

Opérateur de fermeture, correspondance de Galois Treillis de Galois, Treillis de Concepts

Université de Montpellier - Faculté des Sciences
Master Informatique 1ère année
Ordres, treillis et induction (HAI715I)

16 octobre 2022

Opérateur de fermeture

Une application h est un opérateur de fermeture sur un ensemble ordonné (E, \leq_E) si :

- h est croissante : $x \leq_E y \Rightarrow h(x) \leq_E h(y)$
- h est extensive : $x \leq_E h(x)$
- h est idempotente : $h(h(x)) = h(x)$

Élément fermé pour h

x est un **élément fermé** ssi $h(x) = x$

Opérateur de fermeture

Exemple : fermeture en coordonnées entières

Soit l'ensemble des points dans le plan en coordonnées réelles $E = \mathbb{R} \times \mathbb{R}$, muni de l'ordre \leq_E avec $(x_1, y_1) \leq_E (x_2, y_2)$ ssi $x_1 \leq x_2$ et $y_1 \leq y_2$

et $h : E \rightarrow E$, avec $h(x, y) = (\lceil x \rceil, \lceil y \rceil)$

où $\lceil x \rceil$ est la partie entière supérieure de x

Exemple : $h(3.2, 6.8) = (4, 7)$

h est un opérateur de fermeture

- h est croissante : $(x_1, y_1) \leq_E (x_2, y_2) \Rightarrow (\lceil x_1 \rceil, \lceil y_1 \rceil) \leq_E (\lceil x_2 \rceil, \lceil y_2 \rceil)$
- h est extensive : $(x_1, y_1) \leq_E (\lceil x_1 \rceil, \lceil y_1 \rceil)$
- h est idempotente : $h(h((x_1, y_1))) = h((\lceil x_1 \rceil, \lceil y_1 \rceil)) = (\lceil x_1 \rceil, \lceil y_1 \rceil) = h((x_1, y_1))$

Élément fermé pour h

(x_1, y_1) est un **élément fermé** si x_1 et y_1 sont des entiers

Opérateur de fermeture

Exemple : fermeture transitive d'une relation

Soit un ensemble E , une relation R sur E est un élément de $P = 2^{E \times E}$.

On munit P de l'ordre \subseteq .

Soit $h : P \longrightarrow P$, avec $h(R) = R_{trans}$

où R_{trans} est la fermeture transitive de R (plus petite relation transitive contenant R) :

(1) $R \subseteq R_{trans}$

(2) si $x, y, z \in E$, $(x, y) \in R$ et $(y, z) \in R$ alors $(x, z) \in R_{trans}$

h est un opérateur de fermeture

- h est croissante : si $R_1 \subseteq R_2$ alors $R_{1_{trans}} \subseteq R_{2_{trans}}$
- h est extensive : $R \subseteq R_{trans}$
- h est idempotente : $h(h(R)) = h(R_{trans}) = R_{trans} = h(R)$

Élément fermé pour h

R est un **élément fermé** si R est transitive.

Exercice : Formaliser la notion de fermeture réflexive d'une relation binaire sur un ensemble E .

Treillis des fermés

Treillis des fermés

Soit h un opérateur de fermeture sur un treillis T et F l'ensemble des fermés de h .
 (F, \wedge_F, \vee_F) est un treillis avec :

$$x \wedge_F y = x \wedge_T y$$

$$x \vee_F y = h(x \vee_T y)$$

Exemple : Treillis des relations binaires transitives sur un ensemble E

Soit h l'opérateur de fermeture sur le treillis $(2^{E \times E}, \subseteq)$ qui associe à une relation binaire sur un ensemble E sa fermeture transitive et F l'ensemble des relations binaires transitives sur E .

(F, \wedge_F, \vee_F) est un treillis et plus précisément :

$$R_1 \wedge_F R_2 = R_1 \cap R_2$$

$$R_1 \vee_F R_2 = h(R_1 \cup R_2) \text{ c'est-à-dire la fermeture transitive de } R_1 \cup R_2$$

Exercice : Dessiner une petite partie du treillis des relations transitives sur l'ensemble $E = \{a, b, c\}$ en partant du top.

Vers des fermetures associées à une relation binaire

Contexte formel (O, A, R)

- O et A sont deux ensembles finis (par exemple représentant des objets et leurs attributs)
- $R \subseteq O \times A$ est une relation binaire (par exemple "possède")

Applications associées à R

- f associe à un ensemble d'objets les attributs qu'ils partagent
 $f : \mathcal{P}(O) \rightarrow \mathcal{P}(A)$
 $X \mapsto f(X) = \{y \in A \mid \forall x \in X, (x, y) \in R\}$
- g associe à un ensemble d'attributs les objets qui les possèdent tous
 $g : \mathcal{P}(A) \rightarrow \mathcal{P}(O)$
 $Y \mapsto g(Y) = \{x \in O \mid \forall y \in Y, (x, y) \in R\}$

Rappel, on note $\mathcal{P}(O)$ ou 2^O l'ensemble des parties de O






Opérateurs de fermeture associés à R

$f \circ g$ et $g \circ f$ sont des opérateurs de fermeture.

