

## Chapitre 9 : Logique propositionnelle (le problème SAT)

slides disponibles sur <http://cahier-de-prepa/mp2i-fermat>

# Plan

1. Syntaxe de la logique propositionnelle
2. Algèbre de Boole
3. Sémantique de la logique propositionnelle
4. Mise sous forme normale
5. Le problème SAT

# Plan

1. Syntaxe de la logique propositionnelle
2. Algèbre de Boole
3. Sémantique de la logique propositionnelle
4. Mise sous forme normale
5. Le problème SAT

# Plan

1. Syntaxe de la logique propositionnelle
2. Algèbre de Boole
3. **Sémantique de la logique propositionnelle**
4. Mise sous forme normale
5. Le problème SAT

# Plan

1. Syntaxe de la logique propositionnelle
2. Algèbre de Boole
3. Sémantique de la logique propositionnelle
4. Mise sous forme normale
5. Le problème SAT

# Plan

1. Syntaxe de la logique propositionnelle
2. Algèbre de Boole
3. Sémantique de la logique propositionnelle
4. Mise sous forme normale
5. Le problème SAT
  - Définitions
  - Réductions
  - Modéliser des FND ou FNC au regard de la satisfiabilité
  - FND-SAT : un problème facile
  - Puissance d'encodage de 3-SAT

# Le problème

Dans cette section on s'intéresse à la satisfiabilité des formules (sur un ensemble de variables fini, *i.e.*  $\text{Card}(\mathcal{Q}) \in \mathbb{N}$ .)

**Pourquoi ?**

# Le problème

Dans cette section on s'intéresse à la satisfiabilité des formules (sur un ensemble de variables fini, *i.e.*  $\text{Card}(\mathcal{Q}) \in \mathbb{N}$ .)

**Pourquoi ?** Une formule propositionnelle peut modéliser un problème concret dont une solution serait donnée par un environnement satisfaisant la formule.



# Le problème

Dans cette section on s'intéresse à la satisfiabilité des formules (sur un ensemble de variables fini, *i.e.*  $\text{Card}(\mathcal{Q}) \in \mathbb{N}$ .)

**Pourquoi ?** Une formule propositionnelle peut modéliser un problème concret dont une solution serait donnée par un environnement satisfaisant la formule.

exemple du sudoku : la valeur de vérité d'une variable  $p_{i,j,k}$  indique si la case  $i,j$  contient la valeur  $k$ , l'environnement complet décrit une solution, et s'il satisfait la formule, alors c'est une solution valide.

# Le problème

Dans cette section on s'intéresse à la satisfiabilité des formules (sur un ensemble de variables fini, *i.e.*  $\text{Card}(\mathcal{Q}) \in \mathbb{N}$ .)

**Pourquoi ?** Une formule propositionnelle peut modéliser un problème concret dont une solution serait donnée par un environnement satisfaisant la formule.

exemple du sudoku : la valeur de vérité d'une variable  $p_{i,j,k}$  indique si la case  $i,j$  contient la valeur  $k$ , l'environnement complet décrit une solution, et s'il satisfait la formule, alors c'est une solution valide.

Mais on s'intéresse déjà au problème de décision (plus simple) de savoir si une formule est satisfiable, sans demander par quel environnement.

**SAT** || entrée :  $A \in \mathbb{F}_p(\mathcal{Q})$   
|| sortie : oui si  $A$  est satisfiable, non sinon

# Le problème

Dans cette section on s'intéresse à la satisfiabilité des formules (sur un ensemble de variables fini, *i.e.*  $\text{Card}(Q) \in \mathbb{N}$ .)

**Pourquoi ?** Une formule propositionnelle peut modéliser un problème concret dont une solution serait donnée par un environnement satisfaisant la formule.

exemple du sudoku : la valeur de vérité d'une variable  $p_{i,j,k}$  indique si la case  $i,j$  contient la valeur  $k$ , l'environnement complet décrit une solution, et s'il satisfait la formule, alors c'est une solution valide.

Mais on s'intéresse déjà au problème de décision (plus simple) de savoir si une formule est satisfiable, sans demander par quel environnement.

**SAT**  $\left\| \begin{array}{l} \text{entrée : } A \in \mathbb{F}_p(Q) \\ \text{sortie : oui si } A \text{ est satisfiable, non sinon} \end{array} \right.$

**Remarque** : puisque la satisfiabilité d'une formule ne dépend que de sa classe, on peut résoudre le problème sur une formule ayant une forme particulière, mais attention au coût de transformation.

# Variantes du problème SAT

**SAT**      || entrée :  $A \in \mathbb{F}_p(Q)$   
             || sortie : oui si  $A$  est satisfiable, non sinon

## Variantes du problème SAT

**SAT**      || entrée :  $A \in \mathbb{F}_p(Q)$   
             || sortie : oui si  $A$  est satisfiable, non sinon

**FND-SAT** || entrée :  $A \in \mathbb{F}_p(Q)$  sous FND  
             || sortie : oui si  $A$  est satisfiable, non sinon

# Variantes du problème SAT

**SAT** || entrée :  $A \in \mathbb{F}_p(\mathcal{Q})$   
|| sortie : oui si  $A$  est satisfiable, non sinon

**FND-SAT** || entrée :  $A \in \mathbb{F}_p(\mathcal{Q})$  sous FND  
|| sortie : oui si  $A$  est satisfiable, non sinon

**FNC-SAT** || entrée :  $A \in \mathbb{F}_p(\mathcal{Q})$  sous FNC  
|| sortie : oui si  $A$  est satisfiable, non sinon

## Variantes du problème SAT

**SAT** || entrée :  $A \in \mathbb{F}_p(Q)$   
sortie : oui si  $A$  est satisfiable, non sinon

**FND-SAT** || entrée :  $A \in \mathbb{F}_p(Q)$  sous FND  
sortie : oui si  $A$  est satisfiable, non sinon

**FNC-SAT** || entrée :  $A \in \mathbb{F}_p(Q)$  sous FNC  
sortie : oui si  $A$  est satisfiable, non sinon

**3-SAT** || entrée :  $A \in \mathbb{F}_p(Q)$  sous FNC ac des clauses de 3 littéraux seulement  
sortie : oui si  $A$  est satisfiable, non sinon

