Application de la synthèse de contrôleurs à l'ordonnancement de systèmes temps-réel

Karine Altisen

le 17 décembre 2001

Thèse préparée à Verimag

sous la direction de Joseph Sifakis

Systèmes temps-réel

- Systèmes temps-réel critiques (ex : avion, métro)
- Systèmes temps-réel grand public (ex : télécom, multimédia)
  - Produits de grande consommation
  - Contraintes temporelles de type qualité de service
  - Faibles coûts
  - Limitations de ressources (taille, consommation d'énergie)

Ordonnancement temps-réel

Problème

Application temps-réel

n processus P_1, ..., P_n

+ ressources partagées

(processeurs, mémoires, bande passante...)

+ contraintes temporelles

(date d'activation, temps d'exécution, échéances)

Ordonnanceur

Contrôle les accès concurrents des processus aux ressources

Problème de l'ordonnançabilité

Trouver un ordonnanceur qui satisfait les contraintes temporelles

Approches

Approche analytique :

ex : Liu et Layland 1973

Modèle :	n processus périodiques, $P_i = (E_i, T_i)$, un processeur
+ Politique d'ordonnancement :	« Rate Monotonic »
+ Condition d'ordonnançabilité :	$\sum_{i=1}^{n} \frac{E_i}{T_i} \leq n(2^{\frac{1}{n}} - 1)$

- faible complexité ; spécificité !

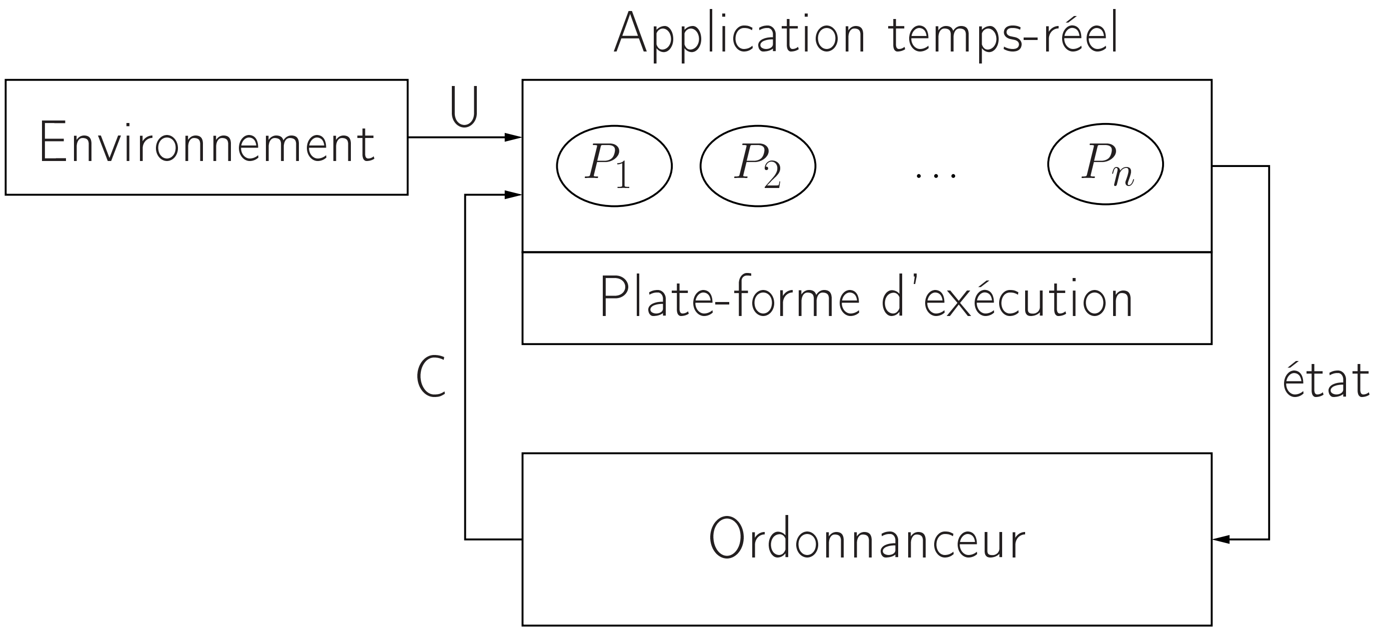
Approche basée sur un modèle :

Modèle opérationnel de l'application + Validation (vérification, synthèse)

- généralité ; complexité !

