information Privacy & Security
- Managing Bio-Diversity Information
- Entity Resolution
- Web Information
- Peer to Peer Systems

Information Privacy & Security

- How to build a SECURE database system?
 - good performance
 - usable

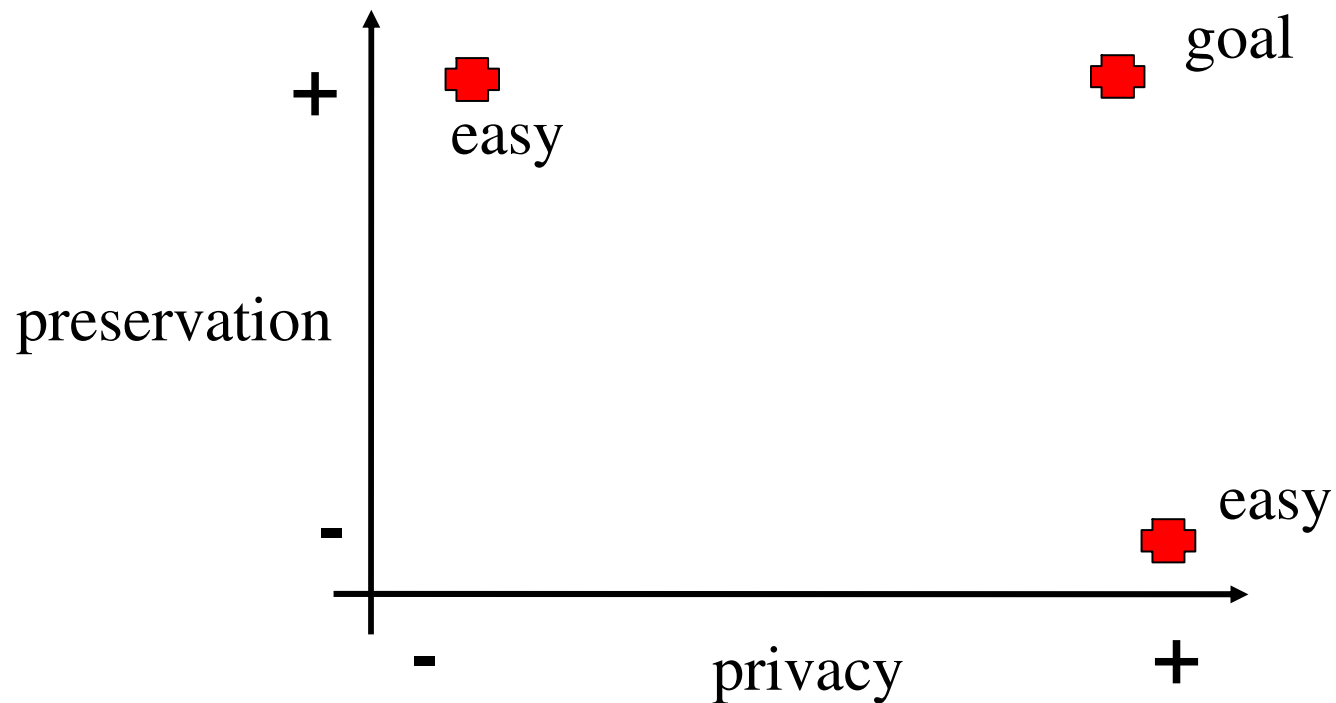
Information Privacy & Security

- How to build a SECURE database system?
 - good performance
 - usable



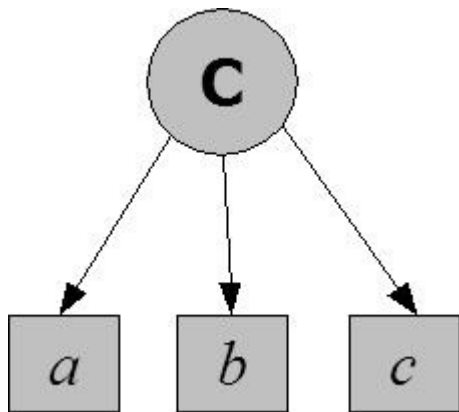
Information Privacy & Security

- Preservation, Performance, Functionality

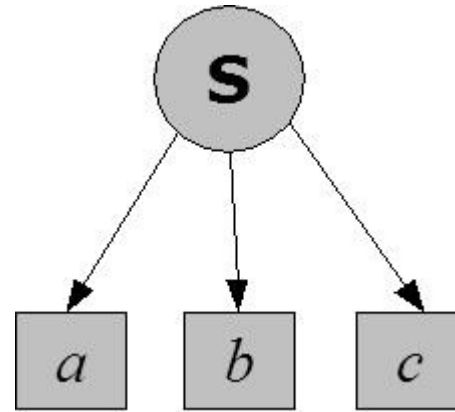


Key Configuration Management

- Safeguarding *sensitive data*
 - *against data loss*
 - *against unauthorized access*



