Vérification de formules de logique temporelle à temps linéaire

Alexandre Duret-Lutz

mars 2009

Menu du jour

1. Notations
   - ω-mots, expressions ω-rationnelles
2. Introduction à LTL
   - Logique des propositions : l'instant présent
   - F1S : Logique monadique du 1^{er} ordre à un successeur
   - S1S : Logique monadique du 2^{nd} ordre à un successeur
   - LTL : Logique Temporelle à temps Linéaire
3. Automates de Büchi, GBA et TGBA
   - Propriétés
4. Approche automate
5. Traduction de LTL en automates
   - Opérateurs X et U
   - Tableau
   - Tableau vers GBA
   - Tableau vers TGBA

ω : premier ordinal infini

Les entiers naturels peuvent être construits avec des ensembles :

$$0 = \{\}$$
$$1 = \{0\} = \{\{\}\}$$
$$2 = \{0, 1\} = \{\{\}, \{\{\}\}\}$$
$$3 = \{0, 1, 2\} = \{\{\}, \{\{\}\}, \{\{\}, \{\{\}\}\}\}$$
$$\vdots$$

Tout entier correspond à un ensemble.
L'inclusion sur les ensembles se traduit par un ordre sur les entiers.

ω : premier ordinal infini

Les entiers naturels peuvent être construits avec des ensembles :

$$
\begin{aligned}
0 &= \{\} \\
1 &= \{0\} = \{\{\}\} \\
2 &= \{0,1\} = \{\{\},\{\{\}\}\} \\
3 &= \{0,1,2\} = \{\{\},\{\{\}\},\{\{\},\{\{\}\}\}\} \\
&\vdots \\
n+1 &= n \cup \{n\} \\
&\vdots
\end{aligned}
$$

Tout entier correspond à un ensemble.
L'inclusion sur les ensembles se traduit par un ordre sur les entiers.

ω : premier ordinal infini

Les entiers naturels peuvent être construits avec des ensembles :

$$
\begin{aligned}
0 &= \{\} \\
1 &= \{0\} = \{\{\}\} \\
2 &= \{0,1\} = \{\{\},\{\{\}\}\} \\
3 &= \{0,1,2\} = \{\{\},\{\{\}\},\{\{\},\{\{\}\}\}\} \\
&\vdots \\
n+1 &= n \cup \{n\} \\
&\vdots \\
&= \mathbb{N}
\end{aligned}
$$

Tout entier correspond à un ensemble.
L'inclusion sur les ensembles se traduit par un ordre sur les entiers.

ω : premier ordinal infini

Les entiers naturels peuvent être construits avec des ensembles :

$$
\begin{aligned}
0 &= \{\} \\
1 &= \{0\} = \{\{\}\} \\
2 &= \{0,1\} = \{\{\},\{\{\}\}\} \\
3 &= \{0,1,2\} = \{\{\},\{\{\}\},\{\{\},\{\{\}\}\}\} \\
&\vdots \\
n+1 &= n \cup \{n\} \\
&\vdots \\
\omega &= \mathbb{N}
\end{aligned}
$$

Tout entier correspond à un ensemble.
L'inclusion sur les ensembles se traduit par un ordre sur les entiers.

ω : premier ordinal infini

Les ordinaux peuvent être construits avec des ensembles :

$$
\begin{aligned}
0 &= \{\} \\
1 &= \{0\} = \{\{\}\} \\
2 &= \{0,1\} = \{\{\},\{\{\}\}\} \\
3 &= \{0,1,2\} = \{\{\},\{\{\}\},\{\{\},\{\{\}\}\}\} \\
&\vdots \\
n+1 &= n \cup \{n\} \\
&\vdots \\
\omega &= \mathbb{N} \\
\omega+1 &= \mathbb{N} \cup \{\omega\}
\end{aligned}
$$

Tout ordinal correspond à un ensemble.
L'inclusion sur les ensembles se traduit par un ordre sur les ordinaux.
(Paradoxe : les ordinaux ne forment pas un ensemble.)

ω-mots

Soient Σ un alphabet (ensemble de lettres) et $n \in \mathbb{N} \cup \{\omega\}$ un ordinal.

Une séquence de taille n (ou n-mot) de Σ est une fonction $\sigma : [\![0, n[\![\mapsto \Sigma$ associant une lettre chaque entier naturel inférieur à n.

Notations :

- Σ^n l'ensemble des séquences de taille n,
- $\Sigma^\star$ l'ensemble des séquences finies ($n < \omega$),
- Σ^ω l'ensemble des séquences infinies ($n = \omega$),
- $\Sigma^\infty = \Sigma^\star \cup \Sigma^\omega$ l'ensemble des séquences ($n \leq \omega$),
- σ^i suffixe de σ commençant à la position i : $\sigma^i(j) = \sigma(i + j)$ pour les j tels que $i + j < n$,
- ε_Σ la séquence de taille nulle sur Σ.

Expressions ω-rationnelles

L'ensemble des expressions ω-rationnelles sur Σ, par induction :

- un mot $m \in \Sigma^\infty$ est une expression ω-rationnelle ;
- si w_1 et w_2 sont deux expressions ω-rationnelles, alors w_1^ω, w_1^*, $(w_1 + w_2)$ et $(w_1 \cdot w_2)$ sont des expressions ω-rationnelles.

$\mathscr{L}(w)$, le langage d'une expression ω-rationnelle w, est défini par

$$\begin{aligned}
\mathscr{L}(m) &= \{m\} \\
\mathscr{L}\big((w_1 + w_2)\big) &= \mathscr{L}(w_1) \cup \mathscr{L}(w_2) \\
\mathscr{L}\big((w_1 \cdot w_2)\big) &= \{m_1 \cdot m_2 \mid m_1 \in \mathscr{L}(w_1), m_2 \in \mathscr{L}(w_2)\} \\
\mathscr{L}(w_1^*) &= \{\varepsilon_\Sigma\} \cup \bigcup_{n \in \mathbb{N}} \{m_0 \cdot m_1 \cdots m_n \mid \forall i \leq n,\ m_i \in \mathscr{L}(w_1)\} \\
\mathscr{L}(w_1^\omega) &= \{m_0 \cdot m_1 \cdot m_2 \cdots \mid \forall i \in \mathbb{N},\ m_i \in \mathscr{L}(w_1)\}
\end{aligned}$$

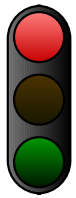
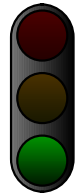
Logique des propositions : l'instant présent

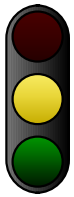
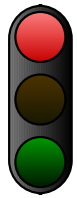
La logique propositionnelle peut caractériser un instant.

r : feu rouge allumé

o : feu orange allumé

v : feu vert allumé

$r \wedge o \wedge v =$  , $r \wedge \neg o \wedge \neg v =$  , $\neg r \wedge \neg o \wedge v =$  ,
$\neg r \wedge \neg o \wedge \neg v =$  .

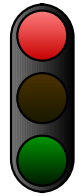
Comment dire que  précède  ?
Comment dire que le système ne reste pas toujours sur  ?

$\Rightarrow$ besoin de faire apparaître le temps

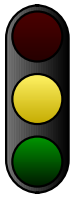
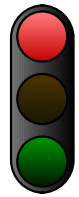
F1S : Logique monadique du 1er ordre à un succ.

Les prop. deviennent des prédicats unaires, paramétrés par le temps.

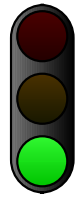
$r(t)$, $o(t)$, $v(t)$: feux allumés à l'instant t
$t+1$: instant successeur immédiat
$t \leq u$: ordre total sur les instants
$\exists t$, $\forall t$: quantificateurs du premier ordre

$\neg\forall t.(r(t) \wedge \neg o(t) \wedge \neg v(t))$: le système ne reste pas tout le temps  .

$\forall t.((\neg r(t) \wedge o(t) \wedge \neg v(t)) \rightarrow (r(t+1) \wedge \neg o(t+1) \wedge \neg v(t+1)))$:

toute configuration  est immédiatement suivie de  .

$\forall t.\exists u.(t \leq u) \wedge (\neg r(u) \wedge \neg o(u) \wedge v(u))$:

le système passe infiniment souvent par la configuration  .

S1S : Logique monadique du 2^{nd} ordre à un succ.

$r(t)$, $o(t)$, $v(t)$: feux allumés à l'instant t

0 : instant initial

$t+1$: instant successeur immédiat

$t \leq u$: ordre total sur les instants

$\exists t$, $\forall t$: quantificateurs du premier ordre

$\exists^2 X$, $\forall^2 X$: quantificateurs du second ordre

$t \in X$: appartenance d'une variable du premier ordre a une variable du second

$$\exists^2 X. \underbrace{(0 \in X \wedge (\forall t.(t \in X \rightarrow (\neg(t+1 \in X) \wedge (t+1+1 \in X)))))}_{Pair(X)}$$

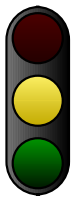
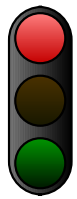
$\exists^2 X.Pair(X) \wedge \forall t.(t \in X \rightarrow r(t))$: le feu rouge doit toujours être allumé aux instants pairs.

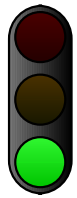
LTL : Logique Temporelle à temps Linéaire

Next	$\mathbf{X}\, f$	f est vraie à l'instant suivant
Always	$\mathbf{G}\, f$	f est vraie a tout instant
Eventually	$\mathbf{F}\, f$	f sera vraie à un instant (présent ou futur)
Until	$f\, \mathbf{U}\, g$	f est toujours vraie jusqu'à ce que g le soit

Équivalente à la logique monadique du premier ordre à un successeur.

$\neg\, \mathbf{G}(r \wedge \neg o \wedge \neg v)$: le système ne reste pas tout le temps  .

$\mathbf{G}((\neg r \wedge o \wedge \neg v) \rightarrow \mathbf{X}(r \wedge \neg o \wedge \neg v))$:  est tjs imm. suivi de  .

$\mathbf{G}\, \mathbf{F}(\neg r \wedge \neg o \wedge v)$: le système passe infiniment souvent par  .

LTL : Logique Temporelle à temps Linéaire

Next	$\mathbf{X}\,f$	f est vraie à l'instant suivant
Always	$\mathbf{G}\,f$	f est vraie a tout instant
Eventually	$\mathbf{F}\,f$	f sera vraie à un instant (présent ou futur)
Until	$f\,\mathbf{U}\,g$	f est toujours vraie jusqu'à ce que g le soit

F, **G** et **R** (Release) peuvent être vus comme du sucre :

$$\mathbf{F}\,f = \top\,\mathbf{U}\,f$$
$$f\,\mathbf{R}\,g = \neg(\neg f\,\mathbf{U}\,\neg g)$$
$$\mathbf{G}\,f = \neg\,\mathbf{F}\,\neg f = \neg(\top\,\mathbf{U}\,\neg f) = \bot\,\mathbf{R}\,f$$

D'autre part on a :

$$\neg\,\mathbf{X}\,f = \mathbf{X}\,\neg f$$
$$\neg\,\mathbf{F}\,f = \mathbf{G}\,\neg f \qquad \neg(f\,\mathbf{U}\,g) = (\neg f)\,\mathbf{R}(\neg g)$$
$$\neg\,\mathbf{G}\,f = \mathbf{F}\,\neg f \qquad \neg(f\,\mathbf{R}\,g) = (\neg f)\,\mathbf{U}(\neg g)$$

Interprétation sur une séquence

Pour toute proposition atomique p_i et toutes formules LTL f_1 et f_2, la satisfaction d'une formule LTL f par rapport à $\sigma \in (2^{AP})^\omega$ est notée $\sigma \models f$ et définie inductivement de la façon suivante :

$$
\begin{array}{ll}
\sigma \models p & \text{ssi } p \in \sigma(0) \\
\sigma \models \neg f_1 & \text{ssi } \neg(\sigma \models f_1) \\
\sigma \models f_1 \wedge f_2 & \text{ssi } \sigma \models f_1 \text{ et } \sigma \models f_2 \\
\sigma \models \mathbf{X}\, f_1 & \text{ssi } \sigma^1 \models f_1 \\
\sigma \models f_1\, \mathbf{U}\, f_2 & \text{ssi } \exists i \geq 0 \text{ tel que } \sigma^i \models f_2 \text{ et } \forall j \in [\![0, i-1]\!],\ \sigma^j \models f_1
\end{array}
$$

Le langage de la formule φ est l'ensemble des séquence infinies sur 2^{AP} qui satisfont φ.

$$\mathscr{L}_{AP}(\varphi) = \{\sigma \in (2^{AP})^\omega \mid \sigma \models \varphi\}$$

Lien avec F1S

Notons $p_1, p_2, \ldots$ les propositions atomiques de LTL, et $p_1(t), p_2(t), \ldots$ les prédicats correspondants en F1S. Une formule LTL f correspond à l'expression F1S $[f]_0$ définie inductivement de la façon suivante (où f_1 et f_2 sont des formules LTL) :

$$
\begin{aligned}
[p_i]_t &= p_i(t) \\
[\neg f_1]_t &= \neg [f_1]_t \\
[f_1 \wedge f_2]_t &= [f_1]_t \wedge [f_2]_t \\
[\mathbf{X}\, f_1]_t &= [f_1]_{t+1} \\
[f_1\, \mathbf{U}\, f_2]_t &= \exists u.((\forall v.((t \leq v) \wedge (v+1 \leq u)) \rightarrow [f_1]_v) \wedge [f_2]_u)
\end{aligned}
$$

(Le nom des variables u et v étant bien entendu choisi de façon unique dans le cas où plusieurs **U** sont imbriqués.)

On a $\forall \sigma \in (2^{AP})^\omega$, $\sigma \models f \iff \sigma \models [f]_0$.

Forme Normale Positive

$$\neg \mathbf{X}\, f = \mathbf{X}\, \neg f$$
$$\neg \mathbf{F}\, f = \mathbf{G}\, \neg f \qquad \neg(f\, \mathbf{U}\, g) = (\neg f)\, \mathbf{R} (\neg g)$$
$$\neg \mathbf{G}\, f = \mathbf{F}\, \neg f \qquad \neg(f\, \mathbf{R}\, g) = (\neg f)\, \mathbf{U} (\neg g)$$

Les négations ($\neg$, mais aussi $\rightarrow$ et $\leftrightarrow$) ne portent que sur les propositions atomiques.

$\neg \mathbf{G}(r \wedge \neg o \wedge \neg v) =$

$\neg \mathbf{G}\, \mathbf{F}(a \vee \neg b) =$

$\neg(a\, \mathbf{U}((b \leftrightarrow \mathbf{X}\, c)\, \mathbf{U}\, d)) =$

Forme Normale Positive

$$\neg \mathbf{X}\, f = \mathbf{X}\, \neg f$$
$$\neg \mathbf{F}\, f = \mathbf{G}\, \neg f \qquad \neg(f\, \mathbf{U}\, g) = (\neg f)\, \mathbf{R} (\neg g)$$
$$\neg \mathbf{G}\, f = \mathbf{F}\, \neg f \qquad \neg(f\, \mathbf{R}\, g) = (\neg f)\, \mathbf{U} (\neg g)$$

Les négations ($\neg$, mais aussi $\rightarrow$ et $\leftrightarrow$) ne portent que sur les propositions atomiques.

$\neg \mathbf{G}(r \wedge \neg o \wedge \neg v) = \mathbf{F}(\neg r \vee o \vee v)$

$\neg \mathbf{G}\, \mathbf{F}(a \vee \neg b) =$

$\neg(a\, \mathbf{U}((b \leftrightarrow \mathbf{X}\, c)\, \mathbf{U}\, d)) =$

Forme Normale Positive

$$\neg \mathbf{X}\, f = \mathbf{X}\, \neg f$$
$$\neg \mathbf{F}\, f = \mathbf{G}\, \neg f \qquad \neg(f\, \mathbf{U}\, g) = (\neg f)\, \mathbf{R} (\neg g)$$
$$\neg \mathbf{G}\, f = \mathbf{F}\, \neg f \qquad \neg(f\, \mathbf{R}\, g) = (\neg f)\, \mathbf{U} (\neg g)$$

Les négations ($\neg$, mais aussi $\rightarrow$ et $\leftrightarrow$) ne portent que sur les propositions atomiques.

