Model checking avec « edge-valued decision diagrams »

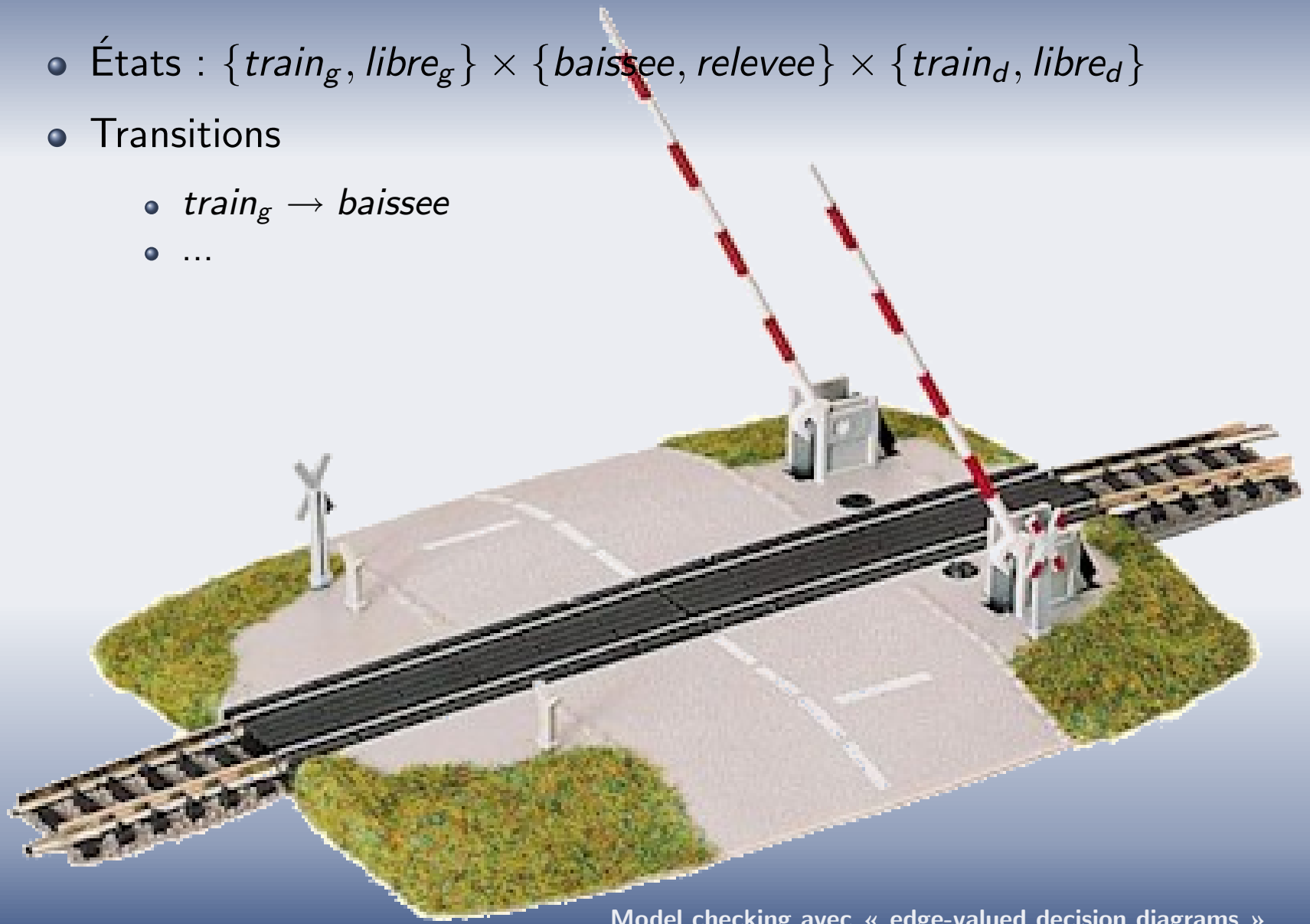
Pierre Roux

16 mars 2010

Stage effectué du 26 janvier au 16 avril 2009 sous la direction de Radu Siminiceanu
au National Institute of Aerospace, Hampton, Virginie.

Système de transitions

- États : $\{train_g, libre_g\} \times \{baissee, relevee\} \times \{train_d, libre_d\}$
- Transitions
  - $train_g \longrightarrow baissee$
  - ...



Representation de fonctions

En model-checking on doit représenter

- fonctions booléennes $f : \{0,1\}^n \to \{0,1\}$
- ensembles $\left\{x \in \{0,1\}^n \mid f(x) = 1\right\}$

Representation de fonctions

En model-checking on doit représenter

- fonctions booléennes $f : \{0,1\}^n \to \{0,1\}$
- ensembles $\left\{x \in \{0,1\}^n \mid f(x) = 1\right\}$

Première idée : table de vérité

a	b	c	$f(a,b,c)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Representation de fonctions

En model-checking on doit représenter

- fonctions booléennes $f : \{0,1\}^n \to \{0,1\}$
- ensembles $\left\{x \in \{0,1\}^n \mid f(x) = 1\right\}$

