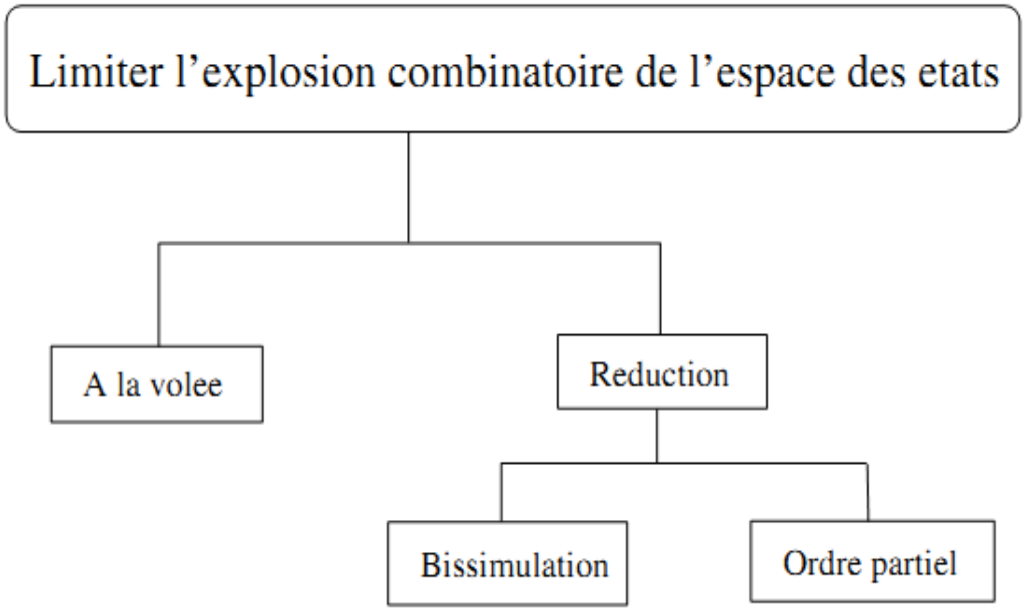
PROBLÈME DE L'EXPLOSION COMBINATOIRE DU GRAPHE D'ÉTAT

Dr. D. Boughareb
boughareb@labged.net

Introduction

- Inconvénient des méthodes de vérification basée sur les modèles (model-checking)
  - Complexité réaliste de système vérifié =>la taille du modèle correspondant devient rapidement abusive.
  - =>L'explosion de l'espace d'états.
    - Besoin d'un espace mémoire important;
    - Temps CPU pour explorer chaque état

Solutions proposées (1)



Approches possibles pour limiter l'explosion combinatoire de l'espace des états

Solutions proposées (2)

- Générer un modèle réduit, mais équivalent au modèle original du point de vue vérification des propriétés à valider.

- L'exploration partielle du modèle pour améliorer les performances, en temps ou en mémoire, en effectuant
  - Une exploration partielle
  - Ou une mémorisation partielle du modèle.

Vérification à la volée(1)

- Vérifier le système durant la génération du graphe d'état.
- La vérification s'arrête après la première erreur détectée (avant la génération du graphe complet)
- Parcours en profondeur:
  - Algorithmes DFS (depth first search) et BFS (breadth first search).

Algorithme DFS (1)

