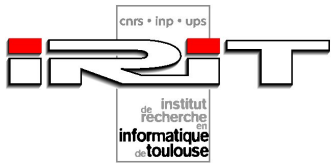
Compilation de CSP en Set-labeled Diagram

Alexandre Niveau Hélène Fargier Cédric Pralet



23 mai 2012

Exemple introductif

- Problème de configuration : T-shirts personnalisés.
  [Hadzic et al., 2008]

[Hadzic et al., 2008] Hadzic, T., Hansen, E., and B. O'Sullivan, B. (2008). On Automata, MDDs and BDDs in Constraint Satisfaction. In *ECAI Workshop on Inference methods based on Graphical Structures of Knowledge (WIGSK)*.

Exemple introductif

- Problème de configuration : T-shirts personnalisés.
  [Hadzic et al., 2008]
- paramètres :
  - image – "Men in Black" (MiB) ou "Sauvons les baleines" (Slb)
  - couleur – noir ou bleu
  - taille – S ou XL
  - manches – avec ou sans

[Hadzic et al., 2008] Hadzic, T., Hansen, E., and B. O'Sullivan, B. (2008). On Automata, MDDs and BDDs in Constraint Satisfaction. In *ECAI Workshop on Inference methods based on Graphical Structures of Knowledge (WIGSK)*.

Exemple introductif

- Problème de configuration : T-shirts personnalisés.
  [Hadzic et al., 2008]
- paramètres :
  - image – "Men in Black" (MiB) ou "Sauvons les baleines" (Slb)
  - couleur – noir ou bleu
  - taille – S ou XL
  - manches – avec ou sans
- règles :
  - les T-shirts « MiB » doivent être noirs
  - l'image « Slb » ne rentre pas sur les T-shirts de taille S
  - les T-shirts XL ont forcément des manches

[Hadzic et al., 2008] Hadzic, T., Hansen, E., and B. O'Sullivan, B. (2008). On Automata, MDDs and BDDs in Constraint Satisfaction. In *ECAI Workshop on Inference methods based on Graphical Structures of Knowledge (WIGSK)*.

Exemple introductif

- Problème de configuration : T-shirts personnalisés.
  [Hadzic et al., 2008]
- paramètres :
  - image – "Men in Black" (MiB) ou "Sauvons les baleines" (Slb)
  - couleur – noir ou bleu
  - taille – S ou XL
  - manches – avec ou sans
- règles :
  - les T-shirts « MiB » doivent être noirs
  - l'image « Slb » ne rentre pas sur les T-shirts de taille S
  - les T-shirts XL ont forcément des manches
- le programme doit préciser si les choix courants respectent les règles

---

[Hadzic et al., 2008] Hadzic, T., Hansen, E., and B. O'Sullivan, B. (2008). On Automata, MDDs and BDDs in Constraint Satisfaction. In *ECAI Workshop on Inference methods based on Graphical Structures of Knowledge (WIGSK)*.

Problème

- produit configurable $\rightarrow$ CSP
  - chaque variable correspond à un choix
  - chaque solution est une configuration possible

Problème

- produit configurable $\rightarrow$ CSP
  - chaque variable correspond à un choix
  - chaque solution est une configuration possible
- processus de configuration :
  - chaque choix fixe une valeur pour une variable donnée
  - le CSP est-il encore cohérent?

Problème

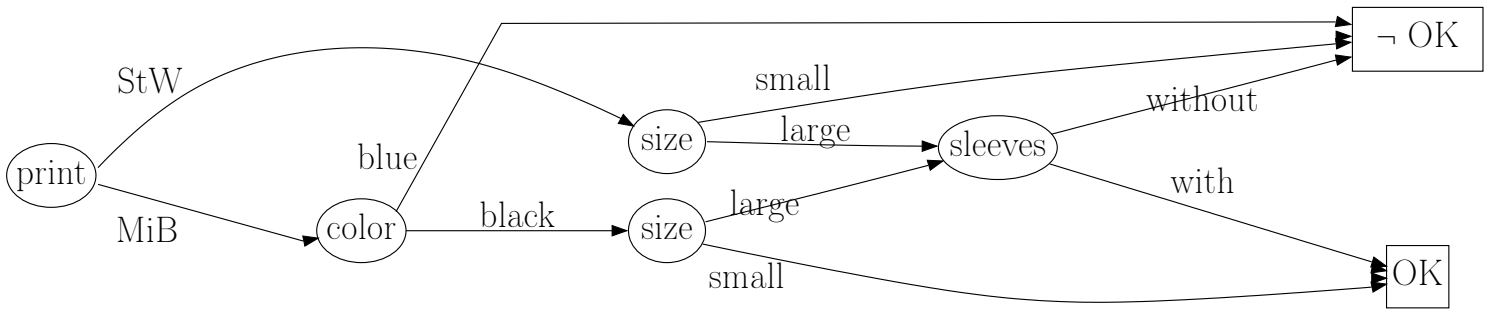
- produit configurable $\rightarrow$ CSP
  - chaque variable correspond à un choix
  - chaque solution est une configuration possible
- processus de configuration :
  - chaque choix fixe une valeur pour une variable donnée
  - le CSP est-il encore cohérent?
- problème NP-complet... or l'utilisateur ne veut pas attendre trop longtemps après chaque choix

Une solution: la compilation

- produit configurable $\rightarrow$ gros CSP qui ne varie jamais

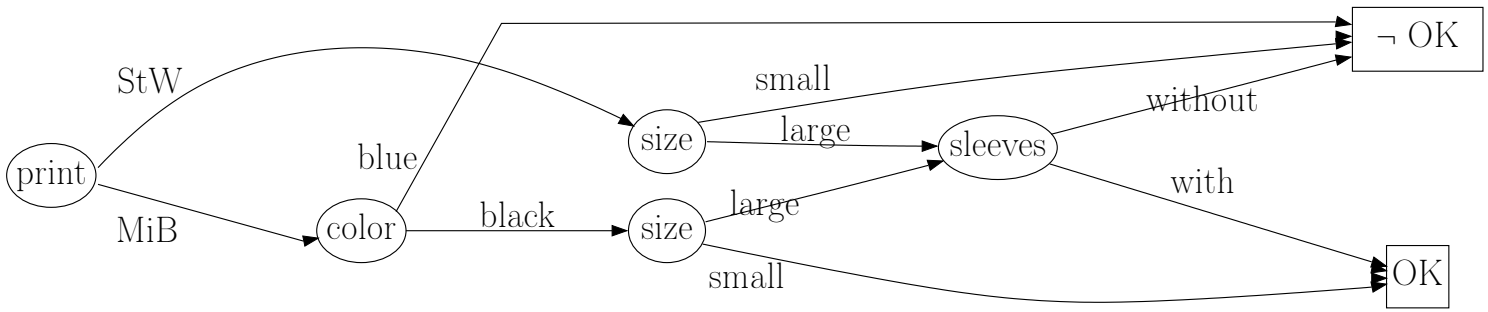
Une solution: la compilation

- produit configurable $\rightarrow$ gros CSP qui ne varie jamais
- compiler ce CSP en OBDD



Une solution: la compilation

- produit configurable → gros CSP qui ne varie jamais
- compiler ce CSP en OBDD

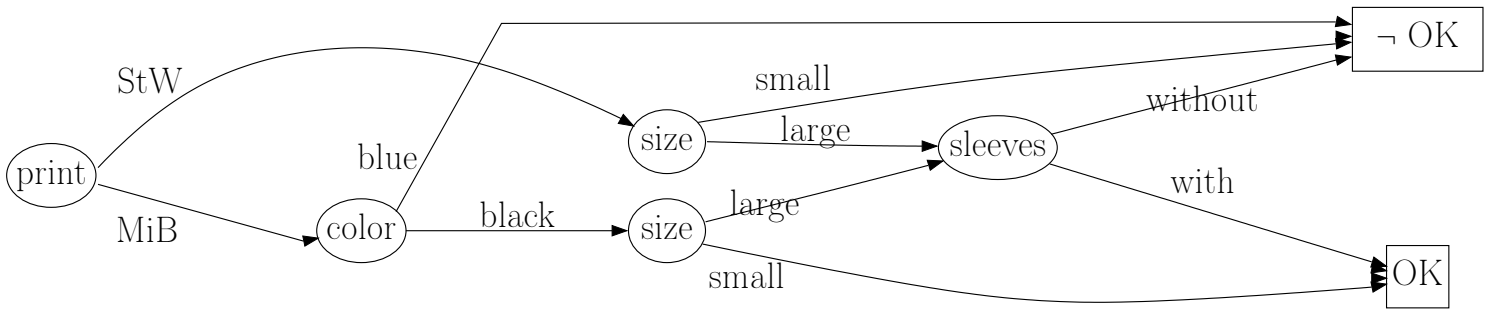


- l'assignation de valeurs aux variables (conditioning) et la cohérence sont polynomiales sur les OBDDs

