

EXTENDING COMPACT-TABLE TO BASIC SMART TABLES

JFPC18

Hélène Verhaeghe¹, Christophe Lecoutre², Yves Deville¹, Pierre Schaus³

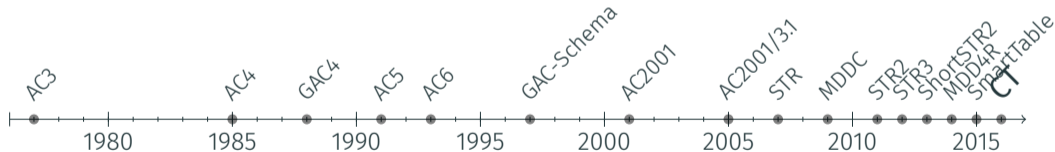
12 Juin 2018

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

²CRIL-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

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


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

COMPACT-TABLE [CP2016]

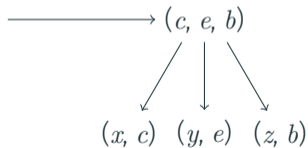
Une Table

	x	y	z
τ_1	a	a	a
τ_2	c	e	b
τ_3	d	d	a
\vdots	\vdots	\vdots	\vdots

Une Table

contient des Tuples

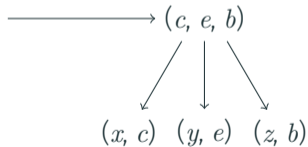
	x	y	z
τ_1	a	a	a
τ_2	c	e	b
τ_3	d	d	a
\vdots	\vdots	\vdots	\vdots



Une Table

contient des Tuples

	x	y	z
τ_1	a	a	a
τ_2	c	e	b
τ_3	d	d	a
\vdots	\vdots	\vdots	\vdots



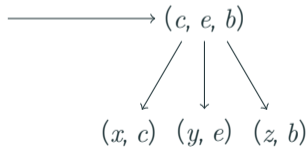
Ensemble des Tuples dans la Table

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

Une Table

contient des Tuples

	x	y	z
τ_1	a	a	a
τ_2	c	e	b
τ_3	d	d	a
\vdots	\vdots	\vdots	\vdots



Ensemble des Tuples dans la Table

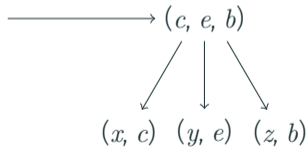
(x, a)	τ	(x, d)
(x, b)		(x, e)
(x, c)		(x, f)

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

Une Table

	x	y	z
τ_1	a	a	a
τ_2	c	e	b
τ_3	d	d	a
\vdots	\vdots	\vdots	\vdots

contient des Tuples



Ensemble des Tuples dans la Table

(x, a)	(x, d)
(x, b)	τ (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	Δ
x	{ a, b, c }	{ }
y	{ a, b, c }	{ }
z	{ a, b, c }	{ }

Etat

	x	y	z	
τ_1	a	a	a	✓
τ_2	a	b	c	✓
τ_3	c	a	b	✓
τ_4	b	c	c	✓
τ_5	a	c	a	✓

Table

τ_1	τ_2	τ_3	τ_4	τ_5
1	1	1	1	1

currTable

But de la Mise à jour

Savoir quels sont les tuples encore possible

	Dom	Δ
x	{ a, x , c }	{ b }
y	{ a, b, c }	{ }
z	{ a, b, c }	{ }

Etat

	x	y	z	
τ_1	a	a	a	✓
τ_2	a	b	c	✓
τ_3	c	a	b	✓
τ_4	b	c	c	✗
τ_5	a	c	a	✓

Table

τ_1	τ_2	τ_3	τ_4	τ_5
1	1	1	0	1

currTable

But de la Mise à jour

Savoir quels sont les tuples encore possible

	Dom	Δ
x	{ a, c }	{ b }
y	{ a, b, c }	{ }
z	{ a, b, c }	{ c }

Etat

	x	y	z	
τ_1	a	a	a	✓
τ_2	a	b	c	✗
τ_3	c	a	b	✓
τ_4	b	c	c	
τ_5	a	c	a	✓

Table

τ_1	τ_2	τ_3	τ_4	τ_5
1	0	1	0	1

currTable

But de la Mise à jour

Savoir quels sont les tuples encore possible

	Dom	Δ
x	{ a, c }	{ a, b }
y	{ a, b, c }	{ }
z	{ a, b }	{ c }

Etat

	x	y	z	
τ_1	a	a	a	X
τ_2	a	b	c	
τ_3	c	a	b	✓
τ_4	b	c	c	
τ_5	a	c	a	X

Table

τ_1	τ_2	τ_3	τ_4	τ_5
0	0	1	0	0

currTable

	x	y	z
τ_1	a	a	a
τ_2	a	b	c
τ_3	c	a	b
τ_4	b	c	c
τ_5	a	c	a

Table

	τ_1	τ_2	τ_3	τ_4	τ_5
(x,a)					
(x,b)					
(x,c)					
(y,a)					
(y,b)					
(y,c)					
(z,a)					
(z,b)					
(z,c)					

support

	x	y	z
τ_1	a	a	a
τ_2	a	b	c
τ_3	c	a	b
τ_4	b	c	c
τ_5	a	c	a

Table

	τ_1	τ_2	τ_3	τ_4	τ_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

	x	y	z
τ_1	a	a	a
τ_2	a	b	c
τ_3	c	a	b
τ_4	b	c	c
τ_5	a	c	a

Table

	τ_1	τ_2	τ_3	τ_4	τ_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

	x	y	z
τ_1	a	a	a
τ_2	a	b	c
τ_3	c	a	b
τ_4	b	c	c
τ_5	a	c	a

Table

	τ_1	τ_2	τ_3	τ_4	τ_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

	x	y	z
τ_1	a	a	a
τ_2	a	b	c
τ_3	c	a	b
τ_4	b	c	c
τ_5	a	c	a

Table

	τ_1	τ_2	τ_3	τ_4	τ_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

	x	y	z
τ_1	a	a	a
τ_2	a	b	c
τ_3	c	a	b
τ_4	b	c	c
τ_5	a	c	a

Table

	τ_1	τ_2	τ_3	τ_4	τ_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

	x	y	z
τ_1	a	a	a
τ_2	a	b	c
τ_3	c	a	b
τ_4	b	c	c
τ_5	a	c	a

Table

	τ_1	τ_2	τ_3	τ_4	τ_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

	x	y	z
τ_1	a	a	a
τ_2	a	b	c
τ_3	c	a	b
τ_4	b	c	c
τ_5	a	c	a

Table

	τ_1	τ_2	τ_3	τ_4	τ_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)	τ_1	τ_2	τ_5
(x,b)			
(x,c)			

