

Cours 12

Logiques temporelles & Vérification de modèle
(Model checking)

Motivations

But : modéliser (puis vérifier) le comportement dynamique d'un système.

Plusieurs langages :

- ▶ LTL : *Linear time temporal logic*
- ▶ CTL : *Computation tree logic*
- ▶ CTL^{*} : combine LTL et CTL

Nous nous restreignons dans ce cours aux formules CTL.

CTL : syntaxe

V un ensemble dénombrable de variables propositionnelles.

C ensemble fini de connecteurs propositionnels : $C = \{\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow\}$

T ensemble fini de connecteurs temporelles : $T = \{X, F, G, U, A, E\}$

Définition (Formules CTL)

L'ensemble des formules CTL est un langage sur $V \cup C \cup T \cup \{(,)\}$ défini inductivement par :

- ▶ $V \subseteq \text{CTL}$
- ▶ si $\varphi_1, \varphi_2 \in \text{CTL}$, alors
 $\neg\varphi_1, (\varphi_1 \wedge \varphi_2), (\varphi_1 \vee \varphi_2), (\varphi_1 \Rightarrow \varphi_2), (\varphi_1 \Leftrightarrow \varphi_2) \in \text{CTL}$
- ▶ si $\varphi_1, \varphi_2 \in \text{CTL}$, alors
 $AX\varphi, EX\varphi, AF\varphi, EF\varphi, AG\varphi, EG\varphi, A(\varphi_1 U \varphi_2), E(\varphi_1 U \varphi_2) \in \text{CTL}$

Sémantique : structures de Kripke

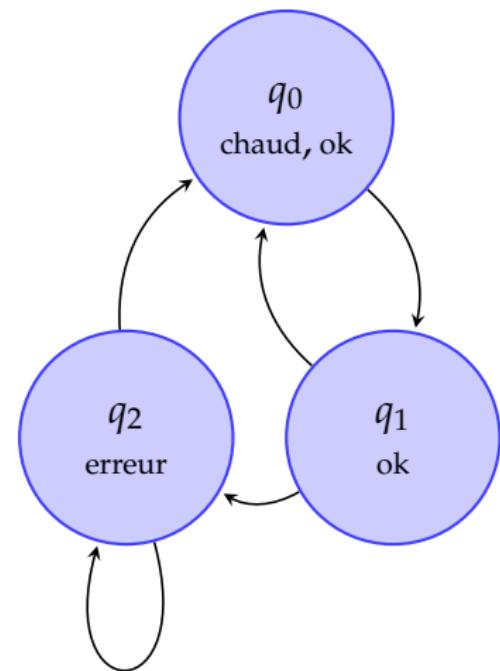
Les formules sont interprétées vis-à-vis d'un modèle particulier : les structures de Kripke.

Définition

Une *structure de Kripke* est un quadruplet
 $\langle S, R, q_0, L \rangle$ tel que

- ▶ S est un ensemble d'états,
- ▶ $R \subseteq S \times S$ est une relation de transition entre états,
- ▶ $q_0 \in S$ est un état initial,
- ▶ $L \in S \rightarrow \mathcal{P}(\mathbb{V})$ est une fonction de labélisation.

Remarque : nous nous restreignons au cas où tout état a au moins un successeur.



Sémantique : structures de Kripke

Définition

Un *chemin* dans une structure de Kripke est une séquence **maximale**

$$\pi = s_0, s_1, s_2, \dots$$

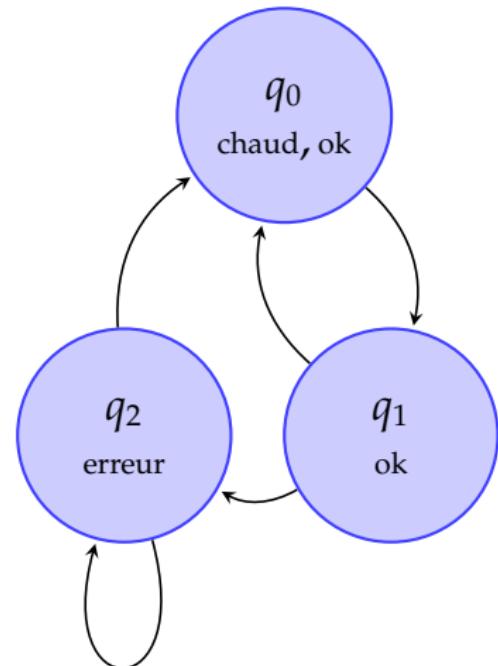
telle que $(s_i, s_{i+1}) \in R$ pour tout $i \geq 0$.

Exemple :

$$\pi_1 = q_0, q_1, q_0, q_1, \dots$$

$$\pi_2 = q_0, q_1, q_2, q_0, \dots$$

$$\pi_3 = q_0, q_1, q_2, q_2, q_2 \dots$$



Sémantique de CTL

Soit \mathcal{K} une structure de Kripke, s un état et φ une formule de CTL.

$\mathcal{K}, s \models \varphi$: à l'état s , φ est vraie

Définition : \mathcal{K} satisfait φ ($\mathcal{K} \models \varphi$) ssi $\mathcal{K}, q_0 \models \varphi$.

Définition de $\mathcal{K}, s \models \varphi$:

$\mathcal{K}, s \models p$ ssi $p \in L(s)$

$\mathcal{K}, s \models \neg\varphi$ ssi $\mathcal{K}, s \not\models \varphi$

$\mathcal{K}, s \models (\varphi_1 \wedge \varphi_2)$ ssi $\mathcal{K}, s \models \varphi_1$ et $\mathcal{K}, s \models \varphi_2$

...

Sémantique de CTL

$\mathcal{K}, s \models AX\varphi$ ssi pour tout s' tel que $(s, s') \in R$, $\mathcal{K}, s' \models \varphi$

$\mathcal{K}, s \models EX\varphi$ ssi il existe s' tel que $(s, s') \in R$ et $\mathcal{K}, s' \models \varphi$

$\mathcal{K}, s \models AF\varphi$ ssi pour tout chemin π tel que $\pi(0) = s$,
il existe j tel que $\mathcal{K}, \pi(j) \models \varphi$

$\mathcal{K}, s \models EF\varphi$ ssi il existe un chemin π tel que $\pi(0) = s$
et il existe j tel que $\mathcal{K}, \pi(j) \models \varphi$

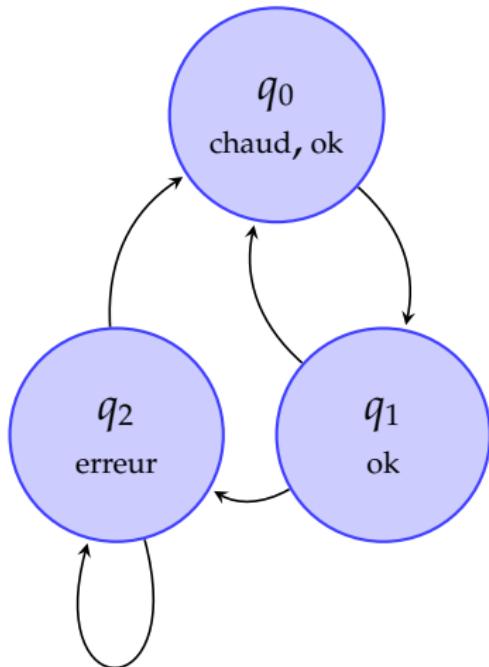
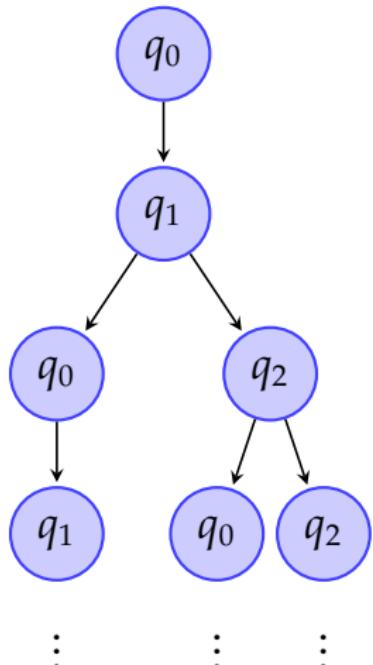
$\mathcal{K}, s \models AG\varphi$ ssi pour tout chemin π tel que $\pi(0) = s$,
pour tout j , $\mathcal{K}, \pi(j) \models \varphi$

$\mathcal{K}, s \models EG\varphi$ ssi il existe un chemin π tel que $\pi(0) = s$
et pour tout j , $\mathcal{K}, \pi(j) \models \varphi$

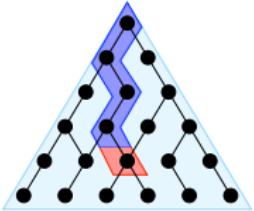
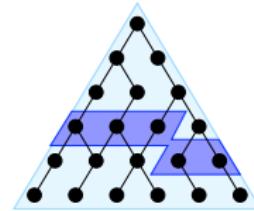
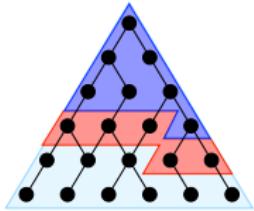
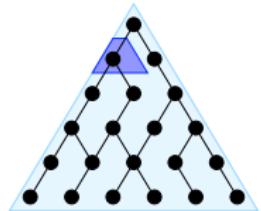
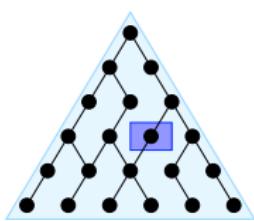
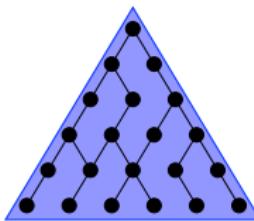
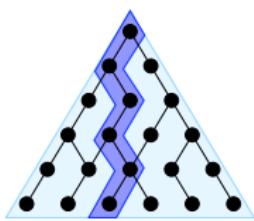
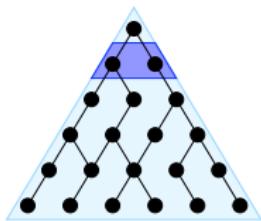
Sémantique de CTL

CTL : Logique du temps arborescent

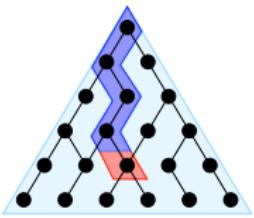
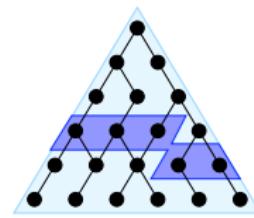
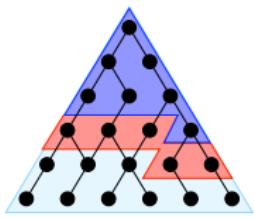
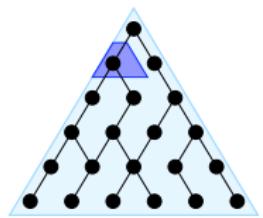
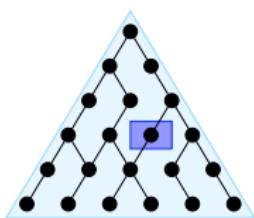
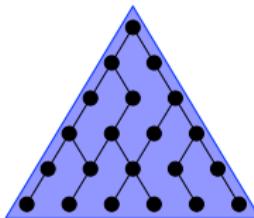
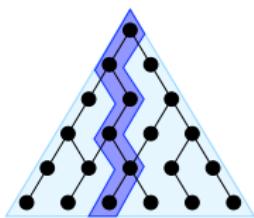
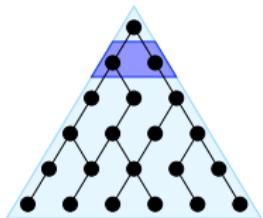
Il est utile de raisonner sur un dépliage d'une structure de Kripke : un arbre infinie (voir un DAG...) représentant tous les chemins de la structure.



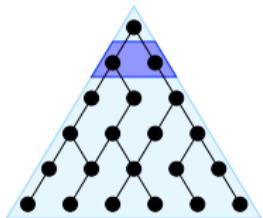
CTL : Logique du temps arborescent



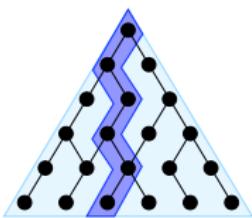
CTL : Logique du temps arborescent



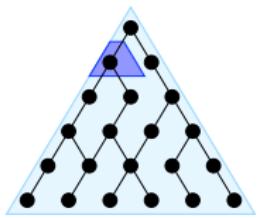
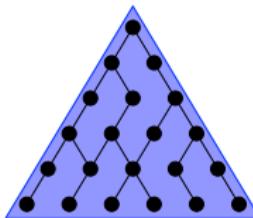
CTL : Logique du temps arborescent



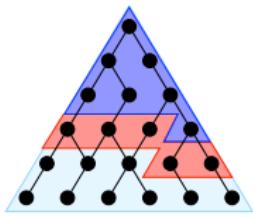
AX_P



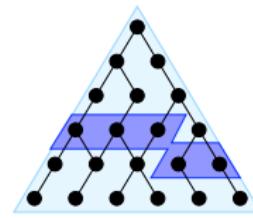
EG_P



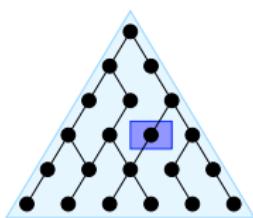
EX_P



EG

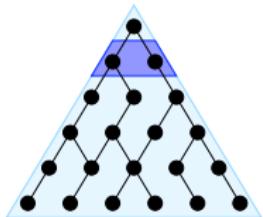


EGX

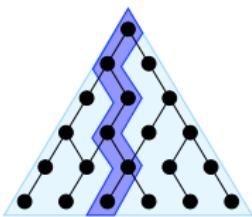


EX

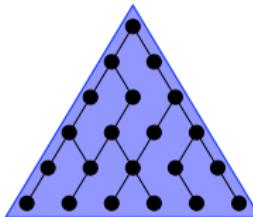
CTL : Logique du temps arborescent



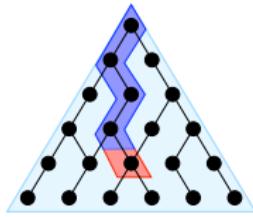
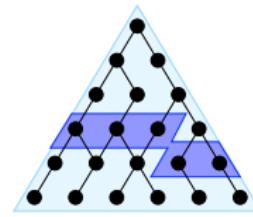
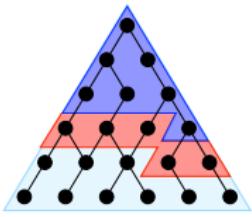
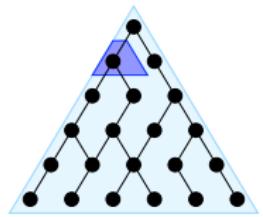
AX_P