→ l'attente de l'utilisateur est réduite

Une solution: la compilation

- produit configurable → gros CSP qui ne varie jamais
- compiler ce CSP en OBDD



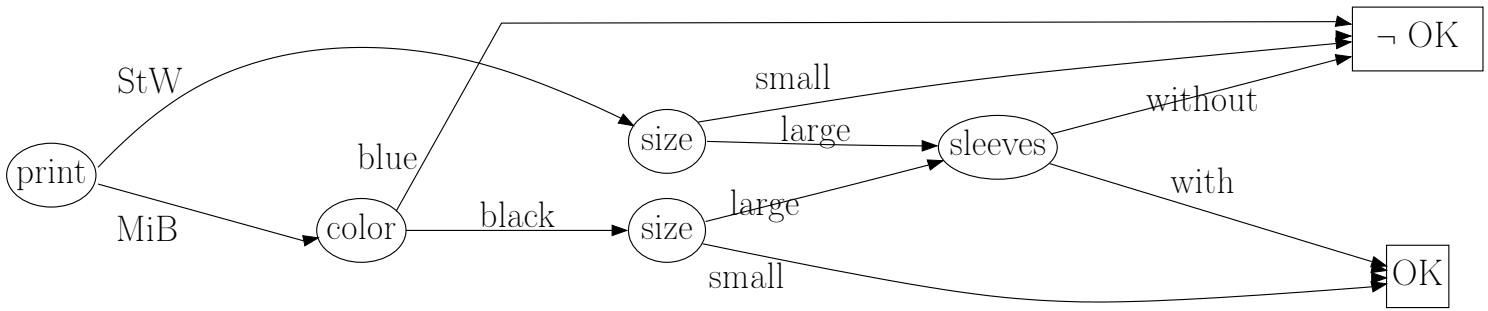
- l'assignation de valeurs aux variables (conditioning) et la cohérence sont polynomiales sur les OBDDs

→ l'attente de l'utilisateur est réduite

- compilation de connaissances : traduction hors ligne de la partie fixe d'un problème (prétraitement, potentiellement difficile) pour rendre les opérations en ligne polynomiales

Multivalued Decision Diagrams

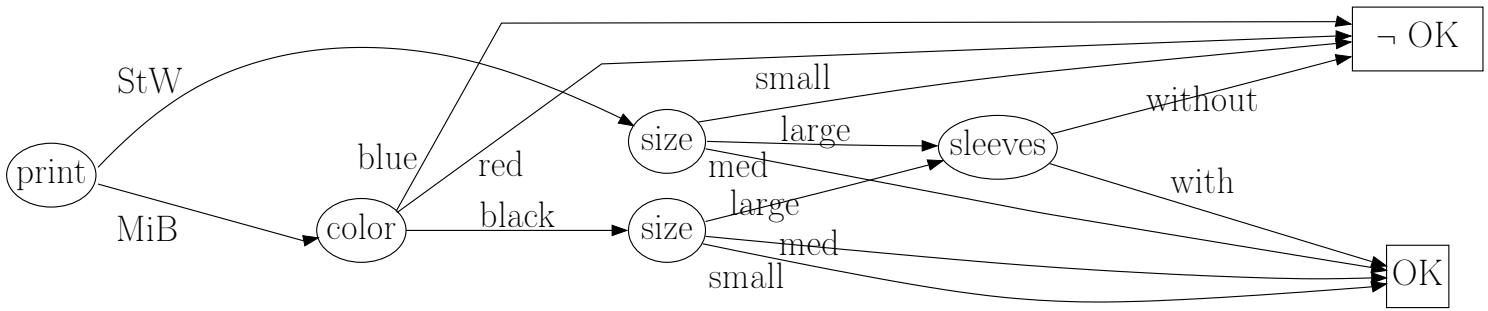
- MDDs : généralisation directe des OBDDs aux variables multivaluées [Srinivasan et al., 1990]



[Srinivasan et al., 1990] Srinivasan, A., Kam, T., Malik, S., and Brayton, R. K. (1990). Algorithms for Discrete Function Manipulation. In *ICCAD*, pages 92–95.

Multivalued Decision Diagrams

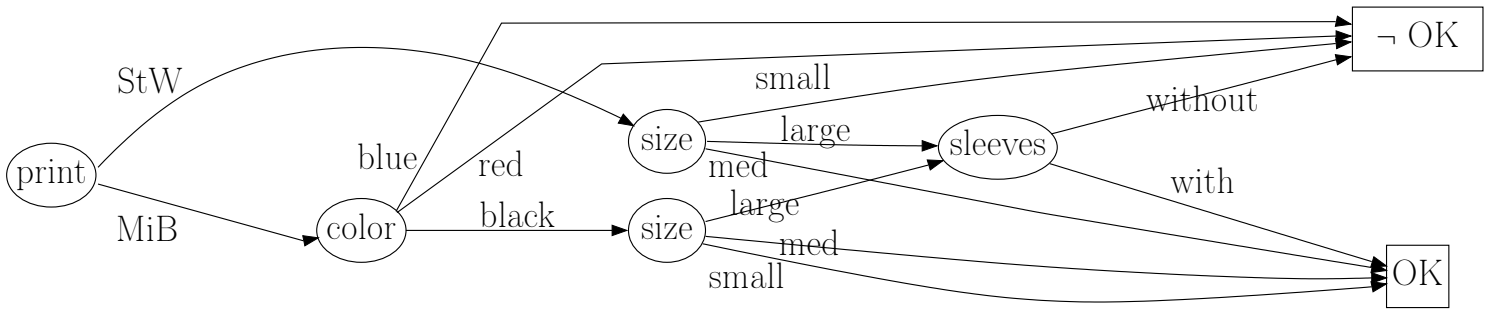
- MDDs : généralisation directe des OBDDs aux variables multivaluées [Srinivasan et al., 1990]



[Srinivasan et al., 1990] Srinivasan, A., Kam, T., Malik, S., and Brayton, R. K. (1990). Algorithms for Discrete Function Manipulation. In *ICCAD*, pages 92–95.

Multivalued Decision Diagrams

- MDDs : généralisation directe des OBDDs aux variables multivaluées [Srinivasan et al., 1990]

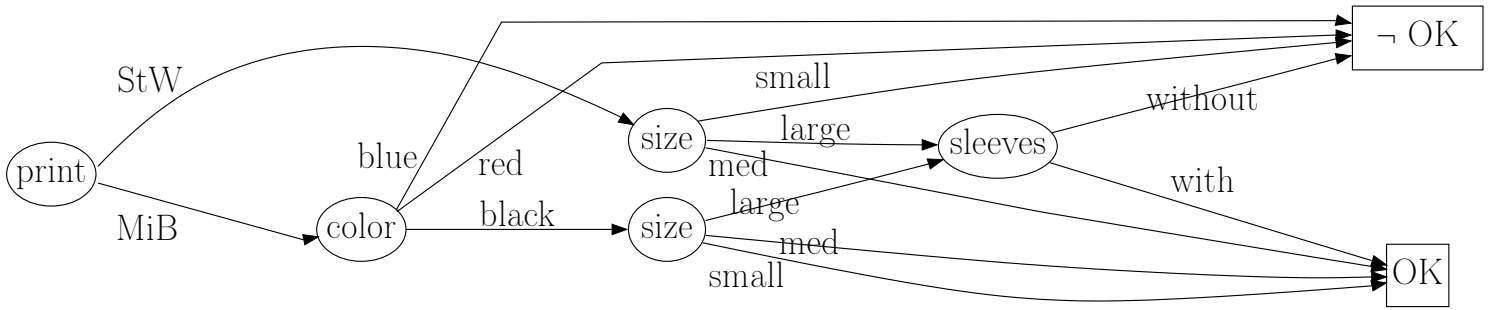


- fortes exigences sur les MDDs : déterminisme, read-once, ordonnancement

[Srinivasan et al., 1990] Srinivasan, A., Kam, T., Malik, S., and Brayton, R. K. (1990). Algorithms for Discrete Function Manipulation. In *ICCAD*, pages 92–95.

