# Éligibilité de contraintes pour la résolution de réseaux de contraintes qualitatives temporelles et spatiales

Jean-François Condotta[1] Gérard Ligozat[2]
Mahmoud Saade[1]

[1] *CRIL-CNRS, Université d'Artois, France*
[2] *LIMSI-CNRS, Université de Paris-Sud, France*

RTE 2007

# Plan du propos

# Définition

## Formalismes qualitatifs

- Un formalisme qualitatif d'arité 2 utilise un ensemble fini B de $k$ relations binaires définies sur un domaine D
- Les relations de B sont appelées relations de base ou encore relations atomiques
- Les entités temporelles sont représentées par les éléments de D

## La propriété JEPD

- Les relations de base sont complètes et mutuellement exclusives: chaque couple d'éléments de D appartient à exactement une relation de base de B

## Relations complexes

- A est l'ensemble des sous-ensembles de B: $\mathrm{A} = 2^{\mathrm{B}} = \{B : B \subseteq \mathrm{B}\}$
- Une relation complexe, representée par $\{\mathrm{B}_1, \ldots, \mathrm{B}_p\}$ où $1 \leq p \leq m$, correspond à $\mathrm{B}_1 \cup \ldots \cup \mathrm{B}_p$

# Opérations

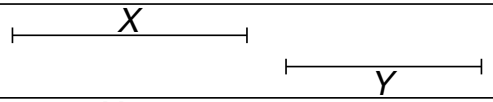
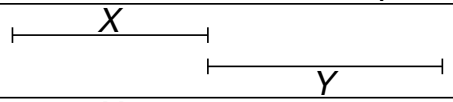
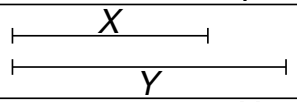
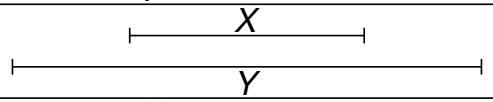
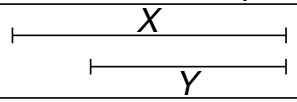
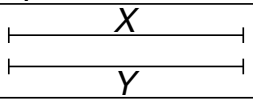
## Inversion

$\forall r \in$ A l'inversion de $r$ est définie comme suit :
$\forall i, j \ (x_i, x_j) \in r^{-1} \ (x_j, x_i) \in r$.

## Composition

La composition est définie comme suit :
soit $r \in$ A : $r \in R \circ S$ ssi $\exists (x, y) \in r$ et $\exists z \in$ D tel que $(x, z) \in R$, $(z, y) \in S$

# Exemple: Algèbre des Intervalles

| *Relation* | *Symbol* | $^{-1}$ | *Meaning* |
|---|---|---|---|
| *precedes* | *b* | *bi* |   |
| *meets* | *m* | *mi* |   |
| *overlaps* | *o* | *oi* |   |
| *starts* | *s* | *si* |   |
| *during* | *d* | *di* |   |
| *finishes* | *f* | *fi* |   |
| *equals* | *eq* | *eq* |   |

# Exemple: Algèbre des Intervalles

| *Relation* | *Symbol* | $^{-1}$ | *Meaning* |
|---|---|---|---|
| *precedes* | *b* | *bi* | X Y |
| *meets* | *m* | *mi* | X Y |
| *overlaps* | *o* | *oi* | X Y |
| *starts* | *s* | *si* | X Y |
| *during* | *d* | *di* | X Y |
| *finishes* | *f* | *fi* | X Y |
| *equals* | *eq* | *eq* | X Y |

## Algèbre des Intervalles

- Les entités sont les intervalles de la droite des réels
- 13 relations de base B $= \{eq, b, bi, m, mi, o, oi, d, di, s, si, f, fi\}$
- Il y a $2^{13}$ relations complexes

# Exemple: Algèbre des Intervalles

| *Relation* | *Symbol* | $^{-1}$ | *Meaning* |
|---|---|---|---|
| *precedes* | *b* | *bi* | X Y |
| *meets* | *m* | *mi* | X Y |
| *overlaps* | *o* | *oi* | X Y |
| *starts* | *s* | *si* | X Y |
| *during* | *d* | *di* | X Y |
| *finishes* | *f* | *fi* | X Y |
| *equals* | *eq* | *eq* | X Y |

## inversion et composition

Soit $R_1 = \{eq, b, si, d\}$, $R_2 = \{m, o\}$,

- $R_1^{-1} = \{eq, bi, s, di\}$ et $R_2^{-1} = \{mi, oi\}$
- $R_1 \circ R_2 = \{b, m, o, d, di, s, fi\}$

# Les réseaux de contraintes qualitatives (Qualitative Constraint Networks - QCN)

**Définition**

Un QCN est un couple $\mathcal{N} = (V, C)$ où:

- $V$ est un ensemble fini de $n$ variables $\{v'_0, \ldots, v'_{p-1}\}$ (où $n > 0$);
- $C$ est une application qui à chaque tuple $(v_i, v_j)$ de $V \times V$ associe un sous-ensemble $C(vi, v_j)$ de l'ensemble des relations de base : $C(v_i, v_j) \subseteq$ B.
- $C(v_i, v_j) =$ B signifie qu'il n'y a pas de contraintes.

# Les réseaux de contraintes qualitatives

## Définition

- Une *instantiation* de $\mathcal{N}$ est une application $\alpha$ de $V$ vers D
- Une *instantiation* est *consistante* ssi $(\alpha(v_0), \ldots, \alpha(v_{n-1})) \in C(v_0, \ldots, v_{n-1})$, pour tout $v_0, \ldots, v_{n-1} \in V$. Une solution est une instantiation consistante.
- Un QCN atomique est un QCN tel que chaque contrainte est définie par une relation de base de B.
- Un *scénario* de $\mathcal{N}$ est un sous-réseau de $\mathcal{N}$ qui est atomique.
- Un *scénario* consistant de $\mathcal{N}$ est un scénario admettant une solution.
- $\mathcal{N}' = (V', C')$ est *équivalent* à $\mathcal{N}$ ssi $V = V'$ et les deux réseaux $\mathcal{N}$ et $\mathcal{N}'$ ont le même ensemble de solutions.

# Les réseaux de contraintes qualitatives (Qualitative Constraint Networks - QCN)

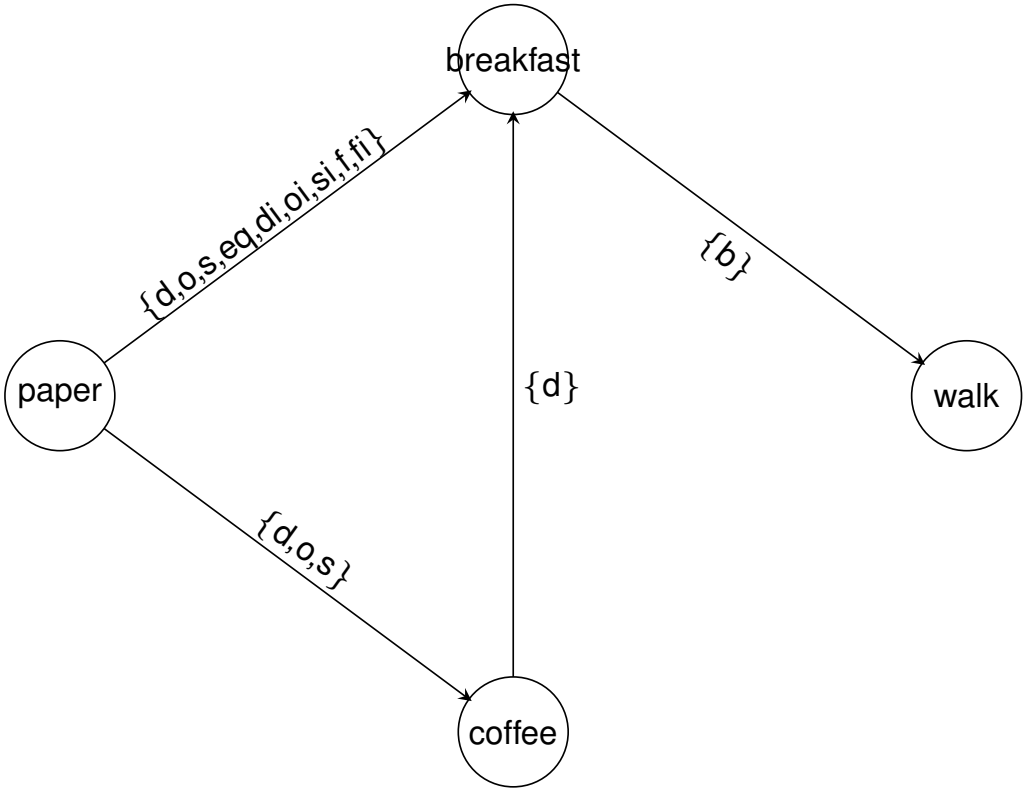
**Exemple**

Fred lit son journal pendant qu'il prend son petit-déjeûner. Il pose son journal et boit sa dernière tasse de café. Après il part faire une promenade.

Le QCN correspondant:

- $V = \{paper, breakfast, coffee, walk\}$
- $C = \{(paper, breakfast, \{eq, d, di, o, oi, s, si, f, fi\}),$
  $(paper, coffee, \{d, o, s\}), (coffee, breakfast, \{d\}), (breakfast, walk, \{b\})\}$

# Les réseaux de contraintes qualitatives



# Les principaux problèmes

## Les principaux problèmes

Étant donné un *QCN* $\mathcal{N}$, les problèmes principaux sont :

- Décider si il existe une solution de $\mathcal{N}$
- trouver une ou toutes les solutions de $\mathcal{N}$
- trouver un ou tous les scénarios de $\mathcal{N}$
- trouver le *QCN* minimal et équivalent à $\mathcal{N}$

# Les principaux problèmes

## Les principaux problèmes

Étant donné un *QCN* $\mathcal{N}$, les problèmes principaux sont :

- Décider si il existe une solution de $\mathcal{N}$
- trouver une ou toutes les solutions de $\mathcal{N}$
- trouver un ou tous les scénarios de $\mathcal{N}$
- trouver le *QCN* minimal et équivalent à $\mathcal{N}$

- Dans la plupart des cas ces problèmes sont des problèmes NP-complets.
- Les algorithmes utilisés pour résoudre ces problèmes sont similaires à ceux des CSP discrets (e.g. méthodes de backtracking).
- Les méthodes de propagation locale de contraintes sont basées sur la méthode de $\circ$-fermeture.

# La méthode de la $\circ$-fermeture

## Définition

La méthode de la $\circ$-fermeture utilise l'opération de composition. Pour chaque tuple $(i, j, k)$, elle remplace la contrainte $C_{ij}$ par:
$C_{ik} \circ C_{kj} \cap C_{ij}$

## Propriétés

- Elle est largement utilisée pour réduire l'espace de recherche.
- Elle est utilisée dans les algorithmes de backtracking.
- La complexité en temps de cette méthode est polynomiale ($O(n^3)$).
- En général, elle ne décide pas de la consistance d'un *QCN*.

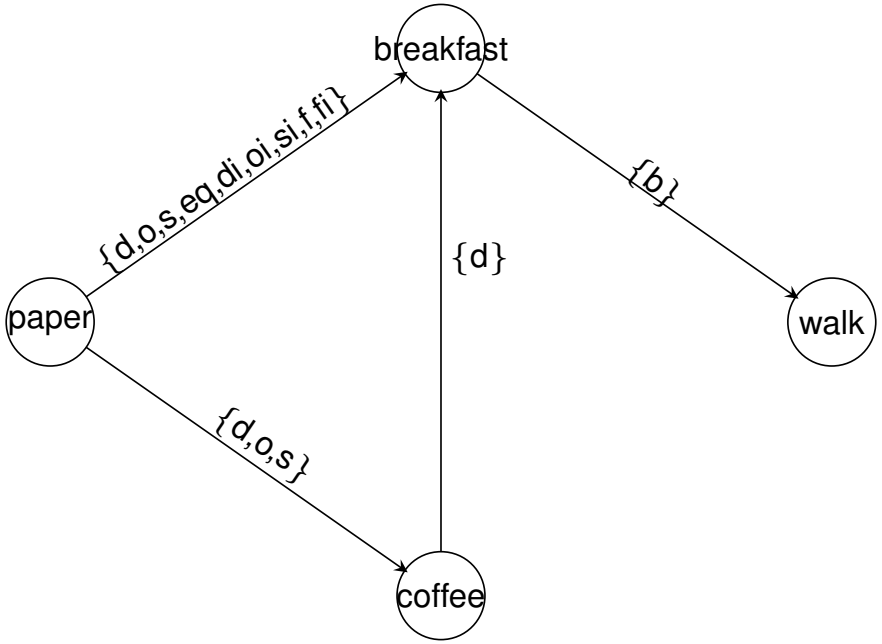
# Algorithme de $\circ$-fermeture

**Algorithm 1**: Function ffc

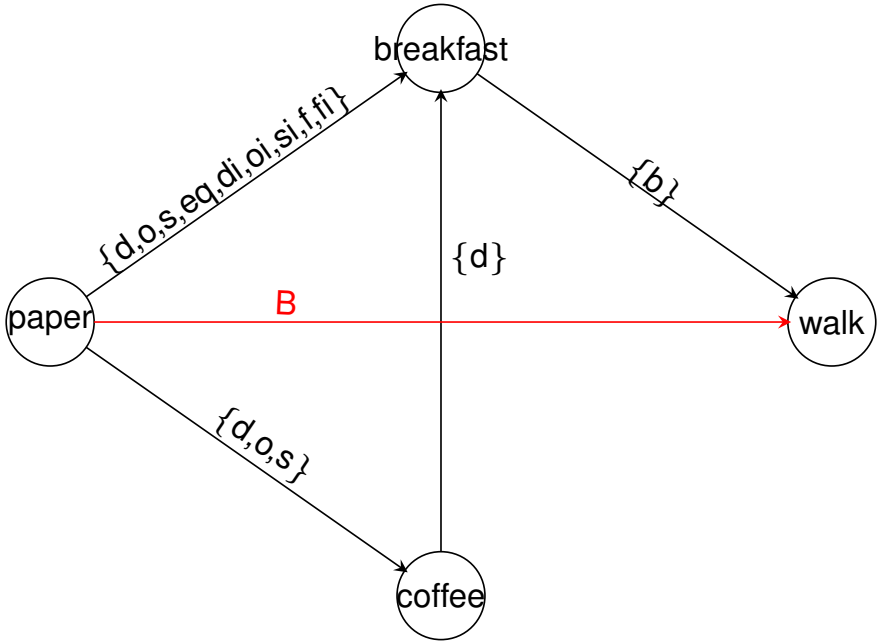
**Data**: $\mathcal{N} = (V, C)$ avec $n = |V|$

```
begin
    repeat
        N' = N
        for i, j, k ∈ 0...n − 1 do
            C_ij ← C_ij ∩ (C_ik ∘ C_kj)
            C_ji ← C_ij^{−1}
            if (C_ij == ∅) then return false
    until N' == N
    return true
end
```

