Transformation des formules LTL en Automates de Büchi

Master GLSD

2016-2017

Plan

1) Idée des méthodes:
   - Les tableaux sémantiques (preuve de formules),
   - Extension pour les formules LTL

2) Application pour la transformation des formules LTL vers AB:
   - Méthode GPVW,
   - Méthode de Couvreur

Idée de base
Preuve par tableaux

- Idée : utiliser une technique pour la preuve des formules de la logique propositionnelle;

- Cette méthode est dite: celle **des tableaux sémantiques;**

- Un tableau est représenté sous forme d'**un arbre;**

- Des règles sont définies pour construire cet arbre;

- L'arbre représente une mise **sous forme disjonctive** de la formule;

- La formule est satisfiable **ssi il y'a au moins une branche qui ne mène pas vers une contradiction;**

Idée de base:

Règles de construction de l'arbre

- **Les feuilles** de l'arbre seront des ensembles de **formules atomiques**

- Les feuilles de l'arbre peuvent contenir des **contradictions**

Idée de base:

Règles de construction de l'arbre

	formule	1^{er} fils	2^{e} fils
(1)	$\neg\neg f$	$\{f\}$	
(2)	$\neg\top$	$\{\bot\}$	
(3)	$\neg\bot$	$\{\top\}$	
(4)	$f \wedge g$	$\{f, g\}$	
(5)	$f \vee g$	$\{f\}$	$\{g\}$
(6)	$\neg(f \wedge g)$	$\{\neg f\}$	$\{\neg g\}$
(7)	$\neg(f \vee g)$	$\{\neg f, \neg g\}$	

Remarque :

Pour prouver que *f* est satisfiable, il peut être plus simple de prouver que $\neg f$ est une contradiction

Idée de base
exemple

Prouver la formule f=

$$\neg(\neg A \vee B) \vee (\neg(A \wedge C) \vee (B \wedge C))$$

On va prouver que $\neg$f est une antilogie

Idée de base exemple

$$\{\neg(\neg(\neg A \vee B) \vee (\neg(A \wedge C) \vee (B \wedge C)))\}$$

$$\downarrow (7)$$

$$\{\neg\neg(\neg A \vee B), \neg(\neg(A \wedge C) \vee (B \wedge C))\}$$

$$\downarrow (1)$$

$$\{\neg A \vee B, \neg(\neg(A \wedge C) \vee (B \wedge C))\}$$

$$\downarrow (7)$$

$$\{\neg A \vee B, \neg\neg(A \wedge C), \neg(B \wedge C)\}$$

$$\downarrow (1)$$

$$\{\neg A \vee B, A \wedge C, \neg(B \wedge C)\}$$

$$\downarrow (4)$$

$$\begin{tikzcd}[column sep=small]
 & \{\neg A \vee B, A, C, \neg(B \wedge C)\} \arrow[dl, "(5)"] \arrow[dr] & & \\
\{\neg A, A, C, \neg(B \wedge C)\} & & \{B, A, C, \neg(B \wedge C)\} \arrow[dl, "(6)"] \arrow[dr] & \\
 & \{B, A, C, \neg B\} & & \{B, A, C, \neg C\}
\end{tikzcd}$$

Extension pour LTL

- La même technique sera utilisée pour prouver que les formules LTL sont satisfiables;
- Des **règles** sont définies pour **les opérateurs temporels;**
- Prouver une formule consiste à **voir les sous-formules à prouver pour l'instant** (nœud présent) et **les sous-formules à prouver dans le futur** (nœuds suivants définis par la présence de l'opérateur X);
- **L'arbre sera infini** mais ses **nœuds sont dénombrables** donc on peut la représenter sous forme d'un graphe;

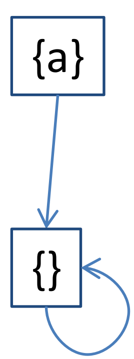
Extension pour LTL

formule	1er fils	2e fils
$\neg \mathsf{X} f$	$\{\mathsf{X}\neg f\}$	
$f \,\mathsf{U}\, g$	$\{g\}$	$\{f, \mathsf{X}(f \,\mathsf{U}\, g)\}$
$\neg(f \,\mathsf{U}\, g)$	$\{\neg f, \neg g\}$	$\{\neg g, \neg \mathsf{X}(f \,\mathsf{U}\, g)\}$

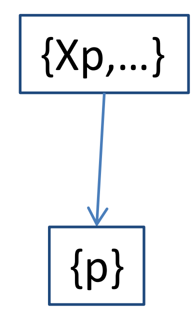
- On applique cette décomposition jusqu'à avoir **des nœuds avec des atomes**, des **nœuds non décomposables ou des nœuds avec des opérateurs X**,

- Après, on applique la décomposition des formules Xp, qui permet de montrer le passage d'un instant vers un autre

Pour les atomes, on aurait un fils:



Pour le X



Extension pour LTL

- l'opérateur X est **toujours le dernier opérateur à traiter** dans la construction de l'arbre (après les: v, $\wedge$, U);