Multivalued Decision Diagrams

- MDDs : généralisation directe des OBDDs aux variables multivaluées [Srinivasan et al., 1990]



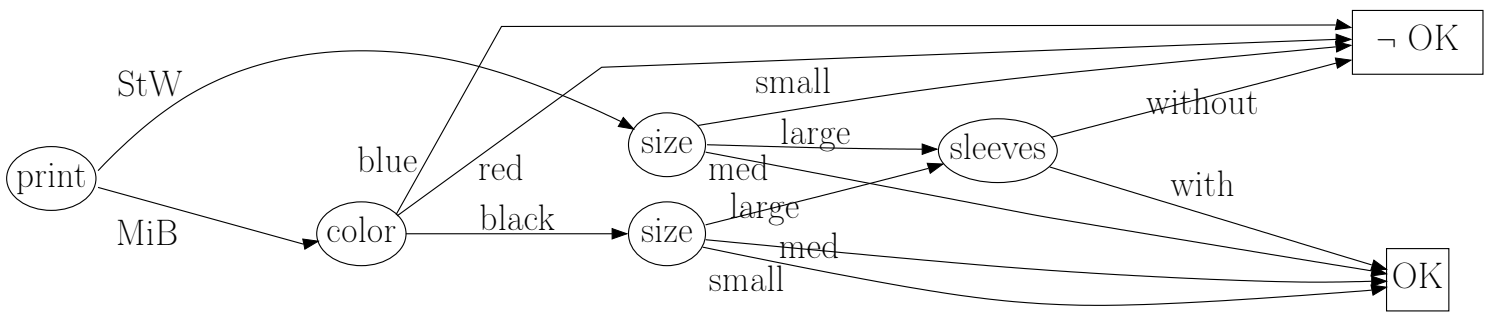
- fortes exigences sur les MDDs : déterminisme, read-once, ordonnancement
- perte en compacité
  or exigences pas toujours nécessaires

→ on utilise une structure plus générale

[Srinivasan et al., 1990] Srinivasan, A., Kam, T., Malik, S., and Brayton, R. K. (1990). Algorithms for Discrete Function Manipulation. In *ICCAD*, pages 92–95.

Set-labeled Diagrams

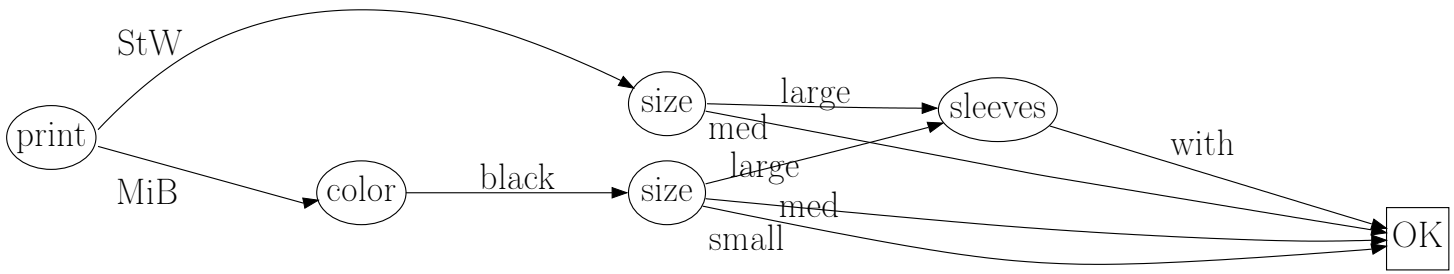
- Set-labeled Diagrams (SDs) : non-déterministes, non ordonnés, arcs étiquetés par un ensemble de valeurs [Niveau et al., 2011]



[Niveau et al., 2011] Niveau, A., Fargier, H., and Pralet, C. (2011). Representing CSPs with set-labeled diagrams: A compilation map. In *Proc. of the 2nd International Workshop on Graph Structures for Knowledge Representation and Reasoning (GKR)*.

Set-labeled Diagrams

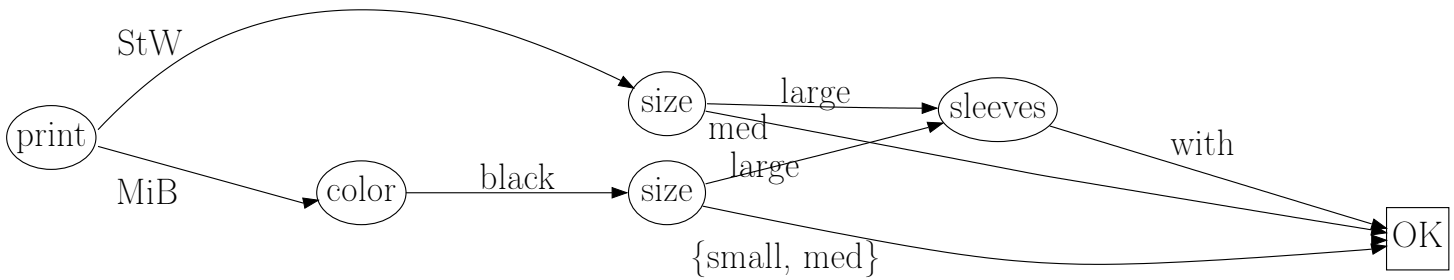
- Set-labeled Diagrams (SDs) : non-déterministes, non ordonnés, arcs étiquetés par un ensemble de valeurs [Niveau et al., 2011]



[Niveau et al., 2011] Niveau, A., Fargier, H., and Pralet, C. (2011). Representing CSPs with set-labeled diagrams: A compilation map. In *Proc. of the 2nd International Workshop on Graph Structures for Knowledge Representation and Reasoning (GKR)*.

Set-labeled Diagrams

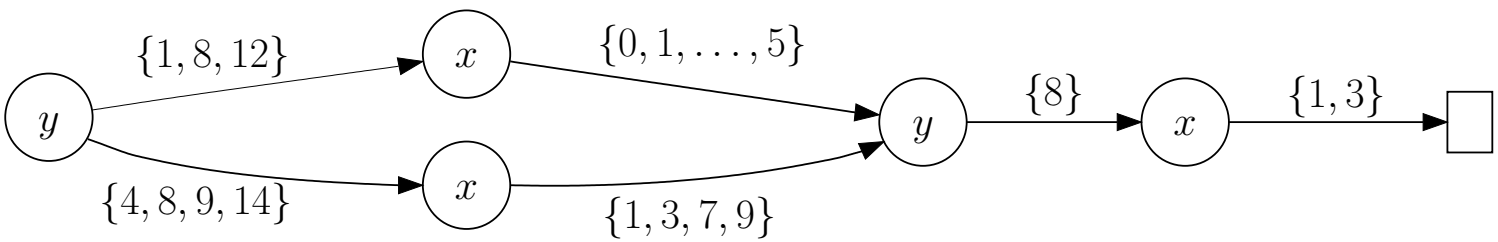
- Set-labeled Diagrams (SDs) : non-déterministes, non ordonnés, arcs étiquetés par un ensemble de valeurs [Niveau et al., 2011]



[Niveau et al., 2011] Niveau, A., Fargier, H., and Pralet, C. (2011). Representing CSPs with set-labeled diagrams: A compilation map. In *Proc. of the 2nd International Workshop on Graph Structures for Knowledge Representation and Reasoning (GKR)*.

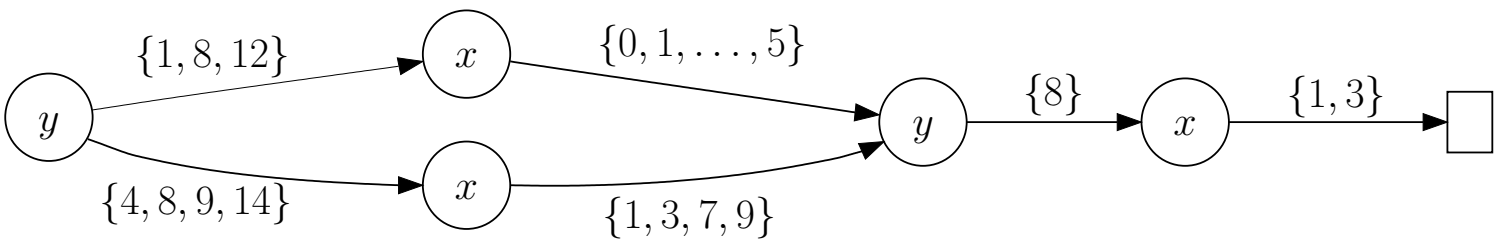
Famille des SDs : convergence

- divers types de SD, plus ou moins adaptés à diverses requêtes
- FSDs : SDs convergents (focusing).



