

Génie Logiciel M2Pro-GL

Methodes Formelles pour Protocoles et Architectures

Burkhart Wolff

(basée sur le materiel
de Fatiha Zaïdi & Marie-Claude Gaudel)

wolff@lri.fr

Plan du Cours

- ◆ Préliminaires sur les modèles du parallélisme
- ◆ Les langages de spécification, généralités sur le langage LOTOS
- ◆ Lotos de base
- ◆ Lotos complet
- ◆ Test de Protocoles
- ◆ Model-Checking

Chap.1 : Sémantique du parallélisme

- ◆ Bibliographie : “*Vérification de logiciels : techniques et outils du model-checking*”, chap. 1 et 2, ouvrage collectif, Vuibert Informatique, Paris, 1999
- ◆ Ensemble de *processus* qui :
 - exécutent des *actions* indépendamment les uns des autres, donc changent d'*état*
 - coopèrent ; donc se *synchronisent* avec des actions d'autres processus
 - indépendance => *non-déterminisme*

1 - Variété des systèmes => variété des modèles

- ◆ *Deux grands mécanismes de coopération :*
 - ◆ - messages (réseaux)
 - ◆ - variables partagées (machines parallèles ou clients-serveurs)

- ◆ *Deux grands types de systèmes :*
 - ◆ - synchronisés (=> horloge commune)
 - ◆ - asynchrones (notions locales de temps ≠)

Difficultés dues à la répartition

- ◆ Difficulté à percevoir l'**état global** (observer les autres prend du temps, et pendant ce temps-là l'état global évolue...)
- ◆ Nécessité de **tolérer les fautes** de transmissions (messages perdus, processeurs ou serveurs défectueux)
- ◆ Possibilité de **blocage**, spécialement pour les systèmes asynchrones
- ◆ non abordé dans ce cours par manque de temps : performances, goulots d'étranglement

Exécutions, comportements

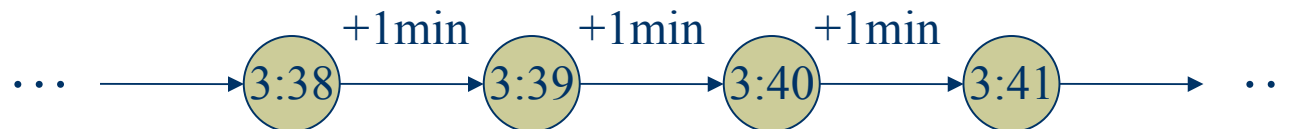
- ◆ Localement, dynamiquement : **traces**, i.e. suite des actions, parfois infinie, exécutées par un processus
- ◆ Localement, statiquement : **arbre des traces** possibles d'un processus
- ◆ Globalement : dépend du modèle de parallélisme considéré. Dans ce cours, **entrelacement** des traces

Modèle d'entrelacement

- ◆ Les actions sont *atomiques* et *non superposables*
- ◆ Un comportement de plusieurs processus parallèles indépendants est UNE trace où sont entrelacées les actions des processus
- ◆ L'ensemble des comportements possibles est l'ensemble des entrelacements (*non déterminisme*)
- ◆ Modèle simple, et suffisant pour décrire et vérifier la plupart des protocoles. Mais ce n'est qu'un modèle...

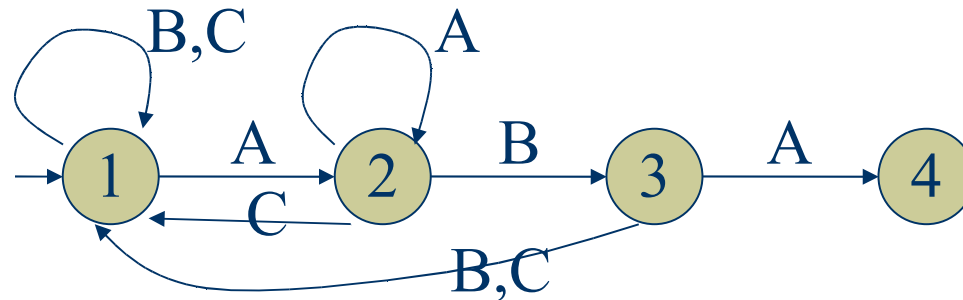
2 - Rappels sur les automates

- ◆ appelés aussi systèmes de transitions ou structures de Kripke
- ◆ États, qui ont un nom, et transitions, qui sont étiquetées
- ◆ Exemple : une montre, $24 \times 60 = 1440$ états, 1440 transitions