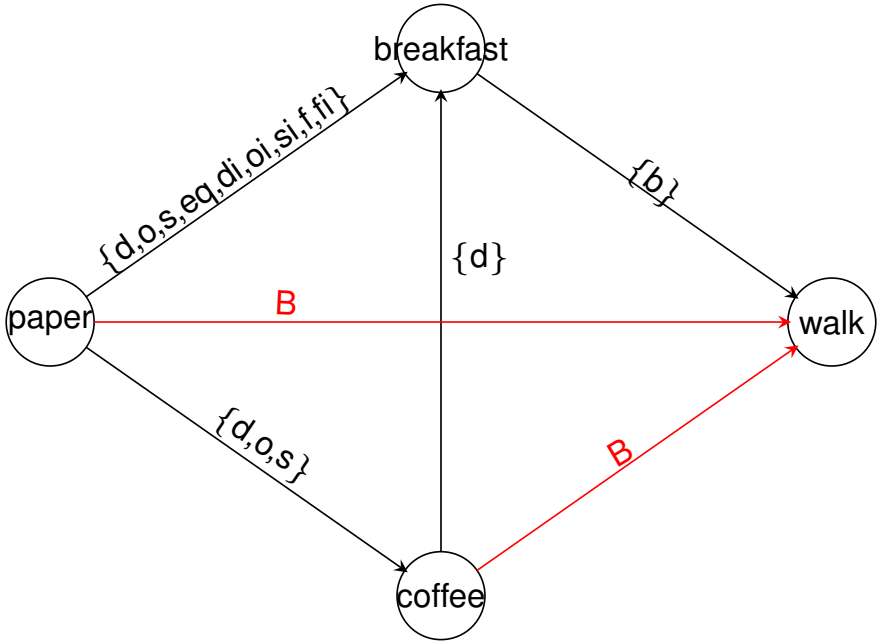
# La méthode de ○-fermeture



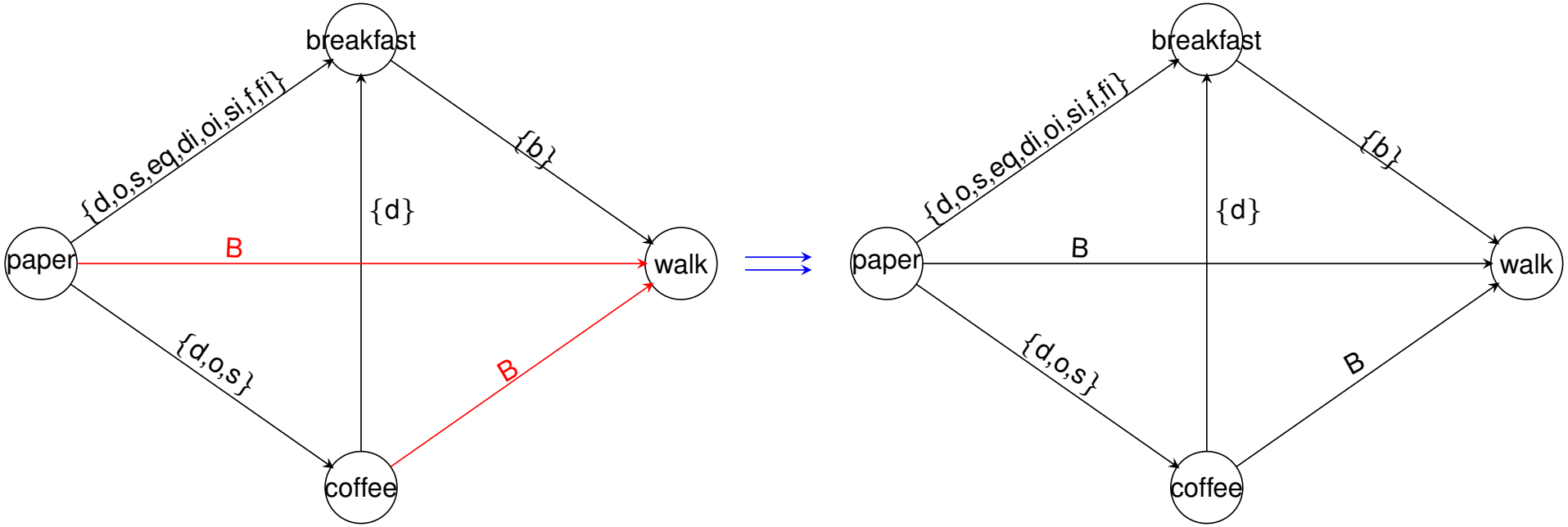
# La méthode de ○-fermeture



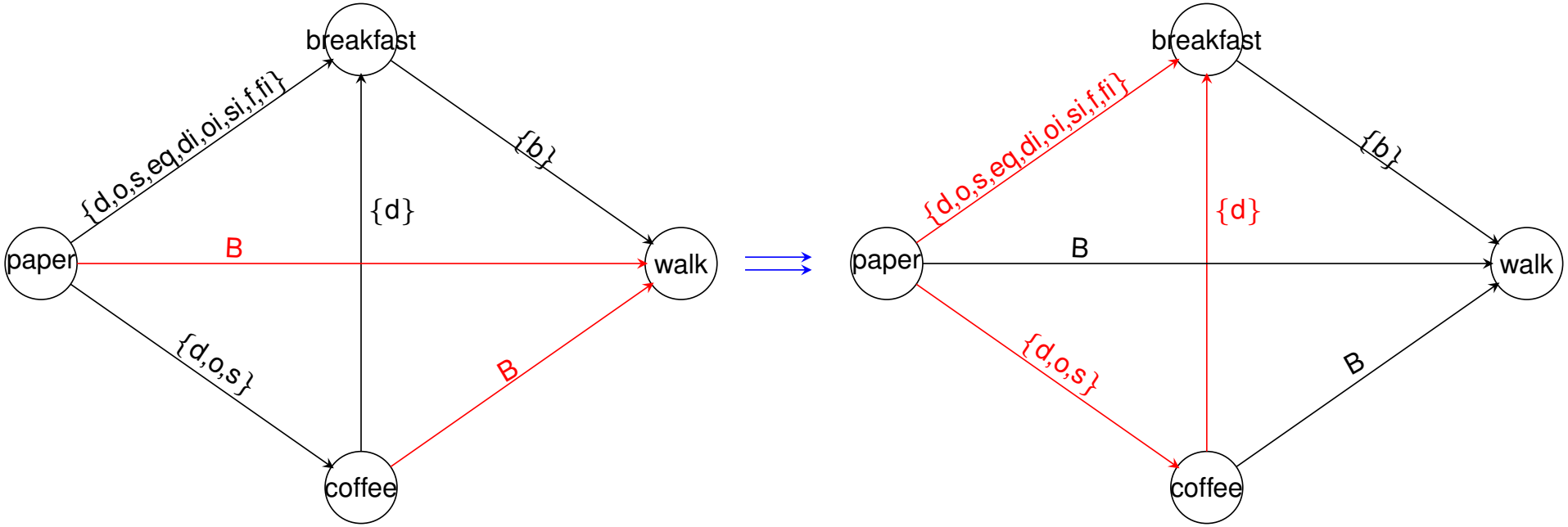
# La méthode de ○-fermeture



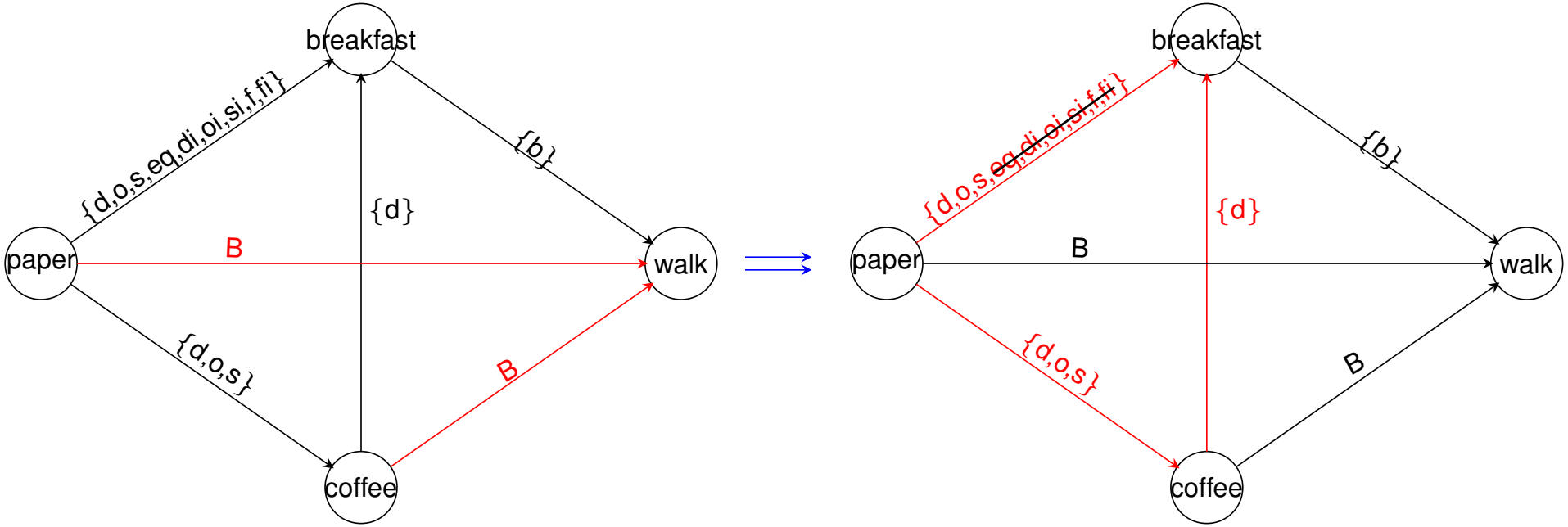
# La méthode de ○-fermeture



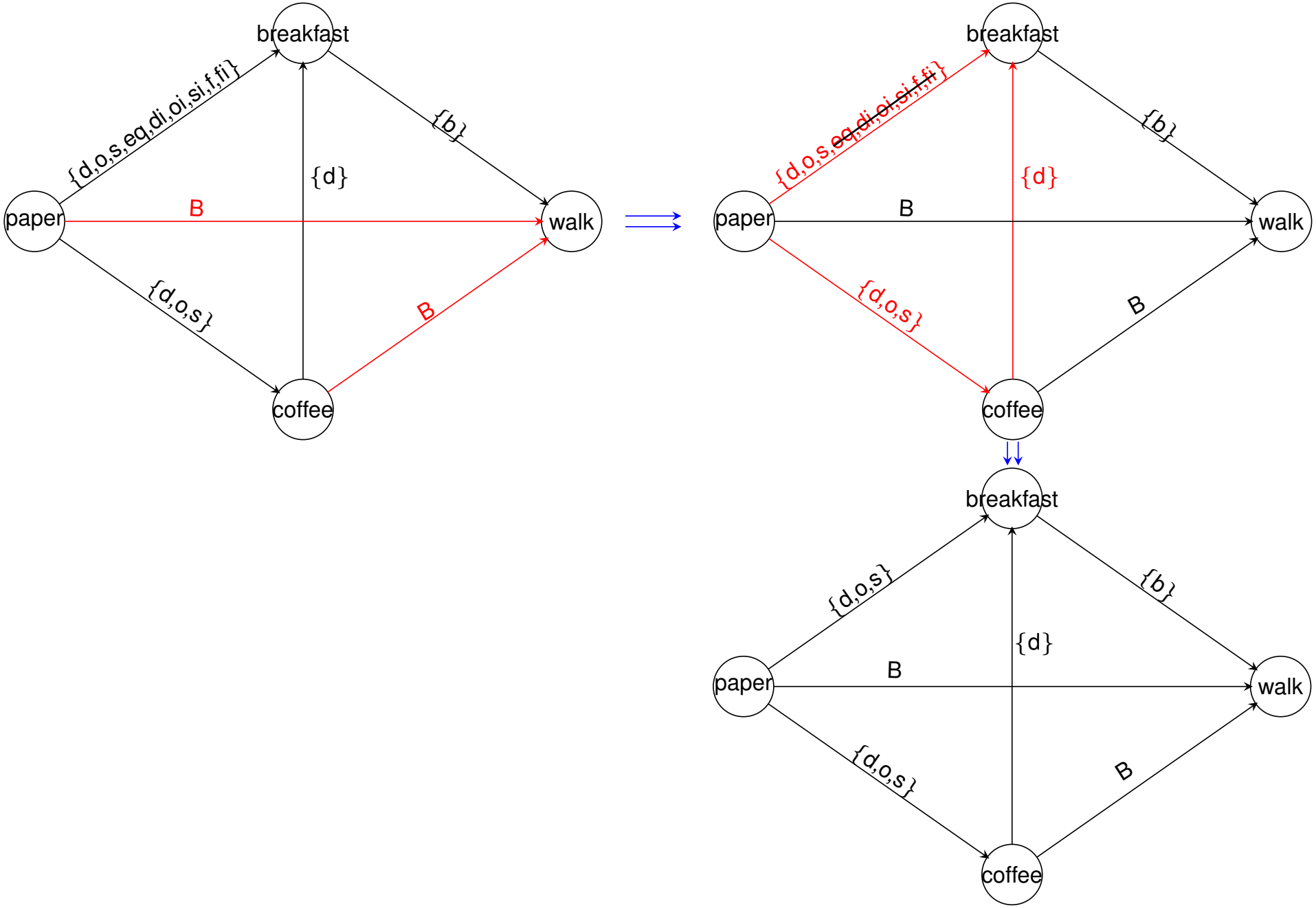
# La méthode de ∘-fermeture



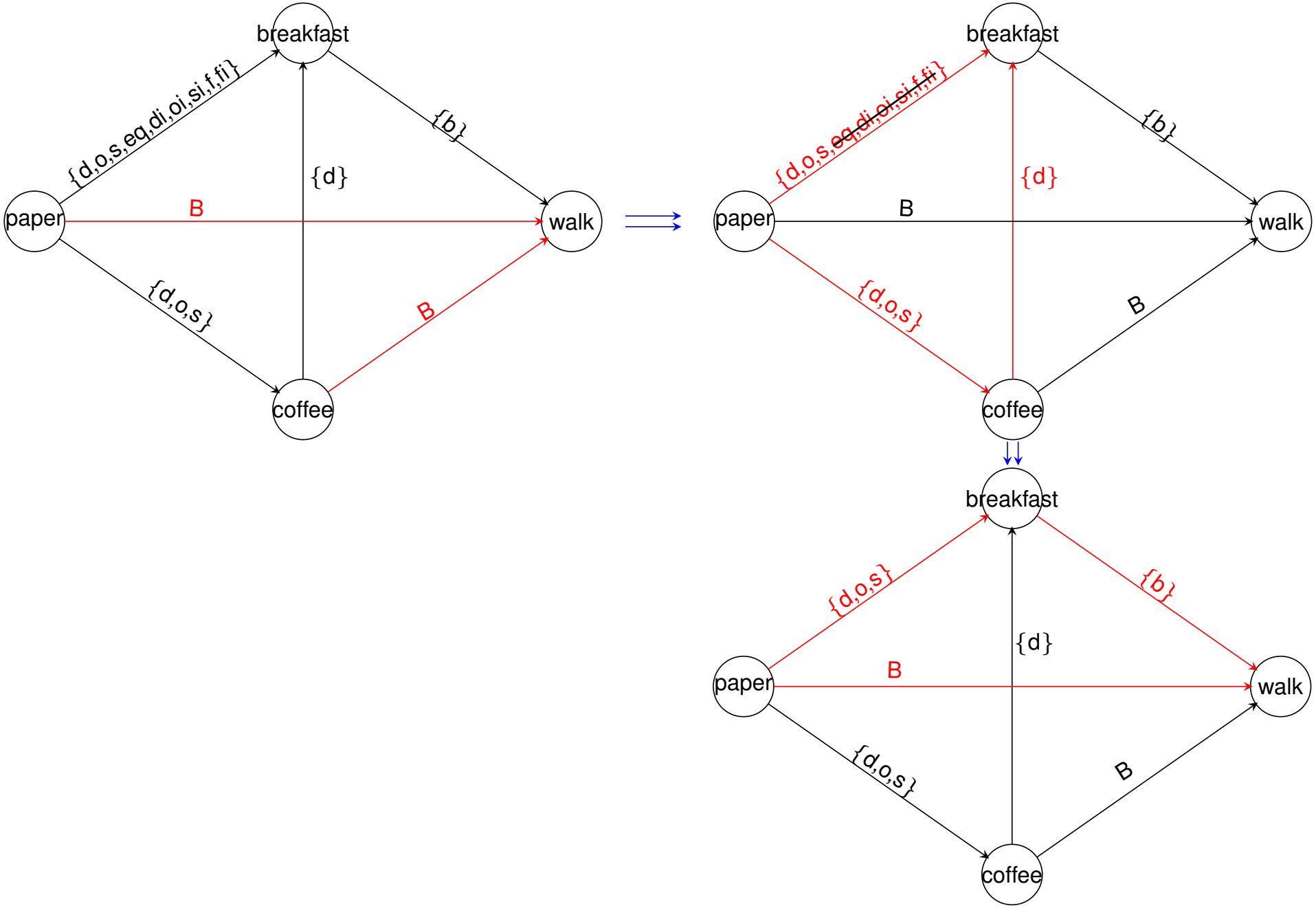
# La méthode de ○-fermeture



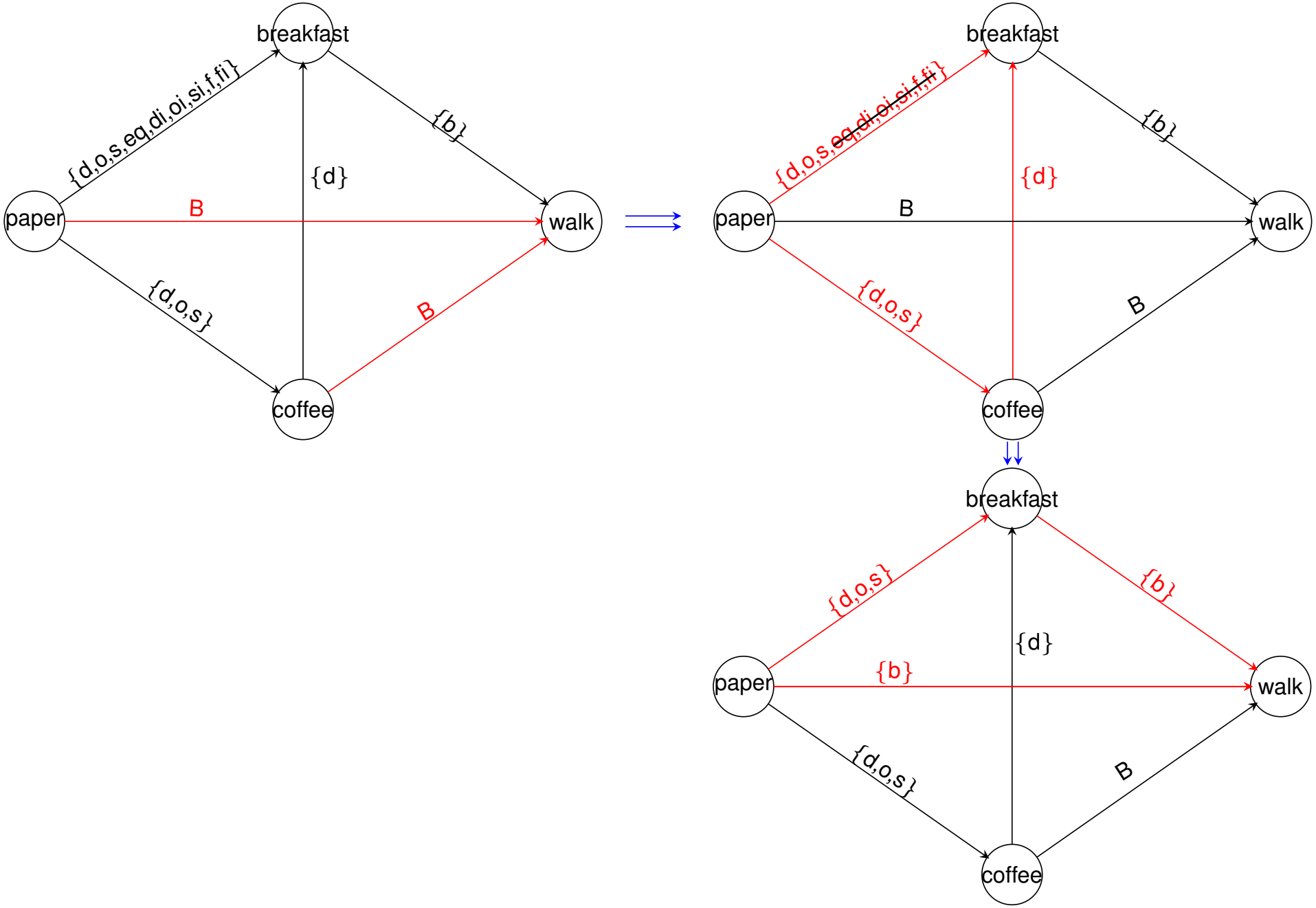
