

# Introduction à la Programmation par Contraintes

Cyril Terrioux  
`cyril.terrioux@univ-amu.fr`



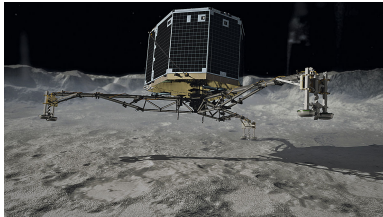
# Plan

- 1 Contexte
- 2 Formalisme
- 3 Résolution
- 4 Quelques exemples de modélisation
- 5 Les limites

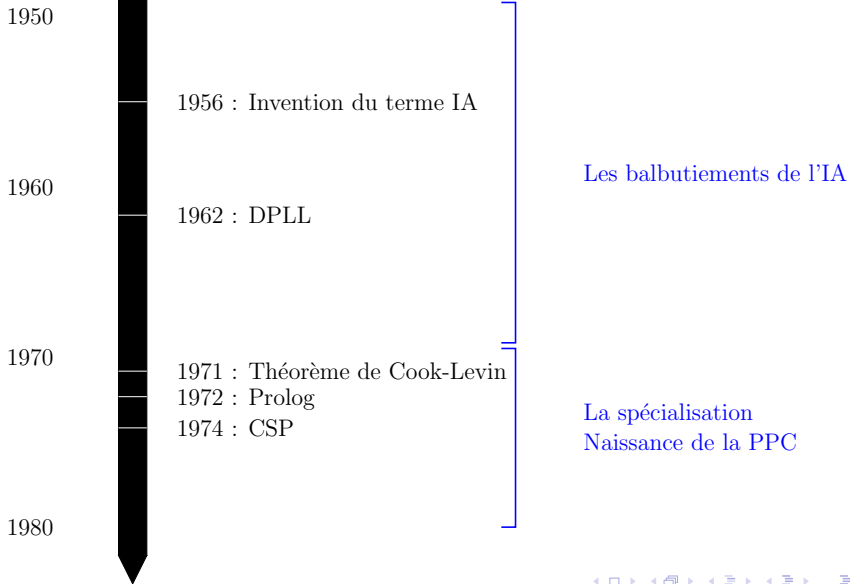
# Plan

- 1 Contexte
- 2 Formalisme
- 3 Résolution
- 4 Quelques exemples de modélisation
- 5 Les limites

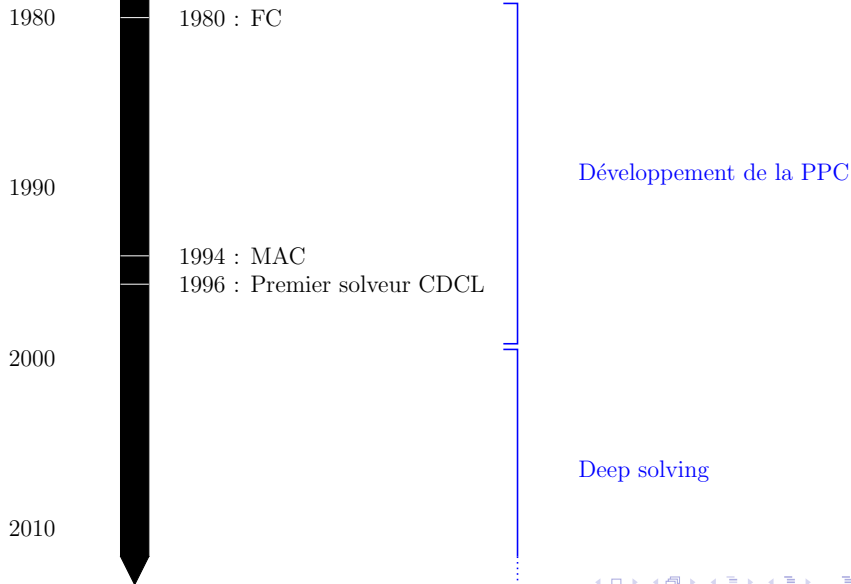
## Omniprésence de la notion de contrainte

[illegible]

# Historique



# Historique



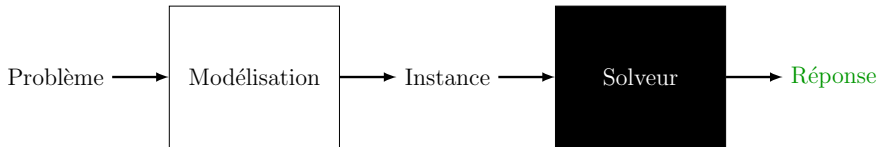
# Principe

*« Constraint Programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming : the user states the problem, the computer solves it. »*  
E. Freuder, 1997

# Principe

*« Constraint Programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming : the user states the problem, the computer solves it. »*

E. Freuder, 1997

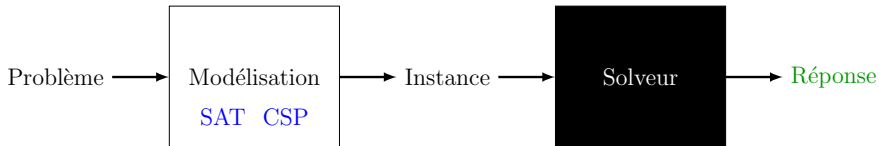




# Principe

*« Constraint Programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming : the user states the problem, the computer solves it. »*

E. Freuder, 1997



# Programmation par contraintes

Interdisciplinaire par nature :

- Intelligence artificielle
- Recherche opérationnelle
- Mathématiques discrètes
- Algorithmique
- Bases de données
- Théorie de la complexité
- Théorie des graphes
- ...

# Applications

- Enchères combinatoires
- Bio-informatique
- Logistique
- Jeux
- Routage
- Transport
- Cryptographie
- Planification
- Télécommunication : Problème d'allocation de fréquences
- Théorie des graphes
- Agriculture
- Chimie
- ...

# Quelques entreprises

- COSLING,
- Cosytec,
- Google,
- Huawei Technologies Ltd,
- IBM (ILOG),
- Innovation 24 (Bouygues E-lab),
- Microsoft Research,
- Orange Labs (France Télécom R&D),
- PROLOGIA,
- ...

# Plan

- 1 Contexte
- 2 Formalisme**
- 3 Résolution
- 4 Quelques exemples de modélisation
- 5 Les limites

# Problème de satisfaction de contraintes (CSP)

Instance  $\mathcal{P} = (X, D, C)$  avec :

- $X = \{x_1, \dots, x_n\}$  un ensemble de  $n$  variables,
- $D = \{d_{x_1}, \dots, d_{x_n}\}$  un ensemble de domaines finis de taille au plus  $d$ ,
- $C$  un ensemble de  $m$  contraintes  $c = (S(c), R(c))$  :
  - $S(c) \subseteq X$  : portée
  - $R(c) \subseteq \prod_{x \in S(c)} d_x$  : relation

# Contraintes

**Arité** =  $|S(c)|$  = nombre de variables soumises à la contrainte

# Contraintes

**Arité** =  $|S(c)|$  = nombre de variables soumises à la contrainte

**Contrainte unaire** = une contrainte portant sur une seule variable

**Contrainte binaire** = une contrainte portant sur deux variables

**Contrainte n-aire** = une contrainte portant sur plus de deux variables



# Contraintes

**Arité** =  $|S(c)|$  = nombre de variables soumises à la contrainte

**Contrainte unaire** = une contrainte portant sur une seule variable

**Contrainte binaire** = une contrainte portant sur deux variables

**Contrainte n-aire** = une contrainte portant sur plus de deux variables

**CSP binaire** = CSP dont toutes les contraintes sont unaires ou binaires

**CSP n-aire** = CSP dont au moins une contrainte est n-aire

# Relations

Ensemble des tuples autorisés par la contrainte  $c_i$

# Relations

Ensemble des tuples autorisés par la contrainte  $c_i$

Relation  $R(c) =$  sous-ensemble du produit cartésien  $\prod_{x \in S(c)} d_x$

# Relations

Ensemble des tuples autorisés par la contrainte  $c_i$

Relation  $R(c) =$  sous-ensemble du produit cartésien  $\prod_{x \in S(c)} d_x$

Plusieurs façons de les définir :

- énumérations des tuples compatibles ou interdits,
- équations, inéquations,
- prédicats,
- ...