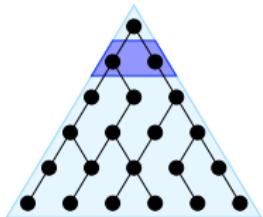
EG_P



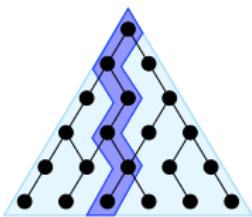
AG_P



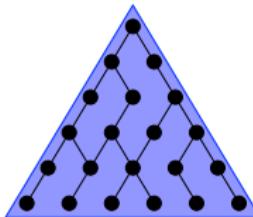
CTL : Logique du temps arborescent



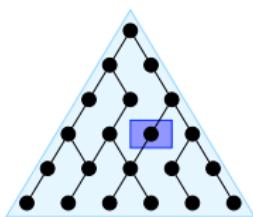
AX P



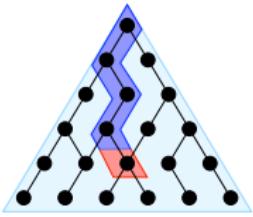
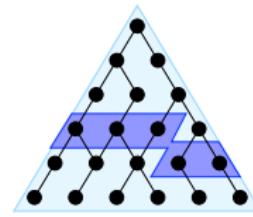
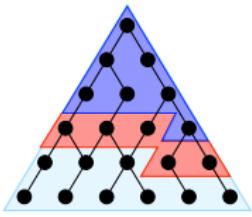
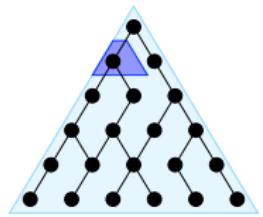
EG P



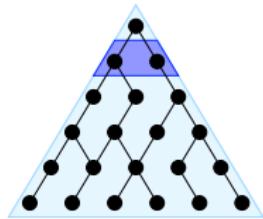
AG P



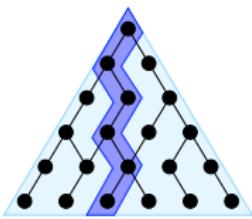
EF P



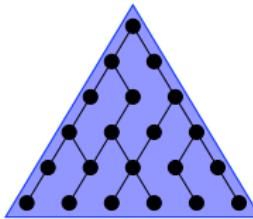
CTL : Logique du temps arborescent



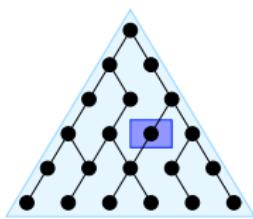
AXP



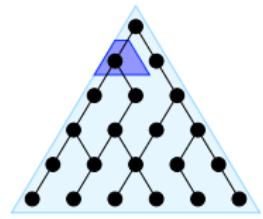
EGP



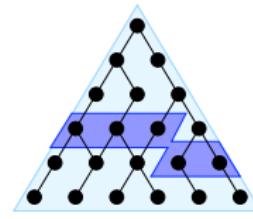
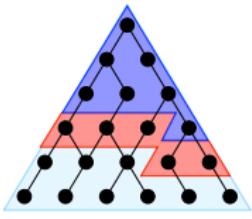
AGP



EF P

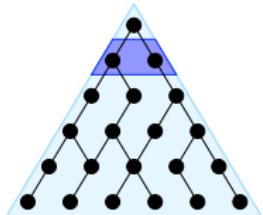


EXP

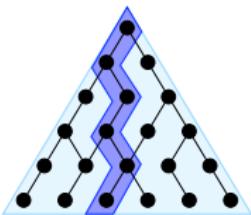


©Alessandro Artale
<http://www.inf.unibz.it/~artale/FM/fm.htm>

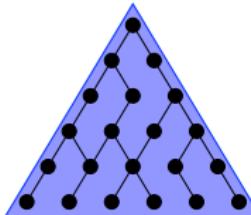
CTL : Logique du temps arborescent



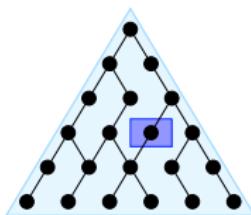
AX P



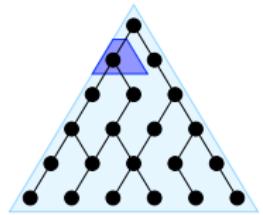
EG P



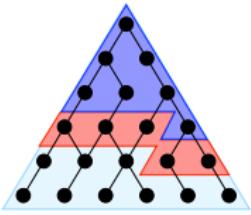
AG P



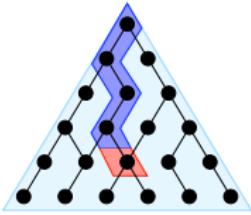
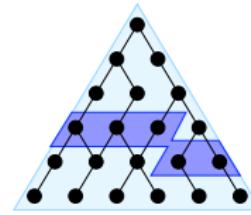
EF P



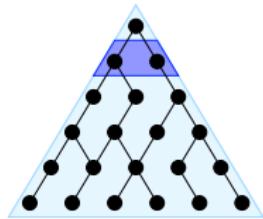
EX P



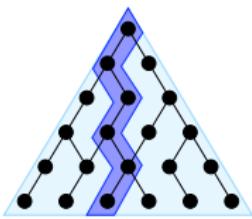
A[p U q]



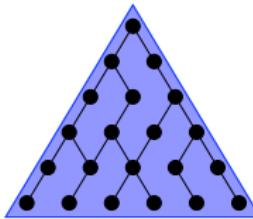
CTL : Logique du temps arborescent



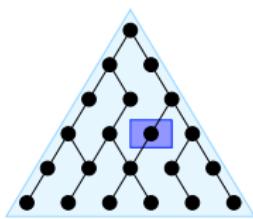
AXP



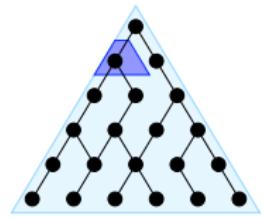
EGP



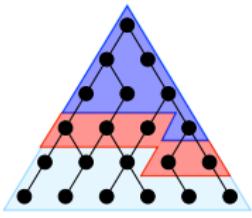
AGP



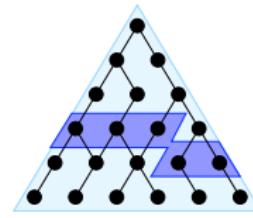
EF P



EXP



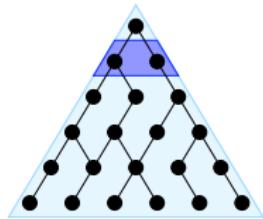
A[p U q]



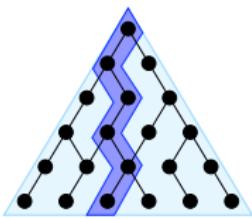
AF P

©Alessandro Artale
<http://www.inf.unibz.it/~artale/FM/fm.htm>

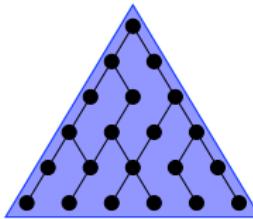
CTL : Logique du temps arborescent



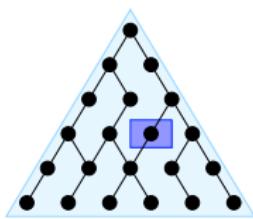
AXP



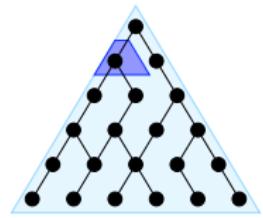
EGP



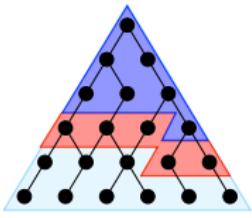
AGP



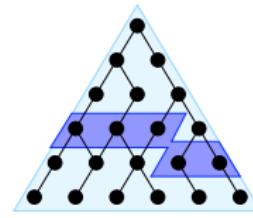
EF P



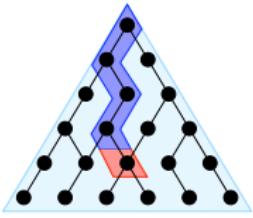
EXP



A[P U q]



AFP



E [P U q]

Equivalences sémantiques

Définition

Deux formules φ_1 et φ_2 sont *équivalentes* ($\varphi_1 \equiv \varphi_2$) ssi pour toute structure de Kripke \mathcal{K} ,

$$\mathcal{K} \models \varphi_1 \text{ ssi } \mathcal{K} \models \varphi_2$$

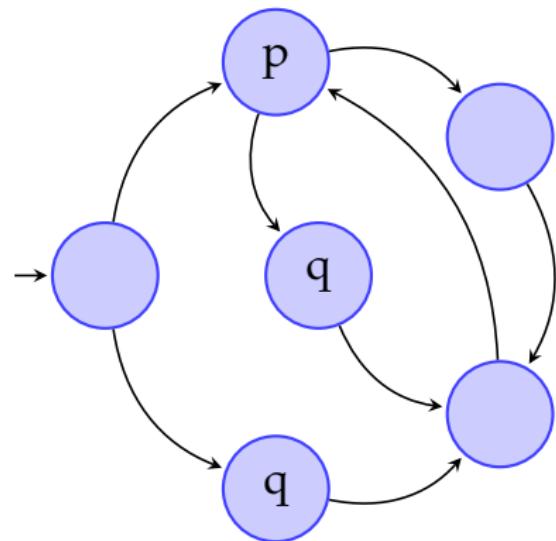
Exemple :

$$AF\varphi \equiv \neg EG\neg\varphi$$

Exercice

Pour chacune des formules de CTL suivantes, indiquer celles qui sont satisfaites par la structure de Kripke ci-contre.

- ▶ AXp
- ▶ $AX(p \Rightarrow AGAFp)$
- ▶ $A(\neg q U q)$
- ▶ $E(\neg q U q)$
- ▶ $AXA(p U q)$



Vérification de modèle

$$\mathcal{K} \models^? \varphi$$

Principe : on construit l'ensemble $\text{SAT}[\varphi]$ des états s qui satisfont φ (*i.e.* tels que $\mathcal{K}, s \models \varphi$).

$\text{SAT}[\varphi]$ est défini par induction sur φ .

SAT

$$\begin{aligned}\text{SAT}[p] &= \{ s \in S \mid p \in L(s) \} \\ \text{SAT}[\neg\varphi] &= \dots \\ \text{SAT}[(\varphi_1 \wedge \varphi_2)] &= \text{SAT}[\varphi_1] \cdots \text{SAT}[\varphi_2] \\ \text{SAT}[(\varphi_1 \vee \varphi_2)] &= \text{SAT}[\varphi_1] \cdots \text{SAT}[\varphi_2] \\ \text{SAT}[(\varphi_1 \Rightarrow \varphi_2)] &= \text{SAT}[(\neg\varphi_1 \vee \varphi_2)] \\ \text{SAT}[(\varphi_1 \Leftrightarrow \varphi_2)] &= \text{SAT}[((\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1))] \\ \text{SAT}[AX\varphi] &= \dots (\text{SAT}[\varphi]) \\ \text{SAT}[EX\varphi] &= \dots (\text{SAT}[\varphi])\end{aligned}$$

SAT

$$\begin{aligned}\text{SAT}[p] &= \{ s \in S \mid p \in L(s) \} \\ \text{SAT}[\neg\varphi] &= S - \text{SAT}[\varphi] \\ \text{SAT}[(\varphi_1 \wedge \varphi_2)] &= \text{SAT}[\varphi_1] \cdots \text{SAT}[\varphi_2] \\ \text{SAT}[(\varphi_1 \vee \varphi_2)] &= \text{SAT}[\varphi_1] \cdots \text{SAT}[\varphi_2] \\ \text{SAT}[(\varphi_1 \Rightarrow \varphi_2)] &= \text{SAT}[(\neg\varphi_1 \vee \varphi_2)] \\ \text{SAT}[(\varphi_1 \Leftrightarrow \varphi_2)] &= \text{SAT}[((\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1))] \\ \text{SAT}[AX\varphi] &= \cdots (\text{SAT}[\varphi]) \\ \text{SAT}[EX\varphi] &= \cdots (\text{SAT}[\varphi])\end{aligned}$$

SAT

$$\begin{aligned}\text{SAT}[p] &= \{ s \in S \mid p \in L(s) \} \\ \text{SAT}[\neg\varphi] &= S - \text{SAT}[\varphi] \\ \text{SAT}[(\varphi_1 \wedge \varphi_2)] &= \text{SAT}[\varphi_1] \cap \text{SAT}[\varphi_2] \\ \text{SAT}[(\varphi_1 \vee \varphi_2)] &= \text{SAT}[\varphi_1] \cdots \text{SAT}[\varphi_2] \\ \text{SAT}[(\varphi_1 \Rightarrow \varphi_2)] &= \text{SAT}[(\neg\varphi_1 \vee \varphi_2)] \\ \text{SAT}[(\varphi_1 \Leftrightarrow \varphi_2)] &= \text{SAT}[((\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1))] \\ \text{SAT}[AX\varphi] &= \cdots (\text{SAT}[\varphi]) \\ \text{SAT}[EX\varphi] &= \cdots (\text{SAT}[\varphi])\end{aligned}$$

SAT

$$\begin{aligned}\text{SAT}[p] &= \{ s \in S \mid p \in L(s) \} \\ \text{SAT}[\neg\varphi] &= S - \text{SAT}[\varphi] \\ \text{SAT}[(\varphi_1 \wedge \varphi_2)] &= \text{SAT}[\varphi_1] \cap \text{SAT}[\varphi_2] \\ \text{SAT}[(\varphi_1 \vee \varphi_2)] &= \text{SAT}[\varphi_1] \cup \text{SAT}[\varphi_2] \\ \text{SAT}[(\varphi_1 \Rightarrow \varphi_2)] &= \text{SAT}[(\neg\varphi_1 \vee \varphi_2)] \\ \text{SAT}[(\varphi_1 \Leftrightarrow \varphi_2)] &= \text{SAT}[((\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1))] \\ \text{SAT}[AX\varphi] &= \dots (\text{SAT}[\varphi]) \\ \text{SAT}[EX\varphi] &= \dots (\text{SAT}[\varphi])\end{aligned}$$

