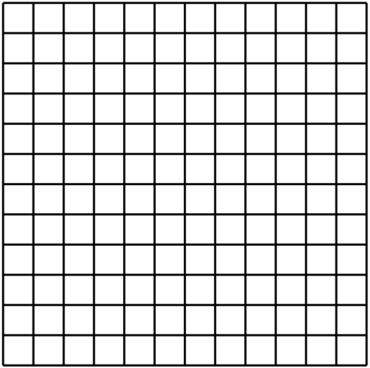
# Résoudre le jeu de Picross

## avec la Programmation par Contraintes

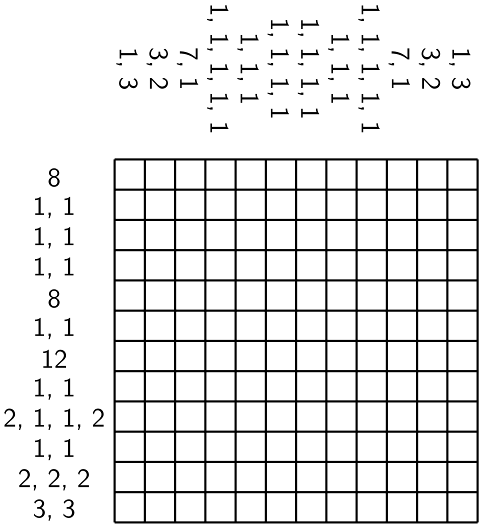
Benoît BOMPOL

14 octobre 2024

# Picross



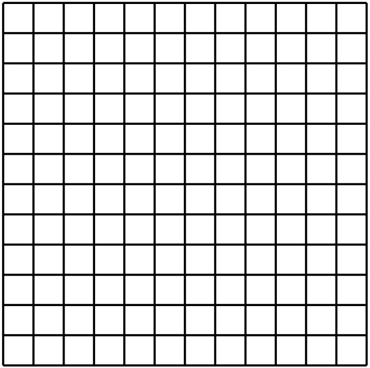
# Picross


1, 3
3, 2
7, 1
1, 1, 1, 1, 1
1, 1, 1
1, 1, 1, 1
1, 1, 1, 1
1, 1, 1
1, 1, 1, 1, 1
7, 1
3, 2
1, 3
8
1, 1
1, 1
1, 1
8
1, 1
12
1, 1
2, 1, 1, 2
1, 1
2, 2, 2
3, 3

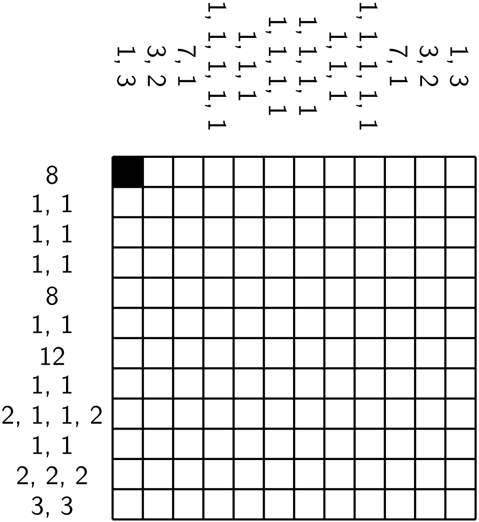
# Picross

1, 3
3, 2
7, 1
1, 1, 1, 1, 1
1, 1, 1
1, 1, 1, 1
1, 1, 1, 1
1, 1, 1
1, 1, 1, 1, 1
7, 1
3, 2
1, 3