# Variantes du problème SAT

**SAT** || entrée :  $A \in \mathbb{F}_p(Q)$   
sortie : oui si  $A$  est satisfiable, non sinon

**FND-SAT** || entrée :  $A \in \mathbb{F}_p(Q)$  sous FND  
sortie : oui si  $A$  est satisfiable, non sinon

**FNC-SAT** || entrée :  $A \in \mathbb{F}_p(Q)$  sous FNC  
sortie : oui si  $A$  est satisfiable, non sinon

**3-SAT** || entrée :  $A \in \mathbb{F}_p(Q)$  sous FNC ac des clauses de 3 littéraux seulement  
sortie : oui si  $A$  est satisfiable, non sinon

**2-SAT** || entrée :  $A \in \mathbb{F}_p(Q)$  sous FNC ac des clauses de 2 littéraux seulement  
sortie : oui si  $A$  est satisfiable, non sinon



## Variantes du problème SAT

**SAT** || entrée :  $A \in \mathbb{F}_p(Q)$   
 || sortie : oui si  $A$  est satisfiable, non sinon

**FND-SAT** || entrée :  $A \in \mathbb{F}_p(Q)$  sous FND  
 || sortie : oui si  $A$  est satisfiable, non sinon

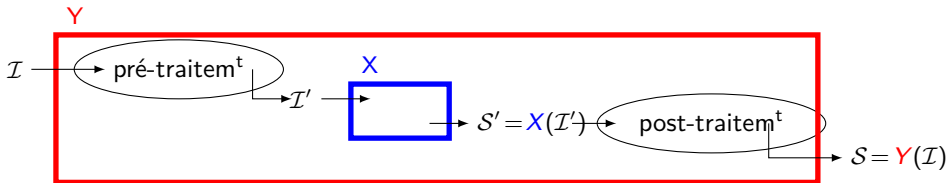
**FNC-SAT** || entrée :  $A \in \mathbb{F}_p(Q)$  sous FNC  
 || sortie : oui si  $A$  est satisfiable, non sinon

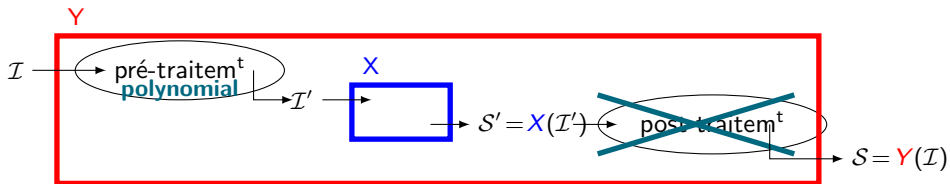
**3-SAT** || entrée :  $A \in \mathbb{F}_p(Q)$  sous FNC ac des clauses de 3 littéraux seulement  
 || sortie : oui si  $A$  est satisfiable, non sinon

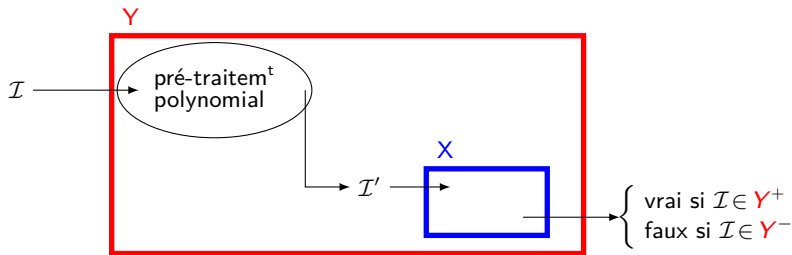
**2-SAT** || entrée :  $A \in \mathbb{F}_p(Q)$  sous FNC ac des clauses de 2 littéraux seulement  
 || sortie : oui si  $A$  est satisfiable, non sinon

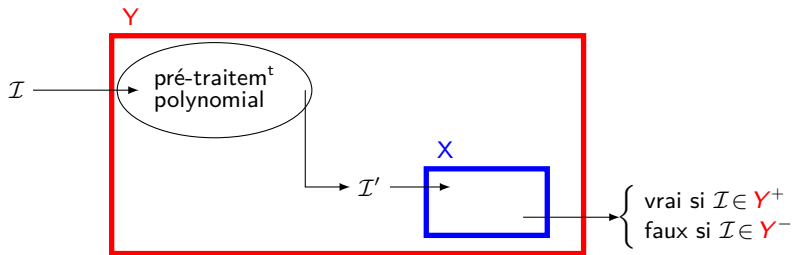
*Quels liens entre ces variantes ?*

# Rappel : réduction entre problèmes



réduction **polynomiale** entre problèmes

Réduction **polynomiale** entre problèmes de décision

Réduction **polynomiale** entre problèmes de décision

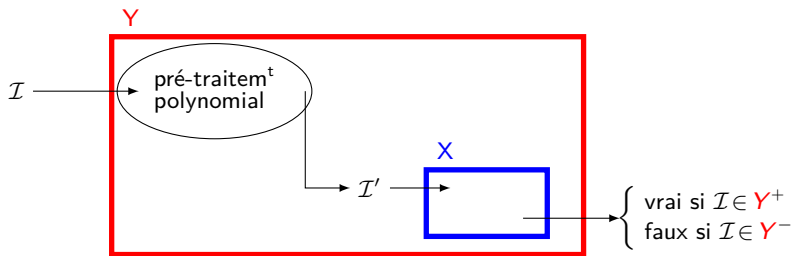
## Définition

Soient  $X$  et  $Y$  deux problèmes de décisions.

On note  $X^+$  (resp.  $X^-$ ) les instances positives (resp. négatives) de  $X$ , i.e. celles pour lesquelles la sol. est vrai (resp. faux). On définit de m<sup>^</sup>  $Y^+$  et  $Y^-$ .

On dit que  $Y$  **se réduit** à  $X$  en temps polynomial

# Réduction **polynomiale** entre problèmes de décision



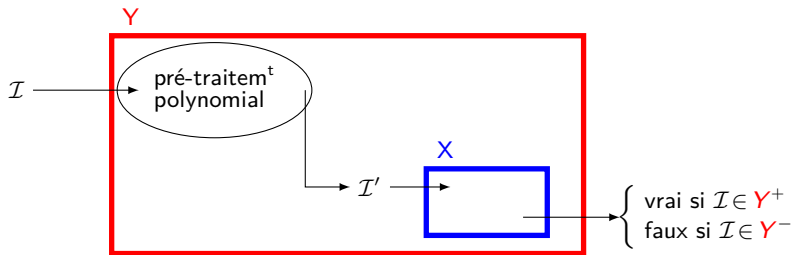
## Définition

Soient  $X$  et  $Y$  deux problèmes de décisions.