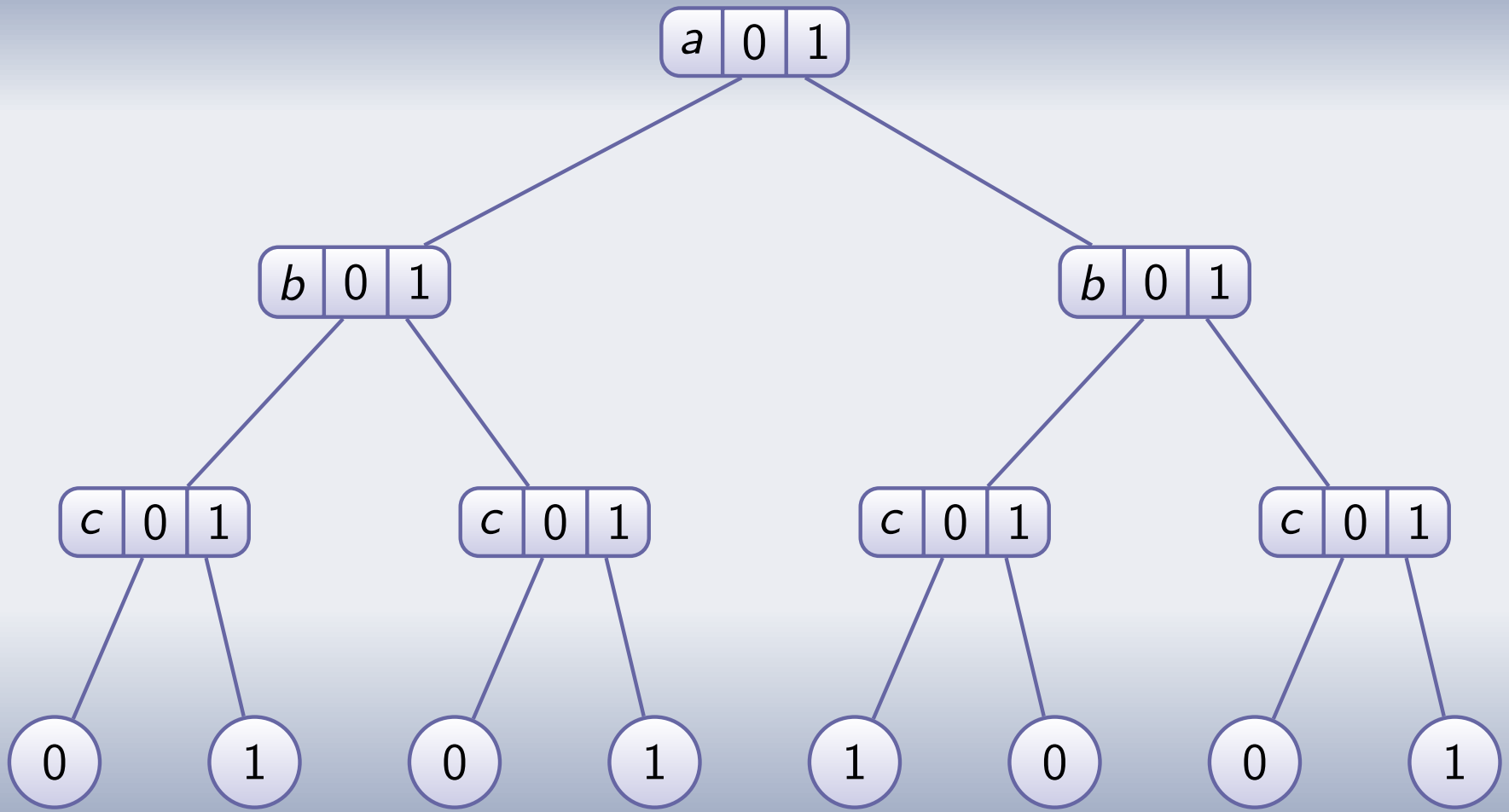
Première idée : table de vérité

a	b	c	$f(a,b,c)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

2^n entrées

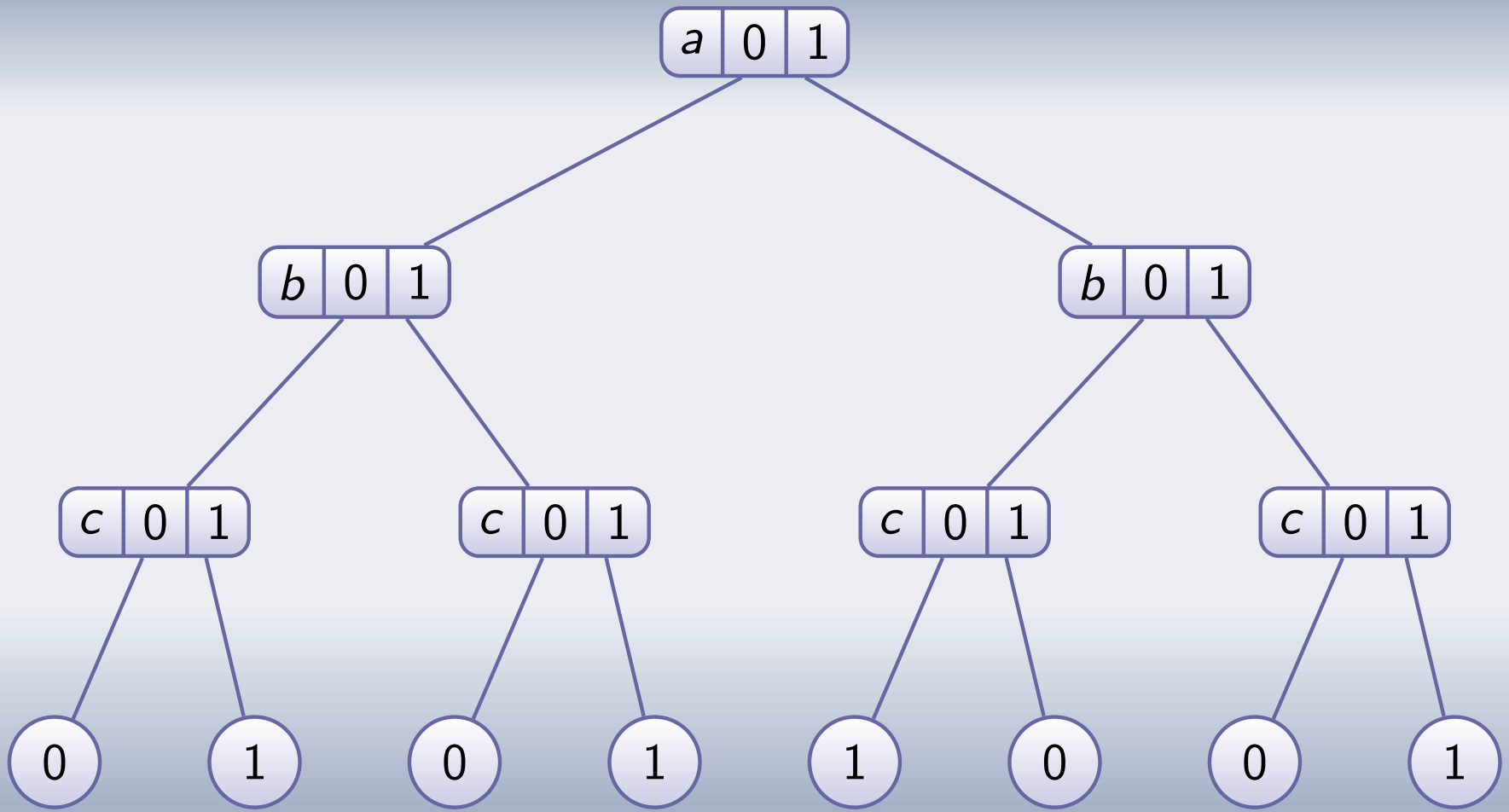
Arbres

Deuxième idée : utiliser des arbres



Arbres

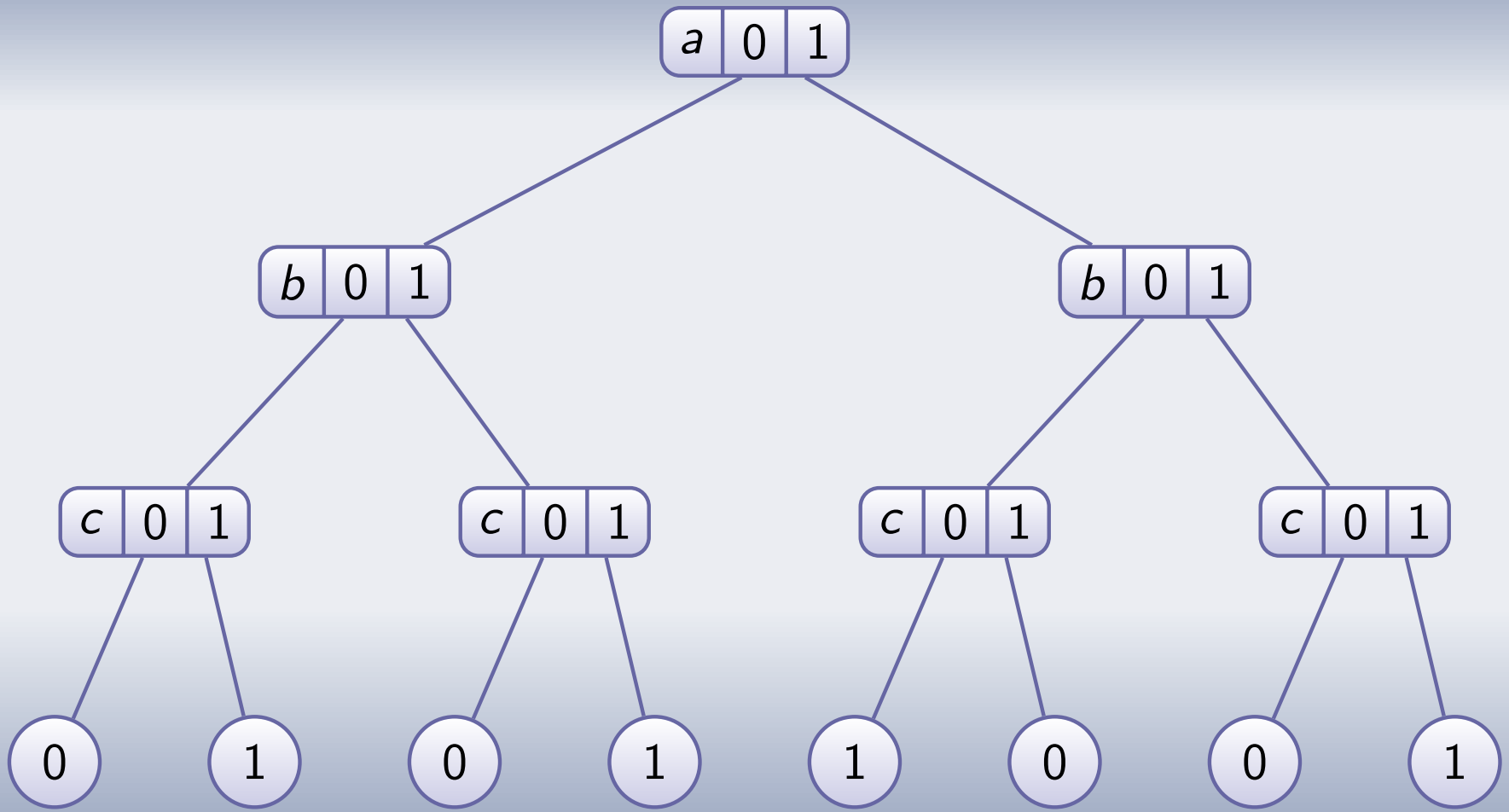
Deuxième idée : utiliser des arbres



encore 2^n feuilles

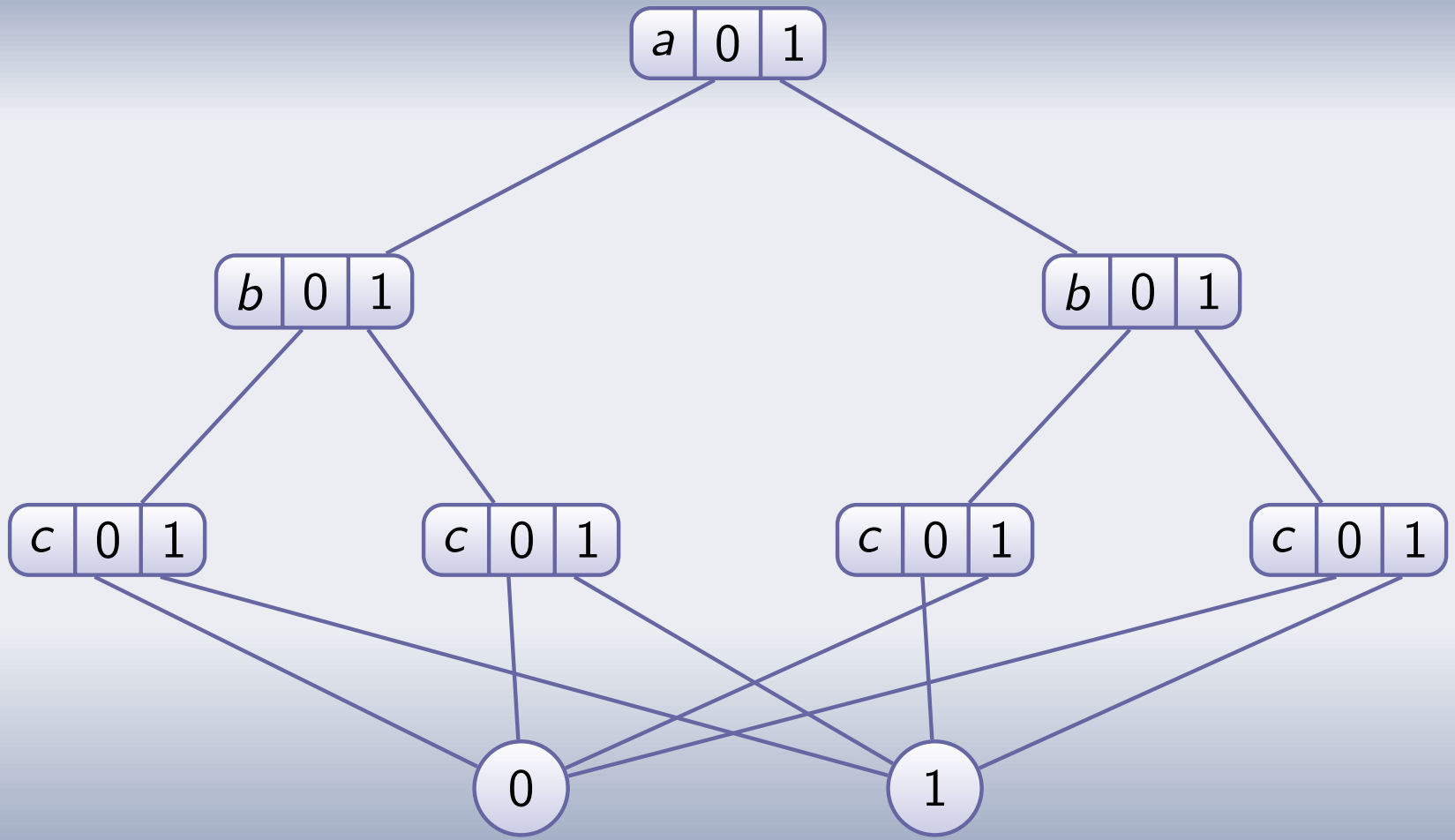
Diagrammes de décision binaires (BDD)

