

## Lois déjà présentes dans la thèse et/ou l'article

Pour certaines lois, la formulation change par rapport à la formulation initiale.

### Lois locales

$$\pi_{\delta_1} \circ \dots \circ \pi_{\delta_n} \equiv \pi_{\delta_1 \cap \dots \cap \delta_n} \quad (1)$$

$$\sigma_{p_1} \circ \dots \circ \sigma_{p_n} \equiv \sigma_{p_1 \wedge \dots \wedge p_n} \quad (2)$$

$$\pi_{\delta} \circ \sigma_p \equiv \sigma_p \circ \pi_{\delta} \quad \text{si } \text{dom}(p) \subset \delta \quad (3)$$

### Lois identité

$$\text{id} \equiv \text{defrag} \circ \text{frag}_{\delta} \quad (4)$$

$$\text{id} \equiv \text{decrypt}_{\alpha, c} \circ \text{crypt}_{\alpha, c} \quad (5)$$

### Lois de projection

$$\pi_{\delta} \circ \text{decrypt}_{\alpha, c} \equiv \text{decrypt}_{\alpha, c} \circ \pi_{\delta} \quad \text{si } \alpha \in \delta \quad (6)$$

$$\pi_{\delta} \circ \text{decrypt}_{\alpha, c} \equiv \pi_{\delta} \quad \text{si } \alpha \notin \delta \quad (7)$$

$$\pi_{\delta} \circ \text{defrag} \equiv \text{defrag} \circ (\pi_{\delta \cap \delta'}, \pi_{\delta \setminus \delta'}) \quad \text{où } \delta' \text{ est le schéma relationnel du premier fragment} \quad (8)$$

### Lois de sélection

Lorsqu'une défragmentation est effectuée, on supposera que le schéma relationnel du fragment de gauche s'appelle  $\delta'$ .

$$\sigma_p \circ \text{decrypt}_{\alpha, c} \equiv \text{decrypt}_{\alpha, c} \circ \sigma_p \quad \text{si } \text{dom}(p) \cap \alpha = \emptyset \quad (9)$$

$$\sigma_p \circ \text{decrypt}_{\alpha, c} \equiv \text{decrypt}_{\alpha, c} \circ \sigma_{c \Rightarrow p} \quad \text{si } p \text{ est compatible avec } c \quad (10)$$

$$\sigma_p \circ \text{defrag} \equiv \text{defrag} \circ (\sigma_p, \text{id}) \quad \text{si } \text{dom}(p) \subset \delta' \quad (11)$$

$$\sigma_p \circ \text{defrag} \equiv \text{defrag} \circ (\text{id}, \sigma_p) \quad \text{si } \text{dom}(p) \subset \Delta \setminus \delta' \quad (12)$$

### Lois d'agrégation

Pour tout chiffrement  $c$ , on appellera  $c'$  le chiffrement qui agit sur une liste en appliquant  $c$  à chacun des éléments de la liste. Lorsqu'une défragmentation est effectuée, on supposera que le

schéma relationnel du fragment de gauche s'appelle  $\delta'$  .

$$\text{group}_\delta \circ \text{decrypt}_{\alpha,c} \equiv \text{decrypt}_{\alpha,c} \circ \text{group}_\delta \quad \text{Si } \alpha \notin \delta \quad (13)$$

$$\text{group}_\delta \circ \text{decrypt}_{\alpha,c} \equiv \text{decrypt}_{\alpha,c} \circ \text{group}_\delta \quad \text{Si } \alpha \in \delta \text{ et } c \text{ est compatible avec l'égalité} \quad (14)$$

$$\text{group}_\delta \circ \text{defrag} \equiv \text{defrag} \circ (\text{send} \circ \text{group}_\delta, \text{receiveAndGroup}()) \quad \text{Si } \delta \subset \delta' \quad (15)$$

$$\text{group}_\delta \circ \text{defrag} \equiv \text{defrag} \circ (\text{receiveAndGroup}(), \text{send} \circ \text{group}_\delta) \quad \text{Si } \delta \cap \delta' = \emptyset \quad (16)$$

## Lois de composition des protections

Lorsqu'une défragmentation est effectuée, on supposera que le schéma relationnel du fragment de gauche s'appelle  $\delta'$  .