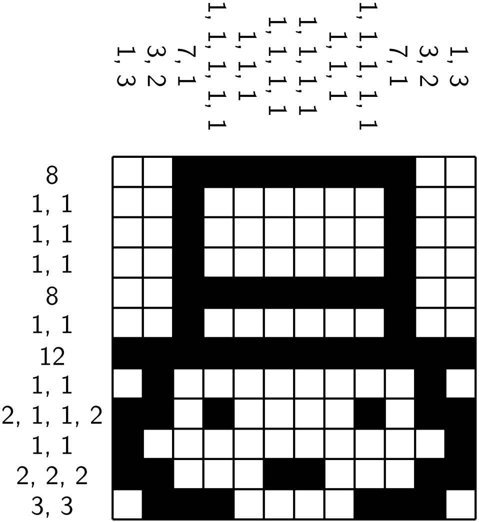
8
1, 1
1, 1
1, 1
8
1, 1
12
1, 1
**2, 1, 1, 2**
1, 1
2, 2, 2
3, 3



# Picross


1, 3
3, 2
7, 1
1, 1, 1, 1, 1
1, 1, 1
1, 1, 1, 1
1, 1, 1, 1
1, 1, 1
1, 1, 1, 1, 1
7, 1
3, 2
1, 3
8
1, 1
1, 1
1, 1
8
1, 1
12
1, 1
2, 1, 1, 2
1, 1
2, 2, 2
3, 3

# Picross


1, 3
3, 2
7, 1
1, 1, 1, 1, 1, 1
1, 1, 1, 1
1, 1, 1, 1, 1
1, 1, 1, 1, 1
1, 1, 1, 1
1, 1, 1, 1, 1, 1
7, 1
3, 2
1, 3
8
1, 1
1, 1
1, 1
8
1, 1
12
1, 1
2, 1, 1, 2
1, 1
2, 2, 2
3, 3

# Enjeux

- Avant tout un problème de **modélisation**
- Découvrir la **réification**
- Concevoir un outil de **résolution** *automatique*

$n$ lignes, $m$ colonnes

# Enjeux

- Avant tout un problème de **modélisation**
- Découvrir la **réification**
- Concevoir un outil de **résolution** *automatique*

$n$ lignes, $m$ colonnes

# Enjeux

- Avant tout un problème de **modélisation**
- Découvrir la **réification**
- Concevoir un outil de **résolution** *automatique*

$n$ lignes, $m$ colonnes

# Enjeux

- Avant tout un problème de **modélisation**
- Découvrir la **réification**
- Concevoir un outil de **résolution** *automatique*

$n$ lignes, $m$ colonnes : $2^{n\times m}$ possibilités, trop pour énumérer

# Représentation

- $n$ lignes, $m$ colonnes
- R[$i$] l'ensemble des contraintes sur la ligne $i$.
  - C[$j$] représente celles sur la colonne $j$.
  - *Exemple :* $\mathrm{R}[8] = \{2, 1, 1, 2\}$

**Solution** :

- Une grille de taille $(n, m)$
- Chaque *bloc* sur une ligne est séparé par (au moins) une case blanche
- Chaque *bloc* sur une colonne est séparé par (au moins) une case blanche
- **On suppose qu'il n'y a qu'une seule solution par instance**

# Représentation

- $n$ lignes, $m$ colonnes
- R[$i$] l'ensemble des contraintes sur la ligne $i$.
  - C[$j$] représente celles sur la colonne $j$.
  - *Exemple :* $\mathrm{R}[8] = \{2, 1, 1, 2\}$

**Solution** :

- Une grille de taille $(n, m)$
- Chaque *bloc* sur une ligne est séparé par (au moins) une case blanche
- Chaque *bloc* sur une colonne est séparé par (au moins) une case blanche
- **On suppose qu'il n'y a qu'une seule solution par instance**

# Représentation

- $n$ lignes, $m$ colonnes
- R[$i$] l'ensemble des contraintes sur la ligne $i$.
  - C[$j$] représente celles sur la colonne $j$.
  - *Exemple :* $\mathrm{R}[8] = \{2, 1, 1, 2\}$

**Solution** :

- Une grille de taille $(n, m)$
- Chaque *bloc* sur une ligne est séparé par (au moins) une case blanche
- Chaque *bloc* sur une colonne est séparé par (au moins) une case blanche
- **On suppose qu'il n'y a qu'une seule solution par instance**

