Cours 11

Vérification de modèle

(*Model checking*)

Motivations

But : Vérifier qu'un modèle vérifie une certaine propriété.

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

Nous nous restreignons dans ce cours aux formules CTL.

CTL : syntaxe (rappel)

$\mathbb{V}$ 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 $\mathbb{V} \cup C \cup T \cup \{(,)\}$ défini inductivement par :

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

Sémantique de CTL

Les formules CTL on une sémantique en tant que formule CTL*, mais nous pouvons donner une sémantique (équivalente) plus directe.

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

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

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

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

$$\begin{aligned}
\mathcal{K}, s \models p &\quad \text{ssi} \quad p \in L(s) \\
\mathcal{K}, s \models \neg\varphi &\quad \text{ssi} \quad \mathcal{K}, \pi, i \not\models \varphi \\
\mathcal{K}, s \models (\varphi_1 \wedge \varphi_2) &\quad \text{ssi} \quad \mathcal{K}, s \models \varphi_1 \text{ et } \mathcal{K}, s \models \varphi_2 \\
&\quad \ldots
\end{aligned}$$

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

$\mathcal{K}, s \models A(\varphi_1 U \varphi_2)$ ssi pour tout chemin π tel que $\pi(0) = s$,
il existe j tel que $\mathcal{K}, \pi(j) \models \varphi_2$ et
pour tout k tel que $k < j$, $\mathcal{K}, \pi(k) \models \varphi_1$

$\mathcal{K}, s \models E(\varphi_1 U \varphi_2)$ ssi il existe un chemin π tel que $\pi(0) = s$,
et il existe j tel que $\mathcal{K}, \pi(j) \models \varphi_2$ et
pour tout k tel que $k < j$, $\mathcal{K}, \pi(k) \models \varphi_1$

Vérification de modèle

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

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

SAT[φ] est défini par induction sur φ.

SAT

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

avec pour tout $X \subseteq S$,

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

SAT

$$
\begin{aligned}
\mathrm{SAT}[AG\varphi] &= \mu X.\, \mathrm{SAT}(\varphi) \cap \mathrm{pre}_{\forall}(X) \\
\mathrm{SAT}[EG\varphi] &= \nu X.\, \mathrm{SAT}(\varphi) \cap \mathrm{pre}_{\exists}(X) \\
\mathrm{SAT}[AF\varphi] &= \nu X.\, \mathrm{SAT}(\varphi) \cup \mathrm{pre}_{\forall}(X) \\
\mathrm{SAT}[EF\varphi] &= \mu X.\, \mathrm{SAT}(\varphi) \cup \mathrm{pre}_{\exists}(X) \\
\mathrm{SAT}[A(\varphi_1 U \varphi_2)] &= \nu X.\, \mathrm{SAT}(\varphi_2) \cup (\mathrm{SAT}(\varphi_1) \cap \mathrm{pre}_{\forall}(X)) \\
\mathrm{SAT}[E(\varphi_1 U \varphi_2)] &= \mu X.\, \mathrm{SAT}(\varphi_2) \cup (\mathrm{SAT}(\varphi_1) \cap \mathrm{pre}_{\exists}(X))
\end{aligned}
$$

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

$$
\begin{aligned}
\mu X.\, F(X) &: \text{le plus petit point fixe de } F \\
\nu X.\, F(X) &: \text{le plus grand point fixe de } F
\end{aligned}
$$

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

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 $F \in \mathcal{P}(S) \to \mathcal{P}(S)$ 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* $F \in \mathcal{P}(S) \to \mathcal{P}(S)$ *une fonction monotone, les fonctions suivantes sont monotones :*

- $\text{pre}_\forall$
- $\text{pre}_\exists$
- $X \mapsto Y \cup F(X)$
- $X \mapsto Y \cap F(X)$