On note  $X^+$  (resp.  $X^-$ ) les instances positives (resp. négatives) de  $X$ , i.e. celles pour lesquelles la sol. est vrai (resp. faux). On définit de m<sup>^</sup>  $Y^+$  et  $Y^-$ .

On dit que  $Y$  **se réduit** à  $X$  **en temps polynomial** s'il existe une transformation  $\varphi$ ,

# Réduction **polynomiale** entre problèmes de décision



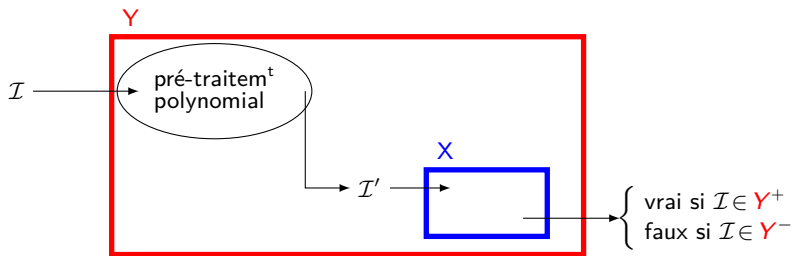
## Définition

Soient  $X$  et  $Y$  deux problèmes de décisions.

On note  $X^+$  (resp.  $X^-$ ) les instances positives (resp. négatives) de  $X$ , i.e. celles pour lesquelles la sol. est vrai (resp. faux). On définit de m<sup>^</sup>  $Y^+$  et  $Y^-$ .

On dit que  $Y$  **se réduit** à  $X$  **en temps polynomial**

s'il existe une transformation  $\varphi$ , calculable en temps polynomial,

Réduction **polynomiale** entre problèmes de décision

## Définition

Soient  $X$  et  $Y$  deux problèmes de décisions.

On note  $X^+$  (resp.  $X^-$ ) les instances positives (resp. négatives) de  $X$ , i.e. celles pour lesquelles la sol. est vrai (resp. faux). On définit de même  $Y^+$  et  $Y^-$ .

On dit que  $Y$  **se réduit** à  $X$  **en temps polynomial**

s'il existe une transformation  $\varphi$ , calculable en temps polynomial, qui transforme toute instance de  $Y$  en une instance de  $X$  de sorte que

$$\forall \mathcal{I}, \varphi(\mathcal{I}) \in X^+ \Leftrightarrow \mathcal{I} \in Y^+$$



## Réductions entre les variantes de SAT - 1/2

- **SAT** est plus dur que **FND-SAT** et **FNC-SAT**

## Réductions entre les variantes de SAT - 1/2

- **SAT** est plus dur que **FND-SAT** et **FNC-SAT**  
↳ **FNC-SAT** se réduit à **SAT**

## Réductions entre les variantes de SAT - 1/2

- **SAT** est plus dur que **FND-SAT** et **FNC-SAT**
  - ↪ **FNC-SAT** se réduit à **SAT** (pour  $\varphi = id$ )
  - ↪ **FND-SAT** se réduit à **SAT** (pour  $\varphi = id$ )

## Réductions entre les variantes de SAT - 1/2

- **SAT** est plus dur que **FND-SAT** et **FNC-SAT**
  - ↪ **FNC-SAT** se réduit à **SAT** (pour  $\varphi = id$ )
  - ↪ **FND-SAT** se réduit à **SAT** (pour  $\varphi = id$ )
- **FNC-SAT** est plus dur que **2-SAT** et **3-SAT**
  - ↪ **2-SAT** se réduit à **FNC-SAT** (pour  $\varphi = id$ )
  - ↪ **3-SAT** se réduit à **FNC-SAT** (pour  $\varphi = id$ )

# Réductions entre les variantes de SAT - 1/2

- **SAT** est plus dur que **FND-SAT** et **FNC-SAT**
  - ↪ **FNC-SAT** se réduit à **SAT** (pour  $\varphi = id$ )
  - ↪ **FND-SAT** se réduit à **SAT** (pour  $\varphi = id$ )
- **FNC-SAT** est plus dur que **2-SAT** et **3-SAT**
  - ↪ **2-SAT** se réduit à **FNC-SAT** (pour  $\varphi = id$ )
  - ↪ **3-SAT** se réduit à **FNC-SAT** (pour  $\varphi = id$ )
- **FNC-SAT** se réduit à **3-SAT**
  - ↪ on transforme chaque clause de taille  $< 3$  en clause de taille 3

$$a \equiv (a \vee a \vee a) \quad (a \vee b) \equiv (a \vee a \vee b)$$

# Réductions entre les variantes de SAT - 1/2

- **SAT** est plus dur que **FND-SAT** et **FNC-SAT**

↪ **FNC-SAT** se réduit à **SAT** (pour  $\varphi = id$ )

↪ **FND-SAT** se réduit à **SAT** (pour  $\varphi = id$ )

- **FNC-SAT** est plus dur que **2-SAT** et **3-SAT**

↪ **2-SAT** se réduit à **FNC-SAT** (pour  $\varphi = id$ )

↪ **3-SAT** se réduit à **FNC-SAT** (pour  $\varphi = id$ )

- **FNC-SAT** se réduit à **3-SAT**

↪ on transforme chaque clause de taille  $< 3$  en clause de taille 3

$$a \equiv (a \vee a \vee a) \quad (a \vee b) \equiv (a \vee a \vee b)$$

↪ on transforme chaque clause de taille  $> 3$  en conjonction de clauses de taille 3 en ajoutant des nouvelles variables

$$(a \vee b \vee c \vee d) \rightsquigarrow$$

# Réductions entre les variantes de SAT - 1/2

- **SAT** est plus dur que **FND-SAT** et **FNC-SAT**

↪ **FNC-SAT** se réduit à **SAT** (pour  $\varphi = id$ )

↪ **FND-SAT** se réduit à **SAT** (pour  $\varphi = id$ )

- **FNC-SAT** est plus dur que **2-SAT** et **3-SAT**

↪ **2-SAT** se réduit à **FNC-SAT** (pour  $\varphi = id$ )

↪ **3-SAT** se réduit à **FNC-SAT** (pour  $\varphi = id$ )

