

# EXTENSION DE COMPACT-TABLE AUX TABLES NÉGATIVES ET CONCISES

JFPC2017

---

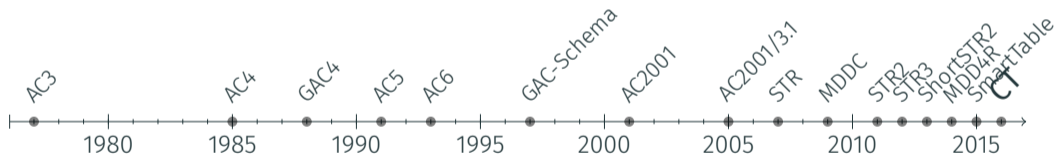
Hélène Verhaeghe<sup>1</sup>, Christophe Lecoutre<sup>2</sup>, Pierre Schaus<sup>3</sup>

14 Juin 2017

<sup>1</sup>UCLouvain, ICTEAM, Place Sainte Barbe 2, 1348 Louvain-la-Neuve, Belgium, *{firstname.lastname}@uclouvain.be*

<sup>2</sup>CRIL-CNRS UMR 8188, Université d'Artois, F-62307 Lens, France, *lecoutre@cril.fr*

Les tables sont les contraintes  
les plus vieilles et les plus utilisées

$$(x, y, z) \in \begin{array}{c|ccc} & x & y & z \\ \hline \tau_1 & a & a & a \\ \tau_2 & d & d & a \\ \tau_3 & c & e & b \\ \vdots & \vdots & \vdots & \vdots \end{array}$$


2016 : Nouvel algorithme! Compact-Table [CP2016], basé sur les opérations bit à bit, devance complètement les algorithmes existants.

# COMPACT-TABLE

---

Une Table

	$x$	$y$	$z$
$\tau_1$	$a$	$a$	$a$
$\tau_2$	$d$	$d$	$a$
$\tau_3$	$c$	$e$	$b$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Une Table

contient des Tuples

	$x$	$y$	$z$
$\tau_1$	$a$	$a$	$a$
$\tau_2$	$d$	$d$	$a$
$\tau_3$	$c$	$e$	$b$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

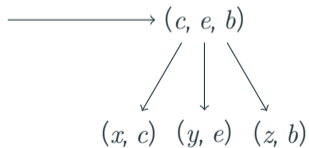
—————→  $(c, e, b)$

# TABLES : DÉFINITION ET VISUALISATION PAR ENSEMBLES

Une Table

contient des Tuples

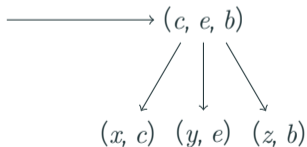
	$x$	$y$	$z$
$\tau_1$	$a$	$a$	$a$
$\tau_2$	$d$	$d$	$a$
$\tau_3$	$c$	$e$	$b$
$\vdots$	$\vdots$	$\vdots$	$\vdots$



Une Table

contient des Tuples

	$x$	$y$	$z$
$\tau_1$	$a$	$a$	$a$
$\tau_2$	$d$	$d$	$a$
$\tau_3$	$c$	$e$	$b$
$\vdots$	$\vdots$	$\vdots$	$\vdots$



Ensemble des Tuples dans la Table

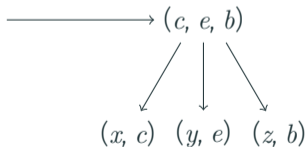
$(x, a)$	$(x, d)$
$(x, b)$	$(x, e)$
$(x, c)$	$(x, f)$

# TABLES : DÉFINITION ET VISUALISATION PAR ENSEMBLES

Une Table

	$x$	$y$	$z$
$\tau_1$	$a$	$a$	$a$
$\tau_2$	$d$	$d$	$a$
$\tau_3$	$c$	$e$	$b$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

contient des Tuples



Ensemble des Tuples dans la Table

$(x, a)$	$\tau$	$(x, d)$
$(x, b)$		$(x, e)$
$(x, c)$		$(x, f)$

Par exemple :  $\tau = (a, b, c)$

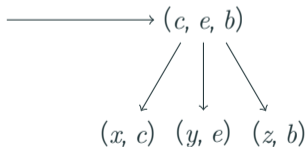


# TABLES : DÉFINITION ET VISUALISATION PAR ENSEMBLES

Une Table

	$x$	$y$	$z$
$\tau_1$	$a$	$a$	$a$
$\tau_2$	$d$	$d$	$a$
$\tau_3$	$c$	$e$	$b$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

contient des Tuples



Ensemble des Tuples dans la Table

$(x, a)$	$(x, d)$
$(x, b)$	$\tau$ $(x, e)$
$(x, c)$	$(x, f)$

Par exemple :  $\tau = (e, c, a)$

1. Quels sont les tuples encore valides?
2. Quels sont les valeurs non supportées?

1. Quels sont les tuples encore valides?

Phase de **mise à jour**

2. Quels sont les valeurs non supportées?

1. Quels sont les tuples encore valides?

Phase de **mise à jour**

2. Quels sont les valeurs non supportées?

Phase de **propagation**

## But de la Mise à jour

Savoir quels sont les tuples encore possible

	Dom	$\Delta$
x	{ a, b, c }	{ }
y	{ a, b, c }	{ }
z	{ a, b, c }	{ }

Etat

	x	y	z	
$\tau_1$	a	a	a	✓
$\tau_2$	a	b	c	✓
$\tau_3$	c	a	b	✓
$\tau_4$	b	c	c	✓
$\tau_5$	a	c	a	✓

Table

$\tau_1$	$\tau_2$	$\tau_3$	$\tau_4$	$\tau_5$
1	1	1	1	1

currTable

## But de la Mise à jour

Savoir quels sont les tuples encore possible

	Dom	$\Delta$
x	{ a, <del>x</del> , c }	{ b }
y	{ a, b, c }	{ }
z	{ a, b, c }	{ }

Etat

	x	y	z	
$\tau_1$	a	a	a	✓
$\tau_2$	a	b	c	✓
$\tau_3$	c	a	b	✓
$\tau_4$	b	c	c	✗
$\tau_5$	a	c	a	✓