Ensembles

	x	y	z
τ_1	a	a	a
τ_2	a	b	c
τ_3	c	a	b
τ_4	b	c	c
τ_5	a	c	a

Table

	τ_1	τ_2	τ_3	τ_4	τ_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)	τ_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)	Tuples valides	(x,e)
(x,c)		(x,f)

But de la Mise à jour

Retirer les tuples invalides de `currTable`

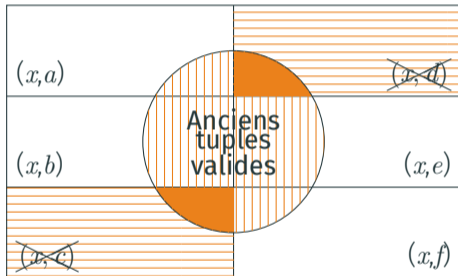
Ensemble des Tuples dans la Table

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

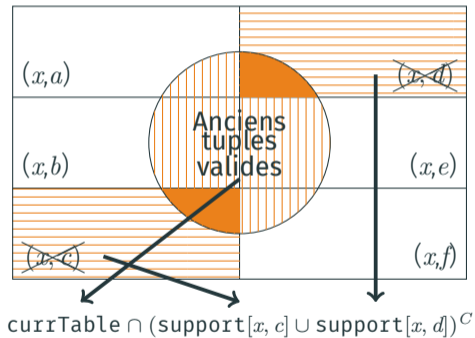
Anciens tuples valides

But de la Mise à jourRetirer les tuples invalides de `currTable`

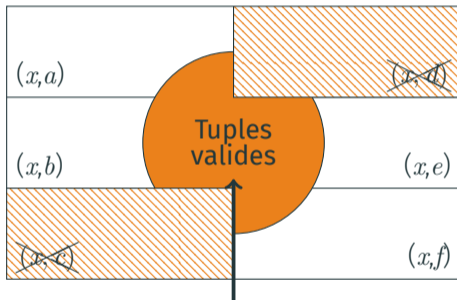
Ensemble des Tuples dans la Table

**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

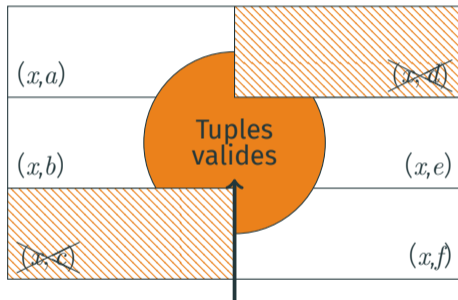


$$\text{currTable} \cap (\text{support}[x, c] \cup \text{support}[x, d])^C$$

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 jourRetirer les tuples invalides de `currTable`**Algorithm:** ClassicalUpdate(x)

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

Ensemble des Tuples dans la Table

(x,a)		(x,d)
(x,b)	Tuples valides	(x,e)
(x,c)		(x,f)

But de la Mise à jour

Retirer les tuples invalides de `currTable`

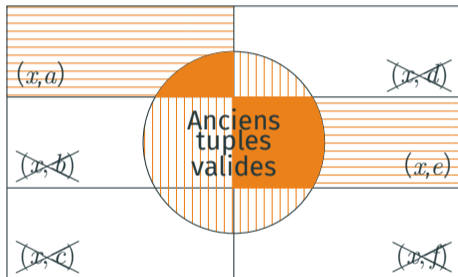
Ensemble des Tuples dans la Table

(x, a)		(x, d)
(x, b)	Anciens tuples valides	(x, e)
(x, c)		(x, f)