Famille des SDs : convergence

- divers types de SD, plus ou moins adaptés à diverses requêtes
- FSDs : SDs convergents (focusing).



→ requêtes de satisfaisabilité et de conditioning polynomiales sur les FSDs

Famille des SDs

- dFSDs : FSDs déterministes.

  $\rightarrow$ perte en compacité, mais requête de validité polynomiale

Famille des SDs

- dFSDs : FSDs déterministes.
  $\rightarrow$ perte en compacité, mais requête de validité polynomiale

- dRSDs : dFSDs read-once.
  $\rightarrow$ négation et comptage de modèles polynomiaux

Famille des SDs

- dFSDs : FSDs déterministes.
  $\rightarrow$ perte en compacité, mais requête de validité polynomiale

- dRSDs : dFSDs read-once.
  $\rightarrow$ négation et comptage de modèles polynomiaux

- MDDs : dRSDs ordonnés.
  $\rightarrow$ perte en compacité, mais équivalence, conjonction et disjonction polynomiales

Compiler un CSP en SD

- méthode classique : bottom-up compilation, combinaison de SDs représentant des contraintes élémentaires
- inconvénient : graphes intermédiaires, pouvant être exponentiellement plus gros que le graphe final
- une autre idée est de construire des SDs en suivant l'arbre de recherche d'un solveur, à la « DPLL with a trace »
  [Huang and Darwiche, 2005]

[Huang and Darwiche, 2005] Huang, J. and Darwiche, A. (2005). DPLL with a Trace: From SAT to Knowledge Compilation. In *IJCAI*, pages 156–162.

CHOCO with a trace

Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :

SD :

CHOCO with a trace

Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

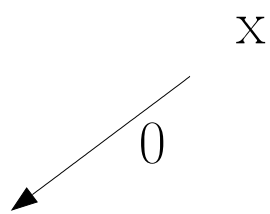
Arbre de recherche :

SD :

CHOCO with a trace

Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :

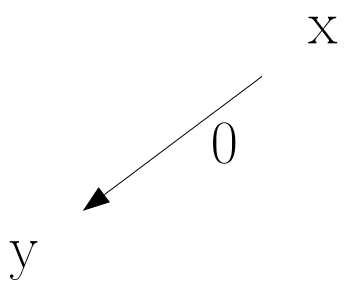


SD :

CHOCO with a trace

Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :

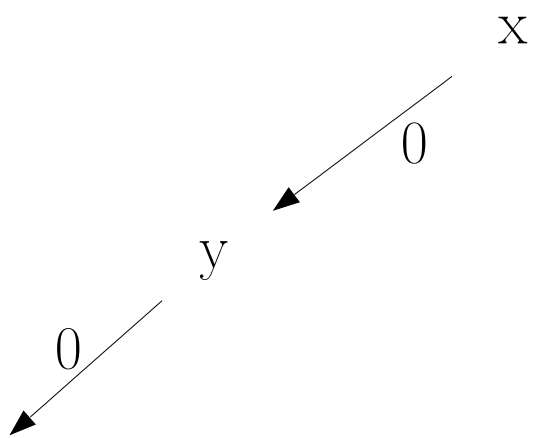


SD :

CHOCO with a trace

Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :

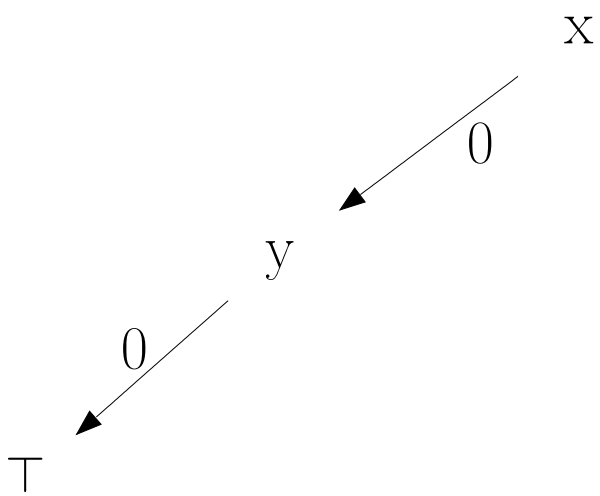


SD :

CHOCO with a trace

Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :



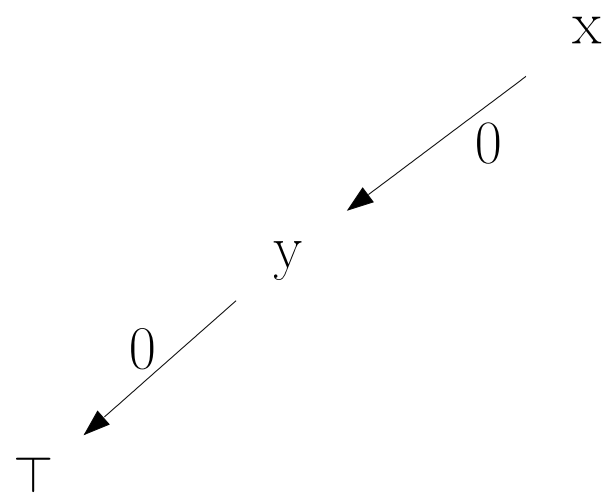
SD :

une solution a été trouvée

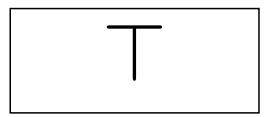
CHOCO with a trace

Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :



SD :

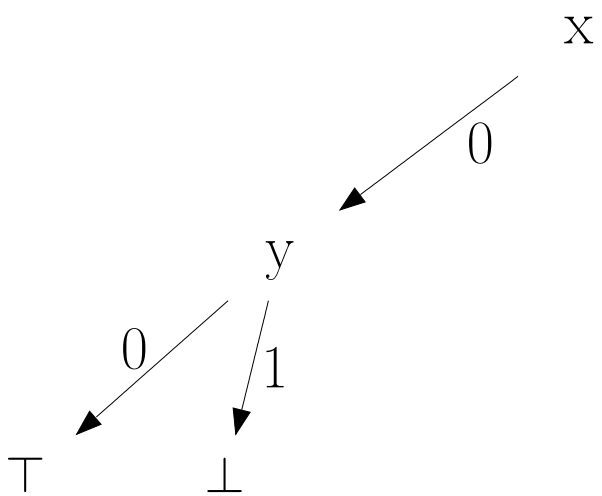


une solution a été trouvée
$\rightarrow$ on ajoute le nœud puits au SD

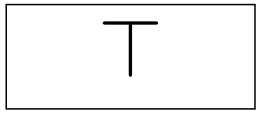
CHOCO with a trace

Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :



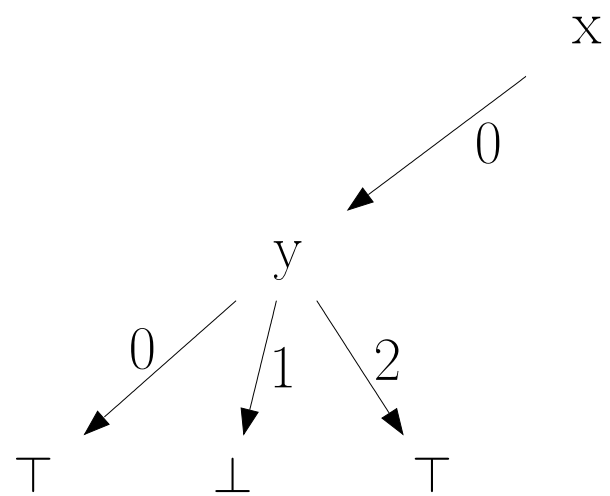
SD :



CHOCO with a trace

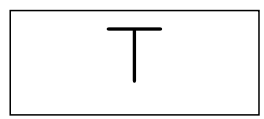
Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :



fermeture du nœud y

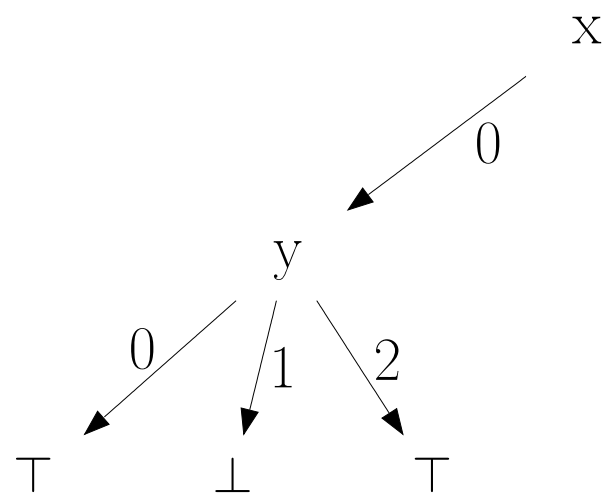
SD :