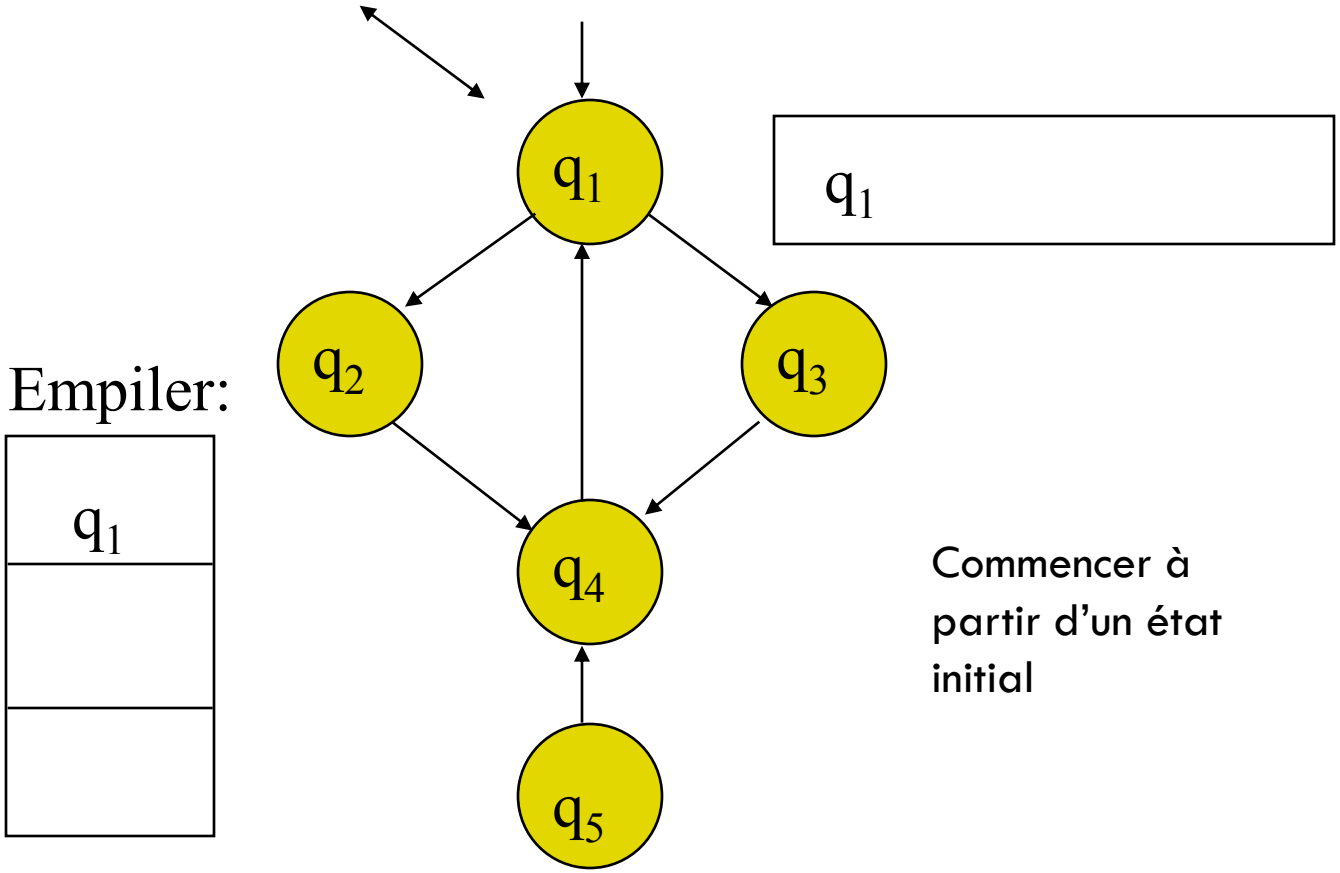
Program DFS

For each s such that
    Init(s)
        dfs(s)
end DFS

Procedure dfs(s)

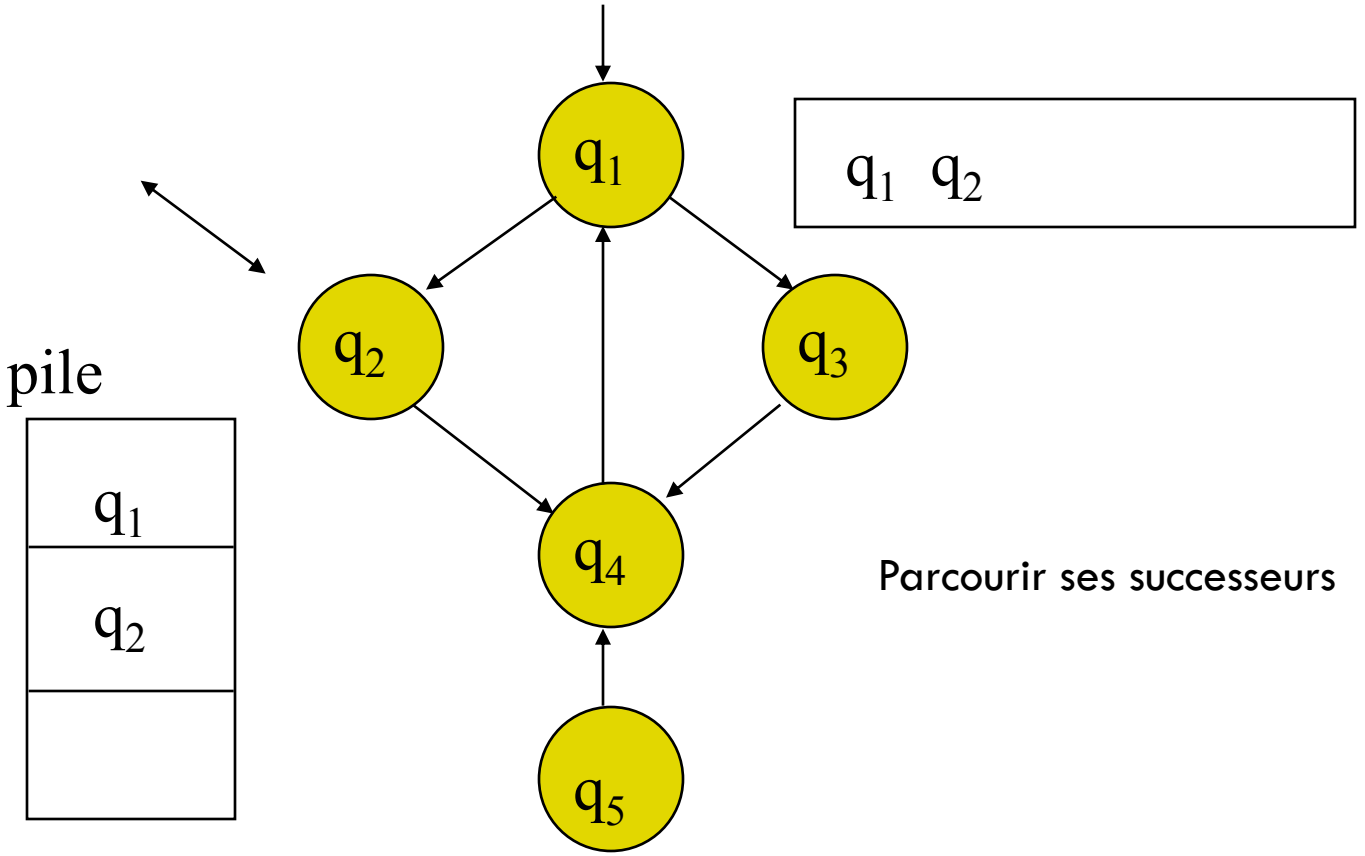
for each s' such that
        R(s,s') do
    If new(s') then dfs(s')
end dfs.