Fusionner les feuilles



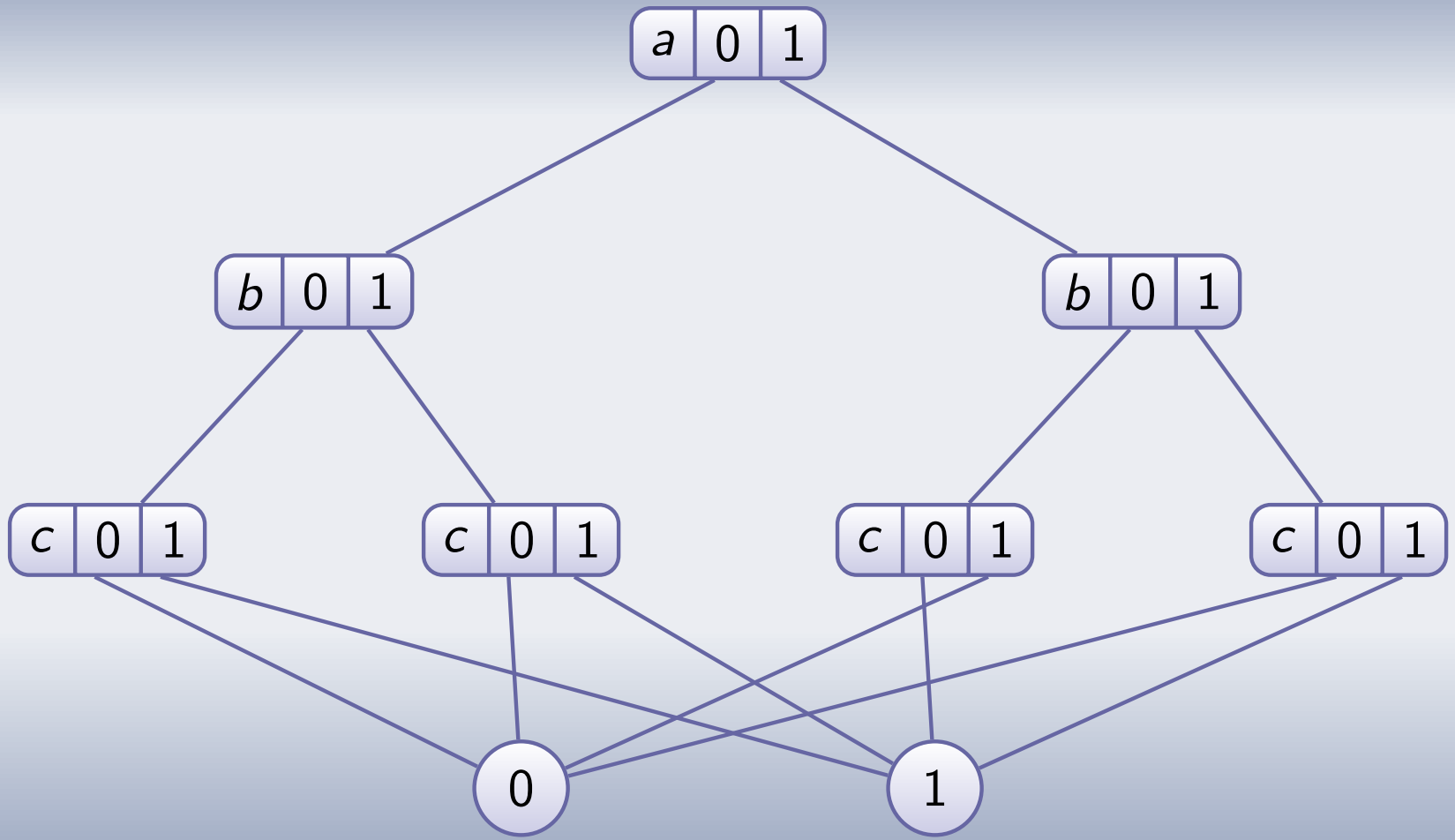
Diagrammes de décision binaires (BDD)

Fusionner les feuilles



Diagrammes de décision binaires (BDD)