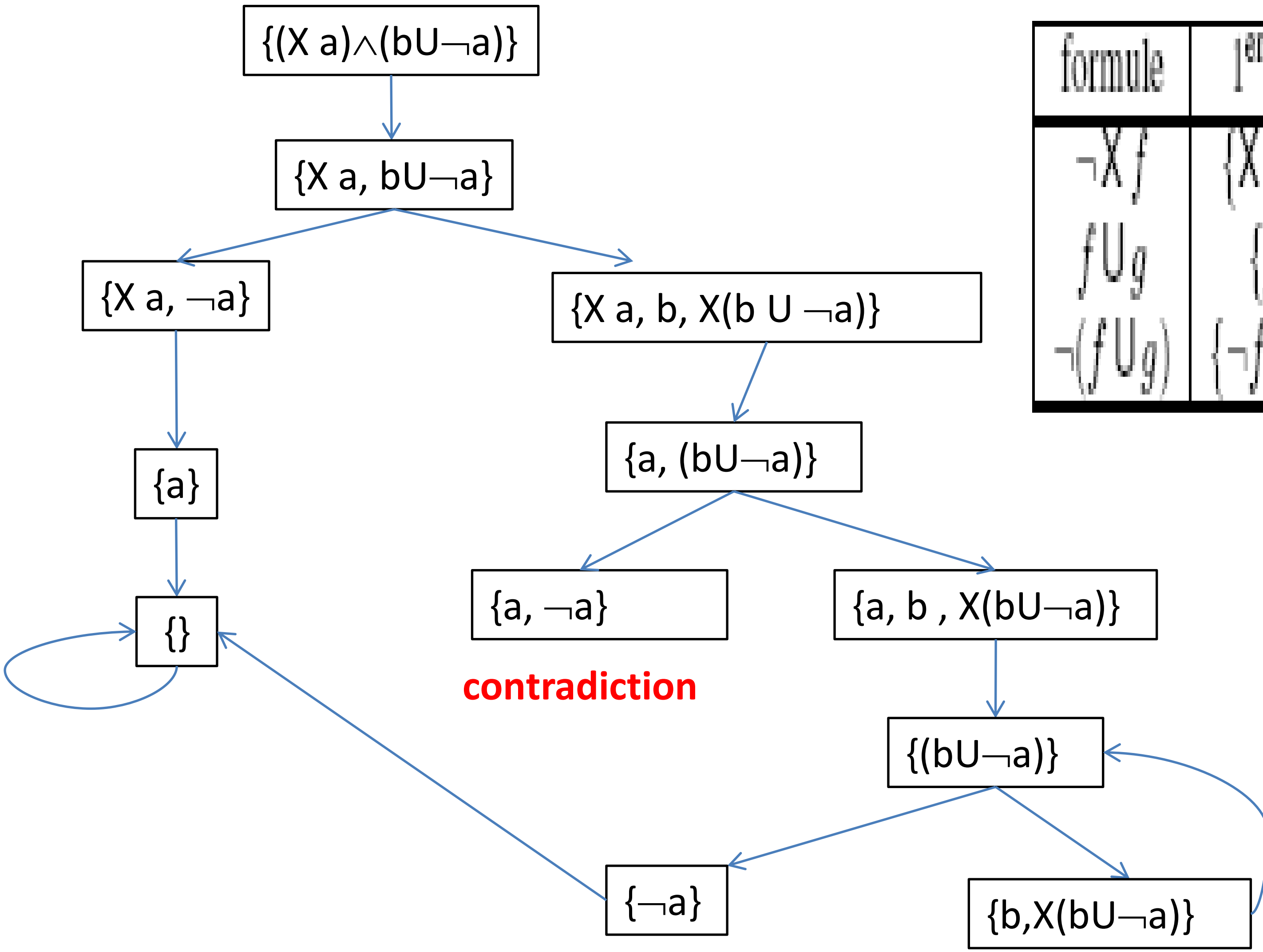
- Lors de la construction de l'arbre, un fils représente l'état suivant du système **uniquement** quand il s'agit de **l'opérateur X**;

$\{X\ \phi, X\varphi$, des atomes sans X….$\}$ L'instant présent

$\downarrow$

$\{\ \phi, \varphi\}$ L'instant suivant

Extension pour LTL
exemple: $(X\ a) \wedge (b U \neg a)$



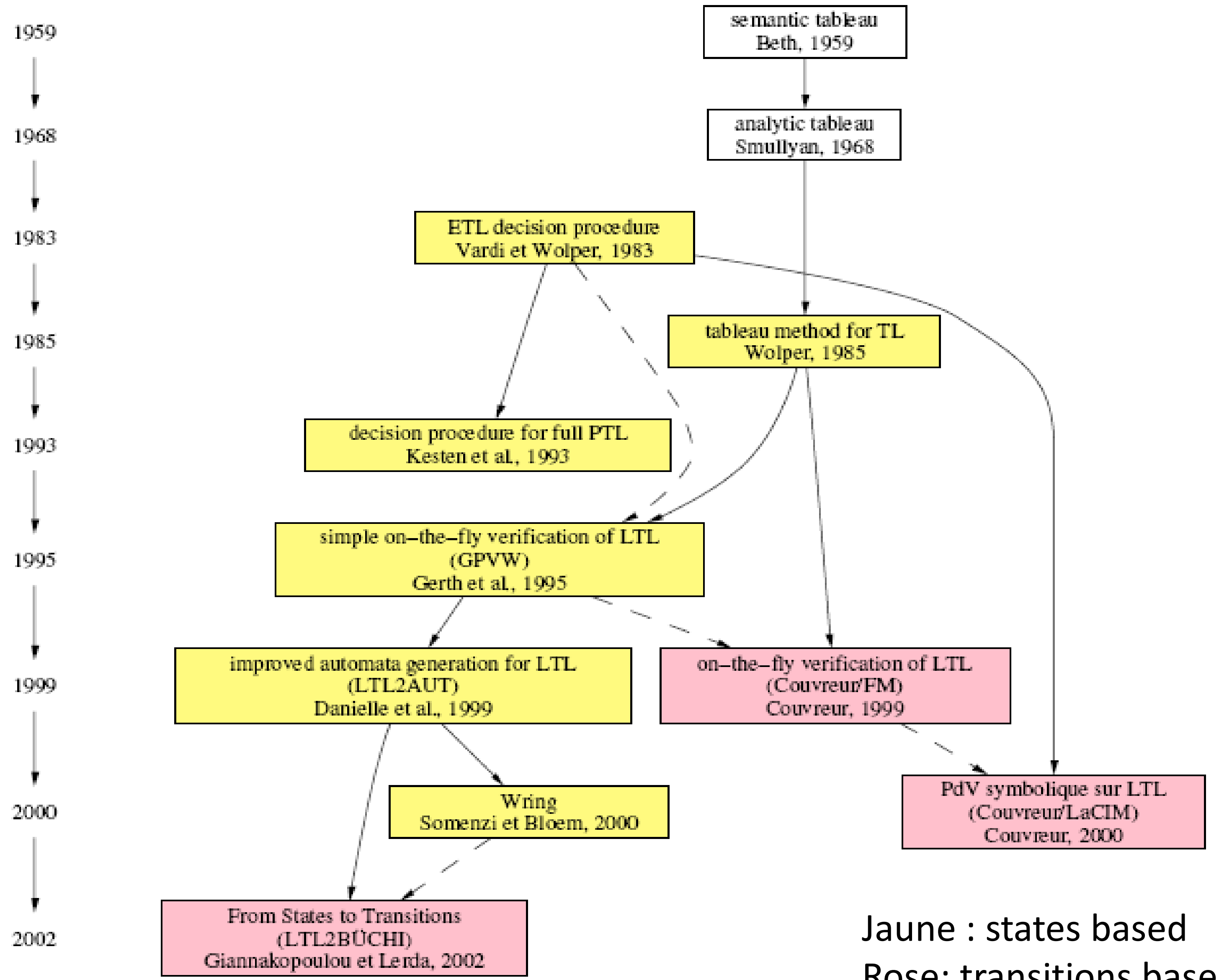
formule	1er fils	2e fils
$\neg X f$	$\{X \neg f\}$	
$f U g$	$\{g\}$	$\{f, X(f U g)\}$
$\neg(f U g)$	$\{\neg f, \neg g\}$	$\{\neg g, \neg X(f U g)\}$