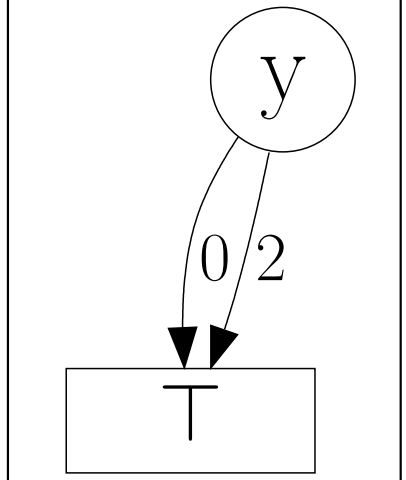
CHOCO with a trace

Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :



SD :

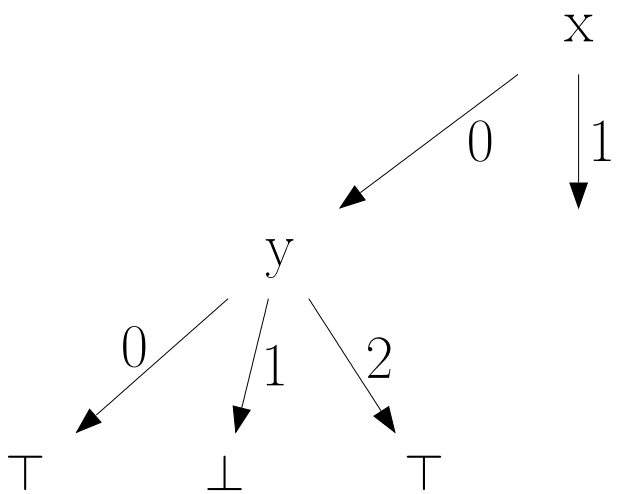


fermeture du nœud y
$\rightarrow$ on l'ajoute au SD

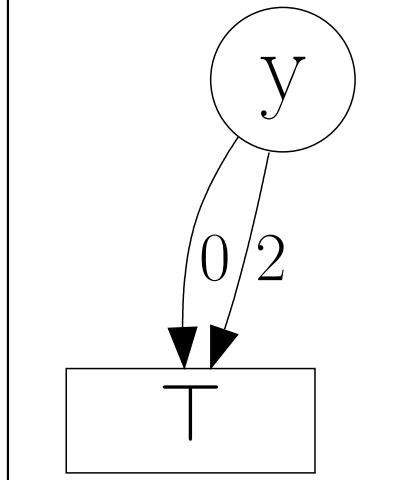
CHOCO with a trace

Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :



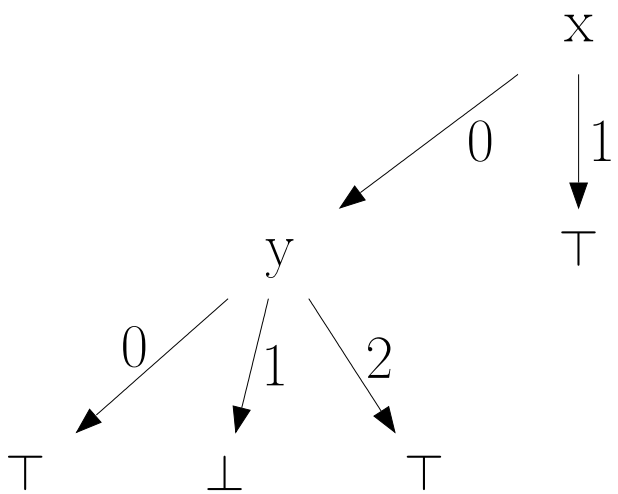
SD :



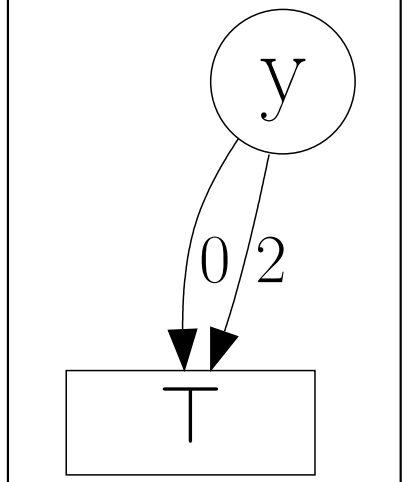
CHOCO with a trace

Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :



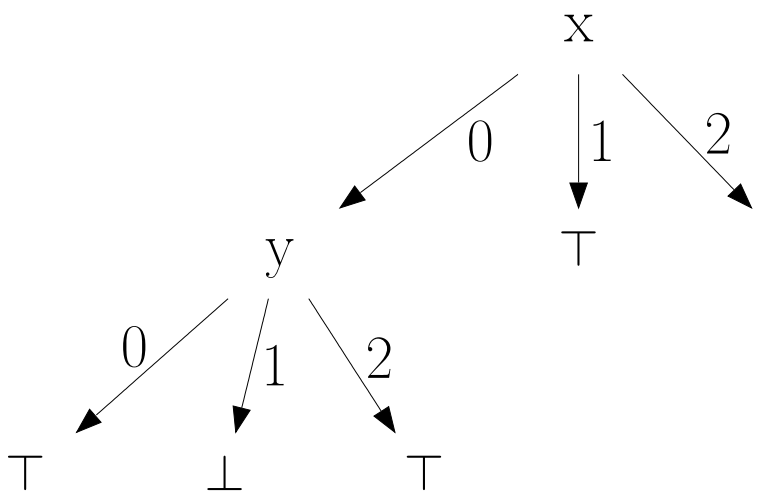
SD :



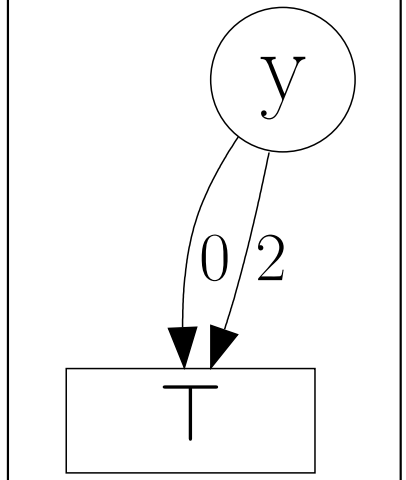
CHOCO with a trace

Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :



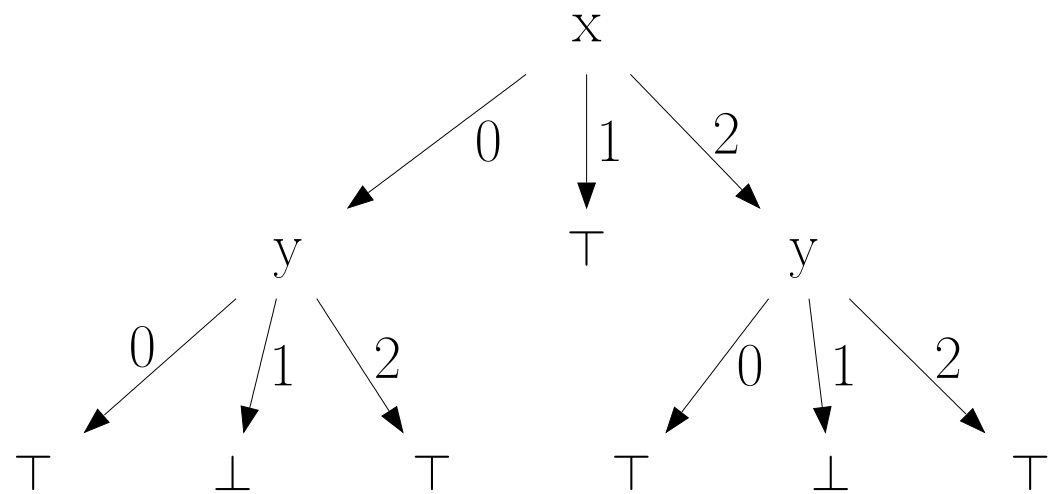
SD :