Corollaire

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

Correction de SAT

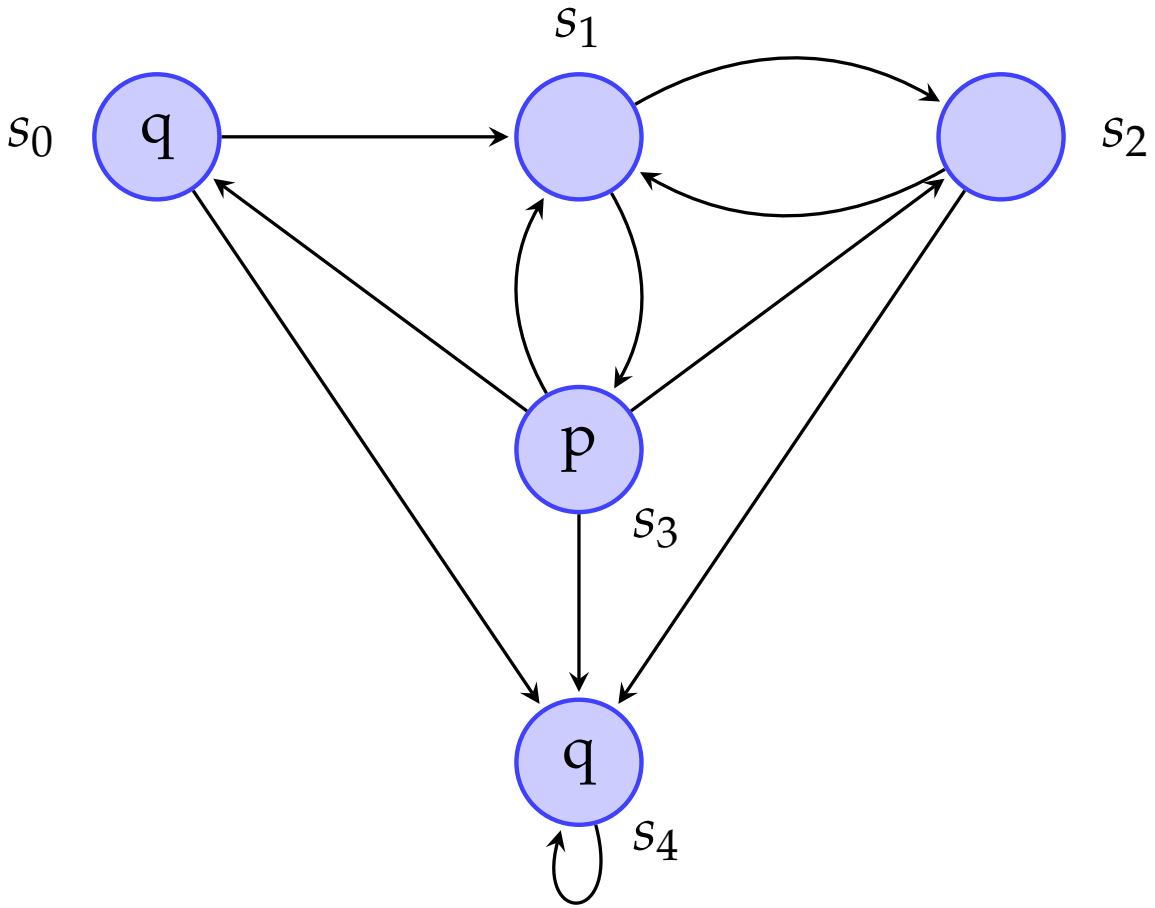
Théorème

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

Exercices

Calculer SAT(EFp).

Calculer SAT(EGq).



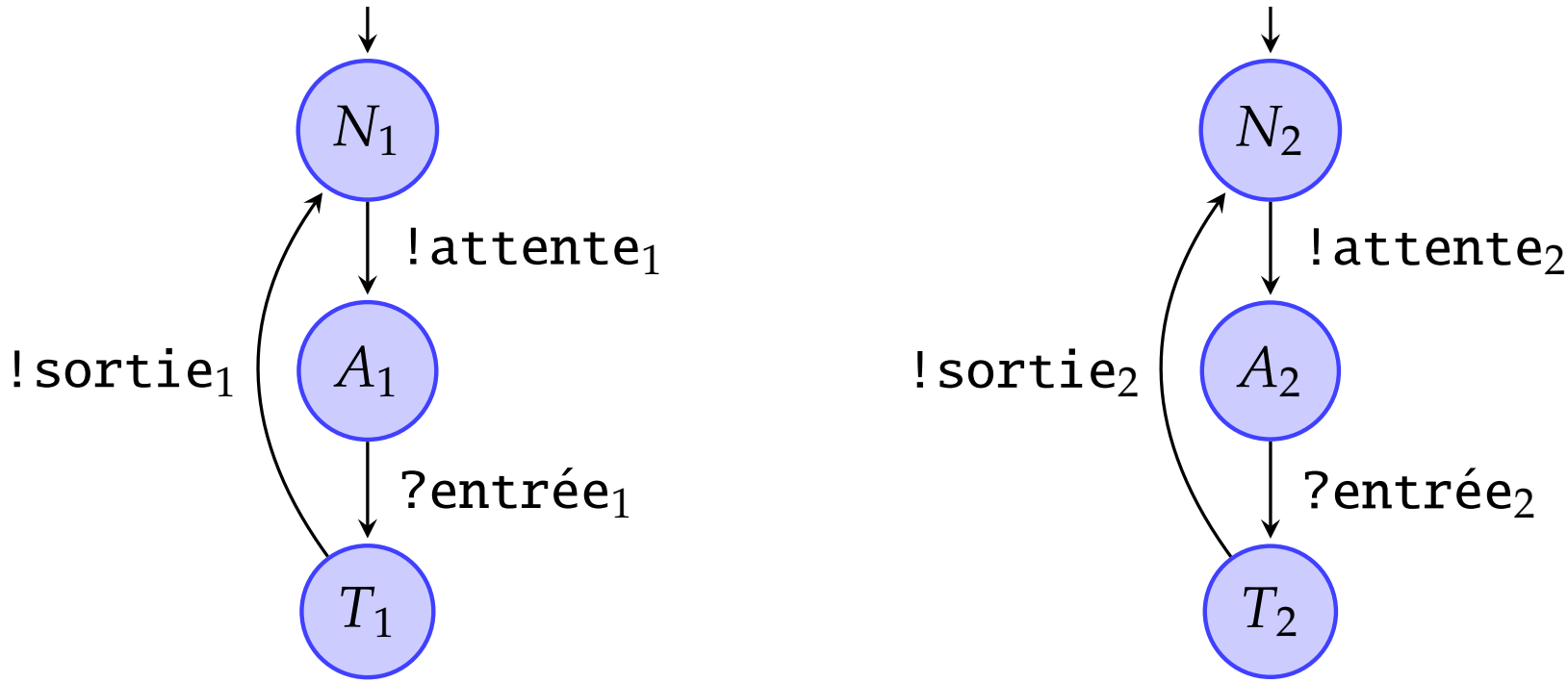
Cas d'étude : problème d'exclusion mutuelle