$\neg \mathbf{G}(r \wedge \neg o \wedge \neg v) = \mathbf{F}(\neg r \vee o \vee v)$

$\neg \mathbf{G}\, \mathbf{F}(a \vee \neg b) = \mathbf{F}\, \mathbf{G}(\neg a \wedge b)$

$\neg(a\, \mathbf{U}((b \leftrightarrow \mathbf{X}\, c)\, \mathbf{U}\, d)) =$

Forme Normale Positive

$$\neg \mathbf{X}\, f = \mathbf{X}\, \neg f$$
$$\neg \mathbf{F}\, f = \mathbf{G}\, \neg f \qquad \neg(f\, \mathbf{U}\, g) = (\neg f)\, \mathbf{R}(\neg g)$$
$$\neg \mathbf{G}\, f = \mathbf{F}\, \neg f \qquad \neg(f\, \mathbf{R}\, g) = (\neg f)\, \mathbf{U}(\neg g)$$

Les négations ($\neg$, mais aussi $\rightarrow$ et $\leftrightarrow$) ne portent que sur les propositions atomiques.

$\neg \mathbf{G}(r \wedge \neg o \wedge \neg v) = \mathbf{F}(\neg r \vee o \vee v)$

$\neg \mathbf{G}\, \mathbf{F}(a \vee \neg b) = \mathbf{F}\, \mathbf{G}(\neg a \wedge b)$

$\neg(a\, \mathbf{U}((b \leftrightarrow \mathbf{X}\, c)\, \mathbf{U}\, d)) = (\neg a)\, \mathbf{R}(((\neg b \wedge \mathbf{X}\, c) \vee (b \wedge \mathbf{X}\, \neg c))\, \mathbf{R}(\neg d))$

Julius Richard Büchi (1924–1984)

J. Richard Büchi, 1983

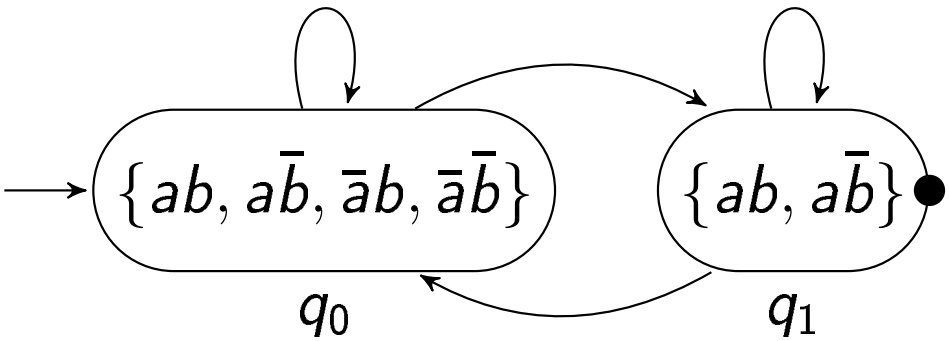
Logicien et mathématicien suisse.
Thèse à Zürich en 1950, s'installe aux USA ensuite.
Montre la décidabilité de S1S.

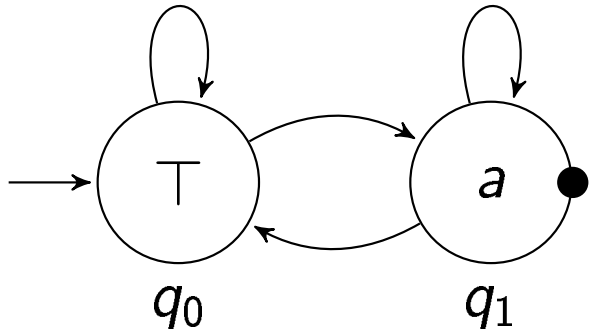
Automates de Büchi

Un automate de Büchi est un sextuplet $A = \langle \Sigma, \mathcal{Q}, \mathcal{Q}^0, \mathcal{F}, \delta, l \rangle$ où

- Σ est un alphabet,
- $\mathcal{Q}$ est un ensemble fini d'états,
- $\mathcal{Q}^0 \subseteq \mathcal{Q}$ est un ensemble d'états initiaux,
- $\mathcal{F} \subseteq \mathcal{Q}$ est un ensemble d'états d'acceptation,
- $\delta : \mathcal{Q} \mapsto 2^{\mathcal{Q}}$ est une fonction indiquant les successeurs d'un état,
- $l : \mathcal{Q} \mapsto 2^{\Sigma} \setminus \{\emptyset\}$ étiquette chaque état par un ensemble non-vide de lettres.

Exemple avec $AP = \{a, b\}$, $\Sigma = 2^{AP}$:





Automates de Büchi : langage

Les chemins de A :

$$\mathrm{Run}(A) = \{q_0 \cdot q_1 \cdot q_2 \cdots \in \mathcal{Q}^\omega \mid q_0 \in \mathcal{Q}^0 \text{ et } \forall i \geq 0,\ q_{i+1} \in \delta(q_i)\}$$

Les chemins acceptants de A sont ceux qui traversent des états d'acceptation infiniment souvent :

$$\mathrm{Acc}(A) = \{r \in \mathrm{Run}(A) \mid \forall i \geq 0,\ \exists j \geq i,\ r(j) \in \mathcal{F}\}$$

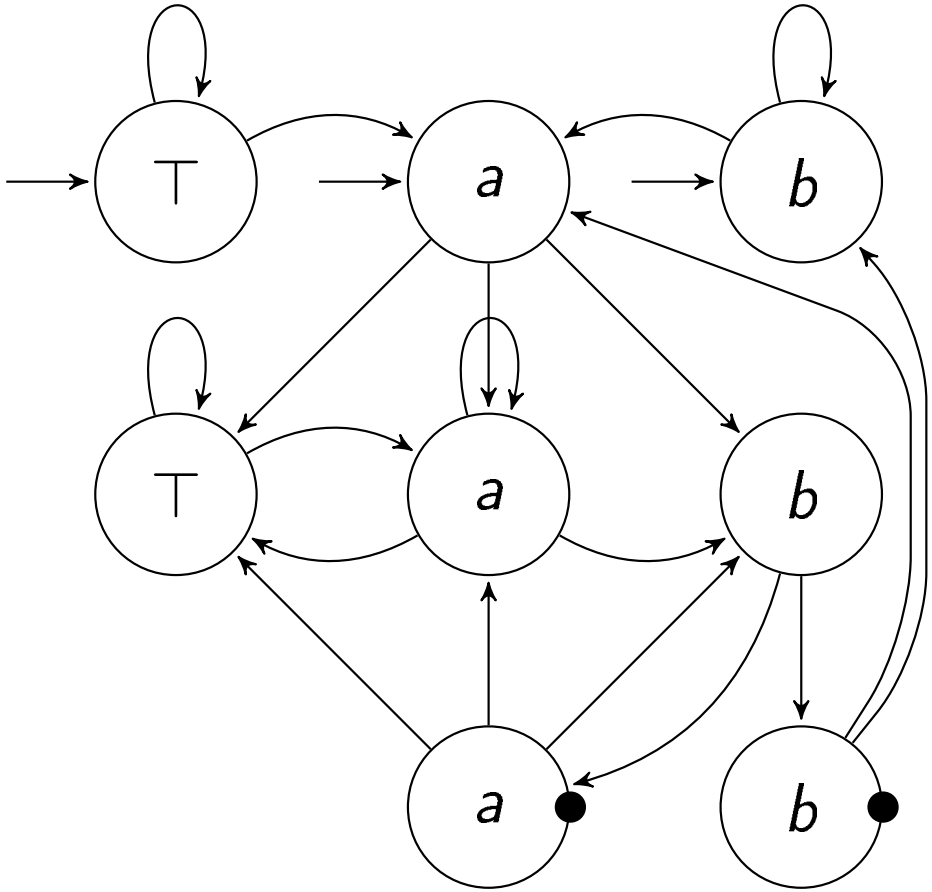
Un exécution de A est une séquence $\sigma \in \Sigma^\omega$ pour laquelle il existe un chemin acceptant $q_0 \cdot q_1 \cdots \in \mathrm{Acc}(A)$ dont les étiquettes en contiennent les lettres : $\forall i \in \mathbb{N},\ \sigma(i) \in l(q_i)$.

Le langage de A est l'ensemble des exécutions de A :

$$\mathscr{L}(A) = \{\sigma \in \Sigma^\omega \mid \exists q_0 \cdot q_1 \cdot q_2 \cdots \in \mathrm{Acc}(A),\ \forall i \in \mathbb{N},\ \sigma(i) \in l(q_i)\}$$

Automate de Büchi : plus d'états acceptants

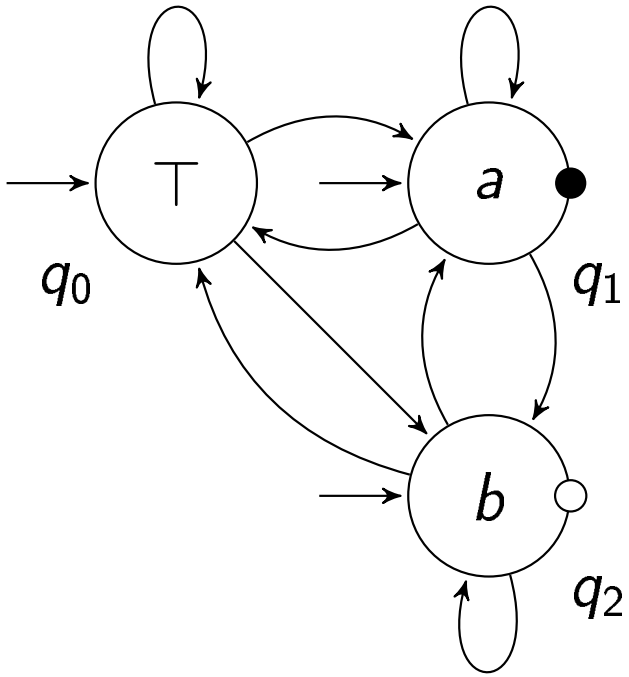
Exemple avec $AP = \{a, b\}$, $\Sigma = 2^{AP}$:



Automate de Büchi généralisé (GBA)

C'est un sextuplet $A = \langle \Sigma, \mathcal{Q}, \mathcal{Q}^0, \mathcal{F}, \delta, l \rangle$ où

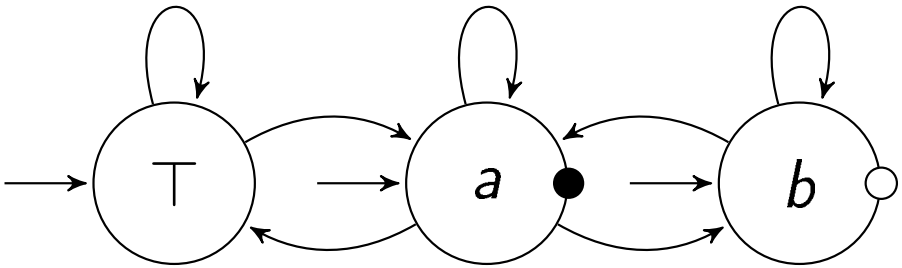
- $\mathcal{F} \subseteq 2^{\mathcal{Q}}$ est un ensemble d'ensembles d'états d'acceptation,



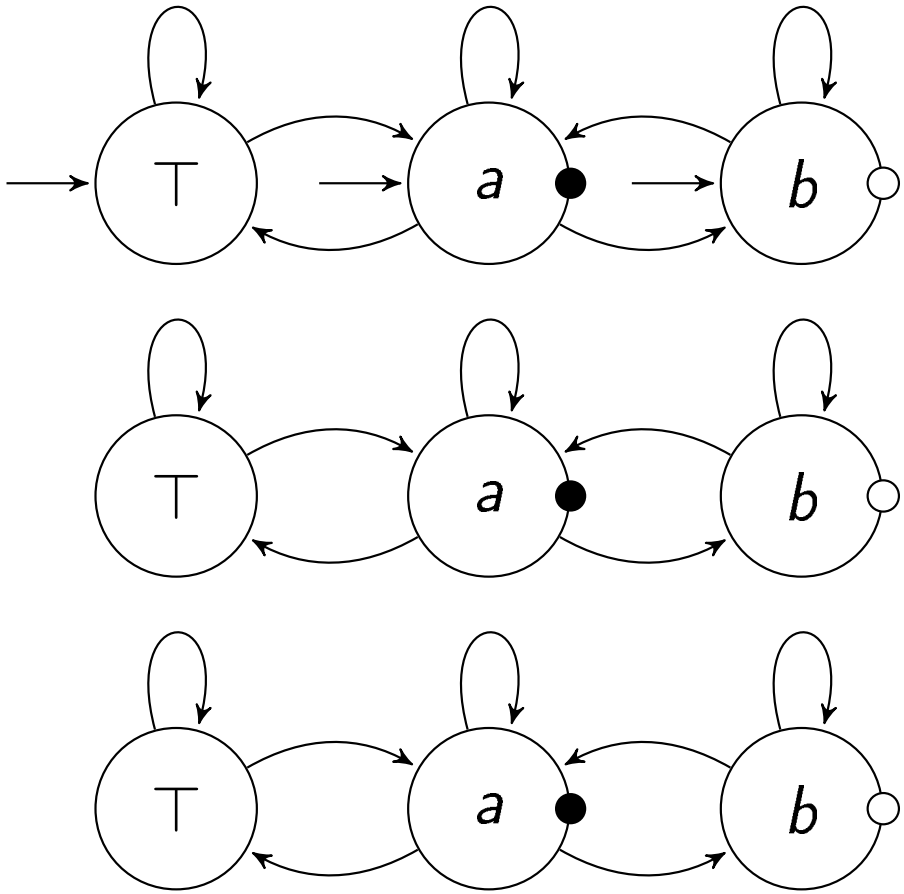
$$\mathrm{Acc}(A) = \{r \in \mathrm{Run}(A) \mid \forall F \in \mathcal{F}, \forall i \geq 0, \exists j \geq i, r(j) \in F\}$$

$$\mathscr{L}(A) = \{\sigma \in \Sigma^\omega \mid \exists q_0 \cdot q_1 \cdot q_2 \cdots \in \mathrm{Acc}(A), \forall i \in \mathbb{N}, \sigma(i) \in l(q_i)\}$$

