Vérification algorithmique: généralités et rappels

Étapes de la vérification

- Étape 1: On modélise le système M.
- Étape 2: On spécifie la propriété à vérifier ϕ.
- Étape 3: On utilise un algorithme / une méthode pour vérifier si $M \models \phi$.

Étapes:

1. M: peut par exemple se décrire par une structure de Kripke
2. ϕ: peut par exemple se décrire par une formule de logique temporelle
3. $M \models \phi$?: Méthode/algorithme.

Systèmes de transitions

- Un *système de transitions* est $M = (S, S_0, R)$
  (noté (Q, I, δ) dans le cours de Stephan Merz)
  où
  1. S est un ensemble (fini ou infini) d'états,
  2. $S_0 \subseteq S$ est l'ensemble des états initiaux,
  3. $R \subseteq S \times S$ est une relation de transition totale.
     i.e. pour tout s, il existe t tel que $R(s, t)$.
- *Un chemin (une trajectoire) de M débutant en s* est une suite $\pi = s_0, s_1, \ldots$ telle que $s_0 = s$ et $R(s_i, s_{i+1})$ pour tout $i \geq 0$.
- Le chemin est *initialisé* si $s \in S_0$.

Structures de Kripke

Soit AP une ensemble de propositions atomiques.

- Une *structure de Kripke* sur AP est $M = (S, S_0, R, L)$ (noté $(Q, I, \delta, \nu, \lambda)$ par S. Merz, ν correspondant à AP) où
  1. (S, S_0, R) est un système de transitions,
  2. $L : S \longrightarrow 2^{AP}$ est une fonction qui étiquette chaque état $s \in S$ par l'ensemble $L(s)$ des propositions atomiques vraies en s.
- Un chemin π de M est un chemin du système de transitions associé.
- La *trace σ de π* est le ω-mot $L(\pi) = L(s_0)L(s_1)\ldots$ sur l'alphabet $\Sigma = 2^{AP}$.

Exemple. Processus Simple

- Processus qui alterne entre NC (je veux rentrer en session critique) et SC (je suis en session critique).
- Etat intial: NC
- Structure de Kripke sur $AP = \{crit\}$
  - $S = \{NC, SC\}$
  - $S_0 = \{NC\}$
  - $R = \{(NC, SC), (SC, NC)\}$
  - $L(NC) = \emptyset$, $L(SC) = \{crit\}$
- Chemin $\pi = (NCSC)^*$.

Exemple. Programme Simple

- Programme:

$$while\ (true)\ do\ x := (x + y)\ mod\ 2.$$

- Etat initial: $x = y = 1$.
- Structure de Kripke:
  - $S = \{0, 1\} \times \{0, 1\}$
  - $S_0 : \{(1, 1)\}$
  - $R = \{((1, 1), (0, 1)), ((0, 1), (1, 1)), ((1, 0), (1, 0)), ((0, 0), (0, 0))\}$
  - $L((1, 1)) = \{x = 1, y = 1\}$,
    $L((1, 0)) = \{x = 1, y = 0\} \ldots$
- Chemin $\pi = (1, 1), (0, 1), (1, 1), (0, 1), \ldots$