Algorithme DFS (2)

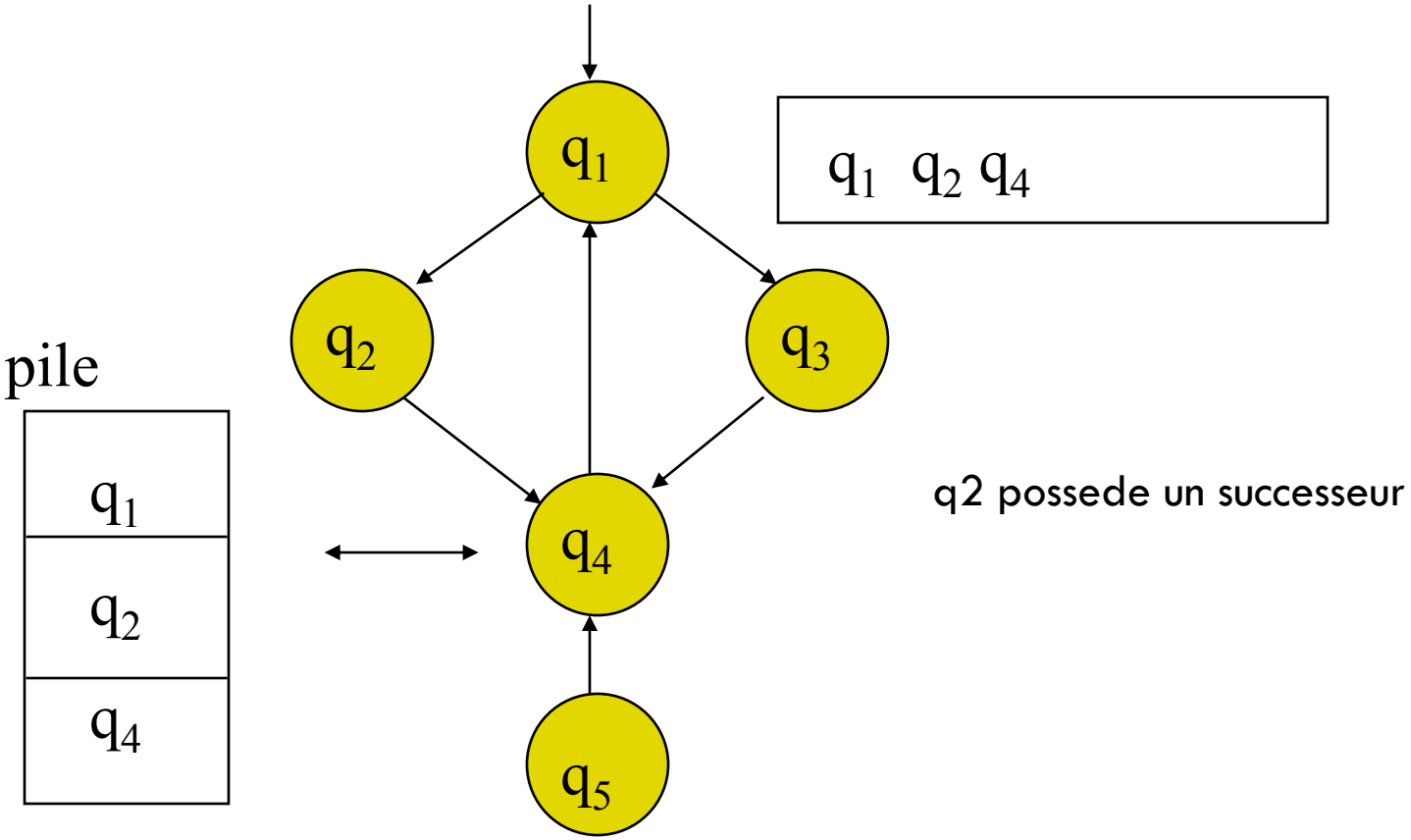


Commencer à partir d'un état initial

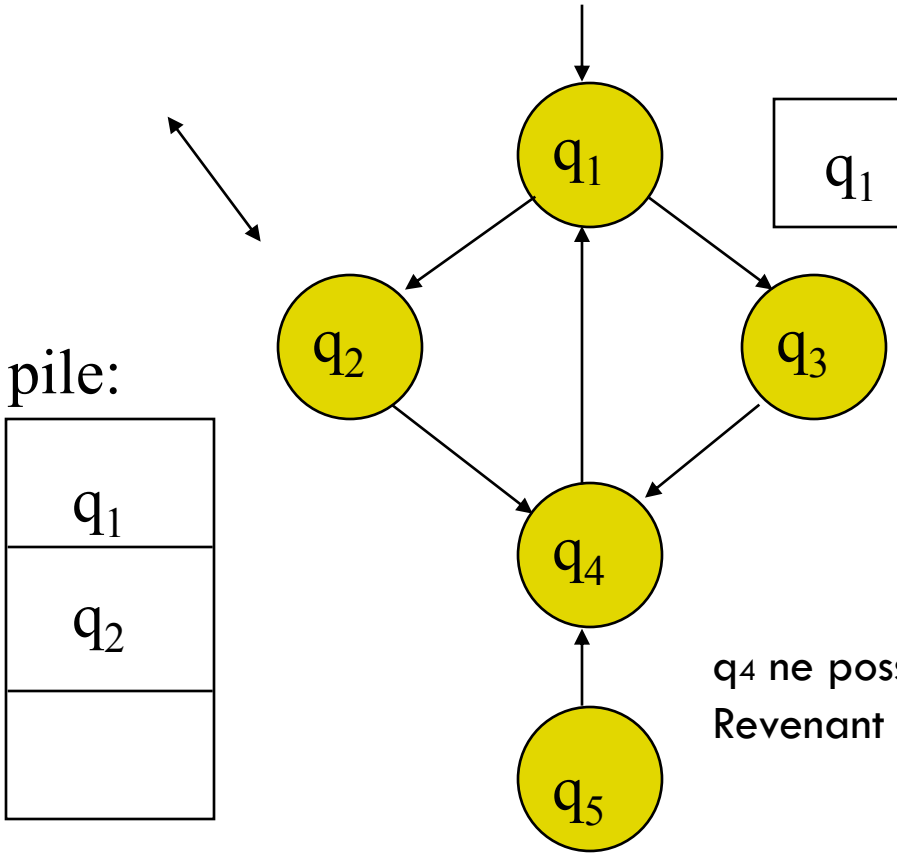
Algorithme DFS (3)