# Représentation

- $n$ lignes, $m$ colonnes
- R[$i$] l'ensemble des contraintes sur la ligne $i$.
  - C[$j$] représente celles sur la colonne $j$.
  - *Exemple :* $\mathrm{R}[8] = \{2, 1, 1, 2\}$

**Solution** :

- Une grille de taille $(n, m)$
- Chaque *bloc* sur une ligne est séparé par (au moins) une case blanche
- Chaque *bloc* sur une colonne est séparé par (au moins) une case blanche
- **On suppose qu'il n'y a qu'une seule solution par instance**

# Méthodologie, choix des variables

- Introduction de *redondance* / *dépendance* dans le choix des variables
- $\text{grid}[i, j] \in \{0, 1\}$, 1 **ssi** la case $(i, j)$ est coloriée.
- $\text{startX}[i, k]$ est la position de début du $k$-ème bloc de la ligne $i$.
- $\text{startY}[j, \ell]$ est la position de début du $\ell$-ème bloc de la colonne $j$.

# Méthodologie, choix des variables

- Introduction de *redondance* / *dépendance* dans le choix des variables
- $\text{grid}[i,j] \in \{0,1\}$, 1 **ssi** la case $(i,j)$ est coloriée.
- $\text{startX}[i,k]$ est la position de début du $k$-ème bloc de la ligne $i$.
- $\text{startY}[j,\ell]$ est la position de début du $\ell$-ème bloc de la colonne $j$.

# Méthodologie, choix des variables

- Introduction de *redondance* / *dépendance* dans le choix des variables
- grid$[i, j] \in \{0, 1\}$, 1 **ssi** la case $(i, j)$ est coloriée.
- startX$[i, k]$ est la position de début du $k$-ème bloc de la ligne $i$.
- startY$[j, \ell]$ est la position de début du $\ell$-ème bloc de la colonne $j$.

# Méthodologie, choix des variables

- Introduction de *redondance* / *dépendance* dans le choix des variables
- $\text{grid}[i,j] \in \{0,1\}$, 1 **ssi** la case $(i,j)$ est coloriée.
- $\text{startX}[i,k]$ est la position de début du $k$-ème bloc de la ligne $i$.
- $\text{startY}[j,\ell]$ est la position de début du $\ell$-ème bloc de la colonne $j$.

# Contraintes (A) : Compter les cases coloriées

- Objectif de la contrainte :
  - S'assurer du bon nombre de cases coloriées dans une unité
- Nombre de cases coloriées sur la ligne $i$ : $\sum_{k \in \mathrm{R}[i]} k$
  - Contrainte associée : $\sum_{j \in 0 \ldots m-1} \mathrm{grid}[i,j] = \sum_{k \in \mathrm{R}[i]} k$
- Contrainte analogue sur la colonne $j$

# Contraintes (A) : Compter les cases coloriées

- Objectif de la contrainte :
  - S'assurer du bon nombre de cases coloriées dans une unité
- Nombre de cases coloriées sur la ligne $i$ : $\sum_{k \in \mathrm{R}[i]} k$
  - Contrainte associée : $\sum_{j \in 0 \ldots m-1} \mathrm{grid}[i,j] = \sum_{k \in \mathrm{R}[i]} k$
- Contrainte analogue sur la colonne $j$

# Contraintes (A) : Compter les cases coloriées

- Objectif de la contrainte :
  - S'assurer du bon nombre de cases coloriées dans une unité
- Nombre de cases coloriées sur la ligne $i$ : $\sum_{k \in \mathrm{R}[i]} k$
  - Contrainte associée : $\sum_{j \in 0 \ldots m-1} \mathrm{grid}[i,j] = \sum_{k \in \mathrm{R}[i]} k$
- Contrainte analogue sur la colonne $j$

# Contraintes (B) : Précédence de deux blocs sur la même unité

- Sur la ligne $i$, le bloc $k+1$ commence après la fin du bloc $k$ **et** une case vide
  - Contrainte analogue sur le bloc $\ell$ de la colonne $j$.