Notre approche

Paradigme : Synthèse de contrôleur



Notre approche

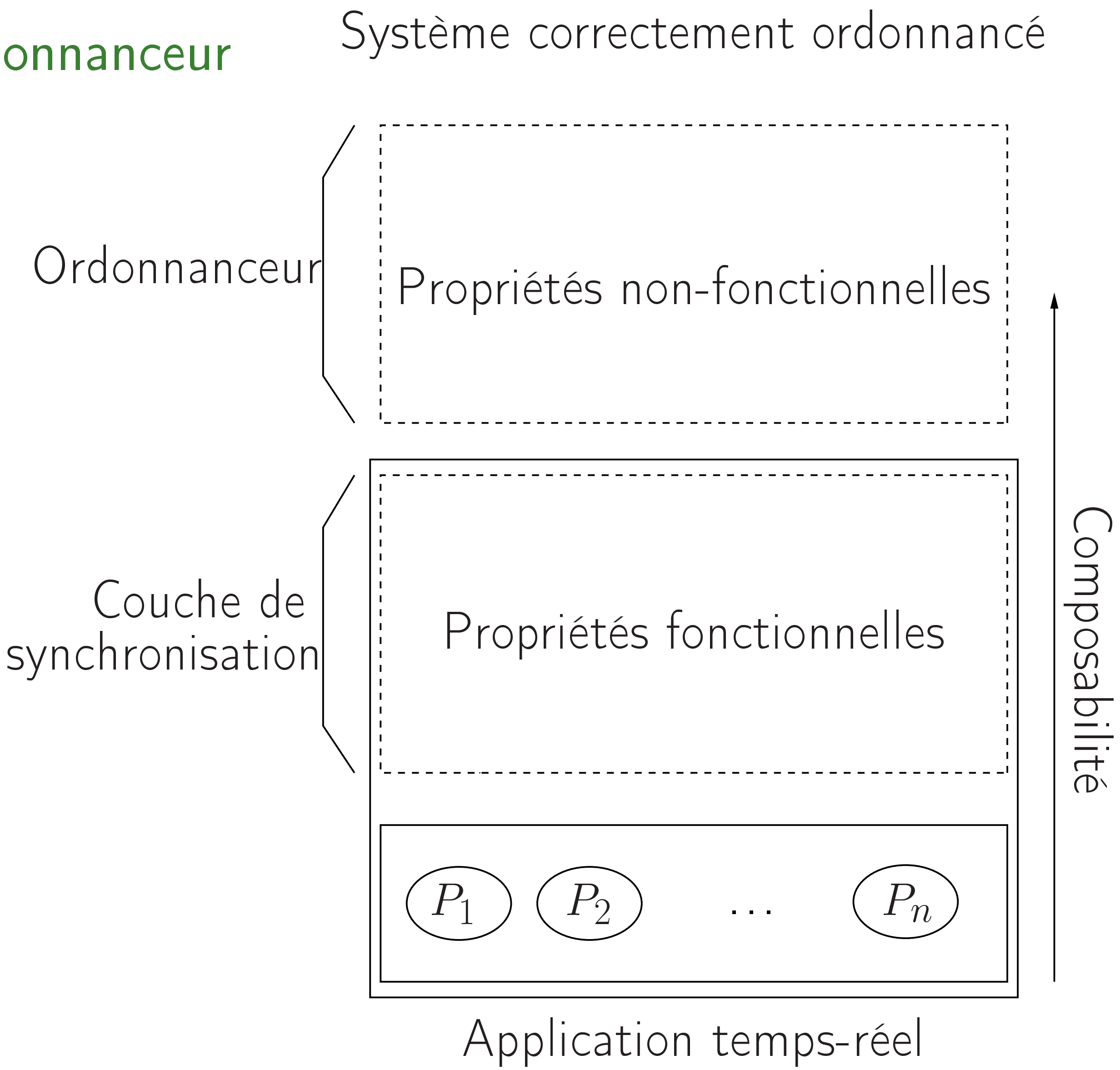
Méthodologie pour le calcul de l'ordonnanceur

itérative et constructive,

autant que possible

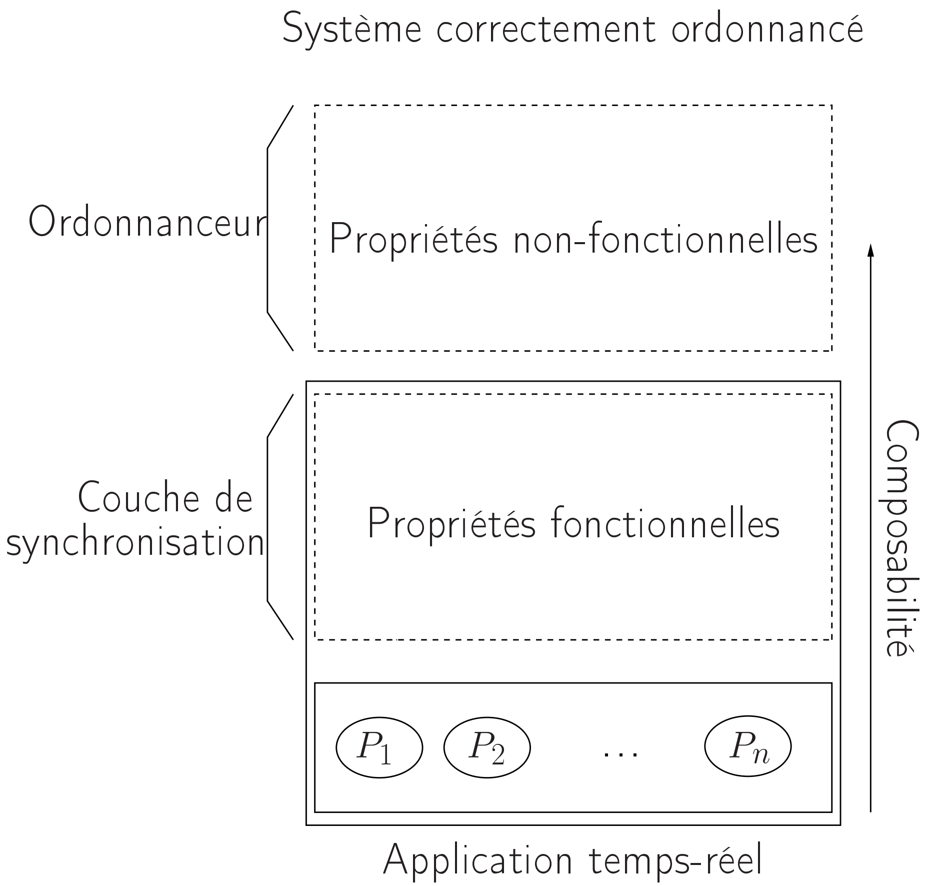
- généralité ;

complexité réduite



Plan

- Introduction
- Modélisation de l'application temps-réel
- Ordonnancement de l'application temps-réel
- Conclusion



Modélisation de l'application en vue de son ordonnancement

Le modèle

Automate temporisé
+ contrôlabilité
+ urgence

$\Bigg\}$ Système temporisé ST

Contrôlabilité

C = allocation de ressources

U = actions externes (env.)