Algorithme DFS (4)



Algorithme DFS (5)



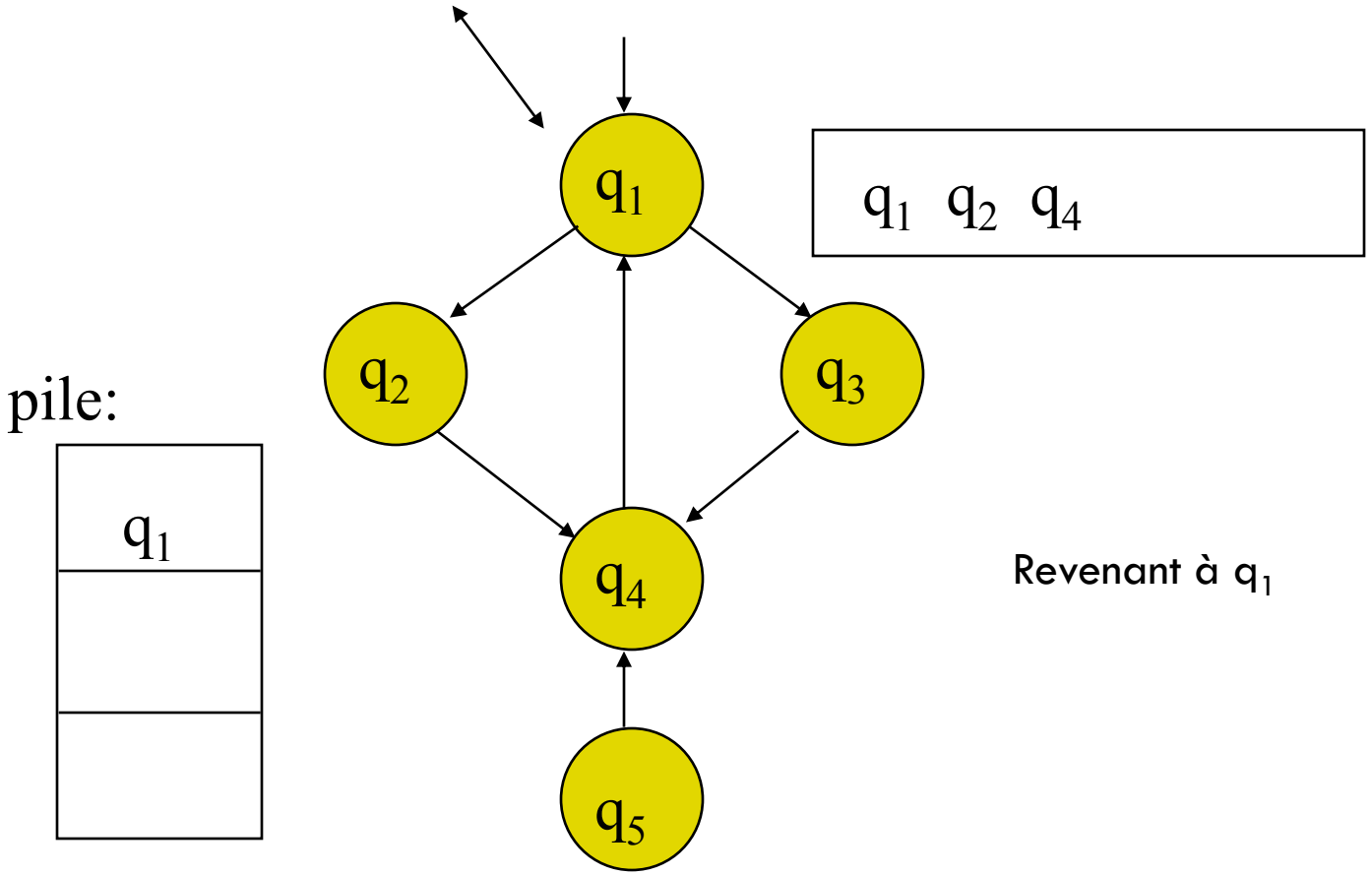
q_1 q_2 q_4

pile:

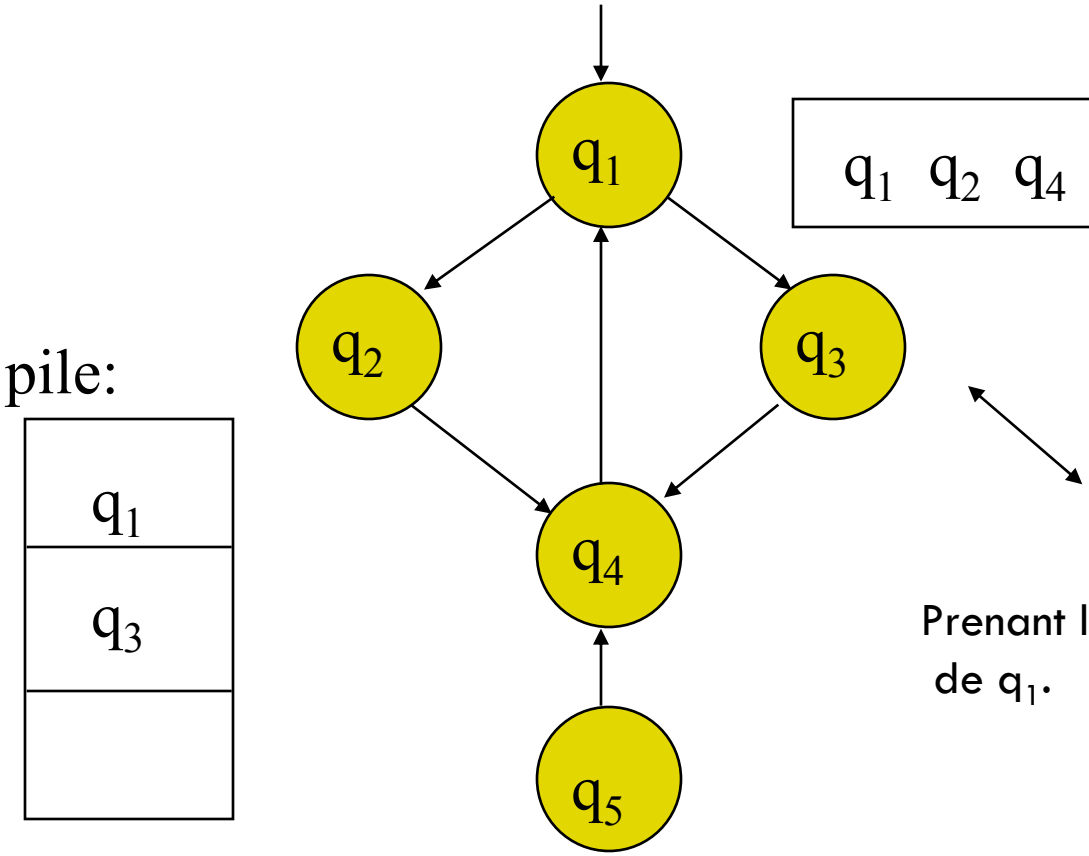
q_1
q_2

q_4 ne possède aucun nouveau successeur
Revenant à q_2.

Algorithme DFS (6)

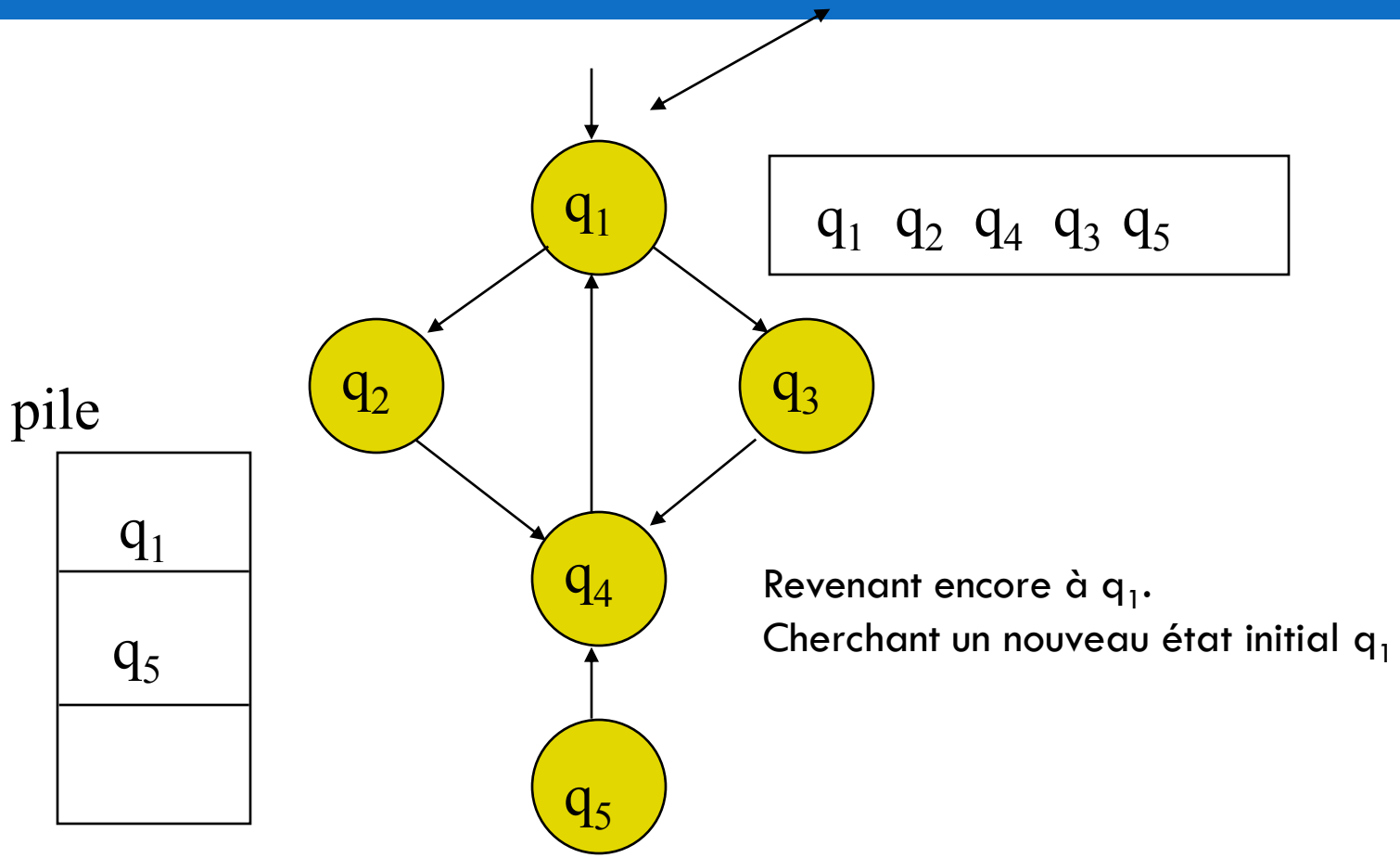


Algorithme DFS (7)



Prenant le deuxième successeur de q_1.

Algorithme DFS (8)



Vérification par réduction (1)

- Construire et prouver qu'un sous ensemble de l'espace des états est suffisant pour vérifier les propriétés
  I. Réduction par bissimulation : de grandes parties du graphe d'états sont remplacés par de plus petits qui sont équivalentes selon un critères d'abstraction.

  II. Réduction basée sur les ordres partiels :

Vérification par réduction (2)

II. Réduction basée sur les ordres partiels :

- Les exécutions qui ne se diffèrent que par l'ordre d'actions indépendantes peuvent être considérées comme équivalentes pour la propriété à vérifier.