Table

$\tau_1$	$\tau_2$	$\tau_3$	$\tau_4$	$\tau_5$
1	1	1	0	1

currTable

## But de la Mise à jour

Savoir quels sont les tuples encore possible

	Dom	$\Delta$
x	{ a, c }	{ b }
y	{ a, b, c }	{ }
z	{ a, b, <del>c</del> }	{ <del>c</del> }

Etat

	x	y	z	
$\tau_1$	a	a	a	✓
$\tau_2$	a	b	c	✗
$\tau_3$	c	a	b	✓
$\tau_4$	b	c	c	
$\tau_5$	a	c	a	✓

Table

$\tau_1$	$\tau_2$	$\tau_3$	$\tau_4$	$\tau_5$
1	0	1	0	1

currTable

## But de la Mise à jour

Savoir quels sont les tuples encore possible

	Dom	$\Delta$
x	<del>{ a, c }</del>	{ a, b }
y	{ a, b, c }	{ }
z	{ a, b }	{ c }

Etat

	x	y	z	
$\tau_1$	a	a	a	<del>X</del>
$\tau_2$	a	b	c	
$\tau_3$	c	a	b	✓
$\tau_4$	b	c	c	
$\tau_5$	a	c	a	<del>X</del>

Table

$\tau_1$	$\tau_2$	$\tau_3$	$\tau_4$	$\tau_5$
0	0	1	0	0

currTable



# SUPPORTS

	$x$	$y$	$z$
$\tau_1$	$a$	$a$	$a$
$\tau_2$	$a$	$b$	$c$
$\tau_3$	$c$	$a$	$b$
$\tau_4$	$b$	$c$	$c$
$\tau_5$	$a$	$c$	$a$

Table

	$\tau_1$	$\tau_2$	$\tau_3$	$\tau_4$	$\tau_5$
$(x,a)$					
$(x,b)$					
$(x,c)$					
$(y,a)$					
$(y,b)$					
$(y,c)$					
$(z,a)$					
$(z,b)$					
$(z,c)$					

support

# SUPPORTS

	x	y	z
$\tau_1$	a	a	a
$\tau_2$	a	b	c
$\tau_3$	c	a	b
$\tau_4$	b	c	c
$\tau_5$	a	c	a

Table

	$\tau_1$	$\tau_2$	$\tau_3$	$\tau_4$	$\tau_5$
$(x,a)$	1				
$(x,b)$	0				
$(x,c)$	0				
$(y,a)$	1				
$(y,b)$	0				
$(y,c)$	0				
$(z,a)$	1				
$(z,b)$	0				
$(z,c)$	0				

support

# SUPPORTS

	x	y	z
$\tau_1$	a	a	a
$\tau_2$	a	b	c
$\tau_3$	c	a	b
$\tau_4$	b	c	c
$\tau_5$	a	c	a

Table

	$\tau_1$	$\tau_2$	$\tau_3$	$\tau_4$	$\tau_5$
$(x,a)$	1	1			
$(x,b)$	0	0			
$(x,c)$	0	0			
$(y,a)$	1	0			
$(y,b)$	0	1			
$(y,c)$	0	0			
$(z,a)$	1	0			
$(z,b)$	0	0			
$(z,c)$	0	1			

support

# SUPPORTS

	x	y	z
$\tau_1$	a	a	a
$\tau_2$	a	b	c
$\tau_3$	c	a	b
$\tau_4$	b	c	c
$\tau_5$	a	c	a

Table

	$\tau_1$	$\tau_2$	$\tau_3$	$\tau_4$	$\tau_5$
$(x,a)$	1	1	0		
$(x,b)$	0	0	0		
$(x,c)$	0	0	1		
$(y,a)$	1	0	1		
$(y,b)$	0	1	0		
$(y,c)$	0	0	0		
$(z,a)$	1	0	0		
$(z,b)$	0	0	1		
$(z,c)$	0	1	0		

support

# SUPPORTS

	x	y	z
$\tau_1$	a	a	a
$\tau_2$	a	b	c
$\tau_3$	c	a	b
$\tau_4$	b	c	c
$\tau_5$	a	c	a

Table

	$\tau_1$	$\tau_2$	$\tau_3$	$\tau_4$	$\tau_5$
$(x,a)$	1	1	0	0	
$(x,b)$	0	0	0	1	
$(x,c)$	0	0	1	0	
$(y,a)$	1	0	1	0	
$(y,b)$	0	1	0	0	
$(y,c)$	0	0	0	1	
$(z,a)$	1	0	0	0	
$(z,b)$	0	0	1	0	
$(z,c)$	0	1	0	1	

support

# SUPPORTS

	x	y	z
$\tau_1$	a	a	a
$\tau_2$	a	b	c
$\tau_3$	c	a	b
$\tau_4$	b	c	c
$\tau_5$	a	c	a

Table

	$\tau_1$	$\tau_2$	$\tau_3$	$\tau_4$	$\tau_5$
$(x,a)$	1	1	0	0	1
$(x,b)$	0	0	0	1	0
$(x,c)$	0	0	1	0	0
$(y,a)$	1	0	1	0	0
$(y,b)$	0	1	0	0	0
$(y,c)$	0	0	0	1	1
$(z,a)$	1	0	0	0	1
$(z,b)$	0	0	1	0	0
$(z,c)$	0	1	0	1	0

support

# SUPPORTS

	x	y	z
$\tau_1$	a	a	a
$\tau_2$	a	b	c
$\tau_3$	c	a	b
$\tau_4$	b	c	c
$\tau_5$	a	c	a

Table

	$\tau_1$	$\tau_2$	$\tau_3$	$\tau_4$	$\tau_5$
$(x,a)$	1	1	0	0	1
$(x,b)$	0	0	0	1	0
$(x,c)$	0	0	1	0	0
$(y,a)$	1	0	1	0	0
$(y,b)$	0	1	0	0	0
$(y,c)$	0	0	0	1	1
$(z,a)$	1	0	0	0	1
$(z,b)$	0	0	1	0	0
$(z,c)$	0	1	0	1	0