But de la Mise à jour

Retirer les tuples invalides de `currTable`

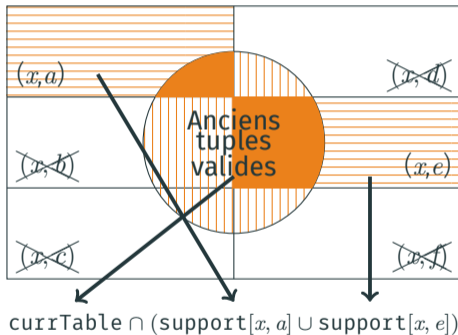
Ensemble des Tuples dans la Table



But de la Mise à jour

Retirer les tuples invalides de `currTable`

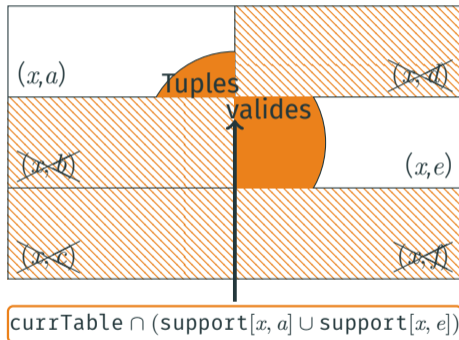
Ensemble des Tuples dans la Table



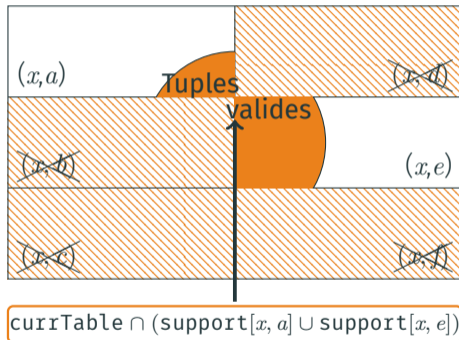
But de la Mise à jour

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`

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` ;
-

- 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);
```

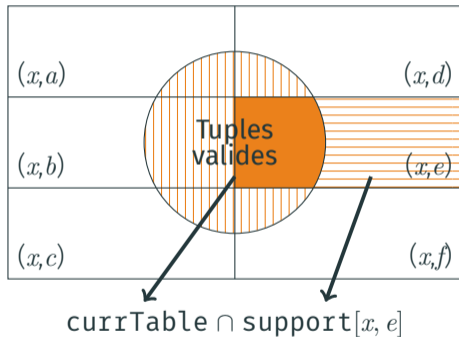
Ensemble des Tuples dans la Table

(x,a)		(x,d)
(x,b)	Tuples valides	(x,e)
(x,c)		(x,f)

But de la propagation

Retirer les valeurs non supportées

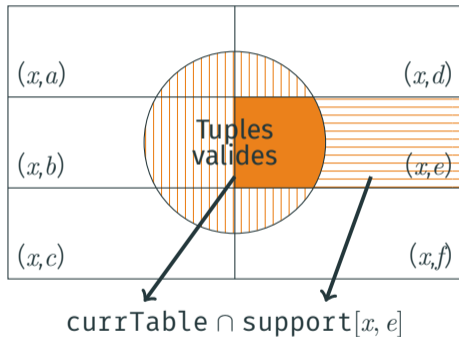
Ensemble des Tuples dans la Table



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
5          $\text{dom}(x) \leftarrow \text{dom}(x) \setminus \{a\}$ ;

```

COMPACT-TABLE POUR LES TABLES SIMPLEMENT INTELLIGENTES

Une Table
Simplément Intelligentes

	x	y	z
τ_1	*	*	$\in \{a, b\}$
τ_2	$\neq a$	c	$\leq a$
τ_3	b	*	*
τ_4	$\geq c$	$\neq b$	a
	\vdots	\vdots	\vdots

Une Table
Simplément Intelligentes

contient des
Elements Intelligents

représentant de
multiples valeurs

	x	y	z
τ_1	*	*	$\in \{a, b\}$
τ_2	$\neq a$	c	$\leq a$
τ_3	b	*	*
τ_4	$\geq c$	$\neq b$	a
	\vdots	\vdots	\vdots

Une Table
Simplement Intelligentes

contient des
Elements Intelligents

représentant de
multiples valeurs

	x	y	z
τ_1	*	*	$\in \{a, b\}$
τ_2	$\neq a$	c	$\leq a$
τ_3	b	*	*
τ_4	$\geq c$	$\neq b$	a
	\vdots	\vdots	\vdots

valeur simple: e

~~a~~ , ~~b~~ , ~~c~~ , ~~d~~ , e , ~~f~~

Une Table
Simplement Intelligentes

contient des
Elements Intelligents

représentant de
multiples valeurs

	x	y	z
τ_1	*	*	$\in \{a, b\}$
τ_2	$\neq a$	c	$\leq a$
τ_3	b	*	*
τ_4	$\geq c$	$\neq b$	a
	\vdots	\vdots	\vdots

valeur simple: e

valeur universelle: *

~~a~~ , ~~b~~ , ~~c~~ , ~~d~~ , **e** , ~~f~~

a , **b** , **c** , **d** , **e** , **f**

Une Table
Simplément Intelligentes

contient des
Elements Intelligents

représentant de
multiples valeurs

	x	y	z
τ_1	*	*	$\in \{a, b\}$
τ_2	$\neq a$	c	$\leq a$
τ_3	b	*	*
τ_4	$\geq c$	$\neq b$	a
	\vdots	\vdots	\vdots

valeur simple: e

valeur universelle: *

exclusion: $\neq e$

~~a~~, ~~b~~, ~~c~~, ~~d~~, **e**, ~~f~~

a, **b**, **c**, **d**, **e**, **f**

a, **b**, **c**, **d**, ~~e~~, **f**

Une Table
Simplément Intelligentes

contient des
Elements Intelligents

représentant de
multiples valeurs

	x	y	z
τ_1	*	*	$\in \{a, b\}$
τ_2	$\neq a$	c	$\leq a$
τ_3	b	*	*
τ_4	$\geq c$	$\neq b$	a
	\vdots	\vdots	\vdots

valeur simple: e

valeur universelle: *

exclusion: $\neq e$

borne supérieure: $\leq c$

~~a~~, ~~b~~, ~~c~~, ~~d~~, **e**, ~~f~~

a, **b**, **c**, **d**, **e**, **f**

a, **b**, **c**, **d**, ~~e~~, **f**

a, **b**, **c**, ~~d~~, ~~e~~, ~~f~~

Une Table
Simplement Intelligentes

contient des
Elements Intelligents

représentant de
multiples valeurs

	x	y	z
τ_1	*	*	$\in \{a, b\}$
τ_2	$\neq a$	c	$\leq a$
τ_3	b	*	*
τ_4	$\geq c$	$\neq b$	a
	\vdots	\vdots	\vdots

valeur simple: e

valeur universelle: *

exclusion: $\neq e$

borne supérieure: $\leq c$

borne inférieure: $\geq c$

~~a~~, ~~b~~, ~~c~~, ~~d~~, **e**, ~~f~~

a, **b**, **c**, **d**, **e**, **f**

a, **b**, **c**, **d**, ~~e~~, **f**

a, **b**, **c**, ~~d~~, ~~e~~, ~~f~~

~~a~~, ~~b~~, **c**, **d**, **e**, **f**

Une Table
Simplément Intelligentes

contient des
Elements Intelligents

représentant de
multiples valeurs

	x	y	z
τ_1	*	*	$\in \{a, b\}$
τ_2	$\neq a$	c	$\leq a$
τ_3	b	*	*
τ_4	$\geq c$	$\neq b$	a
	\vdots	\vdots	\vdots

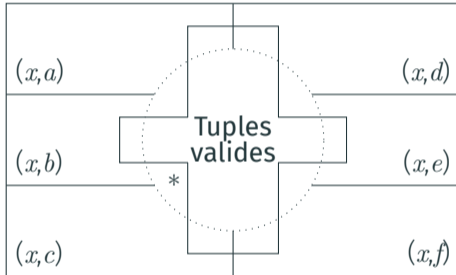
valeur simple: e
 valeur universelle: *
 exclusion: $\neq e$
 borne supérieure: $\leq c$
 borne inférieure: $\geq c$
 ensemble: $\in \{a, c, d\}$