# La méthode de ∘-fermeture



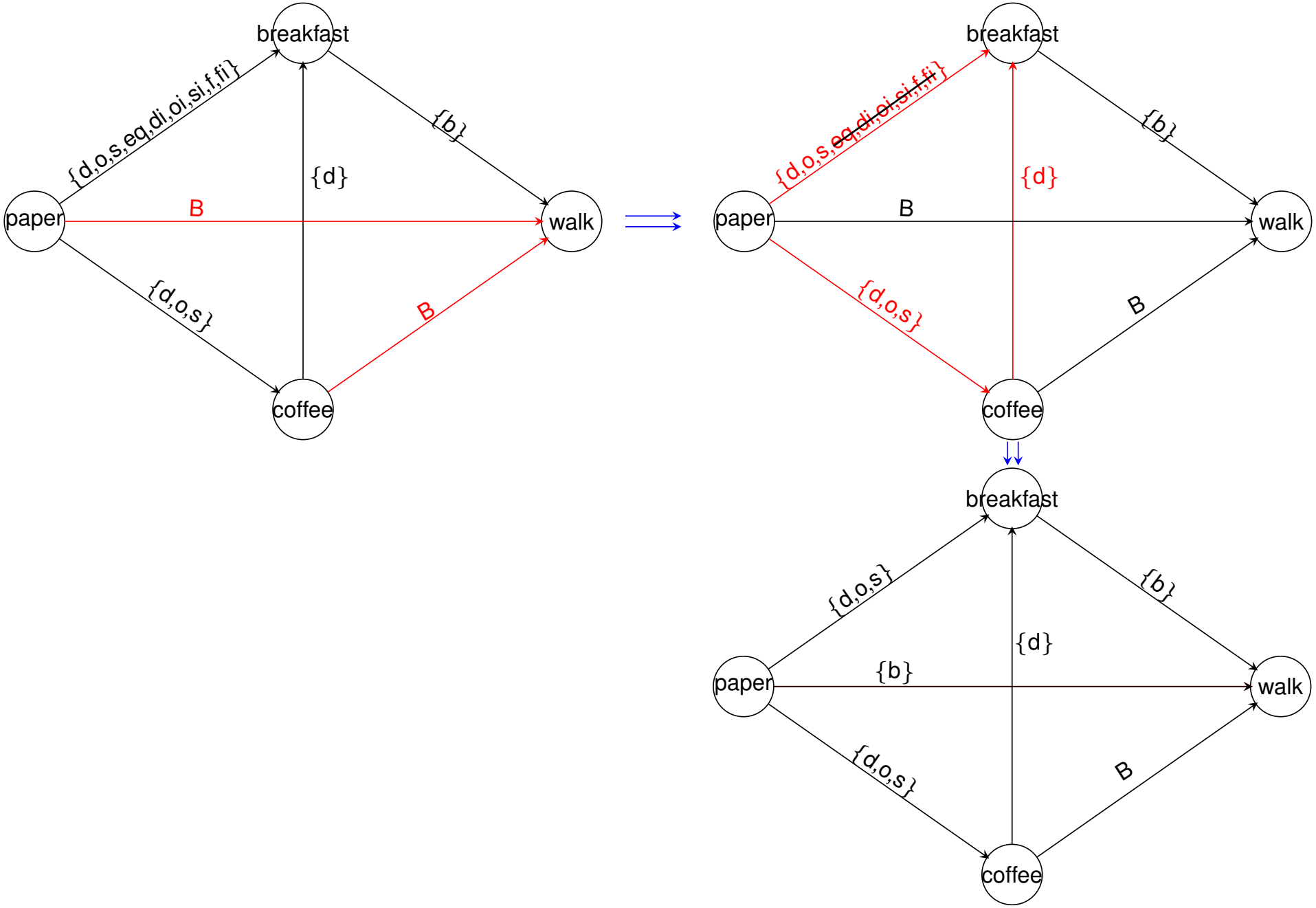
# La méthode de ∘-fermeture



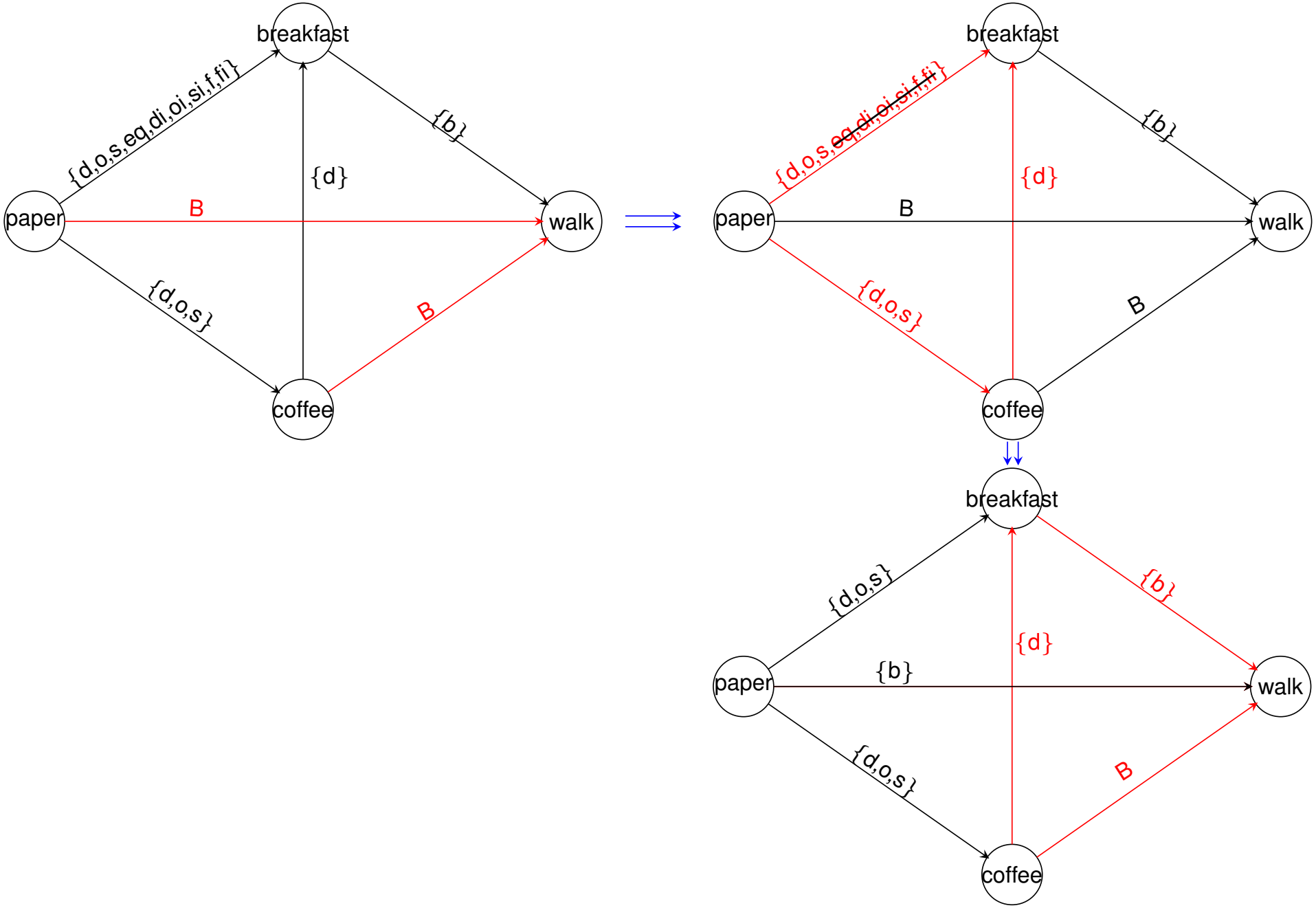
# La méthode de ○-fermeture



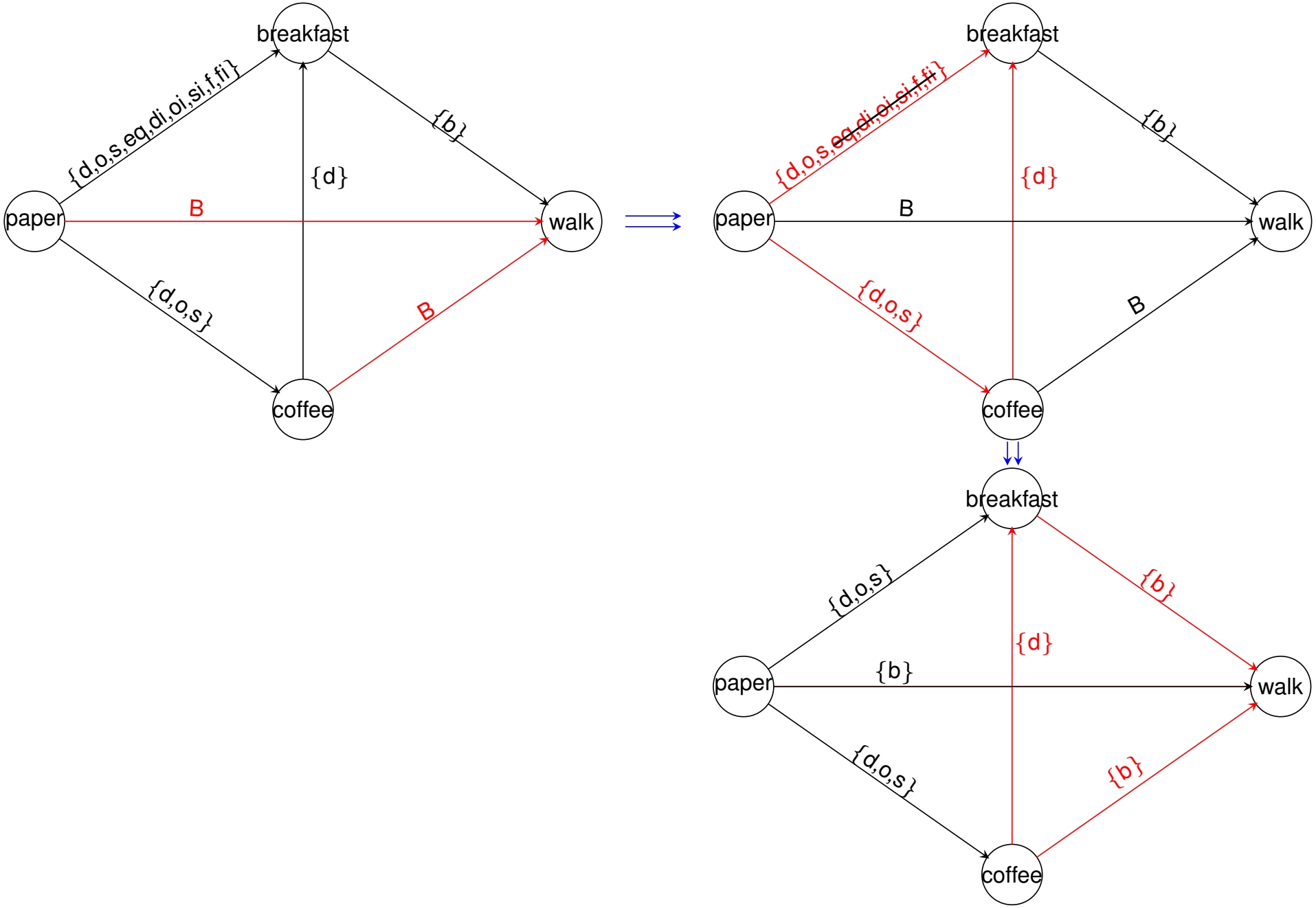
# La méthode de ○-fermeture



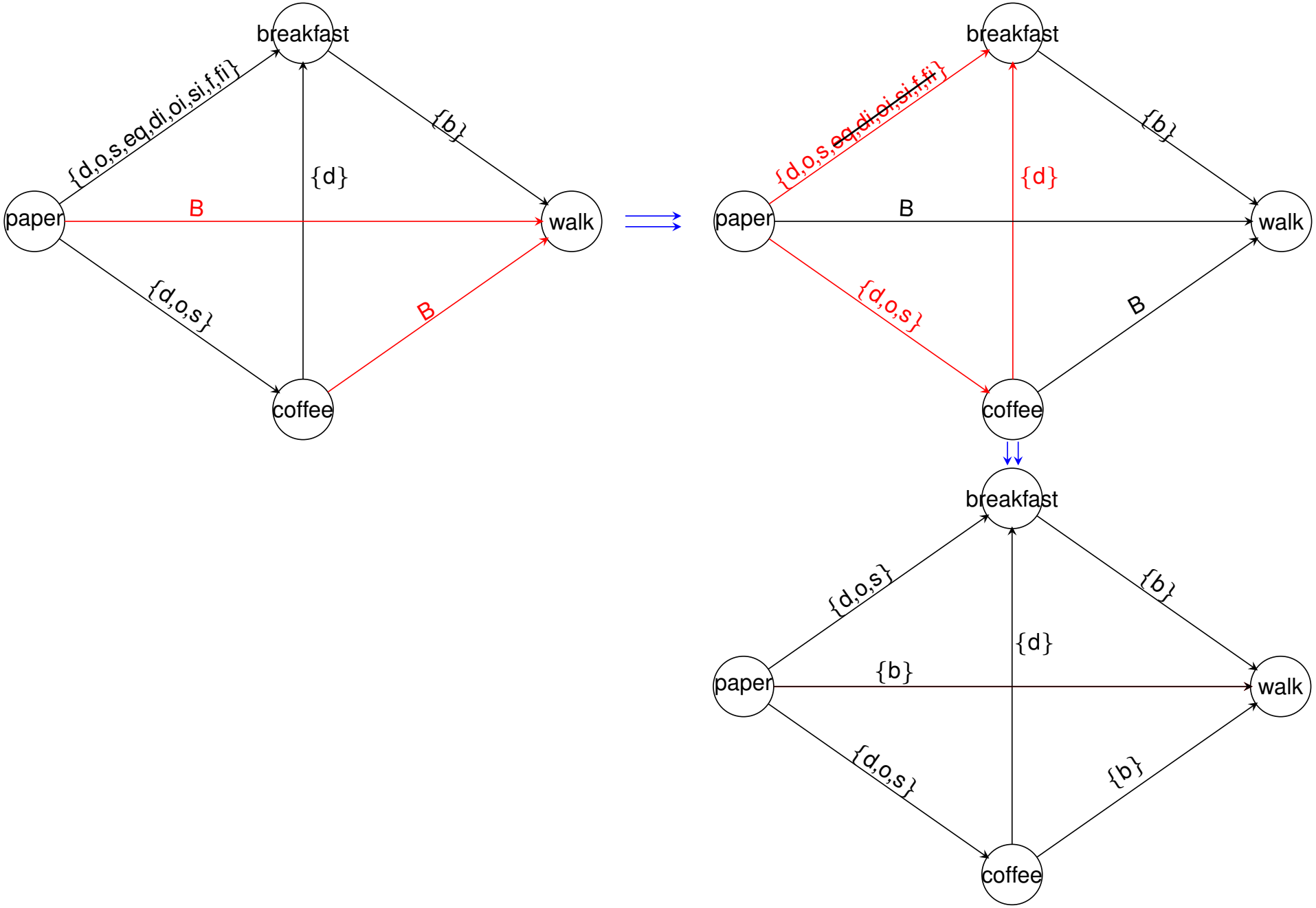
# La méthode de ∘-fermeture



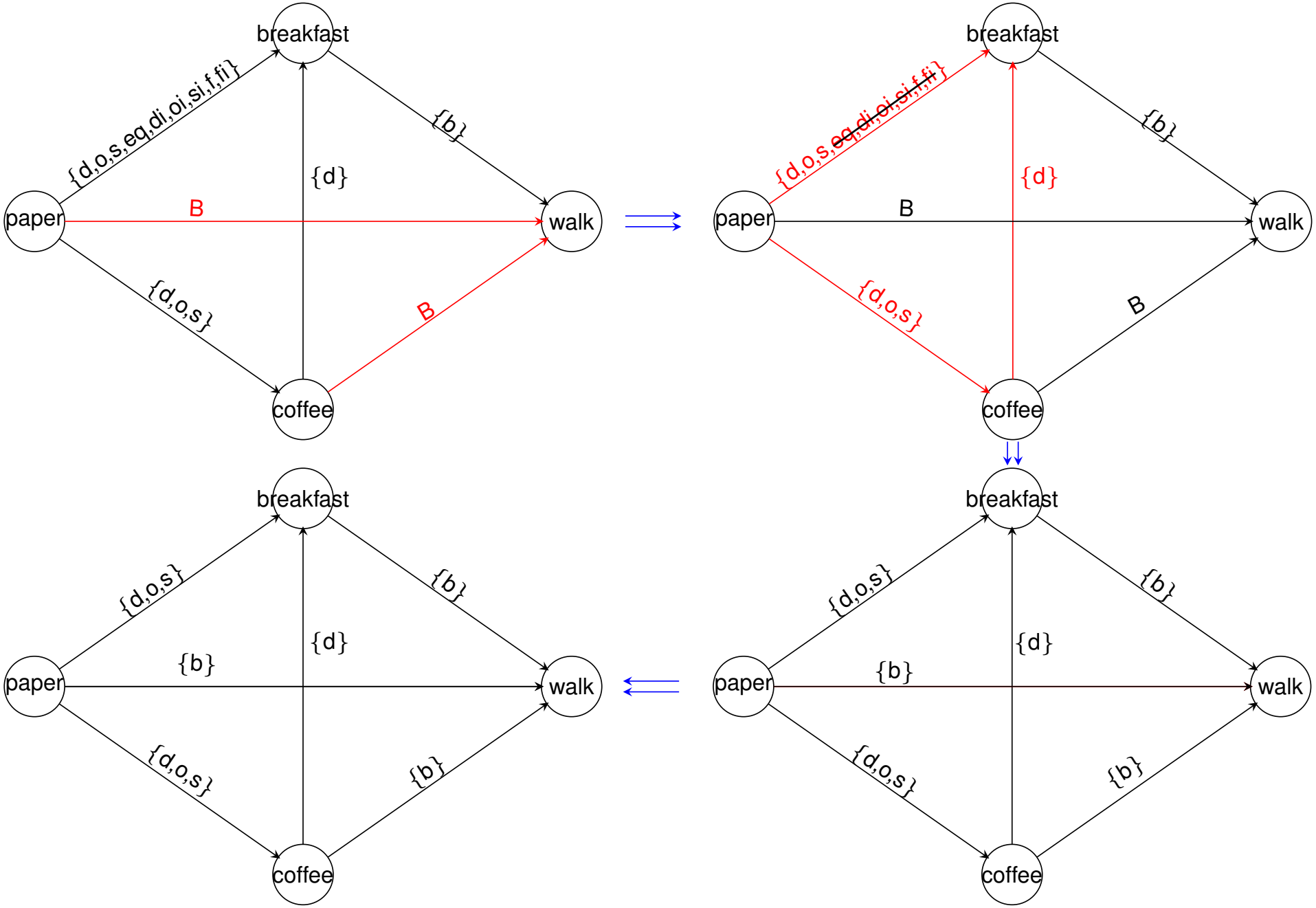
# La méthode de ○-fermeture



# La méthode de ∘-fermeture



# La méthode de ○-fermeture



# Les classes traitables

Soit $\Theta$ un sous ensemble de $A = 2^B$.

## Définition

- Une classe est un ensemble fermé par intersection, inversion et composition.
- $\bar{\Theta}$ est le plus petit ensemble fermé contenant $\Theta$.
- $\Theta$ est une classe ssi $\bar{\Theta} = \Theta$.

## Propriétés

- $ISAT(\Theta)$ est polynomial (resp.NP-complet) ssi $ISAT(\bar{\Theta})$ est polynomial (resp.NP-complet)
- Une classe traitable est une classe pour laquelle on peut répondre à la question (est ce que le réseau est consistent?) en temps polynomial.