Extension pour LTL
Satisfiabilité d’une formule

- Une formule est **valide** ssi la racine de son arbre est valide
- Un nœud est **invalide** si:

1) **contient une contradiction**
2) Ou bien **tous ses fils son invalide**
3) Si il **contient (aUb)** et **ni lui ni ses fils directes ou indirectes ne satisfait b**

LTL vers Büchi (historiques)



Jaune : states based
Rose: transitions based

LTL vers Büchi
Méthodes

- **Méthode 1** (**combinatoire**): elle produit un automate de Büchi étiqueté sur les états. (proposée en 1979). Nombre d'états<=2*taille formule;

- **Méthode des tableaux:**

1)(GPVW: Gerth, Peled, Vardi, et Wolper): basée sur la technique des tableaux. Un automate de Büchi généralisé **étiqueté sur les états**;

2) Technique de Couvreur (1999): génère un automate de Büchi **réduit** généralisé et **étiqueté sur les transitions** (arc entrant)

LTL vers Büchi
Technique: **GPVW**

- Construire le graphe représentant **l'arbre binaire** de la formule

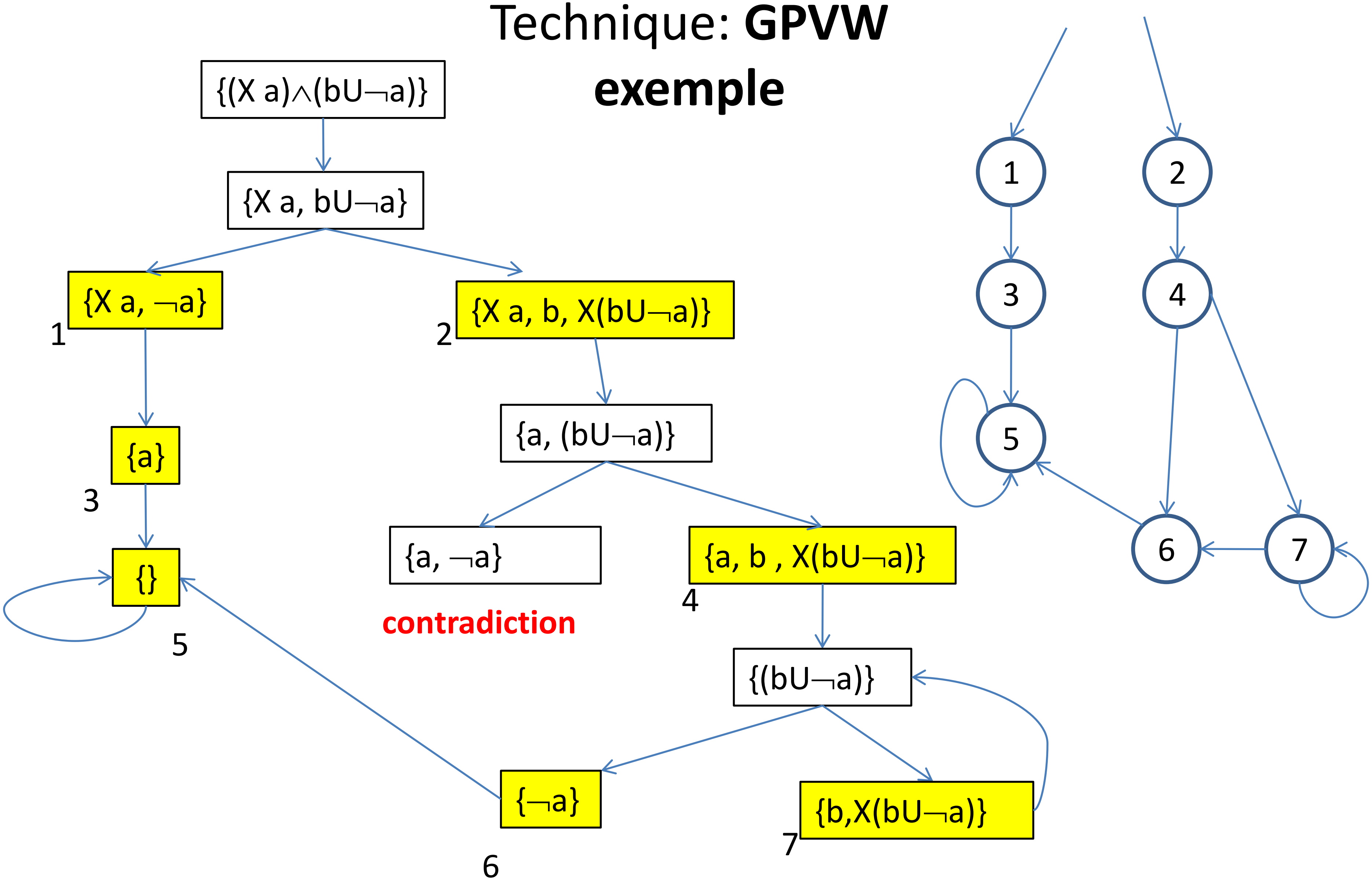
- Chaque **nœud cohérent** du graphe qui contient seulement : - **des formules atomique (ou leurs négations)**, et/ou **des formules précédées par X** correspond à un **état** dans l'automate

- **L'étiquette** de chaque nœud est **l'ensemble des ensemble d'atomes compatibles** avec les **formules du nœud**