- Il est seulement-nécessaire de vérifier un graphe réduit d'états qui contient au moins une exécution représentative de chaque classe d'équivalence. Parmi les techniques qui reposent sur cette idée
  - Ensembles persistants
  - Ensemble dormants
  - Ensemble amples

Techniques symboliques

- Symbolic Model-Checking.

- Représentation symbolique d'un ensemble d'états par un ensemble de formules en logique propositionnelle.

Fonctions booléennes

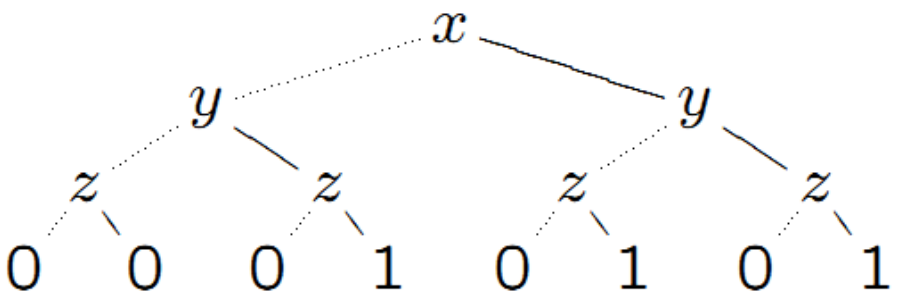
- Représentations possibles
  - Binary decision trees
  - Binary decision diagrams
  - Ordered binary decision diagrams
  - Reduced ordered binary decision diagrams

Fonctions booléennes

- Une fonction booléenne établie le chemin des entrées booléennes (0s et1s) jusqu'aux résultats booléens (0 ou 1).
- Fonctions Basique :
  - $\overline{0} = 1, \overline{1} = 0$ (complémentation).
  - $x+y = 0 \leftrightarrow x = y = 0$ (disjonction, similaire à l'addition).
  - $x \,.\, y = 1 \leftrightarrow x = y = 1$(conjonction, , similaire à la multiplication).
  - $x \oplus y = 1 \leftrightarrow x \neq y$ (or exclusif).

Binary decision trees

- Representation par une table de vérité
- Representation par un arbre de decision binaire



Evaluation:

arc solide pour des 1

arc pointillé pour des 0

x	y	z	$(x+y)\cdot z$
1	1	1	1
1	1	0	0
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	0

Le nombre de variables bouleenes augmente => difficile à verifier

BDD (Binary Decision Diagrams)

- Permet de représenter efficacement de larges fonctions
- Un diagramme binaire utilisé pour représenter des fonctions booléennes.
- Composé de :
  - Nœuds non-Terminaux : nœuds étiquetés par des variables booléennes;
  - Nœuds terminaux: nœuds étiquetés soit par un 0 ou 1;
  - Deux types de lignes (arc) solide et pointillé;
  - Chemins conduisant à un 1 représente les modèles, tandis que les chemins conduisant à un 0 représentent les contre-modèles.

Construction d'un BDD

- Utilisation récursive du théorème de Shannon étendu

$$F = aFa + a'Fa'$$

$$F = a'b + abc' + a'b'c$$

$$F = a(bc') + a'(b+b'c)$$

Fa=(bc')

Fa'=(b+b'c)

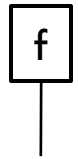
F = a'b + abc' + a'b'c
F = bFb + b'Fb'
F = b(a'+ ac') + b'(a'c)

F = a'b + abc' + a'b'c
F = cFc + c'Fc'
F = c(a'b + a'b') + c'(a'b + ab)

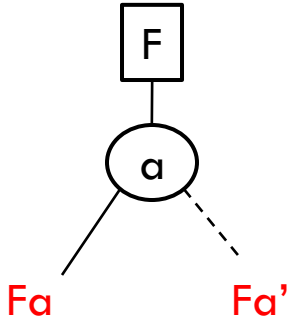
Construction d'un BDD

- Construire le BDD de

$$F = abc + ab'c + a'bc' + a'b'c'$$



- Utiliser l'ordre des variables $a \leq b \leq c$
- $Fa = bc + b'c$
- $Fa' = bc' + b'c'$



Construction d'un BDD

$$F = abc + ab'c + a'bc' + a'b'c'$$

$Fa = bc + b'c$

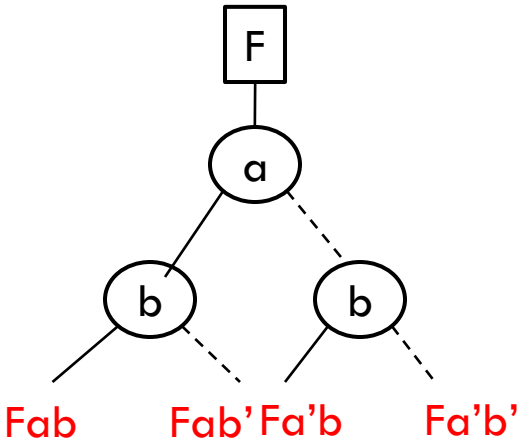
(Fa)b=Fab=c

(Fa)b'=Fab'=c

$Fa' = bc' + b'c'$

(Fa')b=Fa'b=c'

(Fa')b'=Fa'b'=c'



Construction d'un BDD

- F = abc + ab'c + a'bc' + a'b'c'

Fab = c

Fabc=1

Fabc'=0

Fab'= c

Fab'c=1

Fab'c'=0

Fa'b = c'