# Algorithme de recherche classique

**Algorithm 2**: Function cohérence

**Data**: $\mathcal{N} = (V, C)$

**begin**

  **if** $!ffc(\mathcal{N})$ **then return** false

  Selectionner une contrainte $C_{ij}$ non instanciée

  **if** $C_{ij}$ *n'existe pas* **then return** true

  Partage $C_{ij}$ en sous relations $r_1, ..., r_k$ telles que $r_l \in$ B

  **for** $i \leftarrow 1$ **to** $k$ **do**

    $C_{ij} \leftarrow r_l$

    $C_{ji} \leftarrow {C_{ij}}^{-1}$

    **if** *cohérence*($\mathcal{N}$) **then return** true

  **return** false

**end**

# Algorithme de Nebel

**Algorithm 3**: Function cohérenceN

**Data**: $\mathcal{N} = (V, C)$, *Split*

**begin**

  **if** $!ffc(\mathcal{N})$ **then return** false

  Selectionner une contrainte $C_{ij}$ non instanciée

  **if** $C_{ij}$ *n'existe pas* **then return** true

  Partage $C_{ij}$ en sous relations $r_1, ..., r_k$ telles que $r_l \in Split$

  **for** $i \leftarrow 1$ **to** $k$ **do**

    $C_{ij} \leftarrow r_l$

    $C_{ji} \leftarrow {C_{ij}}^{-1}$

    **if** *cohérenceN*($\mathcal{N}$) **then return** true

  **return** false

**end**

# Contraintes éligibles

## Définition

Une contrainte est dite *éligible* si elle doit être prise en compte lors de la recherche. Pendant l'exécution d'un algorithme de recherche, on instancie seulement les contraintes éligibles.

## Exemple

- Toute contrainte définie avec une relation de *Split* à un moment quelconque* deviendra non éligible. *Split* est une classe traitable pour laquelle la fermeture par composirion est complète.

# Contraintes éligibles

## Définition

Une contrainte est dite *éligible* si elle doit être prise en compte lors de la recherche. Pendant l'exécution d'un algorithme de recherche, on instancie seulement les contraintes éligibles.

## Exemple

- Toute contrainte définie avec une relation de *Split* à un moment quelconque* deviendra non éligible. *Split* est une classe traitable pour laquelle la fermeture par composirion est complète.

*

- Initialement avant de faire la propagation.
- Lors de la propagation.
- Après la propgation.

# Contraintes éligibles

## Définition

Une contrainte est dite *éligible* si elle doit être prise en compte lors de la recherche. Pendant l'exécution d'un algorithme de recherche, on instancie seulement les contraintes éligibles.

## Exemple

- Toute contrainte définie avec une relation de *Split* à un moment quelconque* deviendra non éligible. *Split* est une classe traitable pour laquelle la fermeture par composirion est complète.

Dans la suite *Split* désigne un ensemble de relations pour lequel la méthode de fermeture par faible composition est complète.

# Un nouvel algorithme de $\circ$-fermeture

**Algorithm 4**: Function ffc

**Data**: $\mathcal{N} = (V, C)$ avec $n = |V|$

**begin**

  **repeat**

    $\mathcal{N}' = \mathcal{N}$

    **for** $i, j, k \in 0 \ldots n-1$ **do**

      $C_{ij} \leftarrow C_{ij} \cap (C_{ik} \circ C_{kj})$

      $C_{ji} \leftarrow C_{ij}^{-1}$

      **if** $(C_{ij} == \emptyset)$ **then return** false

      **if** $(C_{ij} \in Split)$ **then** Marquer non éligible $C_{ij}$ et $C_{ji}$

  **until** $\mathcal{N}' == \mathcal{N}$

  **return** true

**end**

# Algorithme

**Algorithm 5**: Function cohérenceÉ

**Data**: $\mathcal{N} = (V, C)$, *Split*)

**begin**

  **if** $!ffc(\mathcal{N})$ **then return** false

  Selectionner une contrainte éligible $C_{ij}$

  **if** $C_{ij}$ *n'existe pas* **then return** true

  Partage $C_{ij}$ en sous relations $r1, ..., r_k$ telles que $r_l \in Split$

  **for** $i \leftarrow 1$ **to** $k$ **do**

    $C_{ij} \leftarrow r_l$

    $C_{ji} \leftarrow {C_{ij}}^{-1}$

    Marquer non éligible $C_{ij}$ et $C_{ji}$

    **if** *(cohérenceÉ*$(\mathcal{N})$*)* **then return** true

  **return** false

**end**

# Algorithme

**Algorithm 6**: Function cohérenceE

**Data**: $\mathcal{N} = (V, C)$, $Split$)

**begin**

  **for** $i, j \in 0, ...n-1$ **do**

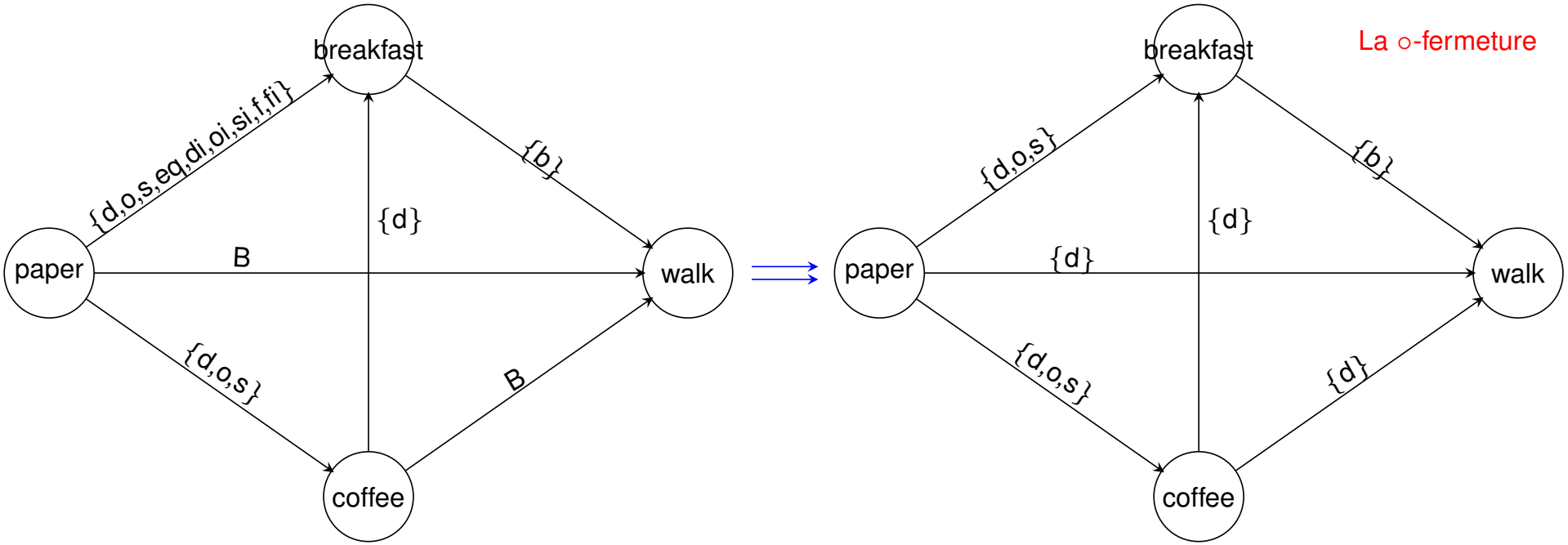
    **if** $(C_{ij} \in Split)$ **then** Marquer non éligible $C_{ij}$

    **else** Marquer éligible $C_{ij}$

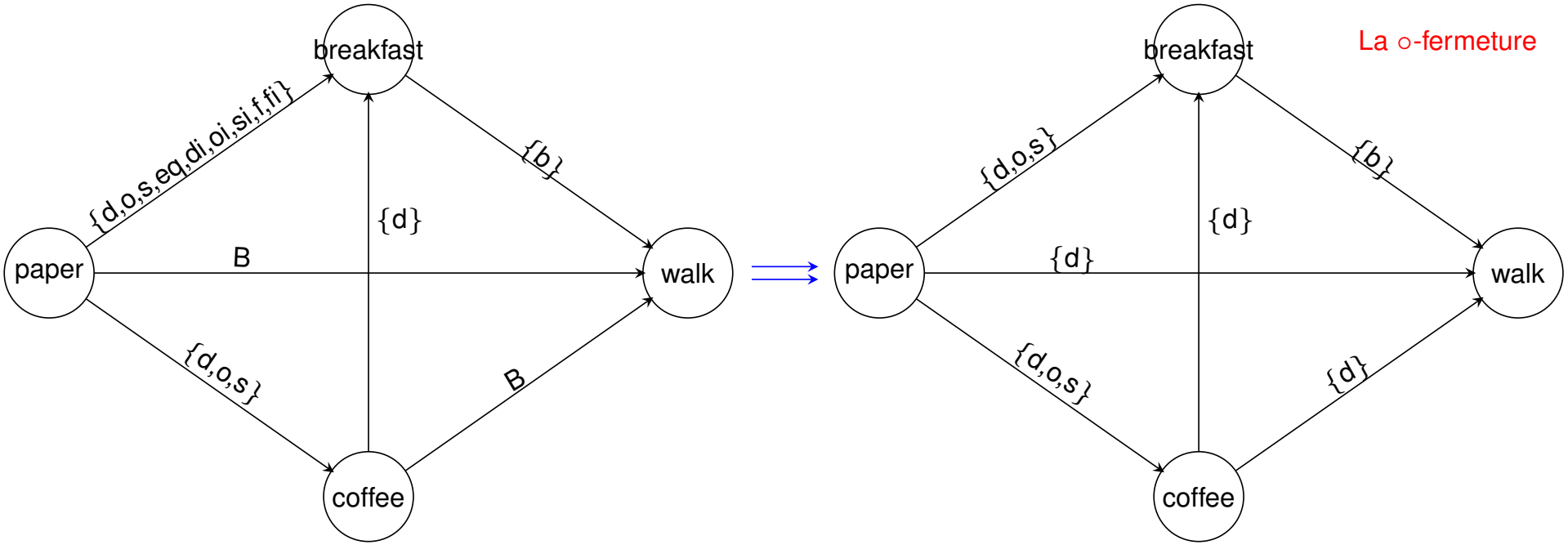
  **return** cohérenceÉ($\mathcal{N}$)

**end**

# Exemple


breakfast
{d,o,s,eq,di,oi,si,f,fi}
{d}
{b}
paper
B
walk
{d,o,s}
B
coffee
breakfast
La ∘-fermeture
{d,o,s}
{d}
{b}
paper
{d}
walk
{d,o,s}
{d}
coffee

# Exemple




Étant donné que toutes les contraintes sont préconvexes alors elles sont toutes non éligibles. Ce réseau est chemin-cohérent donc il est consistent par rappport à la méthode de faible composition

# Contraintes frozen

## Contraintes structurales

Ces contraintes sont utilisées pour représenter la structure du problème. Elles forment un ensemble de contraintes consistent.

## Contraintes critiques

Ce sont les contraintes que nous intéressent.

## Exemple

- Le problème de n-reines.
- Le JobShop.

# Frozen constraints

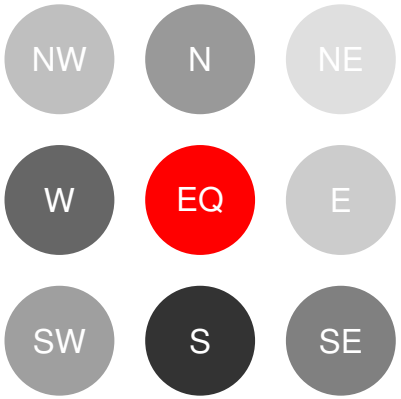
**Définition**

Une contrainte est dite *Frozen* si elle ne doit pas être modifiée ni pendant la propagation ni pendant la recherche.

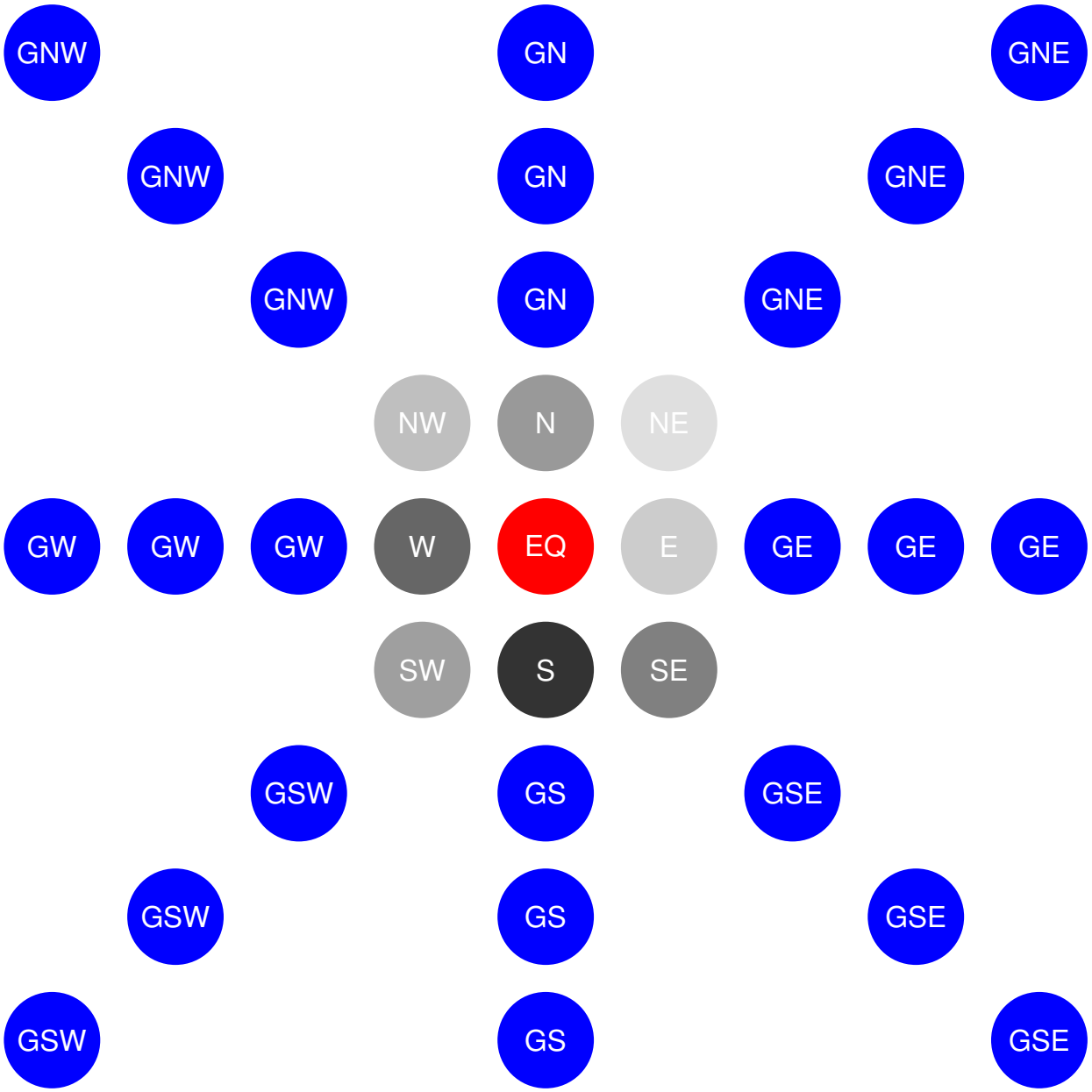
**Exemple**

- Les contraintes entre les cases (Pb de n-reines) sont des contraintes frozen.