support

Ensemble des Tuples

$(x,a)$
$(x,b)$
$(x,c)$

Ensembles

# SUPPORTS

	x	y	z
$\tau_1$	a	a	a
$\tau_2$	a	b	c
$\tau_3$	c	a	b
$\tau_4$	b	c	c
$\tau_5$	a	c	a

Table

	$\tau_1$	$\tau_2$	$\tau_3$	$\tau_4$	$\tau_5$
$(x,a)$	1	1	0	0	1
$(x,b)$	0	0	0	1	0
$(x,c)$	0	0	1	0	0
$(y,a)$	1	0	1	0	0
$(y,b)$	0	1	0	0	0
$(y,c)$	0	0	0	1	1
$(z,a)$	1	0	0	0	1
$(z,b)$	0	0	1	0	0
$(z,c)$	0	1	0	1	0

support

Ensemble des Tuples

$(x,a)$	$\tau_1$	$\tau_2$	$\tau_5$
$(x,b)$			
$(x,c)$			

Ensembles



# SUPPORTS

	x	y	z
$\tau_1$	a	a	a
$\tau_2$	a	b	c
$\tau_3$	c	a	b
$\tau_4$	b	c	c
$\tau_5$	a	c	a

Table

	$\tau_1$	$\tau_2$	$\tau_3$	$\tau_4$	$\tau_5$
$(x,a)$	1	1	0	0	1
$(x,b)$	0	0	0	1	0
$(x,c)$	0	0	1	0	0
$(y,a)$	1	0	1	0	0
$(y,b)$	0	1	0	0	0
$(y,c)$	0	0	0	1	1
$(z,a)$	1	0	0	0	1
$(z,b)$	0	0	1	0	0
$(z,c)$	0	1	0	1	0

support

Ensemble des Tuples

$(x,a)$	
$(x,b)$	
$(x,c)$	$\tau_3$

Ensembles

Ensemble des Tuples dans la Table

$(x,a)$	$(x,d)$
$(x,b)$	$(x,e)$
$(x,c)$	$(x,f)$

**But de la Mise à jour**

Retirer les tuples invalides de `currTable`

Ensemble des Tuples dans la Table

$(x,a)$		$(x,d)$
$(x,b)$	<b>Tuples valides</b>	$(x,e)$
$(x,c)$		$(x,f)$

**But de la Mise à jour**

Retirer les tuples invalides de `currTable`

Ensemble des Tuples dans la Table

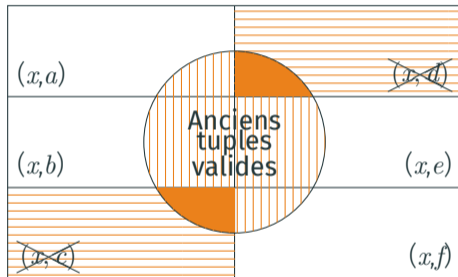
$(x,a)$	<del><math>(x,d)</math></del>
$(x,b)$	$(x,e)$
<del><math>(x,c)</math></del>	$(x,f)$

Anciens tuples valides

But de la Mise à jour

Retirer les tuples invalides de `currTable`

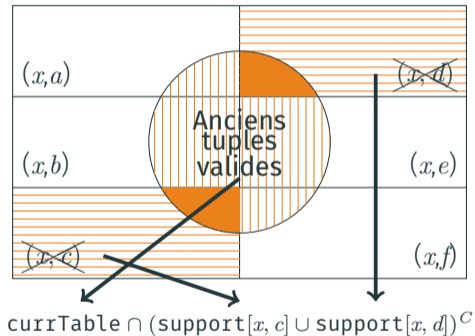
Ensemble des Tuples dans la Table



**But de la Mise à jour**

Retirer les tuples invalides de `currTable`

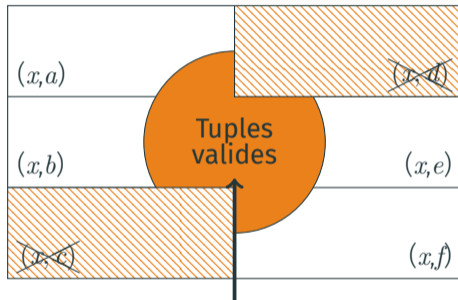
Ensemble des Tuples dans la Table



But de la Mise à jour

Retirer les tuples invalides de `currTable`

Ensemble des Tuples dans la Table

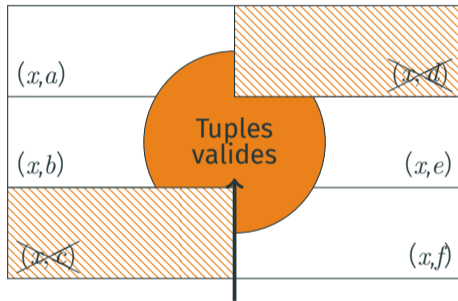


$$\text{currTable} \cap (\text{support}[x, c] \cup \text{support}[x, d])^C$$

## But de la Mise à jour

Retirer les tuples invalides de `currTable`

Ensemble des Tuples dans la Table



$$\text{currTable} \cap (\text{support}[x, c] \cup \text{support}[x, d])^C$$

## But de la Mise à jour

Retirer les tuples invalides de `currTable`


---

**Algorithm:** ClassicalUpdate(x)
 

---

- 1  $\text{mask} \leftarrow 0;$
  - 2 **foreach** value  $a \in \Delta_x$  **do**
  - 3      $\text{mask} \leftarrow \text{mask} \mid \text{supports}[x, a];$
  - 4  $\text{mask} \leftarrow \sim \text{mask};$
  - 5  $\text{currTable} \leftarrow \text{currTable} \& \text{mask};$
-