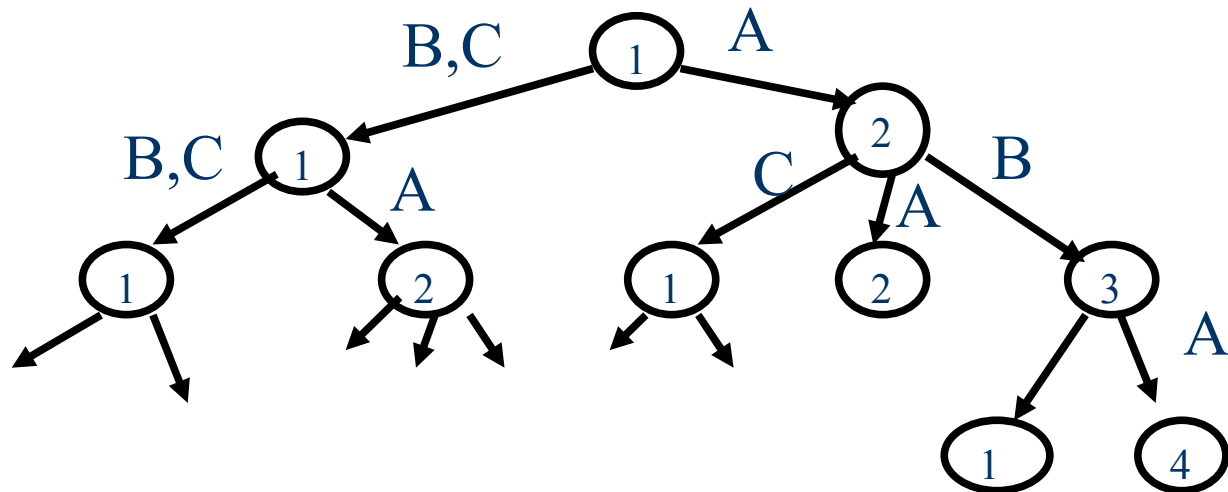
Exemple introductif

- ◆ Un digicode avec 3 touches ABC qui ouvre la porte quand on a tapé ABA



- ◆ 4 états, 9 transitions \Rightarrow automate fini
- ◆ exécutions : 1121, 112223, ...
- ◆ comportements 12234, 11231234, ...
- ◆ il y a des exécutions infinies

Début de l'arbre des exécutions



Note : les sous-arbres de racine 1 (resp. 2, 3) sont toujours les mêmes car cet arbre est le développement d'un automate

Propriétés associées aux états de l'automate

- ◆ “*la porte est ouverte*” est vraie dans l'état 4, fausse dans les autres états (*spécification*)
- ◆ Propriétés structurelles
 - “*on vient de taper A*” (P_A) est vraie en 2 et 4, et fausse ailleurs,
 - “*on vient de taper B*” (P_B) est toujours vraie en 3
 - “*l'état précédent est forcément 2*” (P_2) est vraie en 3 seulement
 - “*l'état précédent est forcément 3*” (P_3) est vraie en 4 seulement, ...

Propriétés de l'automate

- ◆ *“Si la porte s’ouvre, c’est que les trois dernières lettres tapées sont A, B, A”*
 - On ne peut atteindre l’état 4 qu’après 3 transitions étiquetées successivement par A, B, A
- ◆ *“Toute suite de lettres tapées finissant par ABA ouvre la porte”*
 - Quelque soit l’état de départ, après 3 transitions étiquetées successivement par A, B, A on est dans l’état 4
- ◆ Se démontrent à partir des propriétés associées aux états