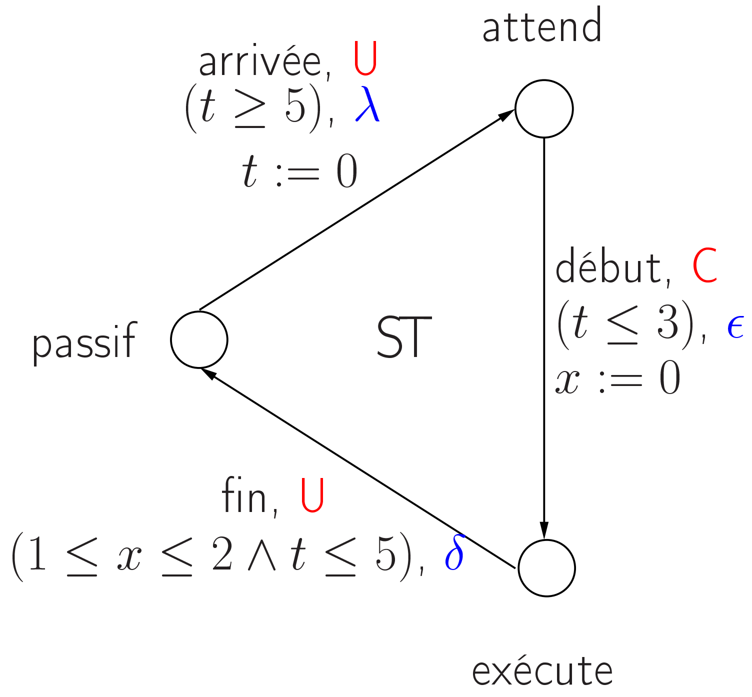
actions internes (plate-forme)

Urgence

ϵ = urgent ;

δ = délayable ;

λ = paresseux ;



ex : un processus sporadique

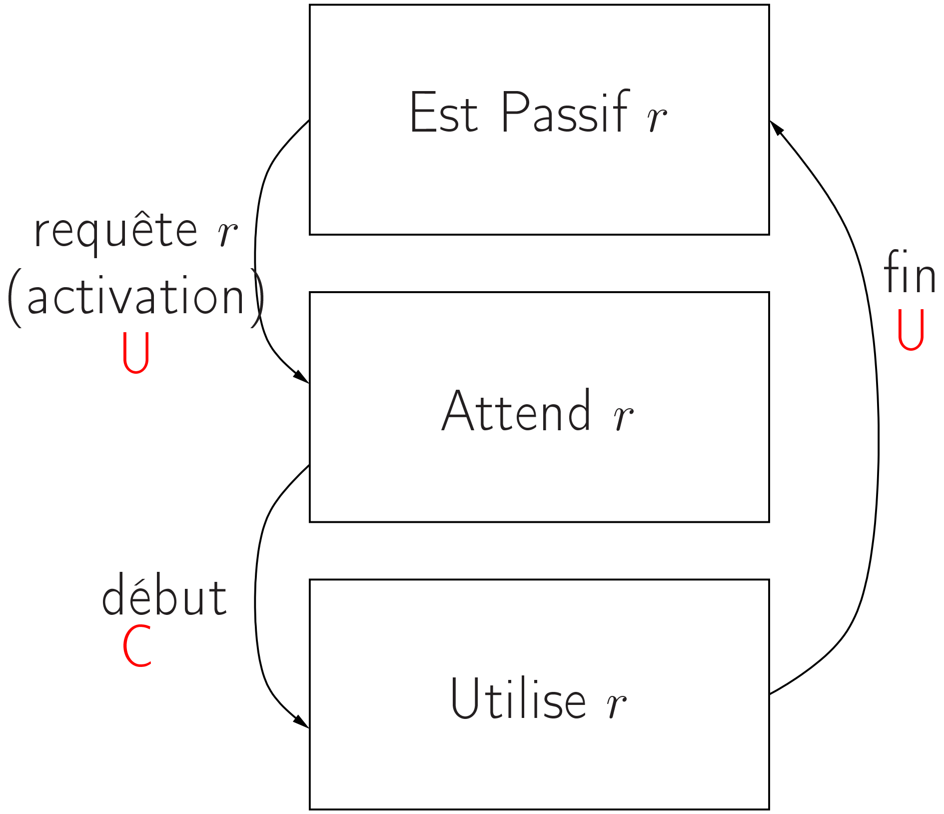
temps min. d'inter-activation 5,

temps d'exécution [1, 2], échéance 5

Modélisation de l'application en vue de son ordonnancement

Construction du ST d'un processus

- Partition des *états* du processus pour chaque ressource r
- Partition des *actions* ; expression de contraintes temporelles propres