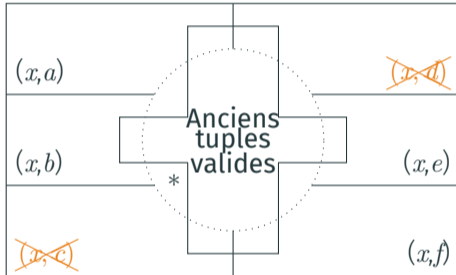
Ensemble des Tuples dans la Table

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

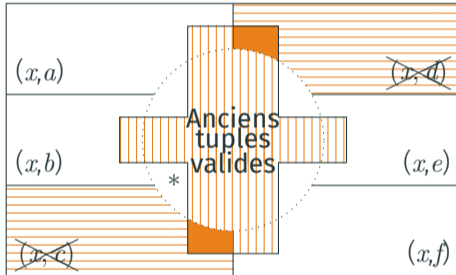
Ensemble des Tuples dans la Table



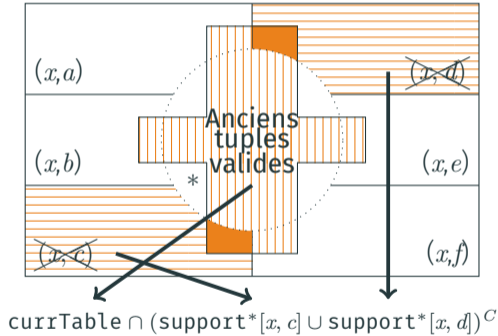
Ensemble des Tuples dans la Table



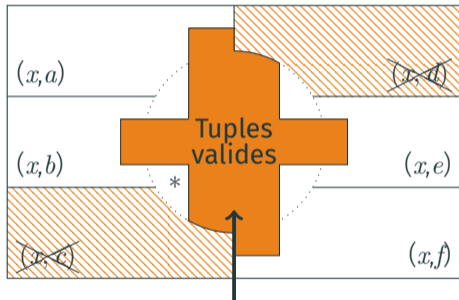
Ensemble des Tuples dans la Table



Ensemble des Tuples dans la Table

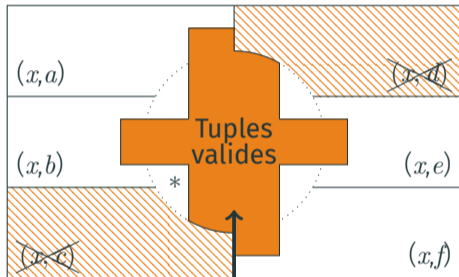


Ensemble des Tuples dans la Table



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

Ensemble des Tuples dans la Table



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

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 ;
-

$$|\text{dom}(x)| == 0$$

$$|\text{dom}(x)| > 1$$

$$|\text{dom}(x)| == 1$$

$$|dom(x)| == 0$$

Trivial!
Variable x s'en charge

$$|dom(x)| == 1$$

$$|dom(x)| > 1$$

$$|dom(x)| == 0$$

Trivial!
Variable x s'en charge

$$|dom(x)| > 1$$

$$|dom(x)| == 1$$

$|\Delta_x| \geq |dom(x)|$ toujours vrai!
ResetUpdate(x) utilisée
et déjà fonctionnelle!

$$|dom(x)| == 0$$

Trivial!
Variable x s'en charge

$$|dom(x)| == 1$$

$|\Delta_x| \geq |dom(x)|$ toujours vrai!
ResetUpdate(x) utilisée
et déjà fonctionnelle!

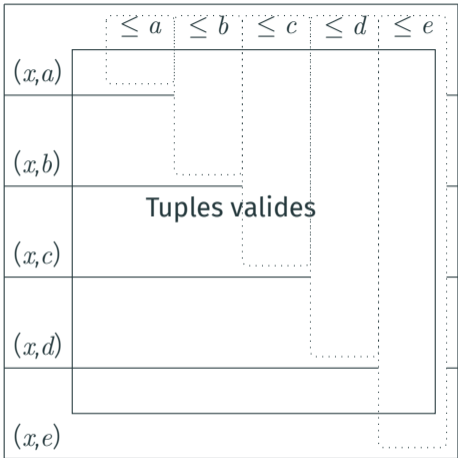
$$|dom(x)| > 1$$

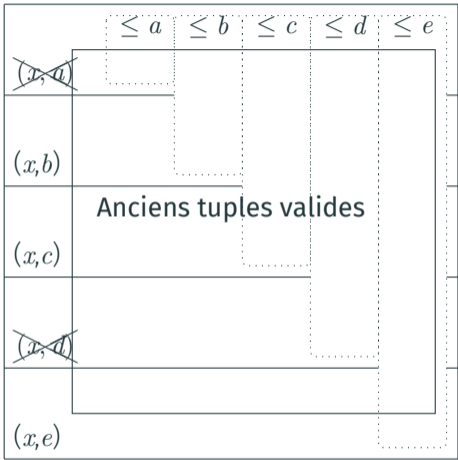
Si $|\Delta_x| < |dom(x)|$

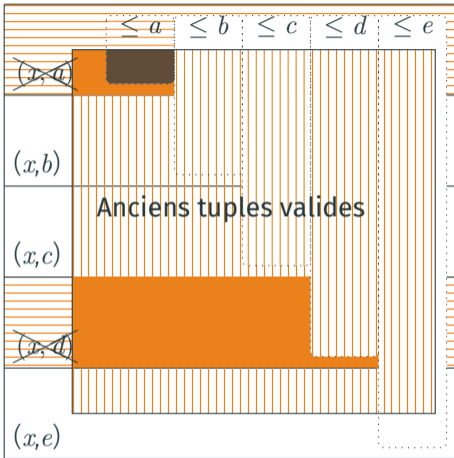
Le tuple est toujours valide!
Au moins une valeur valide
support*[x][τ] = 0

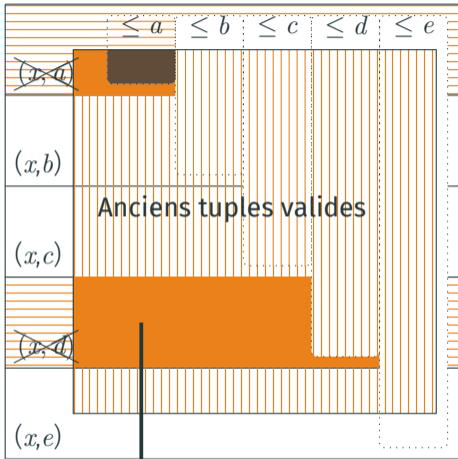
If $|\Delta_x| \geq |dom(x)|$

ResetUpdate(x) utilisée
et déjà fonctionnelle!