CHOCO with a trace

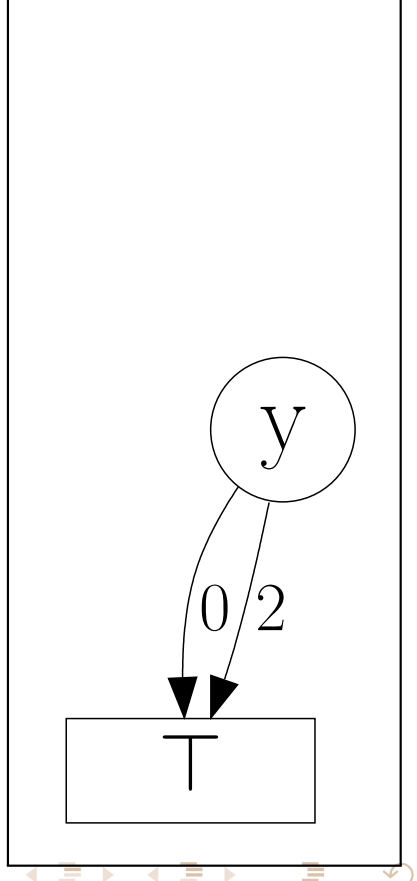
Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :



fermeture du nœud y

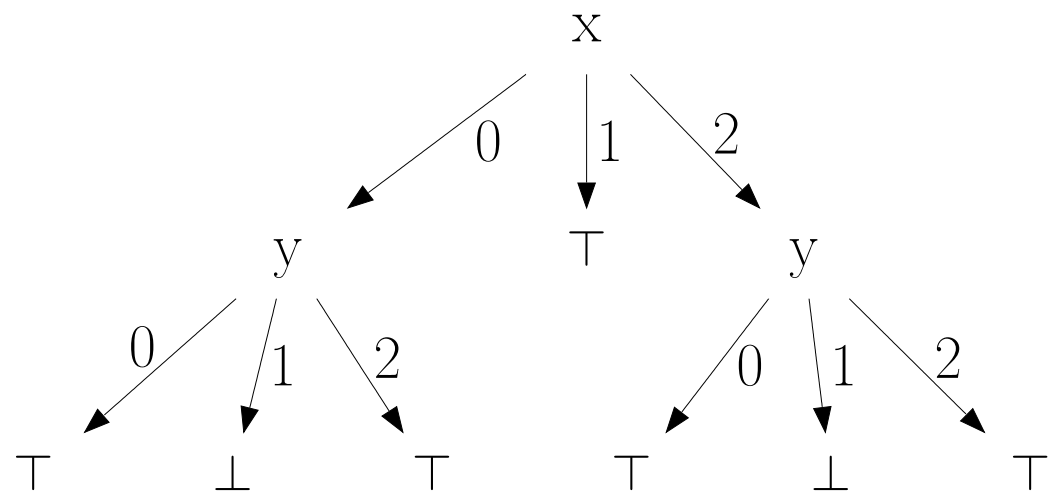
SD :



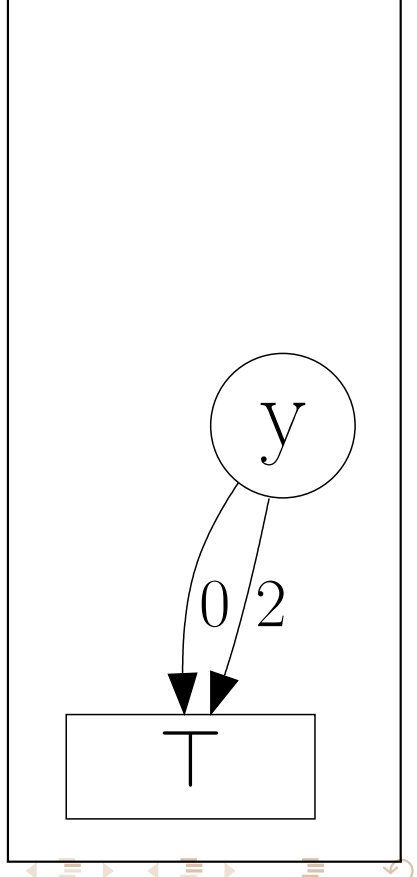
CHOCO with a trace

Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :



SD :

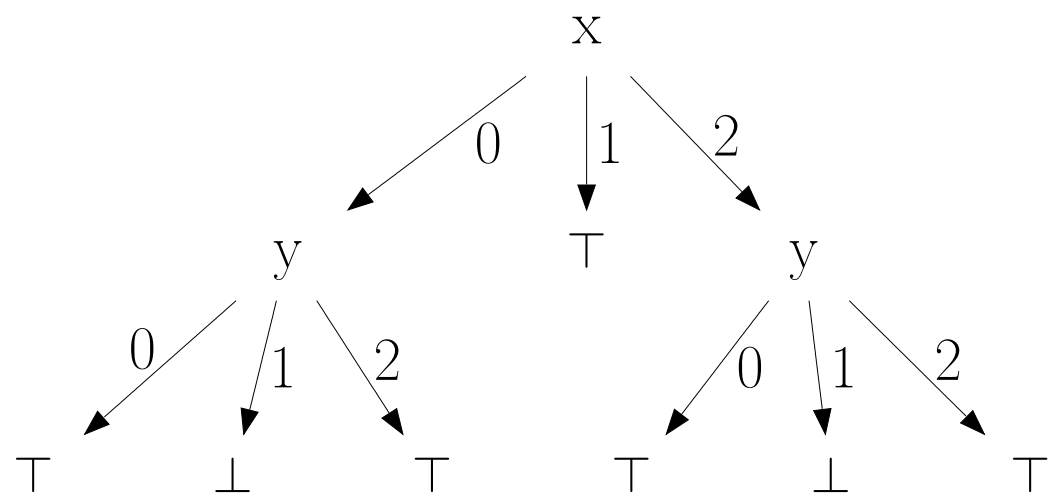


fermeture du nœud y
$\rightarrow$ un nœud équivalent existe déjà dans le SD, on ne l'ajoute pas

CHOCO with a trace

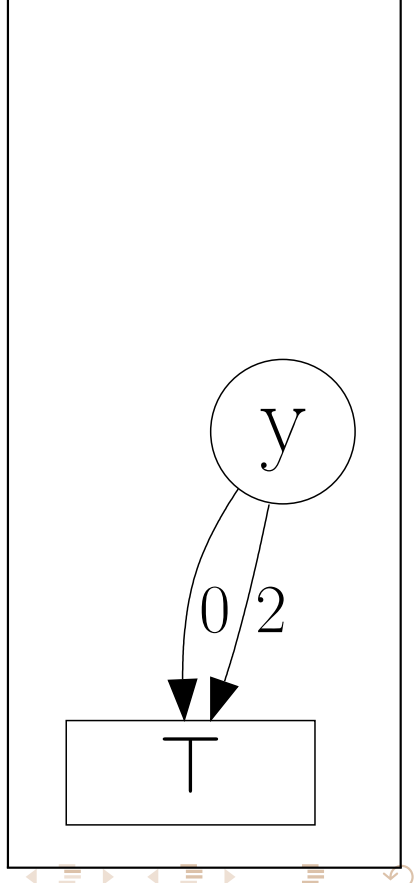
Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :



fermeture du nœud x

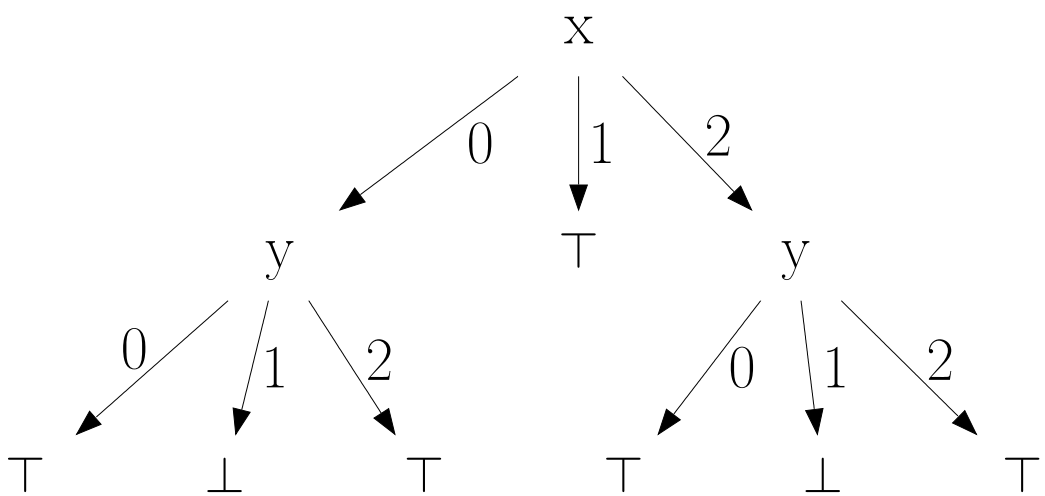
SD :