## Exemple 1 : Sudoku

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

# Exemple 1 : Sudoku

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

# Exemple 1 : Sudoku

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

## Exemple 1 : Sudoku

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					



# Exemple 1 : Sudoku

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

## Exemple 1 : Sudoku

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$
$x_{10}$	$x_{11}$	$x_{12}$	$x_{13}$	$x_{14}$	$x_{15}$	$x_{16}$	$x_{17}$	$x_{18}$
$x_{19}$	$x_{20}$	$x_{21}$	$x_{22}$	$x_{23}$	$x_{24}$	$x_{25}$	$x_{26}$	$x_{27}$
$x_{28}$	$x_{29}$	$x_{30}$	$x_{31}$	$x_{32}$	$x_{33}$	$x_{34}$	$x_{35}$	$x_{36}$
$x_{37}$	$x_{38}$	$x_{39}$	$x_{40}$	$x_{41}$	$x_{42}$	$x_{43}$	$x_{44}$	$x_{45}$
$x_{46}$	$x_{47}$	$x_{48}$	$x_{49}$	$x_{50}$	$x_{51}$	$x_{52}$	$x_{53}$	$x_{54}$
$x_{55}$	$x_{56}$	$x_{57}$	$x_{58}$	$x_{59}$	$x_{60}$	$x_{61}$	$x_{62}$	$x_{63}$
$x_{64}$	$x_{65}$	$x_{66}$	$x_{67}$	$x_{68}$	$x_{69}$	$x_{70}$	$x_{71}$	$x_{72}$
$x_{73}$	$x_{74}$	$x_{75}$	$x_{76}$	$x_{77}$	$x_{78}$	$x_{79}$	$x_{80}$	$x_{81}$

- $X = \{x_1, x_2, \dots, x_{81}\}$

## Exemple 1 : Sudoku

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

2	4	1	1 2 3	1 2 3	1 2 3	1 2 3	1 2 3	1 2 3
4 5 6	4 5 6	4 5 6	4	4 5 6	4 5 6	1	4 5 6	4 5 6
7 8 9	7 8 9	7 8 9		7 8 9	7 8 9		7 8 9	7 8 9
1 2 3	1 2 3	1 2 3		1 2 3	1 2 3		1 2 3	1 2 3
4 5 6	4 5 6	4 5 6	9	4 5 6	4 5 6		4 5 6	4 5 6
7 8 9	7 8 9	7 8 9		7 8 9	7 8 9		7 8 9	7 8 9
1 2 3	1 2 3	1 2 3	2	1 2 3	1 2 3		1 2 3	1 2 3
4 5 6	4 5 6	4 5 6		4 5 6	4 5 6	6	4 5 6	4 5 6
7 8 9	7 8 9	7 8 9		7 8 9	7 8 9	7	7 8 9	7 8 9
1 2 3	1 2 3	1 2 3		1 2 3	1 2 3		1 2 3	1 2 3
4 5 6	4 5 6	4 5 6		4 5 6	4 5 6		4 5 6	4 5 6
7 8 9	7 8 9	7 8 9		7 8 9	7 8 9		7 8 9	7 8 9
1 2 3	1 2 3	1 2 3	7	1 2 3	1 2 3		1 2 3	1 2 3
4 5 6	4 5 6	4 5 6		4 5 6	4 5 6		4 5 6	4 5 6
7 8 9	7 8 9	7 8 9		7 8 9	7 8 9		7 8 9	7 8 9
1 2 3	1 2 3	1 2 3		1 2 3	1 2 3		1 2 3	1 2 3
4 5 6	4 5 6	4 5 6	8	4 5 6	4 5 6		4 5 6	4 5 6
7 8 9	7 8 9	7 8 9		7 8 9	7 8 9		7 8 9	7 8 9
1 2 3	1 2 3	1 2 3		1 2 3	1 2 3		1 2 3	1 2 3
4 5 6	4 5 6	4 5 6		4 5 6	4 5 6	3	4 5 6	4 5 6
7 8 9	7 8 9	7 8 9		7 8 9	7 8 9		7 8 9	7 8 9
1 2 3	1 2 3	1 2 3		1 2 3	1 2 3		1 2 3	1 2 3
4 5 6	4 5 6	4 5 6	8	4 5 6	4 5 6		4 5 6	4 5 6
7 8 9	7 8 9	7 8 9		7 8 9	7 8 9		7 8 9	7 8 9
1 2 3	1 2 3	1 2 3		1 2 3	1 2 3		1 2 3	1 2 3
4 5 6	4 5 6	4 5 6		4 5 6	4 5 6		4 5 6	4 5 6
7 8 9	7 8 9	7 8 9		7 8 9	7 8 9		7 8 9	7 8 9

- $X = \{x_1, x_2, \dots, x_{81}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{81}}\}$  avec
  - $d_{x_1} = \{2\},$
  - $d_{x_2} = \{4\},$
  - $d_{x_3} = \{1\},$
  - $d_{x_4} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},$
  - ...

## Exemple 1 : Sudoku

2	4	1						
			4			1		
	5		9					6
		2				6	7	
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

2	4	1						
			4			1		
	5		9					6
		2				6	7	
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

- $X = \{x_1, x_2, \dots, x_{81}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{81}}\}$  avec
  - $d_{x_1} = \{2\},$
  - $d_{x_2} = \{4\},$
  - $d_{x_3} = \{1\},$
  - $d_{x_4} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},$
  - ...
- $C = \{c_{1,2}, c_{1,3}, \dots\}$  avec
  - $S(c_{i,j}) = \{x_i, x_j\}$
  - $R(c_{i,j}) = \{(v_i, v_j) \in d_{x_i} \times d_{x_j} \mid v_i \neq v_j\}$

## Exemple 1 : Sudoku

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

- $X = \{x_1, x_2, \dots, x_{81}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{81}}\}$  avec
  - $d_{x_1} = \{2\},$
  - $d_{x_2} = \{4\},$
  - $d_{x_3} = \{1\},$
  - $d_{x_4} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},$
  - ...
- $C = \{c_{1,2}, c_{1,3}, \dots\}$  avec
  - $S(c_{i,j}) = \{x_i, x_j\}$
  - $R(c_{i,j}) = \{(v_i, v_j) \in d_{x_i} \times d_{x_j} \mid v_i \neq v_j\}$

## Exemple 1 : Sudoku

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

- $X = \{x_1, x_2, \dots, x_{81}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{81}}\}$  avec
  - $d_{x_1} = \{2\},$
  - $d_{x_2} = \{4\},$
  - $d_{x_3} = \{1\},$
  - $d_{x_4} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},$
  - ...
- $C = \{c_{1,2}, c_{1,3}, \dots\}$  avec
  - $S(c_{i,j}) = \{x_i, x_j\}$
  - $R(c_{i,j}) = \{(v_i, v_j) \in d_{x_i} \times d_{x_j} \mid v_i \neq v_j\}$

# Exemple 2 : Système d'inéquations

$$\left\{ \begin{array}{l} x_1 < x_2 \\ x_1 < x_3 \\ x_1 < x_4 \\ x_1 < x_5 \\ x_3^2 < x_6 \\ x_4 > x_6 \\ x_5 \in \{1, 2\} \\ x_i \in \{1, 2, 3\}, i \neq 5 \end{array} \right.$$

# Exemple 2 : Système d'inéquations

- $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$

$$\left\{ \begin{array}{l} x_1 < x_2 \\ x_1 < x_3 \\ x_1 < x_4 \\ x_1 < x_5 \\ x_3^2 < x_6 \\ x_4 > x_6 \\ x_5 \in \{1, 2\} \\ x_i \in \{1, 2, 3\}, i \neq 5 \end{array} \right.$$



## Exemple 2 : Système d'inéquations

$$\left\{ \begin{array}{l} x_1 < x_2 \\ x_1 < x_3 \\ x_1 < x_4 \\ x_1 < x_5 \\ x_3^2 < x_6 \\ x_4 > x_6 \\ x_5 \in \{1, 2\} \\ x_i \in \{1, 2, 3\}, i \neq 5 \end{array} \right.$$

- $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$
- $D = \{d_{x_1}, d_{x_2}, d_{x_3}, d_{x_4}, d_{x_5}, d_{x_6}\}$  avec
  - $d_{x_i} = \{1, 2, 3\} \ i \neq 5,$
  - $d_{x_5} = \{1, 2\}$

## Exemple 2 : Système d'inéquations

$$\left\{ \begin{array}{l} x_1 < x_2 \\ x_1 < x_3 \\ x_1 < x_4 \\ x_1 < x_5 \\ x_3^2 < x_6 \\ x_4 > x_6 \\ x_5 \in \{1, 2\} \\ x_i \in \{1, 2, 3\}, i \neq 5 \end{array} \right.$$