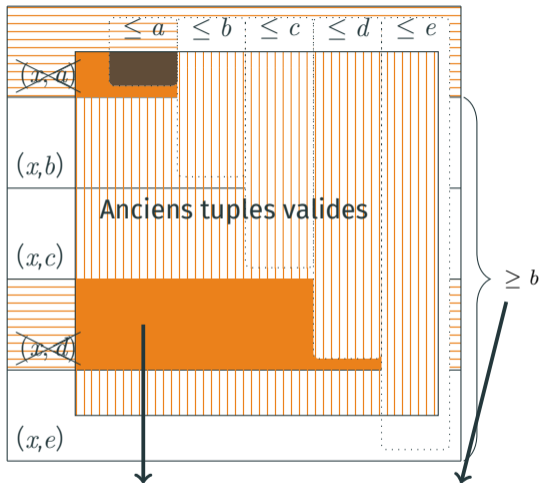




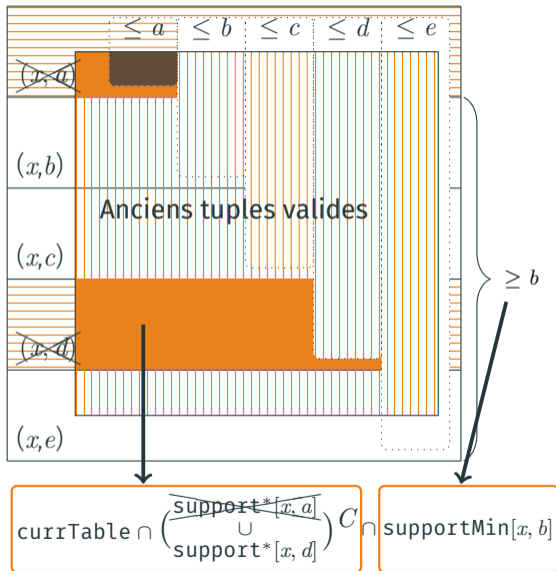


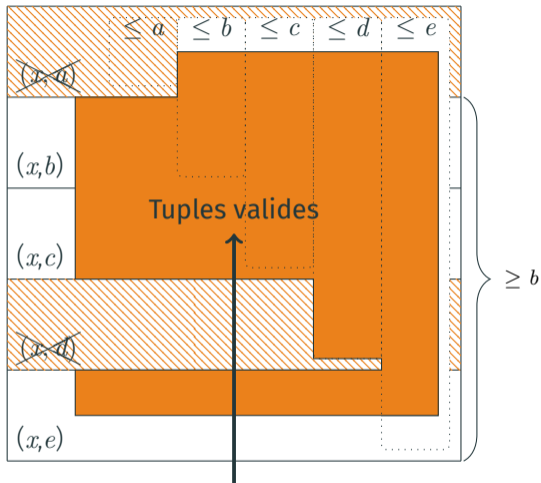


$$\text{currTable} \cap \left(\text{support}^*[x, a] \cup \text{support}^*[x, d] \right) C$$

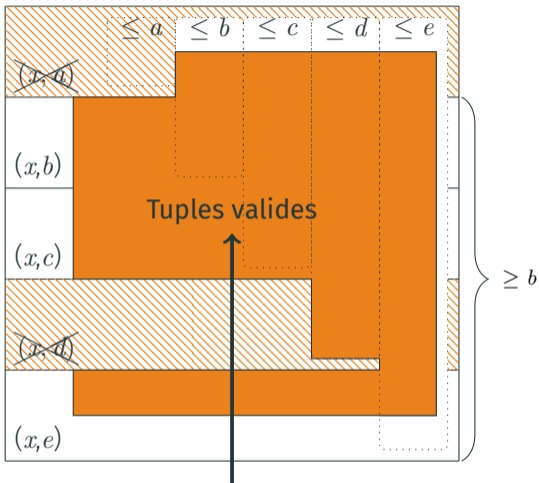


$$\text{currTable} \cap \left(\text{support}^*[x, a] \cup \text{support}^*[x, d] \right) \cap \text{supportMin}[x, b]$$





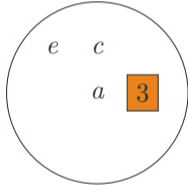
$$\text{currTable} \cap \left(\frac{\text{support}^*[x, a]}{\cup} \right) C \cap \text{supportMin}[x, b]$$

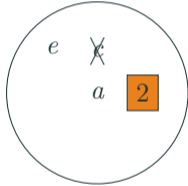


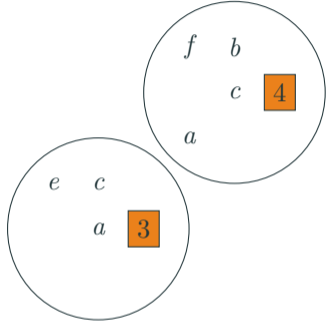
$$\text{currTable} \cap \left(\bigcup_{a \in \text{support}^*[x, a]} \text{support}^*[x, a] \right) \cap \text{supportMin}[x, b]$$