SAT

$$\begin{aligned}\text{SAT}[p] &= \{ s \in S \mid p \in L(s) \} \\ \text{SAT}[\neg\varphi] &= S - \text{SAT}[\varphi] \\ \text{SAT}[(\varphi_1 \wedge \varphi_2)] &= \text{SAT}[\varphi_1] \cap \text{SAT}[\varphi_2] \\ \text{SAT}[(\varphi_1 \vee \varphi_2)] &= \text{SAT}[\varphi_1] \cup \text{SAT}[\varphi_2] \\ \text{SAT}[(\varphi_1 \Rightarrow \varphi_2)] &= \text{SAT}[(\neg\varphi_1 \vee \varphi_2)] \\ \text{SAT}[(\varphi_1 \Leftrightarrow \varphi_2)] &= \text{SAT}[((\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1))] \\ \text{SAT}[AX\varphi] &= \text{pre}_\forall(\text{SAT}[\varphi]) \\ \text{SAT}[EX\varphi] &= \dots (\text{SAT}[\varphi])\end{aligned}$$

avec pour tout $X \subseteq S$,

$$\text{pre}_\forall(X) = \{ s \in S \mid \text{pour tout } s', (s, s') \in R \text{ implique } s' \in X \}$$

SAT

$$\begin{aligned}\text{SAT}[p] &= \{ s \in S \mid p \in L(s) \} \\ \text{SAT}[\neg\varphi] &= S - \text{SAT}[\varphi] \\ \text{SAT}[(\varphi_1 \wedge \varphi_2)] &= \text{SAT}[\varphi_1] \cap \text{SAT}[\varphi_2] \\ \text{SAT}[(\varphi_1 \vee \varphi_2)] &= \text{SAT}[\varphi_1] \cup \text{SAT}[\varphi_2] \\ \text{SAT}[(\varphi_1 \Rightarrow \varphi_2)] &= \text{SAT}[(\neg\varphi_1 \vee \varphi_2)] \\ \text{SAT}[(\varphi_1 \Leftrightarrow \varphi_2)] &= \text{SAT}[((\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1))] \\ \text{SAT}[AX\varphi] &= \text{pre}_\forall(\text{SAT}[\varphi]) \\ \text{SAT}[EX\varphi] &= \text{pre}_\exists(\text{SAT}[\varphi])\end{aligned}$$

avec pour tout $X \subseteq S$,

$$\begin{aligned}\text{pre}_\forall(X) &= \{ s \in S \mid \text{pour tout } s', (s, s') \in R \text{ implique } s' \in X \} \\ \text{pre}_\exists(X) &= \{ s \in S \mid \text{il existe } s' \in X \text{ tel que } (s, s') \in R \}\end{aligned}$$



SAT

$$\text{SAT}[AG\varphi] = \forall X. \text{SAT}(\varphi) \cap \text{pre}_\forall(X)$$

$$\text{SAT}[EG\varphi] = \forall X. \text{SAT}(\varphi) \cap \text{pre}_\exists(X)$$

$$\text{SAT}[AF\varphi] = \exists X. \text{SAT}(\varphi) \cup \text{pre}_\forall(X)$$

$$\text{SAT}[EF\varphi] = \exists X. \text{SAT}(\varphi) \cup \text{pre}_\exists(X)$$

$$\text{SAT}[A(\varphi_1 U \varphi_2)] = \exists X. \text{SAT}(\varphi_2) \cup (\text{SAT}(\varphi_1) \cap \text{pre}_\forall(X))$$

$$\text{SAT}[E(\varphi_1 U \varphi_2)] = \exists X. \text{SAT}(\varphi_2) \cup (\text{SAT}(\varphi_1) \cap \text{pre}_\exists(X))$$

avec pour tout $F \in \mathcal{P}(S) \rightarrow \mathcal{P}(S)$, monotone,

$\mu X. F(X)$: le plus petit point fixe de F

$\nu X. F(X)$: le plus grand point fixe de F

Justifications

Nous devons maintenant :

- ▶ Justifier la notation $\mu X. F(X)$ et $\nu X. F(X)$:
 - ▶ Existence d'un point fixe ?
 - ▶ Un plus petit ?
 - ▶ Un plus grand ?
 - ▶ Le plus petit ?
 - ▶ Le plus grand ?
- ▶ Donner une méthode de calcul pour $\mu X. F(X)$ et $\nu X. F(X)$.
- ▶ Justifier les équations précédentes.

Points fixes

Définition

Soit $F \in \mathcal{P}(S) \rightarrow \mathcal{P}(S)$,

- ① F est dites *monotone* si $X \subseteq Y$ implique $F(X) \subseteq F(Y)$ pour tout $X, Y \subseteq S$.
- ② $X \subseteq S$ est dit *un point fixe* de F si $F(X) = X$.

Théorème

Si S est de cardinal $n \in \mathbb{N}$, si $F \in \mathcal{P}(S) \rightarrow \mathcal{P}(S)$ est monotone, alors $F^n(\emptyset)$ est le plus petit point fixe de F et $F^n(S)$ est le plus grand.

Justifications des équations

Lemme

Soit $Y \subseteq S$ et $F \in \mathcal{P}(S) \rightarrow \mathcal{P}(S)$ une fonction monotone, les fonctions suivantes sont monotones :

- ▶ pre_{\forall}
- ▶ pre_{\exists}
- ▶ $X \mapsto Y \cup F(X)$
- ▶ $X \mapsto Y \cap F(X)$

Corollaire

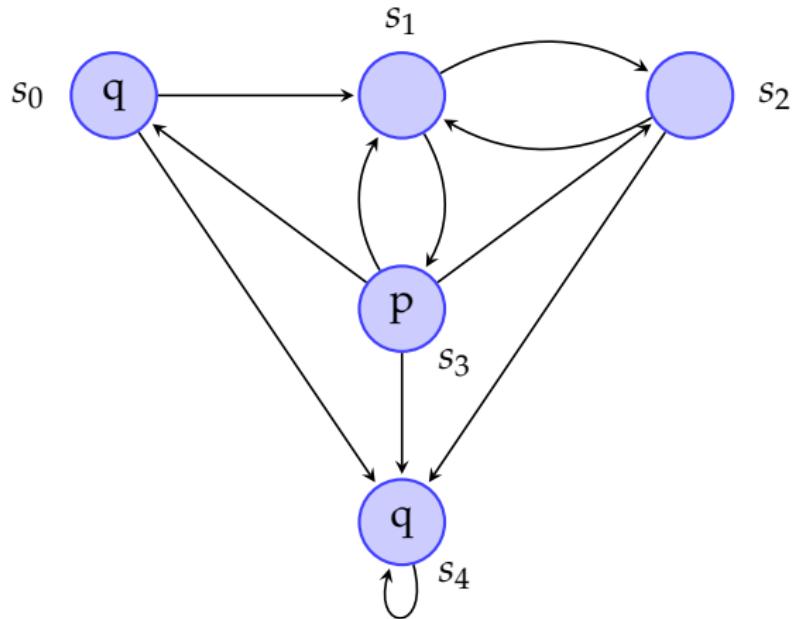
Les équations définissant $\text{SAT}[\text{AG}\varphi]$, $\text{SAT}[\text{EG}\varphi]$, $\text{SAT}[\text{AF}\varphi]$, $\text{SAT}[\text{EF}\varphi]$, $\text{SAT}[A(\varphi_1 \cup \varphi_2)]$ et $\text{SAT}[E(\varphi_1 \cup \varphi_2)]$ sont bien formées.

Correction de SAT

Théorème

Pour toute formule CTL φ , $SAT(\varphi) = \{ s \in S \mid \mathcal{K}, s \models \varphi \}$.

Exercices

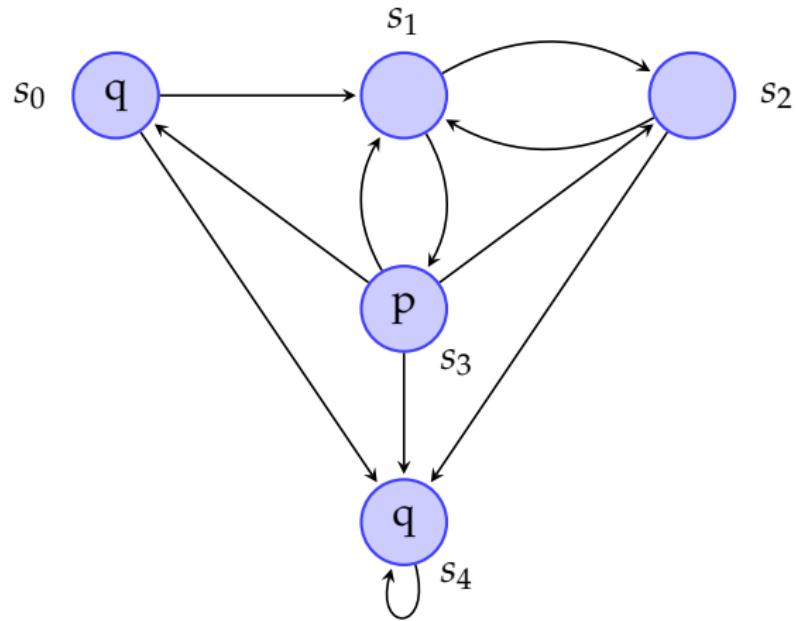


Calculer $\text{SAT}(EFp)$.

Calculer $\text{SAT}(EGq)$.

$SAT(EFp)$

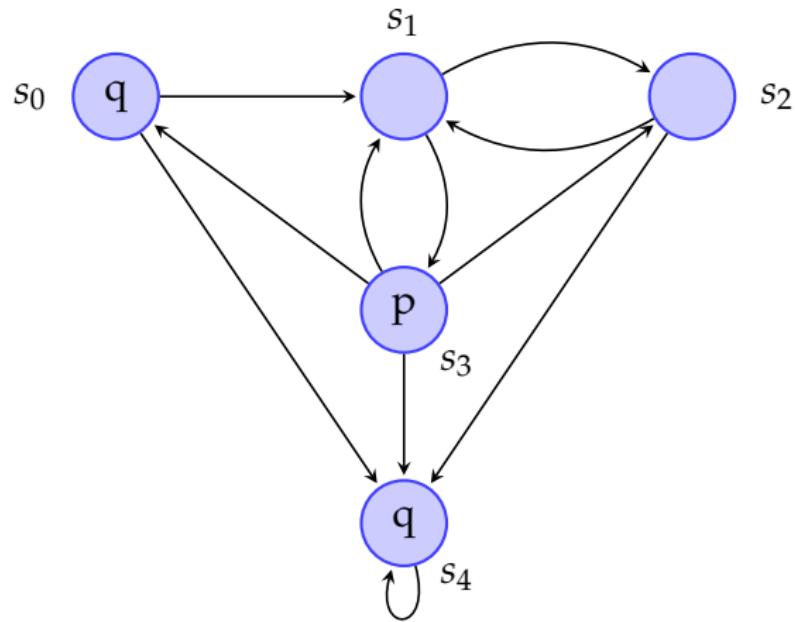
$$SAT(EFp) = \mu X.SAT(p) \cup pre_{\exists}(X) = \mu X.F(X)$$



SAT(EFp)

$$\text{SAT}(EFp) = \mu X.\text{SAT}(p) \cup \text{pre}_{\exists}(X) = \mu X.F(X)$$

$$X_0 = \emptyset$$

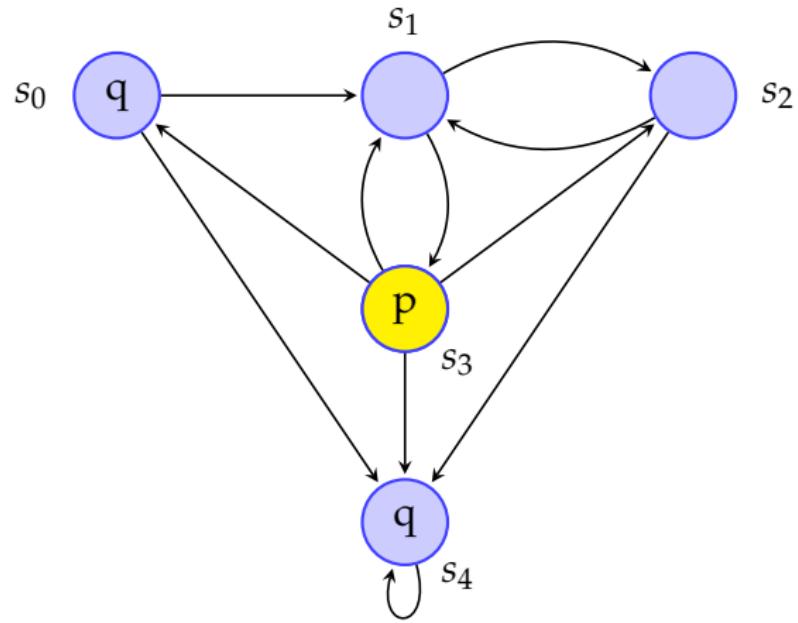


SAT(EFp)

$$\text{SAT}(EFp) = \mu X.\text{SAT}(p) \cup \text{pre}_{\exists}(X) = \mu X.F(X)$$

$$X_0 = \emptyset$$

$$X_1 = F(X_0) = \{s_3\}$$



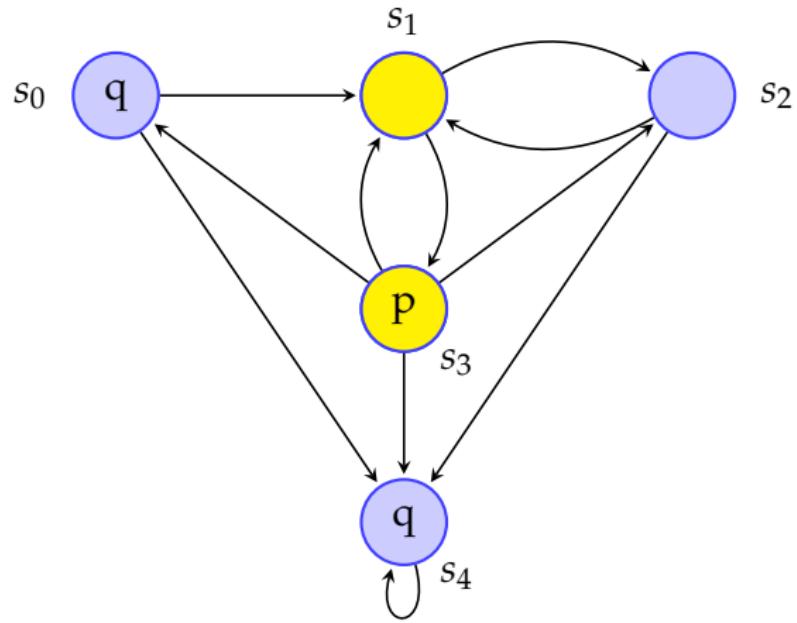
SAT(EFp)

$$\text{SAT}(EFp) = \mu X.\text{SAT}(p) \cup \text{pre}_{\exists}(X) = \mu X.F(X)$$

$$X_0 = \emptyset$$

$$X_1 = F(X_0) = \{s_3\}$$

$$X_2 = F(X_1) = \{s_1, s_3\}$$



SAT(EFp)