LTL vers Büchi

Technique: **GPVW**

exemple



LTL vers Büchi

Technique: **GPVW**

exemple

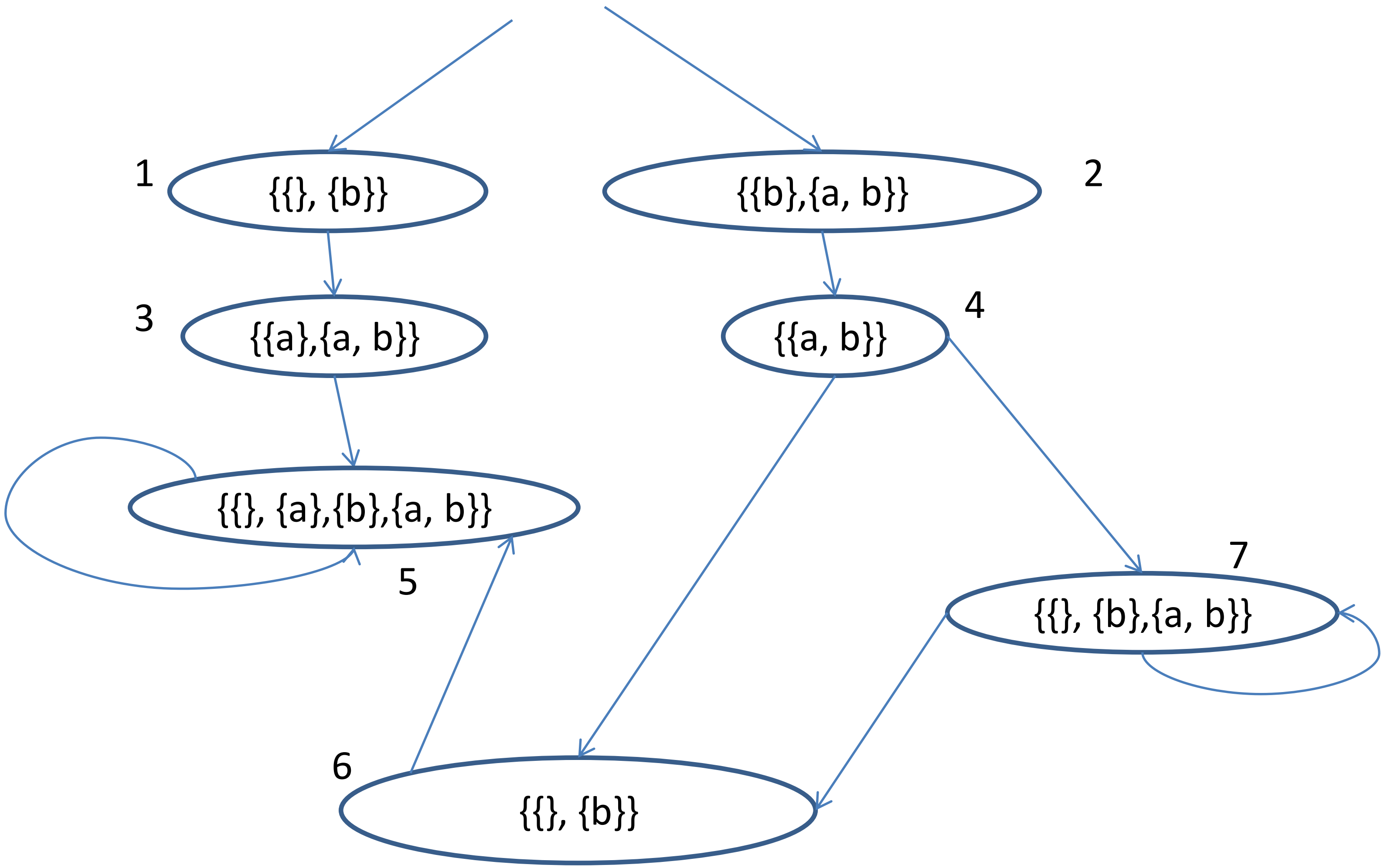
$2^{\{a, b\}}=\{\{\},\{a\},\{b\},\{a, b\}\}$

1	$\boxed{\{X\ a, \neg a\}}$	Étiquette (1)={{}, {b}}
2	$\boxed{\{X\ a, b, X(bU\neg a)\}}$	Étiquette (2)={{b},{a, b}}
3	$\boxed{\{a\}}$	Étiquette (3)={{a},{a, b}}
4	$\boxed{\{a, b, X(bU\neg a)\}}$	Étiquette (4)={{a, b}}
5	$\boxed{\{\}}$	Étiquette (5)={{}, {a},{b},{a, b}}
6	$\boxed{\{\neg a\}}$	Étiquette (6)={{}, {b}}
7	$\boxed{\{b, X(bU\neg a)\}}$	Étiquette (7)={{b},{a, b}}