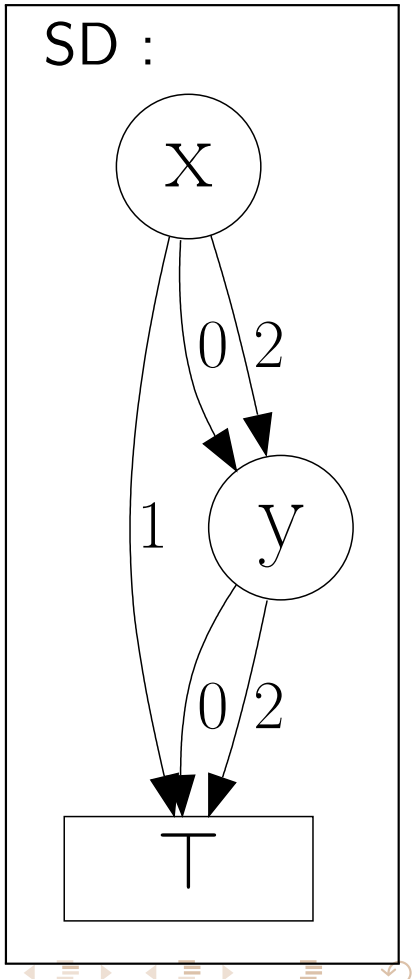
CHOCO with a trace

Déroulement de la compilation de $x = y \vee y \neq 1$, pour x et y de domaine $\{0, 1, 2\}$

Arbre de recherche :



SD :



fermeture du nœud x
$\rightarrow$ on l'ajoute au SD

Caching

- table de nœuds uniques : le compilateur ne crée jamais deux sous-graphes identiques (« isomorphes »)
  $\rightarrow$ le SD résultant peut être exponentiellement plus compact que l'arbre de recherche

- mais le solveur a bien parcouru l'arbre en entier...

[Lecoutre et al., 2007] Lecoutre, C., Sais, L., Tabary, S., and Vidal, V. (2007). Transposition tables for constraint satisfaction. In *AAAI*, pages 243–248.

Caching

- table de nœuds uniques : le compilateur ne crée jamais deux sous-graphes identiques (« isomorphes »)
  $\rightarrow$ le SD résultant peut être exponentiellement plus compact que l'arbre de recherche

- mais le solveur a bien parcouru l'arbre en entier...

$\rightarrow$ caching : table de hachage des sous-problèmes rencontrés

- clé de hachage : domaines des variables non-assignées ou intervenant dans des contraintes non satisfaites
  [Lecoutre et al., 2007]

[Lecoutre et al., 2007] Lecoutre, C., Sais, L., Tabary, S., and Vidal, V. (2007). Transposition tables for constraint satisfaction. In *AAAI*, pages 243–248.

Forme compilée obtenue

- les SDs obtenus sont toujours convergents et déterministes, mais :
  - ils sont read-once si le solveur branche sur des singletons
  - ils sont ordonnés si l'ordre dans lequel les variables sont considérées est statique

$\rightarrow$ on peut obtenir des dFSDs, des dRSDs ou des MDDs

Ordre dynamique

- possibilité d'utiliser des heuristiques de choix de variable
- comment se comportent les heuristiques classiques ?
- quelles heuristiques pour une forme compilée compacte ?

Heuristiques dépendant du CSP

- heuristiques d'ordonnancement basées sur le graphe de contraintes [Amilhastre, 1999]

- pour un ordre partiel courant $\langle x_1, x_2, \ldots, x_k \rangle$
  - **HBW** : choisit en priorité une voisine de x_1, puis une voisine de x_2, etc.

  - **HSBW** : choisit une voisine des variables les plus anciennement choisies

  - **MCSInv** : choisit la variable la plus liée à celles déjà choisies

[Amilhastre, 1999] Amilhastre, J. (1999). *Représentation par automate d'ensemble de solutions de problèmes de satisfaction de contraintes*. PhD thesis, Université Montpellier II.

Heuristique dépendant du solveur

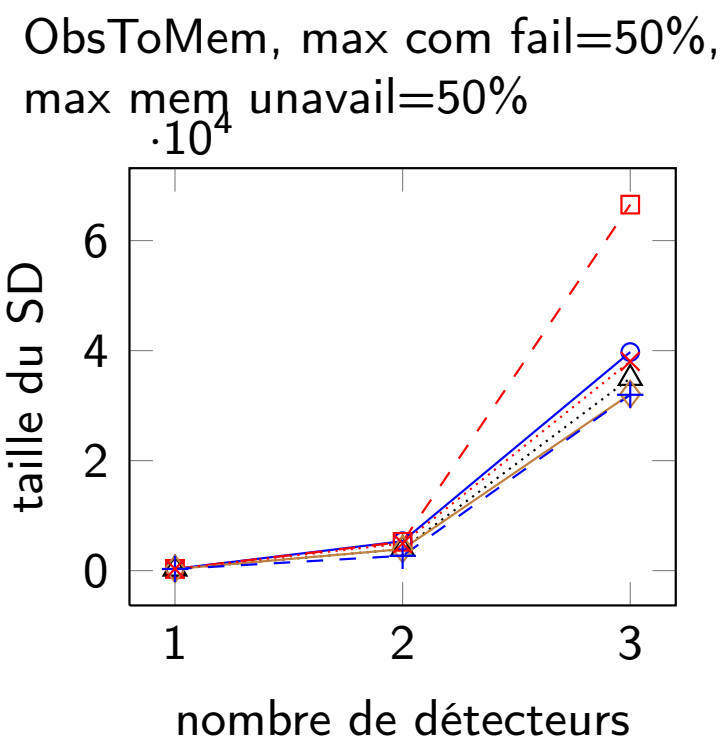
- heuristique basée sur le cache : **MaxHashUse**
- but : limiter le nombre de nouveaux nœuds
- on regarde le nombre de nœuds ouverts pour chaque choix de variable possible
- on choisit celle qui en ouvre le moins

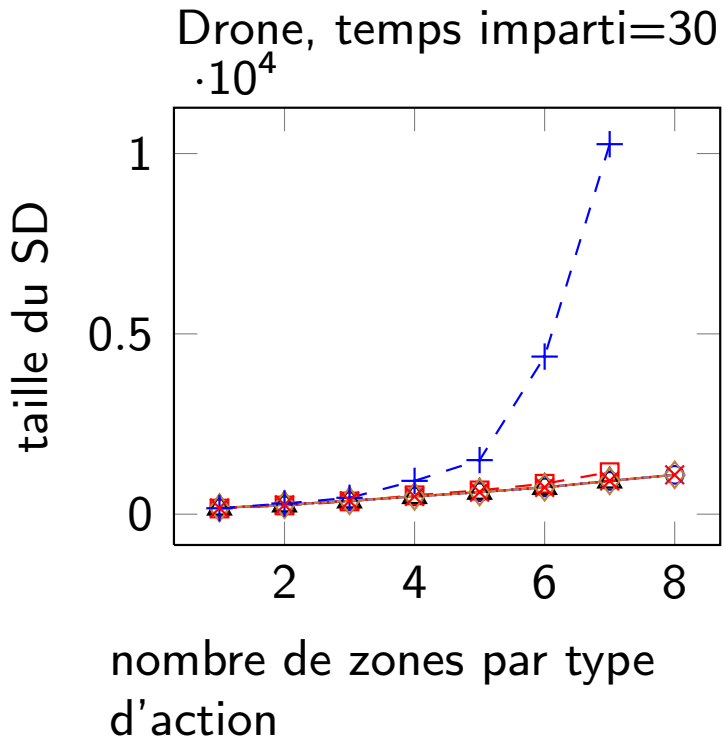
Benchmarks utilisés

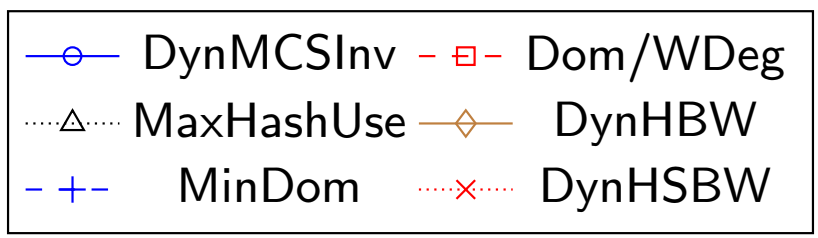
- *Drone* : planification de mission pour un microdrone lors d'une compétition (divers objectifs à remplir)
  $\rightarrow$ compilation d'une table de transition