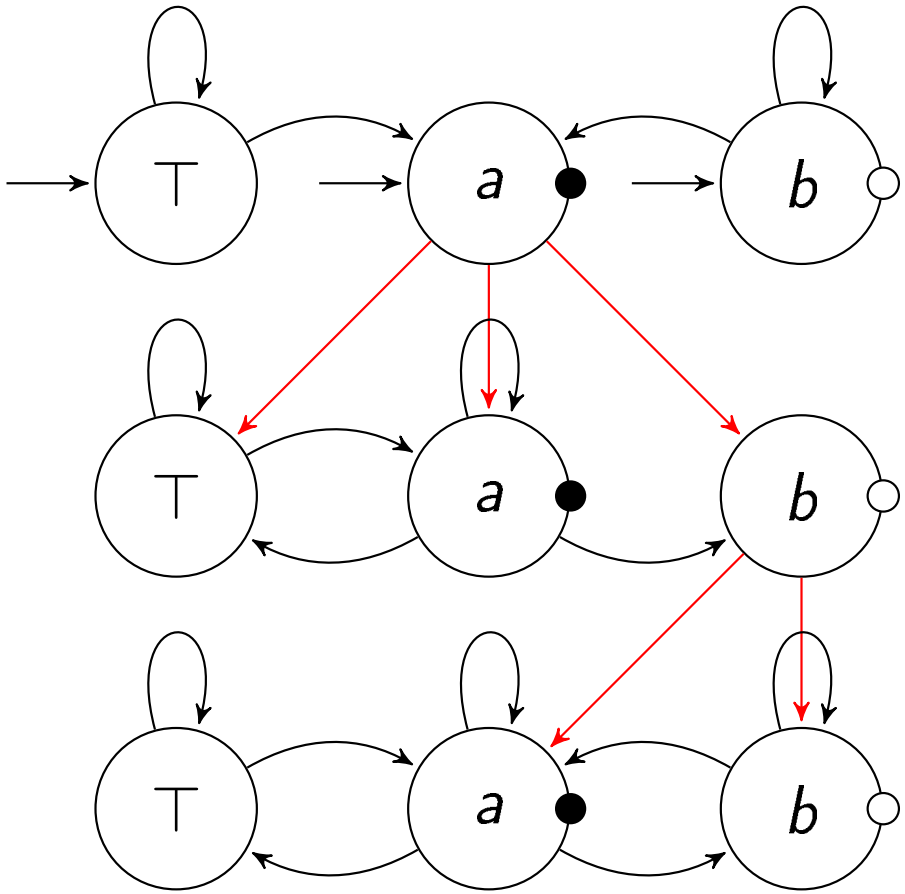
Dégénéralisation : exemple



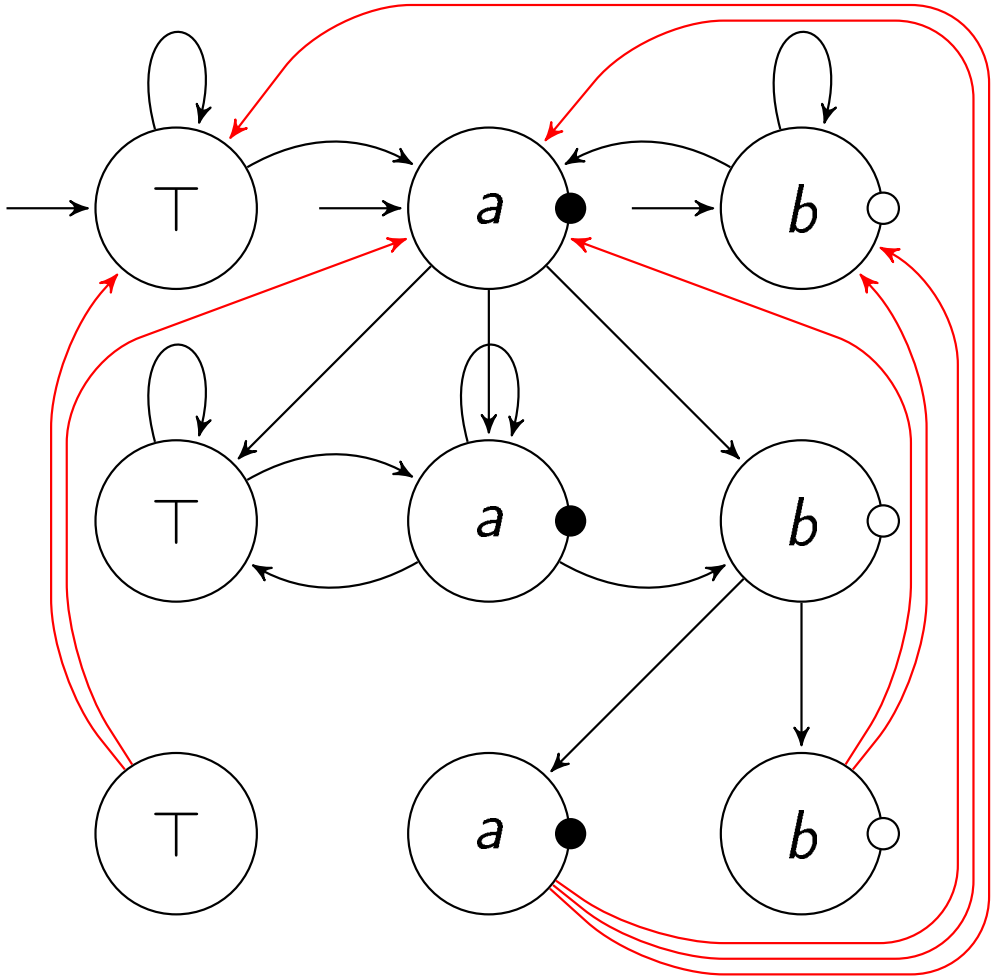
Dégénéralisation : exemple



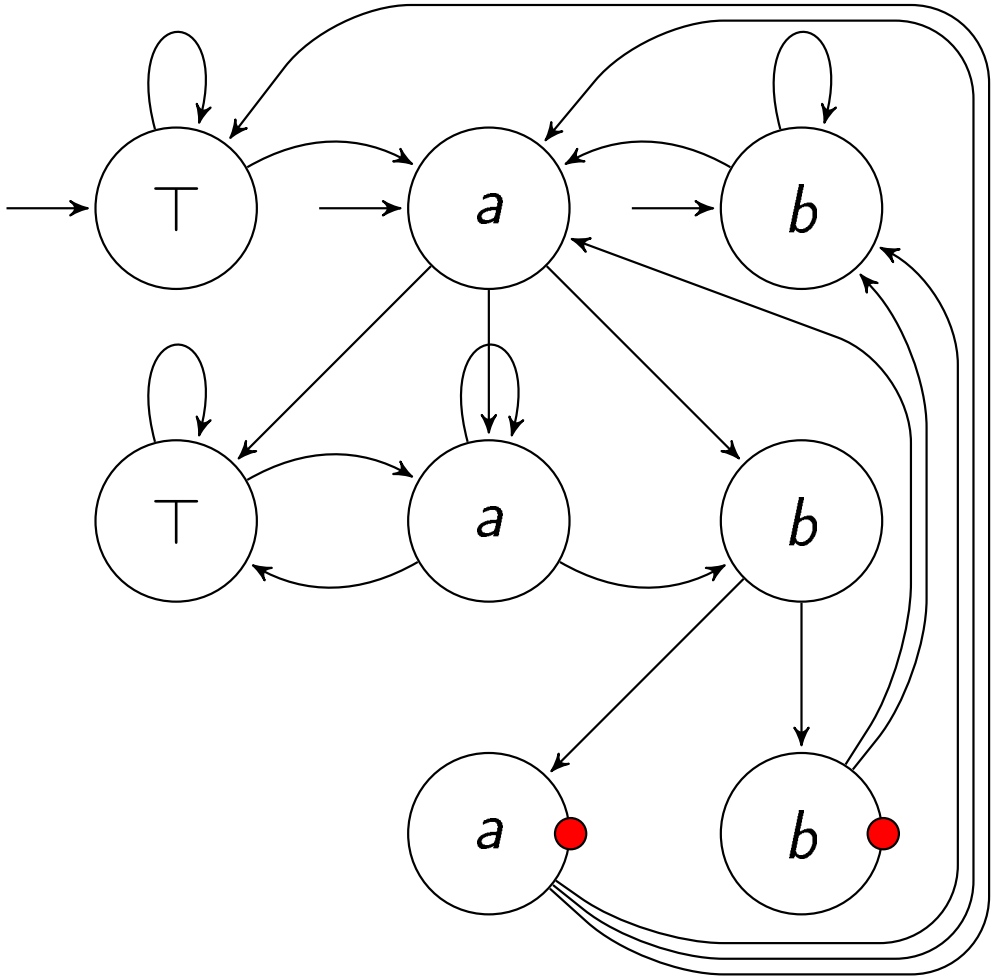
Dégénéralisation : exemple



Dégénéralisation : exemple



Dégénéralisation : exemple



Dégénéralisation : définition

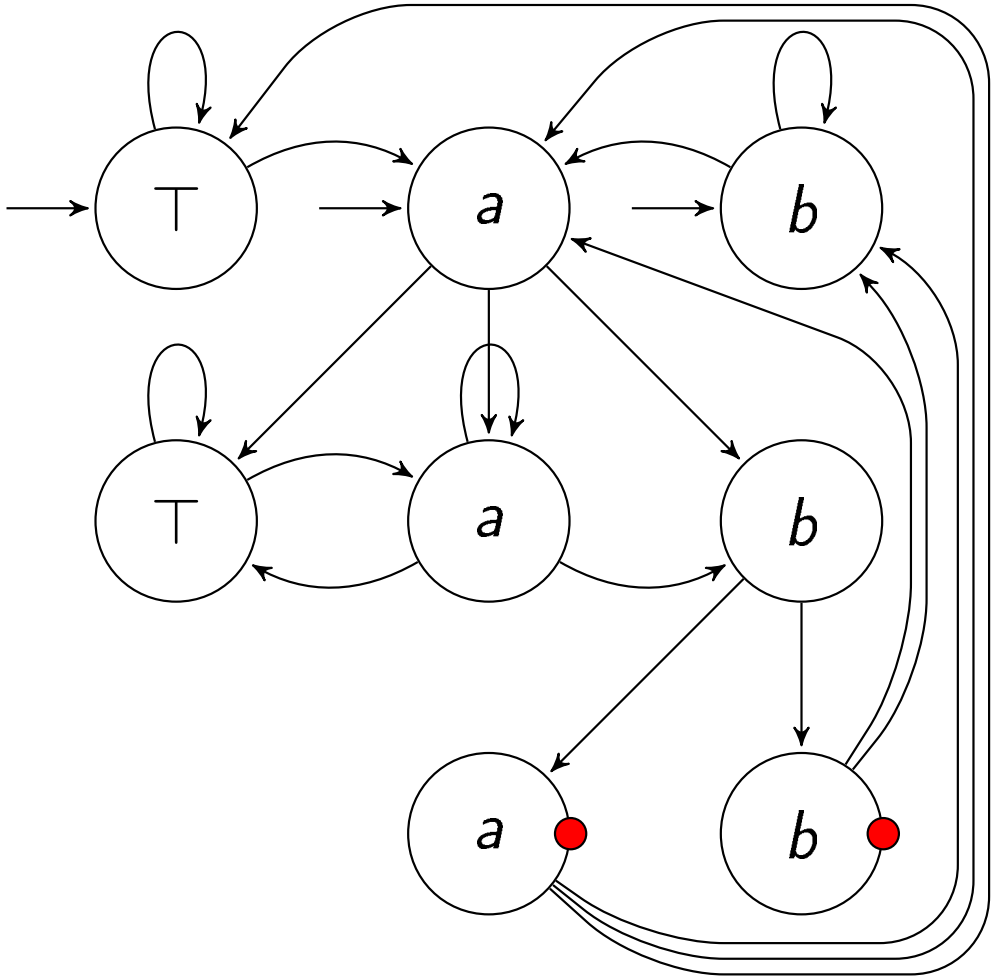
Un automate de Büchi généralisé $A = \langle \Sigma, Q, Q^0, \{\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_{r-1}\}, \delta, l \rangle$ peut être converti en un automate de Büchi non-généralisé $A' = \langle \Sigma, Q', Q'^0, \mathcal{F}', \delta', l' \rangle$ où

- $Q' = Q \times [\![0, r]\!]$
- $Q'^0 = Q \times \{0\}$
- $\mathcal{F}' = Q \times \{r\}$
- $\forall (q,j) \in Q',\ l'((q,j)) = l(q)$
- $\forall (q,j) \in Q',\ \delta'((q,j)) = \{(q', \beta_j(q)) \mid q' \in \delta(q)\}$ avec

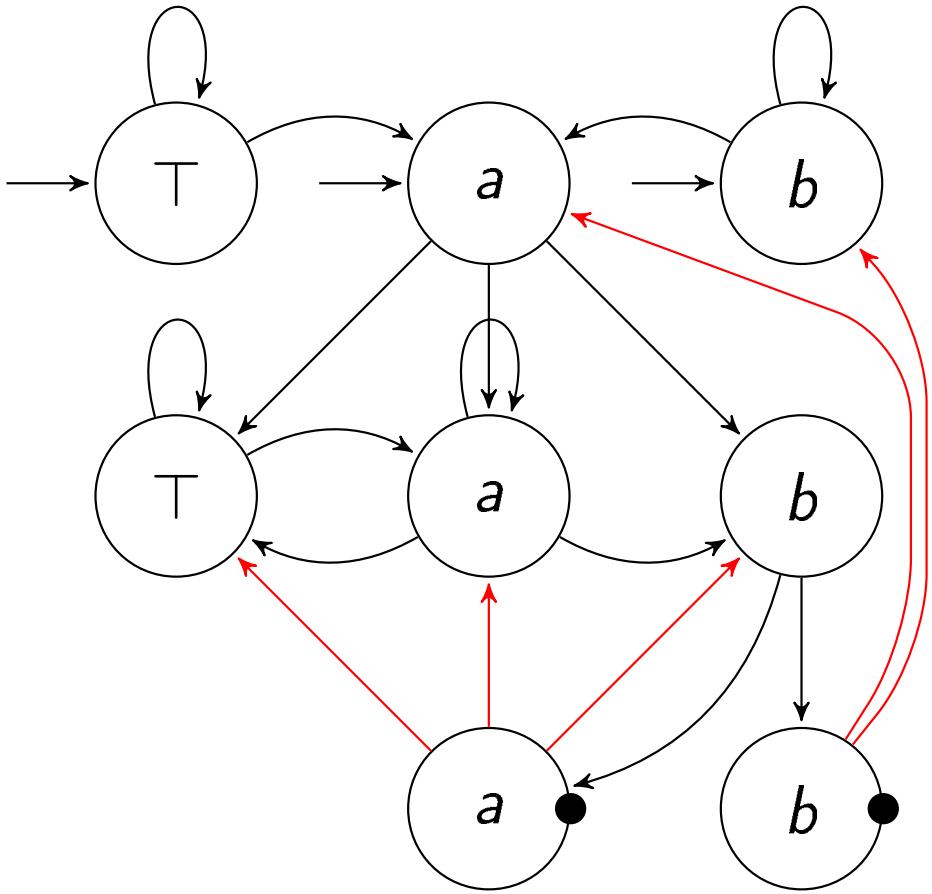
$$\beta_j(q) = \begin{cases} 0 & \text{si } j = r \\ j+1 & \text{si } q \in \mathcal{F}_j \\ j & \text{sinon} \end{cases}$$

Si A possède n états accessibles, A' en possède au pire $n(r+1)$.

Dégénéralisation : exemple



Dégénéralisation : exemple



Dégénéralisation : définition

Un automate de Büchi généralisé $A = \langle \Sigma, Q, Q^0, \{\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_{r-1}\}, \delta, l \rangle$ peut être converti en un automate de Büchi non-généralisé $A' = \langle \Sigma, Q', Q'^0, \mathcal{F}', \delta', l' \rangle$ où

- $Q' = Q \times [\![0, r]\!]$
- $Q'^0 = Q \times \{0\}$
- $\mathcal{F}' = Q \times \{r\}$
- $\forall (q,j) \in Q',\ l'((q,j)) = l(q)$
- $\forall (q,j) \in Q',\ \delta'((q,j)) = \{(q', \beta_j(q)) \mid q' \in \delta(q)\}$ avec

$$\beta_j(q) = \begin{cases} 0 & \text{si } j = r \\ j+1 & \text{si } q \in \mathcal{F}_j \\ j & \text{sinon} \end{cases}$$

Si A possède n états accessibles, A' en possède au pire $n(r+1)$.

Dégénéralisation : définition

Un automate de Büchi généralisé $A = \langle \Sigma, \mathcal{Q}, \mathcal{Q}^0, \{\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_{r-1}\}, \delta, l \rangle$ peut être converti en un automate de Büchi non-généralisé $A' = \langle \Sigma, \mathcal{Q}', \mathcal{Q}'^0, \mathcal{F}', \delta', l' \rangle$ où

- $\mathcal{Q}' = \mathcal{Q} \times [\![0, r]\!]$
- $\mathcal{Q}'^0 = \mathcal{Q} \times \{0\}$
- $\mathcal{F}' = \mathcal{Q} \times \{r\}$
- $\forall (q,j) \in Q',\ l'((q,j)) = l(q)$
- $\forall (q,j) \in Q',\ \delta'((q,j)) = \{(q', \beta_j(q)) \mid q' \in \delta(q)\}$ avec

$$\beta_j(q) = \begin{cases} j & \text{si } j < r,\ q \notin \mathcal{F}_j, \\ \max\{n \in [\![j, r]\!] \mid \forall k \in [\![j, n]\!], q \in \mathcal{F}_k\} & \text{si } j < r,\ q \in \mathcal{F}_j, \\ 0 & \text{si } j = r,\ q \notin \mathcal{F}_0, \\ \max\{n \in [\![0, r]\!] \mid \forall k \in [\![0, n]\!], q \in \mathcal{F}_k\} & \text{si } j = r,\ q \in \mathcal{F}_0 \end{cases}$$

Si A possède n états accessibles, A' en possède au pire $n(r+1)$.