LTL vers Büchi

Technique: **GPVW**

exemple



LTL vers Büchi

Technique: **GPVW**

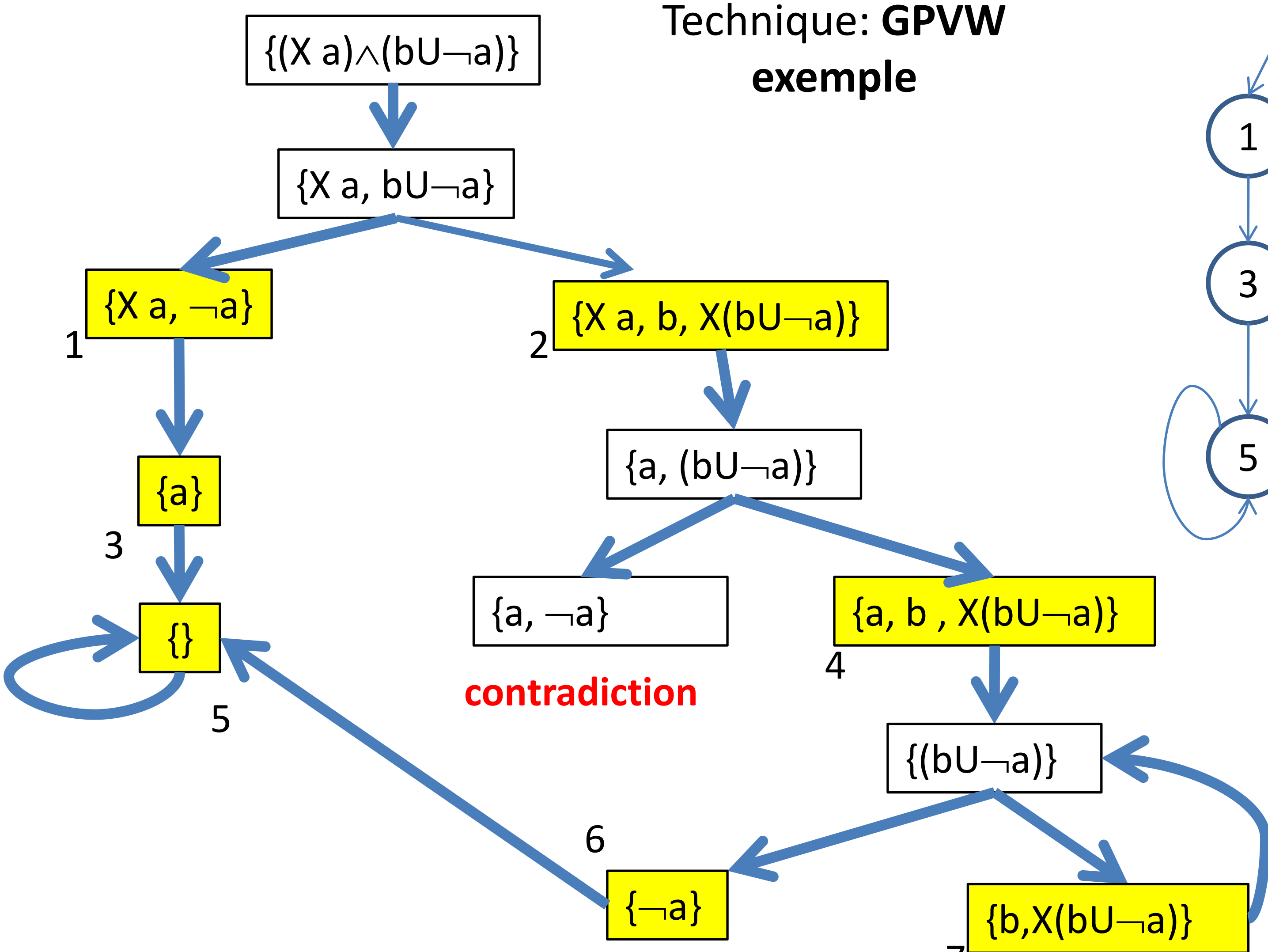
les ensembles d'états finaux

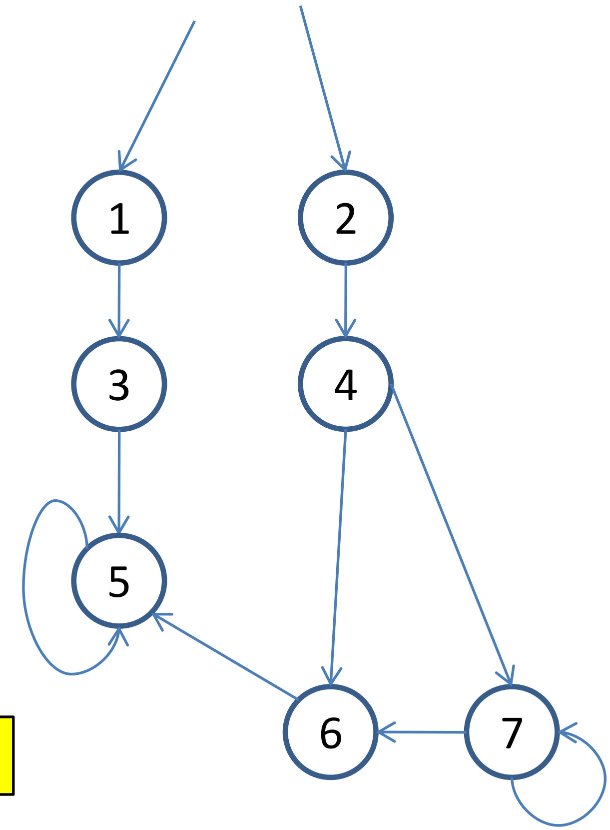
- L'état étiqueté **{}** est un **état final** de l'automate qui fait parti de **tous les ensembles d'états finaux**
- Pour chaque until : **(qUp)**, un ensemble d'états finaux $\mathbf{F_i}$ est définit.
- Les états de $\mathbf{F_i}$ sont ceux qui correspondent aux nœuds de l'arbre **contenant p** ou ne **contenant pas qUp**.