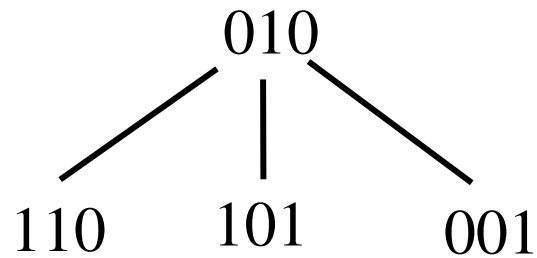
copy



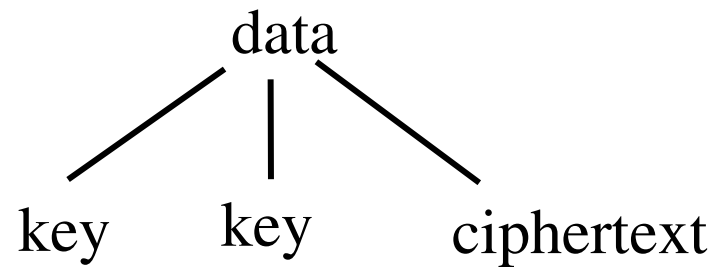
split

Splitting

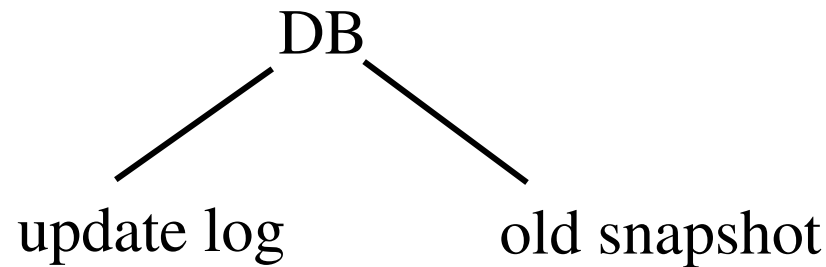
- XOR