Deux ensembles de contraintes à implémenter :

# Contraintes (B) : Précédence de deux blocs sur la même unité

- Sur la ligne $i$, le bloc $k+1$ commence après la fin du bloc $k$ **et** une case vide
  - Contrainte analogue sur le bloc $\ell$ de la colonne $j$.

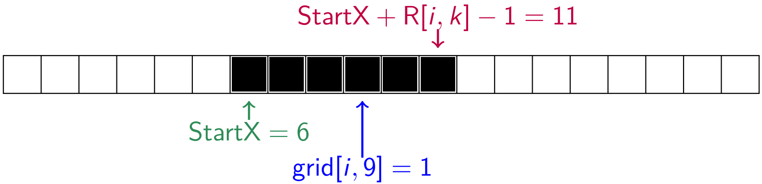
Deux ensembles de contraintes à implémenter :

- $\text{startX}[i, k+1] \geq \text{startX}[i, k] + [R][i, k] + 1$
- Contrainte analogue sur les colonnes

# Contraintes (C) : Lier les variables entre elles

- Principe de la contrainte :
  - Lier le fait que $(i, j)$ soit coloriée, et la position de départ des blocs
- Si le bloc commence avant cette case

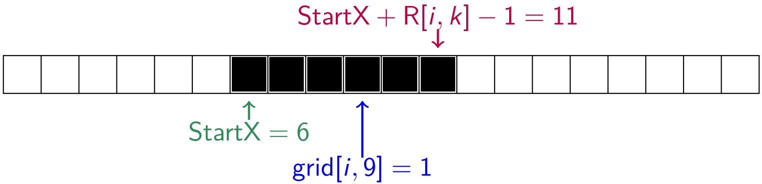
**et** si le **même** bloc finit après cette case



# Contraintes (C) : Lier les variables entre elles

- Principe de la contrainte :
  - Lier le fait que $(i, j)$ soit coloriée, et la position de départ des blocs
- Si le bloc commence avant cette case

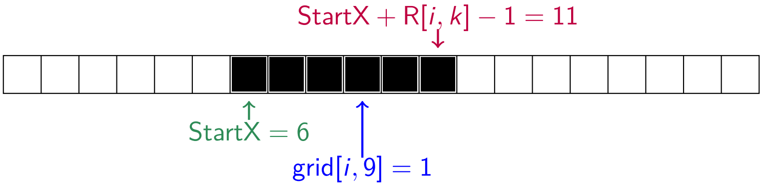
**et** si le **même** bloc finit après cette case



# Contraintes (C) : Lier les variables entre elles

- Principe de la contrainte :
  - Lier le fait que $(i, j)$ soit coloriée, et la position de départ des blocs
- Si le bloc commence avant cette case

**et** si le **même** bloc finit après cette case




$$\text{grid}[i, j] = 1 \iff \bigvee_{k \in 1 \ldots |\text{R}(i)|} (\text{StartX}[i, k] \leq j) \wedge (\text{StartX}[i, k] \geq j - \text{R}[i, k] + 1)$$

Contrainte analogue sur les colonnes.

# Contraintes (D,E) : Stacking des blocs

- Objectif de ces contraintes :
  - Limiter les positions minimales (D), maximales (E) des débuts de blocs
- Méthode utilisée :
  - Compter les blocs coloriés à gauche (position minimale de début de bloc)
  - Compter les blocs coloriés à droite (position maximale de fin de bloc)

- Limiter la position **minimale** : $\mathrm{startX}[i,k] \geq \underbrace{\left(\sum_{k'<k} \mathrm{R}[i,k]\right)}_{\text{Blocs à gauche}} + k$
- Limiter la position **maximale** : $\mathrm{startX}[i,k] \leq m - (|\mathrm{R}(i)| - k - 1) - \underbrace{\sum_{k' \geq k} \mathrm{R}[i,k]}_{\substack{\text{Blocs à droite} \\ \text{incluant le courant}}}$