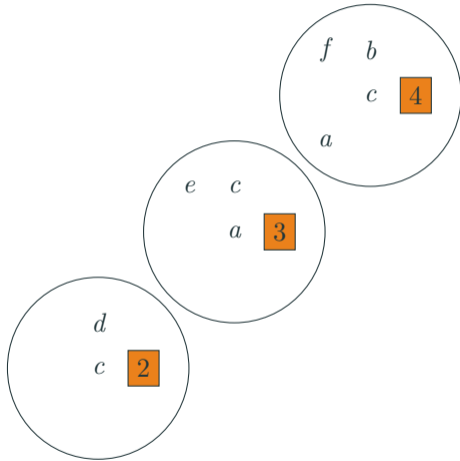
Algorithm: ClassicalUpdate(x)

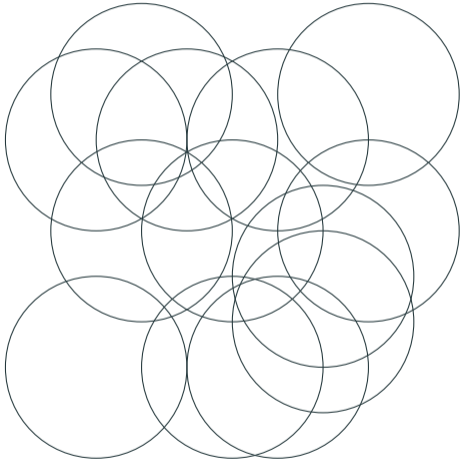
- 1 $\text{mask} \leftarrow 0$;
- 2 **foreach** value $a \in \Delta_x$ **do**
- 3 **if** $a \in [\text{dom}(x).\text{min}; \text{dom}(x).\text{max}]$
- 4 **then**
- 5 $\text{mask} \leftarrow \text{mask} \mid \text{support}^*[x, a]$;
- 6 $\text{mask} \leftarrow \sim \text{mask}$;
- 7 $\text{mask} \leftarrow \text{mask} \& \text{supportMin}[x, \text{dom}(x).\text{min}]$;
- 8 $\text{currTable} \leftarrow \text{currTable} \& \text{mask}$;

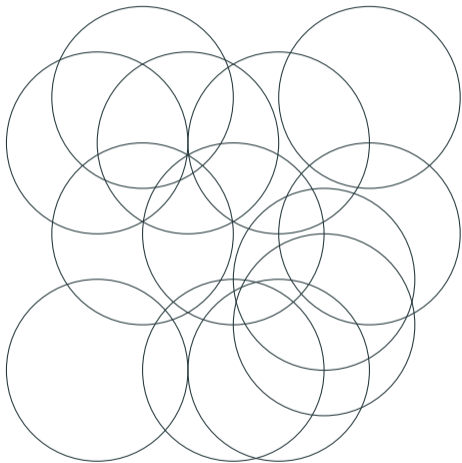












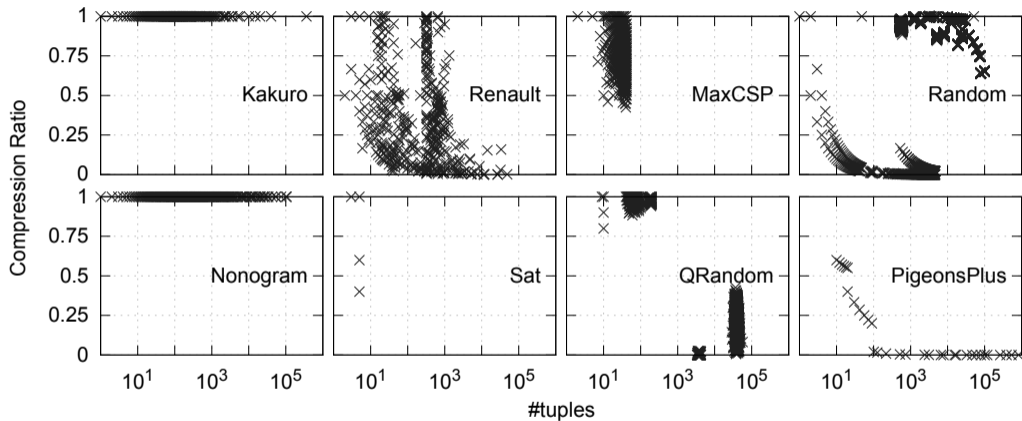
Algorithm: ResetUpdate(x)

- 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` ;
-

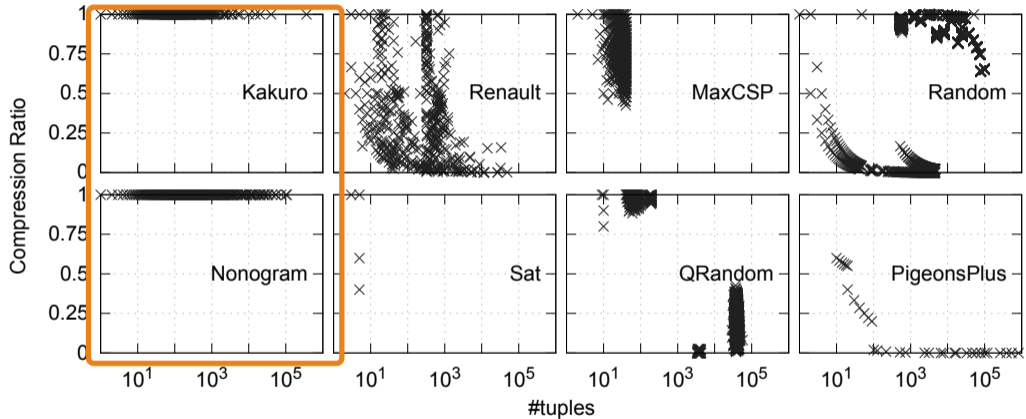
Algorithm: Update(x)

```
1 foreach variable  $x \in \text{scp}_{no \in S}$  do
2   if  $|\Delta_x| < |dom(x)|$  then
3     | ClassicalUpdate(x);
4   else
5     | ResetUpdate(x);
6 foreach variable  $x \in \text{scp}_{with \in S}$  do
7   | ResetUpdate(x);
```