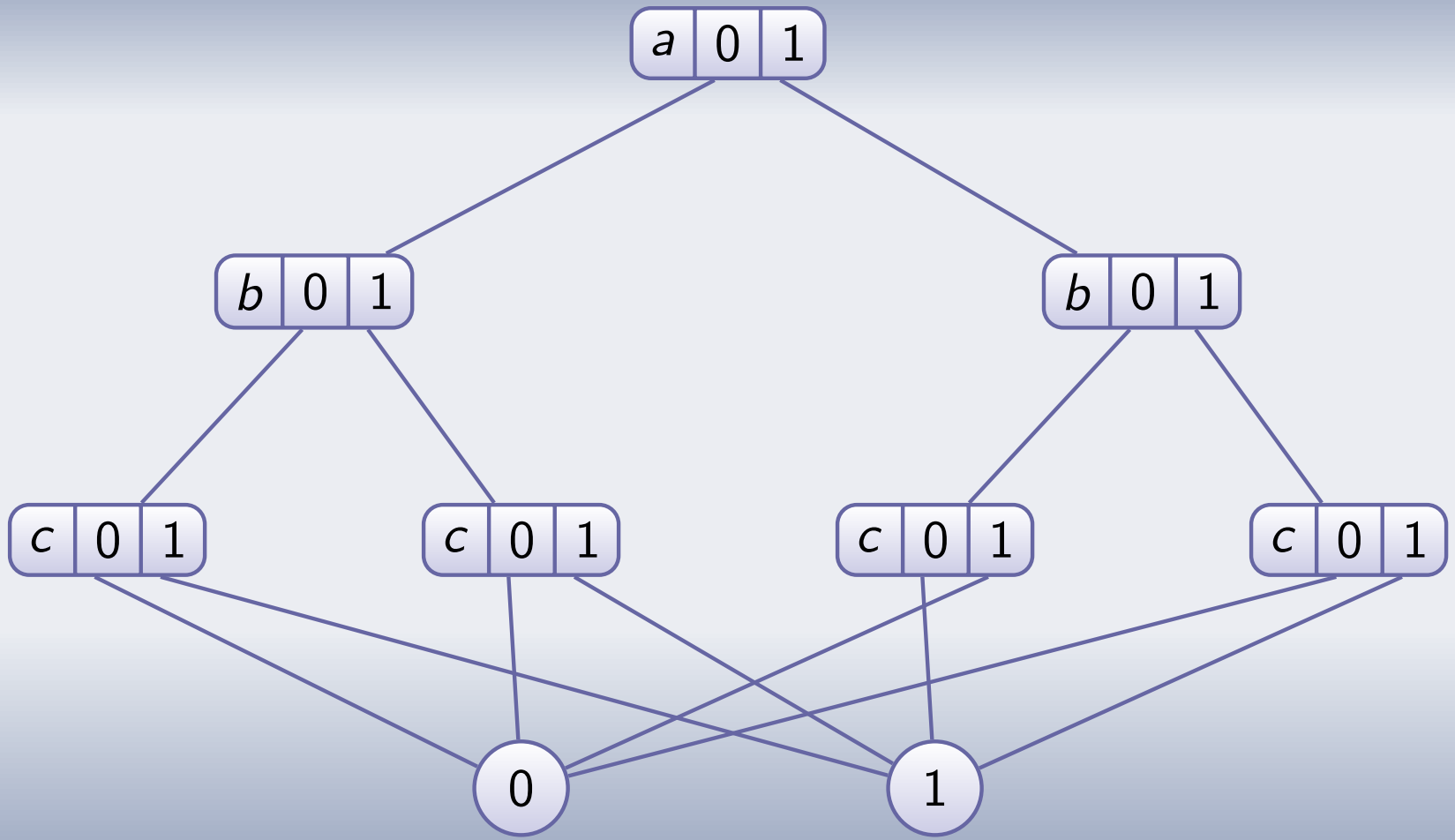
Fusionner les feuilles



toujours exponentiel

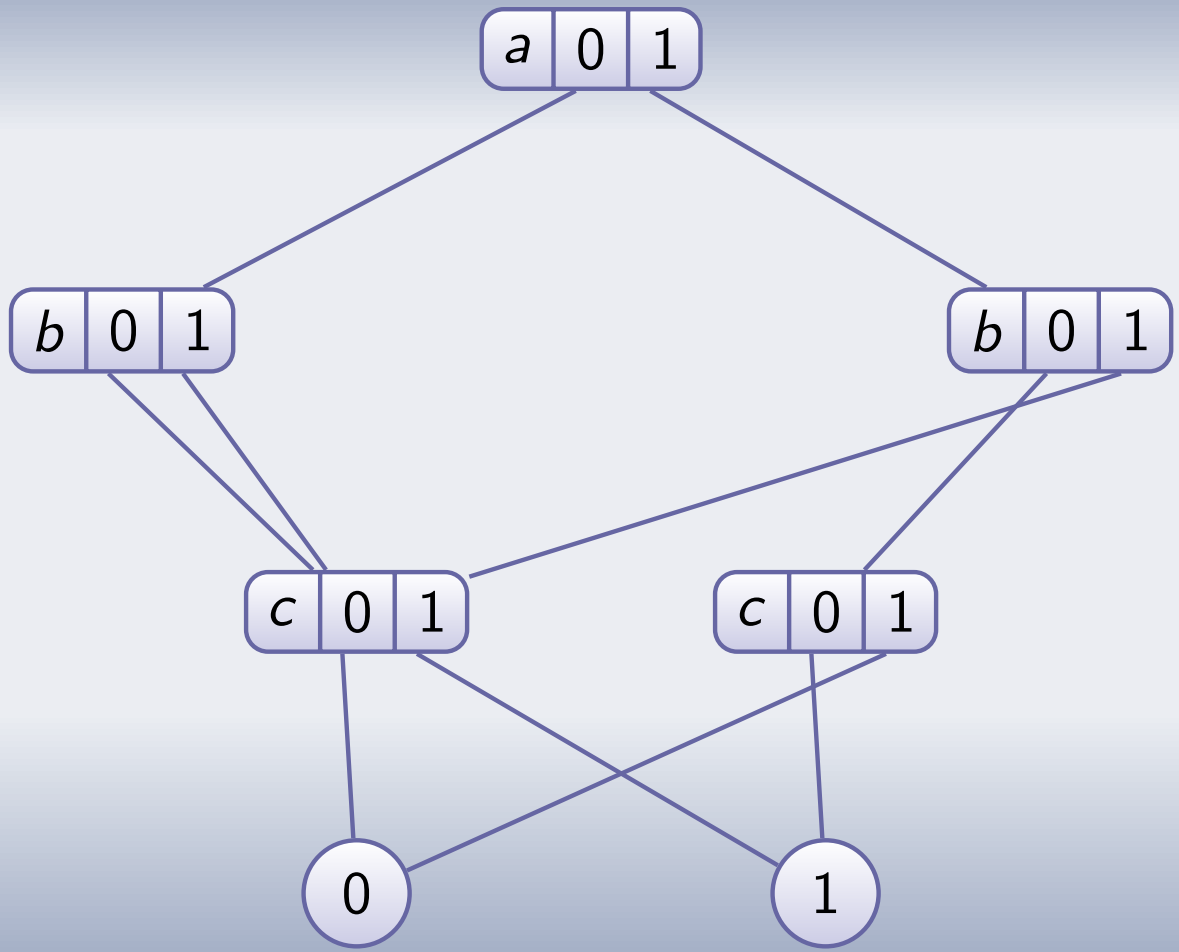
BDD (suite)

Fusionner les noeuds redondants



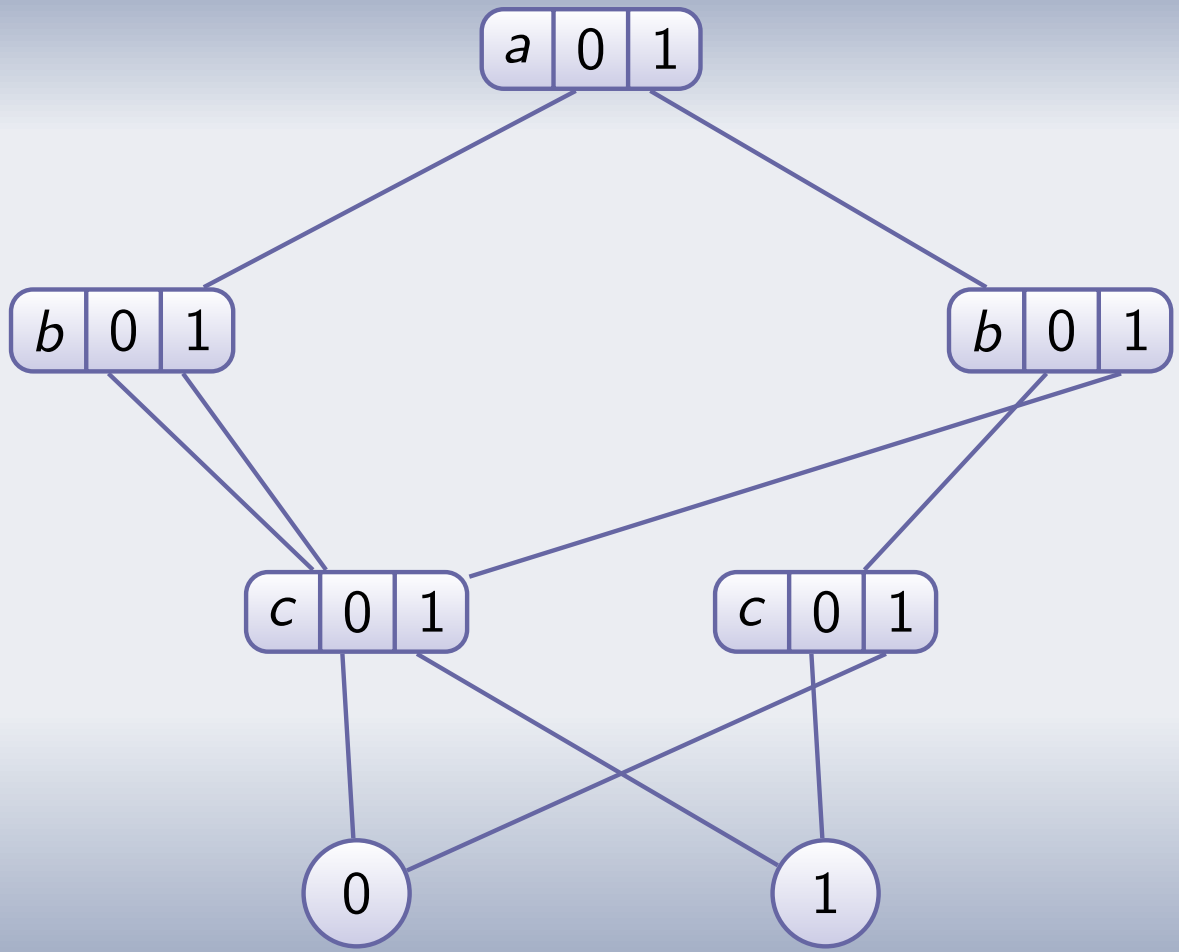
BDD (suite)

Fusionner les noeuds redondants



BDD (suite)