- **FNC-SAT** se réduit à **3-SAT**

↪ on transforme chaque clause de taille  $< 3$  en clause de taille 3

$$a \equiv (a \vee a \vee a) \quad (a \vee b) \equiv (a \vee a \vee b)$$

↪ on transforme chaque clause de taille  $> 3$  en conjonction de clauses de taille 3 en ajoutant des nouvelles variables

$$(a \vee b \vee c \vee d) \rightsquigarrow (a \vee b \vee z) \wedge (\neg z \vee c \vee d)$$

# Réductions entre les variantes de SAT - 1/2

- **SAT** est plus dur que **FND-SAT** et **FNC-SAT**

↪ **FNC-SAT** se réduit à **SAT** (pour  $\varphi = id$ )

↪ **FND-SAT** se réduit à **SAT** (pour  $\varphi = id$ )

- **FNC-SAT** est plus dur que **2-SAT** et **3-SAT**

↪ **2-SAT** se réduit à **FNC-SAT** (pour  $\varphi = id$ )

↪ **3-SAT** se réduit à **FNC-SAT** (pour  $\varphi = id$ )

- **FNC-SAT** se réduit à **3-SAT**

↪ on transforme chaque clause de taille  $< 3$  en clause de taille 3

$$a \equiv (a \vee a \vee a) \quad (a \vee b) \equiv (a \vee a \vee b)$$

↪ on transforme chaque clause de taille  $> 3$  en conjonction de clauses de taille 3 en ajoutant des nouvelles variables

$$(a \vee b \vee c \vee d) \rightsquigarrow (a \vee b \vee z) \wedge (\neg z \vee c \vee d)$$

$$(a \vee b \vee c \vee d \vee e) \rightsquigarrow$$



# Réductions entre les variantes de SAT - 1/2

- **SAT** est plus dur que **FND-SAT** et **FNC-SAT**

↪ **FNC-SAT** se réduit à **SAT** (pour  $\varphi = id$ )

↪ **FND-SAT** se réduit à **SAT** (pour  $\varphi = id$ )

- **FNC-SAT** est plus dur que **2-SAT** et **3-SAT**

↪ **2-SAT** se réduit à **FNC-SAT** (pour  $\varphi = id$ )

↪ **3-SAT** se réduit à **FNC-SAT** (pour  $\varphi = id$ )

- **FNC-SAT** se réduit à **3-SAT**

↪ on transforme chaque clause de taille  $< 3$  en clause de taille 3

$$a \equiv (a \vee a \vee a) \quad (a \vee b) \equiv (a \vee a \vee b)$$

↪ on transforme chaque clause de taille  $> 3$  en conjonction de clauses de taille 3 en ajoutant des nouvelles variables

$$(a \vee b \vee c \vee d) \rightsquigarrow (a \vee b \vee z) \wedge (\neg z \vee c \vee d)$$

$$(a \vee b \vee c \vee d \vee e) \rightsquigarrow (a \vee b \vee z_1) \wedge (\neg z_1 \vee c \vee z_2) \wedge (\neg z_2 \vee d \vee e)$$

# Réductions entre les variantes de SAT - 1/2

- **SAT** est plus dur que **FND-SAT** et **FNC-SAT**

↪ **FNC-SAT** se réduit à **SAT** (pour  $\varphi = id$ )

↪ **FND-SAT** se réduit à **SAT** (pour  $\varphi = id$ )

- **FNC-SAT** est plus dur que **2-SAT** et **3-SAT**

↪ **2-SAT** se réduit à **FNC-SAT** (pour  $\varphi = id$ )

↪ **3-SAT** se réduit à **FNC-SAT** (pour  $\varphi = id$ )

- **FNC-SAT** se réduit à **3-SAT**

↪ on transforme chaque clause de taille  $< 3$  en clause de taille 3

$$a \equiv (a \vee a \vee a) \quad (a \vee b) \equiv (a \vee a \vee b)$$

↪ on transforme chaque clause de taille  $> 3$  en conjonction de clauses de taille 3 en ajoutant des nouvelles variables

$$(a \vee b \vee c \vee d) \rightsquigarrow (a \vee b \vee z) \wedge (\neg z \vee c \vee d)$$

$$(a \vee b \vee c \vee d \vee e) \rightsquigarrow (a \vee b \vee z_1) \wedge (\neg z_1 \vee c \vee z_2) \wedge (\neg z_2 \vee d \vee e)$$

↪ transformations en temps poly. qui préservent la satisfiabilité

## Exercice - pour la réduction de **FNC-SAT** à **3-SAT**

Soit  $C = \ell_1 \vee \ell_2 \vee \ell_3 \vee \dots \vee \ell_k$  une clause de  $\mathbb{F}_p(Q)$  de taille  $k > 3$ .

Soit  $z$  une nouvelle variable (i.e.  $z \notin Q$ ).

On pose alors  $C_1 = (\ell_1 \vee \ell_2 \vee z)$  et  $C_2 = (\neg z \vee \ell_3 \vee \dots \vee \ell_k)$ .

## Exercice - pour la réduction de **FNC-SAT** à **3-SAT**

Soit  $C = \ell_1 \vee \ell_2 \vee \ell_3 \vee \dots \vee \ell_k$  une clause de  $\mathbb{F}_p(Q)$  de taille  $k > 3$ .

Soit  $z$  une nouvelle variable (i.e.  $z \notin Q$ ).

On pose alors  $C_1 = (\ell_1 \vee \ell_2 \vee z)$  et  $C_2 = (\neg z \vee \ell_3 \vee \dots \vee \ell_k)$ .

Montrer les deux propositions suivantes :

$$\rightarrow \forall \rho \in \mathbb{B}^Q, [C]^\rho = V \Rightarrow \left( \exists \tilde{\rho} \in \mathbb{B}^{Q \cup \{z\}}, \tilde{\rho}|_Q = \rho \text{ et } [C_1 \wedge C_2]^{\tilde{\rho}} = V \right)$$

$$\rightarrow \forall \tilde{\rho} \in \mathbb{B}^{Q \cup \{z\}}, [C_1 \wedge C_2]^{\tilde{\rho}} = V \Rightarrow [C]^{\tilde{\rho}} = V$$

## Exercice - pour la réduction de **FNC-SAT** à **3-SAT**

Soit  $C = \ell_1 \vee \ell_2 \vee \ell_3 \vee \dots \vee \ell_k$  une clause de  $\mathbb{F}_p(Q)$  de taille  $k > 3$ .

Soit  $z$  une nouvelle variable (i.e.  $z \notin Q$ ).