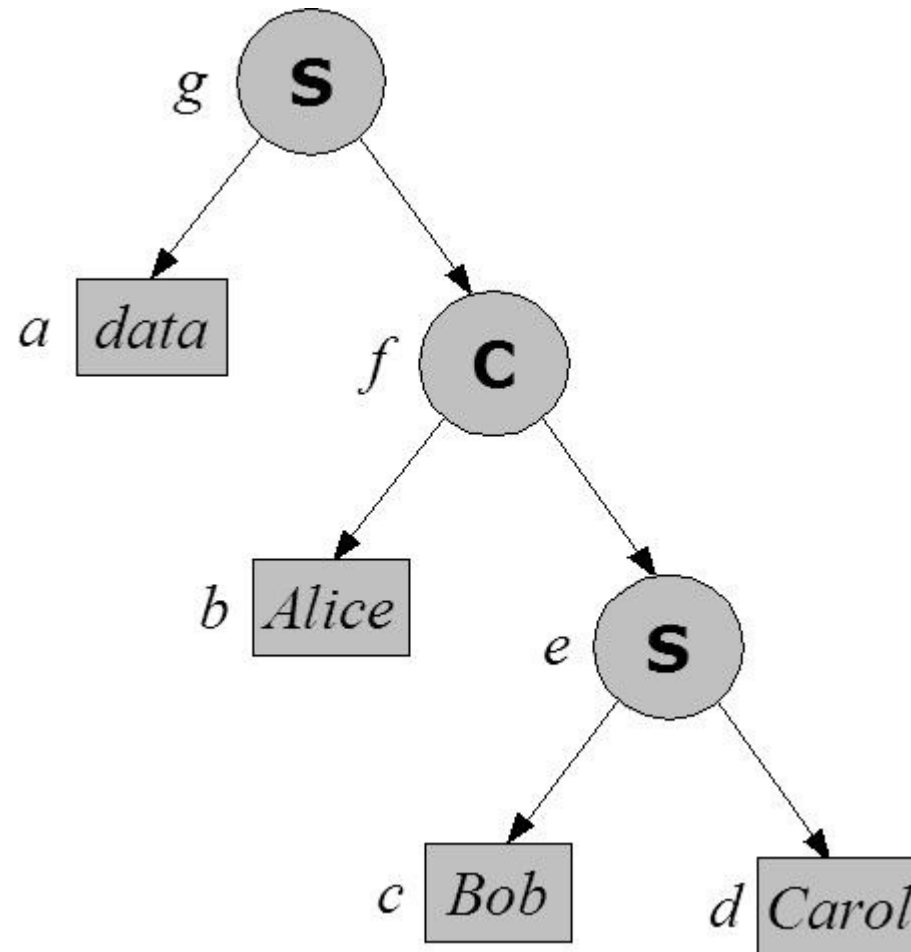
- Encryption



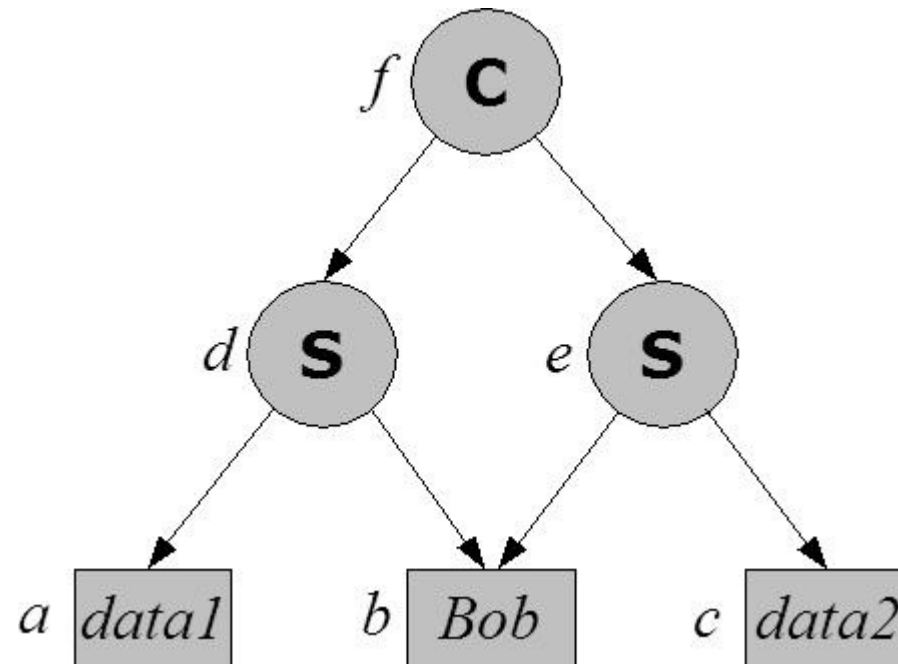
- DB + update logs



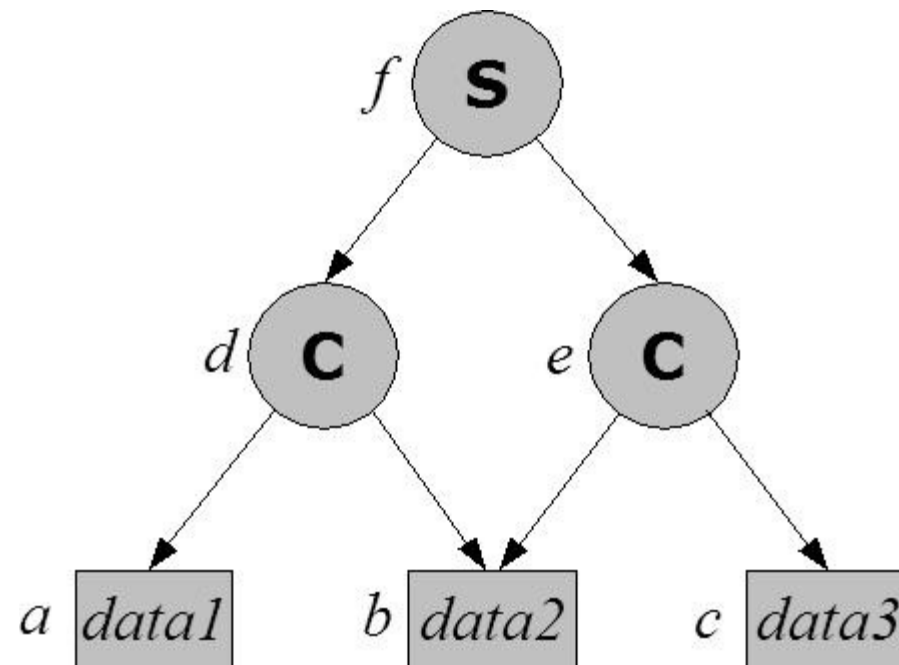