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 :

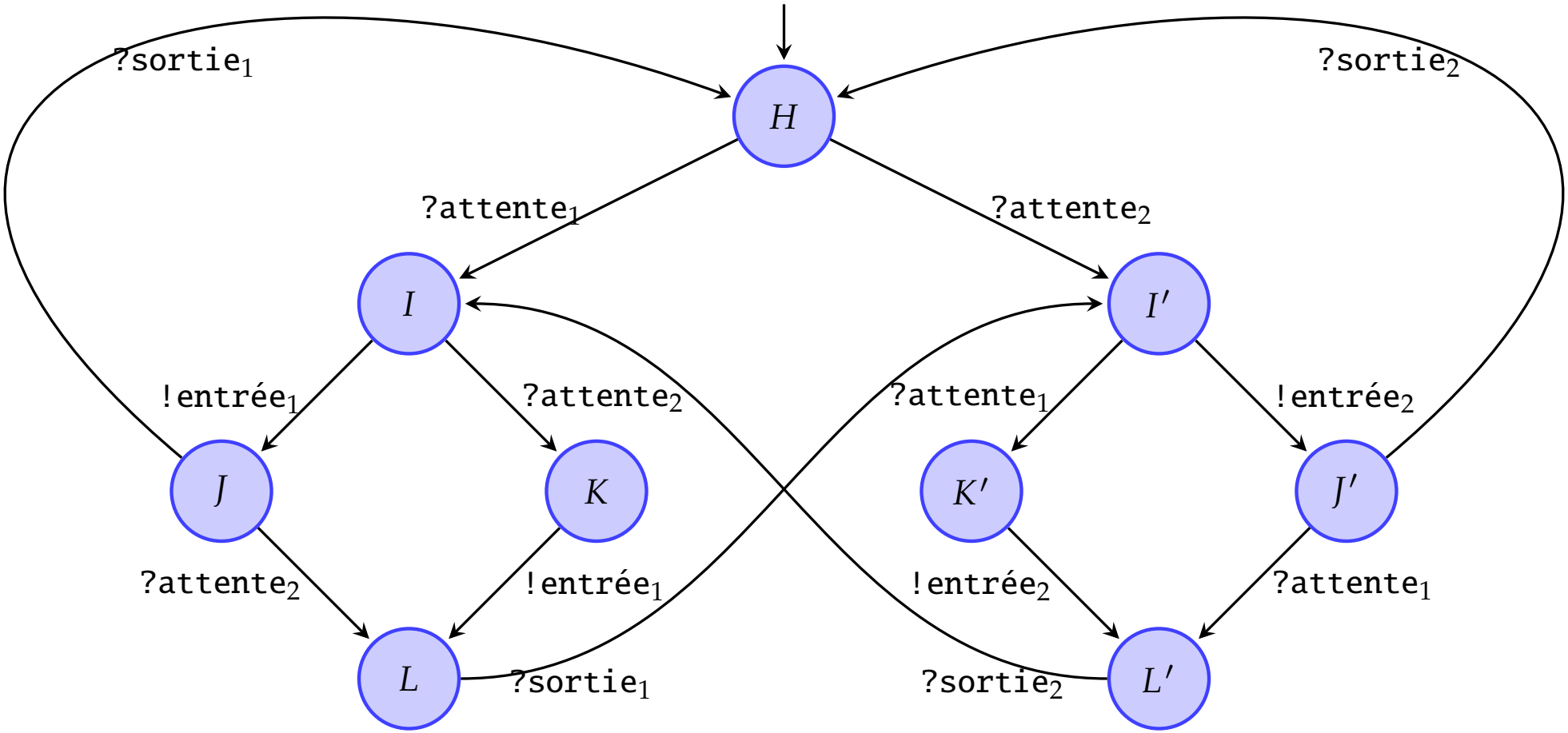
- `attente` : le train veut traverser le tunnel et attend une autorisation,
- `entrée` : le train obtient l'autorisation d'entrer dans le tunnel,
- `sortie` : le train sort du tunnel.

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

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

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

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

Pour une *synchronisation par messages*

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

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 sortie peut toujours se mettre en attente (non-blockage).
- Les trains ne traversent pas forcément le tunnel à tour de rôle.

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