- $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$
- $D = \{d_{x_1}, d_{x_2}, d_{x_3}, d_{x_4}, d_{x_5}, d_{x_6}\}$  avec
  - $d_{x_i} = \{1, 2, 3\} \ i \neq 5,$
  - $d_{x_5} = \{1, 2\}$
- $C = \{c_{12}, c_{13}, c_{14}, c_{15}, c_{36}, c_{46}\}$   
 avec  $S(c_{ij}) = \{x_i, x_j\}$   
 $R(c_{36}) = \{(v_3, v_6) \in d_{x_3} \times d_{x_6} \mid v_3^2 < v_6\},$   
 $R(c_{46}) = \{(v_4, v_6) \in d_{x_4} \times d_{x_6} \mid v_4 > v_6\},$

## Exemple 2 : Système d'inéquations

$$\left\{ \begin{array}{l} x_1 < x_2 \\ x_1 < x_3 \\ x_1 < x_4 \\ x_1 < x_5 \\ x_3^2 < x_6 \\ x_4 > x_6 \\ x_5 \in \{1, 2\} \\ x_i \in \{1, 2, 3\}, i \neq 5 \end{array} \right.$$

- $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$
- $D = \{d_{x_1}, d_{x_2}, d_{x_3}, d_{x_4}, d_{x_5}, d_{x_6}\}$  avec
  - $d_{x_i} = \{1, 2, 3\} \ i \neq 5,$
  - $d_{x_5} = \{1, 2\}$
- $C = \{c_{12}, c_{13}, c_{14}, c_{15}, c_{36}, c_{46}\}$   
 avec  $S(c_{ij}) = \{x_i, x_j\}$   
 $R(c_{36}) = \{(v_3, v_6) \in d_{x_3} \times d_{x_6} \mid v_3^2 < v_6\},$   
 $R(c_{46}) = \{(v_4, v_6) \in d_{x_4} \times d_{x_6} \mid v_4 > v_6\},$   
 $R(c_{ij}) = \{(v_i, v_j) \in d_{x_i} \times d_{x_j} \mid v_i < v_j\}$

## Exemple 2 : Système d'inéquations

$$\left\{ \begin{array}{l} x_1 < x_2 \\ x_1 < x_3 \\ x_1 < x_4 \\ x_1 < x_5 \\ x_3^2 < x_6 \\ x_4 > x_6 \\ x_5 \in \{1, 2\} \\ x_i \in \{1, 2, 3\}, i \neq 5 \end{array} \right.$$

## Exemple 2 : Système d'inéquations

- Contrainte Table positive

$x_1$	$x_2$
1	2
1	3
2	3

$$\left\{ \begin{array}{l} x_1 < x_2 \\ x_1 < x_3 \\ x_1 < x_4 \\ x_1 < x_5 \\ x_3^2 < x_6 \\ x_4 > x_6 \\ x_5 \in \{1, 2\} \\ x_i \in \{1, 2, 3\}, i \neq 5 \end{array} \right.$$

## Exemple 2 : Système d'inéquations

$$\left\{ \begin{array}{l} x_1 < x_2 \\ x_1 < x_3 \\ x_1 < x_4 \\ x_1 < x_5 \\ x_3^2 < x_6 \\ x_4 > x_6 \\ x_5 \in \{1, 2\} \\ x_i \in \{1, 2, 3\}, i \neq 5 \end{array} \right.$$

- Contrainte Table positive

$x_1$	$x_2$
1	2
1	3
2	3

- Contrainte Table négative

$x_1$	$x_5$
1	1
2	1
2	2

## Exemple 2 : Système d'inéquations

$$\left\{ \begin{array}{l} x_1 < x_2 \\ x_1 < x_3 \\ x_1 < x_4 \\ x_1 < x_5 \\ x_3^2 < x_6 \\ x_4 > x_6 \\ x_5 \in \{1, 2\} \\ x_i \in \{1, 2, 3\}, i \neq 5 \end{array} \right.$$

- Contrainte Table positive

$x_1$	$x_2$
1	2
1	3
2	3

- Contrainte Table négative

$x_1$	$x_5$
1	1
2	1
2	2

- Contrainte Table compacte négative

$x_1$	$x_5$
1	1
2	*

# Affectation / Instanciation

**Instanciation** des variables de  $Y = \{y_1, \dots, y_k\} = k\text{-uplet } (v_1, \dots, v_k)$  de  $d_{y_1} \times \dots \times d_{y_k}$ .



# Affectation / Instanciation

**Instanciation** des variables de  $Y = \{y_1, \dots, y_k\} = k\text{-uplet } (v_1, \dots, v_k)$  de  $d_{y_1} \times \dots \times d_{y_k}$ .

**Instanciation complète** = instanciation de toutes les variables

# Affectation / Instanciation

**Instanciation** des variables de  $Y = \{y_1, \dots, y_k\} = k\text{-uplet } (v_1, \dots, v_k)$  de  $d_{y_1} \times \dots \times d_{y_k}$ .

**Instanciation complète** = instanciation de toutes les variables

**Instanciation partielle** = instanciation d'une partie des variables

# Affectation / Instanciation

**Instanciation** des variables de  $Y = \{y_1, \dots, y_k\} = k\text{-uplet } (v_1, \dots, v_k)$  de  $d_{y_1} \times \dots \times d_{y_k}$ .

**Instanciation complète** = instanciation de toutes les variables

**Instanciation partielle** = instanciation d'une partie des variables

Notation :  $\{x_1 = a, x_2 = c\}$

# Affectation / Instanciation

Une affectation  $\mathcal{A}$  de  $Y = \{y_1, \dots, y_k\}$  **satisfait** une contrainte  $c$  si  $\mathcal{A}[S(c)] \in R(c)$ .

$\mathcal{A}$  **viole**  $c$  sinon.

# Affectation / Instanciation

Une affectation  $\mathcal{A}$  de  $Y = \{y_1, \dots, y_k\}$  **satisfait** une contrainte  $c$  si  $\mathcal{A}[S(c)] \in R(c)$ .

$\mathcal{A}$  **viole**  $c$  sinon.

Instanciation **cohérente** = instanciation qui satisfait toutes les contraintes  $c$  telles que  $S(c) \subseteq Y$

# Affectation / Instanciation

Une affectation  $\mathcal{A}$  de  $Y = \{y_1, \dots, y_k\}$  **satisfait** une contrainte  $c$  si  $\mathcal{A}[S(c)] \in R(c)$ .

$\mathcal{A}$  **viole**  $c$  sinon.

Instanciation **cohérente** = instanciation qui satisfait toutes les contraintes  $c$  telles que  $S(c) \subseteq Y$

Instanciation **incohérente** = instanciation qui viole au moins une contrainte  $c$  telle que  $S(c) \subseteq Y$

## Exemple

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

Affectation partielle cohérente

## Exemple

2	4	1		5				
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7	5			3	
		8		6			4	5
4			3					

Affectation partielle incohérente



## Exemple

2	4	1		5				
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7	5			3	
		8		6			4	5
4			3					

Affectation partielle incohérente

# Solution

**Solution** = affectation complète cohérente

# Solution

**Solution** = affectation complète cohérente

Problème **cohérent** = problème possédant au moins une solution

# Solution

**Solution** = affectation complète cohérente

Problème **cohérent** = problème possédant au moins une solution

Problème **incohérent** = problème ne possédant aucune solution

# Exemple

2	4	1	5	3	6	8	7	9
7	9	6	4	8	2	1	5	3
8	5	3	9	1	7	4	2	6
5	8	2	1	4	3	9	6	7
1	7	9	6	2	5	3	8	4
3	6	4	8	7	9	5	1	2
6	1	5	7	9	4	2	3	8
9	3	8	2	6	1	7	4	5
4	2	7	3	5	8	6	9	1

Affectation complète cohérente = solution

# Problèmes posés

- Une instance possède-t-elle une solution ?

# Problèmes posés

- Une instance possède-t-elle une solution ?
- Trouver une solution d'une instance.

# Problèmes posés

- Une instance possède-t-elle une solution ?
- Trouver une solution d'une instance.
- Trouver la meilleure solution d'une instance.



# Problèmes posés

- Une instance possède-t-elle une solution ?
- Trouver une solution d'une instance.
- Trouver la meilleure solution d'une instance.
- Trouver toutes les solutions d'une instance.

# Problèmes posés

- Une instance possède-t-elle une solution ?
- Trouver une solution d'une instance.
- Trouver la meilleure solution d'une instance.
- Trouver toutes les solutions d'une instance.
- Déterminer le nombre de solutions d'une instance.

# Problèmes posés

- Une instance possède-t-elle une solution ?
- Trouver une solution d'une instance.
- Trouver la meilleure solution d'une instance.
- Trouver toutes les solutions d'une instance.
- Déterminer le nombre de solutions d'une instance.
- ...