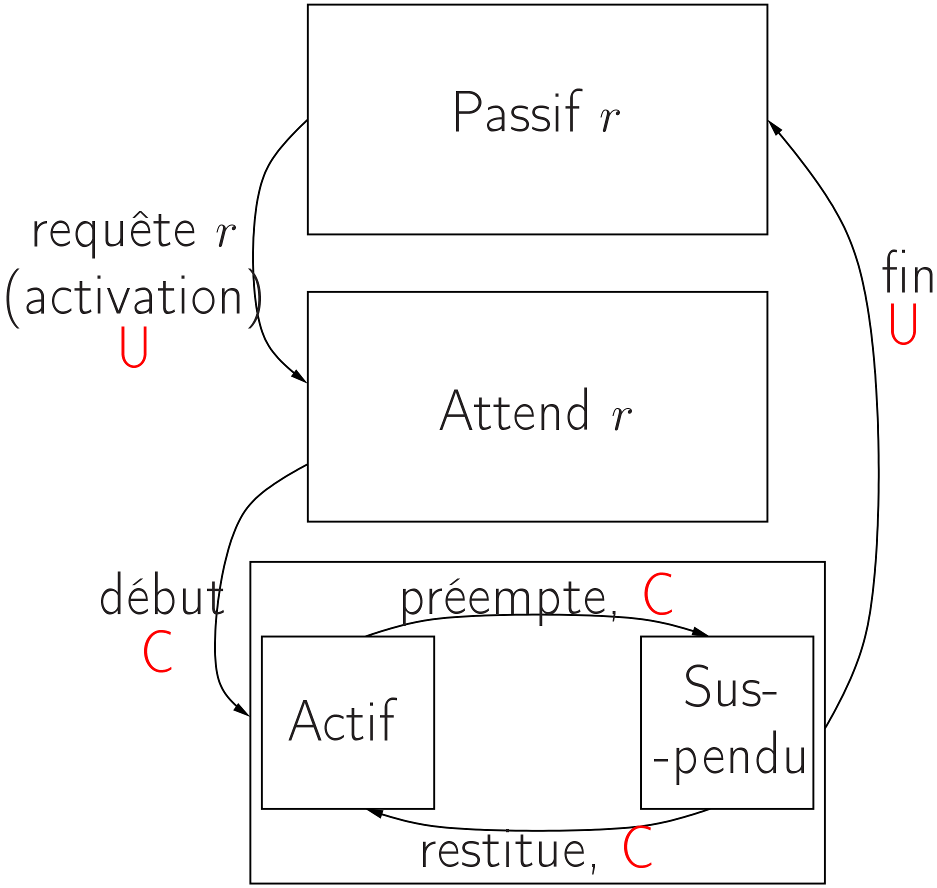
Modélisation de l'application en vue de son ordonnancement

Construction du ST d'un processus

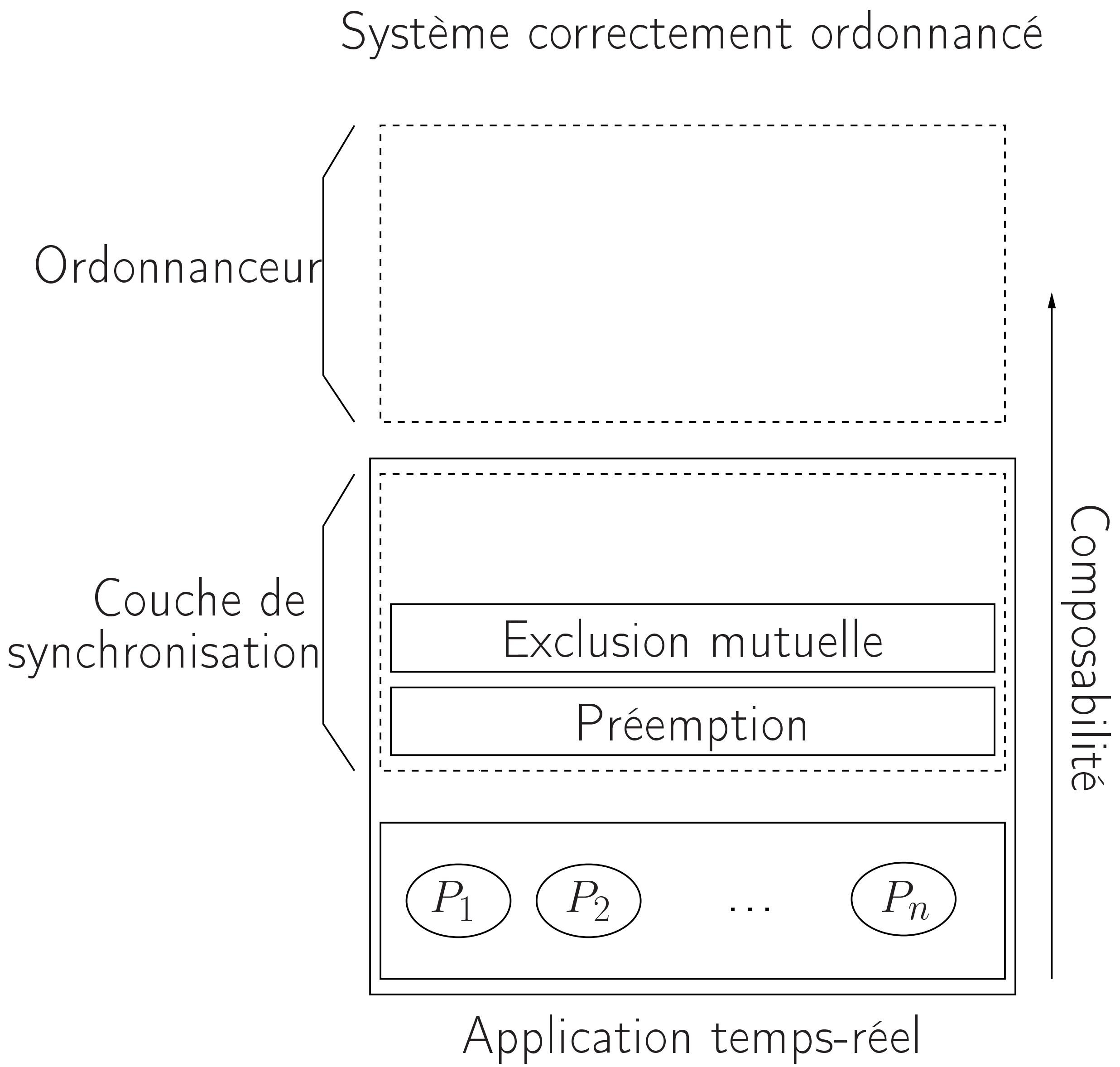
– Partition des *états* du processus pour chaque ressource r