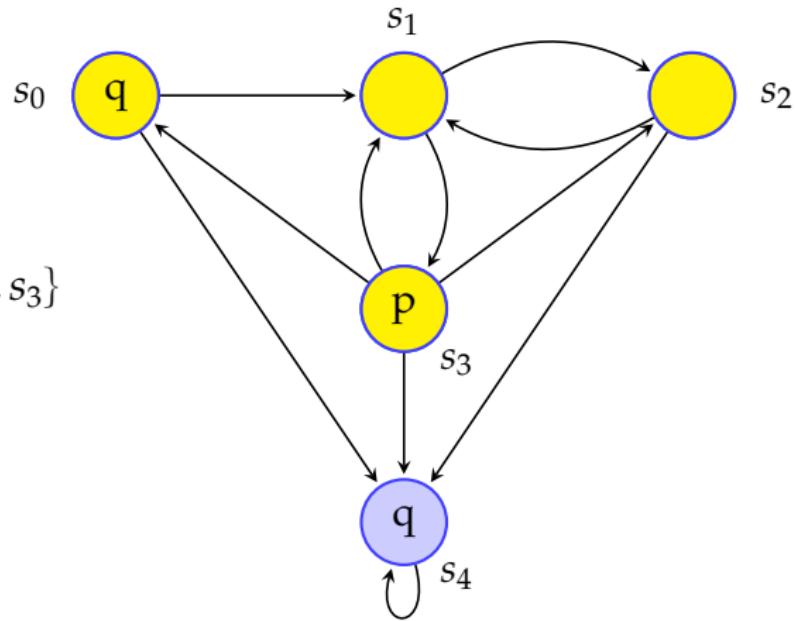
$$\text{SAT}(EFp) = \mu X.\text{SAT}(p) \cup \text{pre}_{\exists}(X) = \mu X.F(X)$$

$$X_0 = \emptyset$$

$$X_1 = F(X_0) = \{s_3\}$$

$$X_2 = F(X_1) = \{s_1, s_3\}$$

$$X_3 = F(X_2) = \{s_0, s_1, s_2, s_3\}$$



SAT(EFp)

$$\text{SAT}(EFp) = \mu X.\text{SAT}(p) \cup \text{pre}_{\exists}(X) = \mu X.F(X)$$

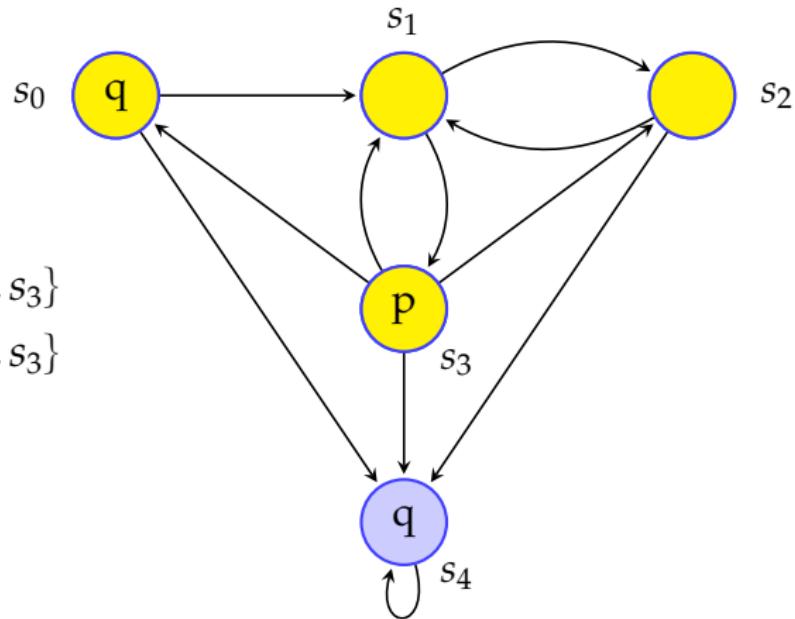
$$X_0 = \emptyset$$

$$X_1 = F(X_0) = \{s_3\}$$

$$X_2 = F(X_1) = \{s_1, s_3\}$$

$$X_3 = F(X_2) = \{s_0, s_1, s_2, s_3\}$$

$$X_4 = F(X_3) = \{s_0, s_1, s_2, s_3\}$$



SAT(EFp)

$$\text{SAT}(EFp) = \mu X.\text{SAT}(p) \cup \text{pre}_{\exists}(X) = \mu X.F(X)$$

$$X_0 = \emptyset$$

$$X_1 = F(X_0) = \{s_3\}$$

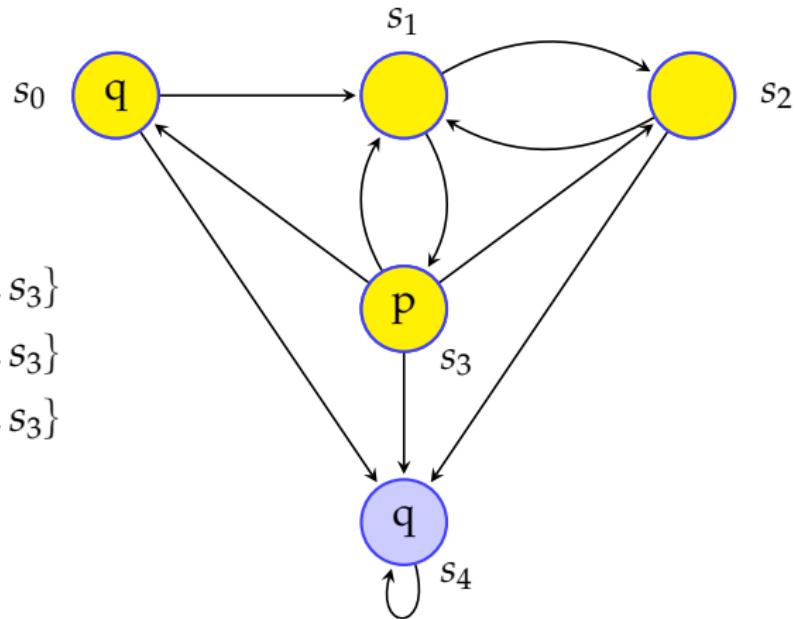
$$X_2 = F(X_1) = \{s_1, s_3\}$$

$$X_3 = F(X_2) = \{s_0, s_1, s_2, s_3\}$$

$$X_4 = F(X_3) = \{s_0, s_1, s_2, s_3\}$$

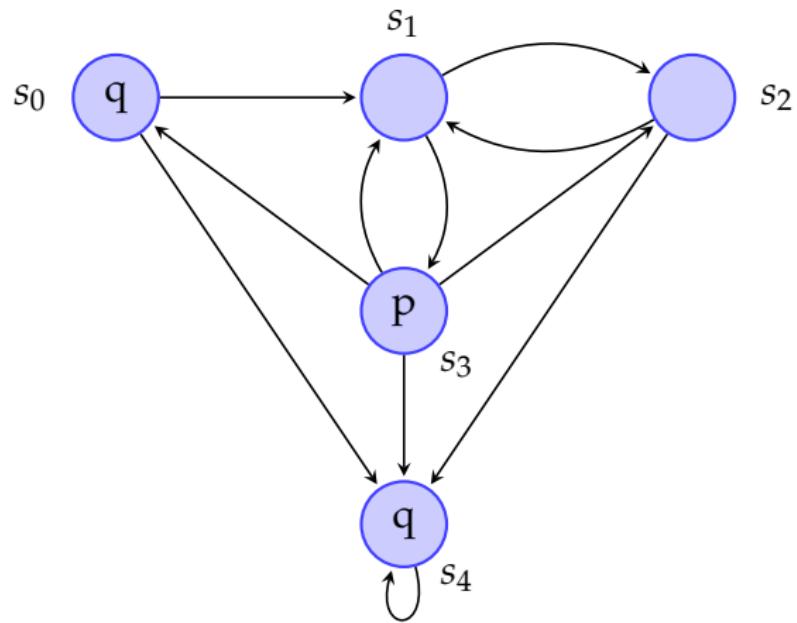
$$X_5 = F(X_4) = \{s_0, s_1, s_2, s_3\}$$

$$\text{SAT}(EFp) = \{s_0, s_1, s_2, s_3\}$$



SAT(EGq)

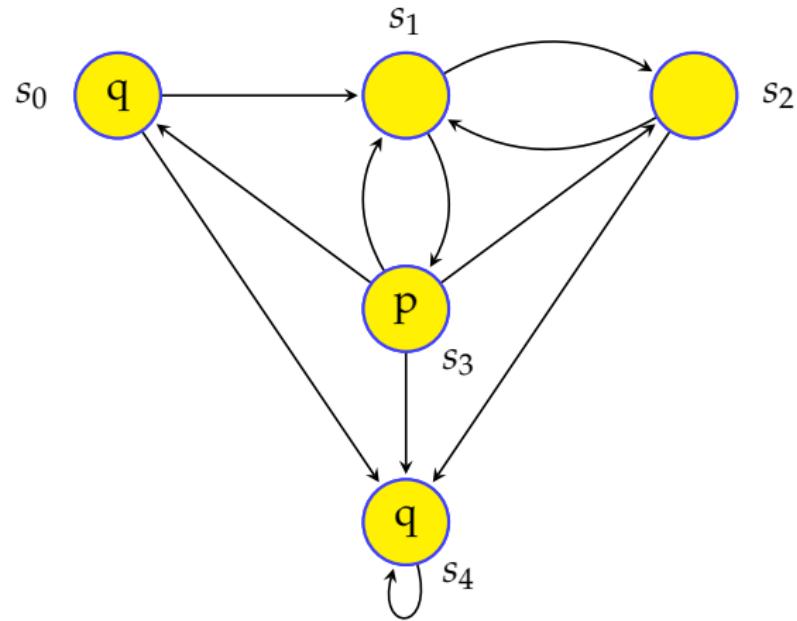
$$\text{SAT}(EGq) = \forall X.\text{SAT}(q) \cap \text{pre}_{\exists}(X) = \forall X.F(X)$$



SAT(EGq)

$$\text{SAT}(EGq) = \forall X.\text{SAT}(q) \cap \text{pre}_{\exists}(X) = \forall X.F(X)$$

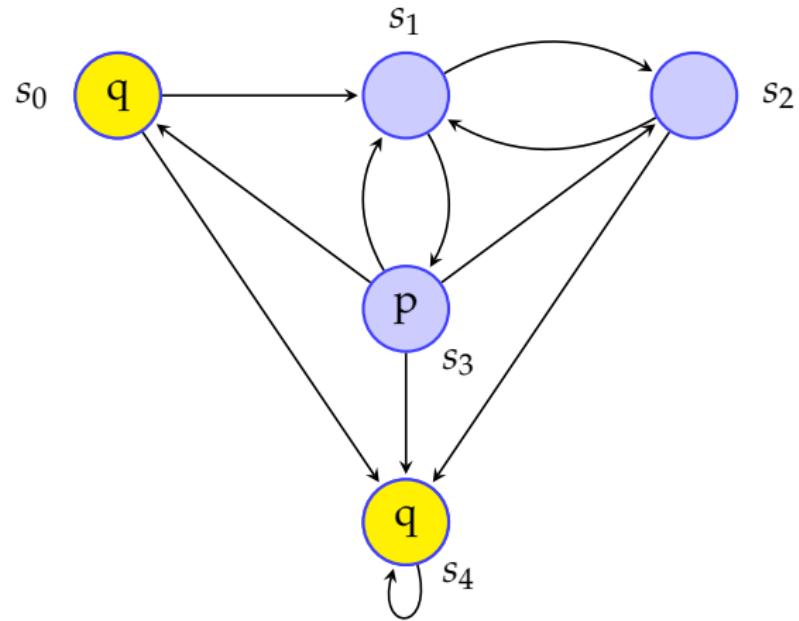
$$X_0 = \{s_0, s_1, s_2, s_3\}$$



SAT(EGq)

$$\text{SAT}(EGq) = \forall X.\text{SAT}(q) \cap \text{pre}_{\exists}(X) = \forall X.F(X)$$

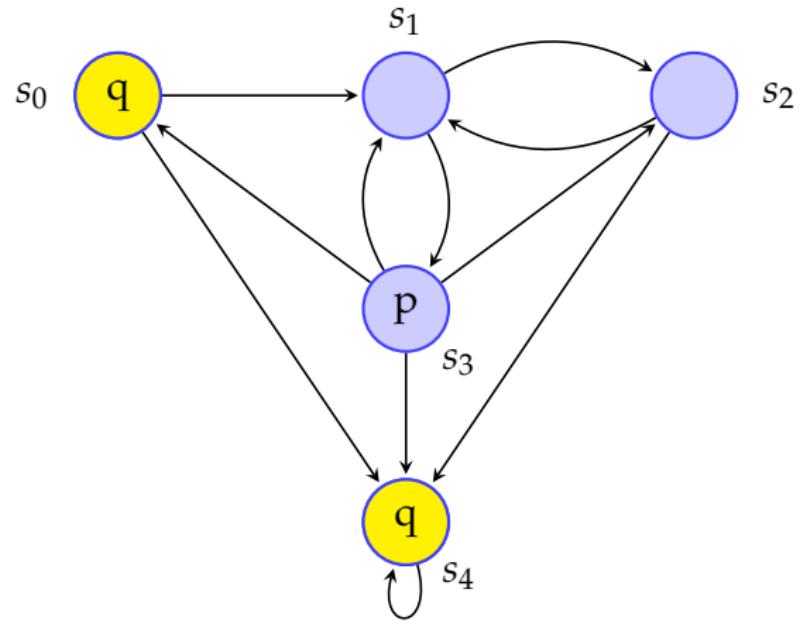
$$\begin{aligned} X_0 &= \{s_0, s_1, s_2, s_3\} \\ X_1 &= F(X_0) = \{s_0, s_4\} \end{aligned}$$



SAT(EGq)

$$\text{SAT}(EGq) = \forall X.\text{SAT}(q) \cap \text{pre}_{\exists}(X) = \forall X.F(X)$$