Example Configuration



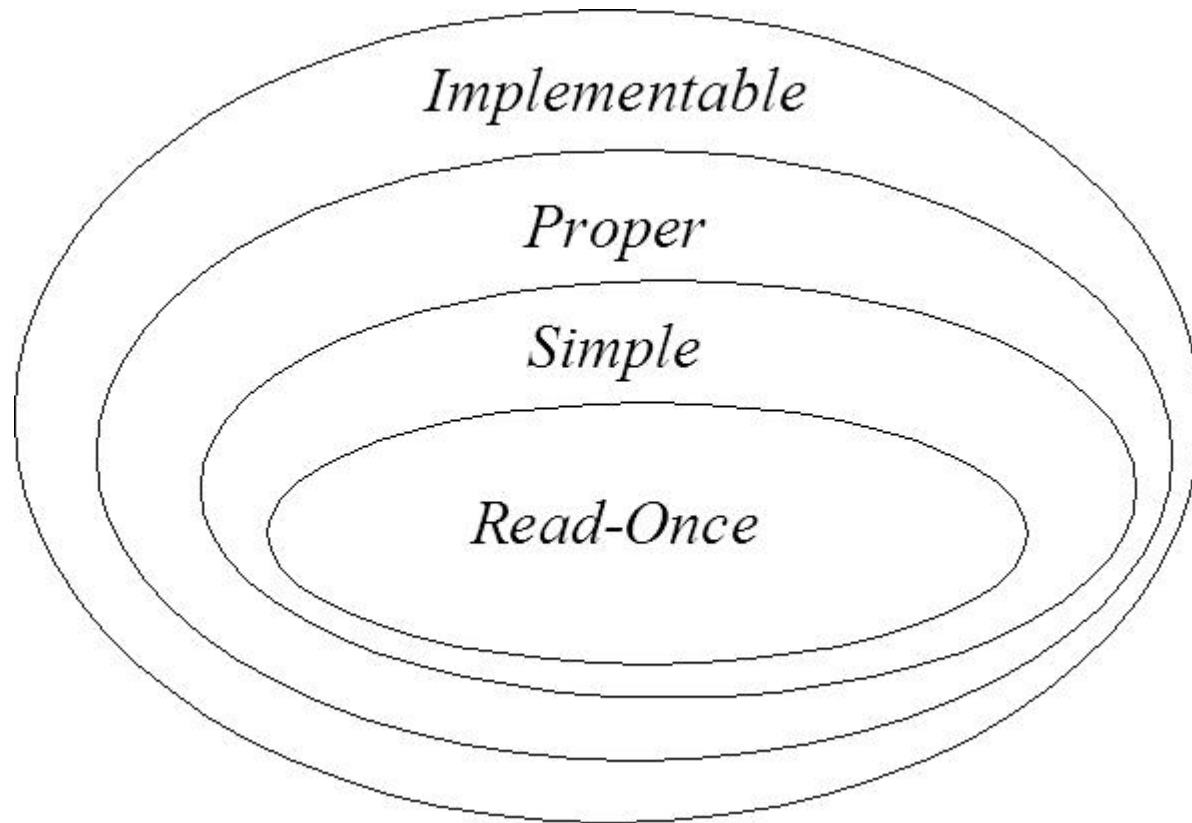
Sharing Keys



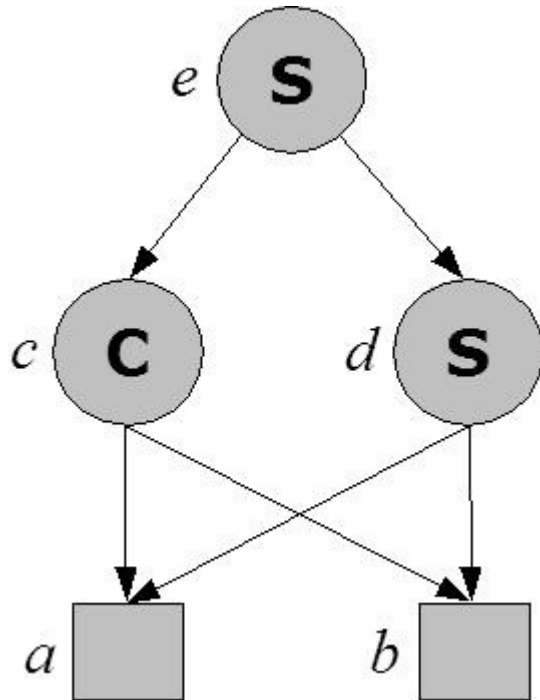
“Problematic” Configuration



Configurations