– Partition des *actions* ; expression de contraintes temporelles propres

– Préemption



Modélisation de l'application en vue de son ordonnancement



Ordonnancement de l'application

Ordonnanceur

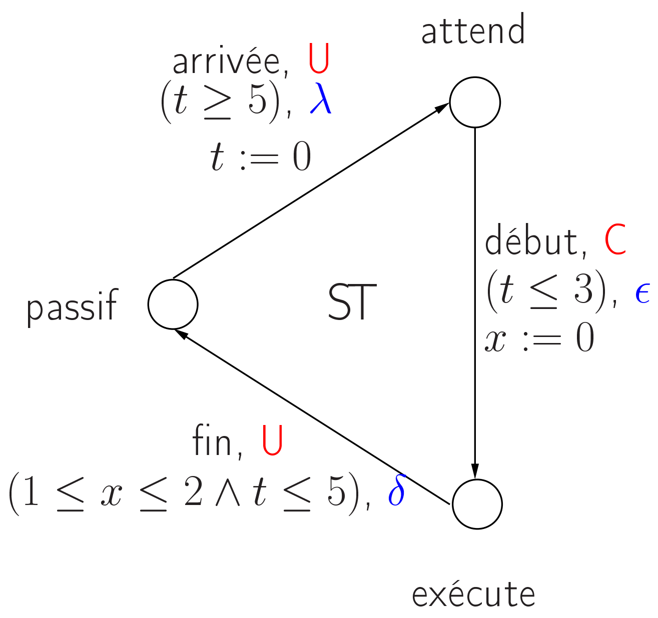
Rôle : contrôle l'accès des processus aux ressources

Objectif : respecter les contraintes temporelles propres à chaque processus

Contrainte d'ordonnançabilité : $\mathsf{K}_{\mathsf{ordo}}$

caractérise les états où le système respecte ces contraintes pour l'ensemble des processus : $\mathsf{K}_{\mathsf{ordo}} = \bigwedge_{P_i} \mathsf{K}^i_{\mathsf{ordo}}$

$$\begin{aligned}\mathsf{K}^i_{\mathsf{ordo}} = \ & \mathsf{passif} \wedge \mathsf{vrai} \vee \\ & \mathsf{attend} \wedge t \leq 3 \vee \\ & \mathsf{exécute} \wedge x \leq 2 \wedge t \leq 5\end{aligned}$$



Ordonnancement de l'application

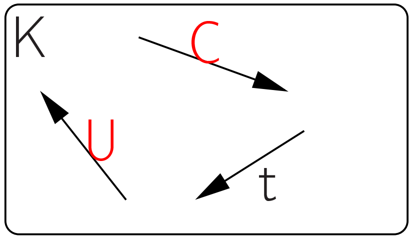
Ordonnanceur

Rôle : contrôle l'accès des processus aux ressources

Objectif : respecter les contraintes temporelles propres à chaque processus

Contrainte d'ordonnançabilité : K_{ordo}

Invariance : $ST \models inv(K_{ordo})$?



$ST \models inv(K)$

Ordonnancement de l'application

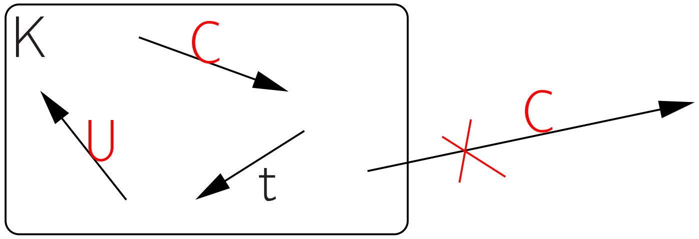
K_{ordo} n'est pas invariant du système en général

$\longrightarrow$ Trouver $K \Rightarrow K_{\text{ordo}}$ tel que

- l'ordonnanceur restreint les actions C pour maintenir K ;
- le temps et les actions U préservent K.

Opérateur de restriction : ST/K

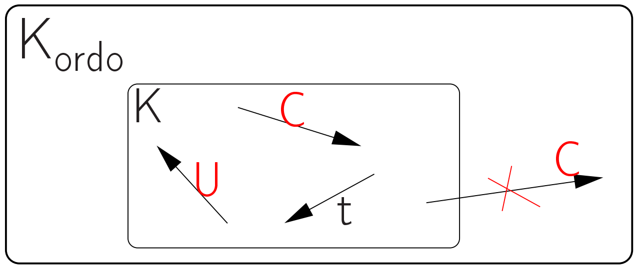
K est invariant de contrôle : $ST/K \models inv(K)$