# Problèmes posés

- Une instance possède-t-elle une solution ? NP-complet
- Trouver une solution d'une instance.
- Trouver la meilleure solution d'une instance.
- Trouver toutes les solutions d'une instance.
- Déterminer le nombre de solutions d'une instance.
- ...

# Problèmes posés

- Une instance possède-t-elle une solution ? NP-complet
- Trouver une solution d'une instance. NP-difficile
- Trouver la meilleure solution d'une instance.
- Trouver toutes les solutions d'une instance.
- Déterminer le nombre de solutions d'une instance.
- ...

# Problèmes posés

- |   |              |
|---|--------------|
| • Une instance possède-t-elle une solution ?        | NP-complet   |
| • Trouver une solution d'une instance.              | NP-difficile |
| • Trouver la meilleure solution d'une instance.     | NP-difficile |
| • Trouver toutes les solutions d'une instance.      |              |
| • Déterminer le nombre de solutions d'une instance. |              |
| • ...   |              |

# Problèmes posés

- Une instance possède-t-elle une solution ? NP-complet
- Trouver une solution d'une instance. NP-difficile
- Trouver la meilleure solution d'une instance. NP-difficile
- Trouver toutes les solutions d'une instance.
- Déterminer le nombre de solutions d'une instance. #P-complet
- ...

# Plan

- 1 Contexte
- 2 Formalisme
- 3 Résolution**
- 4 Quelques exemples de modélisation
- 5 Les limites



# Résolution

Taille de l'espace de recherche :  $O(d^n)$

# Résolution

Taille de l'espace de recherche :  $O(d^n)$

À raison de  $10^9$  affectations par seconde et pour  $d = 2$

$n$	$d^n$	Temps
10	1 024	1 $\mu s$
50	$1,12 \times 10^{15}$	13 jours
100	$1,26 \times 10^{30}$	$4 \times 10^{13}$ ans

# Résolution

Taille de l'espace de recherche :  $O(d^n)$

À raison de  $10^9$  affectations par seconde et pour  $d = 2$

$n$	$d^n$	Temps
10	1 024	1 $\mu s$
50	$1,12 \times 10^{15}$	13 jours
100	$1,26 \times 10^{30}$	$4 \times 10^{13}$ ans

- Méthodes complètes
- Méthodes incomplètes

# Backtrack (BT)

Principe :

- on étend progressivement une affectation cohérente

# Backtrack (BT)

Principe :

- on étend progressivement une affectation cohérente
- en cas d'échec, on change la valeur de la variable courante

# Backtrack (BT)

Principe :

- on étend progressivement une affectation cohérente
- en cas d'échec, on change la valeur de la variable courante
- s'il n'y a plus de valeur, on revient sur la variable précédente

# Backtrack (BT)

Principe :

- on étend progressivement une affectation cohérente
- en cas d'échec, on change la valeur de la variable courante
- s'il n'y a plus de valeur, on revient sur la variable précédente
- on réitère le procédé tant qu'on n'a pas trouvé une solution et qu'on n'a pas essayé toutes les possibilités

# Backtrack (BT)

Principe :

- on étend progressivement une affectation cohérente
- en cas d'échec, on change la valeur de la variable courante
- s'il n'y a plus de valeur, on revient sur la variable précédente
- on réitère le procédé tant qu'on n'a pas trouvé une solution et qu'on n'a pas essayé toutes les possibilités

Complexité :  $O(mrd^n)$



# Backtrack (BT)

$\text{BT}(\mathcal{A}, V)$

**Si**  $V = \emptyset$  **Alors**  $\mathcal{A}$  est une solution

**Sinon**

Choisir  $x \in V$

$d \leftarrow d_x$

**TantQue**  $d \neq \emptyset$

Choisir  $v$  dans  $d$

$d \leftarrow d \setminus \{v\}$

**Si**  $\mathcal{A} \cup \{x = v\}$  est cohérente

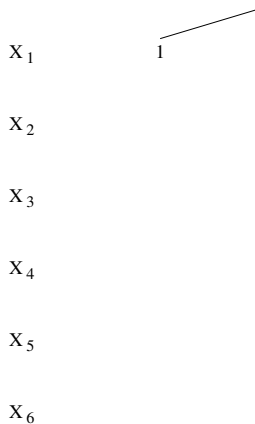
**Alors**  $\text{BT}(\mathcal{A} \cup \{x \leftarrow v\}, V \setminus \{x\})$