TGBA : GBA étiquetés sur les transitions

Un automate de Büchi généralisé étiqueté sur les transitions (TGBA) est un automate de Büchi dans lequel les étiquettes sont portées par les transitions et où les conditions d'acceptation de Büchi généralisées portent sur les transitions. C'est-à-dire un quintuplet $A = \langle \Sigma, \mathcal{Q}, \mathcal{Q}^0, \mathcal{F}, \delta \rangle$ où

- Σ est un alphabet,
- $\mathcal{Q}$ est un ensemble fini d'états,
- $\mathcal{Q}^0 \subseteq \mathcal{Q}$ est l'ensemble des états initiaux,
- $\mathcal{F}$ est un ensemble fini d'éléments appelés conditions d'acceptation,
- $\delta \subseteq \mathcal{Q} \times (2^\Sigma \setminus \{\emptyset\}) \times 2^\mathcal{F} \times \mathcal{Q}$ est la relation de transition de l'automate (chaque transition étant étiquetée par une formule propositionnelle ainsi qu'un ensemble de conditions d'acceptation).

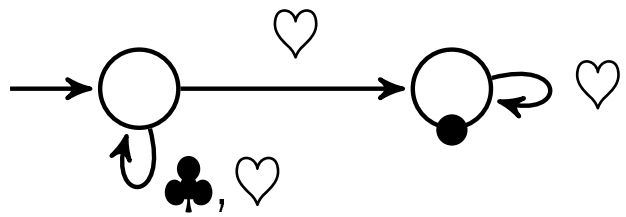
Propriétés des automates de Büchi

Les automates de Büchi, étiquetés sur les états ou les transitions, avec états ou transitions d'acceptation, généralisés ou non, sont tous aussi expressifs. I.e., ils peuvent reconnaître les mêmes langages (pas forcement avec autant d'états ou de transitions).
D'autre part les langages reconnaissables par des automates de Büchi

- sont clos par union (évident)
- sont clos par intersection (produit synchronisé)
- sont clos par complémentation (difficile à montrer)
  Büchi (1960) : construction en $2^{2^{O(n)}}$ états,
  Klarlund (1991), Safra (1992) : $2^{O(n \log n)}$, la borne théorique.
- ont leur vide décidable

Un automate de Büchi n'est pas toujours déterminisable.

Un automate de Büchi n'est pas tjrs déterminisable

$A =$ 

$\mathscr{L}(A) = (\clubsuit + \heartsuit)^\star \heartsuit^\omega$

Supp. $\exists$ automate dét. $B = \langle \{\clubsuit, \heartsuit\}, \mathcal{Q}, \delta, \{q_0\}, F \rangle$ avec un seul ensemble d'acceptation, tel que $\mathscr{L}(B) = \mathscr{L}(A)$.
$u_0 = \heartsuit^\omega \in \mathscr{L}(A)$, donc $\exists v_0$, préfixe fini de u_0 qui amène B dans F.
$u_1 = v_0 \clubsuit \heartsuit^\omega \in \mathscr{L}(A)$, donc il $\exists$ un préfixe fini $v_0 \clubsuit v_1$ de u_1 qui amène B dans F.
$\vdots$
$u_n = v_{n-1} \clubsuit \heartsuit^\omega \in \mathscr{L}(A)$, donc $\exists$ un préfixe fini $v_0 \clubsuit v_1 \clubsuit \cdots \clubsuit v_n$ de u_n qui amène B dans F.
Puisque $\mathcal{Q}$ est fini, il existe i et j, $0 \leqslant i < j$, tels que les mots $v_0 \clubsuit v_1 \clubsuit \cdots \clubsuit v_i$ et $v_0 \clubsuit v_1 \clubsuit \cdots \clubsuit v_i \clubsuit \cdots \clubsuit v_j$ mènent au même état.
Donc $m = v_0 \clubsuit v_1 \clubsuit \cdots \clubsuit v_i (\clubsuit \cdots \clubsuit v_j)^\omega$ est accepté par B.
Or m contient une infinité de $\clubsuit$, il ne peut pas appartenir à $\mathscr{L}(A)$!

Produit synchronisé

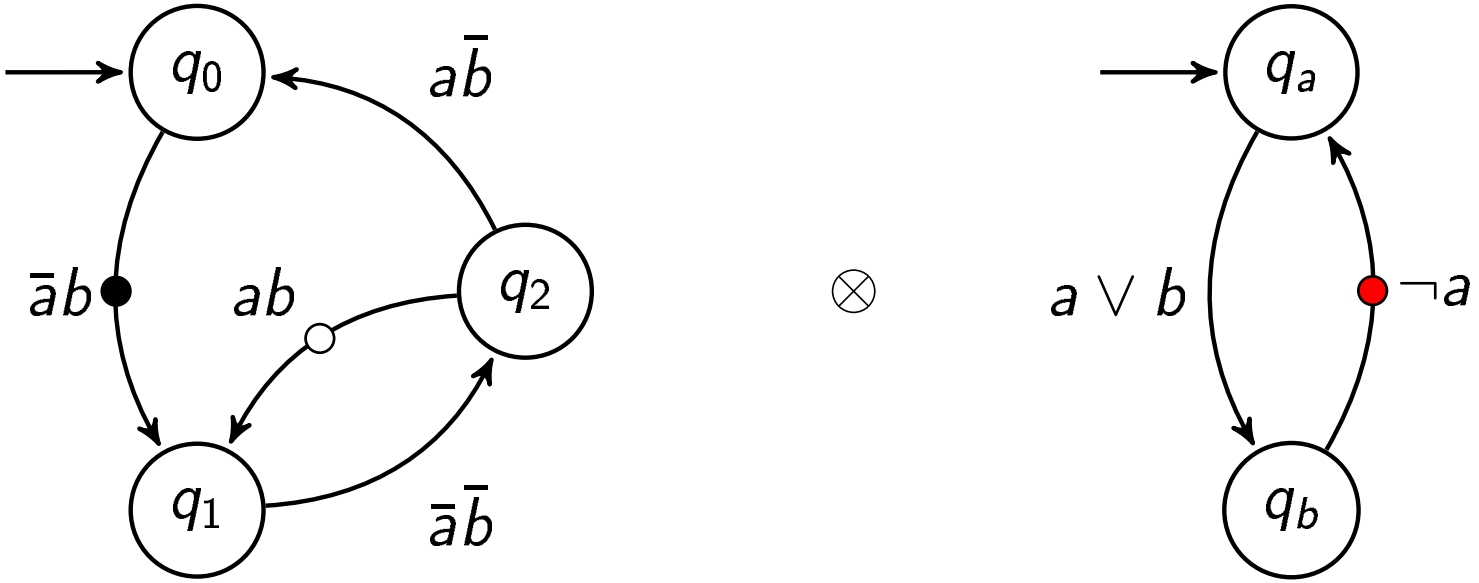
Soient $A_1 = \langle \Sigma, \mathcal{Q}_1, \mathcal{Q}_1^0, \mathcal{F}_1, \delta_1 \rangle$ et $A_2 = \langle \Sigma, \mathcal{Q}_2, \mathcal{Q}_2^0, \mathcal{F}_2, \delta_2 \rangle$ deux TGBA partageant le même ensemble de propositions atomiques.

Le produit synchronisé de A_1 et A_2 est l'automate noté $A_1 \otimes A_2 = \langle \Sigma, \mathcal{Q}, Q^0, \mathcal{F}, \delta_1 \rangle$ où

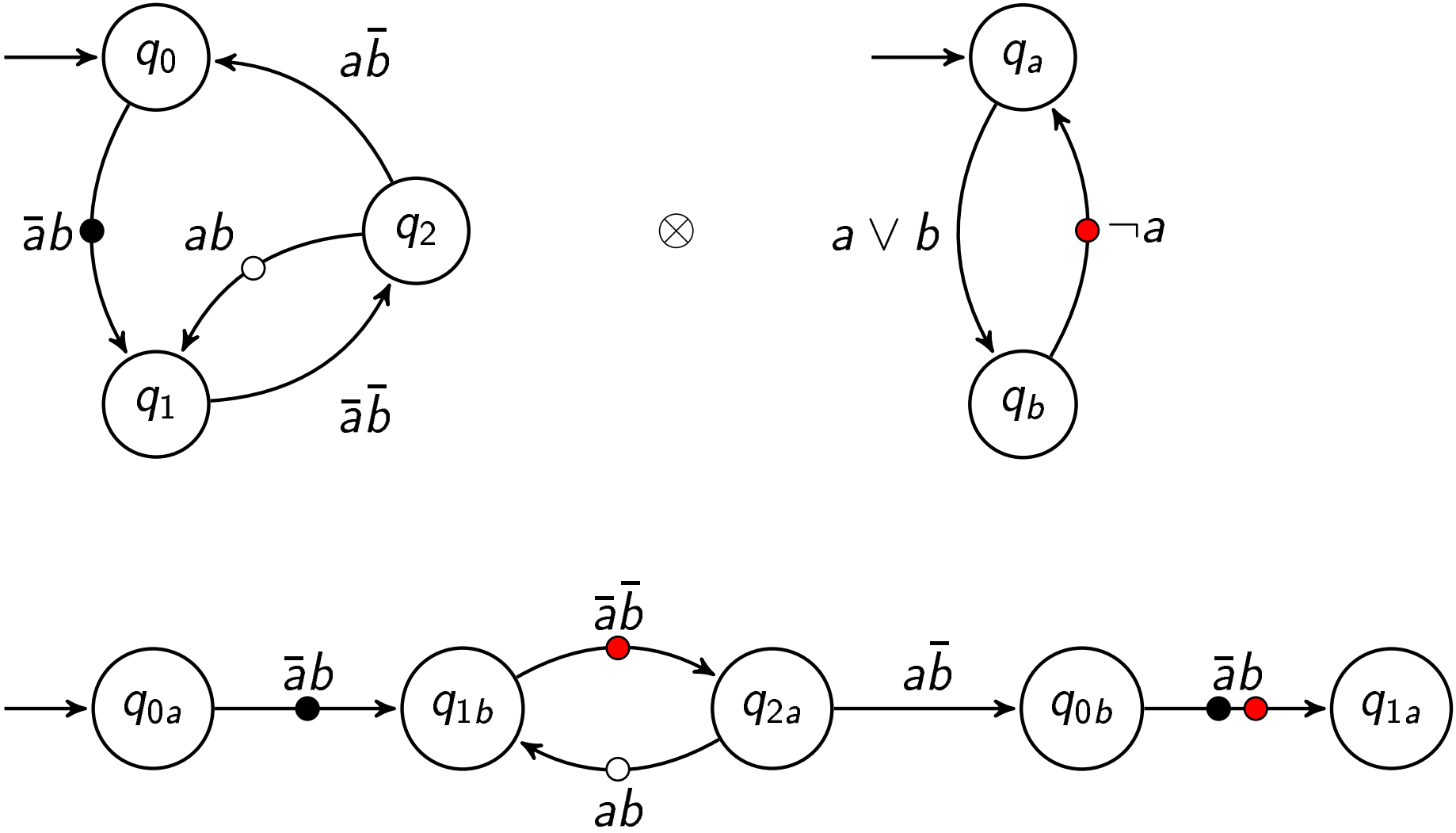
- $\mathcal{Q} = \mathcal{Q}_1 \times \mathcal{Q}_2$
- $\mathcal{Q}^0 = \mathcal{Q}_1^0 \times \mathcal{Q}_2^0$
- $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$ à condition que $\mathcal{F}_1$ et $\mathcal{F}_2$ soient disjoints
- $\delta = \{((t_1^{\text{in}}, t_2^{\text{in}}), t_1^{\text{prop}} \cap t_2^{\text{prop}}, t_1^{\text{acc}} \cup t_2^{\text{acc}}, (t_1^{\text{out}}, t_2^{\text{out}})) \mid t_1 \in \delta_1,\ t_2 \in \delta_2,\ t_1^{\text{prop}} \cap t_2^{\text{prop}} \neq \emptyset\}$

On a alors $\mathscr{L}(A_1 \otimes A_2) = \mathscr{L}(A_1) \cap \mathscr{L}(A_2)$.