$f \circ g$ est un opérateur de fermeture sur $(2^A, \subseteq)$

$g \circ f$ est un opérateur de fermeture sur $(2^O, \subseteq)$

Relation binaire descriptive des animaux

		flying	nocturnal	feathered	migratory	with_crest	with_membrane
	flying squirrel	×					×
	bat	×	×				×
	ostrich			×			
	flamingo	×		×	×		
	chicken	×		×		×	

Concept Formel dans R






Un **concept formel** C est une paire (E, I) telle que
 $f(E) = I$, ou de manière équivalente $E = g(I)$

$E = \{ e \in O \mid \forall i \in I, (e, i) \in R \}$
 est ***l'extension*** (objets couverts)

$I = \{ i \in A \mid \forall e \in E, (e, i) \in R \}$
 est ***l'intension*** (attributs partagés)

E est un fermé de $g \circ f$
 I est un fermé de $f \circ g$

Relation binaire descriptive des animaux

	flyng	nocturnal	feathered	migratory	with_crest	with_membrane
 flying squirrel	×					×
 bat	×	×				×
 ostrich			×			
 flamingo	×		×	×		
 chicken	×		×		×	

(*{flying, feathered}*, *{flamingo, chicken}*) est un concept

(*{flying}*, *{flamingo, chicken}*) n'est pas un concept






(*{flying, feathered}*, *{flamingo}*) n'est pas un concept

Treillis des concepts

L'ensemble de tous les concepts \mathcal{C} forme un treillis \mathcal{L} lorsqu'il est muni de l'ordre suivant :

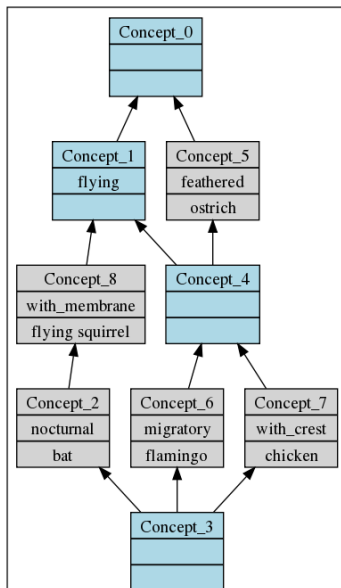
$$(E_1, l_1) \leq_{\mathcal{L}} (E_2, l_2) \Leftrightarrow E_1 \subseteq E_2$$

(ou de manière équivalente $l_2 \subseteq l_1$).

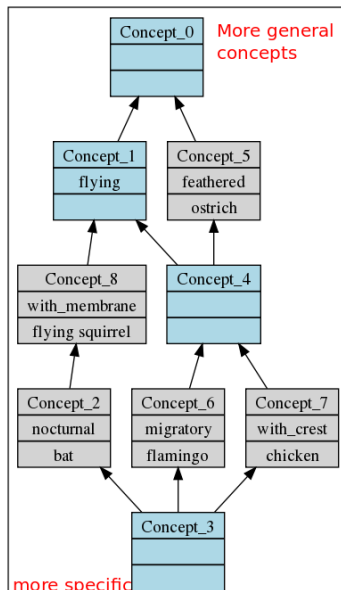
		flying	nocturnal	feathered	migratory	with_crest	with_membrane
	flying squirrel	×					×
	bat	×	×				×
	ostrich			×			
	flamingo	×		×	×		
	chicken	×		×		×	

$(\{flying, feathered\}, \{flamingo, chicken\})$ est un sous-concept de
 $(\{feathered\}, \{flamingo, chicken, ostrich\})$

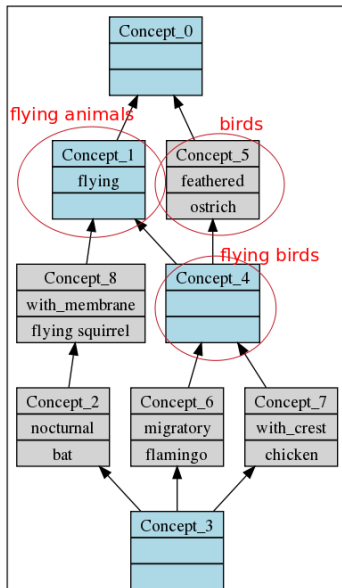
Treillis des animaux



Treillis des animaux



Treillis des animaux



Treillis des concepts de streams C++

Construire le treillis des concepts de la relation suivante qui décrit quelques streams par quelques-unes de leurs opérations

	width	fill	get	put	open	close
io	x					
bio	x	x				
bos	x	x		x		
bis	x	x	x			
bios	x	x	x	x		
bof	x	x		x	x	x
bif	x	x	x		x	x
bf	x	x	x	x	x	x