LTL vers Büchi

Technique: **GPVW**

exemple



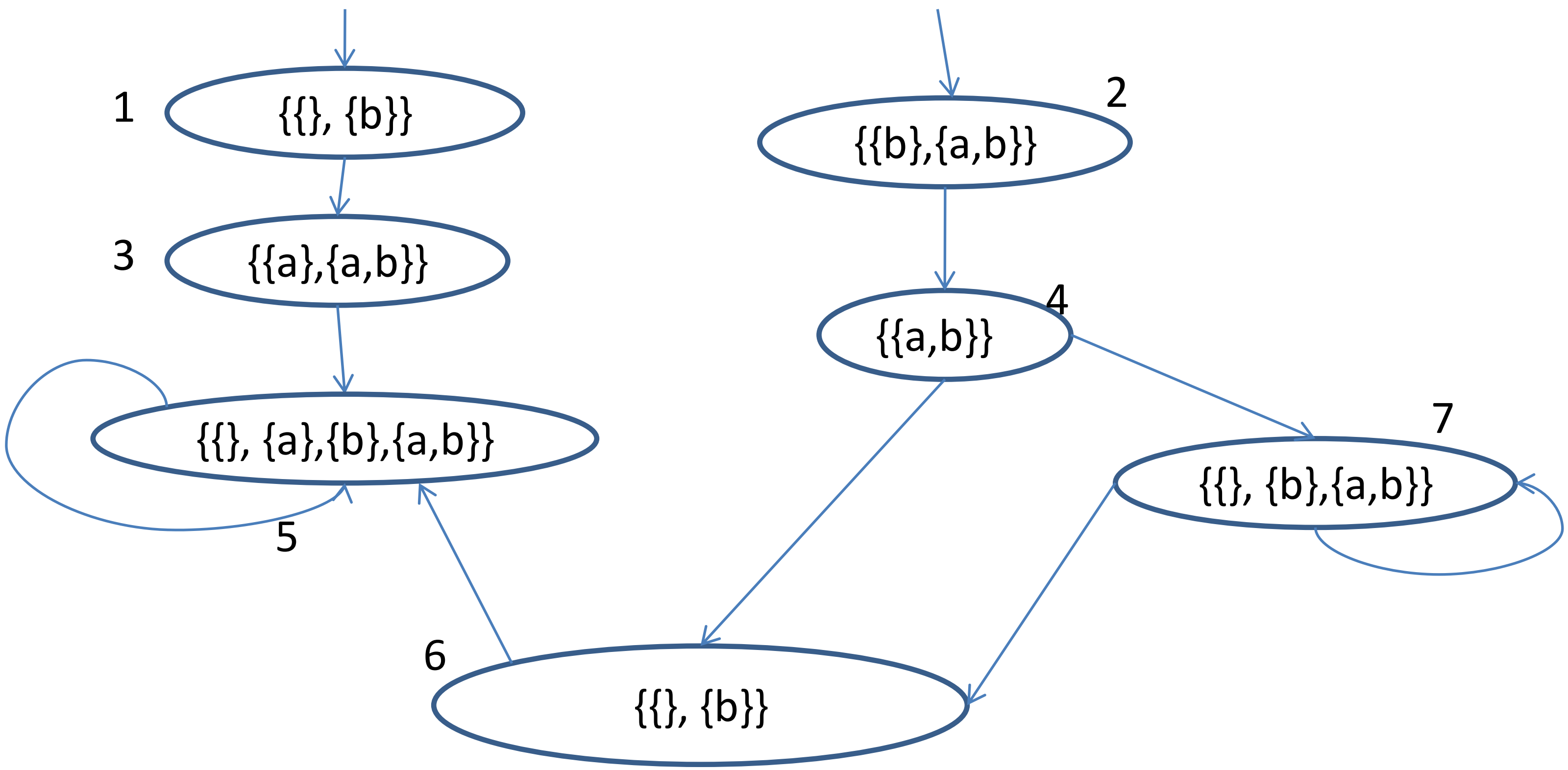


un seul ensemble d'états finaux F={1,3,5,6}

LTL vers Büchi

Technique: **GPVW**

exemple : les ensembles d'états finaux

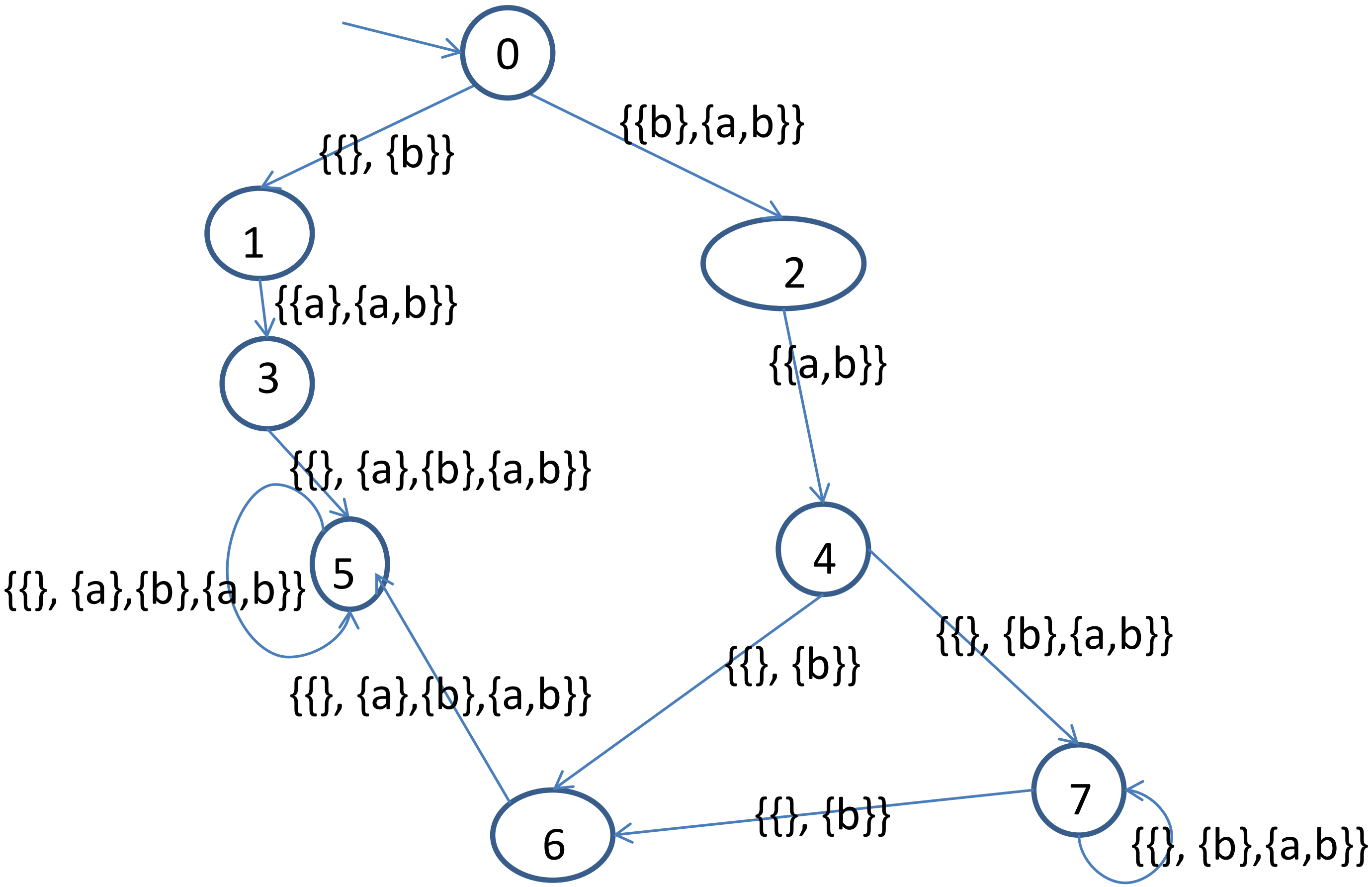


Un seul until : (bU¬a), donc un seul ensemble d'états finaux F={1,3,5,6}

LTL vers Büchi

Technique: **GPVW**

exemple : étiquettes sur les transitions



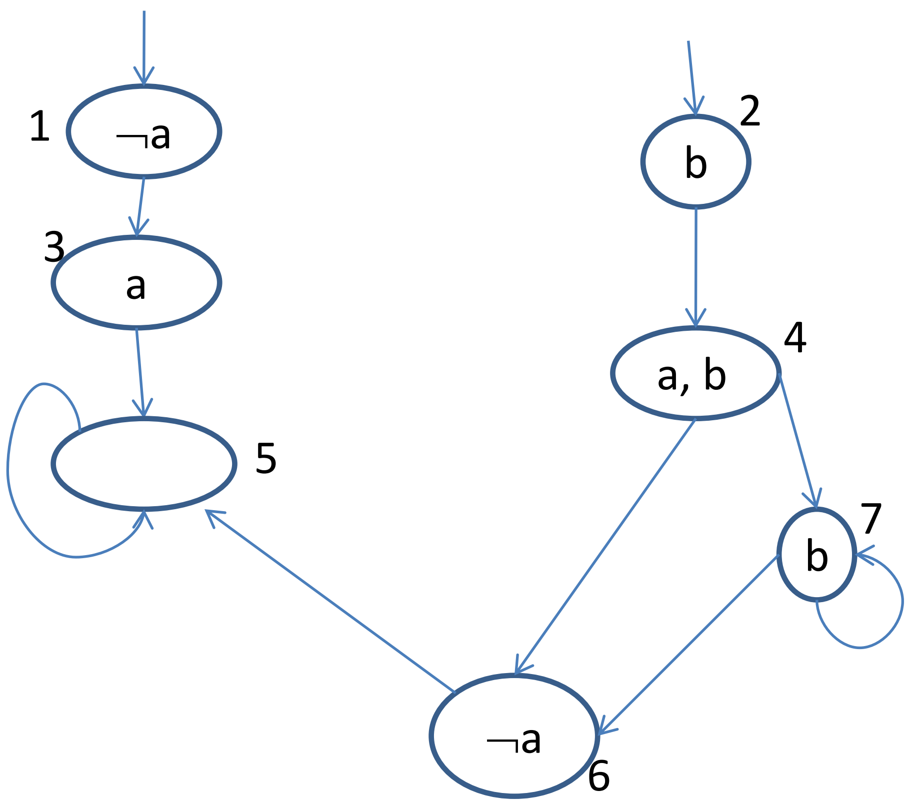
Penser à construire l'automate étiqueté sur les arcs entrants

LTL vers Büchi

Technique: **GPVW**