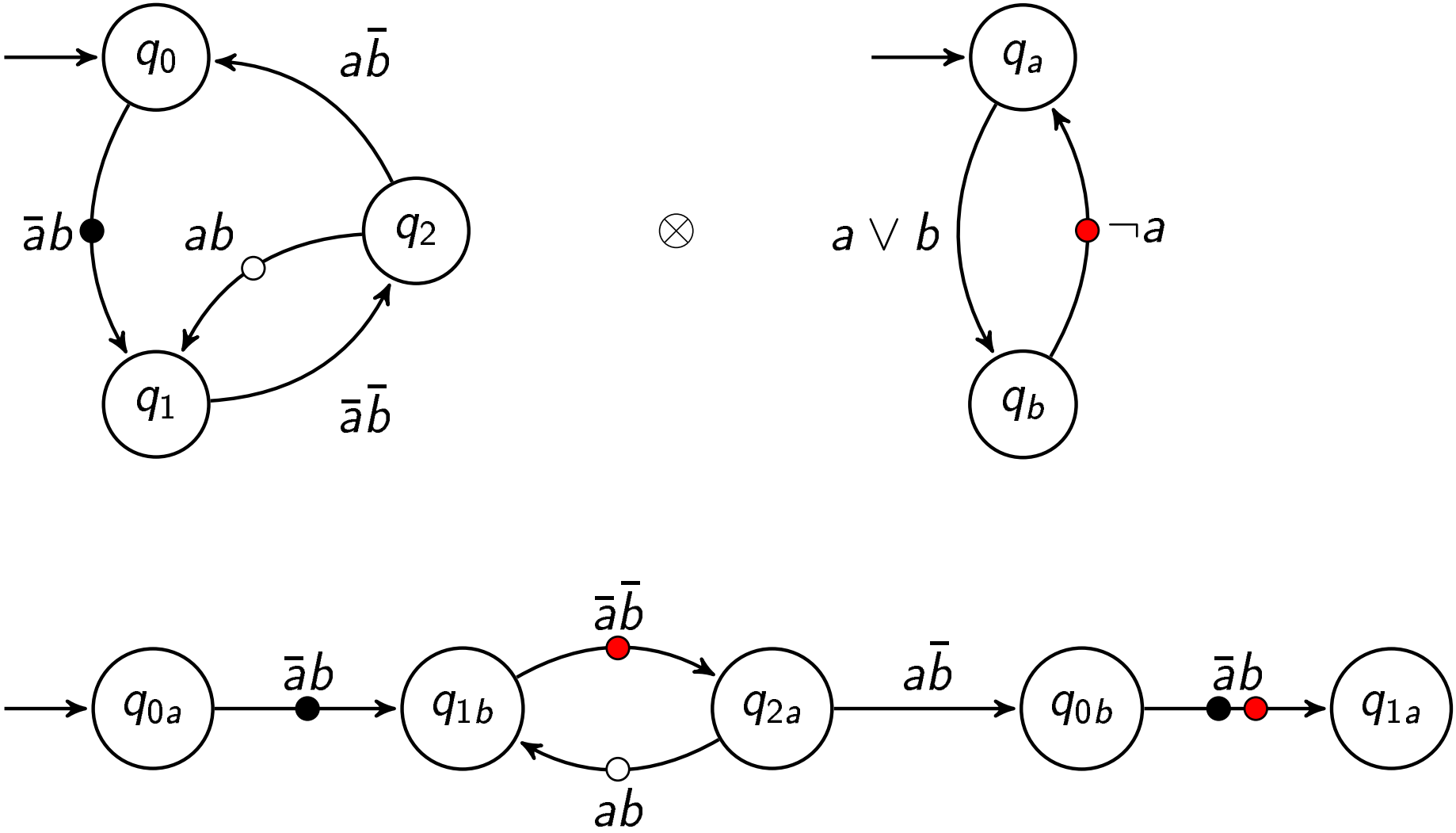
Exemple de produit synchronisé



Exemple de produit synchronisé

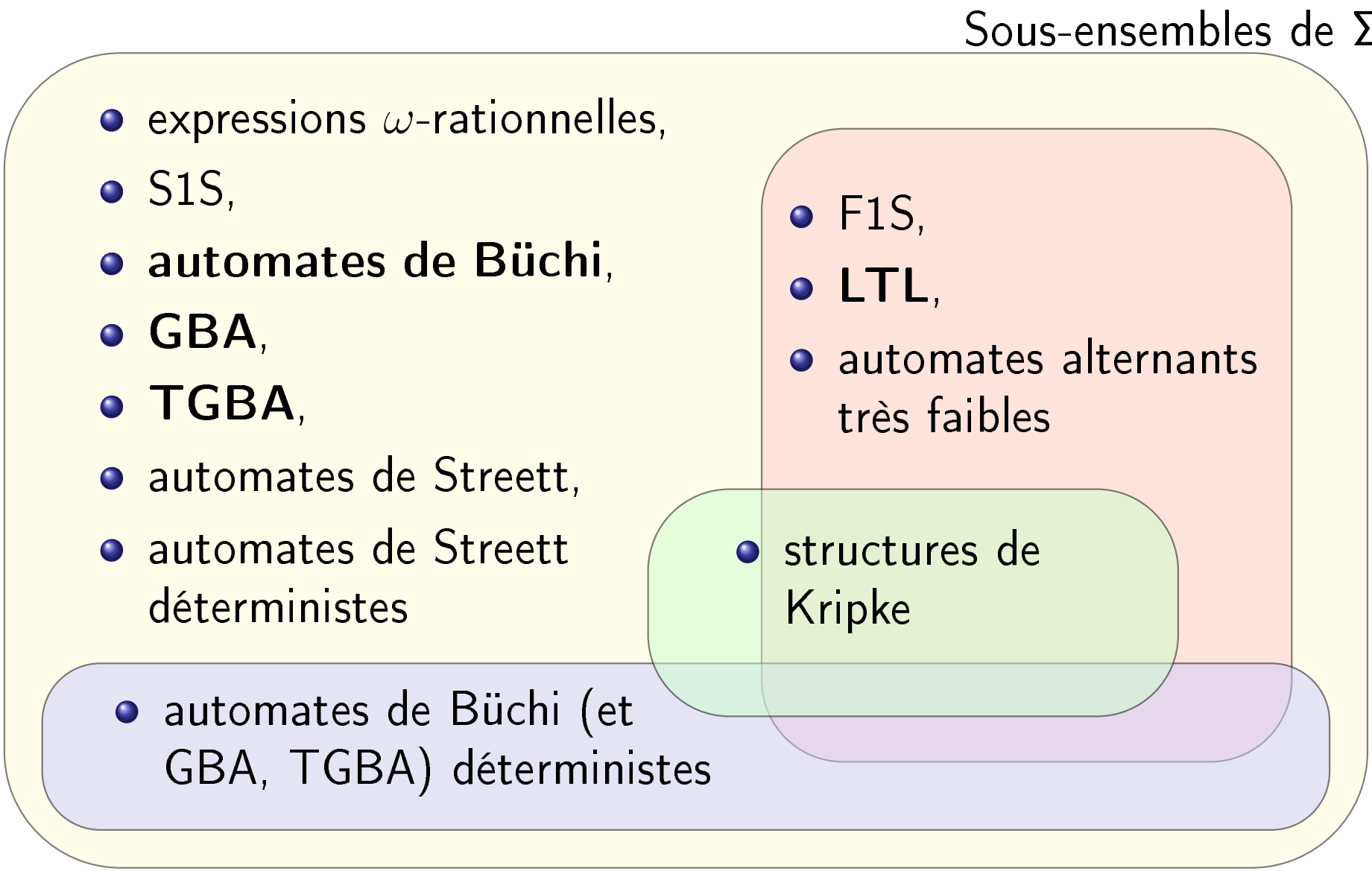


Exemple de produit synchronisé

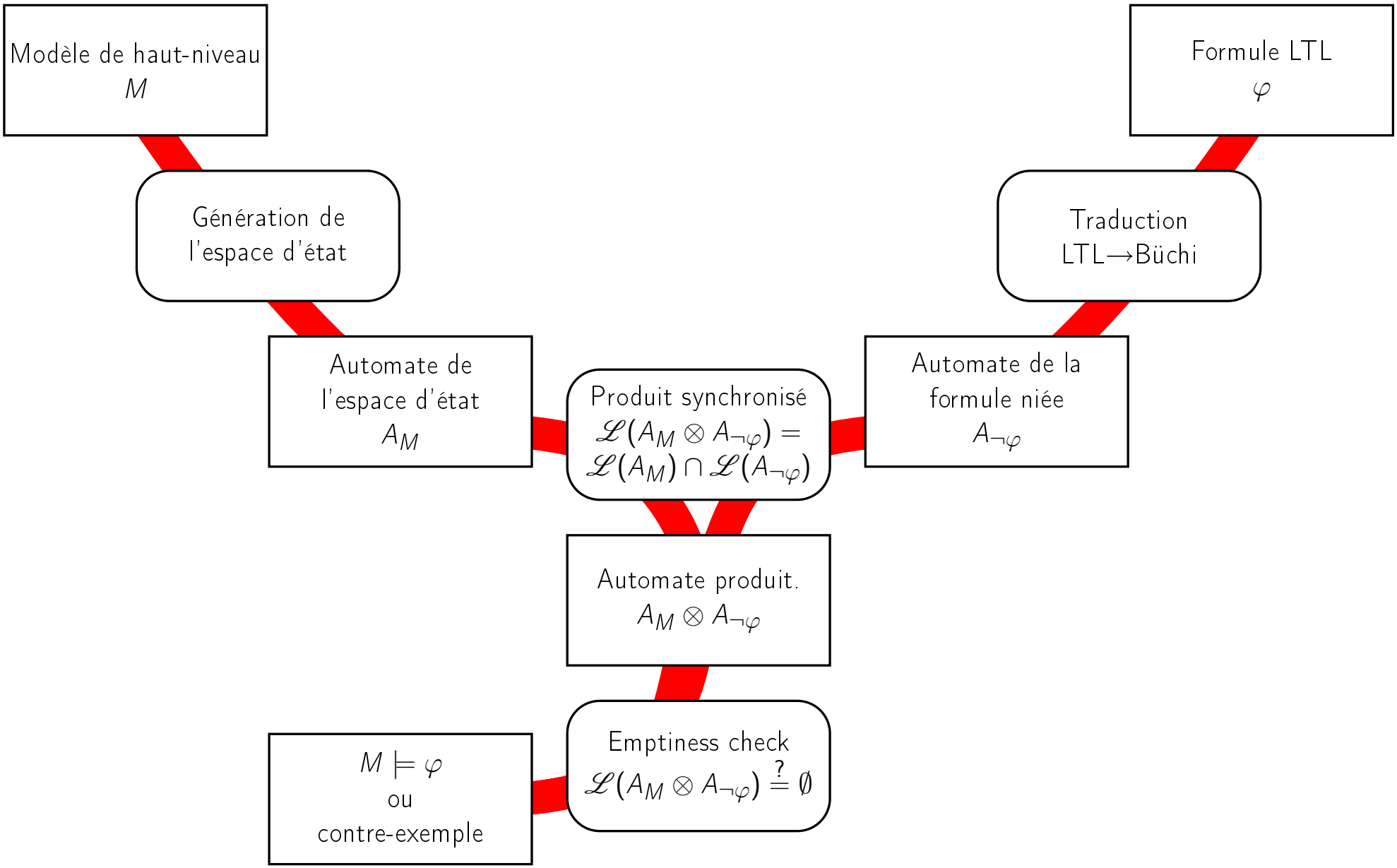


- Que devient le produit si l'on retire « ● » du premier automate ?

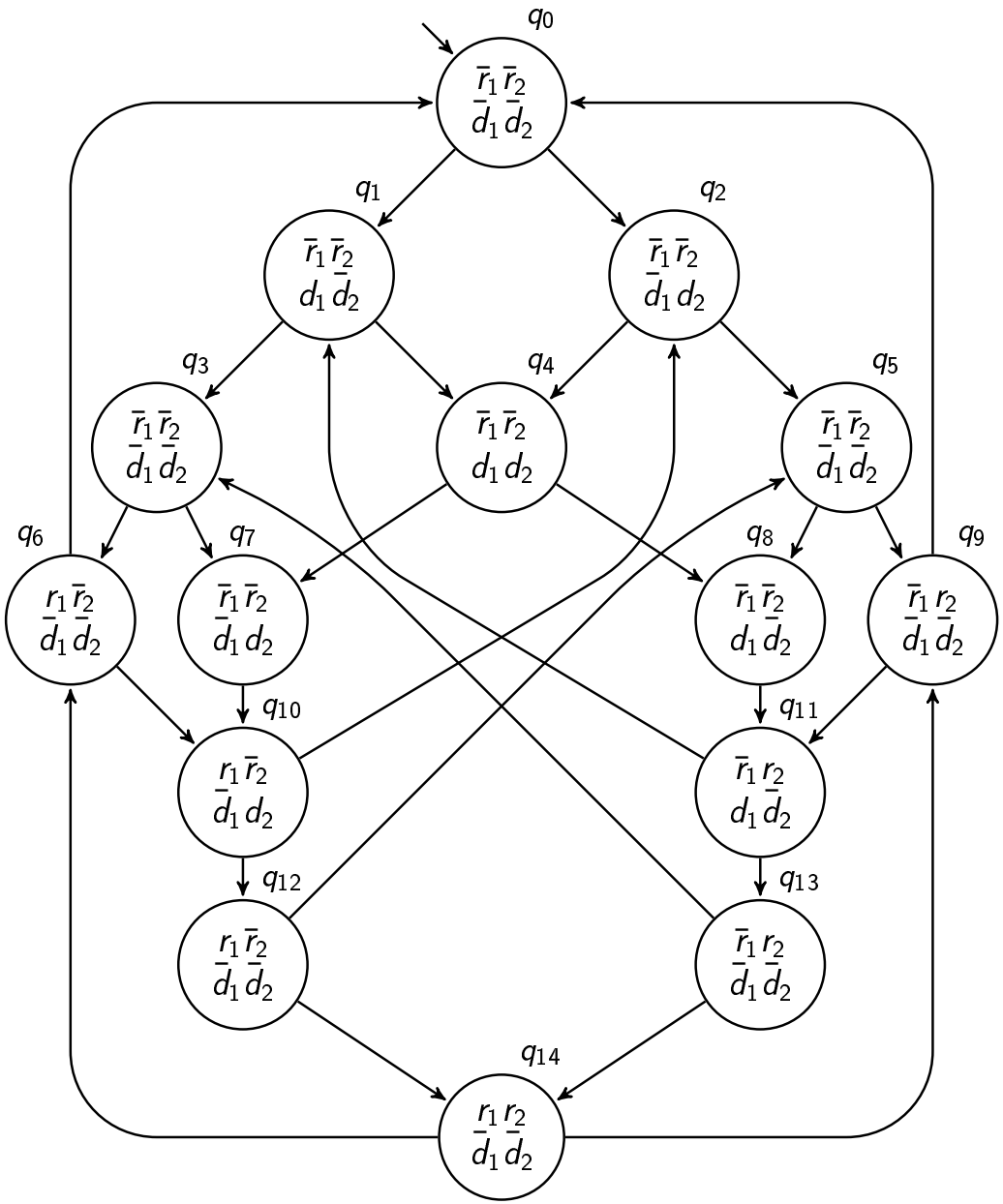
Pouvoir d'expression dans Σ^ω



Approche automate du model checking



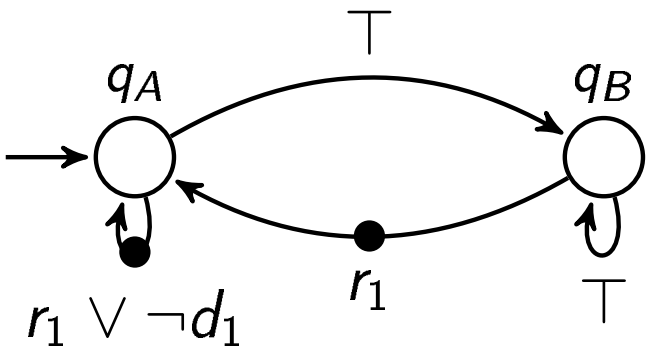
Structure de Kripke pour l'exemple



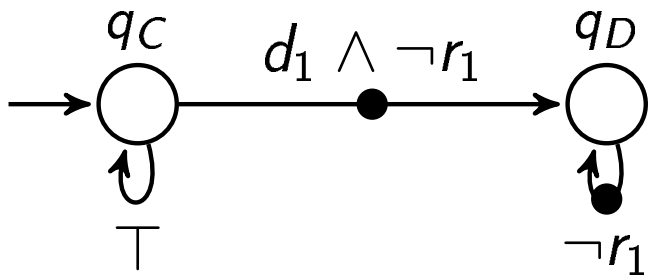
Formule à vérifier

Pour tout $i \in \{1, 2\}$, si un état vérifie d_i alors dans tous ses futurs possibles il possède un successeur qui vérifie r_i.
Par symétrie on peut se limiter à $i = 1$.

En LTL : $\mathbf{G}(d_1 \rightarrow \mathbf{F}\, r_1)$.