Définitions de base sur les automates

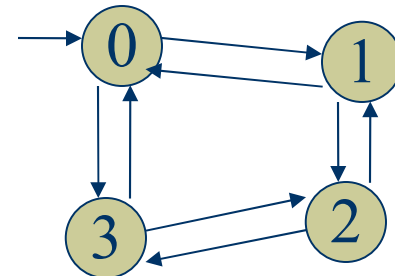
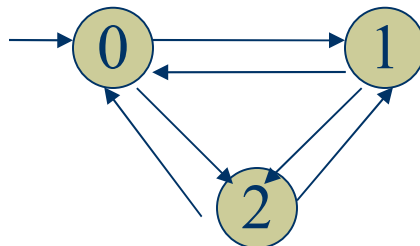
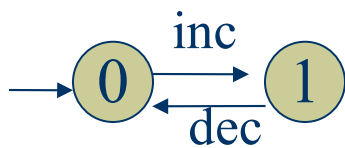
- ◆ **Automate** : $\mathcal{A} = \langle Q, E, T, q_0, \ell \rangle$
 - Q ensemble des états
 - E ensemble des étiquettes des transitions
 - $T \subseteq Q \times E \times Q$ ensemble des transitions
 - $q_0 \in Q$, état initial
 - ℓ application de Q dans $\text{Prop} = \{P_1, \dots\}$
- ◆ **Digicode** :
 - $Q = \{1, 2, 3, 4\}$, $E = \{A, B, C\}$, $T = \{(1, A, 2), (1, B, 1), (1, C, 1), \text{etc}\}$, $q_0 = 1$,
 - $\ell = (1 \rightarrow \neg\text{ouvert}, 2 \rightarrow \neg\text{ouvert}, 3 \rightarrow \neg\text{ouvert}, 4 \rightarrow \text{ouvert})$

Chemins et exécutions

- ◆ *chemin d'un automate*
 - $\sigma = \dots (q_i, e_i, q'_i) \dots$ avec $q'_i = q_{i+1}$
 - **Notation** : $q_1 -e_1-> q_2 -e_2-> q_3 -e_3-> \dots$
- ◆ *exécutions*
 - *exécution partielle* : chemin partant de q_0
 - *comportement* : exécution partielle maximale (qu'on ne peut pas prolonger)
- ◆ Dans certains cas on distingue un sous-ensemble des *états* « *finaux* », Q_f , sans successeurs. Un comportement qui atteint un état final « *termine avec succès* »
- ◆ *états atteignables* : ils apparaissent au moins dans une exécution

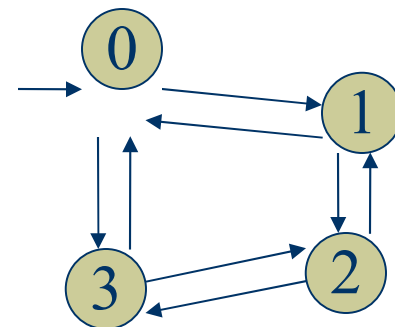
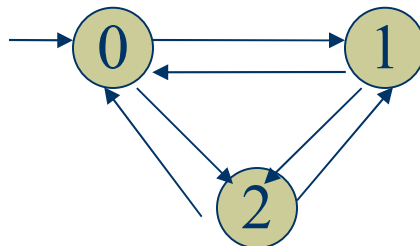
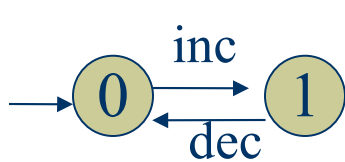
3 - Compositions d'automates : produits

- ◆ $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \dots \times \mathcal{A}_n$
 - **Etats** : vecteurs d'états des automate composants
 - **Transitions** : vecteurs de transitions t_1, \dots, t_n tel que
 - chaque t_i est soit la transition “-”, soit une transition de \mathcal{A}_i
 - il y a au moins un t_i qui n'est pas “-”
- ◆ Exemple : système de 3 compteurs indépendants, un modulo 2, un modulo 3, un modulo 4



Exemple : 3 compteurs indépendants

- ◆ Les états sont les 24 vecteurs :
 - $(0, 0, 0) \dots (1, 2, 3)$
- ◆ Les transitions sont étiquetées par un des 26 vecteurs:
 - $(\text{inc}, -, -)$, $(\text{dec}, -, -)$, $(-, \text{inc}, \text{dec})$, etc
 - NB : ce n'est pas de l'entrelacement mais des multi-ensembles d'actions
- ◆ On a $24 \times 26 = 624$ transitions!



Quelques transitions à partir de l'état $(0, 0, 0)$

Produit \Rightarrow explosion de la taille de l'automate

Définition formelle du produit

- ◆ $\mathcal{A}_i = \langle Q_i, E_i, T_i, q_{0,i}, \ell_i \rangle$ avec $i = 1, \dots, n$
- ◆ $Q = Q_1 \times \dots \times Q_n$
- ◆ $E = \prod_{1 \leq i \leq n} (E_i \cup \{-\})$
- ◆ $T = \{((q_1, \dots, q_n), (e_1, \dots, e_n), (q'_1, \dots, q'_n)) \mid \text{pour tout } i, \text{ soit } e_i = \text{“-”} \text{ et } q'_i = q_i, \text{ soit } (q_i, e_i, q'_i) \in T_i\}$
- ◆ $q_0 = (q_{0,1}, \dots, q_{0,n})$
- ◆ $\ell((q_1, \dots, q_n)) = \bigcup_{1 \leq i \leq n} \ell_i(q_i)$