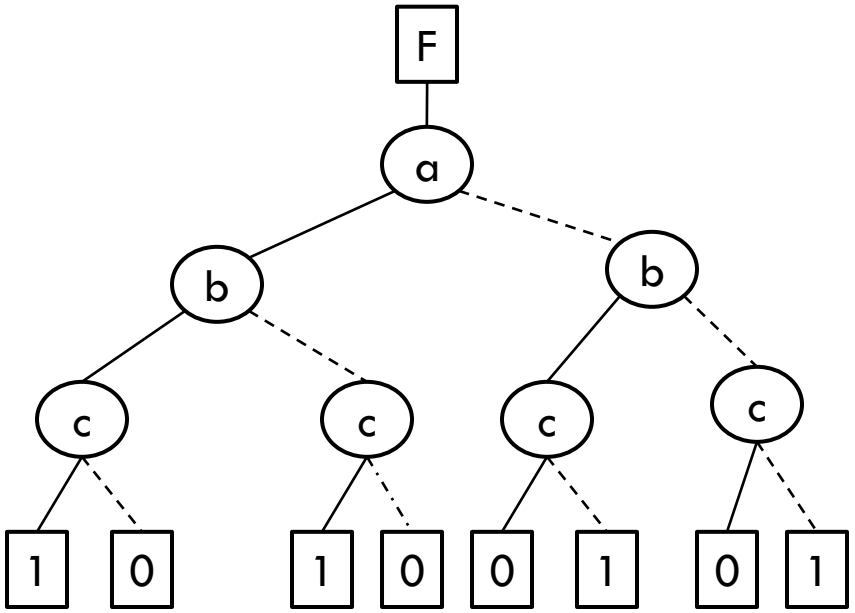
Fa'bc=0

Fa'bc'=1

Fa'b'= c'

Fa'b'c=0

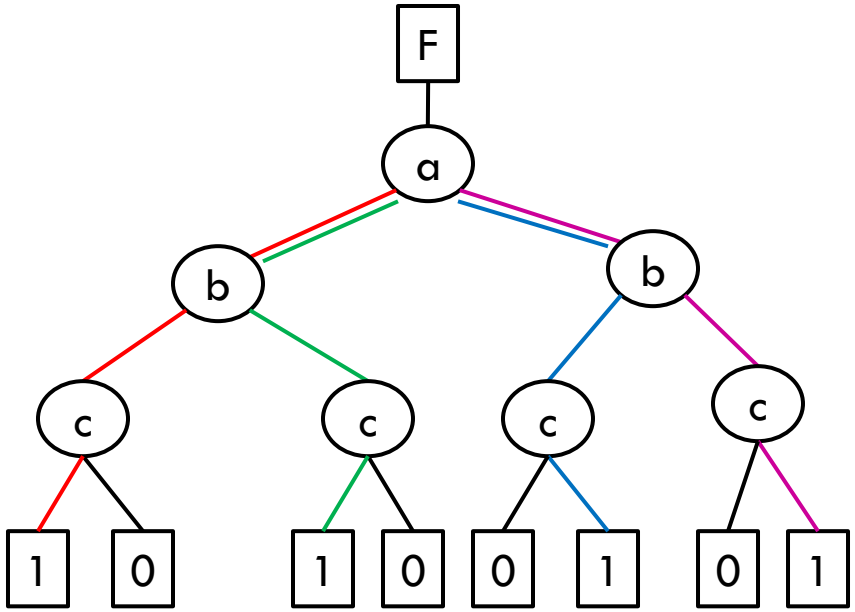
Fa'b'c'=1



Construction d'un BDD

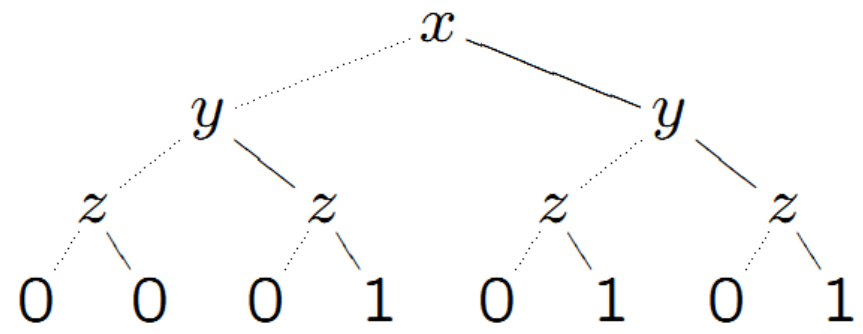
f = abc + ab'c + a'bc' + a'b'c'

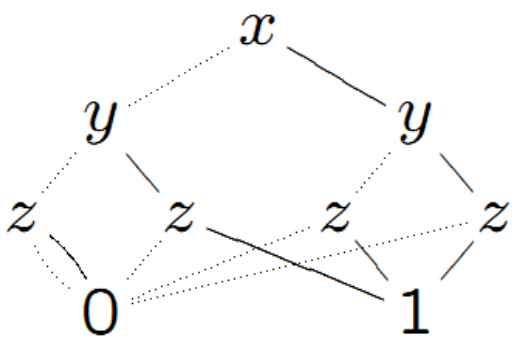
a	b	c	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



Reduction de BDD

- Éliminer les information redondantes depuis le BDD.