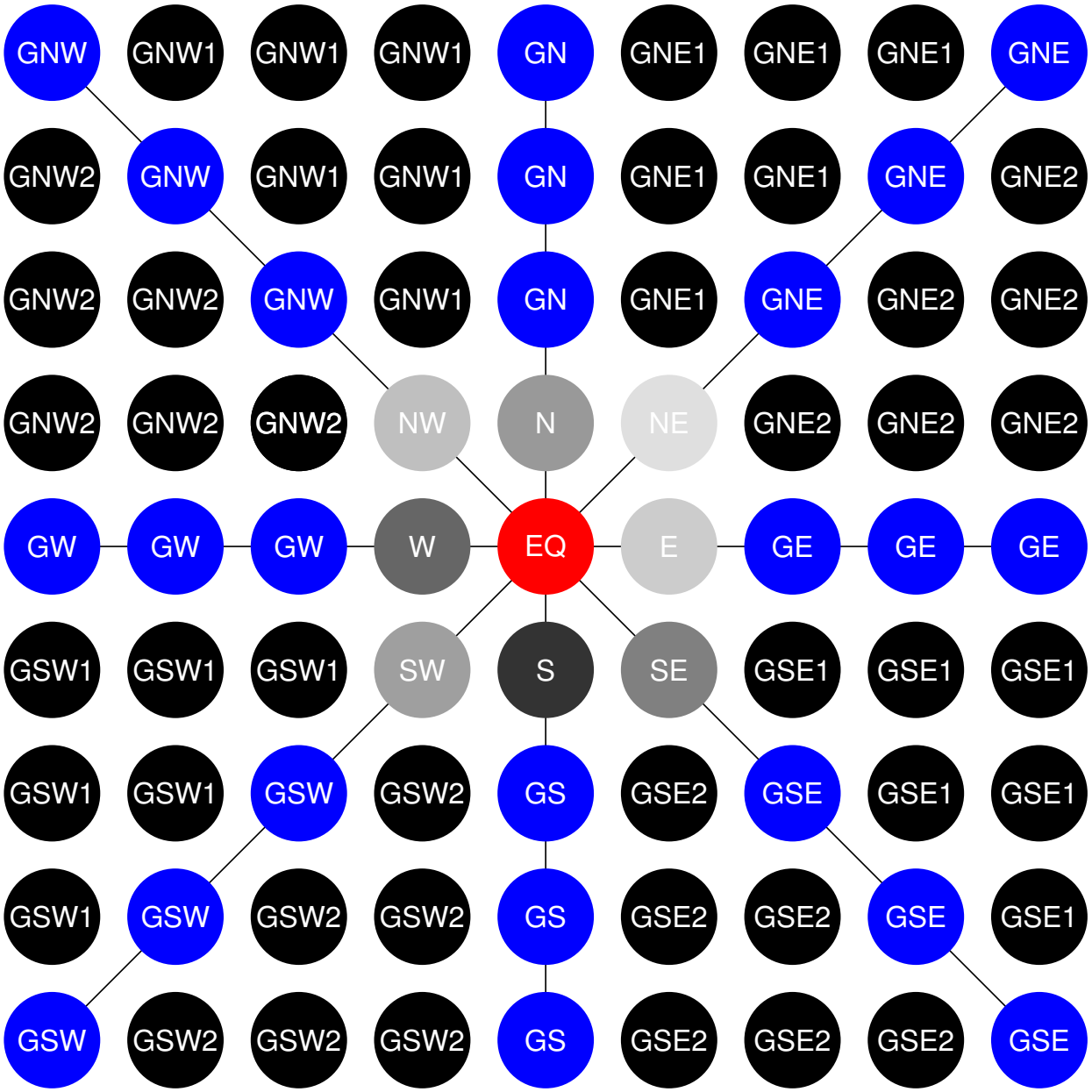
# Algèbre *ACD*25



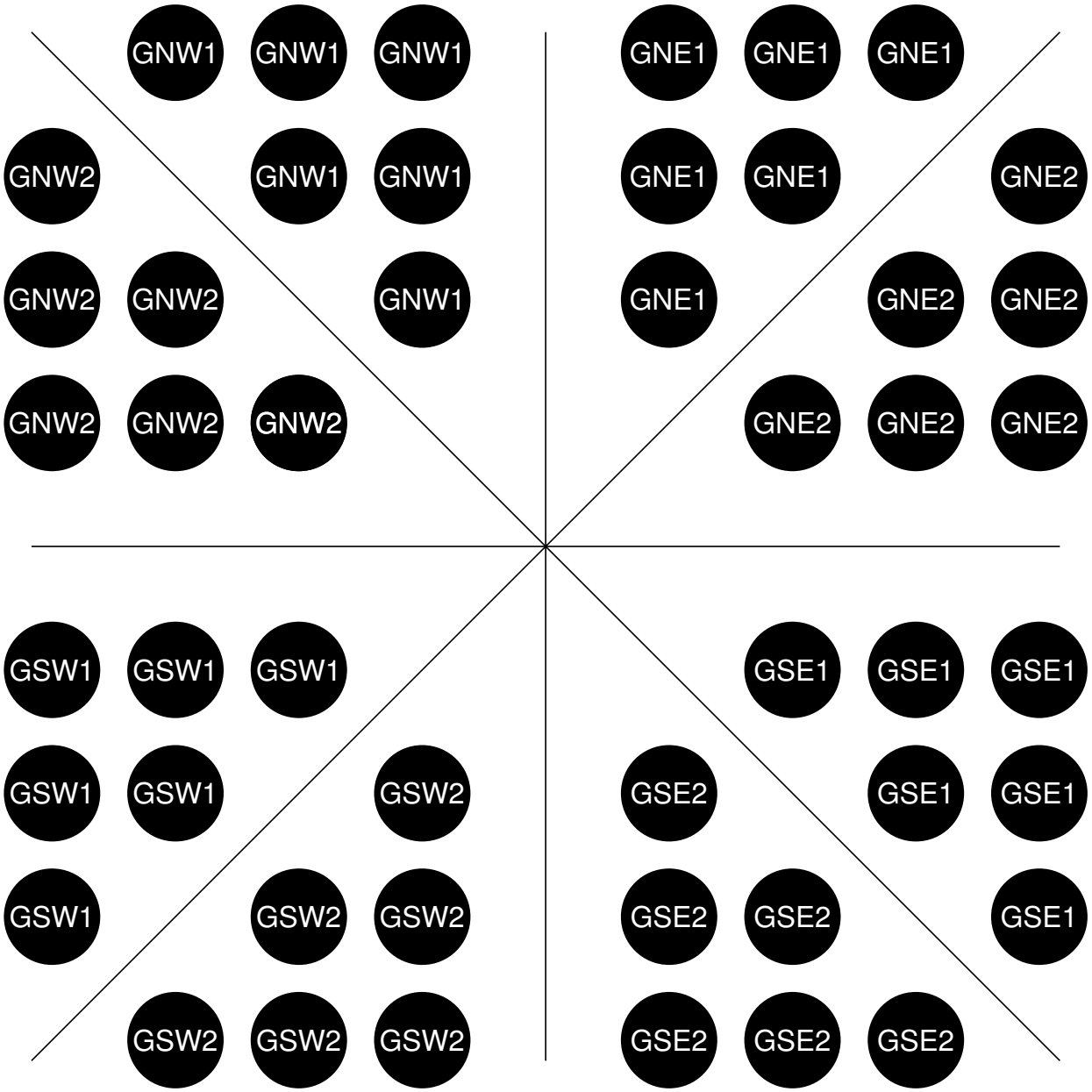
# Algèbre *ACD25*



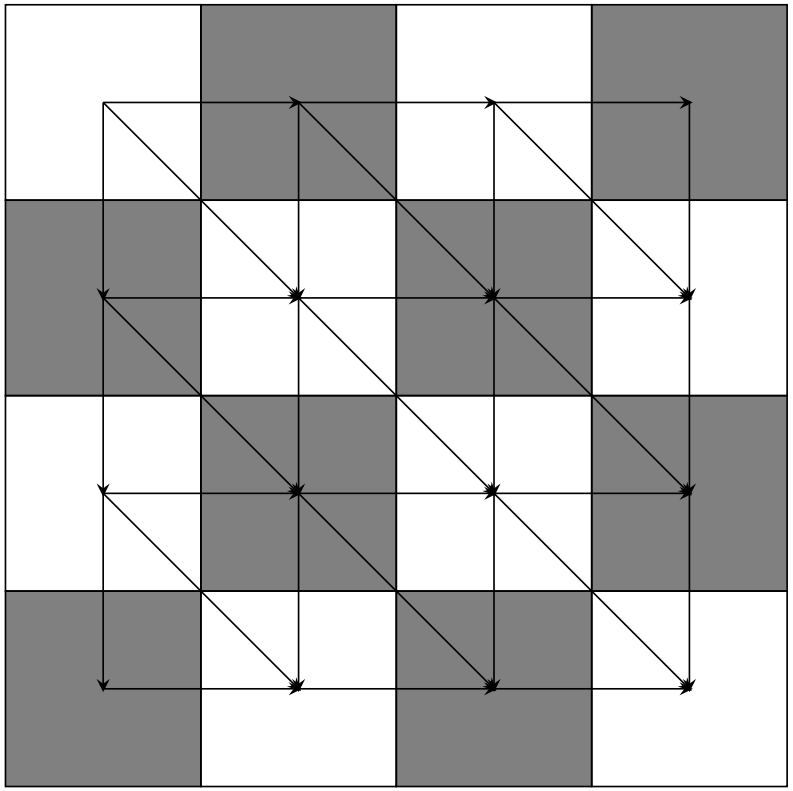
# Algèbre *ACD25*

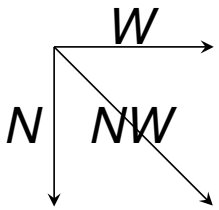


# Représentaion du pb. des reines à l'aide de *ACD25*

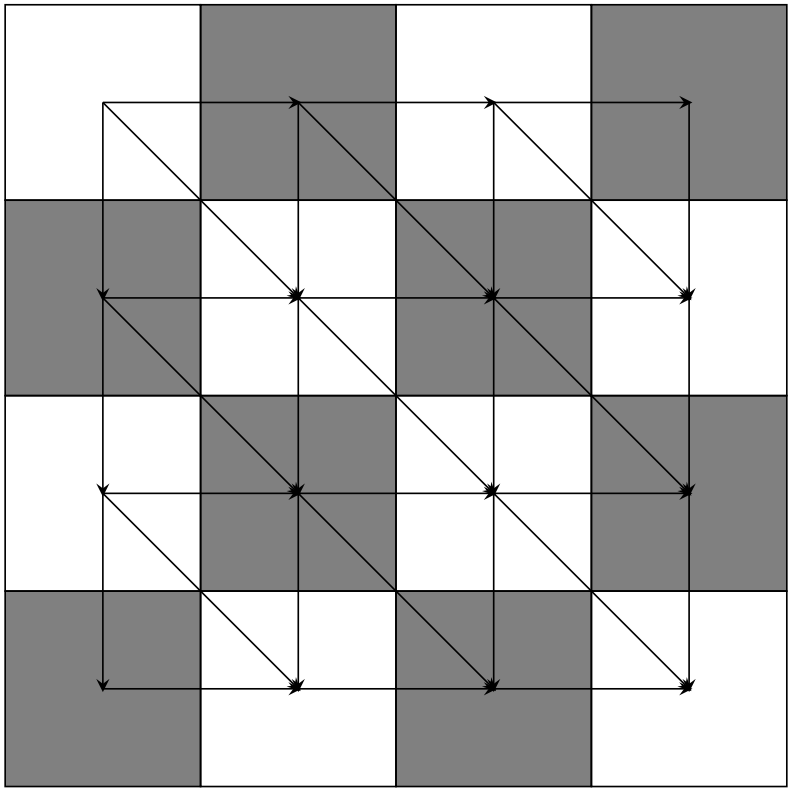


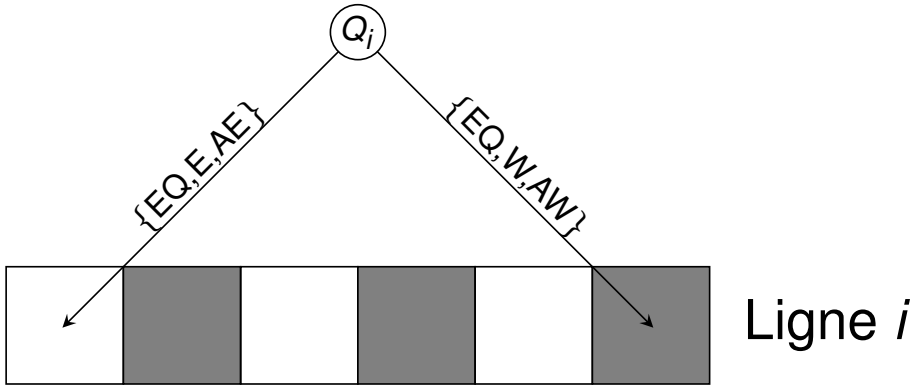
# Représentaion du pb. des reines à l'aide de *ACD*25