Checking a Configuration



$$a = b = c$$

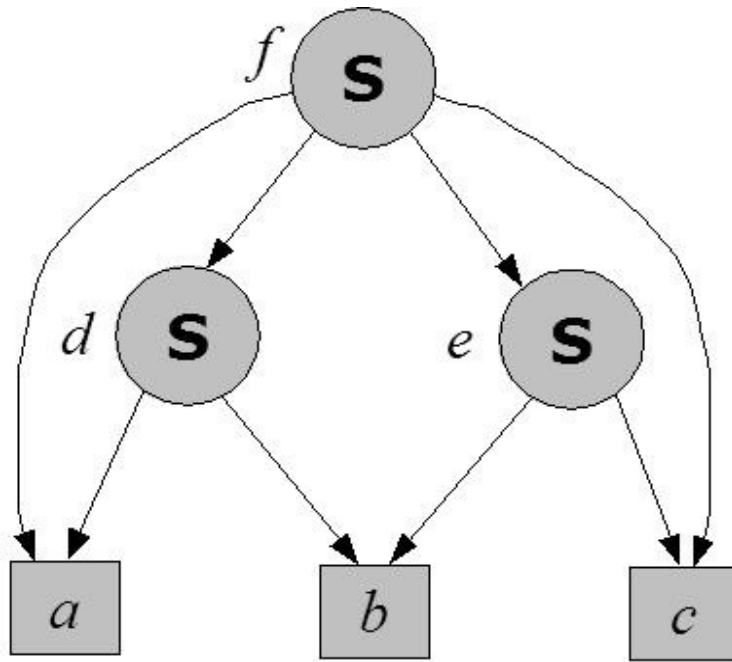
$$(a = -1) \oplus (b = -1)$$

$$(c = -2) \oplus (d = -2)$$

no satisfying assignment!