$$\text{id} \circ f \equiv f \circ \text{id} \equiv f \quad (17)$$

$$\text{frag}_\delta \circ \text{decrypt}_{\alpha,c} \equiv (\text{decrypt}_{\alpha,c}, \text{id}) \circ \text{frag}_\delta \quad \text{si } \alpha \in \delta' \quad (18)$$

$$\text{frag}_\delta \circ \text{decrypt}_{\alpha,c} \equiv (\text{id}, \text{decrypt}_{\alpha,c}) \circ \text{frag}_\delta \quad \text{si } \alpha \notin \delta' \quad (19)$$

$$\text{decrypt}_{\alpha,c} \circ \text{defrag} \equiv \text{defrag} \circ (\text{decrypt}_{\alpha,c}, \text{id}) \quad \text{si } \alpha \in \delta' \quad (20)$$

$$\text{decrypt}_{\alpha,c} \circ \text{defrag} \equiv \text{defrag} \circ (\text{id}, \text{decrypt}_{\alpha,c}) \quad \text{si } \alpha \notin \delta' \quad (21)$$

$$\text{frag}_\delta \circ \text{crypt}_{\alpha,c} \equiv (\text{crypt}_{\alpha,c}, \text{id}) \circ \text{frag}_\delta \quad \text{si } \alpha \in \delta' \quad (22)$$

$$\text{frag}_\delta \circ \text{crypt}_{\alpha,c} \equiv (\text{id}, \text{crypt}_{\alpha,c}) \circ \text{frag}_\delta \quad \text{si } \alpha \notin \delta' \quad (23)$$

## Lois que je propose de rajouter

### Commutation de defrag et crypt

Lorsqu'une défragmentation est effectuée, on supposera que le schéma relationnel du fragment de gauche s'appelle  $\delta'$  .

$$\text{crypt}_{\alpha,c} \circ \text{defrag} \equiv \text{defrag} \circ (\text{crypt}_{\alpha,c}, \text{id}) \quad \text{si } \alpha \in \delta' \quad (24)$$

$$\text{crypt}_{\alpha,c} \circ \text{defrag} \equiv \text{defrag} \circ (\text{id}, \text{crypt}_{\alpha,c}) \quad \text{si } \alpha \notin \delta' \quad (25)$$

### Lois évidentes

$$\text{crypt}_{\alpha,c} \circ \text{crypt}_{\beta,s} \equiv \text{crypt}_{\beta,s} \circ \text{crypt}_{\alpha,c} \quad \text{si } \alpha \neq \beta \quad (26)$$

A priori, chiffrer une donnée déjà chiffrée semble une mauvaise idée de tous points de vue.

$$\text{decrypt}_{\alpha,c} \circ \text{decrypt}_{\beta,s} \equiv \text{decrypt}_{\beta,s} \circ \text{decrypt}_{\alpha,c} \quad \text{si } \alpha \neq \beta \quad (27)$$

A priori, déchiffrer une donnée déjà déchiffrée semble une mauvaise idée de tous points de vue.

## Lois d'agrégation

$$\text{group}_\delta \circ \pi_{\delta'} \equiv \pi_{\delta'} \circ \text{group}_\delta \quad \text{si } \delta \subset \delta' \quad (28)$$

$$\text{group}_\delta \circ \sigma_p \equiv \sigma_p \circ \text{group}_\delta \quad \text{si } \text{dom}(p) \subset \delta \quad (29)$$

$$\text{group ne commute pas avec lui-même} \quad (30)$$

## Lois de jonction

Lorsque les compositions de fonctions considérées à continuations prennent plusieurs arguments, on appelle, de gauche à droite,  $\delta_1, \delta_2, \delta_3, \dots$  leurs schémas relationnels respectifs.