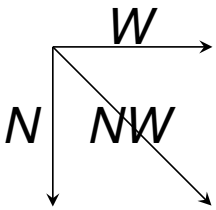




# Représentaion du pb. des reines à l'aide de *ACD*25





# Un nouvel algorithme de $\circ$-fermeture

**Algorithm 7**: Function ffc

**Data**: $\mathcal{N} = (V, C)$ avec $n = |V|$, *SFrozen*

```
begin
    repeat
        N' = N
        for i, j ∈ 0...n − 1 do
            if c_ij ∉ SFrozen then
            for k ∈ 0...n − 1 do
                C_ij ← C_ij ∩ (C_ik ∘ C_kj)
                C_ji ← C_ij^−1
                if (C_ij == ∅) then return false
                if (C_ij ∈ Split) then Marquer non éligible C_ij et C_ji
    until N' == N
    return true
end
```

# Algorithme

**Algorithm 8**: Function cohérenceEF′

**Data**: $\mathcal{N} = (V, C)$ avec $n = |V|$, *Split*)

**begin**

  **if** $!ffc(\mathcal{N})$ **then return** false

  Selectionner une contrainte éligible $C_{ij}$

  **if** $C_{ij}$ *n'existe pas* **then return** true

  Partage $C_{ij}$ en sous relations $r_1, ..., r_k$ telles que $r_l \in Split$

  **for** $i \leftarrow 1$ **to** $k$ **do**

    $C_{ij} \leftarrow r_l$

    $C_{ji} \leftarrow {C_{ij}}^{-1}$

    Marquer non éligible $C_{ij}$ et $C_{ji}$

    **if** *(cohérenceEF′($\mathcal{N}$))* **then return** true

  **return** false

**end**

# Algorithme

**Algorithm 9**: Function cohérenceEF

**Data**: $\mathcal{N} = (V, C)$ avec $n = |V|$, *Split*, *SFrozen*

**begin**

  **for** $i, j \in 0, ...n-1$ **do**

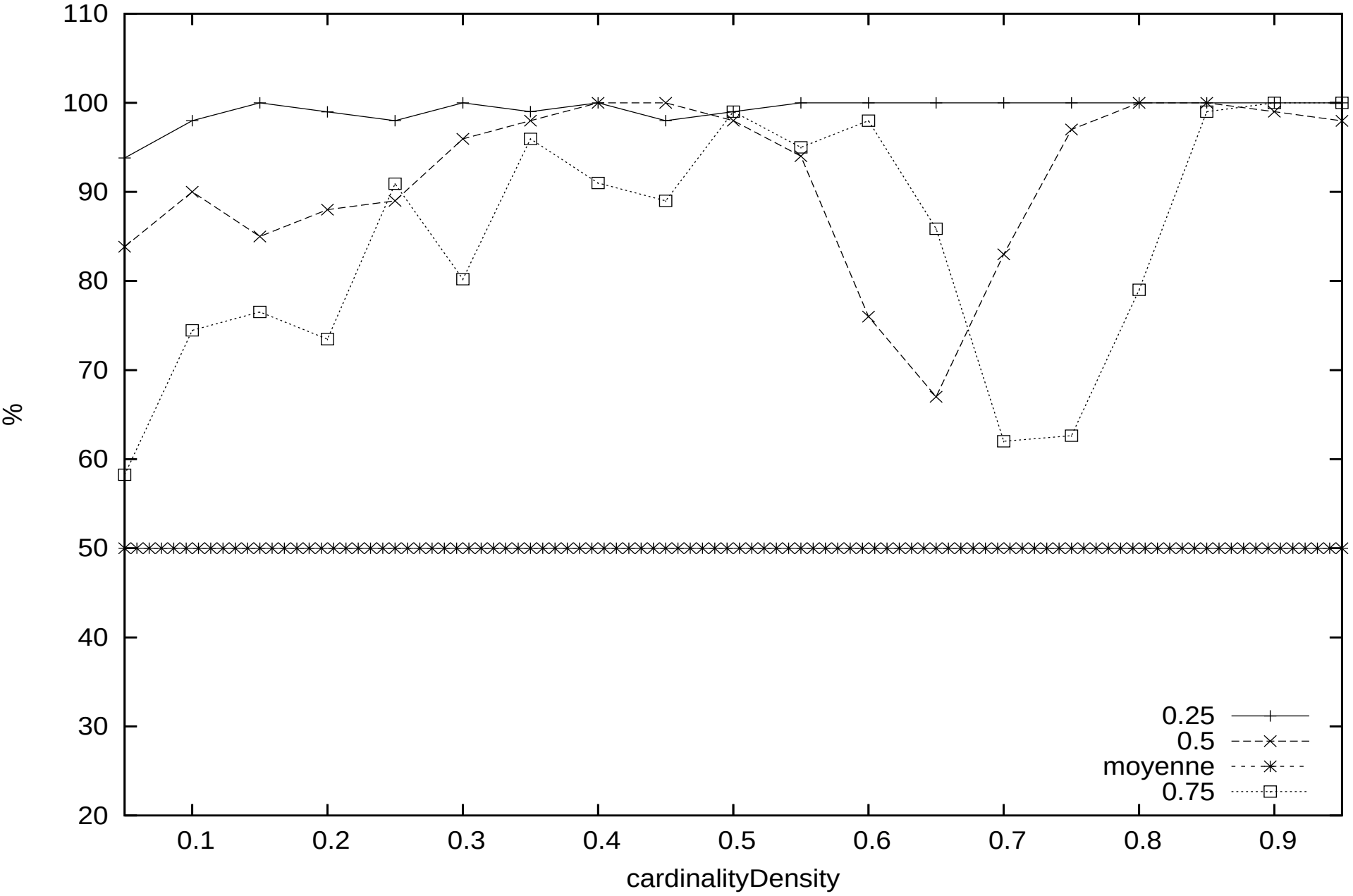
    **if** $((C_{ij} \in Split)$ *ou* $(C_{ij} \in SFrozen))$ **then** Marquer non éligible $C_{ij}$

    **else** Marquer éligible $C_{ij}$

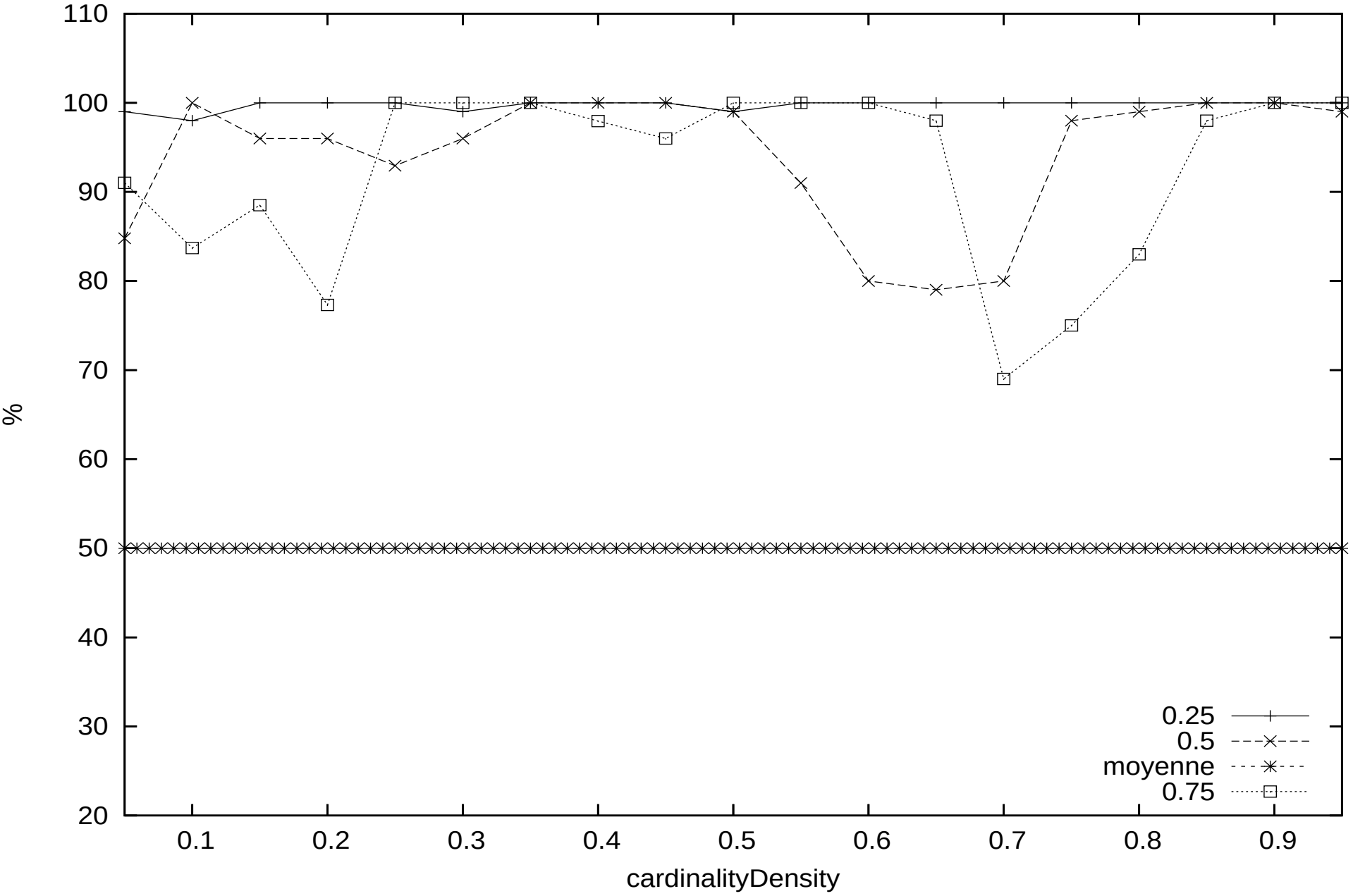
  **return** cohérenceEF $'(\mathcal{N})$

**end**

# cohérenceE vs cohérenceN en utilisant *AscDom*



# cohérenceE vs cohérenceN en utilisant *triangle*



# cohérenceE vs cohérenceN ( $nTD = 0.25$ avec $AscDom$)

| nTD=0.25 | $T_{ffc}$ | $T_E$ | $T_N$ | $BK_E$ | $BK_N$ |
|---|---|---|---|---|---|
| cD=0.05 | 190 | 35 | 64 | 0 | 0 |
| cD=0.1 | 203 | 42 | 78 | 0 | 0 |
| cD=0.15 | 221 | 61 | 112 | 0 | 0 |
| cD=0.2 | 232 | 86 | 143 | 0 | 0 |
| cD=0.25 | 247 | 140 | 226 | 0 | 0 |
| cD=0.3 | 240 | 213 | 322 | 0 | 0 |
| cD=0.35 | 191 | 327 | 467 | 0 | 0 |
| cD=0.4 | 110 | 450 | 613 | 0 | 0 |
| cD=0.45 | 61 | 510 | 670 | 0 | 0 |
| cD=0.5 | 37 | 511 | 689 | 0 | 0 |
| cD=0.55 | 29 | 511 | 748 | 0 | 4 |
| cD=0.6 | 27 | 502 | 692 | 0 | 0 |
| cD=0.65 | 21 | 457 | 684 | 0 | 0 |
| cD=0.7 | 19 | 398 | 610 | 0 | 0 |
| cD=0.75 | 23 | 376 | 569 | 0 | 0 |
| cD=0.8 | 18 | 314 | 495 | 0 | 0 |
| cD=0.85 | 16 | 249 | 403 | 0 | 0 |
| cD=0.9 | 13 | 151 | 252 | 0 | 0 |
| cD=0.95 | 9 | 33 | 66 | 0 | 0 |

# cohérenceE vs cohérenceN ( $nTD = 0.25$ **avec** *triangle*)

| nTD=0.25 | $T_{ffc}$ | $T_E$ | $T_N$ | $BK_E$ | $BK_N$ |
|---|---|---|---|---|---|
| cD=0.05 | 180 | 31 | 60 | 0 | 0 |
| cD=0.1 | 194 | 41 | 79 | 0 | 0 |
| cD=0.15 | 212 | 61 | 109 | 0 | 0 |
| cD=0.2 | 227 | 87 | 143 | 0 | 0 |
| cD=0.25 | 244 | 142 | 227 | 0 | 0 |
| cD=0.3 | 232 | 215 | 318 | 0 | 0 |
| cD=0.35 | 183 | 323 | 459 | 0 | 0 |
| cD=0.4 | 100 | 462 | 628 | 1 | 1 |
| cD=0.45 | 51 | 504 | 673 | 0 | 0 |
| cD=0.5 | 35 | 508 | 775 | 0 | 5 |
| cD=0.55 | 23 | 507 | 698 | 0 | 0 |
| cD=0.6 | 17 | 485 | 701 | 0 | 0 |
| cD=0.65 | 19 | 455 | 676 | 0 | 0 |
| cD=0.7 | 16 | 402 | 614 | 0 | 0 |
| cD=0.75 | 18 | 361 | 559 | 0 | 0 |
| cD=0.8 | 15 | 303 | 486 | 0 | 0 |
| cD=0.85 | 12 | 242 | 397 | 0 | 0 |
| cD=0.9 | 11 | 149 | 247 | 0 | 0 |
| cD=0.95 | 5 | 32 | 63 | 0 | 0 |

# cohérenceE vs cohérenceN ( $nTD = 0.5$ avec $AscDom$)

| nTD=0.5 | $T_{ffc}$ | $T_E$ | $T_N$ | $BK_E$ | $BK_N$ |
|---|---|---|---|---|---|
| cD=0.05 | 137 | 11 | 17 | 0 | 0 |
| cD=0.1 | 138 | 9 | 14 | 0 | 0 |
| cD=0.15 | 151 | 11 | 18 | 0 | 0 |
| cD=0.2 | 162 | 14 | 24 | 0 | 0 |
| cD=0.25 | 175 | 18 | 33 | 0 | 0 |
| cD=0.3 | 198 | 24 | 45 | 0 | 0 |
| cD=0.35 | 231 | 34 | 63 | 0 | 0 |
| cD=0.4 | 268 | 55 | 98 | 0 | 0 |
| cD=0.45 | 296 | 90 | 155 | 0 | 0 |
| cD=0.5 | 290 | 296 | 387 | 0 | 0 |
| cD=0.55 | 108 | 1289 | 1320 | 19 | 12 |
| cD=0.6 | 72 | **37566** | **34338** | 1078 | 839 |
| cD=0.65 | 59 | **28751** | **27061** | 785 | 640 |
| cD=0.7 | 52 | 4964 | 10096 | 136 | 190 |
| cD=0.75 | 47 | 717 | 987 | 0 | 2 |
| cD=0.8 | 48 | 626 | 873 | 0 | 0 |
| cD=0.85 | 42 | 520 | 752 | 0 | 0 |
| cD=0.9 | 31 | 384 | 564 | 0 | 0 |
| cD=0.95 | 17 | 179 | 281 | 0 | 0 |