therefore, unimplementable

Checking Another Configuration



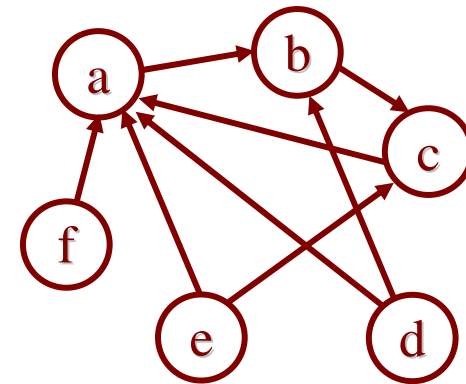
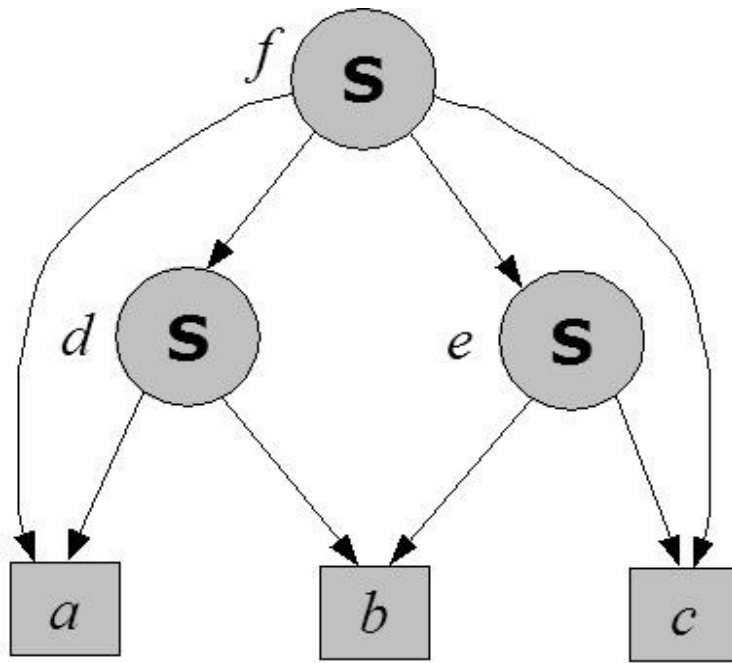
$$(a = -1) \oplus (d = -1) \oplus (e = -1) \oplus (c = -1)$$

$$(a = -2) \oplus (b = -2)$$

$$(b = -3) \oplus (c = -3)$$

Satisfying Assignment: $a = -1, b = -2, c = -3$

Checking Another Configuration



vertex
dependency
graph

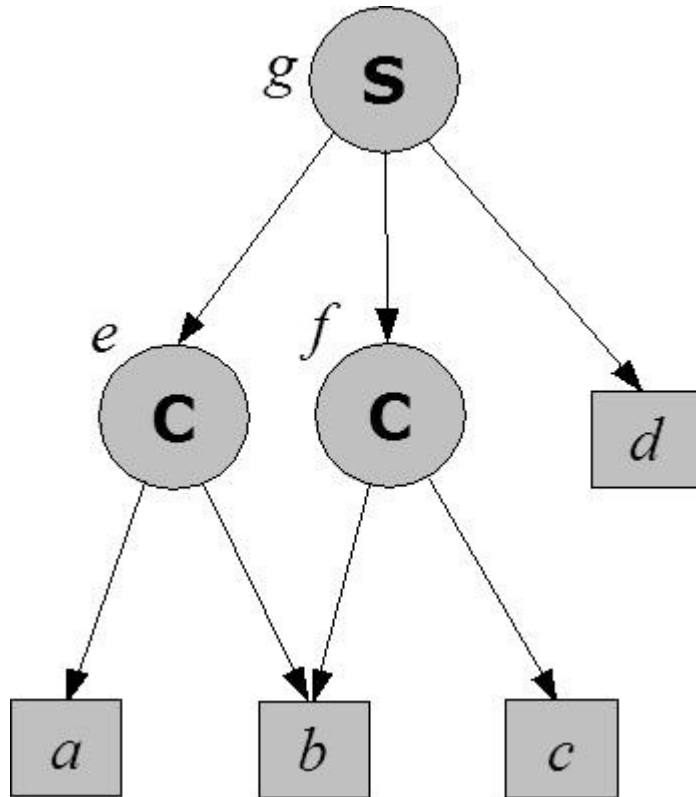
$$(a = -1) \oplus (d = -1) \oplus (e = -1) \oplus (c = -1)$$