Cas particulier de l'entrelacement

- ◆ $\mathcal{A}_i = \langle Q_i, E_i, T_i, q_{0,i}, \ell_i \rangle$ avec $i = 1, \dots, n$
- ◆ $Q = Q_1 \times \dots \times Q_n$
- ◆ $E = \prod_{1 \leq i \leq n} (E_i \cup \{-\})$
- ◆ $T = \{((q_1, \dots, q_n), (e_1, \dots, e_n), (q'_1, \dots, q'_n)) \mid \text{il existe } i \text{ tel que } (q_i, e_i, q'_i) \in T_i \text{ et pour tout } j \neq i, e_j = \text{“-” et } q'_j = q_j\}$
- ◆ $q_0 = (q_{0,1}, \dots, q_{0,n})$
- ◆ $\ell((q_1, \dots, q_n)) = \cup_{1 \leq i \leq n} \ell_i(q_i)$
- ◆ Notation : $\mathcal{A} = \mathcal{A}_1 \parallel \mathcal{A}_2 \parallel \dots \parallel \mathcal{A}_n$

Autre cas particulier : produit synchronisé

- ◆ Certains automates doivent effectuer certaines actions simultanément
 - Seules certaines combinaisons d'étiquettes sont autorisées
- ◆ Exemple : on synchronise les incréments et les décréments des 3 compteurs
 - $\text{Sync} = \{(inc, inc, inc), (dec, dec, dec)\}$
 - Certains états ne sont plus atteignables
 - cf figure 19 : seule la transition de (000) à (111) demeure

Vue globale du produit synchronisé

La détermination des états atteignables est un problème difficile ☹

Définition formelle du produit synchronisé

- ◆ La même que pour le produit, mais
 - $T_{\text{Synch}} = \{((q_1, \dots, q_n), (e_1, \dots, e_n), (q'_1, \dots, q'_n)) \mid (e_1, \dots, e_n) \in \text{Synch}, \text{ et pour tout } i, \text{ soit } e_i = \text{“-”} \text{ et } q'_i = q_i, \text{ soit } (q_i, e_i, q'_i) \in T_i\}$
 - On se restreint aux états atteignables
- ◆ Notation :
 - $\mathcal{A} = \mathcal{A}_1 \parallel \mathcal{A}_2 \parallel \dots \parallel \mathcal{A}_n$ avec $\text{Synch} = \{\dots\}$

4 - Synchronisation par messages

- ◆ Étiquettes spéciales : !m, ?m
- ◆ Si un n-uplet de Synch contient !m, il contient ?m, et vice versa
- ◆ Intuition (un peu fausse) : envoi de m et attente de réception de m
- ◆ Exemple : verrou d'une porte et contrôleur