Ensemble des Tuples dans la Table

$(x,a)$		$(x,d)$
$(x,b)$	<b>Tuples valides</b>	$(x,e)$
$(x,c)$		$(x,f)$

**But de la Mise à jour**

Retirer les tuples invalides de `currTable`

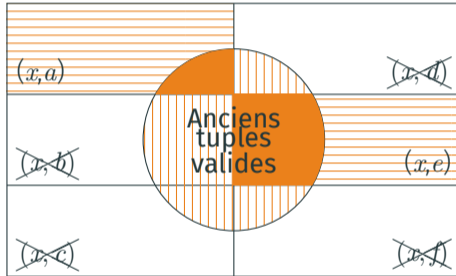
Ensemble des Tuples dans la Table

$(x, a)$		<del><math>(x, d)</math></del>
<del><math>(x, b)</math></del>	Anciens tuples valides	$(x, e)$
<del><math>(x, c)</math></del>		<del><math>(x, f)</math></del>

But de la Mise à jour

Retirer les tuples invalides de `currTable`

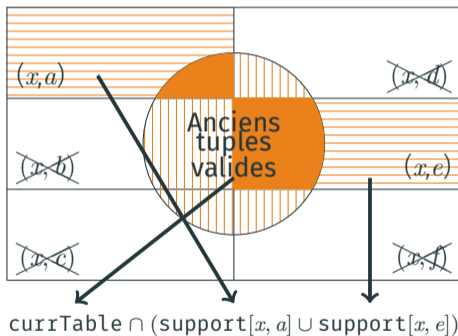
Ensemble des Tuples dans la Table



**But de la Mise à jour**

Retirer les tuples invalides de `currTable`

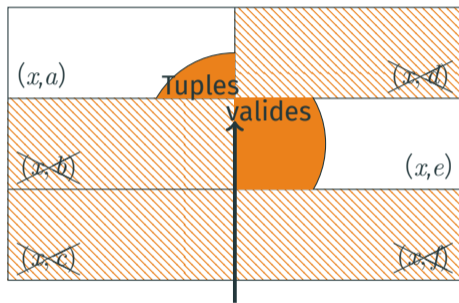
Ensemble des Tuples dans la Table



But de la Mise à jour

Retirer les tuples invalides de `currTable`

Ensemble des Tuples dans la Table

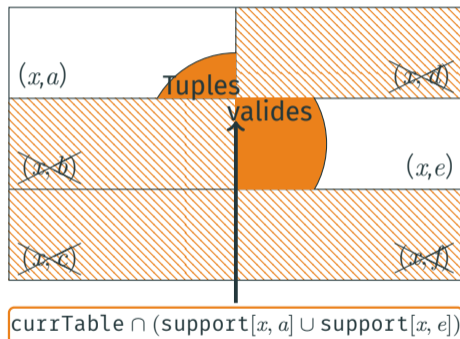


$\text{currTable} \cap (\text{support}[x, a] \cup \text{support}[x, e])$

But de la Mise à jour

Retirer les tuples invalides de `currTable`

Ensemble des Tuples dans la Table

**But de la Mise à jour**Retirer les tuples invalides de `currTable`**Algorithm:** ResetUpdate(x)

- 1  $\text{mask} \leftarrow 0$  ;
- 2 **foreach** value  $a \in \text{dom}(x)$  **do**
- 3      $\text{mask} \leftarrow \text{mask} \mid \text{supports}[x, a]$  ;
- 4  $\text{currTable} \leftarrow \text{currTable} \& \text{mask}$  ;

- Mise à jour classique :

$$\mathcal{O}(|\Delta_x|)$$

- Mise à jour avec reset :

$$\mathcal{O}(|dom(x)|)$$

## But de la Mise à jour

Retirer les tuples invalides de `currTable`

---

### Algorithm: Update(x)

---

```
1 foreach variable  $x \in scp$  do
2   if  $|\Delta_x| < |dom(x)|$  then
3     ClassicalUpdate(x);
4   else
5     ResetUpdate(x);
```

---

Ensemble des Tuples dans la Table

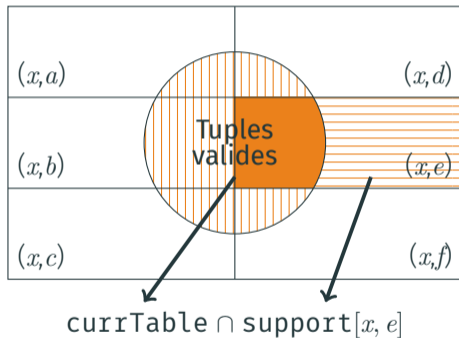
$(x, a)$		$(x, d)$
$(x, b)$	<b>Tuples valides</b>	$(x, e)$
$(x, c)$		$(x, f)$

### But de la propagation

Retirer les valeurs non supportées

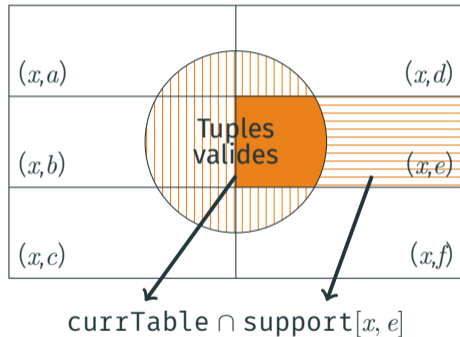


Ensemble des Tuples dans la Table

**But de la propagation**

Retirer les valeurs non supportées

Ensemble des Tuples dans la Table

**But de la propagation**

Retirer les valeurs non supportées

**Algorithm: Propagate()**


---

```

1 foreach variable  $x \in \text{scp}$  do
2   |   foreach value  $a \in \text{dom}(x)$  do
3     |   |   if  $\text{currTable} \& \text{supports}[x, a] = 0$ 
4     |   |   |   then
4     |   |   |   |    $\text{dom}(x) \leftarrow \text{dom}(x) \setminus \{a\}$ ;