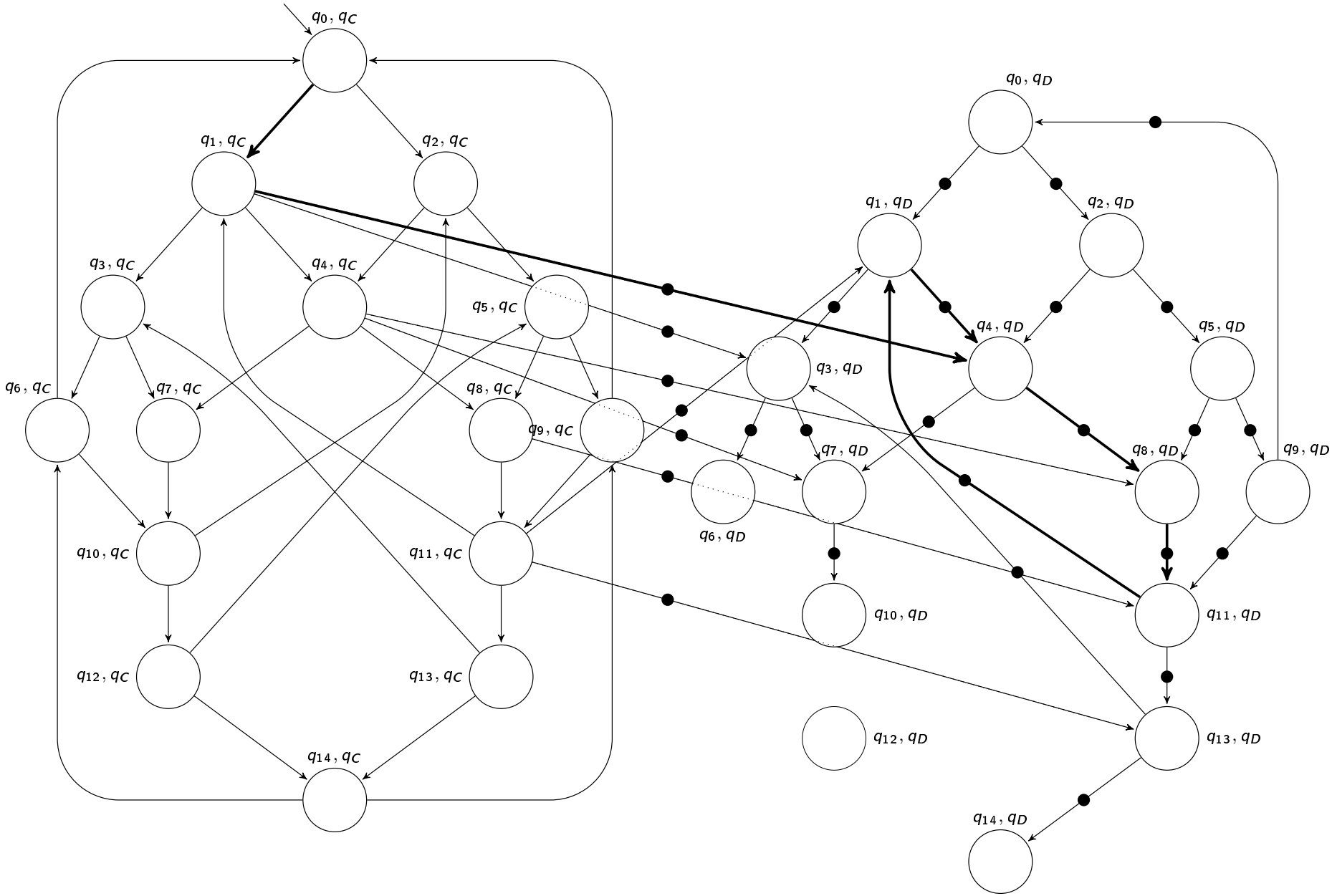


$A_{\mathbf{G}(d_1 \rightarrow \mathbf{F}\, r_1)}$



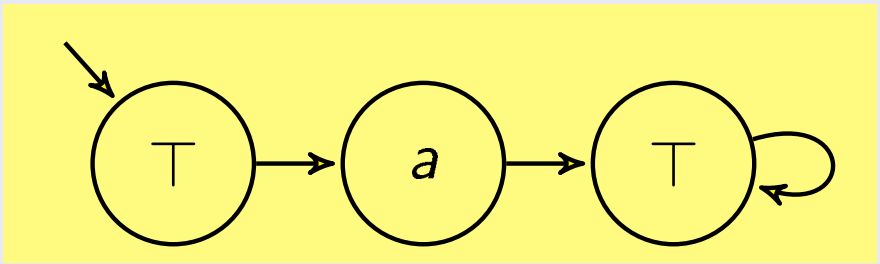
$A_{\neg\, \mathbf{G}(d_1 \rightarrow \mathbf{F}\, r_1)}$

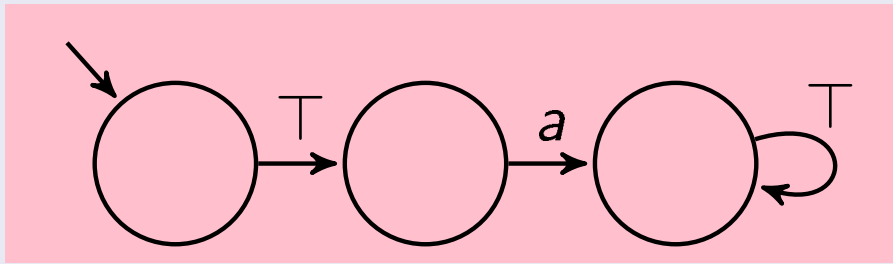
Produit structure de Kripke/Automate



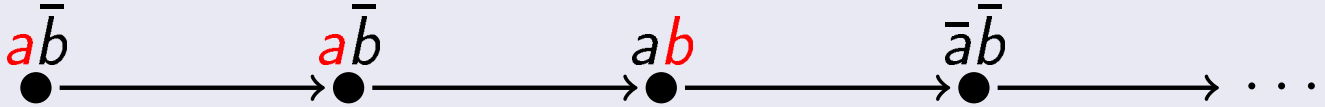
LTL et automates sur états ou transitions

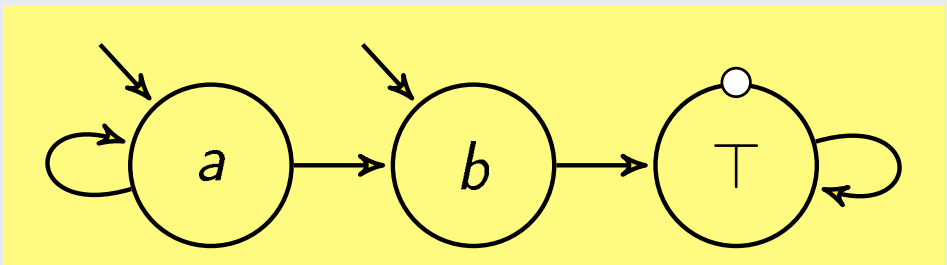
$\mathbf{X}\, a$

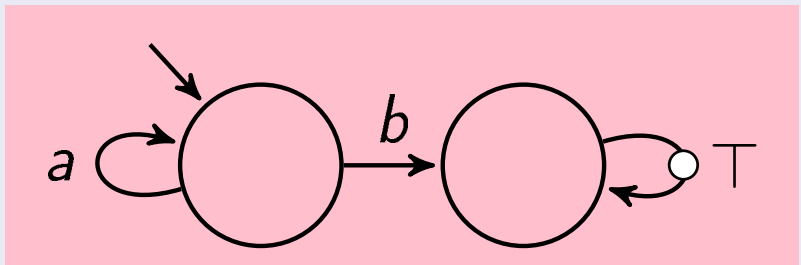




$a\,\mathbf{U}\,b$





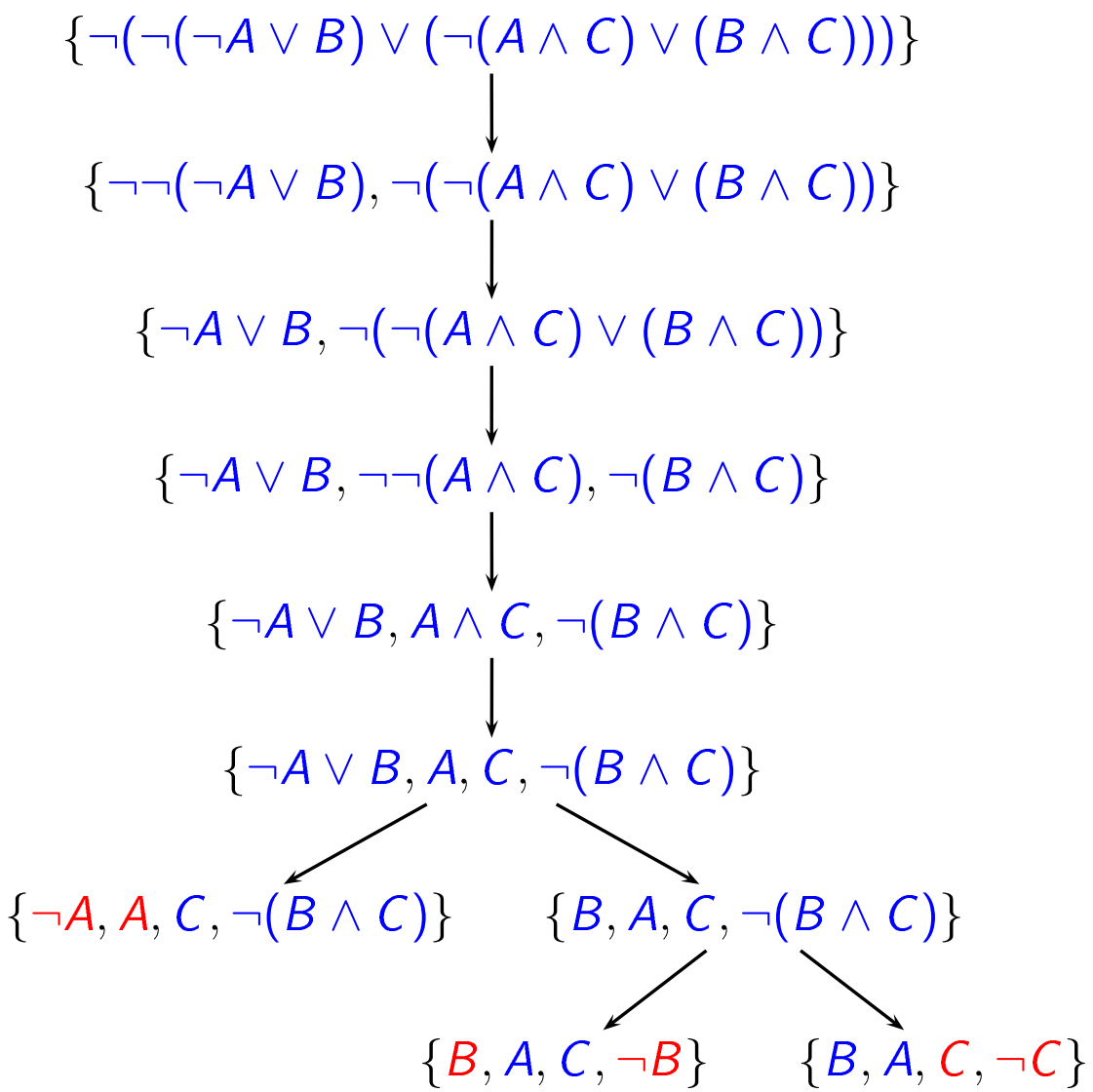


$$a\,\mathbf{U}\,b \equiv b \vee (a \wedge \mathbf{X}(a\,\mathbf{U}\,b))$$

Règles de tableau

formule	1er fils	2e fils
$\Gamma \cup \{\neg\top\}$	$\Gamma \cup \{\bot\}$	
$\Gamma \cup \{\neg\bot\}$	$\Gamma \cup \{\top\}$	
$\Gamma \cup \{\neg\neg f\}$	$\Gamma \cup \{f\}$	
$\Gamma \cup \{f \wedge g\}$	$\Gamma \cup \{f, g\}$	
$\Gamma \cup \{f \vee g\}$	$\Gamma \cup \{f\}$	$\Gamma \cup \{g\}$
$\Gamma \cup \{\neg(f \wedge g)\}$	$\Gamma \cup \{\neg f\}$	$\Gamma \cup \{\neg g\}$
$\Gamma \cup \{\neg(f \vee g)\}$	$\Gamma \cup \{\neg f, \neg g\}$	

Tableau de $\neg\varphi$ avec $\varphi = \neg(\neg A \vee B) \vee (\neg(A \wedge C) \vee (B \wedge C))$



Règles de tableau

formule	1^{er} fils	2^{e} fils
$\Gamma \cup \{\neg\top\}$	$\Gamma \cup \{\bot\}$	
$\Gamma \cup \{\neg\bot\}$	$\Gamma \cup \{\top\}$	
$\Gamma \cup \{\neg\neg f\}$	$\Gamma \cup \{f\}$	
$\Gamma \cup \{f \wedge g\}$	$\Gamma \cup \{f, g\}$	
$\Gamma \cup \{f \vee g\}$	$\Gamma \cup \{f\}$	$\Gamma \cup \{g\}$
$\Gamma \cup \{\neg(f \wedge g)\}$	$\Gamma \cup \{\neg f\}$	$\Gamma \cup \{\neg g\}$
$\Gamma \cup \{\neg(f \vee g)\}$	$\Gamma \cup \{\neg f, \neg g\}$	

Règles de tableau

formule	1er fils	2e fils
$\Gamma \cup \{\neg\top\}$	$\Gamma \cup \{\bot\}$	
$\Gamma \cup \{\neg\bot\}$	$\Gamma \cup \{\top\}$	
$\Gamma \cup \{\neg\neg f\}$	$\Gamma \cup \{f\}$	
$\Gamma \cup \{f \wedge g\}$	$\Gamma \cup \{f, g\}$	
$\Gamma \cup \{f \vee g\}$	$\Gamma \cup \{f\}$	$\Gamma \cup \{g\}$
$\Gamma \cup \{\neg(f \wedge g)\}$	$\Gamma \cup \{\neg f\}$	$\Gamma \cup \{\neg g\}$
$\Gamma \cup \{\neg(f \vee g)\}$	$\Gamma \cup \{\neg f, \neg g\}$	
$\Gamma \cup \{\neg \mathbf{X} f\}$	$\Gamma \cup \{\mathbf{X} \neg f\}$	
$\Gamma \cup \{f \,\mathbf{U}\, g\}$	$\Gamma \cup \{g\}$	$\Gamma \cup \{f, \mathbf{X}(f \,\mathbf{U}\, g), \mathrm{P}\, g\}$
$\Gamma \cup \{\neg(f \,\mathbf{U}\, g)\}$	$\Gamma \cup \{\neg f, \neg g\}$	$\Gamma \cup \{\neg g, \mathbf{X} \neg(f \,\mathbf{U}\, g)\}$

$\mathrm{P}\, g$ est une promesse que g sera vérifié

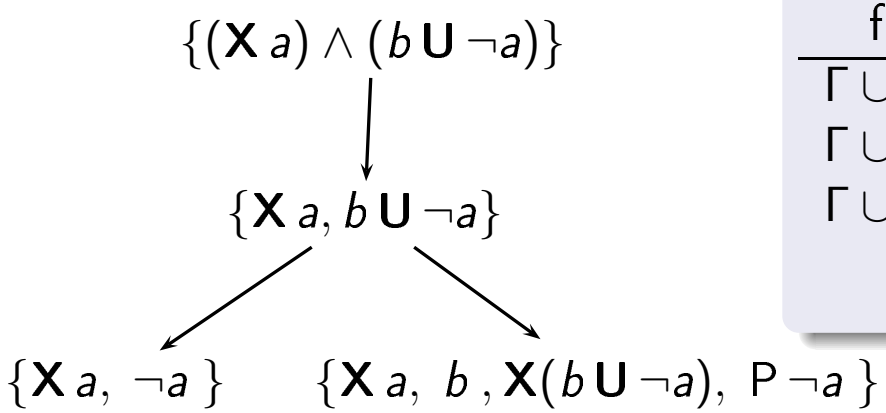
Tableau pour $(\mathbf{X}\, a) \wedge (b\, \mathbf{U}\, \neg a)$

$\{(\mathbf{X}\, a) \wedge (b\, \mathbf{U}\, \neg a)\}$

Règles de tableau

formule	1^{er} fils	2^{e} fils
$\Gamma \cup \{f \wedge g\}$	$\Gamma \cup \{f, g\}$	
$\Gamma \cup \{f \vee g\}$	$\Gamma \cup \{f\}$	$\Gamma \cup \{g\}$
$\Gamma \cup \{f\, \mathbf{U}\, g\}$	$\Gamma \cup \{g\}$	$\Gamma \cup \{f, \mathbf{X}(f\, \mathbf{U}\, g), \mathrm{P}\, g\}$
$\vdots$	$\vdots$	$\vdots$

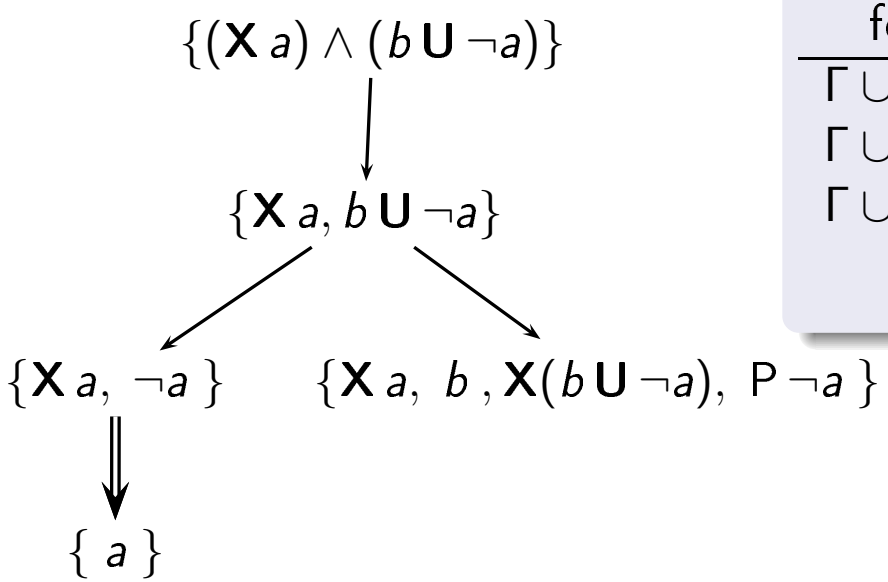
Tableau pour $(\mathbf{X}\, a) \wedge (b\, \mathbf{U}\, \neg a)$