**FinSi**

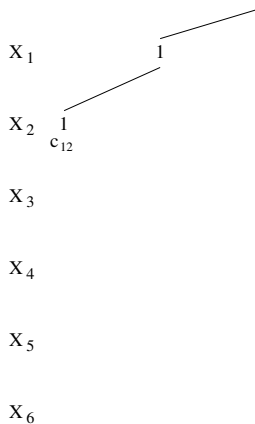
**FinTantQue**

**FinSi**

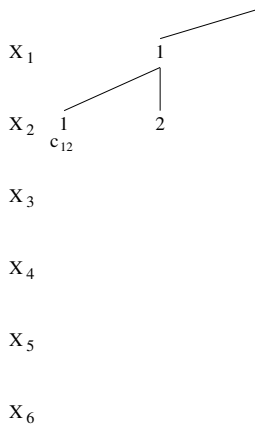
# Exemple



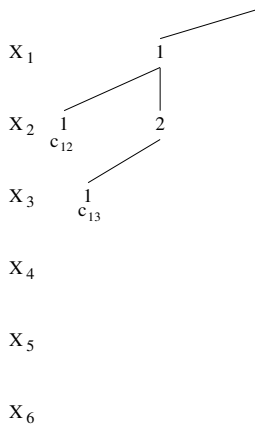
# Exemple



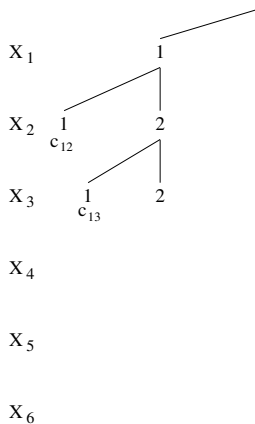
# Exemple



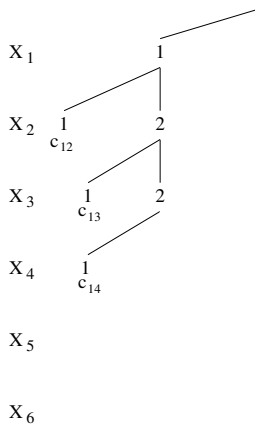
# Exemple



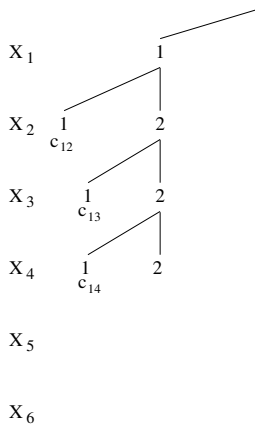
## Exemple



## Exemple

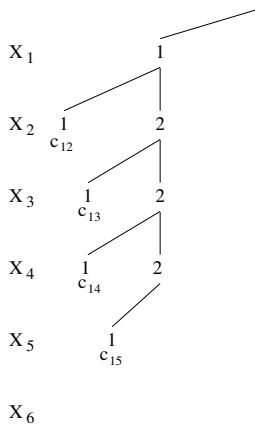


## Exemple

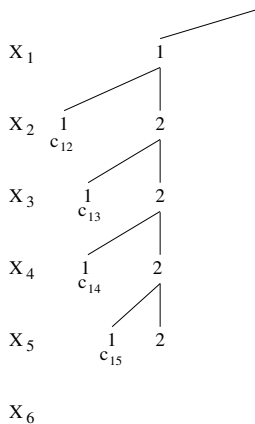




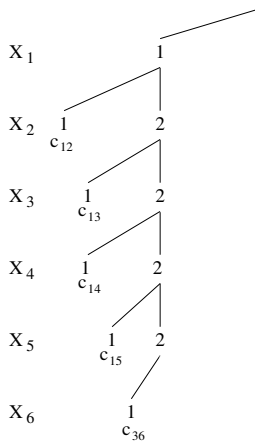
## Exemple



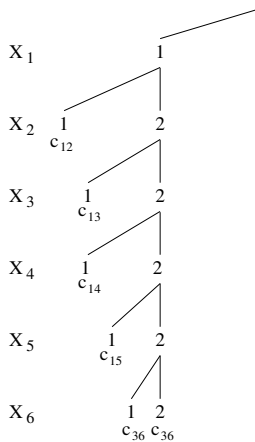
## Exemple



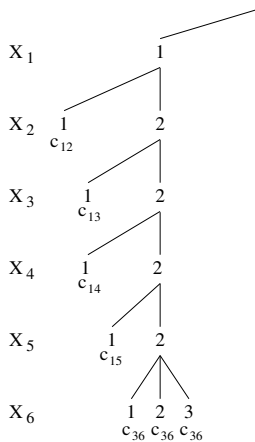
## Exemple



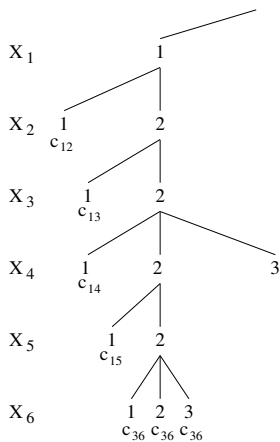
## Exemple



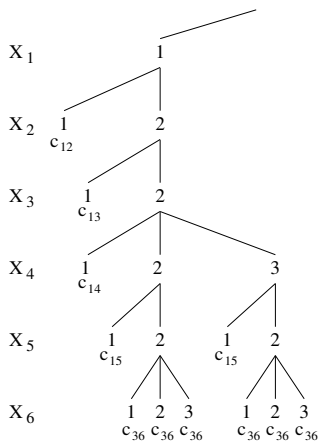
## Exemple



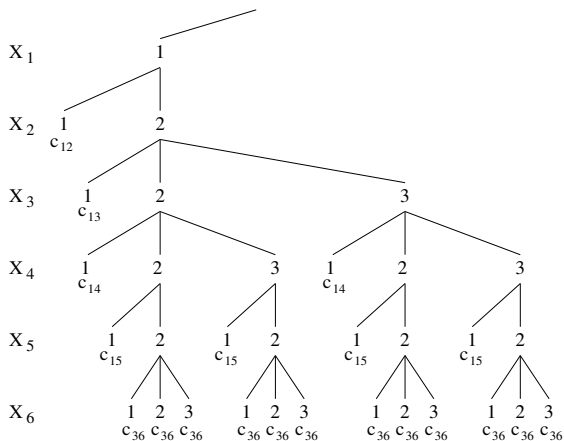
## Exemple



## Exemple

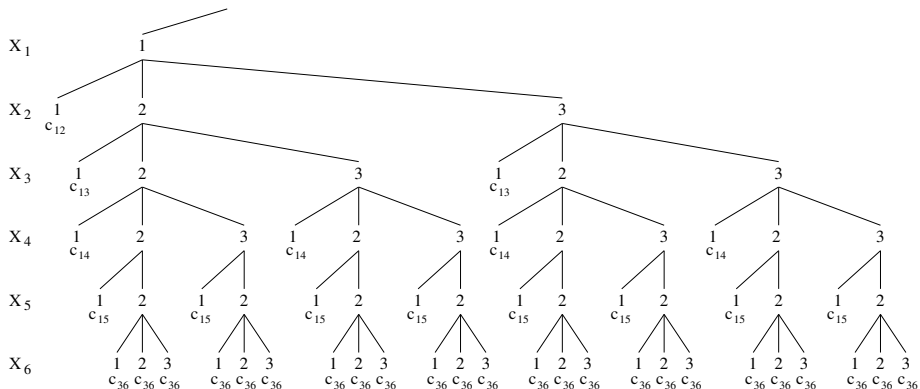


## Exemple





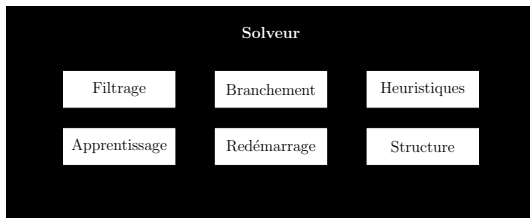
## Exemple



# Solveur

Solveur

# Solveur



# Filtrage

Objectif : éviter de développer certaines branches inutiles

# Filtrage

Objectif : éviter de développer certaines branches inutiles

**branche inutile** = branche ne conduisant pas à une solution

# Filtrage

Objectif : éviter de développer certaines branches inutiles

**branche inutile** = branche ne conduisant pas à une solution

On simplifie le problème en supprimant des valeurs inutiles

# Filtrage

Objectif : éviter de développer certaines branches inutiles

**branche inutile** = branche ne conduisant pas à une solution

On simplifie le problème en supprimant des valeurs inutiles

**valeur inutile** = valeur ne participant pas à une solution

# Filtrage

Objectif : éviter de développer certaines branches inutiles

**branche inutile** = branche ne conduisant pas à une solution

On simplifie le problème en supprimant des valeurs inutiles

**valeur inutile** = valeur ne participant pas à une solution

Avantages :

- Anticipation des échecs immédiats de Backtrack
- Factorisation des tests de contraintes



# Filtrage

Objectif : éviter de développer certaines branches inutiles

**branche inutile** = branche ne conduisant pas à une solution

On simplifie le problème en supprimant des valeurs inutiles

**valeur inutile** = valeur ne participant pas à une solution

Avantages :

- Anticipation des échecs immédiats de Backtrack
- Factorisation des tests de contraintes

L'opération de simplification est appelée **filtrage**.

## Filtrage

2	4	1	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	4	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9	5	1 2 3 4 5 6 7 8 9	9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	6
1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	2	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	6	7
1 2 3 4 5 6 7 8 9	7	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	5	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
3	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	8	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	2
1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	7	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	3	1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	8	1 2 3 4 5 6 7 8 9	6	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	4	5
4	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	3	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9

## Filtrage

2	4	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	9	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6	7
<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	<div>1 2 3 4 5 6 7 8 9</div>	6	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	5
4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>

# Filtrage

2	4	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	9	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6	7
<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	<div>1 2 3 4 5 6 7 8 9</div>	6	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	5
4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>

# Filtrage

2	4	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	9	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6	7
<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	<div>1 2 3 4 5 6 7 8 9</div>	6	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	5
4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>

# Propagation de contraintes

2	4	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	9	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6	7
<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	<div>1 2 3 4 5 6 7 8 9</div>	6	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	5
4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>

# Propagation de contraintes

2	4	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	9	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6	7
<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	<div>1 2 3 4 5 6 7 8 9</div>	6	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	5
4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>

# Propagation de contraintes

2	4	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	9	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6	7
<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	2	6	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	5
4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>



# Propagation de contraintes

2	4	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	9	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6	7
<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	2	6	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	5
4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>

# Propagation de contraintes

2	4	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	9	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2	1	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	6	7
<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	6	<div>1 2 3 4 5 6 7 8 9</div>	5	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>
3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	2
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	7	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>
<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	8	2	6	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	4	5
4	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	3	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>	<div>1 2 3 4 5 6 7 8 9</div>

# Cohérence d'arc (AC)

$t \in R(c)$  est *valide* si  $\forall y \in S(c), t[\{y\}] \in d_y$

Soit une instance CSP  $P = (X, D, C)$

- Une valeur  $v$  de  $d_x$  est *arc-cohérente* si pour chaque contrainte  $c$  t.q.  $x \in S(c)$ ,  $\exists t \in R(c)$  t.q. :
  - $t[\{x\}] = v$  et
  - $t$  valide

# Cohérence d'arc (AC)

$t \in R(c)$  est *valide* si  $\forall y \in S(c), t[\{y\}] \in d_y$

Soit une instance CSP  $P = (X, D, C)$

- Une valeur  $v$  de  $d_x$  est *arc-cohérente* si pour chaque contrainte  $c$  t.q.  
 $x \in S(c), \exists t \in R(c)$  t.q. :
  - $t[\{x\}] = v$  et (  $t$  support de  $v$  vis-à-vis de  $c$  )
  - $t$  valide

# Cohérence d'arc (AC)

$t \in R(c)$  est *valide* si  $\forall y \in S(c), t[\{y\}] \in d_y$

Soit une instance CSP  $P = (X, D, C)$

- Une valeur  $v$  de  $d_x$  est *arc-cohérente* si pour chaque contrainte  $c$  t.q.  
 $x \in S(c), \exists t \in R(c)$  t.q. :
  - $t[\{x\}] = v$  et ( $t$  support de  $v$  vis-à-vis de  $c$ )
  - $t$  valide
- Une variable  $x$  est *arc-cohérente* si chaque valeur  $v$  de  $d_x$  est arc-cohérente.