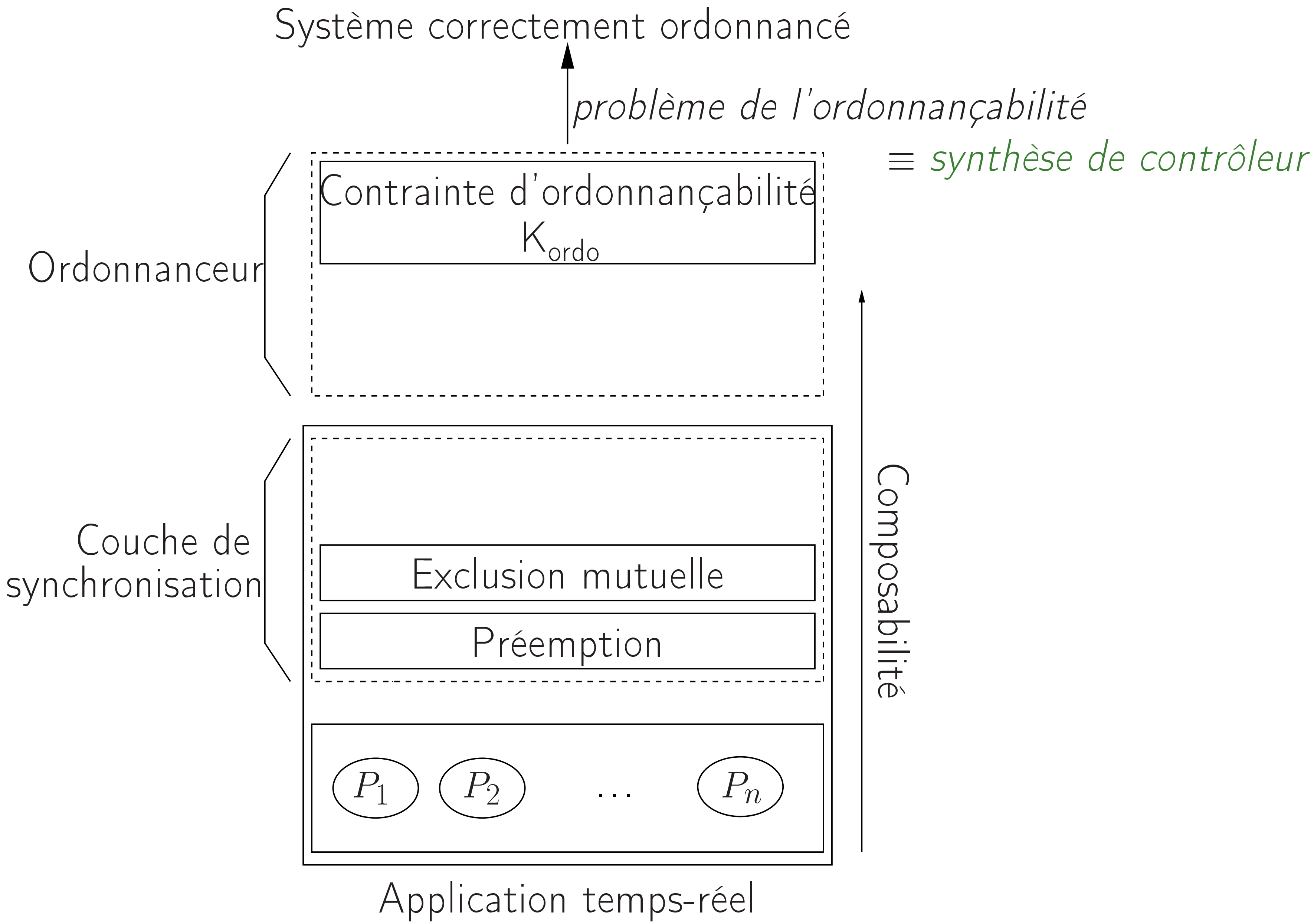
Problème de l'ordonnançabilité :

un problème de synthèse de contrôleur

$$\boxed{\text{trouver } K \text{ tq } K \Rightarrow K_{\text{ordo}} \text{ et } ST/K \models inv(K)}$$