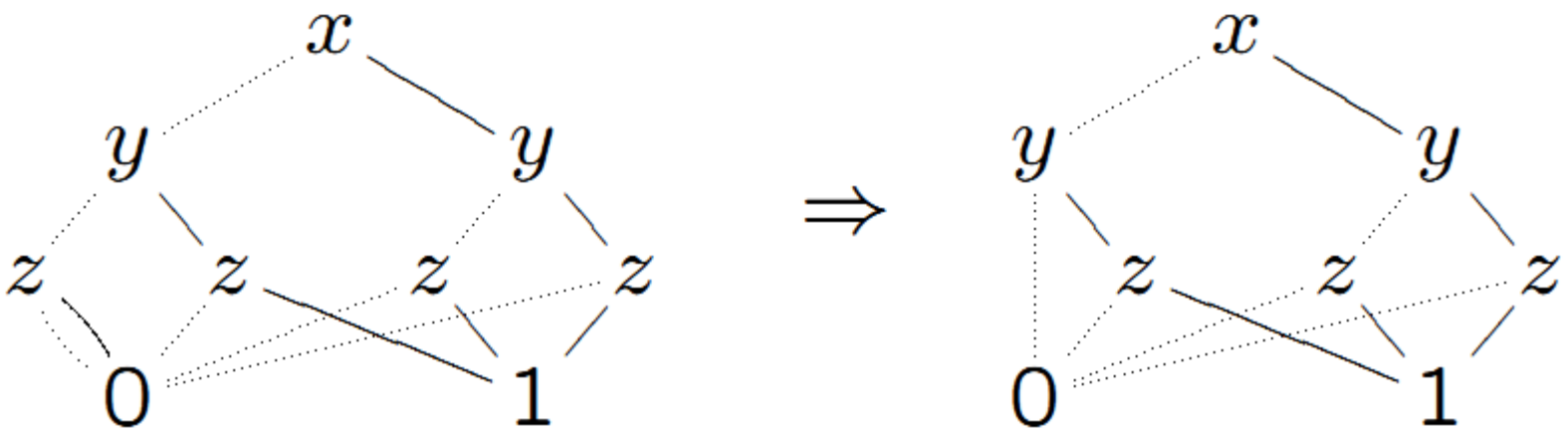




Multiple (0 et 1) en nœuds terminaux (redondance)

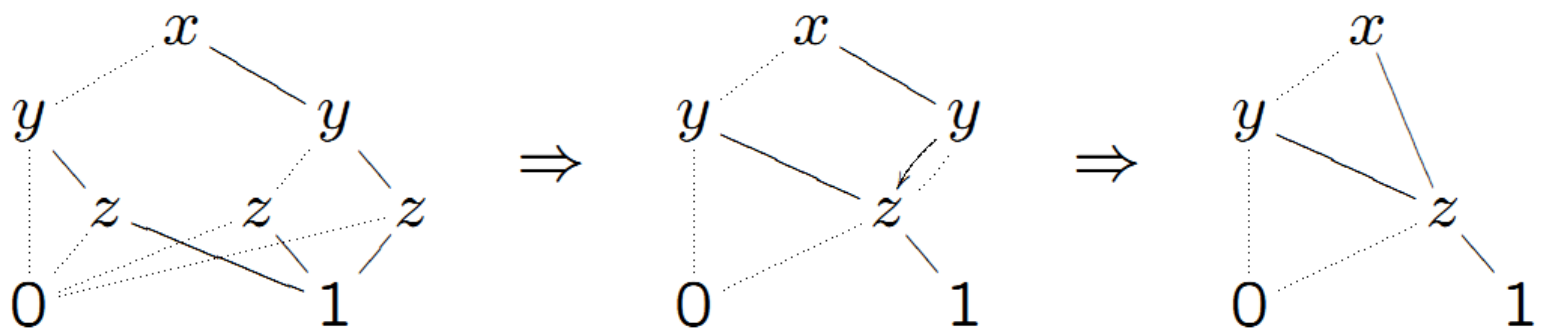
Reduction de BDD

- Si le nœud dispose du même fils dans les deux arcs, le nœud est redondant



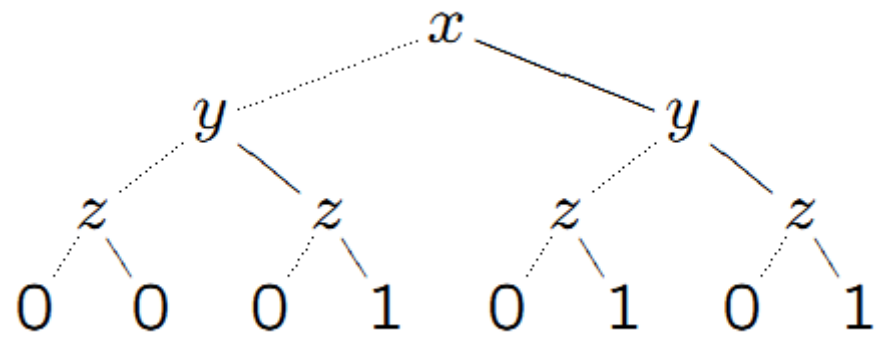
Reduction de BDD

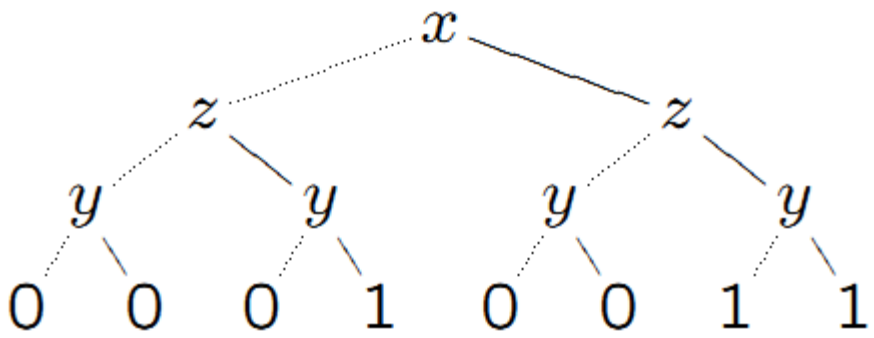
- Si 2 nœuds représentent la même variable et ils possèdent les mêmes descendants, l'un des deus nœuds est redondant.



Ordonnancement des Variables

- Réduction dépend de l'ordre des variables





Ordered BDDs (OBDDs)

- Problèmes avec les BDDs:
  - Structure dépend de l'ordre de représentation des variables
- Solution: fixer l'ordre de variables pour tout les chemins.

Ordonné : [y,z,x] :	Non ordonné :