Règles de tableau

formule	1^{er} fils	2^{e} fils
$\Gamma \cup \{f \wedge g\}$	$\Gamma \cup \{f, g\}$	
$\Gamma \cup \{f \vee g\}$	$\Gamma \cup \{f\}$	$\Gamma \cup \{g\}$
$\Gamma \cup \{f\, \mathbf{U}\, g\}$	$\Gamma \cup \{g\}$	$\Gamma \cup \{f, \mathbf{X}(f\, \mathbf{U}\, g), \mathrm{P}\, g\}$
$\vdots$	$\vdots$	$\vdots$

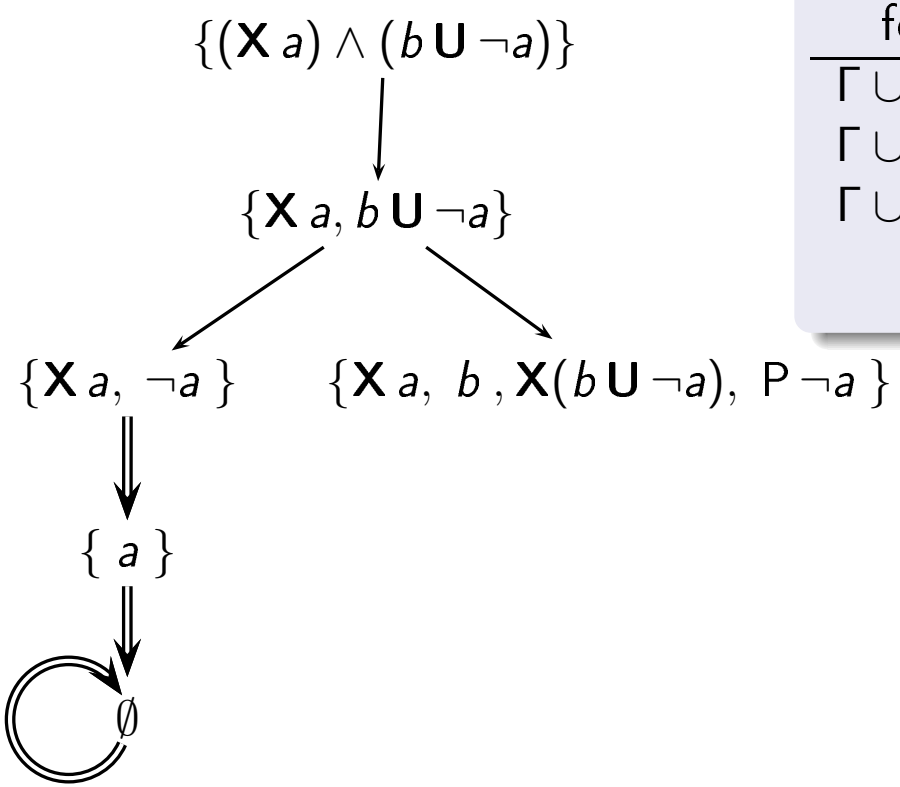
Tableau pour $(\mathbf{X}\, a) \wedge (b\, \mathbf{U}\, \neg a)$



Règles de tableau

formule	1^{er} fils	2^{e} fils
$\Gamma \cup \{f \wedge g\}$	$\Gamma \cup \{f, g\}$	
$\Gamma \cup \{f \vee g\}$	$\Gamma \cup \{f\}$	$\Gamma \cup \{g\}$
$\Gamma \cup \{f\, \mathbf{U}\, g\}$	$\Gamma \cup \{g\}$	$\Gamma \cup \{f, \mathbf{X}(f\, \mathbf{U}\, g), \mathsf{P}\, g\}$
$\vdots$	$\vdots$	$\vdots$

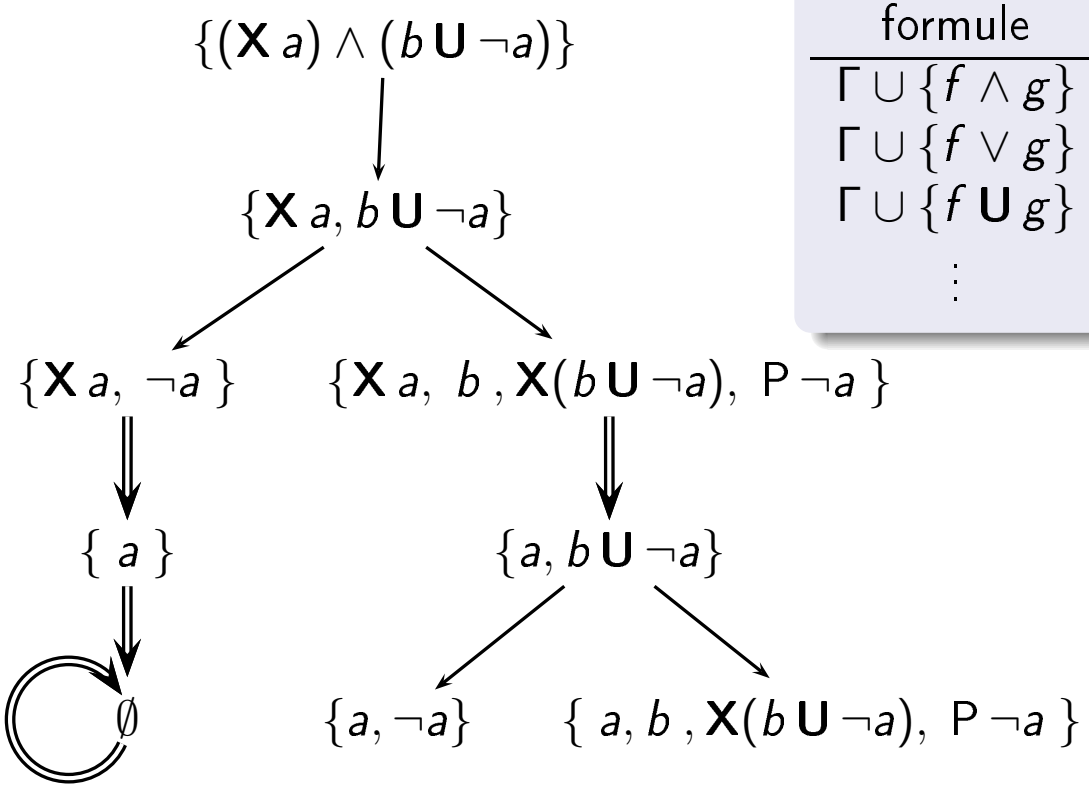
Tableau pour $(\mathbf{X}\, a) \wedge (b\, \mathbf{U}\, \neg a)$



Règles de tableau

formule	1^{er} fils	2^{e} fils
$\Gamma \cup \{f \wedge g\}$	$\Gamma \cup \{f, g\}$	
$\Gamma \cup \{f \vee g\}$	$\Gamma \cup \{f\}$	$\Gamma \cup \{g\}$
$\Gamma \cup \{f\, \mathbf{U}\, g\}$	$\Gamma \cup \{g\}$	$\Gamma \cup \{f, \mathbf{X}(f\, \mathbf{U}\, g), \mathsf{P}\, g\}$
$\vdots$	$\vdots$	$\vdots$

Tableau pour $(\mathbf{X}\, a) \wedge (b\, \mathbf{U}\, \neg a)$



Règles de tableau

formule	1^{er} fils	2^{e} fils
$\Gamma \cup \{f \wedge g\}$	$\Gamma \cup \{f, g\}$	
$\Gamma \cup \{f \vee g\}$	$\Gamma \cup \{f\}$	$\Gamma \cup \{g\}$
$\Gamma \cup \{f\, \mathbf{U}\, g\}$	$\Gamma \cup \{g\}$	$\Gamma \cup \{f, \mathbf{X}(f\, \mathbf{U}\, g), \mathsf{P}\, g\}$
$\vdots$	$\vdots$	$\vdots$

Tableau pour $(\mathbf{X}\, a) \wedge (b\, \mathbf{U}\, \neg a)$

Règles de tableau

formule	1^{er} fils	2^{e} fils
$\Gamma \cup \{f \wedge g\}$	$\Gamma \cup \{f, g\}$	
$\Gamma \cup \{f \vee g\}$	$\Gamma \cup \{f\}$	$\Gamma \cup \{g\}$
$\Gamma \cup \{f\, \mathbf{U}\, g\}$	$\Gamma \cup \{g\}$	$\Gamma \cup \{f, \mathbf{X}(f\, \mathbf{U}\, g), \mathrm{P}\, g\}$
$\vdots$	$\vdots$	$\vdots$

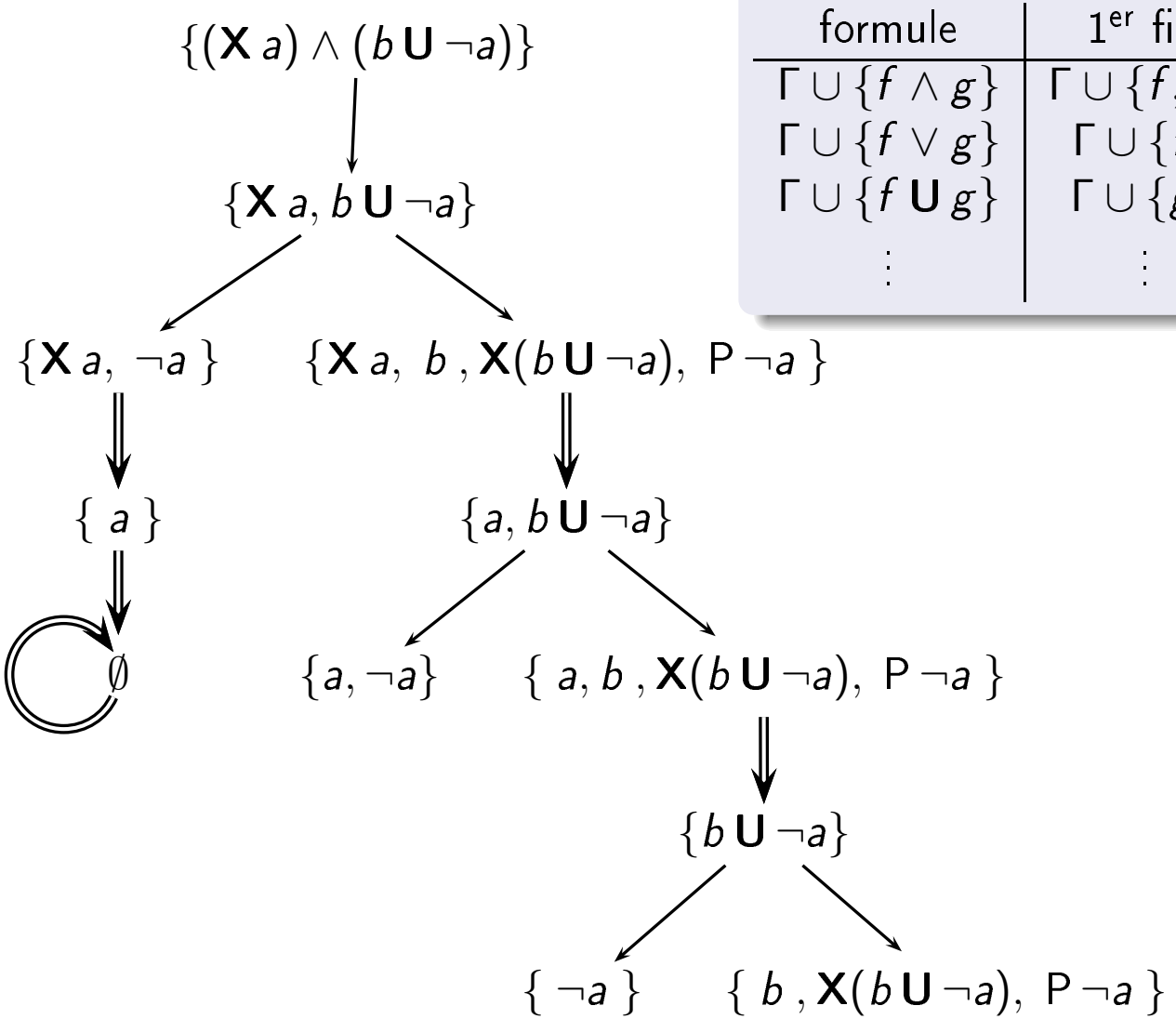
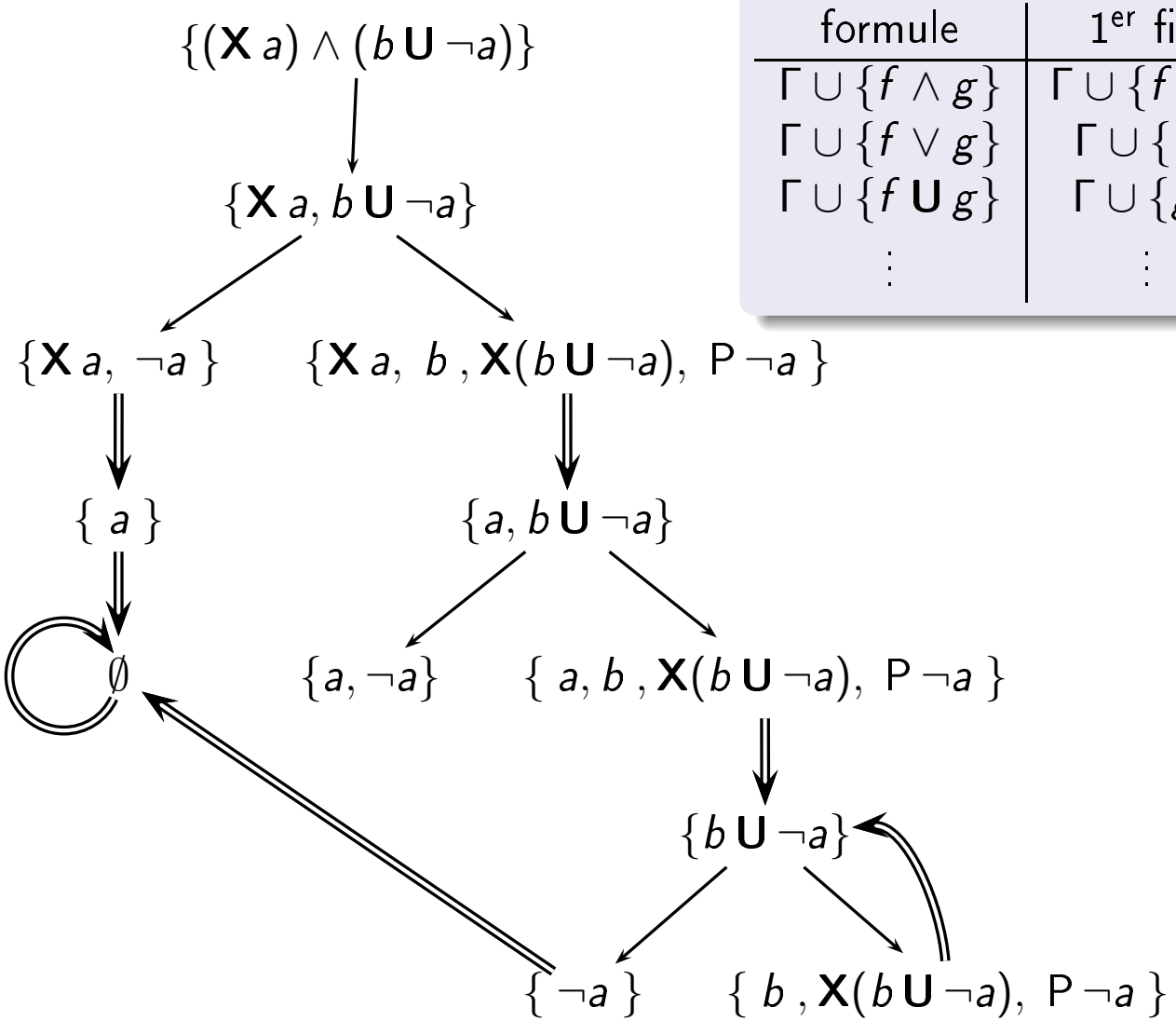