$$(a = -2) \oplus (b = -2)$$

$$(b = -3) \oplus (c = -3)$$

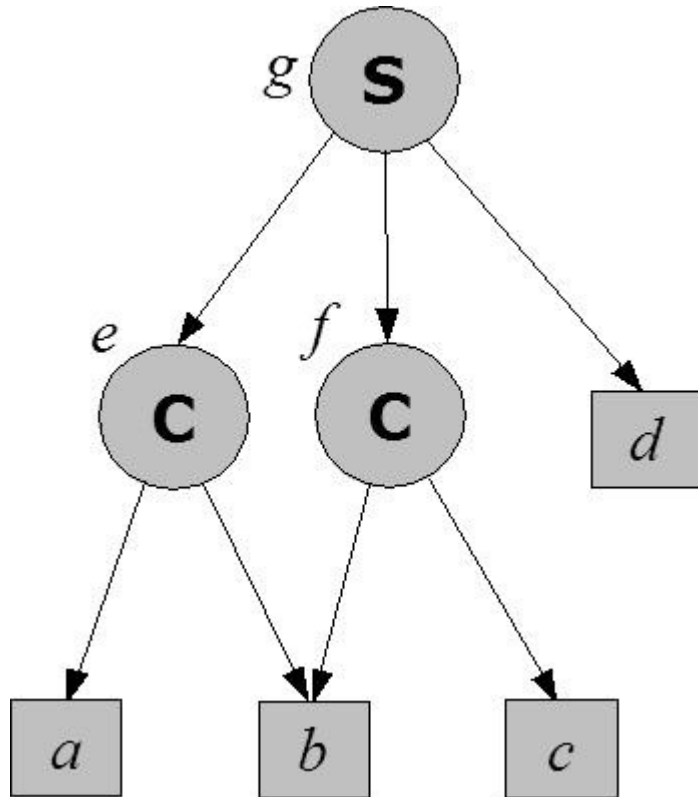
Satisfying Assignment: $a = -1, b = -2, c = -3$

Proper Configurations



$$(e = -1) \oplus (f = -1) \oplus (d = -1)$$
$$a = e = b = f = c$$

Proper Configurations



$$(e = -1) \oplus (f = -1) \oplus (d = -1)$$
$$a = e = b = f = c$$

added constraint:

$$e \neq f \neq d$$

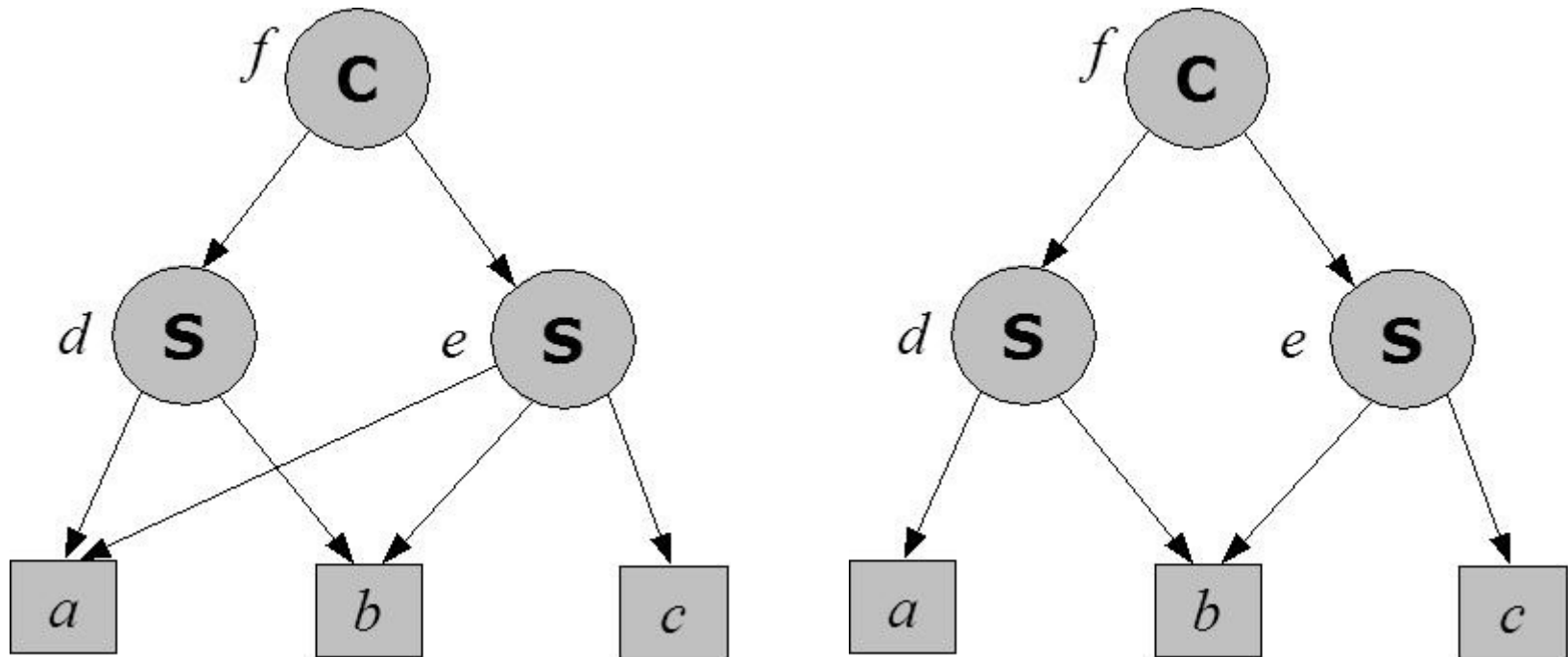
Improper configuration, since e and f are forced to be equal.