```

---

CT\* : CT POUR TABLES CONCISES

---

Une Table Concise

	<i>x</i>	<i>y</i>	<i>z</i>
$\tau_1$	<i>a</i>	<i>b</i>	*
$\tau_2$	<i>a</i>	<i>c</i>	<i>a</i>
$\tau_3$	*	<i>a</i>	<i>a</i>
$\tau_4$	<i>c</i>	*	*

Une Table Concise

contient des Tuples Concis

	x	y	z
$\tau_1$	a	b	*
$\tau_2$	a	c	a
$\tau_3$	*	a	a
$\tau_4$	c	*	*

→ (c, \*, \*)

Une Table Concise

contient des Tuples Concis

	x	y	z
$\tau_1$	a	b	*
$\tau_2$	a	c	a
$\tau_3$	*	a	a
$\tau_4$	c	*	*



$(c, *, *)$



valeurs universelles

Une Table Concise

	x	y	z
$\tau_1$	a	b	*
$\tau_2$	a	c	a
$\tau_3$	*	a	a
$\tau_4$	c	*	*

contient des Tuples Concis

$(c, *, *)$

valeurs universelles

représentants  
plusieurs tuples

$(c, a, a)$

$(c, a, b)$

$(c, a, c)$

$(c, b, a)$

$(c, b, b)$

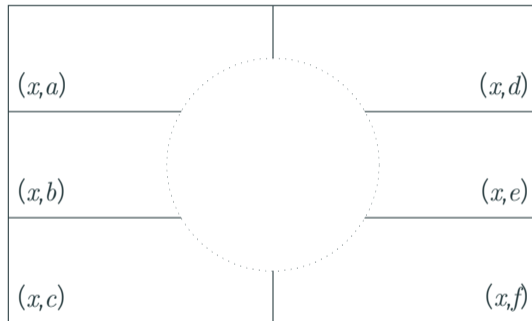
$(c, b, c)$

$(c, c, a)$

$(c, c, b)$

$(c, c, c)$

Ensemble des Tuples dans la Table



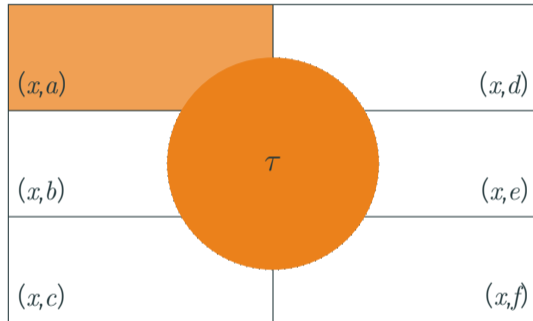


Ensemble des Tuples dans la Table

$(x,a)$	$\tau$	$(x,d)$
$(x,b)$		$(x,e)$
$(x,c)$		$(x,f)$

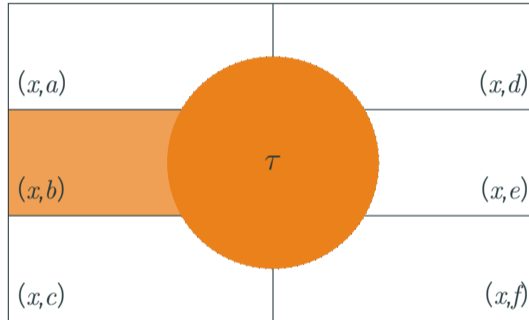
Par exemple :  $\tau = (a, b, c)$

Ensemble des Tuples dans la Table



Par exemple :  $\tau = (*, c, a)$

Ensemble des Tuples dans la Table



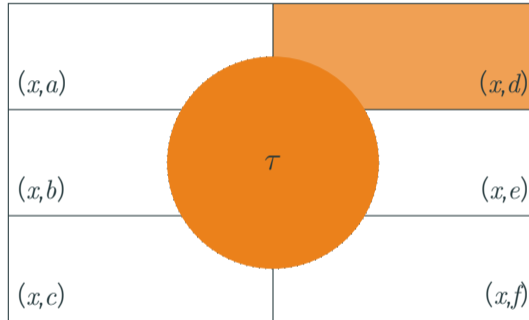
Par exemple :  $\tau = (*, c, a)$

Ensemble des Tuples dans la Table

$(x,a)$		$(x,d)$
$(x,b)$	$\tau$	$(x,e)$
$(x,c)$		$(x,f)$

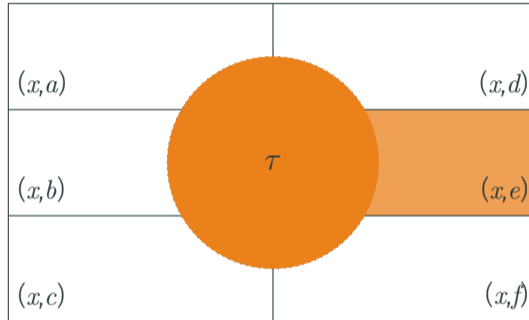
Par exemple :  $\tau = (*, c, a)$

Ensemble des Tuples dans la Table



Par exemple :  $\tau = (*, c, a)$

Ensemble des Tuples dans la Table



Par exemple :  $\tau = (*, c, a)$

Ensemble des Tuples dans la Table

$(x,a)$	$(x,d)$
$(x,b)$	$(x,e)$
$(x,c)$	$(x,f)$

Par exemple :  $\tau = (*, c, a)$

Ensemble des Tuples dans la Table

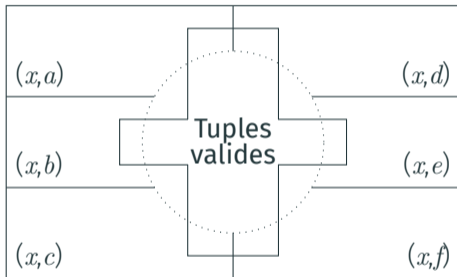
$(x, a)$		$(x, d)$
$(x, b)$		$(x, e)$
$(x, c)$		$(x, f)$