Remarques

- Il est possible d'étiqueter les états uniquement par les atomes des nœuds correspondant dans le tableau



L'état 5 n'exige pas la satisfaction d'un atome, donc on peut l'étiqueter par True

LTL vers Büchi

Technique: **Couvreur 1999**

- Technique de Couvreur (1999): génère un automate de Büchi **réduit** généralisé et étiqueté sur les transitions (arc entrant)

- À partir du même tableau sémantique, les états de l'automate sont maintenant :

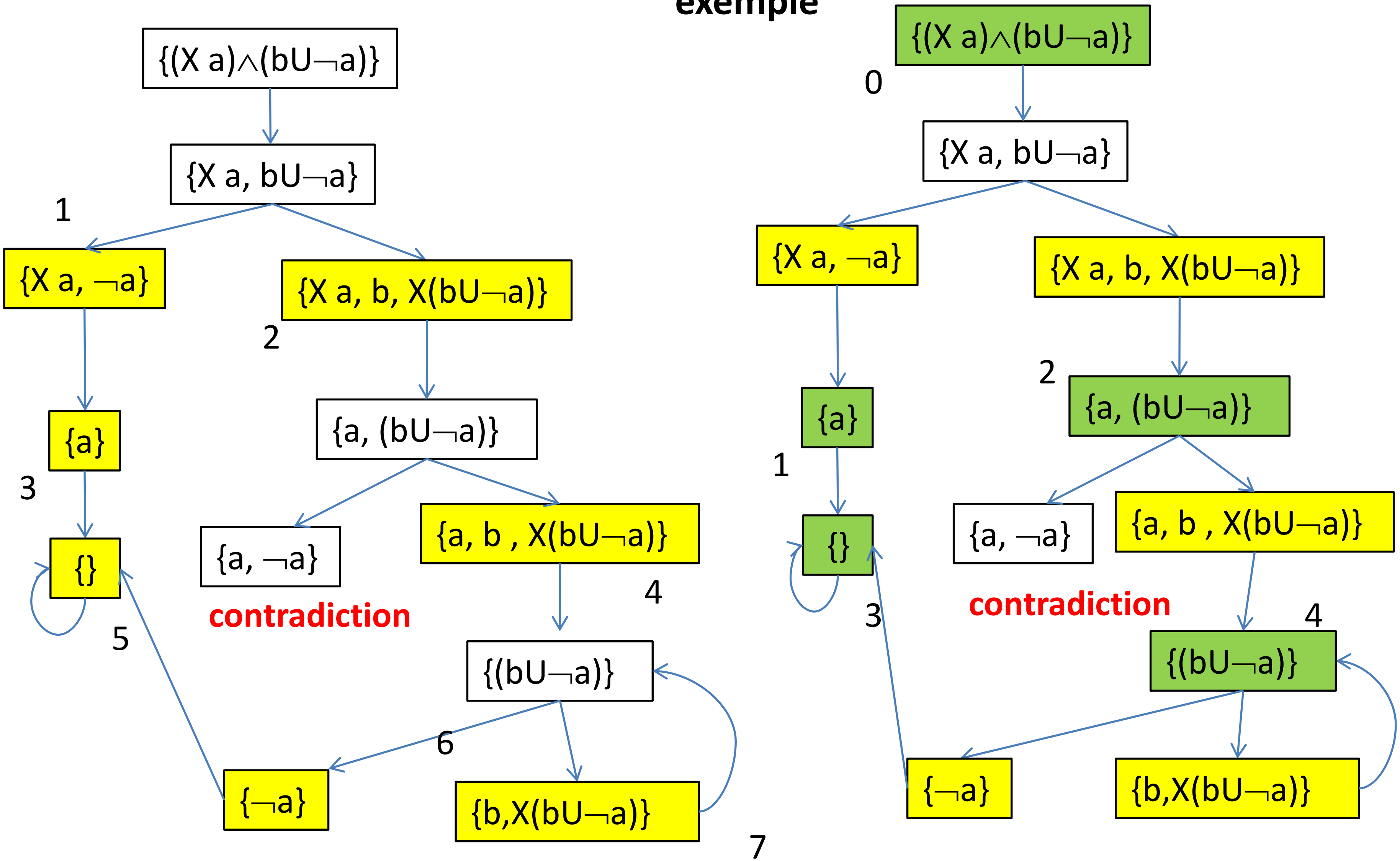
1) la **racine** de l'arbre est **l'état initial**