On pose alors  $C_1 = (\ell_1 \vee \ell_2 \vee z)$  et  $C_2 = (\neg z \vee \ell_3 \vee \dots \vee \ell_k)$ .

Montrer les deux propositions suivantes :

$$\rightarrow \forall \rho \in \mathbb{B}^Q, [C]^\rho = V \Rightarrow \left( \exists \tilde{\rho} \in \mathbb{B}^{Q \cup \{z\}}, \tilde{\rho}|_Q = \rho \text{ et } [C_1 \wedge C_2]^{\tilde{\rho}} = V \right)$$

$$\rightarrow \forall \tilde{\rho} \in \mathbb{B}^{Q \cup \{z\}}, [C_1 \wedge C_2]^{\tilde{\rho}} = V \Rightarrow [C]^{\tilde{\rho}} = V$$

*La même stratégie permet-elle de réduire **FNC-SAT** à **2-SAT** ?*

## Réductions entre les variantes de SAT - 2/2

- **FNC-SAT** ne se réduit pas à **2-SAT**

## Réductions entre les variantes de SAT - 2/2

- **FNC-SAT** ne se réduit pas à **2-SAT**  
↳ on verra que **2-SAT** se résout en temps poly. (cf chap. graphes),

## Réductions entre les variantes de SAT - 2/2

- **FNC-SAT** ne se réduit pas à **2-SAT**
  - ↪ on verra que **2-SAT** se résout en temps poly. (cf chap. graphes),
  - ↪ vous verrez que **3-SAT** est NP-complet (th. de Cook, 2eme année)



## Réductions entre les variantes de SAT - 2/2

- **FNC-SAT** ne se réduit pas à **2-SAT**
  - ↪ on verra que **2-SAT** se résout en temps poly. (cf chap. graphes),
  - ↪ vous verrez que **3-SAT** est NP-complet (th. de Cook, 2eme année)
  - ↪ à moins que  $P=NP$ , ces deux problèmes ne sont pas équivalents

## Réductions entre les variantes de SAT - 2/2

- **FNC-SAT** ne se réduit pas à **2-SAT**
  - ↪ on verra que **2-SAT** se résout en temps poly. (cf chap. graphes),
  - ↪ vous verrez que **3-SAT** est NP-complet (th. de Cook, 2eme année)
  - ↪ à moins que  $P=NP$ , ces deux problèmes ne sont pas équivalents
- **FNC-SAT** ne se réduit pas à **FND-SAT**

## Réductions entre les variantes de SAT - 2/2

- **FNC-SAT** ne se réduit pas à **2-SAT**
  - ↪ on verra que **2-SAT** se résout en temps poly. (cf chap. graphes),
  - ↪ vous verrez que **3-SAT** est NP-complet (th. de Cook, 2eme année)
  - ↪ à moins que  $P=NP$ , ces deux problèmes ne sont pas équivalents
- **FNC-SAT** ne se réduit pas à **FND-SAT**
  - ↪ la transfo. d'une forme à l'autre se fait en temps exponentiel et peut aussi augmenter la taille de l'entrée de manière exponentielle

## Réductions entre les variantes de SAT - 2/2

- **FNC-SAT** ne se réduit pas à **2-SAT**
  - ↪ on verra que **2-SAT** se résout en temps poly. (cf chap. graphes),
  - ↪ vous verrez que **3-SAT** est NP-complet (th. de Cook, 2eme année)
  - ↪ à moins que  $P=NP$ , ces deux problèmes ne sont pas équivalents
- **FNC-SAT** ne se réduit pas à **FND-SAT**
  - ↪ la transfo. d'une forme à l'autre se fait en temps exponentiel et peut aussi augmenter la taille de l'entrée de manière exponentielle
  - ↪ on verra que **FND-SAT** se résout en temps polynomial (juste après),

## Réductions entre les variantes de SAT - 2/2

- **FNC-SAT** ne se réduit pas à **2-SAT**
  - ↪ on verra que **2-SAT** se résout en temps poly. (cf chap. graphes),
  - ↪ vous verrez que **3-SAT** est NP-complet (th. de Cook, 2eme année)
  - ↪ à moins que  $P=NP$ , ces deux problèmes ne sont pas équivalents
- **FNC-SAT** ne se réduit pas à **FND-SAT**
  - ↪ la transfo. d'une forme à l'autre se fait en temps exponentiel et peut aussi augmenter la taille de l'entrée de manière exponentielle
  - ↪ on verra que **FND-SAT** se résout en temps polynomial (juste après),
  - ↪ à moins que  $P=NP$ , ces deux problèmes ne sont pas équivalents

## Réductions entre les variantes de SAT - 2/2

- **FNC-SAT** ne se réduit pas à **2-SAT**
  - ↪ on verra que **2-SAT** se résout en temps poly. (cf chap. graphes),
  - ↪ vous verrez que **3-SAT** est NP-complet (th. de Cook, 2eme année)
  - ↪ à moins que  $P=NP$ , ces deux problèmes ne sont pas équivalents
- **FNC-SAT** ne se réduit pas à **FND-SAT**
  - ↪ la transfo. d'une forme à l'autre se fait en temps exponentiel et peut aussi augmenter la taille de l'entrée de manière exponentielle
  - ↪ on verra que **FND-SAT** se résout en temps polynomial (juste après),
  - ↪ à moins que  $P=NP$ , ces deux problèmes ne sont pas équivalents

**À retenir** : importance de la forme sous laquelle sont données les entrées d'un problème !

# Modéliser une FND

FND = disjonction de conjonctions de littéraux.

D'un point de vue sémantique (*i.e.* dans  $\mathbb{F}_p(\mathcal{Q})/\equiv$ ),

- l'ordre des littéraux au sein d'une conjonction est-il important ? et leur multiplicité ?
- quel objet mathématique est adapté pour modéliser les conjonctions de littéraux ?
- l'ordre des conjonctions au sein de la disjonction est-il important ? et leur multiplicité ?
- quel objet mathématique est adapté pour modéliser une FND ?
- quelle structure de données est adaptée pour modéliser une FND ?

# Modéliser une FND

FND = disjonction de conjonctions de littéraux.

D'un point de vue sémantique (*i.e.* dans  $\mathbb{F}_p(\mathcal{Q})/\equiv$ ),

- l'ordre des littéraux au sein d'une conjonction est-il important ? et leur multiplicité ? **non**, une conjonction est satisfaite par un env. prop. ssi il satisfait l'ensemble de ses termes.
- quel objet mathématique est adapté pour modéliser les conjonctions de littéraux ?
- l'ordre des conjonctions au sein de la disjonction est-il important ? et leur multiplicité ?
- quel objet mathématique est adapté pour modéliser une FND ?
- quelle structure de données est adaptée pour modéliser une FND ?