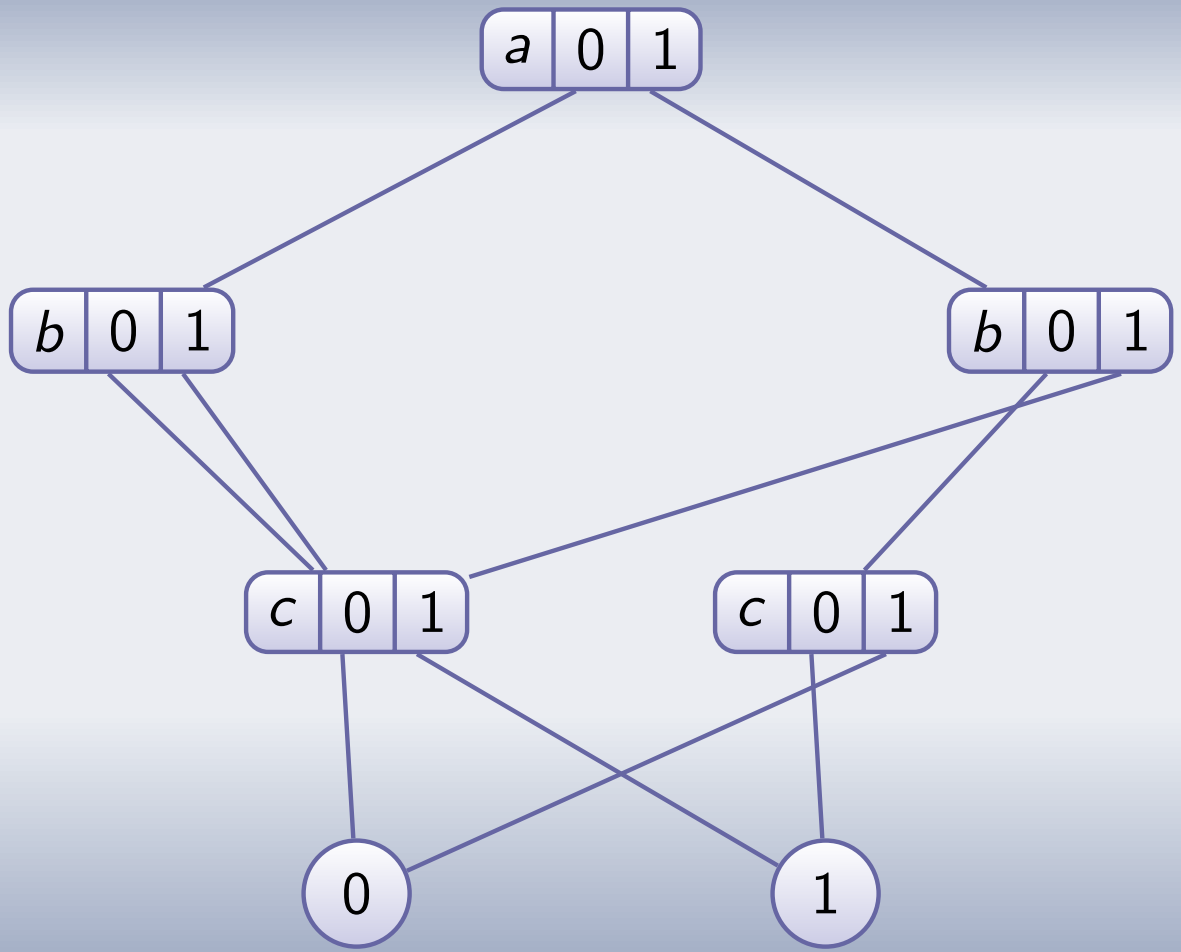
Fusionner les noeuds redondants



exponentiel au pire cas, souvent meilleur en pratique

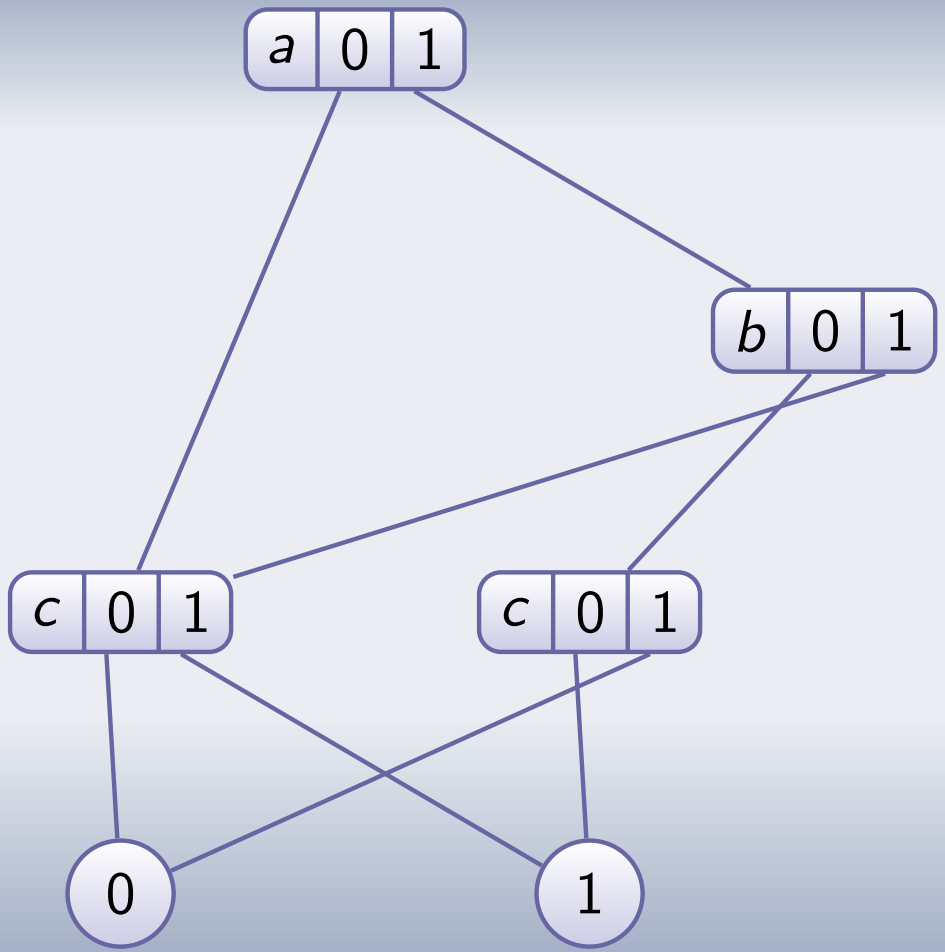
BDD (fin)

Supprimer les noeuds redondants



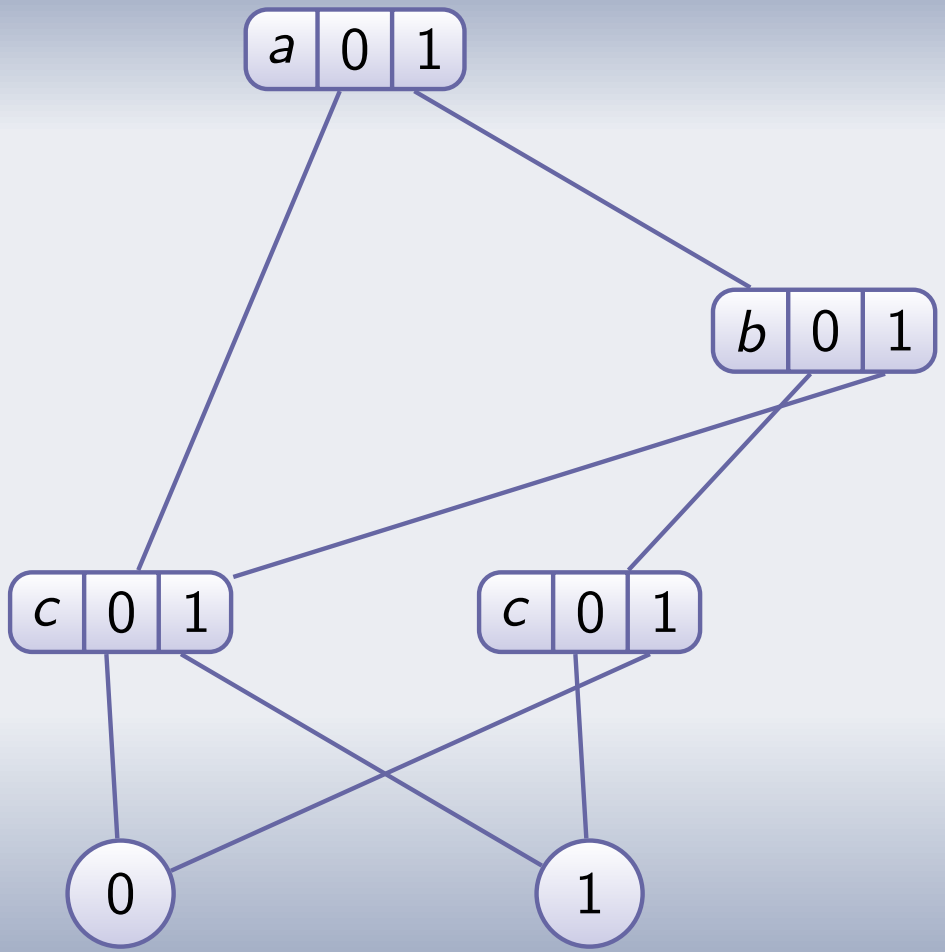
BDD (fin)

Supprimer les noeuds redondants



BDD (fin)

Supprimer les noeuds redondants



moins important (mais quand même)

Charactéristiques des BDD

Ils sont compacts mais surtout :

- canoniques :
  deux BDD représentent la même fonction ssi ils sont isomorphes
- faciles à manipuler :
  pour f, g représenté par deux BDD de taille $|f|$ et $|g|$
  $f * g$ calculé en $\mathrm{O}(|f|\,|g|)$