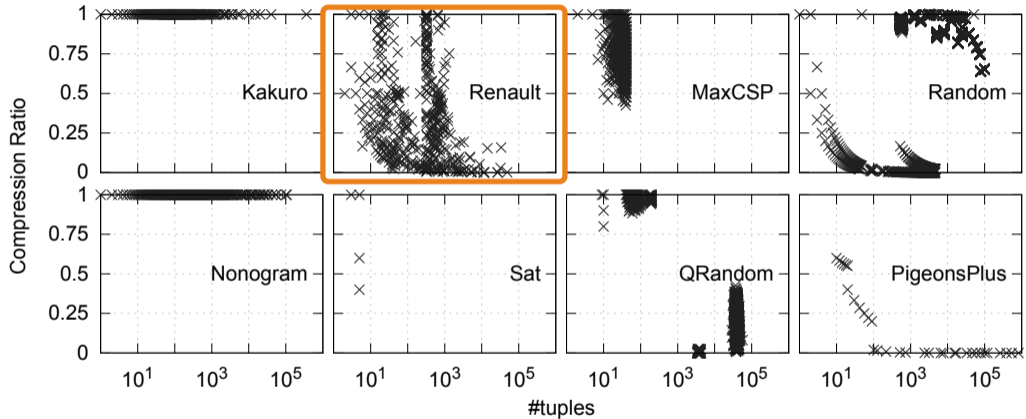
RESULTATS



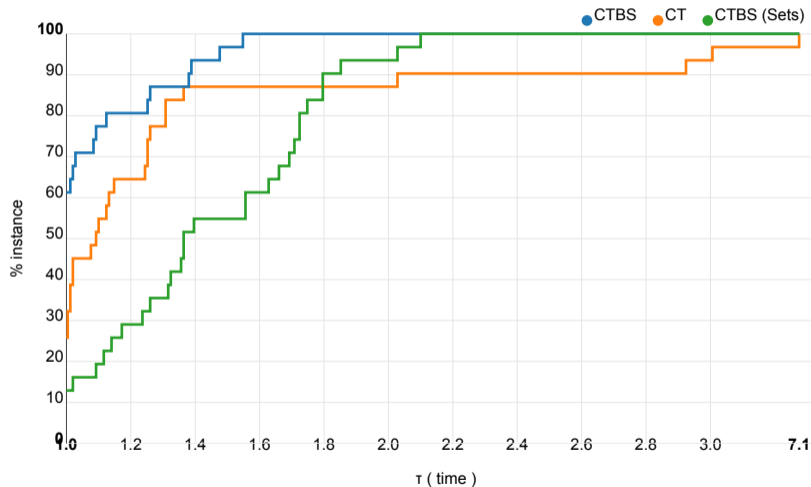
Algorithme glouton de compression générant des \leq et \geq



Algorithme glouton de compression générant des \leq et \geq

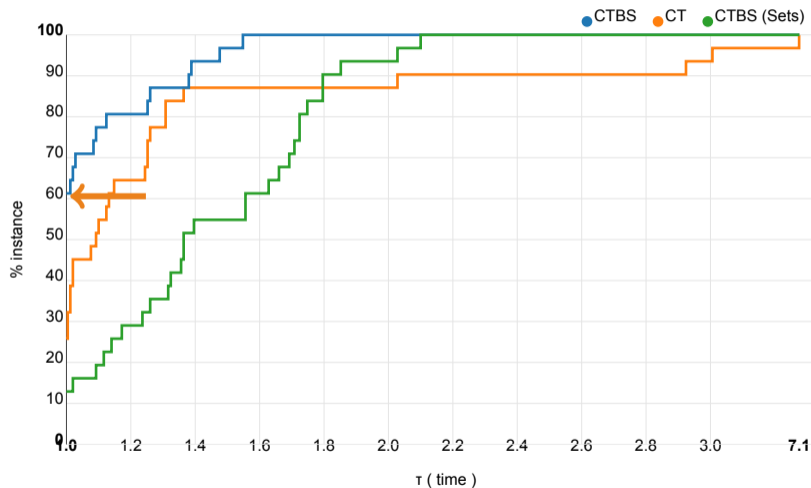


Algorithme glouton de compression générant des \leq et \geq



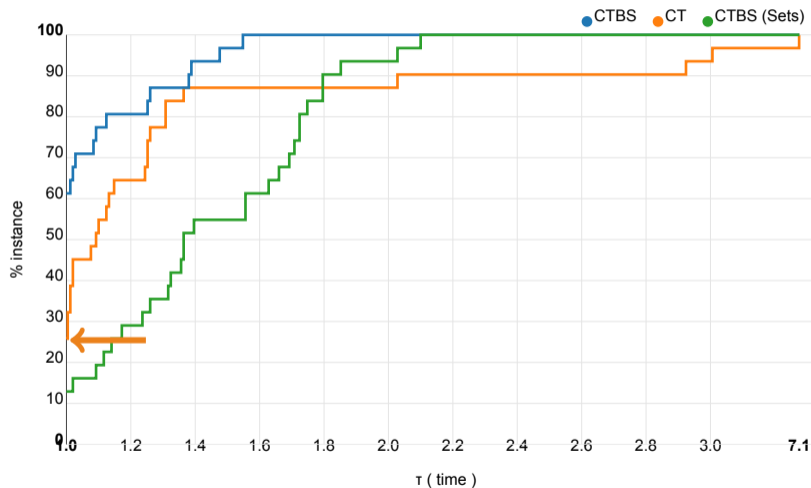
CTBS: compression
 \leq & \geq + post-
 traitement
 \neq & *