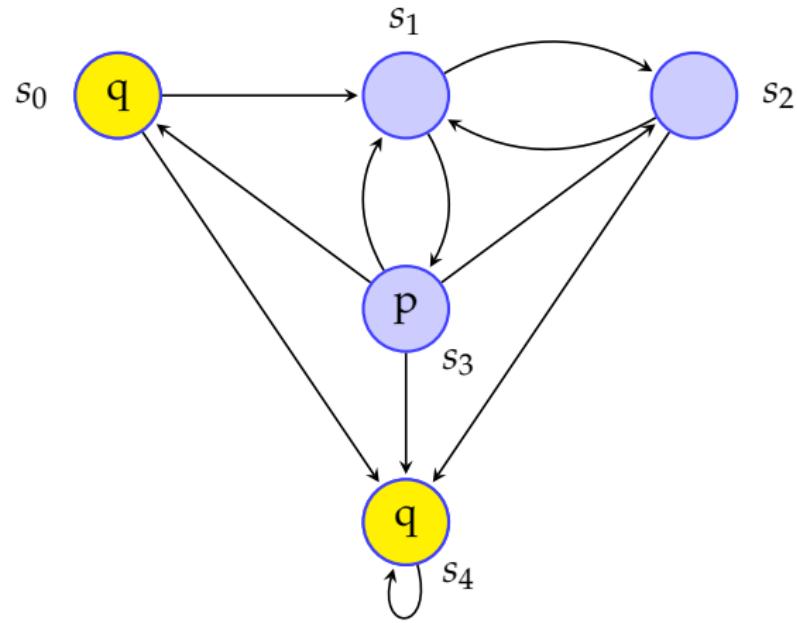
$$\begin{aligned}X_0 &= \{s_0, s_1, s_2, s_3\} \\X_1 &= F(X_0) = \{s_0, s_4\} \\X_2 &= F(X_1) = \{s_0\}\end{aligned}$$



SAT(EGq)

$$\text{SAT}(EGq) = \forall X.\text{SAT}(q) \cap \text{pre}_{\exists}(X) = \forall X.F(X)$$

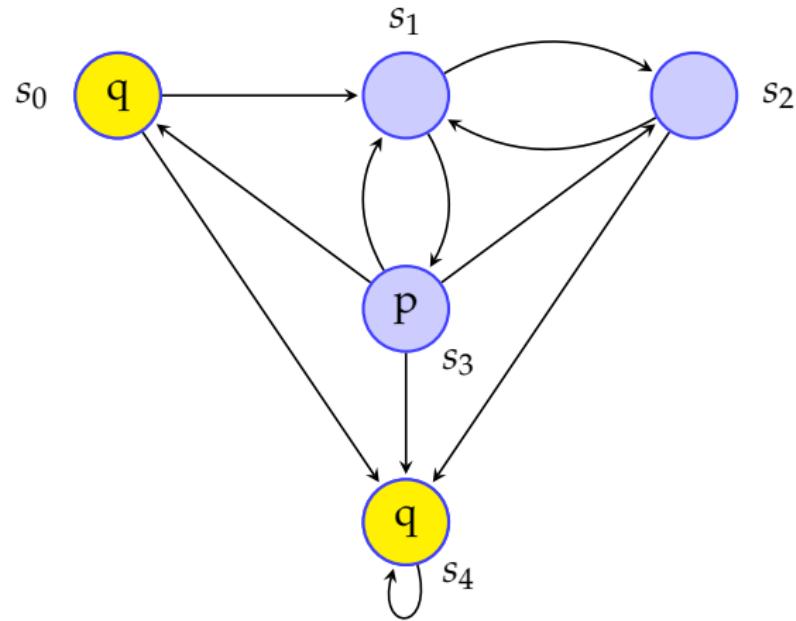
$$\begin{aligned}X_0 &= \{s_0, s_1, s_2, s_3\} \\X_1 &= F(X_0) = \{s_0, s_4\} \\X_2 &= F(X_1) = \{s_0, s_4\} \\X_3 &= F(X_2) = \{s_0, s_4\}\end{aligned}$$



SAT(EGq)

$$\text{SAT}(EGq) = \forall X.\text{SAT}(q) \cap \text{pre}_{\exists}(X) = \forall X.F(X)$$

$$\begin{aligned}X_0 &= \{s_0, s_1, s_2, s_3\} \\X_1 &= F(X_0) = \{s_0, s_4\} \\X_2 &= F(X_1) = \{s_0, s_4\} \\X_3 &= F(X_2) = \{s_0, s_4\} \\X_4 &= F(X_3) = \{s_0, s_4\}\end{aligned}$$

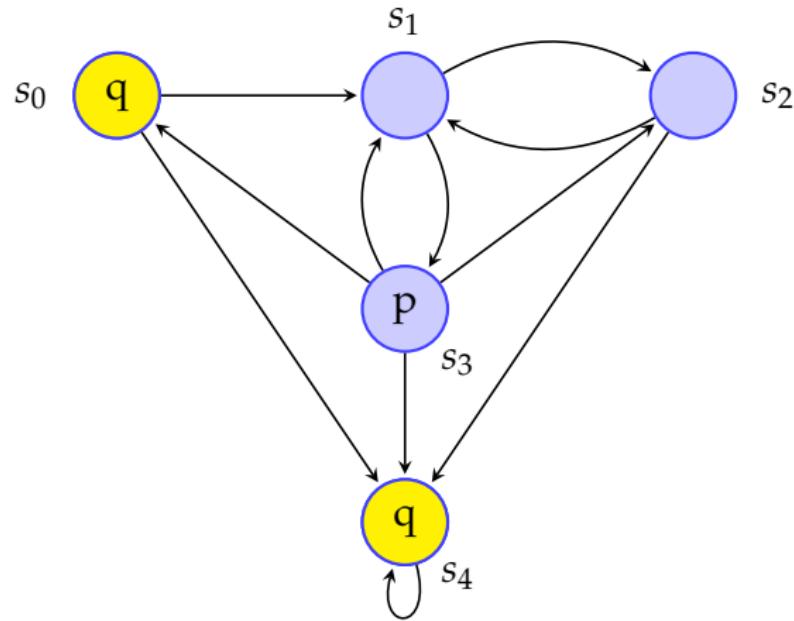


SAT(EGq)

$$\text{SAT}(EGq) = \forall X.\text{SAT}(q) \cap \text{pre}_{\exists}(X) = \forall X.F(X)$$

$$\begin{aligned}X_0 &= \{s_0, s_1, s_2, s_3\} \\X_1 &= F(X_0) = \{s_0, s_4\} \\X_2 &= F(X_1) = \{s_0, s_4\} \\X_3 &= F(X_2) = \{s_0, s_4\} \\X_4 &= F(X_3) = \{s_0, s_4\} \\X_5 &= F(X_4) = \{s_0, s_4\}\end{aligned}$$

$$\text{SAT}(EGq) = \{s_0, s_4\}$$



Cas d'étude : problème d'exclusion mutuelle

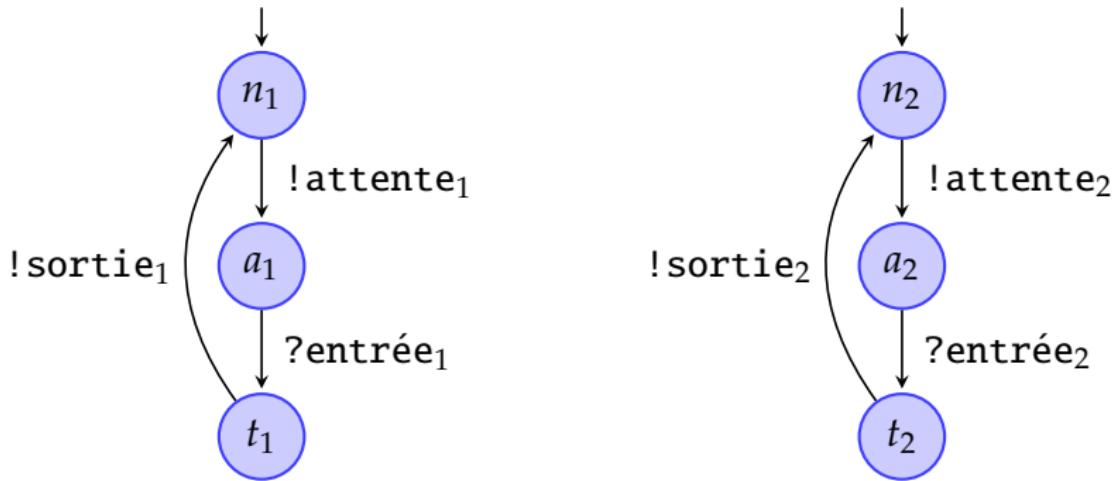
On considère un tunnel de montagne spécialement étroit qui ne permet le passage que d'un seul train à la fois. Deux trains circulent sur cette ligne.

Afin d'assurer la sécurité des voyageurs, chaque train peut échanger des signaux avec l'ordinateur qui assure le trafic dans le tunnel (le médiateur). Ces signaux sont de trois types :

- ▶ **attente** : le train veut traverser le tunnel et attend une autorisation,
- ▶ **entrée** : le train obtient l'autorisation d'entrer dans le tunnel,
- ▶ **sortie** : le train sort du tunnel.

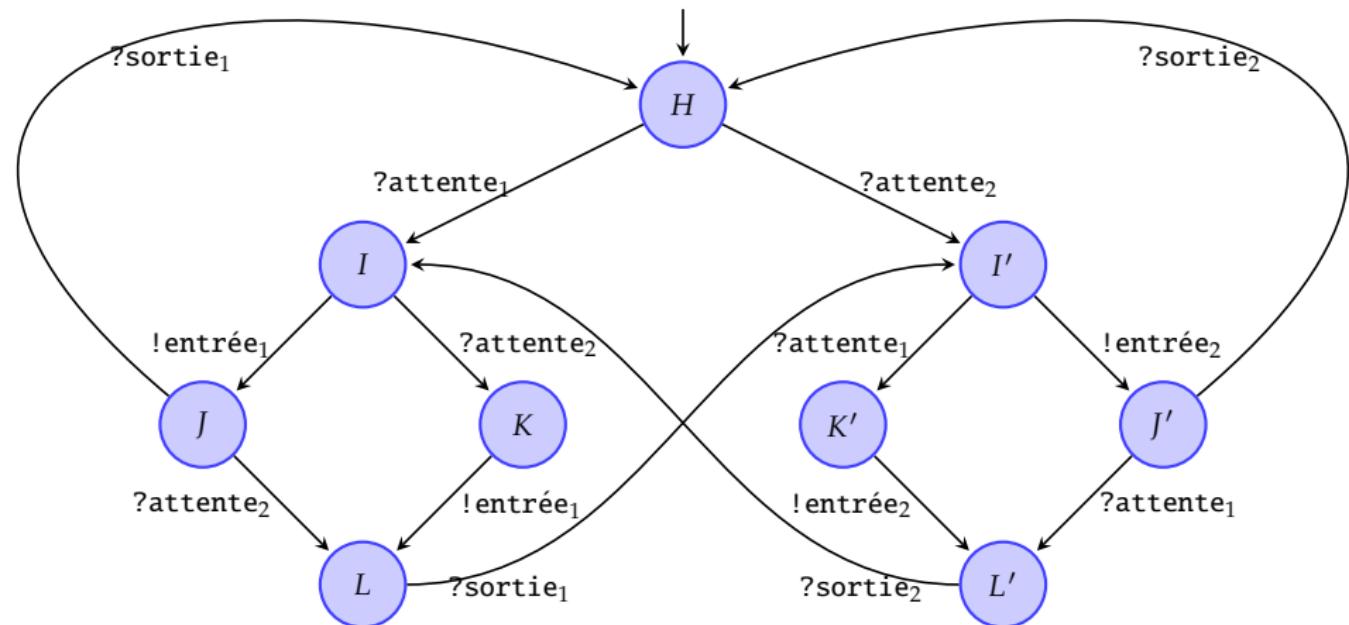
Chaque signal est indexé par le numéro du train concerné. Les signaux émis sont notés $!x$, les signaux reçus $?x$.

Le comportement des deux trains est ainsi modélisé par les automates suivants



n_i représente l'état neutre, a_i l'état d'attente et t_i l'état de traversée, pour le train i .

Le fonctionnement du médiateur est modélisé par l'automate suivant



Pour modéliser le système global, on réalise un *produit synchrone* des trois automates.

Un produit synchrone est un produit d'automates pour lequel on ne considère qu'un certain type de transitions.

Dans le cas présent, on réalise une synchronisation par messages et on se limite donc aux transitions qui contiennent à la fois $?x$ et $!x$ pour un signal x donné.