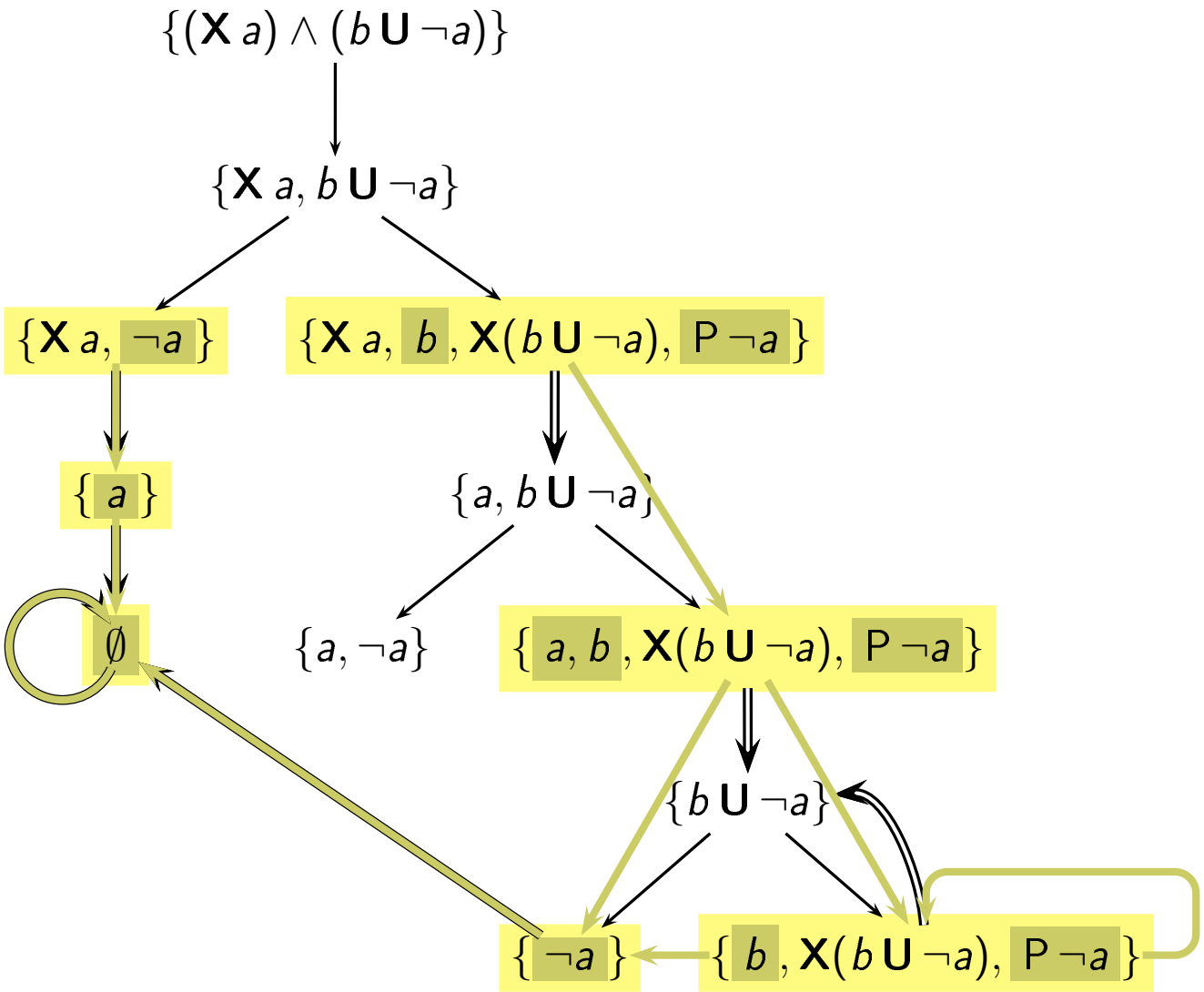
Tableau pour $(\mathbf{X}\, a) \wedge (b\, \mathbf{U}\, \neg a)$

Règles de tableau

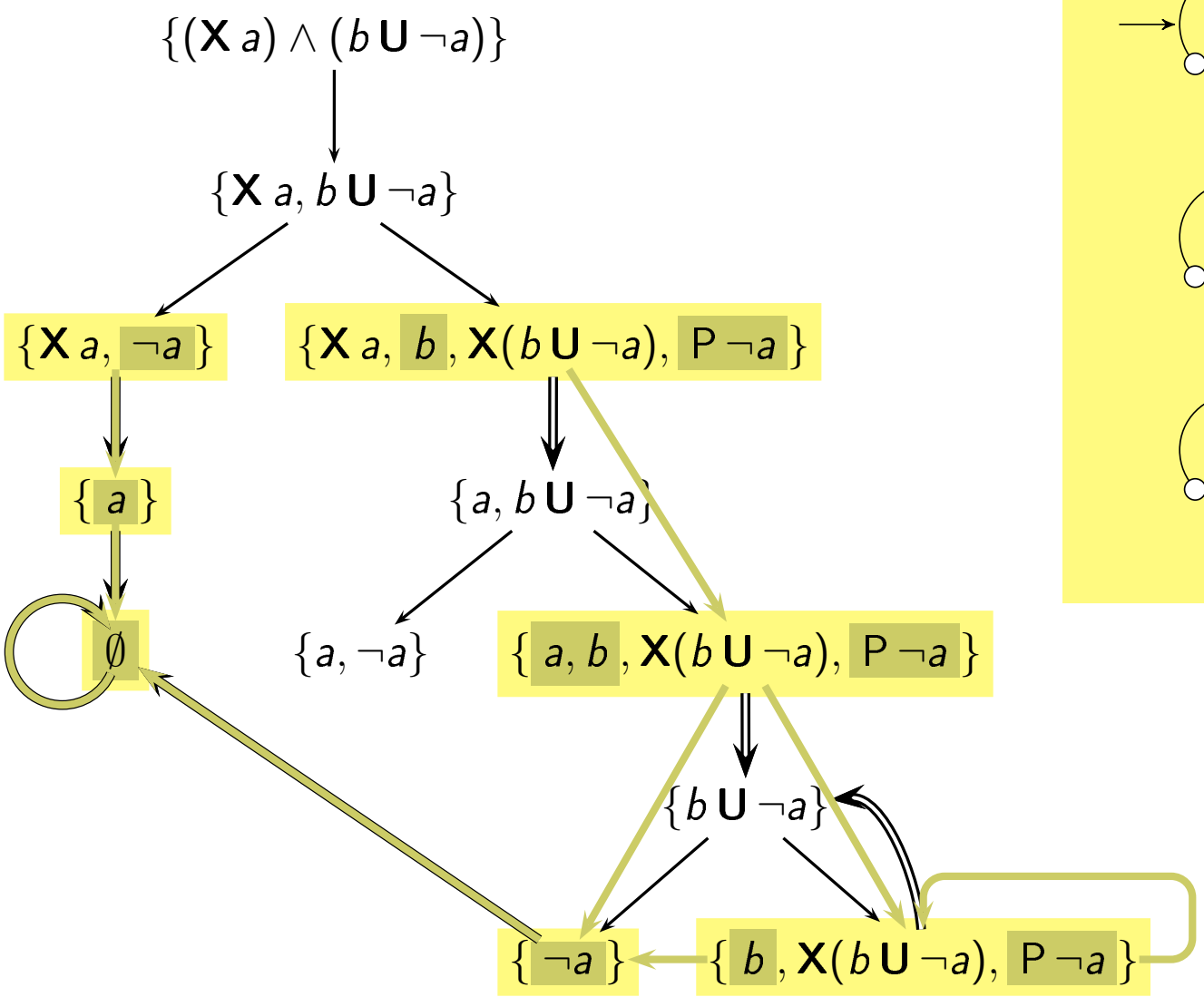
formule	1^{er} fils	2^{e} fils
$\Gamma \cup \{f \wedge g\}$	$\Gamma \cup \{f, g\}$	
$\Gamma \cup \{f \vee g\}$	$\Gamma \cup \{f\}$	$\Gamma \cup \{g\}$
$\Gamma \cup \{f\, \mathbf{U}\, g\}$	$\Gamma \cup \{g\}$	$\Gamma \cup \{f, \mathbf{X}(f\, \mathbf{U}\, g), \mathrm{P}\, g\}$
$\vdots$	$\vdots$	$\vdots$

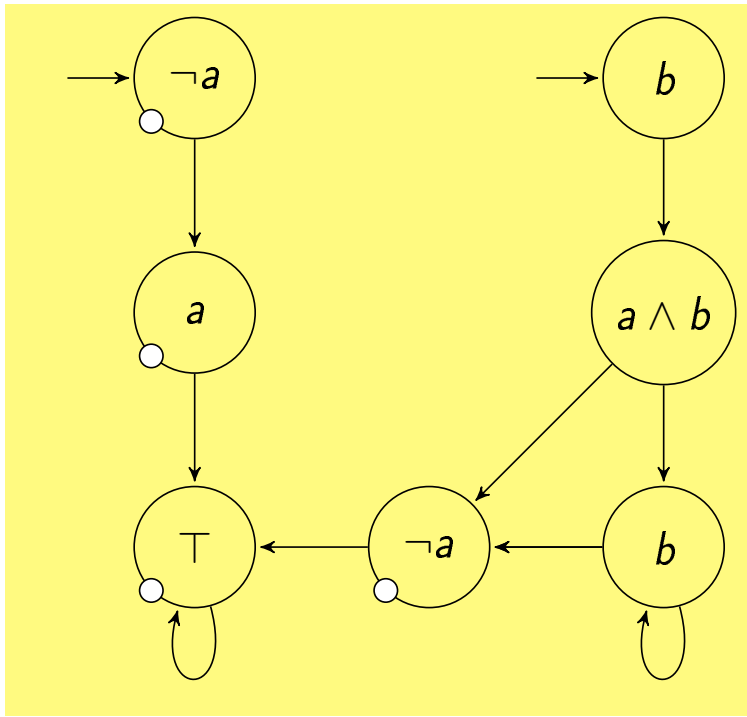


$(\mathbf{X}\, a) \wedge (b\, \mathbf{U}\, \neg a)$ vers GBA basé sur les états

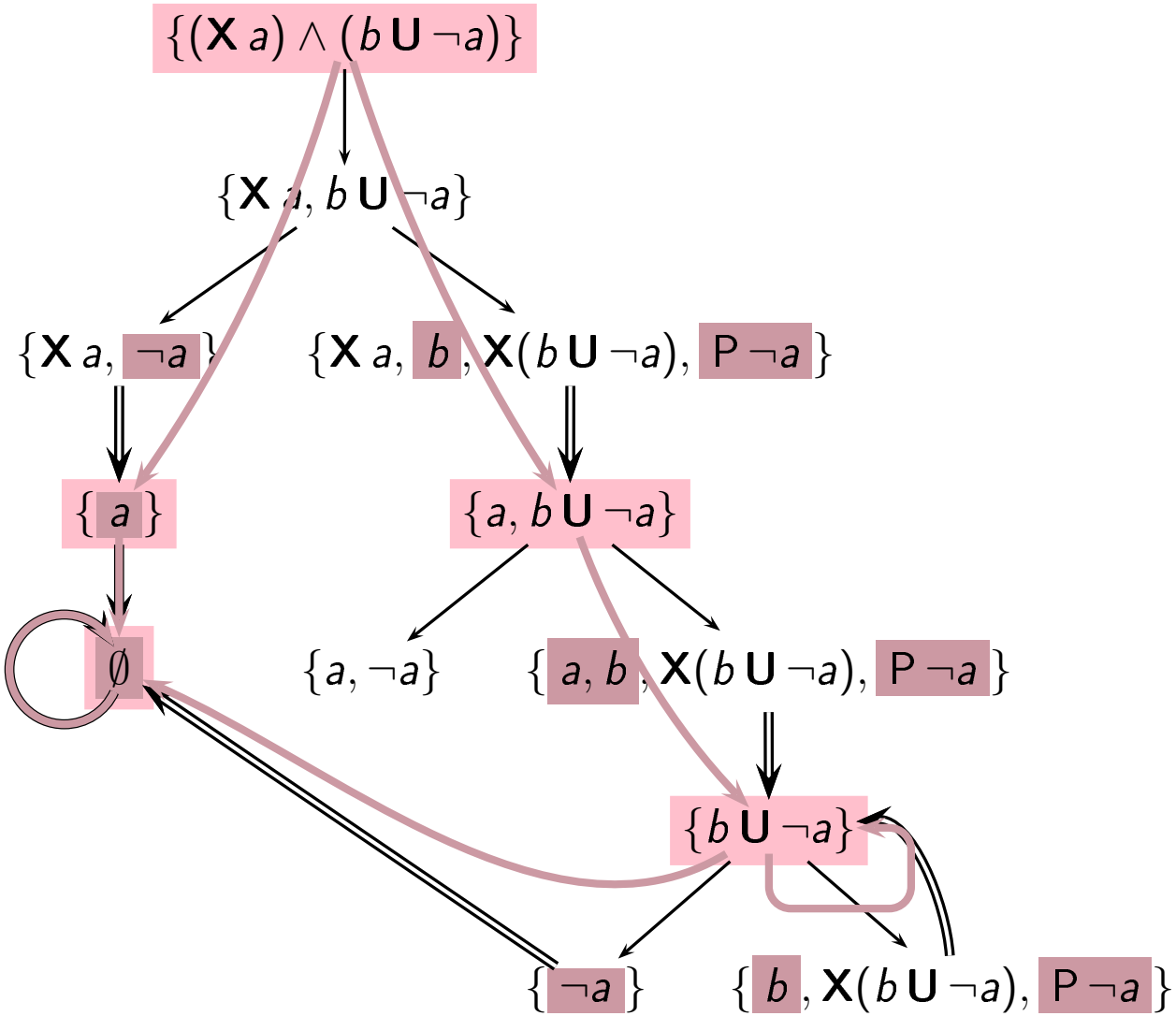


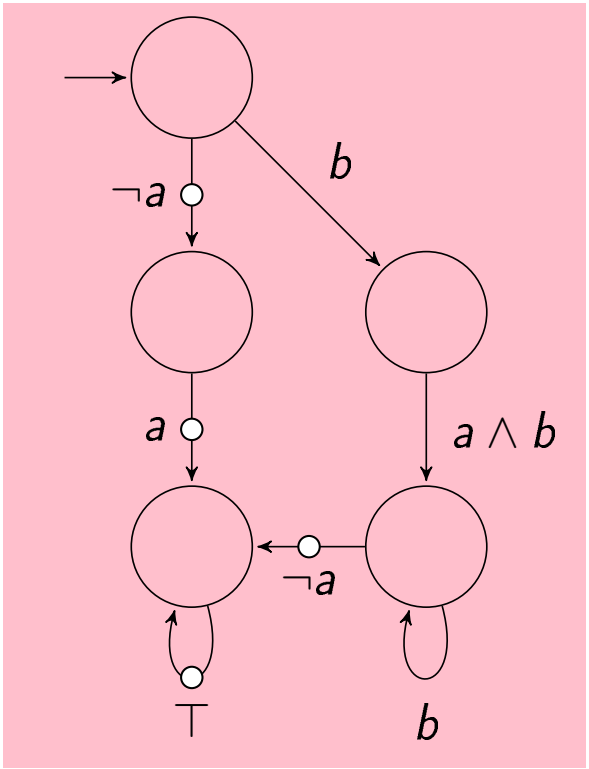
$(\mathbf{X}\, a) \wedge (b\, \mathbf{U}\, \neg a)$ vers GBA basé sur les états





$(\mathbf{X}\, a) \wedge (b\, \mathbf{U}\, \neg a)$ vers GBA basé sur les transitions





Traduisez vos formules LTL en ligne

`http://spot.lip6.fr/cgi-bin/ltl2tgba.py`