# Modéliser une FND

FND = disjonction de conjonctions de littéraux.

D'un point de vue sémantique (*i.e.* dans  $\mathbb{F}_p(\mathcal{Q})/\equiv$ ),

- l'ordre des littéraux au sein d'une conjonction est-il important ? et leur multiplicité ? **non**, une conjonction est satisfaite par un env. prop. ssi il satisfait l'ensemble de ses termes.
- quel objet mathématique est adapté pour modéliser les conjonctions de littéraux ? un **ensemble de littéraux** ou un **couple d'ensembles de variables** : d'une part celles apparaissant dans les littéraux positifs, d'autre part celles apparaissant dans les littéraux négatifs
- l'ordre des conjonctions au sein de la disjonction est-il important ? et leur multiplicité ?
- quel objet mathématique est adapté pour modéliser une FND ?
- quelle structure de données est adaptée pour modéliser une FND ?

# Modéliser une FND

FND = disjonction de conjonctions de littéraux.

D'un point de vue sémantique (*i.e.* dans  $\mathbb{F}_p(\mathcal{Q})/\equiv$ ),

- l'ordre des littéraux au sein d'une conjonction est-il important ? et leur multiplicité ? **non**, une conjonction est satisfaite par un env. prop. ssi il satisfait l'ensemble de ses termes.
- quel objet mathématique est adapté pour modéliser les conjonctions de littéraux ? un **ensemble de littéraux** ou un **couple d'ensembles de variables** : d'une part celles apparaissant dans les littéraux positifs, d'autre part celles apparaissant dans les littéraux négatifs
- l'ordre des conjonctions au sein de la disjonction est-il important ? et leur multiplicité ? **non**, une disjonction est satisfaite par un env. prop. ssi il satisfait l'un de ses termes.
- quel objet mathématique est adapté pour modéliser une FND ?
- quelle structure de données est adaptée pour modéliser une FND ?

# Modéliser une FND

FND = disjonction de conjonctions de littéraux.

D'un point de vue sémantique (*i.e.* dans  $\mathbb{F}_p(\mathcal{Q})/\equiv$ ),

- l'ordre des littéraux au sein d'une conjonction est-il important ? et leur multiplicité ? **non**, une conjonction est satisfaite par un env. prop. ssi il satisfait l'ensemble de ses termes.
- quel objet mathématique est adapté pour modéliser les conjonctions de littéraux ? un **ensemble de littéraux** ou un **couple d'ensembles de variables** : d'une part celles apparaissant dans les littéraux positifs, d'autre part celles apparaissant dans les littéraux négatifs
- l'ordre des conjonctions au sein de la disjonction est-il important ? et leur multiplicité ? **non**, une disjonction est satisfaite par un env. prop. ssi il satisfait l'un de ses termes.
- quel objet mathématique est adapté pour modéliser une FND ? un ensemble de conjonction
- quelle structure de données est adaptée pour modéliser une FND ?

# Modéliser une FND

FND = disjonction de conjonctions de littéraux.

D'un point de vue sémantique (*i.e.* dans  $\mathbb{F}_p(\mathcal{Q})/\equiv$ ),

- l'ordre des littéraux au sein d'une conjonction est-il important ? et leur multiplicité ? **non**, une conjonction est satisfaite par un env. prop. ssi il satisfait l'ensemble de ses termes.
- quel objet mathématique est adapté pour modéliser les conjonctions de littéraux ? un **ensemble de littéraux** ou un **couple d'ensembles de variables** : d'une part celles apparaissant dans les littéraux positifs, d'autre part celles apparaissant dans les littéraux négatifs
- l'ordre des conjonctions au sein de la disjonction est-il important ? et leur multiplicité ? **non**, une disjonction est satisfaite par un env. prop. ssi il satisfait l'un de ses termes.
- quel objet mathématique est adapté pour modéliser une FND ? un ensemble de conjonction
- quelle structure de données est adaptée pour modéliser une FND ? une liste de liste de littéraux convient

# FND-SAT

On suppose que  $\mathcal{Q} = \{q_1, q_2, \dots, q_N\}$ .

**FND-SAT** || entrée :  $((\ell_{i,j})_{i \in [1..n_j]})_{j \in [1..n]}$  une famille de littéraux de  $\mathcal{Q}$   
 || sortie : oui si  $A = \bigvee_{j=1}^n \bigwedge_{i=1}^{n_j} \ell_{i,j}$  est satisfiable,  
 || non sinon

# FND-SAT (FNC-SAT)

On suppose que  $\mathcal{Q} = \{q_1, q_2, \dots, q_N\}$ .

**FND-SAT** || entrée :  $((\ell_{i,j})_{i \in [1..n_j]})_{j \in [1..n]}$  une famille de littéraux de  $\mathcal{Q}$   
 || sortie : oui si  $A = \bigvee_{j=1}^n \bigwedge_{i=1}^{n_j} \ell_{i,j}$  est satisfiable,  
 || non sinon

**FNC-SAT** || entrée :  $((\ell_{i,j})_{i \in [1..n_j]})_{j \in [1..n]}$  une famille de littéraux de  $\mathcal{Q}$   
 || sortie : oui si  $A = \bigwedge_{j=1}^n \bigvee_{i=1}^{n_j} \ell_{i,j}$  est satisfiable,  
 || non sinon

## FND-SAT - exemple 1/2

la formule initiale :

$$(a \wedge b \wedge \neg c) \vee (\neg b \wedge \neg c) \vee (a \wedge c)$$

sous forme d'ensemble d'ensembles de littéraux :

sous forme d'ensemble de couples :

la formule initiale :

$$(a \wedge b \wedge \neg c) \vee (a \wedge c \wedge a) \vee (\neg b \wedge \neg c)$$

sous forme d'ensemble d'ensembles de littéraux :

sous forme d'ensemble de couples :

## FND-SAT - exemple 1/2

la formule initiale :

$$(a \wedge b \wedge \neg c) \vee (\neg b \wedge \neg c) \vee (a \wedge c)$$

sous forme d'ensemble d'ensembles de littéraux :

$$\{\{a, b, \neg c\}, \{\neg b, \neg c\}, \{a, c\}\}$$

sous forme d'ensemble de couples :

la formule initiale :

$$(a \wedge b \wedge \neg c) \vee (a \wedge c \wedge a) \vee (\neg b \wedge \neg c)$$

sous forme d'ensemble d'ensembles de littéraux :

sous forme d'ensemble de couples :