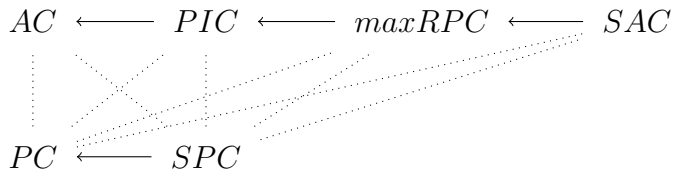
# Cohérence d'arc (AC)

$t \in R(c)$  est *valide* si  $\forall y \in S(c), t[\{y\}] \in d_y$

Soit une instance CSP  $P = (X, D, C)$

- Une valeur  $v$  de  $d_x$  est *arc-cohérente* si pour chaque contrainte  $c$  t.q.  $x \in S(c)$ ,  $\exists t \in R(c)$  t.q. :
  - $t[\{x\}] = v$  et ( $t$  support de  $v$  vis-à-vis de  $c$ )
  - $t$  valide
- Une variable  $x$  est *arc-cohérente* si chaque valeur  $v$  de  $d_x$  est arc-cohérente.
- $P$  est dite *arc-cohérente* si chaque variable de  $P$  est arc-cohérente.

# Quelques filtrages



# Restauration de l'état précédent

Une nécessité en cas d'échec

Trois possibilités :

- Copier les domaines
- Mémoriser les valeurs supprimées
- Mémoriser la taille des domaines des variables impactées



# Restauration de l'état précédent

Une nécessité en cas d'échec

Trois possibilités :

- Copier les domaines
- Mémoriser les valeurs supprimées
- Mémoriser la taille des domaines des variables impactées

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

# Restauration de l'état précédent

Une nécessité en cas d'échec

Trois possibilités :

- Copier les domaines
- Mémoriser les valeurs supprimées
- Mémoriser la taille des domaines des variables impactées

0	1	2	3	7	5	6	4	7
---	---	---	---	---	---	---	---	---

Suppression de 4

# Restauration de l'état précédent

Une nécessité en cas d'échec

Trois possibilités :

- Copier les domaines
- Mémoriser les valeurs supprimées
- Mémoriser la taille des domaines des variables impactées

2	1	0	3	7	5	6	4	1
---	---	---	---	---	---	---	---	---

Affectation de 2

# Restauration de l'état précédent

Une nécessité en cas d'échec

Trois possibilités :

- Copier les domaines
- Mémoriser les valeurs supprimées
- Mémoriser la taille des domaines des variables impactées

2	1	0	3	7	5	6	4	7
---	---	---	---	---	---	---	---	---

Retour en arrière (désaffectation de 2)

# Restauration de l'état précédent

Une nécessité en cas d'échec

Trois possibilités :

- Copier les domaines
- Mémoriser les valeurs supprimées
- Mémoriser la taille des domaines des variables impactées

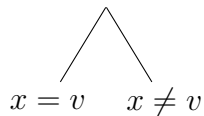
2	1	0	3	7	5	6	4	8
---	---	---	---	---	---	---	---	---

Retour en arrière (restauration de 4)

# Branchement

Deux types de branchement possibles :

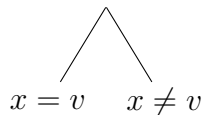
- Branchement binaire :



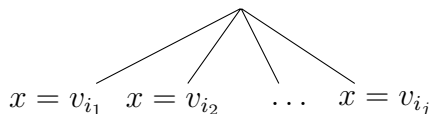
# Branchement

Deux types de branchement possibles :

- Branchement binaire :



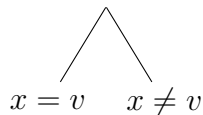
- Branchement non-binaire :



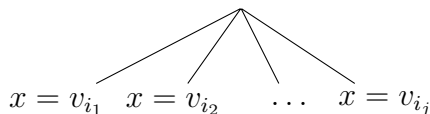
# Branchement

Deux types de branchement possibles :

- Branchement binaire :



- Branchement non-binaire :



Le branchement binaire est théoriquement meilleur.



# Heuristiques

Une traduction de la notion d'intuition

# Heuristiques

Une traduction de la notion d'intuition

Objectif : guider la recherche

# Heuristiques

Une traduction de la notion d'intuition

Objectif : guider la recherche

Aucun impact sur la complexité théorique

# Heuristiques

Une traduction de la notion d'intuition

Objectif : guider la recherche

Aucun impact sur la complexité théorique

Mais des gains significatifs possibles en pratique

# Heuristiques sur les variables

Objectif : rencontrer l'échec le plus tôt possible (first-fail principle)

# Heuristiques sur les variables

Objectif : rencontrer l'échec le plus tôt possible (first-fail principle)

Exemples :

- Heuristique dom = choisir la variable  $x$  ayant le plus petit domaine

# Heuristiques sur les variables

Objectif : rencontrer l'échec le plus tôt possible (first-fail principle)

Exemples :

- Heuristique dom = choisir la variable  $x$  ayant le plus petit domaine
- Heuristique dom/deg = minimiser le rapport  $\frac{|d_x|}{\text{degré}(x)}$

# Heuristiques sur les variables

Objectif : rencontrer l'échec le plus tôt possible (first-fail principe)

Exemples :

- Heuristique dom = choisir la variable  $x$  ayant le plus petit domaine
- Heuristique dom/deg = minimiser le rapport  $\frac{|d_x|}{\text{degré}(x)}$
- Heuristique dom/wdeg = minimiser le rapport

$$\frac{|d_x|}{\sum_{c \in C | x \in S(c) \wedge \text{Fut}(S(c)) > 1} wdeg(c)}$$



# Heuristiques sur les valeurs

Objectif : choisir la valeur ayant le plus de chance de mener à une solution

# Heuristiques sur les valeurs

Objectif : choisir la valeur ayant le plus de chance de mener à une solution

Exemple :

- Heuristique = choisir la valeur qui maximise  $\prod_{x \in \text{Fut}(X)} |d_x|$

# Apprentissage

Mémoriser des informations explicitées durant la recherche

Objectifs :

- Éviter certaines redondances
- Accélérer la recherche

# Apprentissage

Mémoriser des informations explicitées durant la recherche

Objectifs :

- Éviter certaines redondances
- Accélérer la recherche

Inconvénients :

- un surcoût en temps pas toujours compensé par les économies réalisées
- le nombre d'informations à mémoriser peut être exponentiel

# Apprentissage

Mémoriser des informations explicitées durant la recherche

Objectifs :

- Éviter certaines redondances
- Accélérer la recherche

Inconvénients :

- un surcoût en temps pas toujours compensé par les économies réalisées
- le nombre d'informations à mémoriser peut être exponentiel

En pratique, on limite la quantité d'informations mémorisées.

# Apprentissage

Mémoriser des informations explicitées durant la recherche

Objectifs :

- Éviter certaines redondances
- Accélérer la recherche

Inconvénients :

- un surcoût en temps pas toujours compensé par les économies réalisées
- le nombre d'informations à mémoriser peut être exponentiel

En pratique, on limite la quantité d'informations mémorisées.

Exemple : enregistrement de nogood (= tuple interdit par une contrainte)

# Redémarrage

À effectuer quand la recherche ne progresse pas

# Redémarrage

À effectuer quand la recherche ne progresse pas

Détection basée sur le nombre d'échecs ou le nombre de retour en arrière



# Redémarrage

À effectuer quand la recherche ne progresse pas

Détection basée sur le nombre d'échecs ou le nombre de retour en arrière

Conservation d'une partie des informations apprises

# Redémarrage

À effectuer quand la recherche ne progresse pas

Détection basée sur le nombre d'échecs ou le nombre de retour en arrière

Conservation d'une partie des informations apprises

Introduction d'aléas

# Redémarrage

À effectuer quand la recherche ne progresse pas

Détection basée sur le nombre d'échecs ou le nombre de retour en arrière

Conservation d'une partie des informations apprises

Introduction d'aléas

Problème de terminaison

# Structure

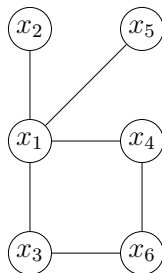
Hypergraphe de contraintes  $(X, C)$  :

- sommets = les variables
- hyper-arêtes = les portées des contraintes

# Structure

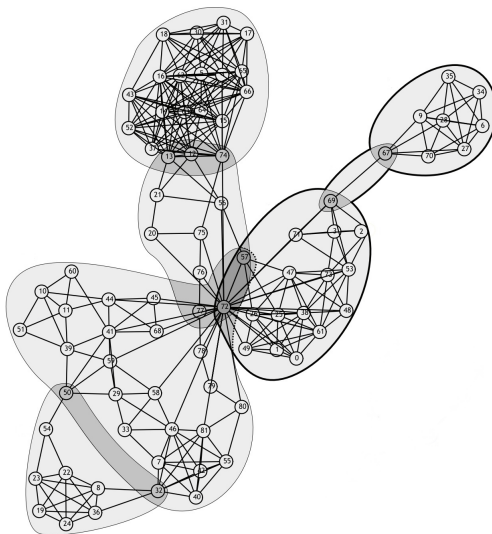
Hypergraphe de contraintes  $(X, C)$  :

- sommets = les variables
- hyper-arêtes = les portées des contraintes



# Structure

Possibilité d'identifier des sous-problèmes indépendants



# Quelques solveurs boîtes noires

De nombreux solveurs :

- AbsCon,
- BTD,
- Concrete,
- Choco,
- cosoco,
- Gecode,
- Mistral,
- OR-Tools,
- OscalaR,
- PicatSAT,
- SAT4J,
- ...