Formellement, on considère une famille de n automates $\mathcal{A}_i = (Q_i, \Sigma_i, \delta_i, q_{0,i}, l_i)$, $i = 1, \dots, n$. Le *produit cartésien* est alors défini comme l'automate $\mathcal{A} = (Q, \Sigma, \delta, q_0, l)$ avec

- ▶ $Q = Q_1 \times \cdots \times Q_n$
- ▶ $\Sigma = \prod_{1 \leq i \leq n} (\Sigma_i \cup \varepsilon)$
- ▶ $\delta = \left\{ \begin{array}{l} ((q_1, \dots, q_n), (a_1, \dots, a_n), (q'_1, \dots, q'_n)) \mid \text{pour tout } i, \\ a_i = \varepsilon \text{ et } q'_i = q_i \text{ ou } a_i \neq \varepsilon \text{ et } (q_i, a_i, q'_i) \in \Sigma_i \end{array} \right\}$
- ▶ $q_0 = (q_{0,1}, \dots, q_{0,n})$
- ▶ $l((q_1, \dots, q_n)) = \bigcup_{1 \leq i \leq n} l_i(q_i)$

Pour obtenir un *produit synchrone*, on restreint les transitions autorisées dans le produit cartésien en se donnant un *ensemble de synchronisations*

$$\text{Sync} \subseteq \prod_{1 \leq i \leq n} (\Sigma_i \cup \varepsilon)$$

Pour une *synchronisation par messages*

$$\text{Sync} = \left\{ (a_1, \dots, a_n) \mid \begin{array}{l} \text{il existe } i, j \text{ tels que } a_i = !x, a_j = ?x \\ \text{et } a_k = \varepsilon \text{ sinon} \end{array} \right\}$$

Exercices

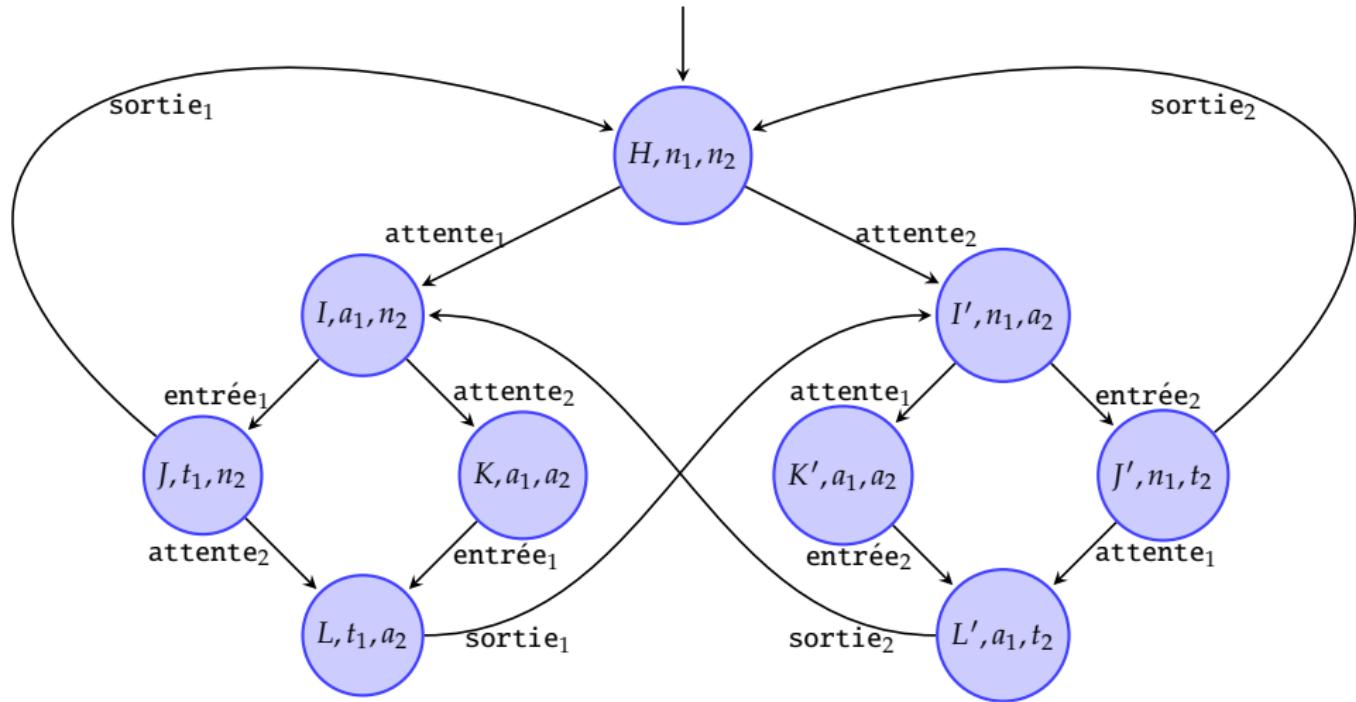
Construire l'automate résultant de ce produit.

Exprimer en CTL les propriétés suivantes :

- ▶ Les deux trains ne traversent jamais en même temps le tunnel (sûreté de l'accès).
- ▶ Un train en attente fini toujours par traverser le tunnel (inéluctabilité de l'accès).
- ▶ Un train qui est sorti peut toujours se mettre en attente (non-blockage).

Utiliser SAT pour déterminer les états de l'automate produit qui vérifient ces propriétés.

Automate produit



Propriétés CTL

- ▶ Les deux trains ne traversent jamais en même temps le tunnel.
- ▶ Un train en attente fini toujours par traverser le tunnel.
- ▶ Un train qui est sorti peut toujours se mettre en attente.

Propriétés CTL

- ▶ Les deux trains ne traversent jamais en même temps le tunnel.

$$AG\neg(t_1 \wedge t_2)$$

- ▶ Un train en attente fini toujours par traverser le tunnel.
- ▶ Un train qui est sorti peut toujours se mettre en attente.

Propriétés CTL

- ▶ Les deux trains ne traversent jamais en même temps le tunnel.

$$AG\neg(t_1 \wedge t_2)$$

- ▶ Un train en attente fini toujours par traverser le tunnel.

$$AG(a_i \Rightarrow AFt_i) \ i \in \{1, 2\}$$

- ▶ Un train qui est sorti peut toujours se mettre en attente.

Propriétés CTL

- ▶ Les deux trains ne traversent jamais en même temps le tunnel.

$$AG\neg(t_1 \wedge t_2)$$

- ▶ Un train en attente fini toujours par traverser le tunnel.

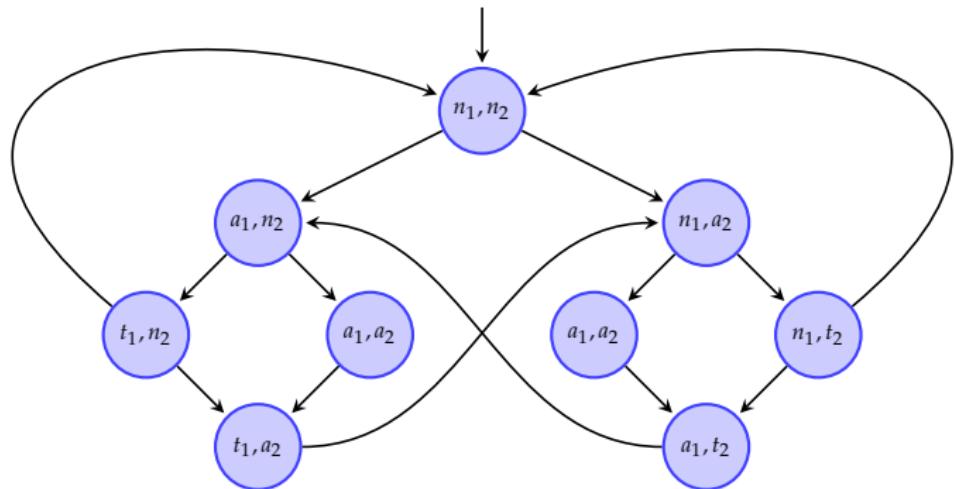
$$AG(a_i \Rightarrow AFt_i) \ i \in \{1, 2\}$$

- ▶ Un train qui est sorti peut toujours se mettre en attente.

$$AG(n_i \Rightarrow EXa_i) \ i \in \{1, 2\}$$

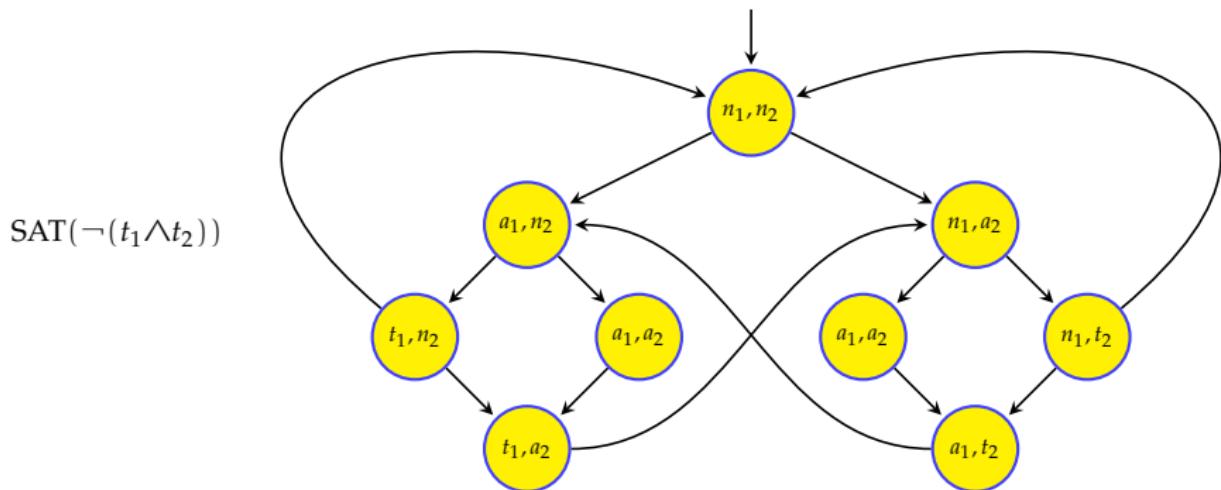
Les deux trains ne traversent jamais en même temps le tunnel

$$\begin{aligned}\text{SAT}(AG\neg(t_1 \wedge t_2)) &= \nu X. \text{SAT}(\neg(t_1 \wedge t_2)) \cap \text{pre}_\forall(X) \\ &= F^9(S)\end{aligned}$$



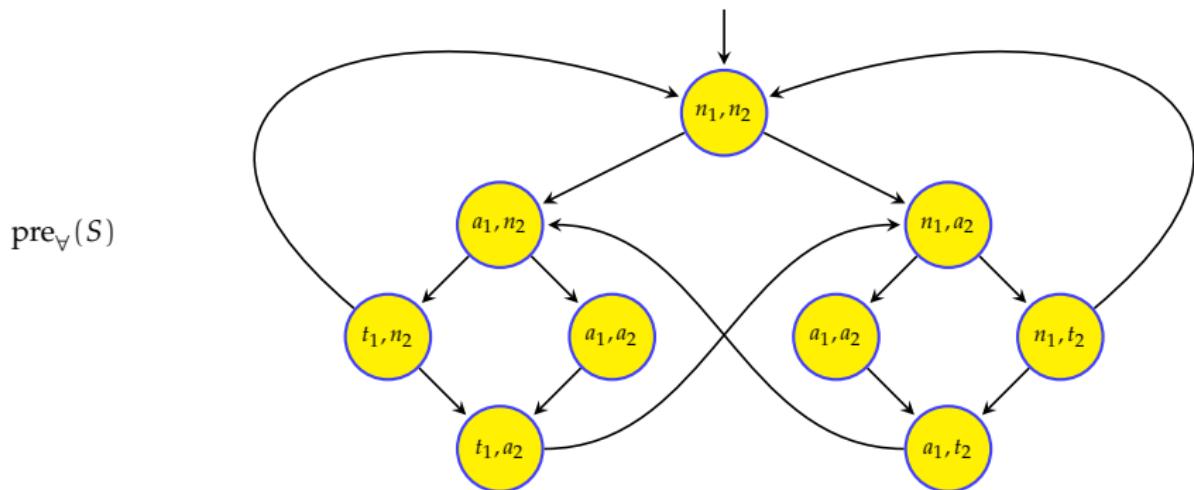
Les deux trains ne traversent jamais en même temps le tunnel

$$\begin{aligned}\text{SAT}(AG\neg(t_1 \wedge t_2)) &= \nu X. \text{SAT}(\neg(t_1 \wedge t_2)) \cap \text{pre}_\forall(X) \\ &= F^9(S)\end{aligned}$$



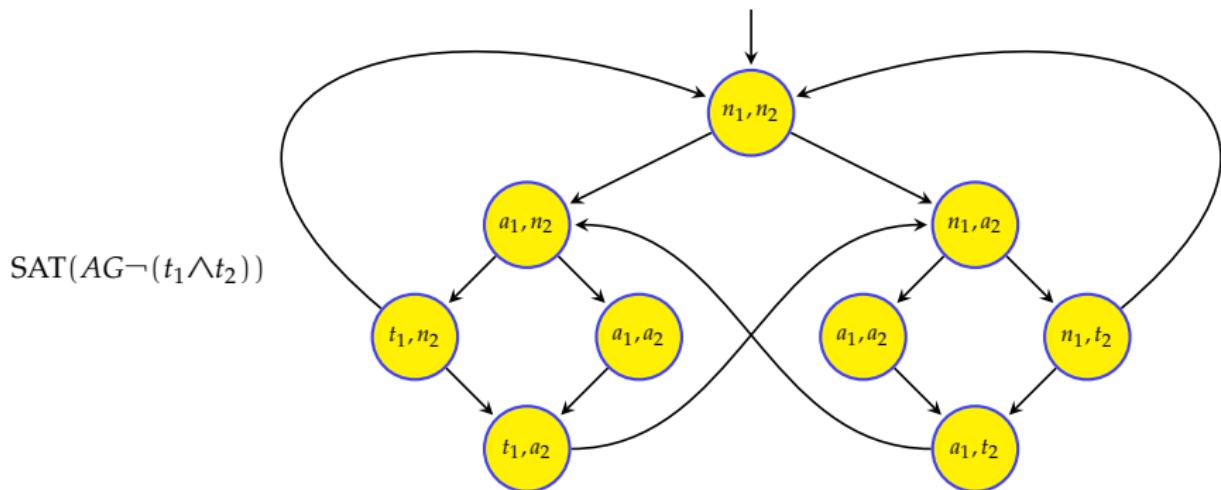
Les deux trains ne traversent jamais en même temps le tunnel

$$\begin{aligned}\text{SAT}(AG\neg(t_1 \wedge t_2)) &= \nu X. \text{SAT}(\neg(t_1 \wedge t_2)) \cap \text{pre}_\vee(X) \\ &= F^9(S)\end{aligned}$$



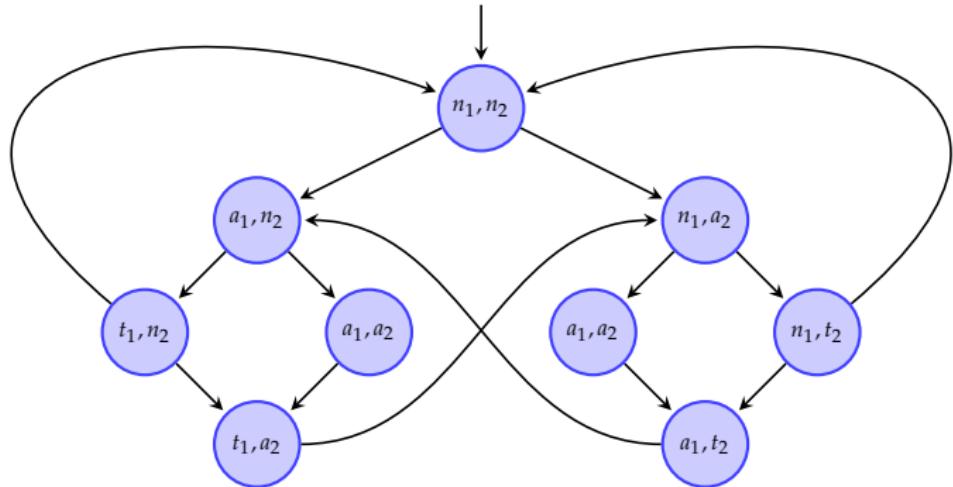
Les deux trains ne traversent jamais en même temps le tunnel

$$\begin{aligned}\text{SAT}(AG\neg(t_1 \wedge t_2)) &= \nu X. \text{SAT}(\neg(t_1 \wedge t_2)) \cap \text{pre}_\forall(X) \\ &= F^9(S)\end{aligned}$$