MTBDD

- $f : \{0,1\}^n \rightarrow \mathbb{Z}$

MTBDD

- $f : \{0,1\}^n \to \mathbb{Z}$
- Étendre les BDD en Multiple Terminal BDD

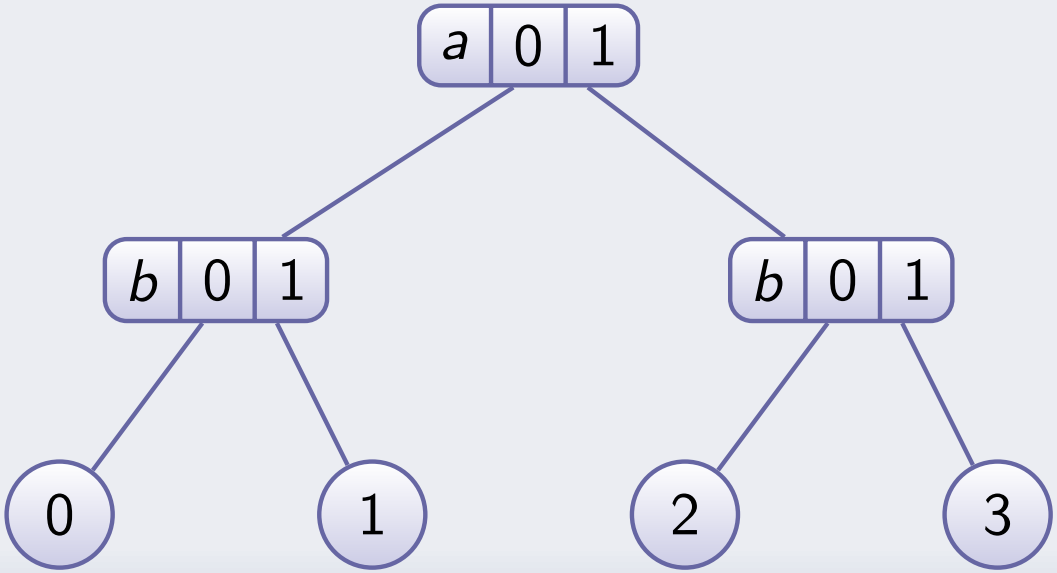


FIG.: $f : (a, b) \mapsto 2a + b$

MTBDD

- $f : \{0,1\}^n \to \mathbb{Z}$
- Étendre les BDD en Multiple Terminal BDD

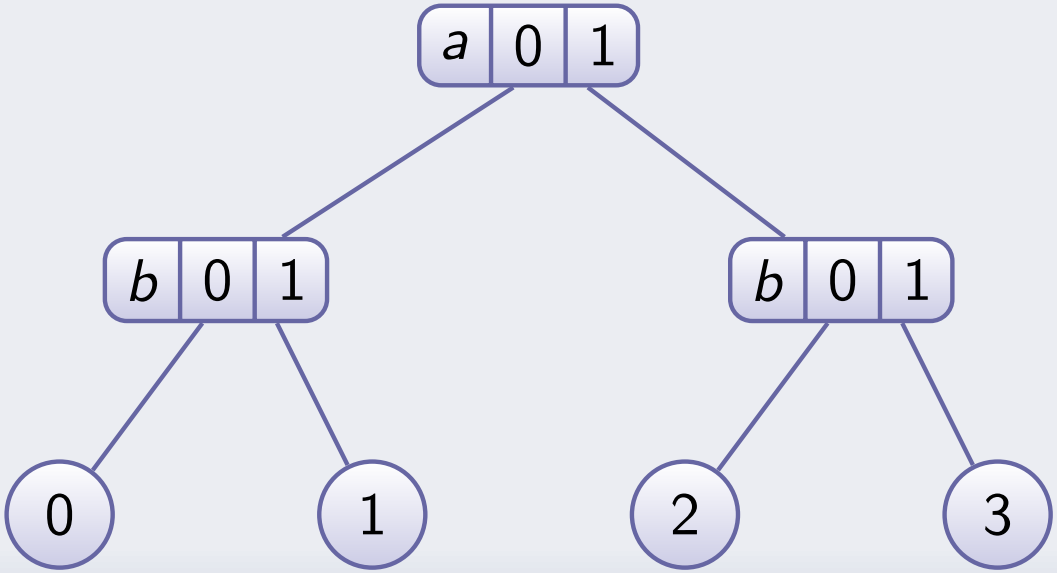
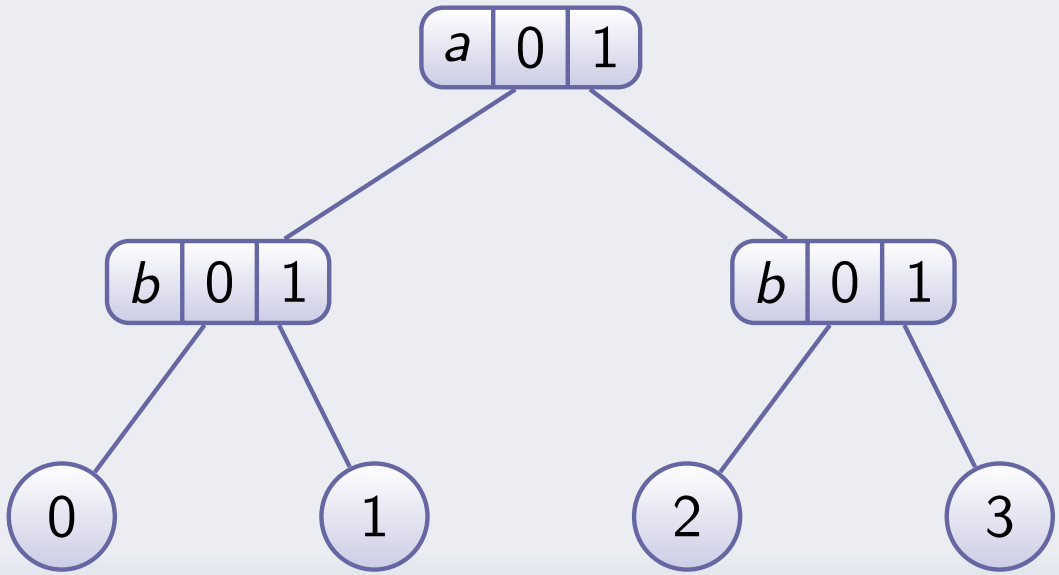


FIG.: $f : (a, b) \mapsto 2a + b$

- Mauvais quand $\mathrm{Img}(f)$ est trop grand

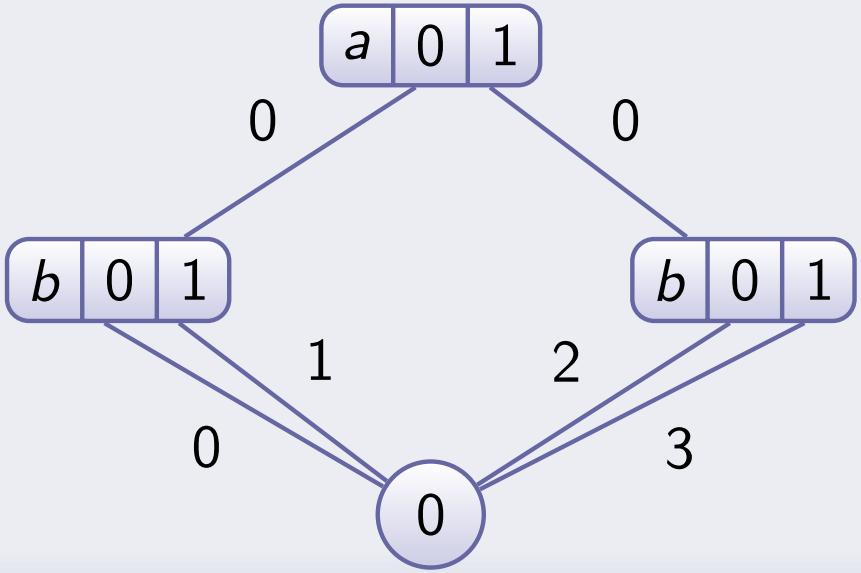
Edge Valued BDD (EVBDD)

Fusionner toutes les feuilles en 0 et mettre les valeurs sur les arêtes



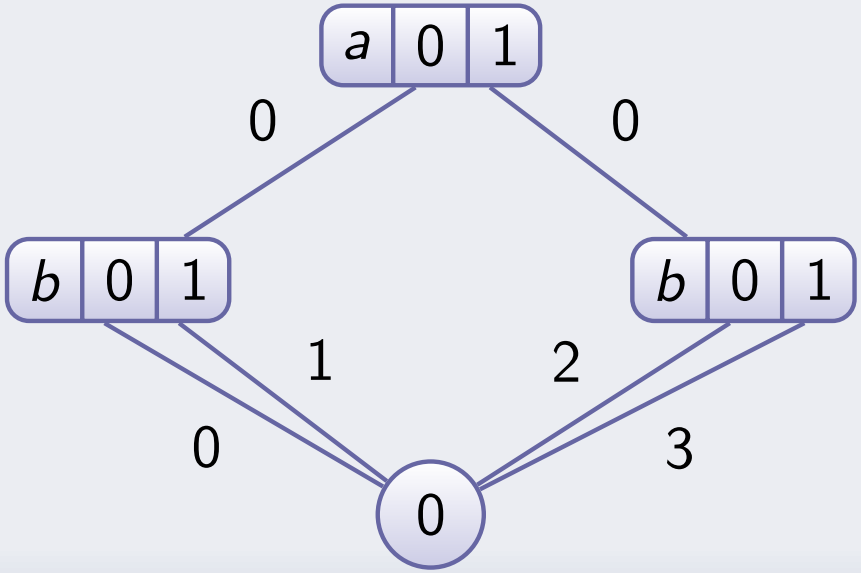
Edge Valued BDD (EVBDD)

Fusionner toutes les feuilles en 0 et mettre les valeurs sur les arêtes



Edge Valued BDD (EVBDD)

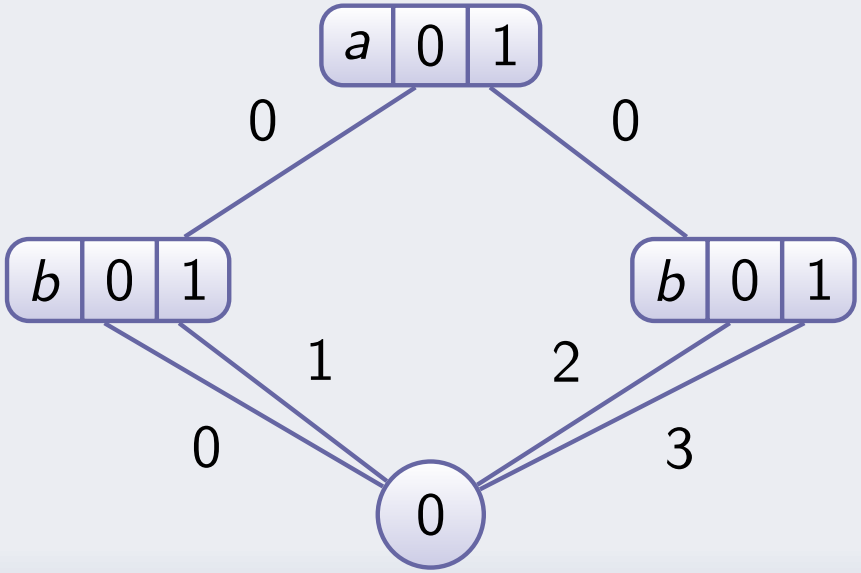
Fusionner toutes les feuilles en 0 et mettre les valeurs sur les arêtes



Évaluation : sommes sur le chemin de la racine à la feuille

Edge Valued BDD (EVBDD)

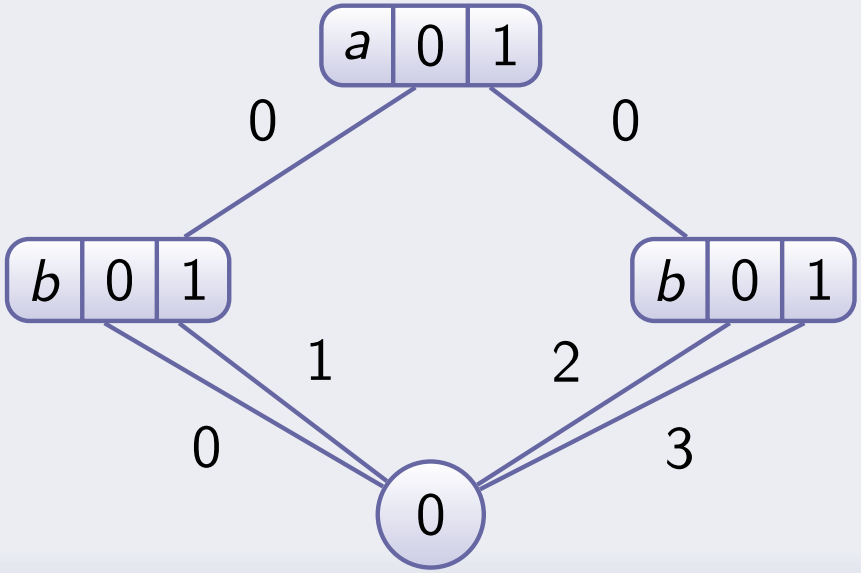
Fusionner toutes les feuilles en 0 et mettre les valeurs sur les arêtes



Évaluation : sommes sur le chemin de la racine à la feuille
pas mieux...