Exemple. Exclusion mutuelle simple

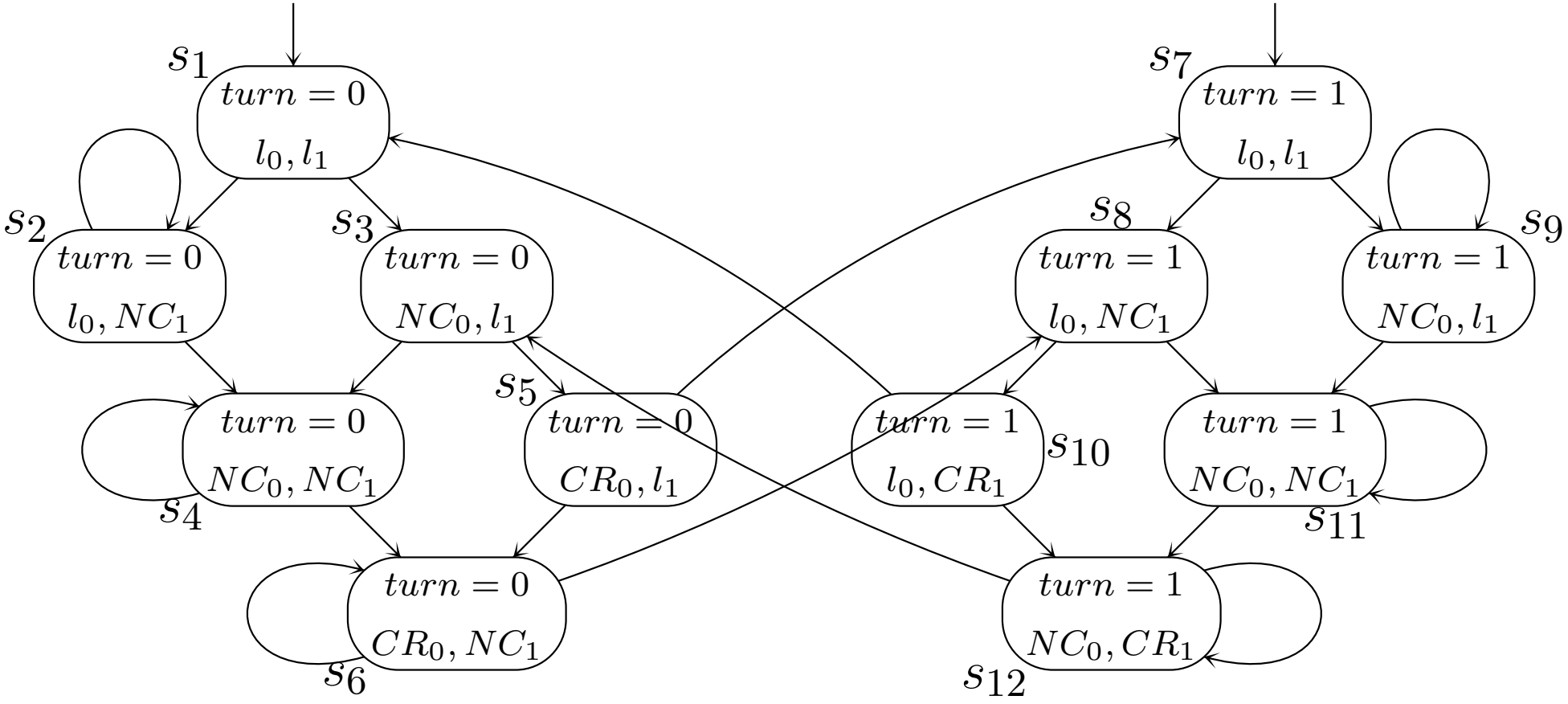
- Algorithme simple d'exclusion mutuelle.

$$P_1 :: \left[\begin{array}{ll} l_0 : & while\ true\ do \\ & \left[\begin{array}{ll} NC_0 : & wait(turn = 0); \\ CR_0 : & turn := 1; \end{array} \right] \end{array} \right]$$

$$\|$$

$$P_2 :: \left[\begin{array}{ll} l_1 : & while\ true\ do \\ & \left[\begin{array}{ll} NC_1 : & wait(turn = 1); \\ CR_1 : & turn := 0; \end{array} \right] \end{array} \right]$$

Partie atteignable



Spécification: logiques temporelles

Les logiques temporelles permettent d'exprimer les propriétés des systèmes que l'on cherche à vérifier.
Logiques temporelles:

- Logique LTL: Logique Temporelle Linéaire.
  - c'est une logique de traces: $\sigma \models \phi$ est définit sur les traces σ.
- Logique CTL: Logique Calcul Arborescente.
  - c'est une logique d'états: $s \models \phi$ est définit sur les états s.
- Logique CTL*:
  - c'est une logique mixte qui contient LTL et CTL.

Écriture des formules

- Symboles:

Clarke/Emerson	Manna/Pnueli
X	$\circ$
U	U
F	$\diamond$
G	$\Box$
A	$\forall$
E	$\exists$

- Exemples:
  - $\forall \Box \neg (CR_0 \wedge CR_1)$
  - $\forall \diamond (CR_0)$

Rappel: LTL

AP: ensemble de propositions atomiques.
C'est une logique de traces: les formules LTL s'interprètent sur les traces d'une structure de Kripke, i.e. sur les ω-mots $\sigma : \sigma_0, \sigma_1, \ldots$ sur l'alphabet $\Sigma = 2^{AP}$.
Grammaire: $p|\neg\phi|\phi \vee \psi|\circ\phi|\phi U\psi|\diamond\phi|\Box\phi$.

$$
\begin{array}{lll}
\sigma \models p & \text{ssi} & p \in \sigma_0 \text{ où } p \in AP \\
\sigma \models \neg\phi & \text{ssi} & \sigma \models \phi \text{ est faux} \\
\sigma \models \phi \vee \psi & \text{ssi} & \sigma \models \phi \text{ ou } \sigma \models \psi \\
\sigma \models \circ\phi & \text{ssi} & \sigma^1 \models \phi \\
\sigma \models \phi U \psi & \text{ssi} & \text{pour } k \geq 0, \sigma^k \models \psi \\
 & & \text{et } \sigma^j \models \phi \text{ pour } 0 \leq j < k
\end{array}
$$

où σ^i est le ω-mot $\sigma_i, \sigma_{i+1}, \ldots$.