### Pour join et defrag

$$\bowtie \circ (\text{defrag}, \text{id}) \equiv \text{defrag} \circ (\text{id}, \bowtie) \quad \text{si } \delta_1 \cap (\delta_2 \cup \delta_3) = \emptyset \quad (31)$$

### Pour join et decrypt

$$\text{decrypt}_{\alpha, c} \circ \bowtie \equiv \bowtie \circ (\text{decrypt}_{\alpha, c}, \text{id}) \quad \text{si } \alpha \in \delta_1 \quad (32)$$

$$\text{decrypt}_{\alpha, c} \circ \bowtie \equiv \bowtie \circ (\text{id}, \text{decrypt}_{\alpha, c}) \quad \text{si } \alpha \in \delta_2 \quad (33)$$

### Pour join et join on a envie d'écrire

$$\bowtie \circ (\bowtie, \text{id}) \equiv \bowtie \circ (\text{id}, \bowtie) \quad (34)$$

La correction de cette formule est à vérifier.

### Join et group

$$\text{group}_\delta \circ \bowtie \equiv \bowtie \circ (\text{group}_\delta, \text{group}_\delta) \quad \text{si } \delta = \delta_1 \cap \delta_2 \quad (35)$$

## Lois du fold

$$\text{fold}_{\alpha, f, z} \circ \pi_\delta \equiv \pi_\delta \circ \text{fold}_{\alpha, f, z} \quad \text{si } \alpha \in \delta \quad (36)$$

$$\text{fold}_{\alpha, f, z} \circ \pi_\delta \equiv \pi_\delta \quad \text{si } \alpha \notin \delta \quad (37)$$

$$\pi_\delta \circ \text{fold}_{\alpha, f, z} \equiv \pi_\delta \quad \text{si } \alpha \notin \delta \quad (38)$$

$$\sigma_p \circ \text{fold}_{\alpha, f, z} = \text{fold}_{\alpha, f, z} \circ \sigma_p \quad \text{si } \alpha \notin \text{dom}(p) \quad (39)$$

On appelle  $\delta'$  le schéma relationnel du premier argument.

$$\text{fold}_{\alpha, f, z} \circ \text{defrag} = \text{defrag} \circ (\text{fold}_{\alpha, f, z}, \text{id}) \quad \text{si } \alpha \in \delta' \quad (40)$$

$$\text{fold}_{\alpha, f, z} \circ \text{defrag} = \text{defrag} \circ (\text{id}, \text{fold}_{\alpha, f, z}) \quad \text{si } \alpha \notin \delta' \quad (41)$$

$$\text{fold}_{\alpha,f,z} \circ \text{decrypt}_{\beta,c} = \text{decrypt}_{\beta,c} \circ \text{fold}_{\alpha,f,z} \quad \text{si } \alpha \neq \beta \quad (42)$$

$$\text{fold}_{\alpha,f,z} \circ \text{decrypt}_{\alpha,c} = \text{decrypt}_{\alpha,c} \circ \text{fold}_{\alpha,c \Rightarrow f, c \Rightarrow z} \quad \text{si } c \text{ est compatible avec } f \quad (43)$$

$$\text{fold}_{\alpha,f,z} \circ \text{fold}_{\beta,g,z'} = \text{fold}_{\beta,g,z'} \circ \text{fold}_{\alpha,f,z} \quad \text{si } \alpha \neq \beta \quad (44)$$

$$\text{fold}_{\alpha,f,z} \circ \text{group}_{\delta} = \text{group}_{\delta} \circ \text{fold}_{\alpha,f,z} \quad \text{si } \text{red}_{\alpha,f,z,\bullet} \text{ est injective} \quad (45)$$

On appelle  $\delta_1$  et  $\delta_2$  les schémas relationnels respectifs du premier et du deuxième argument.

$$\text{fold}_{\alpha,f,z} \circ \bowtie = \bowtie \circ (\text{fold}_{\alpha,f,z}, \text{id}) \quad \text{si } \alpha \in \delta_1 \setminus \delta_2 \quad (46)$$

$$\text{fold}_{\alpha,f,z} \circ \bowtie = \bowtie \circ (\text{id}, \text{fold}_{\alpha,f,z}) \quad \text{si } \alpha \in \delta_2 \setminus \delta_1 \quad (47)$$

$$\text{fold}_{\alpha,f,z} \circ \bowtie = \bowtie \circ (\text{fold}_{\alpha,f,z}, \text{fold}_{\alpha,f,z}) \quad \text{si } \text{red}_{\alpha,f,z,\bullet} \text{ est injective} \quad (48)$$