## FND-SAT - exemple 1/2

la formule initiale :

$$(a \wedge b \wedge \neg c) \vee (\neg b \wedge \neg c) \vee (a \wedge c)$$

sous forme d'ensemble d'ensembles de littéraux :

$$\{\{a, b, \neg c\}, \{\neg b, \neg c\}, \{a, c\}\}$$

sous forme d'ensemble de couples :

$$\{(\{a, b\}, \{c\}), (\emptyset, \{b, c\}), (\{a, c\}, \emptyset)\}$$

la formule initiale :

$$(a \wedge b \wedge \neg c) \vee (a \wedge c \wedge a) \vee (\neg b \wedge \neg c)$$

sous forme d'ensemble d'ensembles de littéraux :

sous forme d'ensemble de couples :

## FND-SAT - exemple 1/2

la formule initiale :

$$(a \wedge b \wedge \neg c) \vee (\neg b \wedge \neg c) \vee (a \wedge c)$$

sous forme d'ensemble d'ensembles de littéraux :

sous forme d'ensemble de couples :

$$\{(\{a, b\}, \{c\}), (\emptyset, \{b, c\}), (\{a, c\}, \emptyset)\}$$

la formule initiale :

$$(a \wedge b \wedge \neg c) \vee (a \wedge c \wedge a) \vee (\neg b \wedge \neg c)$$

sous forme d'ensemble d'ensembles de littéraux :

$$\{\{a, b, \neg c\}, \{\neg b, \neg c\}, \{a, c\}\}$$

sous forme d'ensemble de couples :

## FND-SAT - exemple 1/2

la formule initiale :

$$(a \wedge b \wedge \neg c) \vee (\neg b \wedge \neg c) \vee (a \wedge c)$$

sous forme d'ensemble d'ensembles de littéraux :

sous forme d'ensemble de couples :

$$\{(\{a, b\}, \{c\}), (\emptyset, \{b, c\}), (\{a, c\}, \emptyset)\}$$

la formule initiale :

$$(a \wedge b \wedge \neg c) \vee (a \wedge c \wedge a) \vee (\neg b \wedge \neg c)$$

sous forme d'ensemble d'ensembles de littéraux :

$$\{\{a, b, \neg c\}, \{\neg b, \neg c\}, \{a, c\}\}$$

sous forme d'ensemble de couples :

$$\{(\{a, b\}, \{c\}), (\emptyset, \{b, c\}), (\{a, c\}, \emptyset)\}$$

## FND-SAT - exemple 2/2

la formule initiale :

$$(a \wedge b \wedge \neg a) \vee (a \wedge c \wedge \neg a) \vee (\neg b \wedge \neg c)$$

sous forme d'ensemble d'ensembles de littéraux :

sous forme d'ensemble de couples :

## FND-SAT - exemple 2/2

la formule initiale :

$$(a \wedge b \wedge \neg a) \vee (a \wedge c \wedge \neg a) \vee (\neg b \wedge \neg c)$$

sous forme d'ensemble d'ensembles de littéraux :

$$\{\{a, b, \neg a\}, \{a, \neg a, \neg c\}, \{\neg b, \neg c\}\}$$

sous forme d'ensemble de couples :

## FND-SAT - exemple 2/2

la formule initiale :

$$(a \wedge b \wedge \neg a) \vee (a \wedge c \wedge \neg a) \vee (\neg b \wedge \neg c)$$

sous forme d'ensemble d'ensembles de littéraux :

$$\{\{a, b, \neg a\}, \{a, \neg a, \neg c\}, \{\neg b, \neg c\}\}$$

sous forme d'ensemble de couples :

$$\{(\{a, b\}, \{a\}), (\{a\}, \{a, c\}), (\emptyset, \{b, c\}), (\emptyset, \emptyset)\}$$

## FND-SAT - exemple 2/2

la formule initiale :

$$(a \wedge b \wedge \neg a) \vee (a \wedge c \wedge \neg a) \vee (\neg b \wedge \neg c)$$

sous forme d'ensemble d'ensembles de littéraux :

$$\{\{a, b, \neg a\}, \{a, \neg a, \neg c\}, \{\neg b, \neg c\}\}$$

sous forme d'ensemble de couples :

$$\{(\{a, b\}, \{a\}), (\{a\}, \{a, c\}), (\emptyset, \{b, c\}), (\emptyset, \emptyset)\}$$

*Cette formule est-elle satisfiable ? Justifier.*

*Par quelle procédure décider ?*

# Algorithme pour FND-SAT

---

Algorithme FND-SAT sur  $\mathcal{Q} = \{q_1, q_2, \dots, q_N\}$

---

entrée :  $((\ell_{i,j})_{i \in [1..n_j]})_{j \in [1..n]}$  une famille de littéraux de  $\mathcal{Q}$

sortie : vrai si  $A = \bigvee_{j=1}^n \bigwedge_{i=1}^{n_j} \ell_{i,j}$ , faux sinon

---

**Pour**  $j$  allant de 1 à  $n$  :

Créer  $T$  un tableau indicé par  $[1..N]$  initialisé à  $-1$

Essayer :

**Pour**  $i$  allant de 1 à  $n_j$  :

**Si**  $\ell_{i,j} = q_k$

**alors** si  $T[k] = -1$ , alors  $T[k] \leftarrow 1$

sinon si  $T[k] = 0$  alors déclencher "conj. non sat."

**Si**  $\ell_{i,j} = \neg q_k$

**alors** si  $T[k] = -1$ , alors  $T[k] \leftarrow 0$

sinon si  $T[k] = 1$  alors déclencher l'exception "conj. non sat."

Retourner **vrai**

Rattraper "conj. non sat."

Retourner **faux**

---



## Modélisation - exemple du solitaire 1/3

**État initial** du jeu : Sur chaque case d'un damier carré de  $m \times m$  cases, il y a une pierre bleue (**b**), ou bien une pierre rouge (**r**) ou rien.