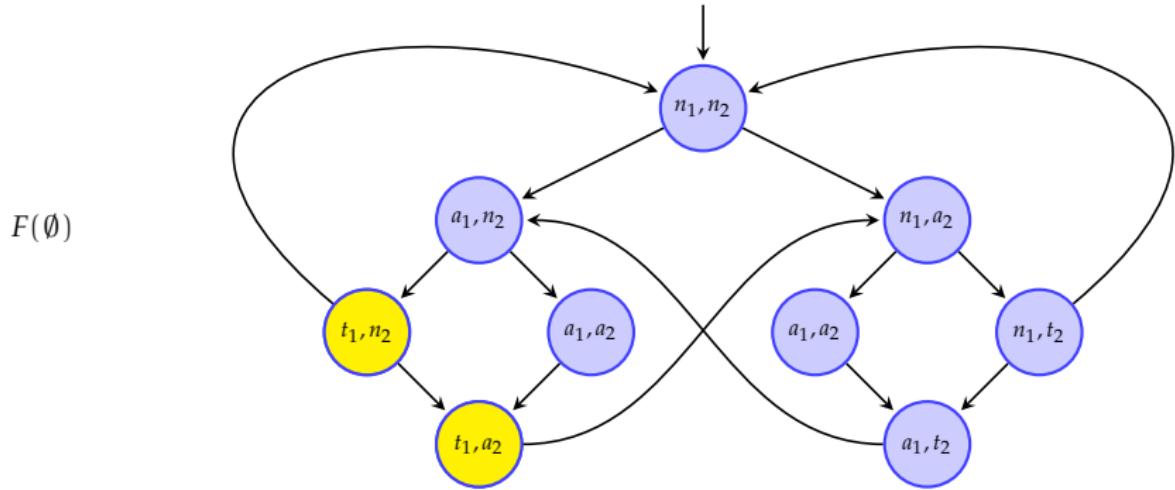
Un train en attente fini toujours par traverser le tunnel

$$\begin{aligned}\text{SAT}(\text{AG}(a_1 \Rightarrow AFt_1)) &= \forall X. (\text{SAT}(\neg a_1) \cup \text{SAT}(AFt_1)) \cap \text{pre}_\forall(X) \\ \text{SAT}(AFt_1) &= \mu X. \text{SAT}(t_1) \cup \text{pre}_\forall(X) = F^9(\emptyset)\end{aligned}$$



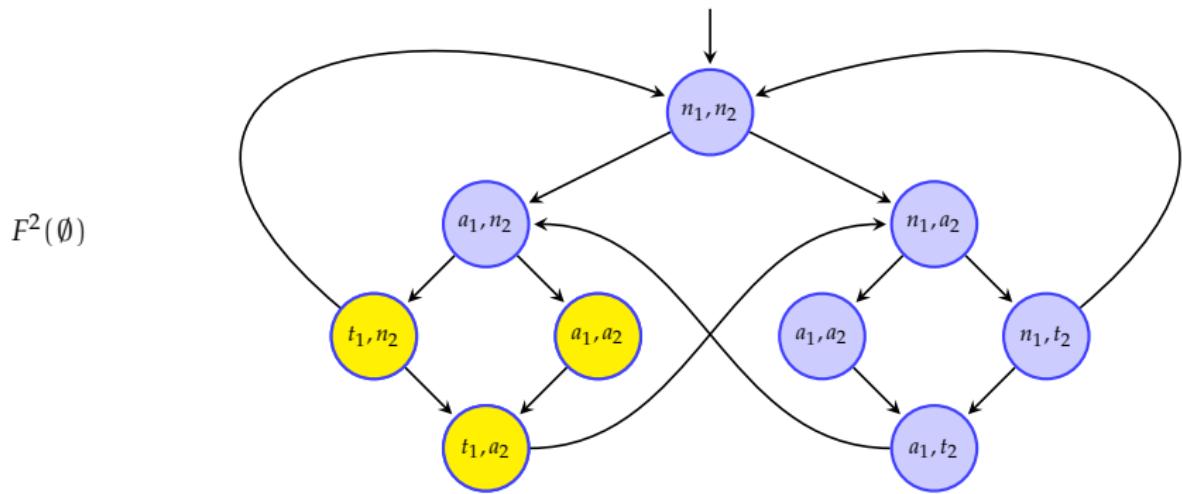
Un train en attente fini toujours par traverser le tunnel

$$\begin{aligned}\text{SAT}(\text{AG}(a_1 \Rightarrow AFt_1)) &= \forall X. (\text{SAT}(\neg a_1) \cup \text{SAT}(AFt_1)) \cap \text{pre}_\forall(X) \\ \text{SAT}(AFt_1) &= \mu X. \text{SAT}(t_1) \cup \text{pre}_\forall(X) = F^9(\emptyset)\end{aligned}$$



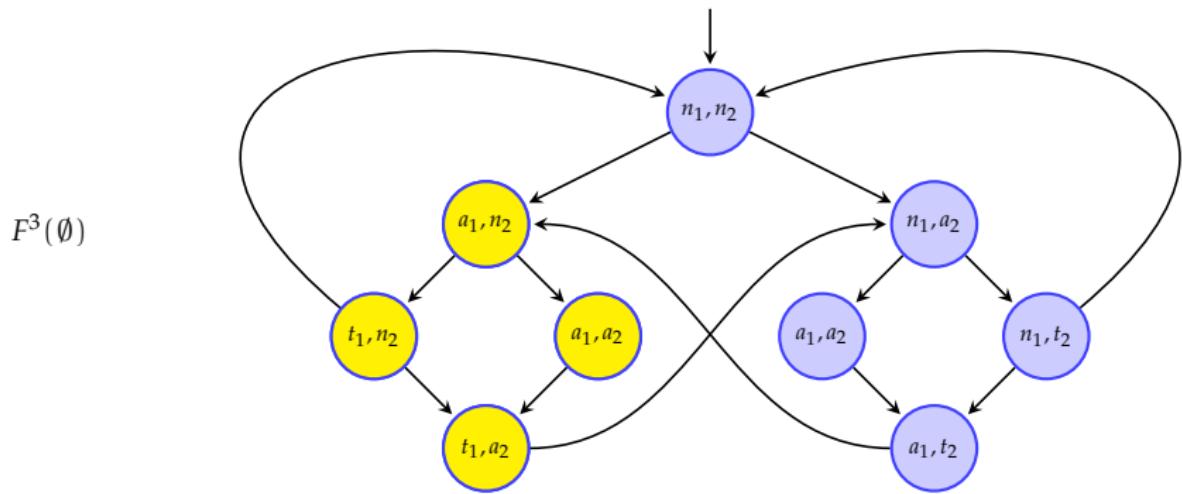
Un train en attente fini toujours par traverser le tunnel

$$\begin{aligned}\text{SAT}(\text{AG}(a_1 \Rightarrow AFt_1)) &= \forall X. (\text{SAT}(\neg a_1) \cup \text{SAT}(AFt_1)) \cap \text{pre}_\forall(X) \\ \text{SAT}(AFt_1) &= \mu X. \text{SAT}(t_1) \cup \text{pre}_\forall(X) = F^9(\emptyset)\end{aligned}$$



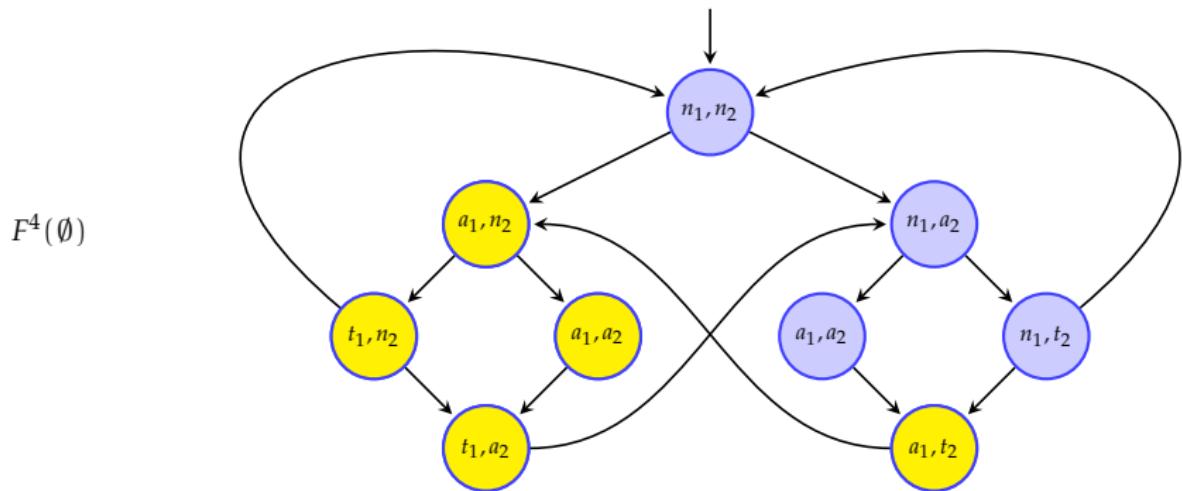
Un train en attente fini toujours par traverser le tunnel

$$\begin{aligned}\text{SAT}(\text{AG}(a_1 \Rightarrow AFt_1)) &= \forall X. (\text{SAT}(\neg a_1) \cup \text{SAT}(AFt_1)) \cap \text{pre}_\forall(X) \\ \text{SAT}(AFt_1) &= \mu X. \text{SAT}(t_1) \cup \text{pre}_\forall(X) = F^9(\emptyset)\end{aligned}$$



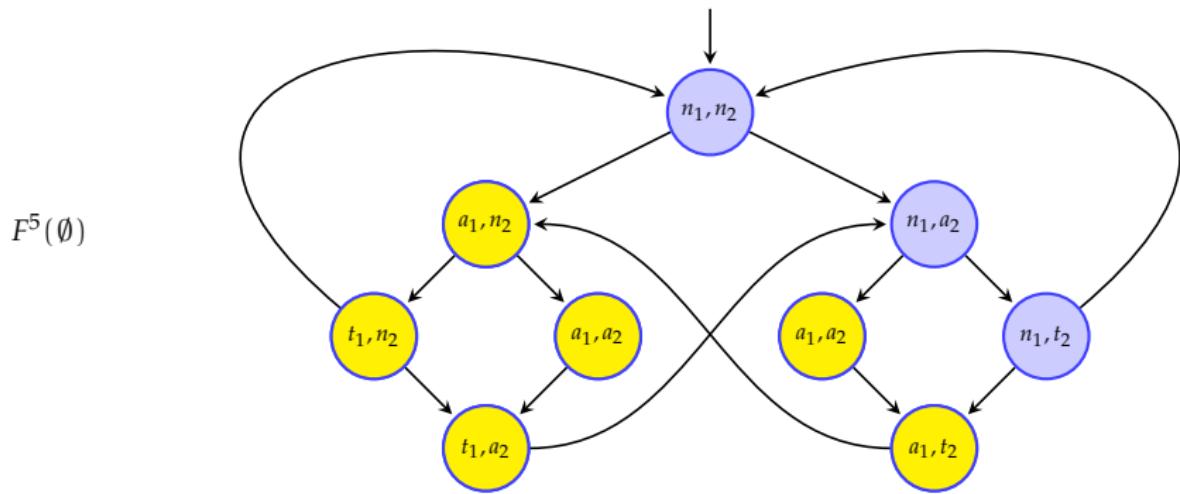
Un train en attente fini toujours par traverser le tunnel

$$\begin{aligned}\text{SAT}(\text{AG}(a_1 \Rightarrow AFt_1)) &= \forall X. (\text{SAT}(\neg a_1) \cup \text{SAT}(AFt_1)) \cap \text{pre}_\forall(X) \\ \text{SAT}(AFt_1) &= \mu X. \text{SAT}(t_1) \cup \text{pre}_\forall(X) = F^9(\emptyset)\end{aligned}$$



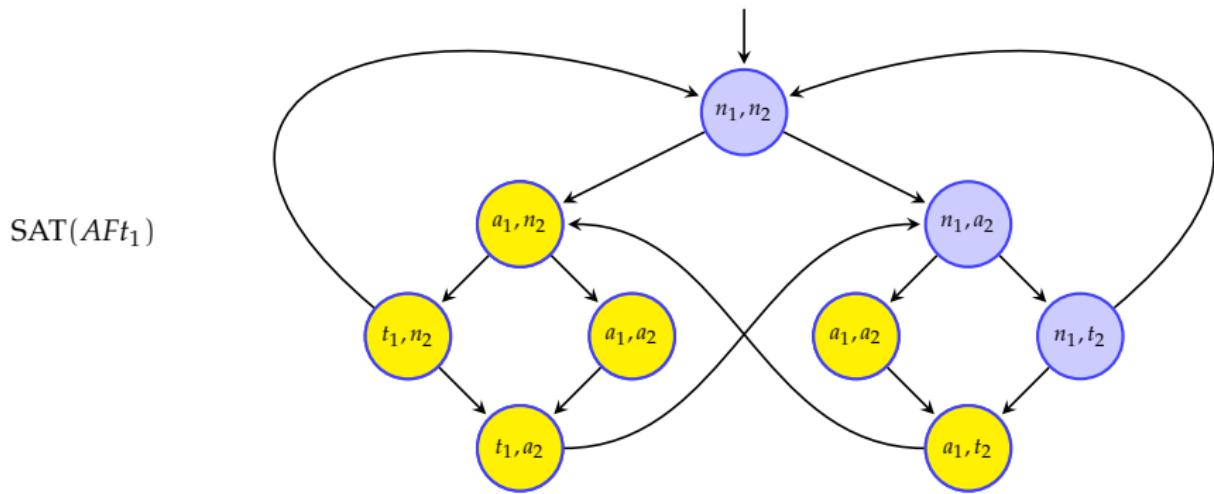
Un train en attente fini toujours par traverser le tunnel

$$\begin{aligned}\text{SAT}(\text{AG}(a_1 \Rightarrow AFt_1)) &= \forall X. (\text{SAT}(\neg a_1) \cup \text{SAT}(AFt_1)) \cap \text{pre}_\forall(X) \\ \text{SAT}(AFt_1) &= \mu X. \text{SAT}(t_1) \cup \text{pre}_\forall(X) = F^9(\emptyset)\end{aligned}$$