# cohérenceE vs cohérenceN ( $nTD = 0.5$ **avec** *triangle*)

| nTD=0.5 | $T_{ffc}$ | $T_E$ | $T_N$ | $BK_E$ | $BK_N$ |
|---|---|---|---|---|---|
| cD=0.05 | 135 | 11 | 17 | 0 | 0 |
| cD=0.1 | 132 | 6 | 13 | 0 | 0 |
| cD=0.15 | 142 | 9 | 17 | 0 | 0 |
| cD=0.2 | 154 | 13 | 24 | 0 | 0 |
| cD=0.25 | 175 | 19 | 34 | 0 | 0 |
| cD=0.3 | 198 | 26 | 47 | 0 | 0 |
| cD=0.35 | 225 | 33 | 62 | 0 | 0 |
| cD=0.4 | 262 | 56 | 98 | 0 | 0 |
| cD=0.45 | 298 | 94 | 153 | 0 | 0 |
| cD=0.5 | 281 | 290 | 380 | 0 | 0 |
| cD=0.55 | 109 | 1642 | 1724 | 29 | 23 |
| cD=0.6 | 68 | 6272 | 6325 | 162 | 139 |
| cD=0.65 | 51 | **14188** | **11260** | 429 | 297 |
| cD=0.7 | 45 | 1078 | 1283 | 7 | 7 |
| cD=0.75 | 40 | 715 | 935 | 0 | 0 |
| cD=0.8 | 38 | 625 | 879 | 0 | 0 |
| cD=0.85 | 36 | 522 | 748 | 0 | 0 |
| cD=0.9 | 24 | 368 | 570 | 0 | 0 |
| cD=0.95 | 13 | 167 | 263 | 0 | 0 |

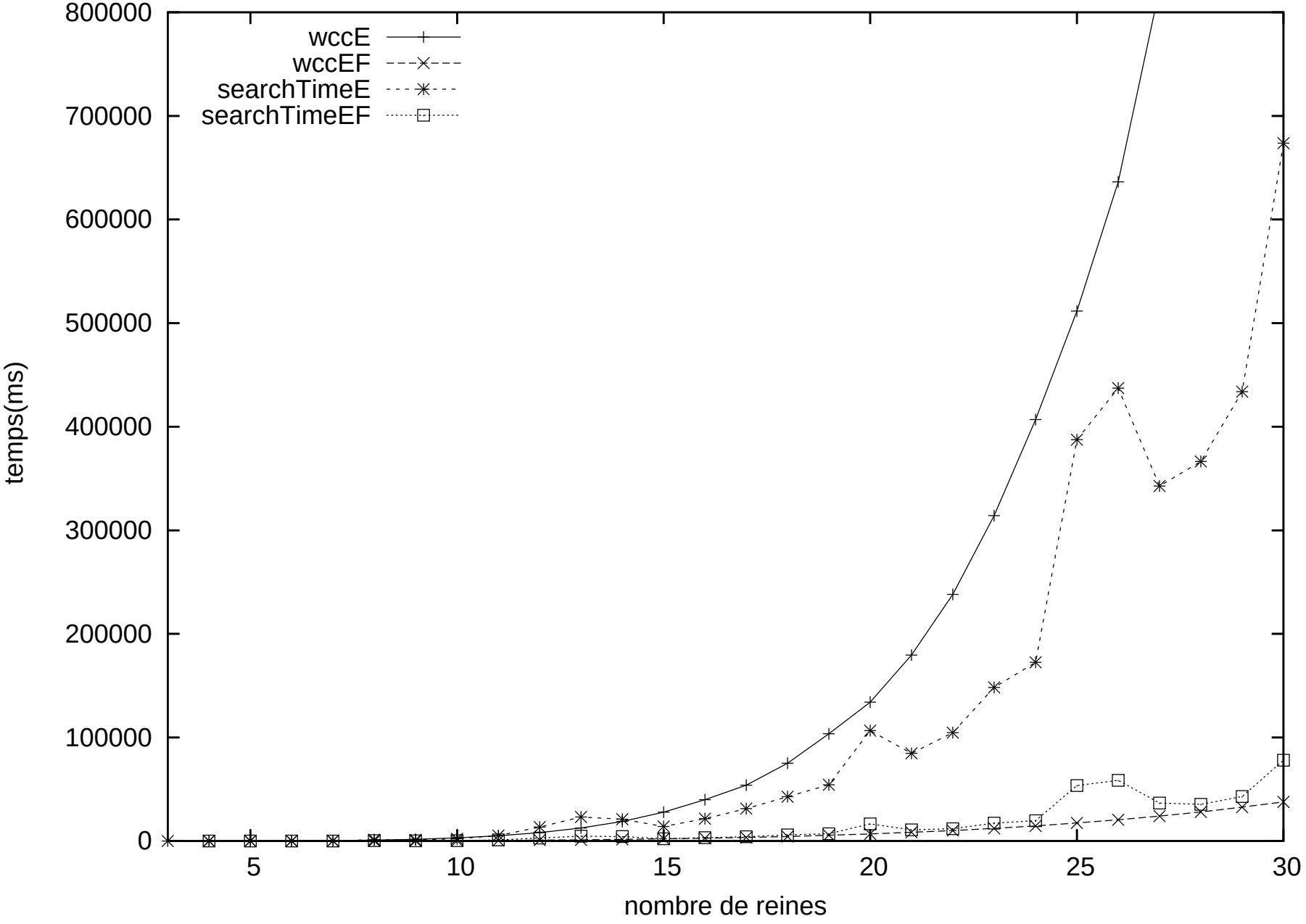
# cohérenceE vs cohérenceN ( $nTD = 0.75$ avec $AscDom$)

| nTD=0.75 | $T_{ffc}$ | $T_E$ | $T_N$ | $BK_E$ | $BK_N$ |
|---|---|---|---|---|---|
| cD=0.05 | 132 | 14 | 13 | 0 | 0 |
| cD=0.1 | 134 | 9 | 12 | 0 | 0 |
| cD=0.15 | 140 | 9 | 12 | 0 | 0 |
| cD=0.2 | 149 | 9 | 13 | 0 | 0 |
| cD=0.25 | 157 | 6 | 11 | 0 | 0 |
| cD=0.3 | 180 | 13 | 17 | 0 | 0 |
| cD=0.35 | 184 | 10 | 19 | 0 | 0 |
| cD=0.4 | 223 | 20 | 35 | 0 | 0 |
| cD=0.45 | 264 | 26 | 46 | 0 | 0 |
| cD=0.5 | 282 | 32 | 63 | 0 | 0 |
| cD=0.55 | 348 | 331 | 485 | 47 | 58 |
| cD=0.6 | 186 | 561 | 681 | 1 | 1 |
| cD=0.65 | 120 | **3551** | **3312** | 63 | 46 |
| cD=0.7 | 110 | **50848** | **43879** | 1295 | 931 |
| cD=0.75 | 94 | **78982** | **66304** | 1635 | 1160 |
| cD=0.8 | 81 | 1516 | 2338 | 14 | 22 |
| cD=0.85 | 75 | 756 | 990 | 0 | 0 |
| cD=0.9 | 55 | 572 | 784 | 0 | 0 |
| cD=0.95 | 32 | 318 | 474 | 0 | 0 |

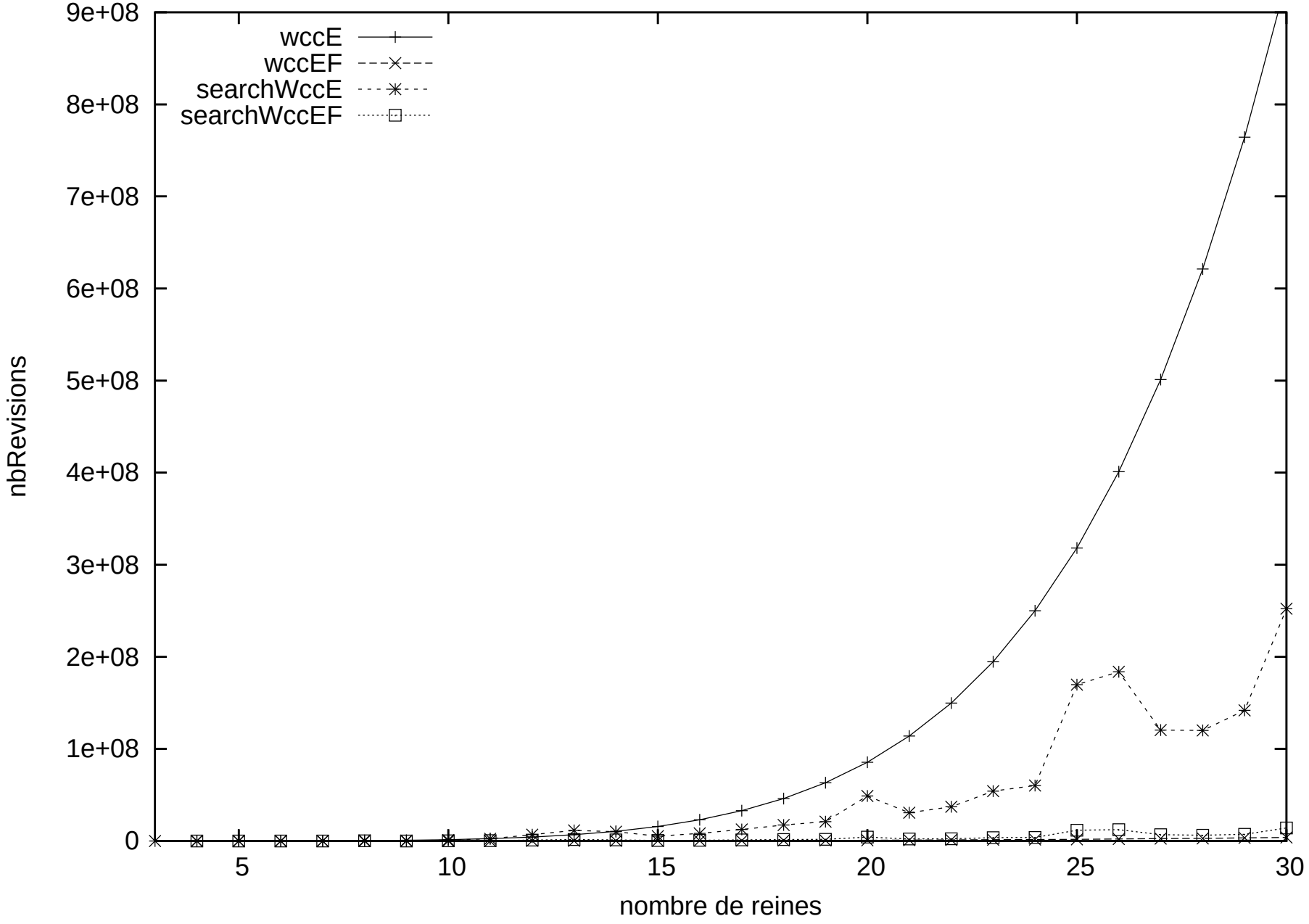
# cohérenceE vs cohérenceN ( $nTD = 0.75$ **avec** *triangle*)

| nTD=0.75 | $T_{ffc}$ | $T_E$ | $T_N$ | $BK_E$ | $BK_N$ |
|---|---|---|---|---|---|
| cD=0.05 | 117 | 10 | 13 | 0 | 0 |
| cD=0.1 | 125 | 8 | 12 | 0 | 0 |
| cD=0.15 | 132 | 8 | 12 | 0 | 0 |
| cD=0.2 | 143 | 9 | 13 | 0 | 0 |
| cD=0.25 | 149 | 5 | 10 | 0 | 0 |
| cD=0.3 | 163 | 8 | 15 | 0 | 0 |
| cD=0.35 | 181 | 9 | 19 | 0 | 0 |
| cD=0.4 | 216 | 19 | 37 | 0 | 0 |
| cD=0.45 | 258 | 27 | 46 | 0 | 0 |
| cD=0.5 | 286 | 34 | 62 | 0 | 0 |
| cD=0.55 | 329 | 72 | 115 | 0 | 0 |
| cD=0.6 | 175 | 543 | 650 | 0 | 0 |
| cD=0.65 | 110 | **3447** | **3179** | 79 | 58 |
| cD=0.7 | 91 | **49158** | **42696** | 1178 | 969 |
| cD=0.75 | 84 | 73930 | 87883 | 1493 | 1536 |
| cD=0.8 | 75 | 1308 | 1328 | 10 | 6 |
| cD=0.85 | 65 | 768 | 1007 | 0 | 0 |
| cD=0.9 | 48 | 567 | 775 | 0 | 0 |
| cD=0.95 | 22 | 314 | 465 | 0 | 0 |

# *consistentE* vs *consistentEF* (au temps)



# *consistentE* vs *consistentEF* (au nb de revisions)



# Conclusion et Perspectives

## Conclusion et Perspectives

- Nous avons donné la définition de ”contrainte éligible” et de ”contrainte gelée”.
- Nous avons proposé un nouvel algorithme, basé sur ces deux définitions, qui étend celui présenté par Nebel. Ce nouvel algorithme est correct et complet.
- Les expérimentations qu’on a faites sur des QCN d’Allen montrent l’intérêt pratique de ce nouvel algorithme.
- Regarder l’intérêt pratique de ces deux nouvelles notions avec d’autre méthode de propagation (*SAC* par exemple)
- Utiliser d’autre ensemble de *Split*
- Ces deux nouveaux algorithmes peuvent être parfois utilisés comme approximation.
- Faire des expérimentations sur des réseaux de constraintes d’autre algébre (RCC-5, RCC-8, ...)