# Contraintes (D,E) : Stacking des blocs

- Objectif de ces contraintes :
  - Limiter les positions minimales (D), maximales (E) des débuts de blocs
- Méthode utilisée :
  - Compter les blocs coloriés à gauche (position minimale de début de bloc)
  - Compter les blocs coloriés à droite (position maximale de fin de bloc)

  Dans le détail :

  - Limiter la position **minimale** : $\text{startX}[i,k] \geq \underbrace{\left(\sum_{k'<k} \text{R}[i,k]\right)}_{\text{Blocs à gauche}} + k$
  - Limiter la position **maximale** : $\text{startX}[i,k] \leq m - (|\text{R}(i)| - k - 1) - \underbrace{\sum_{k' \geq k} \text{R}[i,k]}_{\substack{\text{Blocs à droite} \\ \text{incluant le courant}}}$

# Contraintes (D,E) : Stacking des blocs

- Objectif de ces contraintes :
  - Limiter les positions minimales (D), maximales (E) des débuts de blocs
- Méthode utilisée :
  - Compter les blocs coloriés à gauche (position minimale de début de bloc)
  - Compter les blocs coloriés à droite (position maximale de fin de bloc)

Dans le détail :

- Limiter la position **minimale** : $\text{startX}[i,k] \geq \underbrace{\left(\sum_{k'<k} \text{R}[i,k]\right)}_{\text{Blocs à gauche}} + k$
- Limiter la position **maximale** : $\text{startX}[i,k] \leq m - (|\text{R}(i)| - k - 1) - \underbrace{\sum_{k' \geq k} \text{R}[i,k]}_{\text{Blocs à droite incluant le courant}}$

# Contraintes (D,E) : Stacking des blocs

- Objectif de ces contraintes :
  - Limiter les positions minimales (D), maximales (E) des débuts de blocs
- Méthode utilisée :
  - Compter les blocs coloriés à gauche (position minimale de début de bloc)
  - Compter les blocs coloriés à droite (position maximale de fin de bloc)

Dans le détail :

- Limiter la position **minimale** : $\mathrm{startX}[i,k] \geq \underbrace{\left(\sum_{k'<k} \mathrm{R}[i,k]\right)}_{\text{Blocs à gauche}} + k$
- Limiter la position **maximale** : $\mathrm{startX}[i,k] \leq m - (|\mathrm{R}(i)| - k - 1) - \underbrace{\sum_{k' \geq k} \mathrm{R}[i,k]}_{\substack{\text{Blocs à droite} \\ \text{incluant le courant}}}$

# Contraintes (D,E) : Stacking des blocs

- Objectif de ces contraintes :
  - Limiter les positions minimales (D), maximales (E) des débuts de blocs
- Méthode utilisée :
  - Compter les blocs coloriés à gauche (position minimale de début de bloc)
  - Compter les blocs coloriés à droite (position maximale de fin de bloc)

Dans le détail :

- Limiter la position **minimale** : $\text{startX}[i,k] \geq \underbrace{\left(\sum_{k'<k} \text{R}[i,k]\right)}_{\text{Blocs à gauche}} + k$
- Limiter la position **maximale** : $\text{startX}[i,k] \leq m - (|\text{R}(i)| - k - 1) - \underbrace{\sum_{k' \geq k} \text{R}[i,k]}_{\substack{\text{Blocs à droite} \\ \text{incluant le courant}}}$

Méthode analogue pour les colonnes

# Contraintes (F) : Entourer un bloc de vide

Objectif des contraintes :