# Quelques solveurs boîtes noires

De nombreux solveurs :

- AbsCon,
- BTD,
- Concrete,
- Choco,
- cosoco,
- Gecode,
- Mistral,
- OR-Tools,
- OscalaR,
- PicatSAT,
- SAT4J,
- ...

Différents langages : C++, Java, Scala, ...



# Quelques solveurs boîtes noires

De nombreux solveurs :

- AbsCon,
- BTM,
- Concrete,
- Choco,
- cosoco,
- Gecode,
- Mistral,
- OR-Tools,
- OscanR,
- PicatSAT,
- SAT4J,
- ...

Différents langages : C++, Java, Scala, ...

Des compétitions :

- Compétition XCSP3 : <http://www.cril.univ-artois.fr/XCSP18/>
- MiniZinc Challenge :  
<http://www.minizinc.org/challenge2018/challenge.html>

# Plan

- 1 Contexte
- 2 Formalisme
- 3 Résolution
- 4 Quelques exemples de modélisation**
- 5 Les limites

# Contraintes globales

Un patron de contrainte ayant une certaine sémantique et applicable à un nombre quelconque de variables

# Contraintes globales

Un patron de contrainte ayant une certaine sémantique et applicable à un nombre quelconque de variables

Avantages :

- facilitent la modélisation
- bénéficient d'algorithmes de filtrage adaptés
- rendent souvent la résolution plus efficace

# Contraintes globales

Un patron de contrainte ayant une certaine sémantique et applicable à un nombre quelconque de variables

Avantages :

- facilitent la modélisation
- bénéficient d'algorithmes de filtrage adaptés
- rendent souvent la résolution plus efficace

Inconvénient :

- requièrent plus d'expertise lors de la modélisation

# La contrainte All-different

Une des contraintes les plus fréquentes

# La contrainte All-different

Une des contraintes les plus fréquentes

Soit  $Y = \{x_{i_1}, \dots, x_{i_r}\}$  avec  $r \geq 2$

*All-different*( $Y$ ) impose que les valeurs des  $x_{i_j}$  soient toutes distinctes.

# La contrainte All-different

Une des contraintes les plus fréquentes

Soit  $Y = \{x_{i_1}, \dots, x_{i_r}\}$  avec  $r \geq 2$

*All-different*( $Y$ ) impose que les valeurs des  $x_{i_j}$  soient toutes distinctes.

Existence de plusieurs variantes :

- *All-different-matrix*
- *All-different-except*
- ...



## Sudoku : une deuxième modélisation

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

- $X = \{x_1, x_2, \dots, x_{81}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{81}}\}$  avec
  - $d_{x_1} = \{2\},$
  - $d_{x_2} = \{4\},$
  - $d_{x_3} = \{1\},$
  - $d_{x_4} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},$
  - $\dots$

## Sudoku : une deuxième modélisation

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

- $X = \{x_1, x_2, \dots, x_{81}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{81}}\}$  avec
  - $d_{x_1} = \{2\},$
  - $d_{x_2} = \{4\},$
  - $d_{x_3} = \{1\},$
  - $d_{x_4} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},$
  - $\dots$
- $C = \{c_{l_1}, \dots, c_{l_9}, c_{c_1}, \dots, c_{c_9}, c_{r_1}, \dots, c_{r_9}\}$

## Sudoku : une deuxième modélisation

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

- $X = \{x_1, x_2, \dots, x_{81}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{81}}\}$  avec
  - $d_{x_1} = \{2\},$
  - $d_{x_2} = \{4\},$
  - $d_{x_3} = \{1\},$
  - $d_{x_4} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},$
  - $\dots$
- $C = \{c_{l_1}, \dots, c_{l_9}, c_{c_1}, \dots, c_{c_9}, c_{r_1}, \dots, c_{r_9}\}$   
 $c_{l_1} = \text{All-Different}(\{x_1, x_2, \dots, x_9\})$

## Sudoku : une deuxième modélisation

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

- $X = \{x_1, x_2, \dots, x_{81}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{81}}\}$  avec
  - $d_{x_1} = \{2\},$
  - $d_{x_2} = \{4\},$
  - $d_{x_3} = \{1\},$
  - $d_{x_4} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},$
  - $\dots$
- $C = \{c_{l_1}, \dots, c_{l_9}, c_{c_1}, \dots, c_{c_9}, c_{r_1}, \dots, c_{r_9}\}$ 
  - $c_{l_1} = \text{All-Different}(\{x_1, x_2, \dots, x_9\})$
  - $c_{c_1} = \text{All-Different}(\{x_1, x_{10}, x_{19}, \dots, x_{73}\})$

## Sudoku : une deuxième modélisation

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

- $X = \{x_1, x_2, \dots, x_{81}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{81}}\}$  avec
  - $d_{x_1} = \{2\},$
  - $d_{x_2} = \{4\},$
  - $d_{x_3} = \{1\},$
  - $d_{x_4} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},$
  - $\dots$
- $C = \{c_{l_1}, \dots, c_{l_9}, c_{c_1}, \dots, c_{c_9}, c_{r_1}, \dots, c_{r_9}\}$ 
  - $c_{l_1} = \text{All-Different}(\{x_1, x_2, \dots, x_9\})$
  - $c_{c_1} = \text{All-Different}(\{x_1, x_{10}, x_{19}, \dots, x_{73}\})$
  - $c_{r_4} = \text{All-Different}(\{x_{28}, x_{29}, x_{30}, x_{37}, \dots, x_{48}\})$

## Sudoku : une deuxième modélisation

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

- $X = \{x_1, x_2, \dots, x_{81}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{81}}\}$  avec
  - $d_{x_1} = \{2\},$
  - $d_{x_2} = \{4\},$
  - $d_{x_3} = \{1\},$
  - $d_{x_4} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},$
  - $\dots$
- $C = \{c_{l_1}, \dots, c_{l_9}, c_{c_1}, \dots, c_{c_9}, c_{r_1}, \dots, c_{r_9}\}$ 
  - $c_{l_1} = \text{All-Different}(\{x_1, x_2, \dots, x_9\})$
  - $c_{c_1} = \text{All-Different}(\{x_1, x_{10}, x_{19}, \dots, x_{73}\})$
  - $c_{r_4} = \text{All-Different}(\{x_{28}, x_{29}, x_{30}, x_{37}, \dots, x_{48}\})$

Intérêt de cette modélisation ?

## Sudoku : une deuxième modélisation

2	4	1						
			4			1		
	5		9					6
		2					6	7
	7				5			
3			8					2
			7				3	
		8		6			4	5
4			3					

- $X = \{x_1, x_2, \dots, x_{81}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{81}}\}$  avec
  - $d_{x_1} = \{2\},$
  - $d_{x_2} = \{4\},$
  - $d_{x_3} = \{1\},$
  - $d_{x_4} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},$
  - $\dots$
- $C = \{c_{l_1}, \dots, c_{l_9}, c_{c_1}, \dots, c_{c_9}, c_{r_1}, \dots, c_{r_9}\}$ 
  - $c_{l_1} = \text{All-Different}(\{x_1, x_2, \dots, x_9\})$
  - $c_{c_1} = \text{All-Different}(\{x_1, x_{10}, x_{19}, \dots, x_{73}\})$
  - $c_{r_4} = \text{All-Different}(\{x_{28}, x_{29}, x_{30}, x_{37}, \dots, x_{48}\})$

Intérêt de cette modélisation ?