**But de la Mise à jour**

Retirer les tuples invalides de `currTable`



Ensemble des Tuples dans la Table



**But de la Mise à jour**

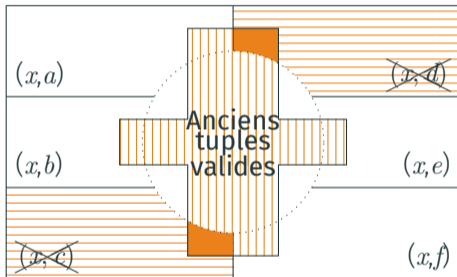
Retirer les tuples invalides de `currTable`

Ensemble des Tuples dans la Table

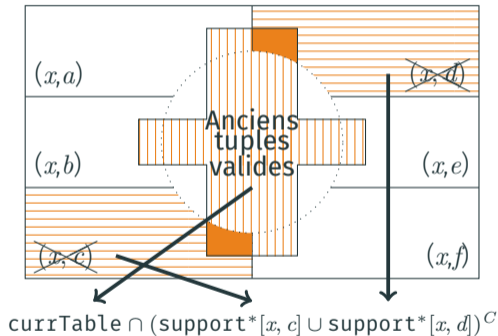
$(x, a)$		<del><math>(x, d)</math></del>
$(x, b)$	Anciens tuples valides	$(x, e)$
<del><math>(x, c)</math></del>		$(x, f)$

**But de la Mise à jour**Retirer les tuples invalides de `currTable`

Ensemble des Tuples dans la Table

**But de la Mise à jour**Retirer les tuples invalides de `currTable`

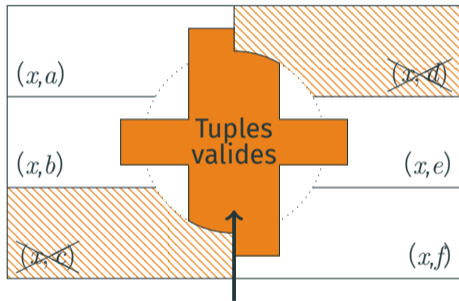
Ensemble des Tuples dans la Table



But de la Mise à jour

Retirer les tuples invalides de `currTable`

Ensemble des Tuples dans la Table

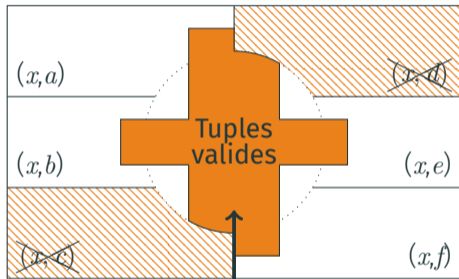


$$\text{currTable} \cap (\text{support}^*[x, c] \cup \text{support}^*[x, d])^C$$

But de la Mise à jour

Retirer les tuples invalides de `currTable`

Ensemble des Tuples dans la Table



$$\text{currTable} \cap (\text{support}^*[x, c] \cup \text{support}^*[x, d])^C$$

## But de la Mise à jour

Retirer les tuples invalides de `currTable`

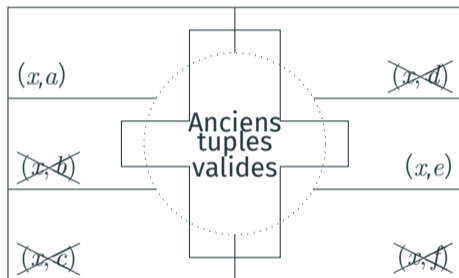

---

**Algorithm:** ClassicalUpdate(x)
 

---

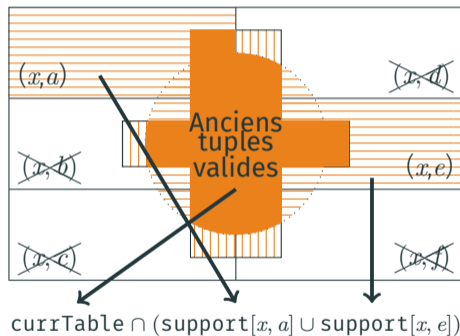
- 1 `mask`  $\leftarrow$  0 ;
  - 2 **foreach** value  $a \in \Delta_x$  **do**
  - 3   `mask`  $\leftarrow$  `mask` | `support`\*[ $x, a$ ] ;
  - 4 `mask`  $\leftarrow$   $\sim$  `mask` ;
  - 5 `currTable`  $\leftarrow$  `currTable` & `mask` ;
-

Ensemble des Tuples dans la Table

**But de la Mise à jour**Retirer les tuples invalides de `currTable`**Algorithm:** ResetUpdate(x)

- 1 `mask`  $\leftarrow$  0 ;
- 2 **foreach** value  $a \in dom(x)$  **do**
- 3     `mask`  $\leftarrow$  `mask` | `supports`[ $x, a$ ] ;
- 4 `currTable`  $\leftarrow$  `currTable` & `mask` ;

Ensemble des Tuples dans la Table



But de la Mise à jour

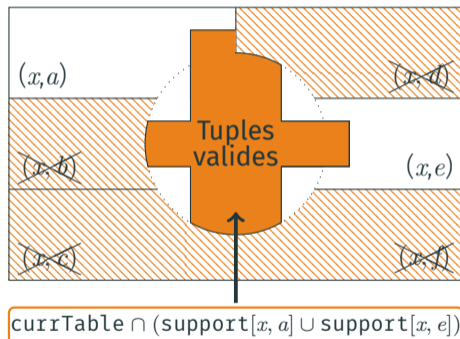
Retirer les tuples invalides de currTable

Algorithm: ResetUpdate(x)

- 1 mask  $\leftarrow$  0 ;
- 2 foreach value  $a \in \text{dom}(x)$  do
- 3     | mask  $\leftarrow$  mask | supports[x, a] ;
- 4 currTable  $\leftarrow$  currTable & mask ;