Ordonnancement de l'application



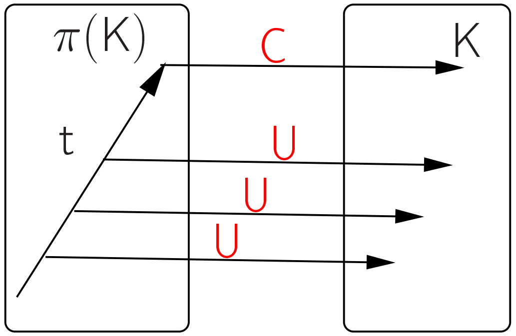
Ordonnancement de l'application

Synthèse de contrôleur maximal

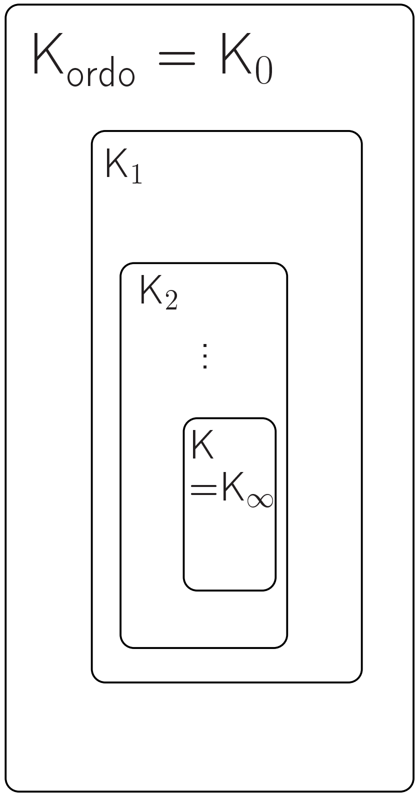
$$K \Rrightarrow K_{\mathsf{ordo}}$$

$$K = K \wedge \pi(K)$$

$\pi(K)$ = prédécesseurs contrôlables de K



Problème : trop coûteux

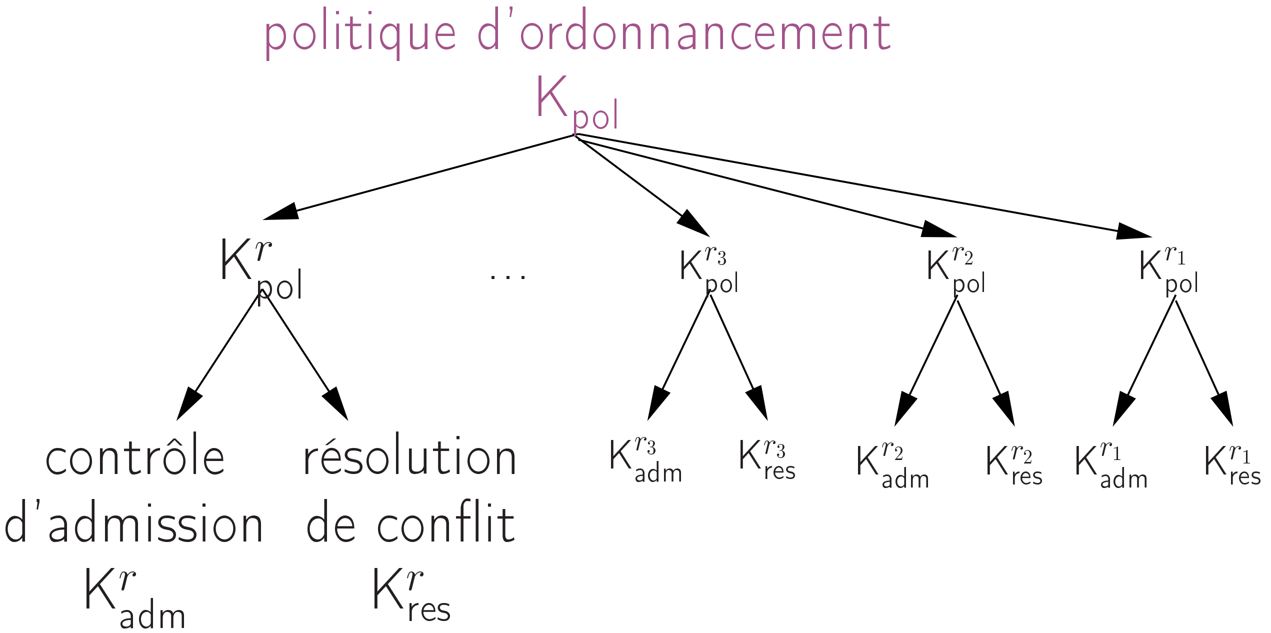


Restriction de l'ordonnanceur

Politiques d'ordonnancement

utilisées *en pratique* pour restreindre l'ordonnanceur
(ex : RM, EDF, priority ceiling protocol)

$\longrightarrow$ Spécification de l'ordonnanceur par $K = K_{ordo} \wedge K_{pol}$



Restriction de l'ordonnanceur

Politiques d'ordonnancement

$\longrightarrow$ L'ensemble des $\mathsf{K}^r_{\mathsf{pol}}$ sont des invariants de contrôle

Problème : les invariants de contrôle ne sont pas préservés par conjonction

$$\left.\begin{array}{l} \mathsf{K}_1 \text{ invariant de contrôle de ST} \\ \mathsf{K}_2 \text{ invariant de contrôle de } (\mathsf{ST}/\mathsf{K}_1) \end{array}\right\} \not\Rightarrow (\mathsf{K}_1 \wedge \mathsf{K}_2) \text{ invariant de contrôle de ST !}$$

Cause : l'opérateur de restriction ST/K

Restriction de l'ordonnanceur

Composabilité

Invariants composables = sous-classe des invariants de contrôle préservés par conjonction

K est composable s'il reste invariant de contrôle après toute autre restriction du système.

Caractérisation : K est composable $\iff \mathsf{ST}^u \models \mathsf{inv}(\mathsf{K})$