**But du jeu** : enlever des pierres de manière à ce que :

- sur chaque colonne toutes les pierres sont de la même couleur
- sur chaque ligne il y a au moins une pierre

**Exercice** :

1. Formaliser le problème de décision consistant à savoir si une partie peut être gagnée.
2. Réduire ce problème à **3-SAT**
3. Réduire **3-SAT** à ce problème (bonus)

**À retenir** : puissance d'encodage de **3-SAT**

## Modélisation - exemple du solitaire 2/3

1. **JEU** || entrée :

## Modélisation - exemple du solitaire 2/3

1. **JEU** || entrée :  $m \in \mathbb{N}^*$  la taille du damier

## Modélisation - exemple du solitaire 2/3

1. **JEU** || entrée :  $m \in \mathbb{N}^*$  la taille du damier  
||  $B \subseteq [1..m]^2$  l'ensemble des cases init. bleues.

# Modélisation - exemple du solitaire 2/3

1. **JEU** || entrée :  $m \in \mathbb{N}^*$  la taille du damier
- $B \subseteq [1..m]^2$  l'ensemble des cases init. bleues.
- $R \subseteq [1..m]^2 \setminus B$  l'ensemble des cases init. rouges.

## Modélisation - exemple du solitaire 2/3

### 1. JEU

entrée :  $m \in \mathbb{N}^*$  la taille du damier

$B \subseteq [1..m]^2$  l'ensemble des cases init. bleues.

$R \subseteq [1..m]^2 \setminus B$  l'ensemble des cases init. rouges.

sortie : oui s'il existe  $B' \subseteq B$  et  $R' \subseteq R$  tel que :

# Modélisation - exemple du solitaire 2/3

## 1. JEU

entrée :  $m \in \mathbb{N}^*$  la taille du damier

$B \subseteq [1..m]^2$  l'ensemble des cases init. bleues.

$R \subseteq [1..m]^2 \setminus B$  l'ensemble des cases init. rouges.

sortie : oui s'il existe  $B' \subseteq B$  et  $R' \subseteq R$  tel que :

$\rightarrow \forall j \in [1..m], B' \cap [1..n] \times \{j\} = \emptyset$  ou  $R' \cap [1..m] \times \{j\} = \emptyset$

$\rightarrow \forall i \in [1..m], (R' \cup B') \cap \{i\} \times [1..m] \neq \emptyset$

## Modélisation - exemple du solitaire 2/3

1. **JEU** | entrée :  $m \in \mathbb{N}^*$  la taille du damier  
           |  $B \subseteq [1..m]^2$  l'ensemble des cases init. bleues.  
           |  $R \subseteq [1..m]^2 \setminus B$  l'ensemble des cases init. rouges.  
           | sortie : oui s'il existe  $B' \subseteq B$  et  $R' \subseteq R$  tel que :  
           |  $\rightarrow \forall j \in [1..m], B' \cap [1..n] \times \{j\} = \emptyset$  ou  $R' \cap [1..m] \times \{j\} = \emptyset$   
           |  $\rightarrow \forall i \in [1..m], (R' \cup B') \cap \{i\} \times [1..m] \neq \emptyset$

2. Soit  $(m, B, R)$  une instance de **JEU**.

On cherche à construire, en temps polynomial, une FNC  $A$  telle que  $A \in \mathbf{3-SAT}^+$  ssi  $(m, B, R) \in \mathbf{JEU}^+$ , c'est-à-dire telle que  $A$  est satisfiable ssi il existe  $B' \subseteq B$  et  $R' \subseteq R$  tel que ...



## Modélisation - exemple du solitaire 2/3

1. **JEU** | entrée :  $m \in \mathbb{N}^*$  la taille du damier  
           |  $B \subseteq [1..m]^2$  l'ensemble des cases init. bleues.  
           |  $R \subseteq [1..m]^2 \setminus B$  l'ensemble des cases init. rouges.  
           | sortie : oui s'il existe  $B' \subseteq B$  et  $R' \subseteq R$  tel que :  
           |  $\rightarrow \forall j \in [1..m], B' \cap [1..m] \times \{j\} = \emptyset$  ou  $R' \cap [1..m] \times \{j\} = \emptyset$   
           |  $\rightarrow \forall i \in [1..m], (R' \cup B') \cap \{i\} \times [1..m] \neq \emptyset$

2. Soit  $(m, B, R)$  une instance de **JEU**.

On cherche à construire, en temps polynomial, une FNC  $A$  telle que  $A \in \mathbf{3-SAT}^+$  ssi  $(m, B, R) \in \mathbf{JEU}^+$ , c'est-à-dire telle que  $A$  est satisfiable ssi il existe  $B' \subseteq B$  et  $R' \subseteq R$  tel que ...

Pour chaque  $(i, j) \in [1..m]^2$ , on introduit 2 variables :

- $b'_{i,j}$  modélisant  $(i, j) \in B'$
- $r'_{i,j}$  modélisant  $(i, j) \in R'$

## Modélisation - exemple du solitaire 3/3

On  $A$  doit modéliser les 4 contraintes ci-dessous :

$$\rightarrow B' \subseteq B$$

$$\rightarrow R' \subseteq R$$

$$\rightarrow \forall j \in [1..m], B' \cap [1..n] \times \{j\} = \emptyset \text{ ou } R' \cap [1..m] \times \{j\} = \emptyset$$

$$\rightarrow \forall i \in [1..m], (R' \cup B') \cap \{i\} \times [1..m] \neq \emptyset$$

## Modélisation - exemple du solitaire 3/3

On  $A$  doit modéliser les 4 contraintes ci-dessous :

$$\rightarrow B' \subseteq B$$

$$\rightarrow R' \subseteq R$$

$$\rightarrow \forall j \in [1..m], B' \cap [1..n] \times \{j\} = \emptyset \text{ ou } R' \cap [1..m] \times \{j\} = \emptyset$$

$$\rightarrow \forall i \in [1..m], (R' \cup B') \cap \{i\} \times [1..m] \neq \emptyset$$

On pose alors  $A$  la conjonction des 4 formules ci-dessous :

$$\rightarrow \bigwedge_{(i,j) \in [1..m] \setminus B} \neg b_{i,j}$$

$$\rightarrow \bigwedge_{(i,j) \in [1..m] \setminus R} \neg r_{i,j}$$

$$\rightarrow \bigwedge_{j \in [1..m]} \bigwedge_{(i,i') \in [1..m]^2} \neg b_{i,j} \vee \neg r_{i',j}$$

$$\rightarrow \bigwedge_{i \in [1..m]} \bigvee_{j \in [1..m]} r_{i,j} \vee b_{i,j}$$