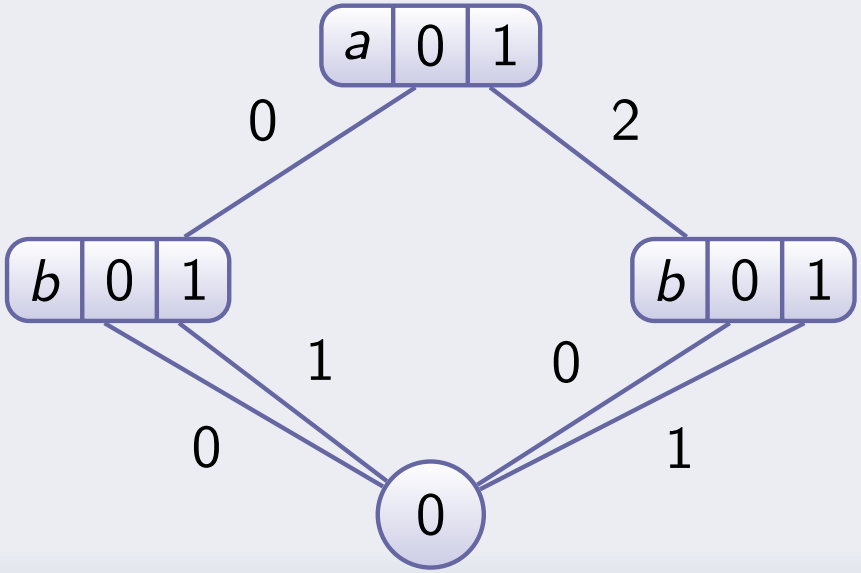
EVBDD (suite)

Noeud canonique : l'arête la plus à gauche porte la valeur 0



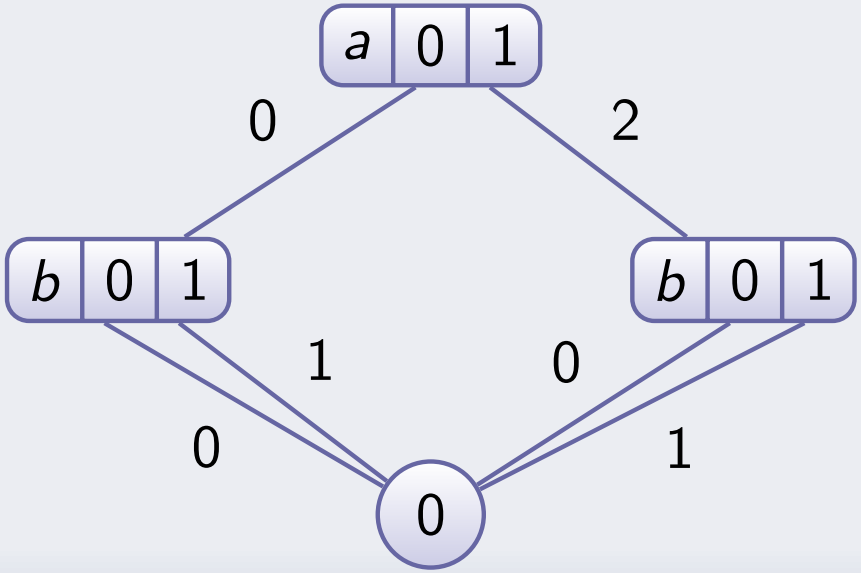
EVBDD (suite)

Noeud canonique : l'arête la plus à gauche porte la valeur 0



EVBDD (suite)

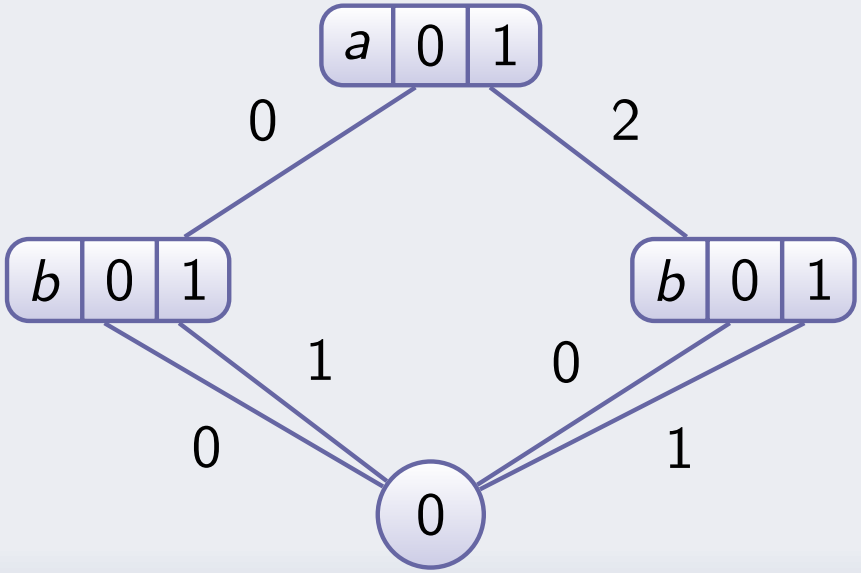
Noeud canonique : l'arête la plus à gauche porte la valeur 0



pas mieux...

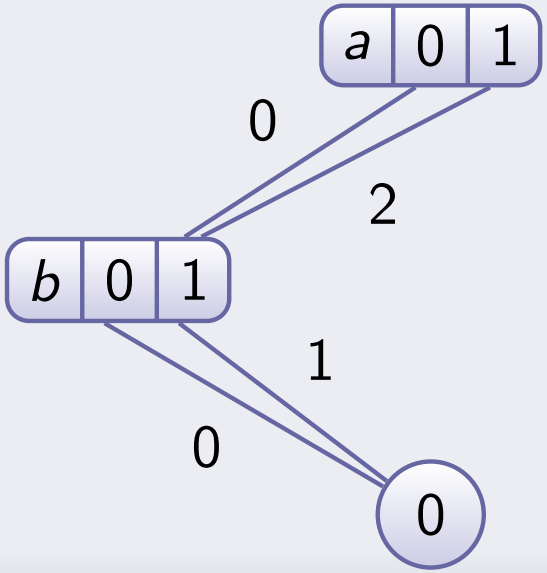
EVBDD (fin)

Fusionner les noeuds redondants



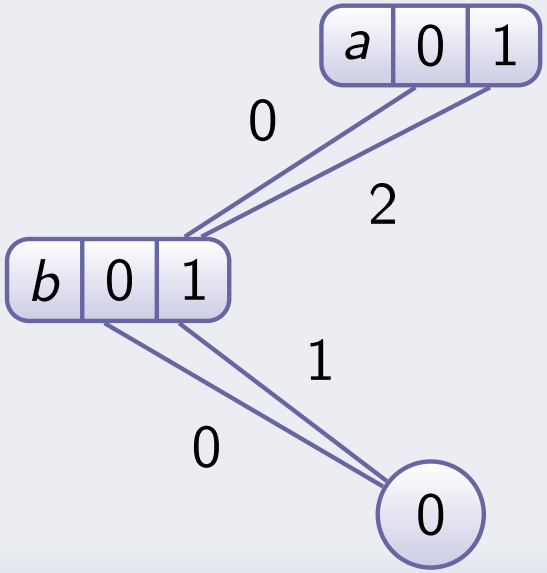
EVBDD (fin)

Fusionner les noeuds redondants



EVBDD (fin)

Fusionner les noeuds redondants



ici, on peut y gagner (ici linéaire au lieu d'exponentiel)

Caractéristiques des EVBDD

- Les EVBDD sont toujours plus petit que les MTBDD
- Parfois linéaire contre exponentiel
- manipulation : algorithme général
  pour f, g représentées par des EVBDD de tailles $|f|$ et $|g|$
  $f * g$ calculé en $\mathrm{O}\left(|f|\,|g|\,|\mathrm{Img}(f)|\,|\mathrm{Img}(g)|\right)$
- Théorème : **exactement la même complexité**
  que l'équivalent sur MTBDD
- Peut on faire mieux ?