1. Si un bloc commence à un endroit, la case qui précède est vide
   - $\text{startX}[i,k] = j \implies \text{grid}[i, j-1] = 0$
   - $\text{startY}[j,\ell] = i \implies \text{grid}[i-1, j] = 0$
2. Si un bloc commence à un endroit, la case qui suit la dernière est vide
   - $\text{startX}[i,k] = j \implies \text{grid}[i, j + \text{R}[i,k]] = 0$
   - $\text{startY}[j,\ell] = i \implies \text{grid}[i + \text{C}[j,\ell], j] = 0$

# Contraintes (F) : Entourer un bloc de vide

Objectif des contraintes :

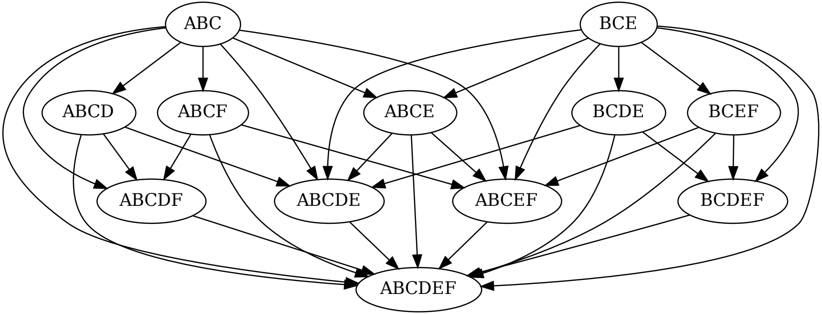
1. Si un bloc commence à un endroit, la case qui précède est vide
   - $\text{startX}[i,k] = j \implies \text{grid}[i, j-1] = 0$
   - $\text{startY}[j,\ell] = i \implies \text{grid}[i-1, j] = 0$
2. Si un bloc commence à un endroit, la case qui suit la dernière est vide
   - $\text{startX}[i,k] = j \implies \text{grid}[i, j+\text{R}[i,k]] = 0$
   - $\text{startY}[j,\ell] = i \implies \text{grid}[i+\text{C}[j,\ell], j] = 0$

# Travail effectué

**Objectif principal** : Déterminer l'importance de chaque bloc de contraintes.
Pour chaque instance :

- On trace un graphe $D = (V, A)$ avec :
  - Chaque sommet $v \in V$ est une **combinaison de contraintes** suffisante pour résoudre l'instance
  - $uv \in A \iff (u \in V) \wedge (v \in V) \wedge (u \subseteq v)$
- On cherche tous les sommets $u$ sans arc entrant. (Ce sont les minimaux)

# Résultats




Deux points (*minimaux*) d'entrée : $A, B, C$ et $B, C, E$

- Synthétiser les *points d'entrée* sur chaque instance permettrait de la classifier suivant sa difficulté apparente, le *nombre* de raisonnements **différents** nécessaires pour la résoudre

# Synthèse des résultats

| | bird | clock | ratz | knife | layton | pikachu | tardis | spade | godzilla | hare | kabuki | ouhbatman | panda | qrcode | zen |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BC | X | X | | | | X | X | | X | X | X | X | X | X | X |
| CDE | X | X | | | | | X | | | | | | | | |
| CEF | X | X | | | | X | | | | | | | | | |
| ABC | | | X | X | X | | | | | | | | | | |
| BCE | | | X | X | X | | | | | | | | | | |
| ACE | X | X | | | | | | | | | | | | | |
| C | | | | | | | | X | | | | | | | |
| ACD | X | | | | | | | | | | | | | | |
| CDF | X | | | | | | | | | | | | | | |
| ACDF | | X | | | | | | | | | | | | | |
| BCF | | | X | | | | | | | | | | | | |

Quelques observations :

1. **spade** est de loin l'instance la plus facile à résoudre
2. Les contraintes $B, C$ permettent de résoudre, seules, 80% des instances
3. Les instances `ratz`, `knife`, `layton` nécessitent au minimum trois types de raisonnements, et sont plus complexes à résoudre
4. Les instances `clock`, `bird` sont similaires en termes de difficultés