Simple Configurations

- Simple configurations have special structure:
 - S-vertices have at least one unshared child
 - C-vertices have no shared children
- Theorem:

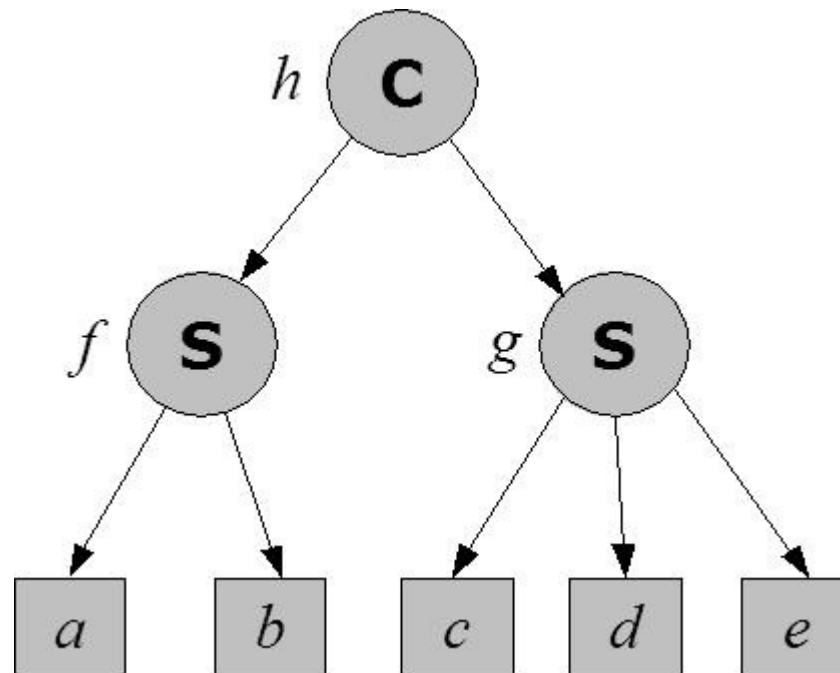
All simple configurations are proper.

Simple Configurations



The configuration on the right is simple.

Read-Once Configurations



Comments

- Logical transformations **do not** necessarily preserve properties
 - $d (b + (ac(a+c)) + ae) \dots$ unimplementable
 - $d (a + b) (b + c + e) \dots$ implementable
 - $d (b + ac + ae) \dots$ simple

Comments