$\longrightarrow$ $\mathsf{K_{pol}}$ est composable

Problème de l'ordonnançabilité

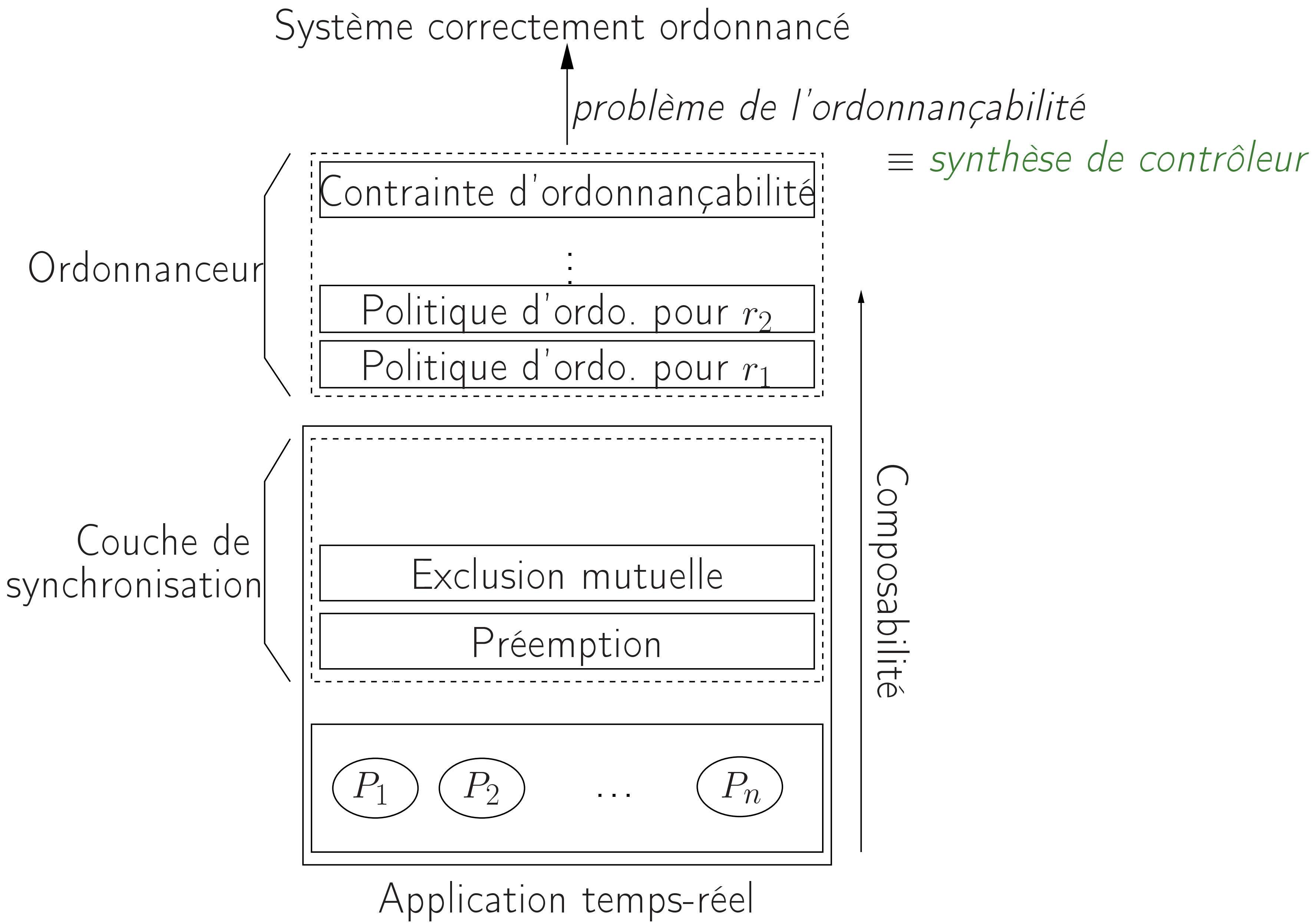
$$\text{trouver K tq K} \Rightarrow (\mathsf{K_{ordo}} \wedge \mathsf{K_{pol}}) \text{ et } \mathsf{ST}/\mathsf{K} \models \mathsf{inv}(\mathsf{K})$$

$\longrightarrow$

$$\iff$$

$$\boxed{\text{trouver K tq K} \Rightarrow \mathsf{K_{ordo}} \text{ et } (\mathsf{ST}/\mathsf{K_{pol}})/\mathsf{K} \models \mathsf{inv}(\mathsf{K})}$$

Méthode



Système correctement ordonnancé
problème de l'ordonnançabilité
≡ synthèse de contrôleur
Contrainte d'ordonnançabilité
Ordonnanceur
Politique d'ordo. pour r_2
Politique d'ordo. pour r_1
Couche de synchronisation
Exclusion mutuelle
Préemption
Composabilité
P_1
P_2
$\dots$
P_n
Application temps-réel

Algorithmes de synthèse

Synthèse de contrôleur maximal par point fixe

- implanté dans KRONOS

Synthèse de contrôleur à la volée sur une représentation discrète

- un nouvel algorithme
- n'explore que les états nécessaires pour trouver un contrôleur
- calcule une solution (pas forcément maximale)
- implanté : FLY-SYNTH

Représentation discrète obtenue par

- bisimulation par abstraction du temps
- discrétisation du temps

Applications

Dispositif de pompage

- objectif : actionner la pompe en fonction du niveau d'eau dans la mine et du taux de CH4
- modélisation : échéances entre événements appartenant à des processus distincts ; mélange d'activations par événement et par date
- taille et performances : 10 processus ; 46 s, 56 Mo

Dispositif chimique (projet VHS, cas 7)

- objectif : actionner les valves pour maintenir un flot de production permanent en sortie
- modélisation : non déterminisme ; interactions complexes entre processus
- taille et performances : 8 processus ; 3 m 30 s, 104 Mo de mémoire utilisés

Conclusion

Interprétation : Contrôleur $\longleftrightarrow$ Ordonnanceur

Problème de l'ordonnançabilité = un problème de synthèse de contrôleur

Méthodologie depuis la modélisation de l'application jusqu'à son ordonnancement

Modélisation incrémentale en vue d'ordonnancer

- Vision en couche du système
- Composabilité
- Intègre des propriétés fonctionnelles et non-fonctionnelles
- Intègre des pratiques analytiques (politiques d'ordonnancement)

Algorithmique

Deux algorithmes de synthèse de contrôleurs
(proposé pour l'un, implantés, et appliqués)

Perspectives

Synthèse de contrôleurs

- approfondir la notion de restriction
- vers une représentation discrète adaptée à la synthèse de contrôleurs

Ordonnancement temps-réel

- combiner les approches analytique et basée sur un modèle opérationnel
- application au multi-ordonnancement
- implantation de l'ordonnanceur