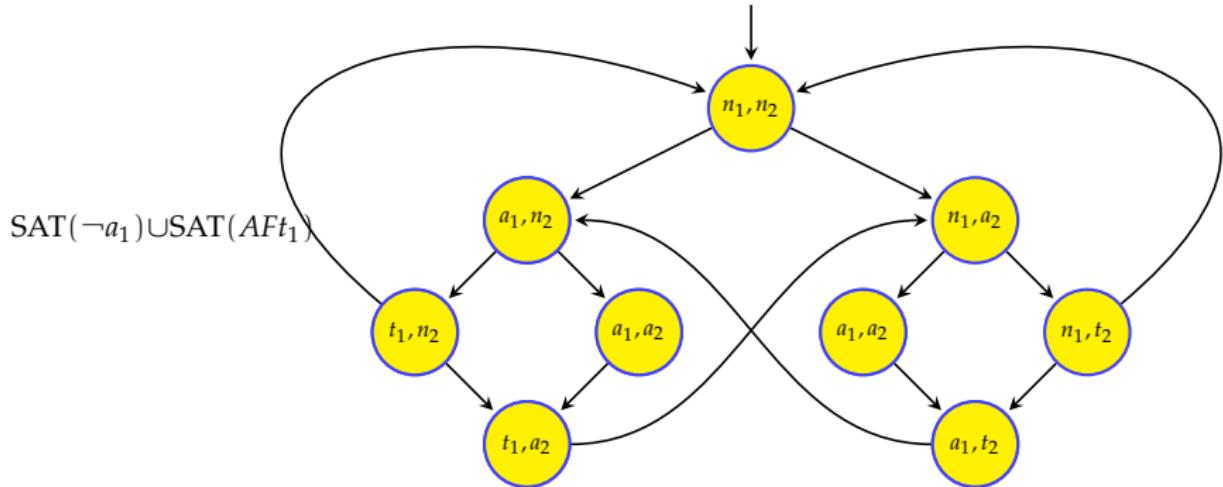
Un train en attente fini toujours par traverser le tunnel

$$\begin{aligned}\text{SAT}(\text{AG}(a_1 \Rightarrow AFt_1)) &= \forall X. (\text{SAT}(\neg a_1) \cup \text{SAT}(AFt_1)) \cap \text{pre}_\forall(X) \\ \text{SAT}(AFt_1) &= \mu X. \text{SAT}(t_1) \cup \text{pre}_\forall(X) = F^9(\emptyset)\end{aligned}$$



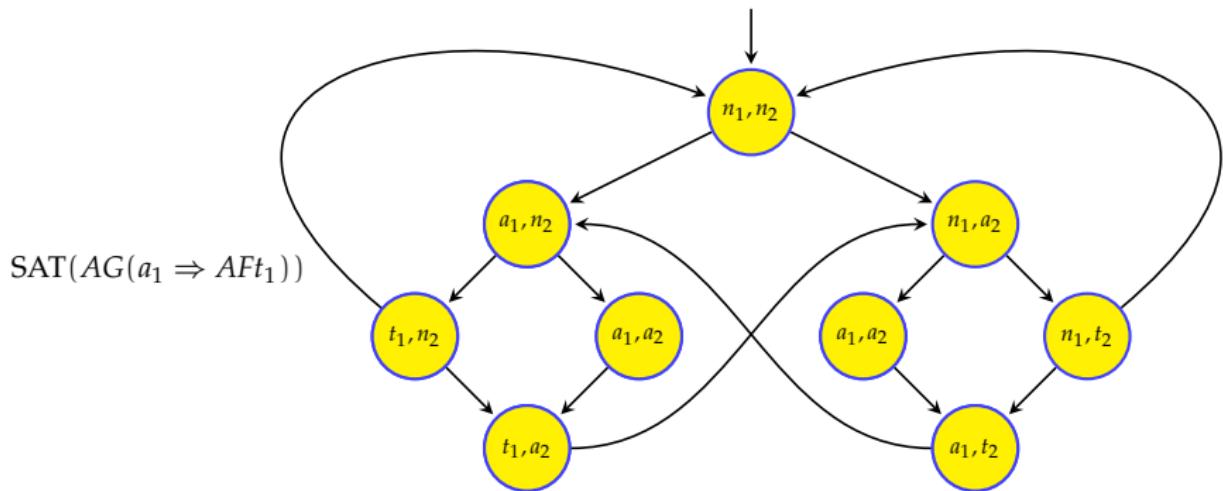
Un train en attente fini toujours par traverser le tunnel

$$\begin{aligned}\text{SAT}(\text{AG}(a_1 \Rightarrow AFt_1)) &= \forall X. (\text{SAT}(\neg a_1) \cup \text{SAT}(AFt_1)) \cap \text{pre}_\forall(X) \\ \text{SAT}(AFt_1) &= \mu X. \text{SAT}(t_1) \cup \text{pre}_\forall(X) = F^9(\emptyset)\end{aligned}$$



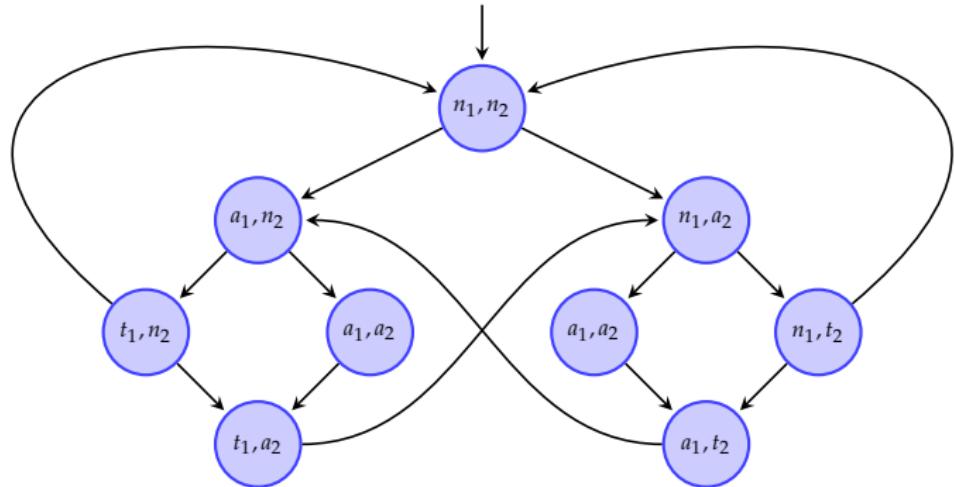
Un train en attente fini toujours par traverser le tunnel

$$\begin{aligned}\text{SAT}(\text{AG}(a_1 \Rightarrow AFt_1)) &= \forall X. (\text{SAT}(\neg a_1) \cup \text{SAT}(AFt_1)) \cap \text{pre}_\forall(X) \\ \text{SAT}(AFt_1) &= \mu X. \text{SAT}(t_1) \cup \text{pre}_\forall(X) = F^9(\emptyset)\end{aligned}$$



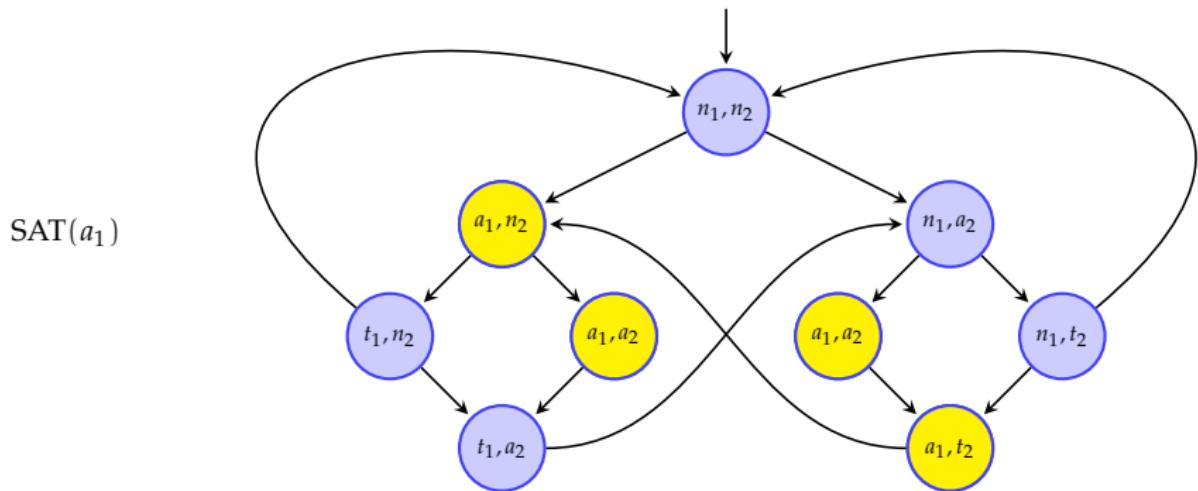
Un train qui est sorti peut toujours se mettre en attente

$$\begin{aligned}\text{SAT}(\text{AG}(n_1 \Rightarrow \text{EX}a_1)) &= \forall X. (\text{SAT}(\neg n_1) \cup \text{SAT}(\text{EX}a_1)) \cap \text{pre}_\forall(X) \\ \text{SAT}(\text{EX}a_1) &= \text{pre}_\exists(\text{SAT}(a_1))\end{aligned}$$



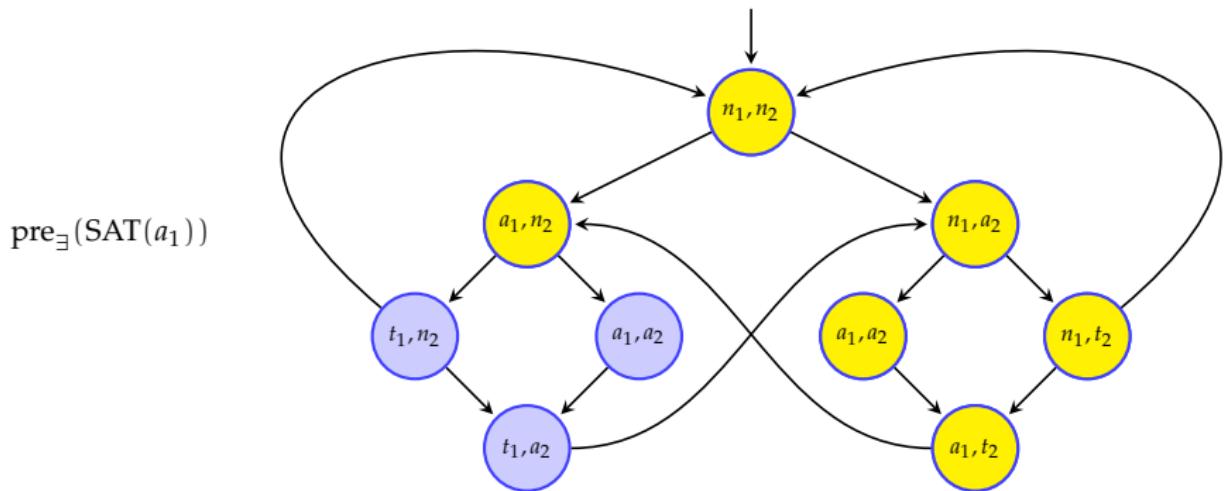
Un train qui est sorti peut toujours se mettre en attente

$$\begin{aligned}\text{SAT}(\text{AG}(n_1 \Rightarrow \text{EX}a_1)) &= \forall X. (\text{SAT}(\neg n_1) \cup \text{SAT}(\text{EX}a_1)) \cap \text{pre}_\forall(X) \\ \text{SAT}(\text{EX}a_1) &= \text{pre}_\exists(\text{SAT}(a_1))\end{aligned}$$



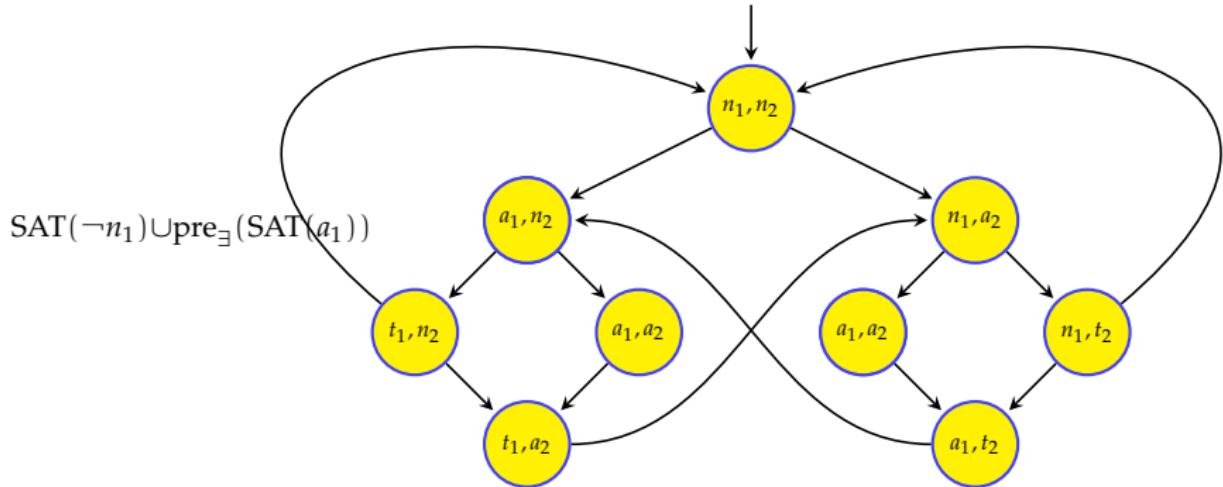
Un train qui est sorti peut toujours se mettre en attente

$$\begin{aligned}\text{SAT}(\text{AG}(n_1 \Rightarrow \text{EX}a_1)) &= \forall X. (\text{SAT}(\neg n_1) \cup \text{SAT}(\text{EX}a_1)) \cap \text{pre}_\forall(X) \\ \text{SAT}(\text{EX}a_1) &= \text{pre}_\exists(\text{SAT}(a_1))\end{aligned}$$



Un train qui est sorti peut toujours se mettre en attente

$$\begin{aligned}\text{SAT}(\text{AG}(n_1 \Rightarrow \text{EX}a_1)) &= \forall X. (\text{SAT}(\neg n_1) \cup \text{SAT}(\text{EX}a_1)) \cap \text{pre}_\forall(X) \\ \text{SAT}(\text{EX}a_1) &= \text{pre}_\exists(\text{SAT}(a_1))\end{aligned}$$



Un train qui est sorti peut toujours se mettre en attente

$$\begin{aligned}\text{SAT}(\text{AG}(n_1 \Rightarrow \text{EX}a_1)) &= \forall X. (\text{SAT}(\neg n_1) \cup \text{SAT}(\text{EX}a_1)) \cap \text{pre}_\forall(X) \\ \text{SAT}(\text{EX}a_1) &= \text{pre}_\exists(\text{SAT}(a_1))\end{aligned}$$