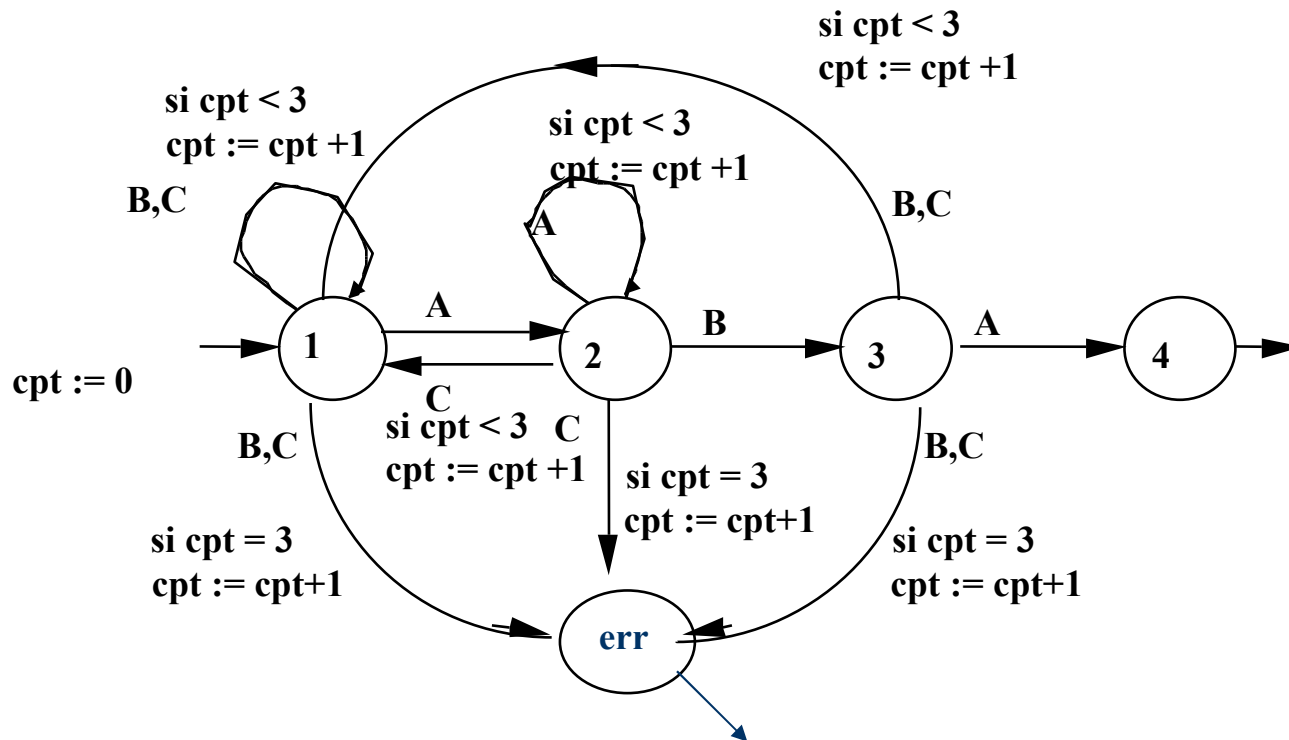
5 - Automates avec variables

- ◆ Notation plus compacte pour les automates complexes : *variables associées à un état*
 - une transition peut modifier la valeur d'une variable
 - une transition peut être conditionnée par la valeur d'une variable
- ◆ Un « état » d'un tel automate est un ensemble d'états sans variable
 - **Exemple** : état q avec $B:\text{Bool} \Leftrightarrow 2$ états, q_vrai et q_faux
- ◆ Même chose pour les « transitions »

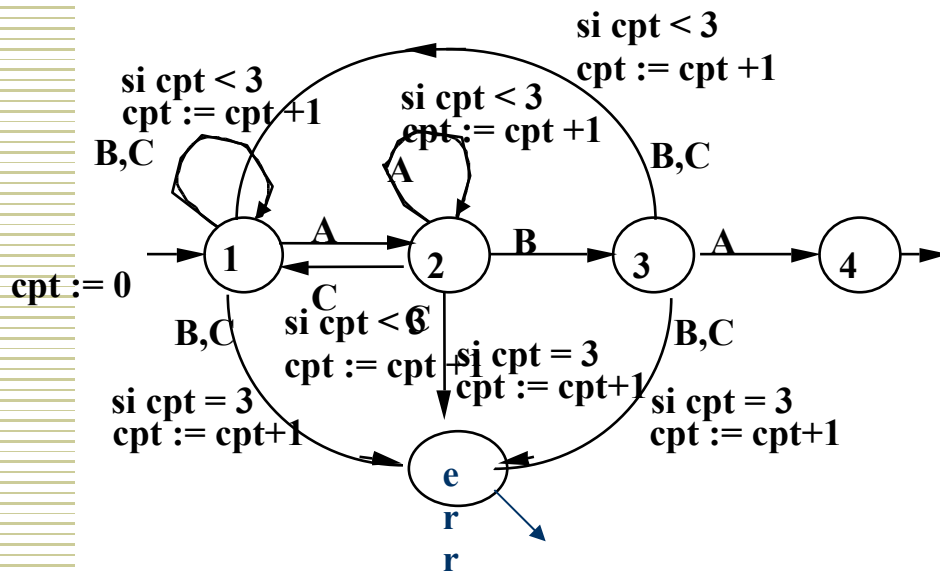
Exemple

- ◆ Digicode qui tolère au maximum 3 erreurs, puis se bloque
- ◆ Une variable « cpt » est incrémentée par les erreurs
- ◆ Si $\text{cpt} < 3$, on effectue les transitions du digicode d'origine
- ◆ Si $\text{cpt} = 3$ on va dans un état « err »

L'automate avec variable



Passage à l'automate sans variable



6 - Synchronisation par variable partagée

- ◆ Une même variable est utilisée par plusieurs automates
- ◆ **exemple** : système composé de 2 utilisateurs A et B qui partagent une imprimante



Produit des deux automates avec « tour » initialisée à A