On définit aussi:

$$\begin{aligned} \Diamond\phi &\equiv true\, U\phi \\ \Box\phi &\equiv \neg \Diamond \neg\phi \end{aligned}$$

Rappel: CTL

C'est une logique d'états: les formules CTL s'interprètent sur les états d'une structure de Kripke.
Grammaire:
$p|\neg\phi|\phi \vee \psi|\exists \circ \phi|\exists\Box\phi|\exists \diamond \phi|\exists\phi U\psi|\forall \circ \phi|\forall\Box\phi|\forall \diamond \phi|\forall\phi U\psi$

$s \models p$ ssi $p \in L(s)$

$s \models \neg f_1$ ssi $s \not\models f_1$

$s \models f_1 \vee f_2$ ssi $s \models f_1$ ou $s \models f_2$

$s \models \exists \circ g_1$ ssi $\exists$ chemin $\pi = s, s_1, s_2 \ldots$ t.q. $s_1 \models g_1$

$s \models \exists \Box g_1$ ssi $\exists$ chemin $\pi = s, s_1, s_2 \ldots$
t.q. pour tout i, $s_i \models g_1$

$s \models \exists g_1 U g_2$ ssi $\exists$ chemin $\pi = s, s_1, \ldots$ et $\exists k \geq 0$
t.q. $s_k \models g_2$ et $\forall 0 \leq j < k$, $s_j \models g_1$

On définit aussi:

$$
\begin{aligned}
\forall \circ f &= \neg\exists \circ \neg f \\
\exists \diamond f &= \exists[trueU\, f] \\
\forall \Box f &= \neg\exists \diamond \neg f \\
\forall \diamond f &= \neg\exists \Box \neg f \\
\forall [f U g] &= \neg\exists[\neg g U (\neg f \wedge \neg g)] \wedge \neg E \Box \neg g
\end{aligned}
$$

Rappel: Logique temporelle CTL*

C'est une logique mixte: les formules CTL* s'interprètent sur les traces et sur les états.

$M, s \models p$	ssi	$p \in L(s)$
$M, s \models \neg f_1$	ssi	$M, s \not\models f_1$
$M, s \models f_1 \vee f_2$	ssi	$M, s \models f_1$ ou $M, s \models f_2$
$M, s \models f_1 \wedge f_2$	ssi	$M, s \models f_1$ et $M, s \models f_2$
$M, s \models \exists g_1$	ssi	$\exists$ chemin $\pi = s, s_1, \ldots$ t.q. $M, \pi \models g_1$
$M, s \models \forall g_1$	ssi	$\forall$ chemin $\pi = s, s_1, \ldots$ $M, \pi \models g_1$
$\vdots$	$\vdots$	$\vdots$

$\vdots$ $\vdots$ $\vdots$

$M, \pi \models f_1$ ssi $\pi = s, s_1, \ldots$ et $M, s \models f_1$

$M, \pi \models \neg g_1$ ssi $M, \pi \not\models g_1$

$M, \pi \models g_1 \vee g_2$ ssi $M, \pi \models g_1$ ou $M, \pi \models g_2$

$M, \pi \models g_1 \wedge g_2$ ssi $M, \pi \models g_1$ et $M, \pi \models g_2$

$M, \pi \models \circ g_1$ ssi $M, \pi^1 \models g_1$

$M, \pi \models \diamond g_1$ ssi $\exists k \geq 0$ t.q. $M, \pi^k \models g_1$

$M, \pi \models \Box g_1$ ssi $\forall k \geq 0$ $M, \pi^k \models g_1$

$M, \pi \models g_1 U g_2$ ssi $\exists k \geq 0$ t.q. $M, \pi^k \models g_2$
et $\forall 0 \leq j < k$, $M, \pi^j \models g_1$

Vérification $M \models \phi$?

1. pour CTL:
   - Complexité: $O(|\phi|(|S| + |R|))$.
     ($O(|\phi|(|S| + |R|)f)$ si f contraintes d'équité).
2. pour LTL:
   - Complexité: $O(2^{|\phi|}(|S| + |R|))$.
3. pour CTL*:
   - Complexité: $O(2^{|\phi|}(|S| + |R|))$.