- *ObsToMem* : synthèse de contrôleur, maintenance de liaisons entre les instruments de détection et la mémoire de masse d'un satellite d'observation
  $\rightarrow$ compilation d'une politique de décision

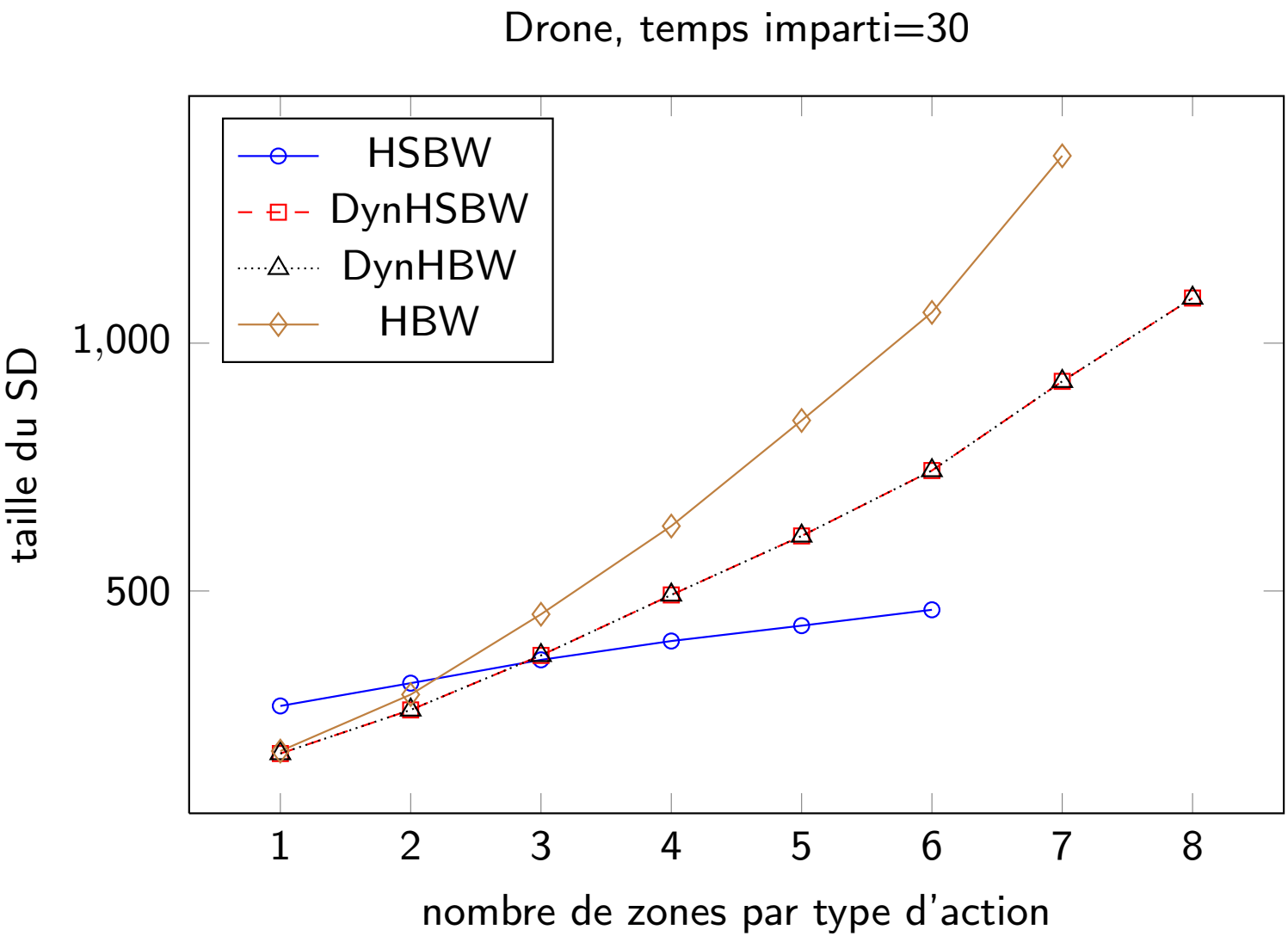
Comparaison des heuristiques



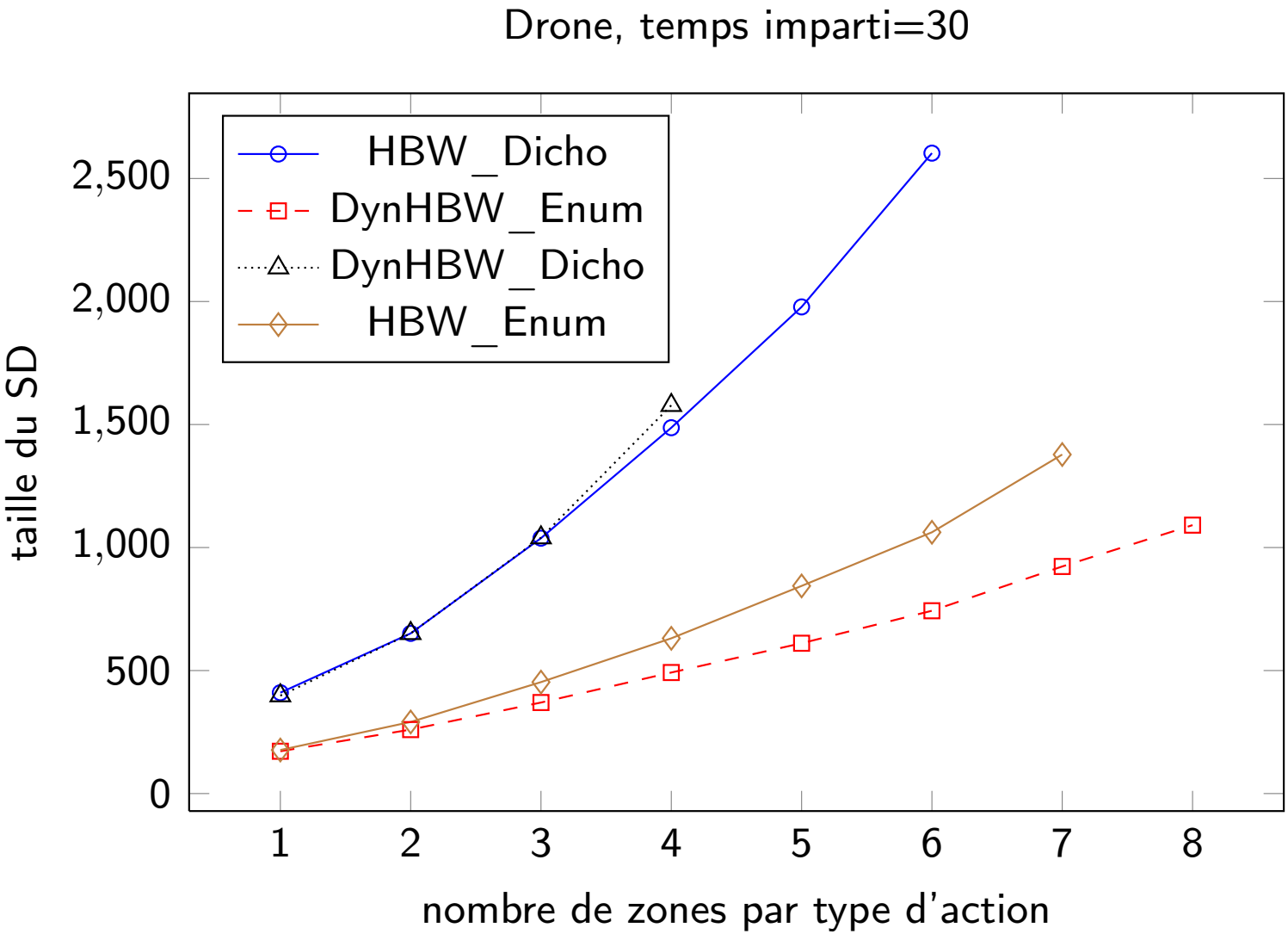




Dynamique vs statique



Fonctions de branchement



Conclusion et perspectives

- algorithme « CHOCO with a trace » : compile des CSPs en diverses variantes de MDD
- limites : compilateur intrinsèquement déterministe
- début de l'étude d'heuristiques de compilation
- Perspectives :
  - heuristiques exploitant le cache
  - comment relâcher le read-once efficacement ?
  - ajout de nœuds « ET » aux SDs