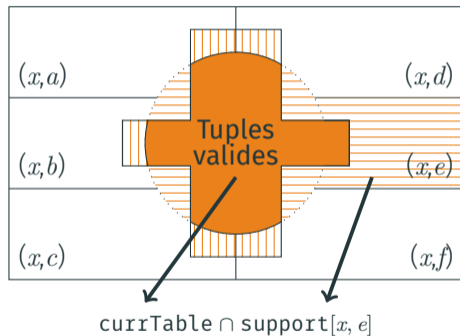


Ensemble des Tuples dans la Table

**But de la Mise à jour**Retirer les tuples invalides de `currTable`**Algorithm:** ResetUpdate(x)

- 1 `mask`  $\leftarrow$  0 ;
- 2 **foreach** value  $a \in \text{dom}(x)$  **do**
- 3     `mask`  $\leftarrow$  `mask` | `supports`[ $x, a$ ] ;
- 4 `currTable`  $\leftarrow$  `currTable` & `mask` ;

Ensemble des Tuples dans la Table

**But de la propagation**

Retirer les valeurs non supportées

**Algorithm: Propagate()**

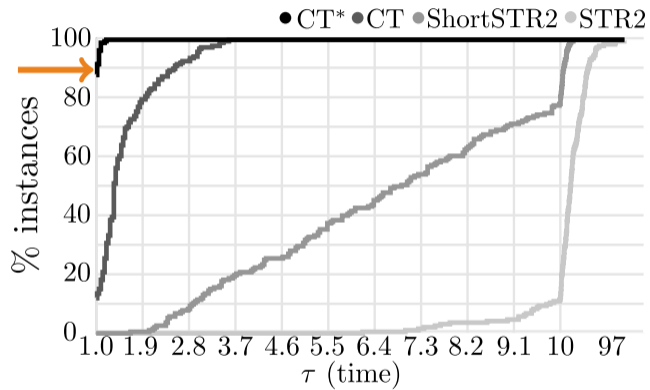

---

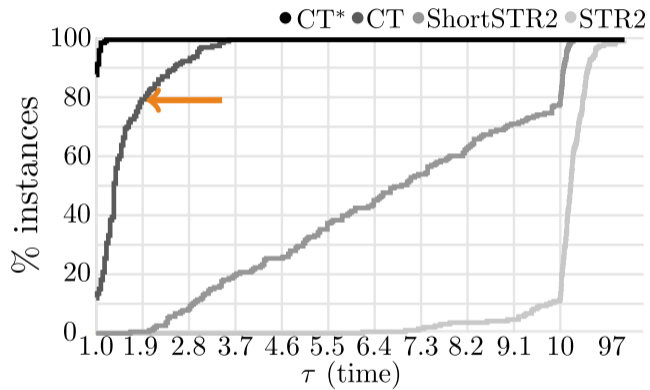
```

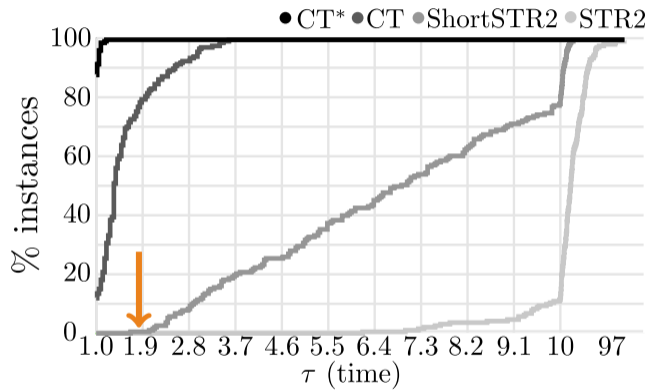
1 foreach variable  $x \in \text{scp}$  do
2   |   foreach value  $a \in \text{dom}(x)$  do
3     |   |   if  $\text{currTable} \ \& \ \text{supports}[x, a] = 0$ 
4     |   |   |   then
4     |   |   |   |    $\text{dom}(x) \leftarrow \text{dom}(x) \setminus \{a\}$ ;

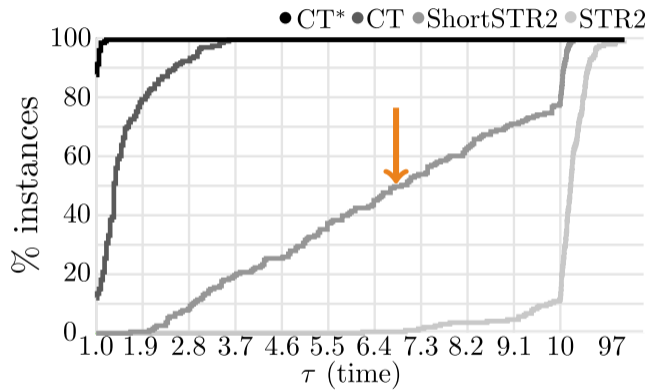
```

---









$CT_{neg}$  : CT POUR TABLES NÉGATIVES

---

Une Table Négative

	$x$	$y$	$z$
$\tau_1$	$a$	$a$	$a$
$\tau_2$	$d$	$d$	$a$
$\tau_3$	$c$	$e$	$b$
$\vdots$	$\vdots$	$\vdots$	$\vdots$



Une Table Négative

contient des Tuples

	$x$	$y$	$z$
$\tau_1$	$a$	$a$	$a$
$\tau_2$	$d$	$d$	$a$
$\tau_3$	$c$	$e$	$b$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

→  $(c, e, b)$

# TABLES NÉGATIVES : DÉFINITION

Une Table Négative

contient des Tuples

qui ne sont pas possibles

	$x$	$y$	$z$
$\tau_1$	$a$	$a$	$a$
$\tau_2$	$d$	$d$	$a$
$\tau_3$	$c$	$e$	$b$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

→  $(c, e, b)$

$x = c, y = e, z = b$

→ n'apparaît dans  
aucune solution

currTable  $\rightarrow$  Tuples dangereux

support[x,a]  $\rightarrow$  Tuples ayant  $(x,a)$

### But de la Mise à jour

Retirer les tuples non dangereux de currTable

---

#### Algorithm: Update(x)

---

```
1 foreach variable  $x \in \text{scp}$  do
2   | if  $|\Delta_x| < |\text{dom}(x)|$  then
3     | ClassicalUpdate(x);
4   else
5     | ResetUpdate(x);
```