CTBS (Sets): même
 compression
 mais vu comme $\in \mathcal{S}$



CTBS: compression
 \leq & \geq + post-
 traitement
 \neq & *

CTBS (Sets): même
 compression
 mais vu comme $\in \mathcal{S}$



CTBS: compression
 \leq & \geq + post-
 traitement
 \neq & $*$

CTBS (Sets): même
 compression
 mais vu comme $\in \mathcal{S}$

$ dom(x) $	ensembles		ensembles structurés
1	$\{a\}$	1	$*$ 1
2	$\{a\}, \{b\}, \{a, b\}$	3	$a, b, *$ 3
3	$\{a\}, \{b\}, \{c\}, \{a, b\},$ $\{a, c\}, \{b, c\}, \{a, b, c\}$	7	$a, b, c, \neq a,$ $\neq b, \neq c, *$ 7
4	$\{a\}, \{b\}, \dots, \{a, b\}, \{a, c\},$ $\{a, d\}, \{b, c\}, \{b, d\}, \{c, d\},$ $\{a, b, c\}, \dots, \{a, b, c, d\}$	15	$a, b, c, d,$ $\leq b, \geq c, \neq a,$ $\neq b, \neq c, \neq d, *$ 11
5	$\{a\}, \{b\}, \dots, \{a, b\}, \{a, c\}, \{a, d\}, \{a, e\},$ $\{b, c\}, \{b, d\}, \{b, e\}, \{c, d\}, \{c, e\},$ $\{a, b, c\}, \{a, b, d\}, \{a, b, e\}, \{a, c, d\},$ $\{a, c, e\}, \dots, \{a, b, c, d\}, \dots, \{a, b, c, d, e\}$	31	a, b, c, d, e $\leq b, \leq c, \geq c,$ $\geq d, \neq a, \neq b,$ $\neq c, \neq d, \neq e, *$ 15

CONCLUSION

Tuples

CT
[CP2016]

	x	y	z
τ_1	a	a	b
τ_2	b	c	a
τ_3	b	a	a
τ_4	c	b	c

Short tuples

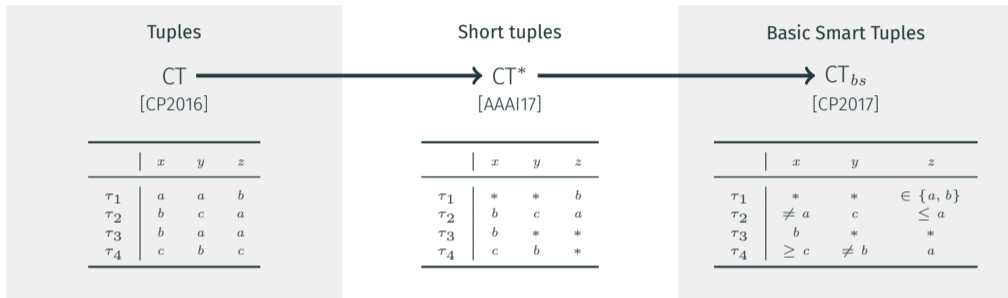
CT*
[AAAI17]

	x	y	z
τ_1	*	*	b
τ_2	b	c	a
τ_3	b	*	*
τ_4	c	b	*

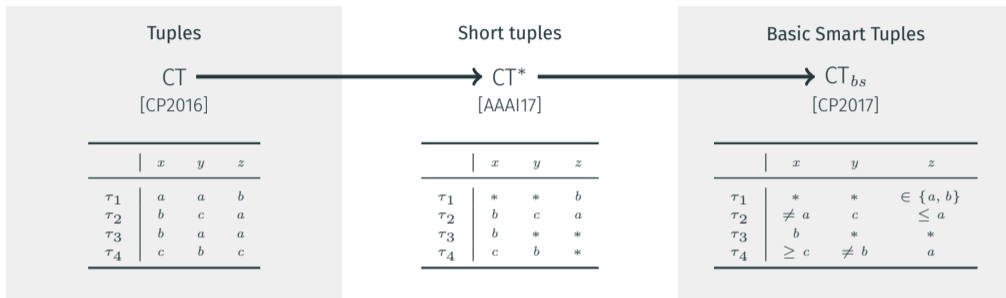
Basic Smart Tuples

CT_{bs}
[CP2017]

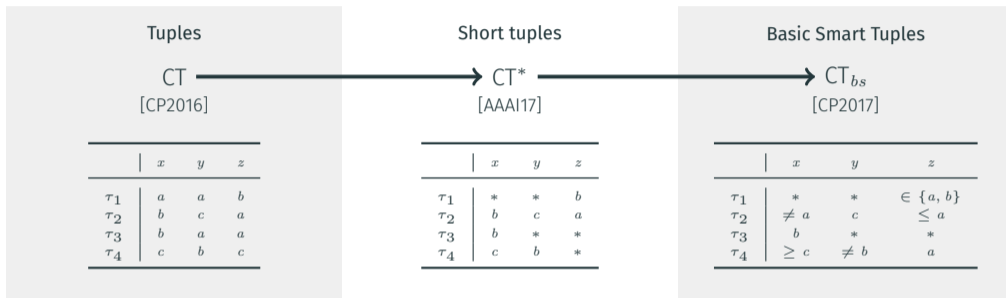
	x	y	z
τ_1	*	*	$\in \{a, b\}$
τ_2	$\neq a$	c	$\leq a$
τ_3	b	*	*
τ_4	$\geq c$	$\neq b$	a



- Augmente l'expressivité

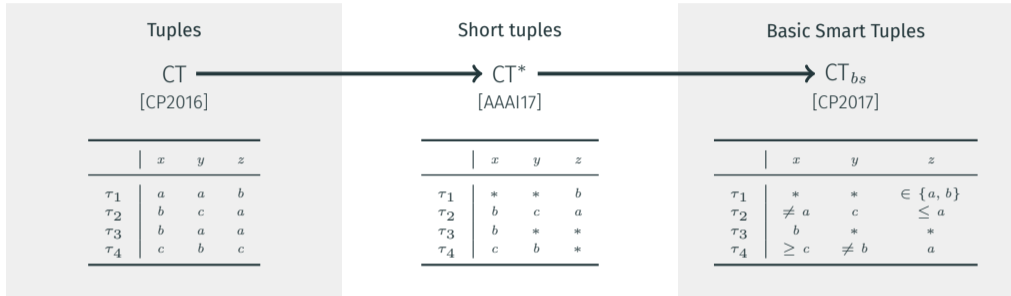


- Augmente l'expressivité
- Diminue le volume de stockage



- Augmente l'expressivité
- Diminue le volume de stockage

- Augmente la vitesse



- Augmente l'expressivité
- Diminue le volume de stockage

- Augmente la vitesse
- Augmente l'efficacité

Merci pour votre attention!

Des questions?