Algorithmes sur les EVBDD

Oui pour certaines opérations

- Ajout de constante :
  $f + c$ calculé en $\mathrm{O}(1)$
- Addition :
  $f + g$ calculé en $\mathrm{O}\left(|f|\,|g|\right)$

Algorithmes sur les EVBDD

Oui pour certaines opérations

- Ajout de constante :
  $f + c$ calculé en $\mathrm{O}(1)$
- Addition :
  $f + g$ calculé en $\mathrm{O}\left(|f|\,|g|\right)$
- Multiplication par une constante :
  $f \times c$ calculé en $\mathrm{O}(|f|)$
- Division par une constante et reste :
  f/c et $f\%c$ calculés en $\mathrm{O}(|f|c)$
- Multiplication :
  $f \times g$ calculé en $\mathrm{O}\left(|f|^2\,|g|^2\,|f \times g|\right)$
  - meilleur sur des cas pratiques
  - produit de taille exponentielle au pire cas

Modèles globalement asynchrones localement synchrones

Relation de transition :

- disjonction de nombreux événements
- chaque événement n'affecte que quelques variables (localité)

D'où l'intérêt de :

- la représentation implicite des identités
- la stratégie d'itération « saturation »
  - ne pas exécuter chaque événement une fois par itération
  - mais calculer un point fixe pour chaque événement

Model checker

Librairie avec

- EVMDD pour les fonctions arithmétiques
- MDD pour les fonctions booléennes
- Construction de l'espace d'état par l'algorithme « saturation »
- Codage « identity reduced » de la relation de transition
- Garbage collector (mark & sweep)

Quelques chiffres

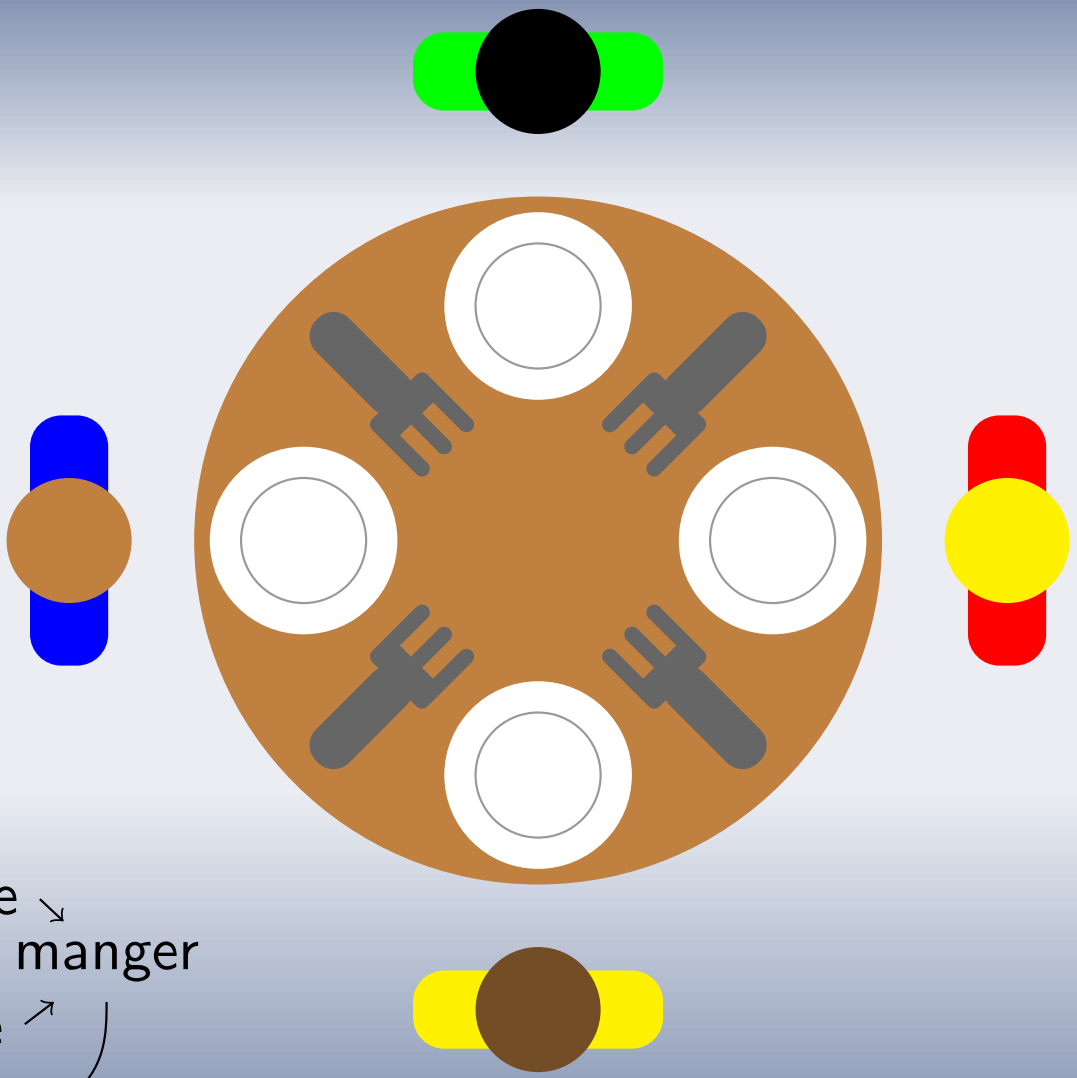
- 4 klocs de C pour la librairie
- 7 klocs pour le model checker

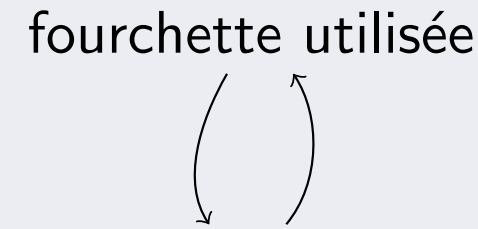
Démo

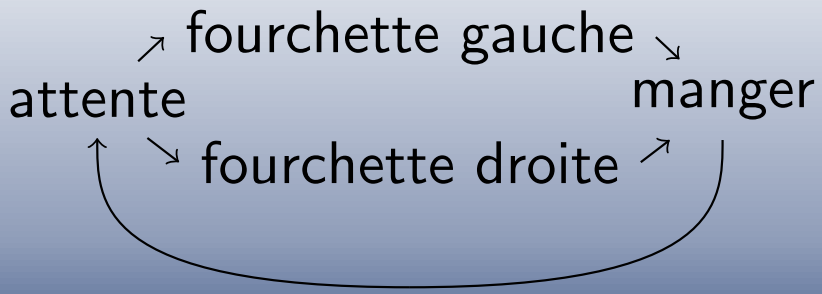
EVMDD et MTBDD

Démo (suite)

Dîner des philosophes







Démo (suite)

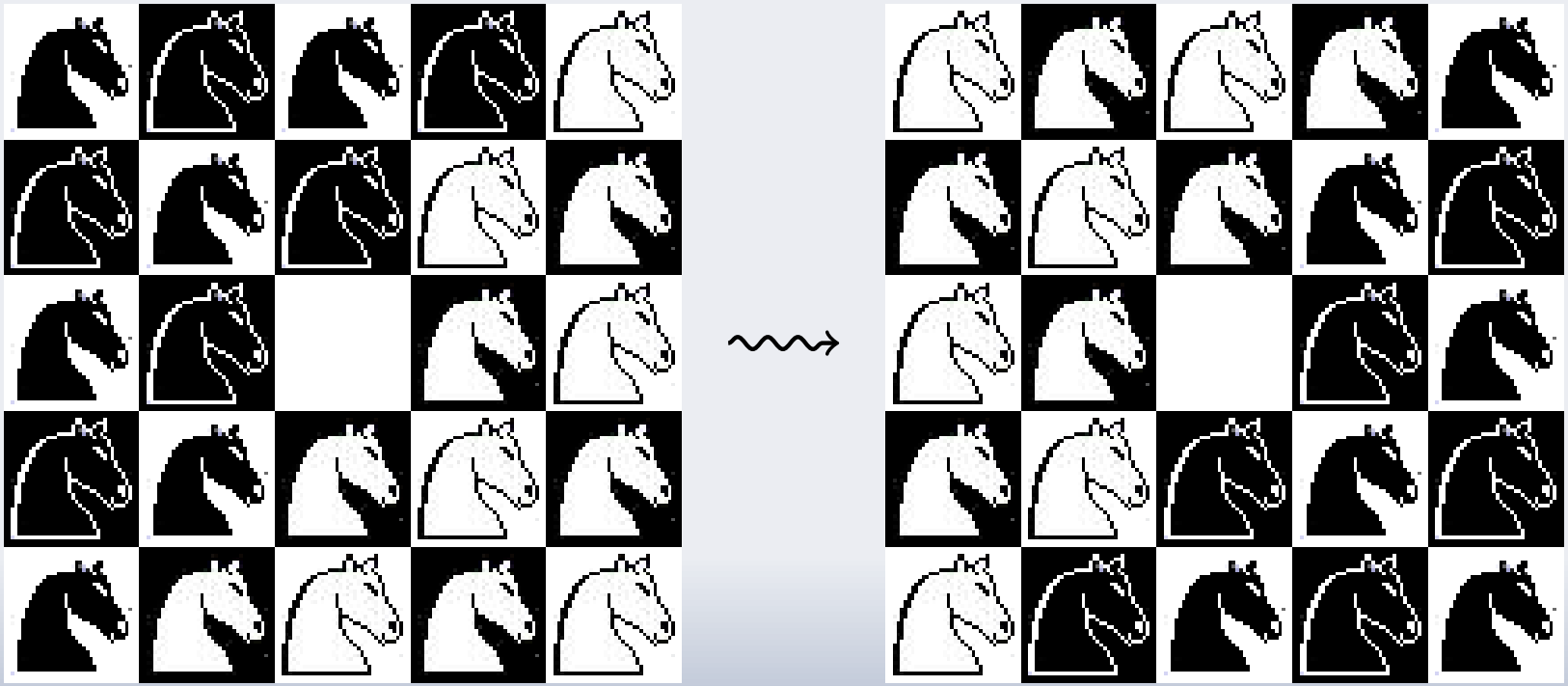
Dîner des philosophes, résultats

Nombre de philosophes	États atteignables	Deadlocks	CUDD (en s)	EVMDD (en s)
100	4×10^{62}	2	5.15	0.04
200	2×10^{125}	2	1493.36	0.10
1000	9×10^{626}	2	—	1.10
5000	6×10^{3134}	2	—	20.48
10000	4×10^{6269}	2	—	77.77
16000	2×10^{10031}	2	—	196.84

TAB.: Recherche de deadlocks, comparé à la librairie CUDD (temps d'exécution, "—" : "plus d'une heure").

Démo (fin)

Chevaliers



Questions

?