---

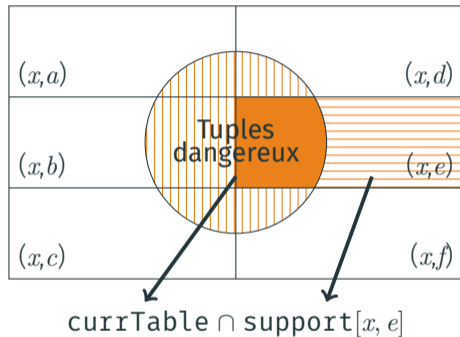
Ensemble des Tuples dans la Table

$(x, a)$		$(x, d)$
$(x, b)$	<b>Tuples dangereux</b>	$(x, e)$
$(x, c)$		$(x, f)$

## But de la propagation

Retirer les valeurs non supportées

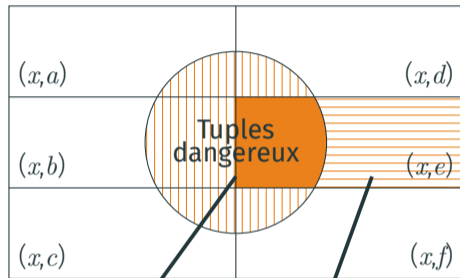
Ensemble des Tuples dans la Table



**But de la propagation**

Retirer les valeurs non supportées

Ensemble des Tuples dans la Table



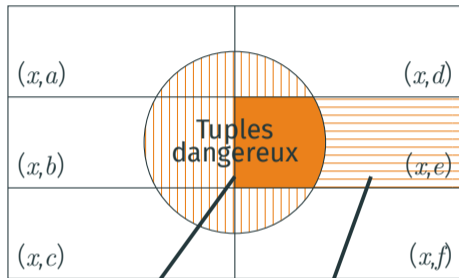
$\text{currTable} \cap \text{support}[x, e]$

$\begin{matrix} \nearrow \\ \text{=} \\ \text{?} \\ \text{---} \\ (e, *, *) \end{matrix}$

### But de la propagation

Retirer les valeurs non supportées

Ensemble des Tuples dans la Table



$\text{currTable} \cap \text{support}[x, e]$

$=?$   
 $(e, *, *)$

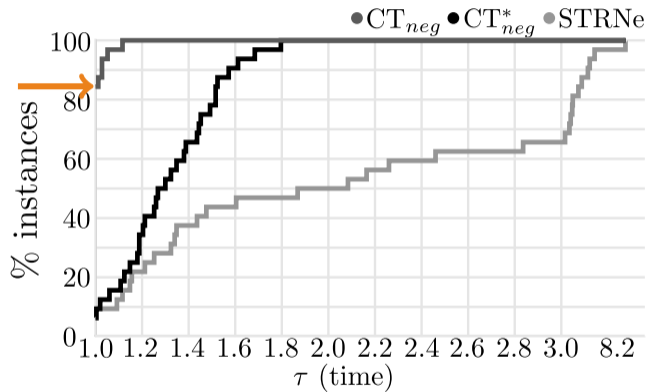
## But de la propagation

Retirer les valeurs non supportées

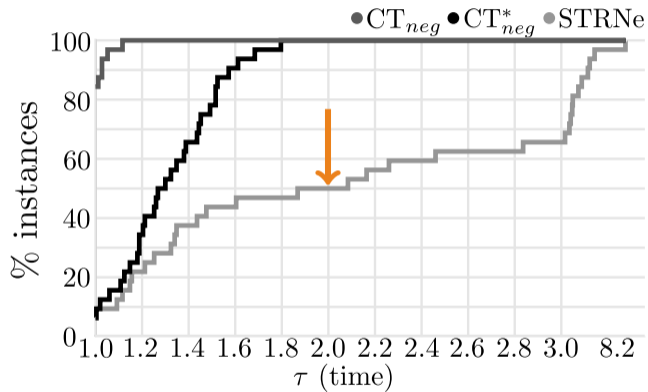
### Algorithm: Propagate()

```

1 foreach variable  $x \in \text{scp}$  do
2    $T \leftarrow \prod_{y \in \text{scp}: y \neq x} |\text{dom}(y)|$ ;
3   foreach value  $a \in \text{dom}(x)$  do
4      $S \leftarrow \text{currTable} \ \& \ \text{supports}[x, a]$ ;
5     if  $\text{bitCount}(S) == T$  then
6        $\text{dom}(x) \leftarrow \text{dom}(x) \setminus \{a\}$ ;
  
```







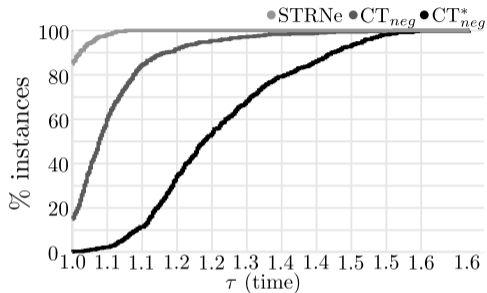
$CT_{neg}^*$  : CT POUR TABLES NÉGATIVES CONCISES

---

## 1. Mise à jour :

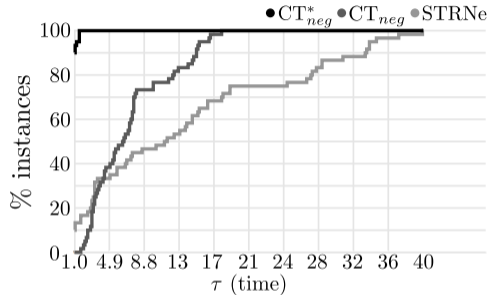
- classique : Comme  $CT^*$
- par reset : Comme  $CT$

## 2. Propagation : Comme $CT_{neg}$ (mais comptage un peu différent)



Petits domaines

(ralentissement dû au comptage, au maximum  
1.6× plus lent)



Grands domaines

(minimum 5× plus rapide pour 60% des  
instances)

# CONCLUSION

---