2) les nœuds successeurs immédiats des états (de la méthode précédente) sont les états de l'automate

LTL vers Büchi

Technique: **Couvreur 1999**

exemple

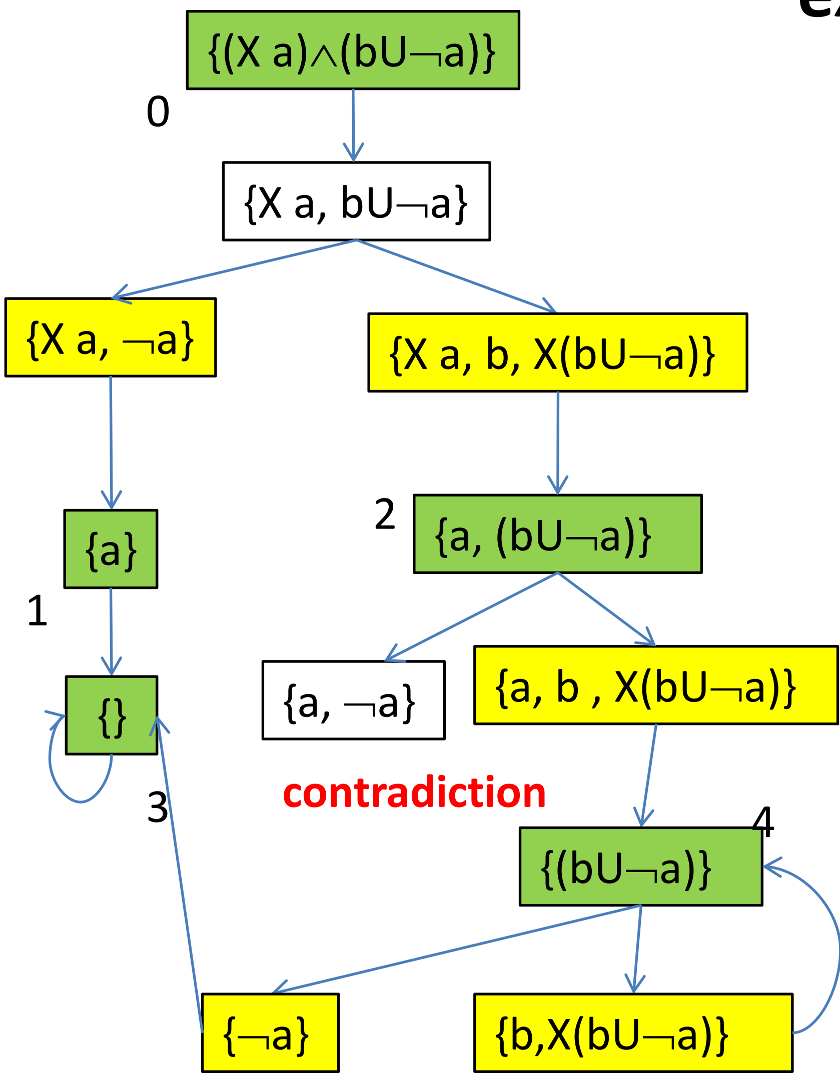


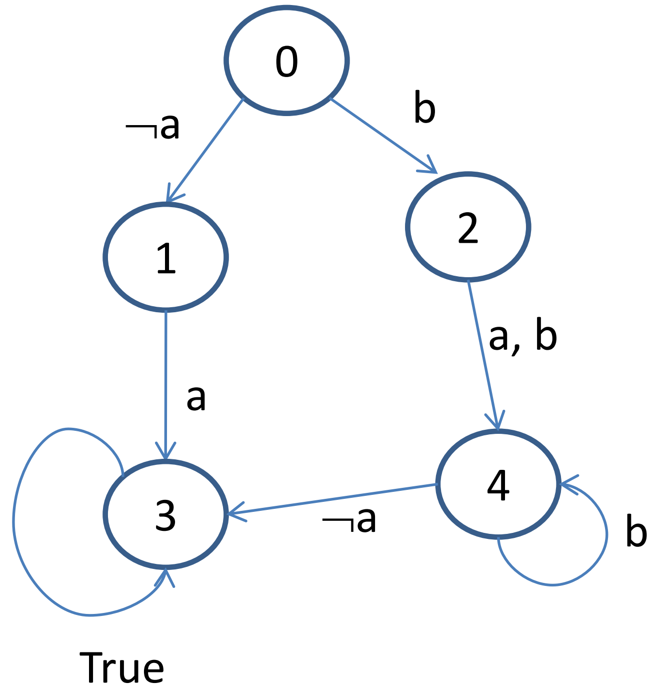
un seul ensemble d'états finaux F={1, 3, 5, 6}

LTL vers Büchi

Technique: **Couvreur 1999**

exemple





Fin

Questions?

Une question donc?

Transformer ça **SVP :**

formule	1^{er} fils	2^e fils
$\neg X f$	$\{X \neg f\}$	
$f \mathrm{U} g$	$\{g\}$	$\{f, X(f \mathrm{U} g)\}$
$\neg(f \mathrm{U} g)$	$\{\neg f, \neg g\}$	$\{\neg g, \neg X(f \mathrm{U} g)\}$

pUq

G(pUq)

(a → X b)U(G¬b)

Rq: G φ ≡ ¬F ¬φ ≡ ¬(**T** U ¬φ)