Filtrage d'*All-Different* plus puissant que celui de  $x \neq y$

# Contrainte de différence vs contrainte All-different

- $X = \{x_1, x_2, x_3\}$
- $D = \{d_{x_1}, d_{x_2}, d_{x_3}\}$  avec
  - $d_{x_1} = d_{x_2} = \{1, 2\}$
  - $d_{x_3} = \{1, 2, 3\}$
- $C = \{c_{12}, c_{13}, c_{23}\}$  avec
  - $S(c_{ij}) = \{x_i, x_j\}$
  - $R(c_{ij}) = \{x_i \neq x_j\}$

- $X = \{x_1, x_2, x_3\}$
- $D = \{d_{x_1}, d_{x_2}, d_{x_3}\}$  avec
  - $d_{x_1} = d_{x_2} = \{1, 2\}$
  - $d_{x_3} = \{1, 2, 3\}$
- $C = \{All\text{-}Different(X)\}$



# Contrainte de différence vs contrainte All-different

- $X = \{x_1, x_2, x_3\}$
- $D = \{d_{x_1}, d_{x_2}, d_{x_3}\}$  avec
  - $d_{x_1} = d_{x_2} = \{1, 2\}$
  - $d_{x_3} = \{1, 2, 3\}$
- $C = \{c_{12}, c_{13}, c_{23}\}$  avec
  - $S(c_{ij}) = \{x_i, x_j\}$
  - $R(c_{ij}) = \{x_i \neq x_j\}$

- $X = \{x_1, x_2, x_3\}$
- $D = \{d_{x_1}, d_{x_2}, d_{x_3}\}$  avec
  - $d_{x_1} = d_{x_2} = \{1, 2\}$
  - $d_{x_3} = \{1, 2, 3\}$
- $C = \{All\text{-}Different(X)\}$

Aucune suppression

# Contrainte de différence vs contrainte All-different

- $X = \{x_1, x_2, x_3\}$
- $D = \{d_{x_1}, d_{x_2}, d_{x_3}\}$  avec
  - $d_{x_1} = d_{x_2} = \{1, 2\}$
  - $d_{x_3} = \{1, 2, 3\}$
- $C = \{c_{12}, c_{13}, c_{23}\}$  avec
  - $S(c_{ij}) = \{x_i, x_j\}$
  - $R(c_{ij}) = \{x_i \neq x_j\}$

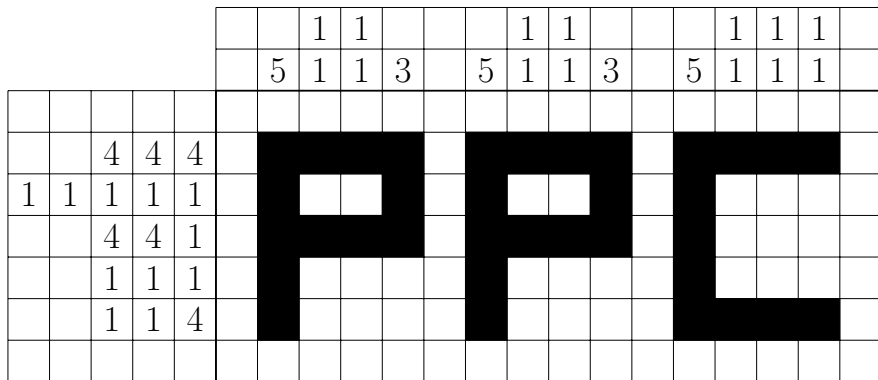
Aucune suppression

- $X = \{x_1, x_2, x_3\}$
- $D = \{d_{x_1}, d_{x_2}, d_{x_3}\}$  avec
  - $d_{x_1} = d_{x_2} = \{1, 2\}$
  - $d_{x_3} = \{1, 2, 3\}$
- $C = \{All\text{-}Different(X)\}$

Suppression de 1 et 2 de  $d_{x_3}$

[illegible]

# Nonogramme



							1	1				1	1				1	1	1	
						5	1	1	3		5	1	1	3		5	1	1	1	
					$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	$x_{11}$	$x_{12}$	$x_{13}$	$x_{14}$	$x_{15}$	$x_{16}$
		4	4	4	$x_{17}$	$x_{18}$	$x_{19}$	$x_{20}$	$x_{21}$	$x_{22}$	$x_{23}$	$x_{24}$	$x_{25}$	$x_{26}$	$x_{27}$	$x_{28}$	$x_{29}$	$x_{30}$	$x_{31}$	$x_{32}$
1	1	1	1	1	$x_{33}$	$x_{34}$	$x_{35}$	$x_{36}$	$x_{37}$	$x_{38}$	$x_{39}$	$x_{40}$	$x_{41}$	$x_{42}$	$x_{43}$	$x_{44}$	$x_{45}$	$x_{46}$	$x_{47}$	$x_{48}$
		4	4	1	$x_{49}$	$x_{50}$	$x_{51}$	$x_{52}$	$x_{53}$	$x_{54}$	$x_{55}$	$x_{56}$	$x_{57}$	$x_{58}$	$x_{59}$	$x_{60}$	$x_{61}$	$x_{62}$	$x_{63}$	$x_{64}$
		1	1	1	$x_{65}$	$x_{66}$	$x_{67}$	$x_{68}$	$x_{69}$	$x_{70}$	$x_{71}$	$x_{72}$	$x_{73}$	$x_{74}$	$x_{75}$	$x_{76}$	$x_{77}$	$x_{78}$	$x_{79}$	$x_{80}$
		1	1	4	$x_{81}$	$x_{82}$	$x_{83}$	$x_{84}$	$x_{85}$	$x_{86}$	$x_{87}$	$x_{88}$	$x_{89}$	$x_{90}$	$x_{91}$	$x_{92}$	$x_{93}$	$x_{94}$	$x_{95}$	$x_{96}$
					$x_{97}$	$x_{98}$	$x_{99}$	$x_{100}$	$x_{101}$	$x_{102}$	$x_{103}$	$x_{104}$	$x_{105}$	$x_{106}$	$x_{107}$	$x_{108}$	$x_{109}$	$x_{110}$	$x_{111}$	$x_{112}$

- $X = \{x_1, x_2, \dots, x_{112}\}$

# Nonogramme

[illegible]

- $X = \{x_1, x_2, \dots, x_{112}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{112}}\}$  avec  $\forall i, d_{x_i} = \{0, 1\}$

[illegible]

- $X = \{x_1, x_2, \dots, x_{112}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{112}}\}$  avec  $\forall i, d_{x_i} = \{0, 1\}$
- $C = \{c_{I_1}, \dots, c_{I_8}, c_{C_1}, \dots, c_{C_{16}}\}$

[illegible]

- ## Comment représenter ces contraintes?



# Nonogramme

[illegible]

- $X = \{x_1, x_2, \dots, x_{112}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{112}}\}$  avec  $\forall i, d_{x_i} = \{0, 1\}$
- $C = \{c_{I_1}, \dots, c_{I_8}, c_{C_1}, \dots, c_{C_{16}}\}$

## Comment représenter ces contraintes?

Contraintes Tables : possible, mais peut être coûteux en mémoire

# La contrainte Regular

Soit  $Y = \{x_{i_1}, \dots, x_{i_r}\}$

Soit un automate fini déterministe  $A$

Soit  $L(A)$  le langage reconnu par  $A$

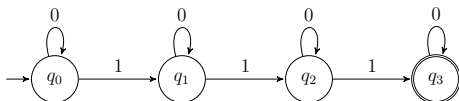
$Regular(Y, A)$  impose que les valeurs des  $x_{i_j}$  forment un mot accepté par  $L(A)$

[illegible]

- $X = \{x_1, x_2, \dots, x_{112}\}$
- $D = \{d_{x_1}, d_{x_2}, \dots, d_{x_{112}}\}$  avec  $\forall i, d_{x_i} = \{0, 1\}$
- $C = \{c_{j_1}, \dots, c_{j_8}, c_{c_1}, \dots, c_{c_{16}}\}$

[illegible]

- Exemple : automate pour  $c_5$



# Plan

- 1 Contexte
- 2 Formalisme
- 3 Résolution
- 4 Quelques exemples de modélisation
- 5 Les limites

# Des limites

Certaines notions posent problèmes :

- Recherche d'une solution optimisant un certain critère

# Des limites

Certaines notions posent problèmes :

- Recherche d'une solution optimisant un certain critère
- Contraintes au-delà du vrai et du faux :
  - possibilité,
  - probabilité,
  - préférence,
  - ...

# Des limites

Certaines notions posent problèmes :

- Recherche d'une solution optimisant un certain critère
- Contraintes au-delà du vrai et du faux :
  - possibilité,
  - probabilité,
  - préférence,
  - ...
- ...



# Des limites

Certaines notions posent problèmes :

- Recherche d'une solution optimisant un certain critère
- Contraintes au-delà du vrai et du faux :
  - possibilité,
  - probabilité,
  - préférence,
  - ...
- ...

D'autres formalismes :

- Problèmes d'optimisation sous contraintes (COP)
- Problèmes de satisfaction de contraintes valués (VCSP)