

# La disponibilité : sûreté ou vivacité ?

Hervé Grall

herve@grall.name

OBASCO (EMN-INRIA)

# Sûreté et vivacité – Une définition

(Alpern et Schneider - 1985)

- propriété décrite par un ensemble de traces (finies ou infinies) d'exécution

# Sûreté et vivacité – Une définition

(Alpern et Schneider - 1985)

- propriété décrite par un ensemble de traces (finies ou infinies) d'exécution
- propriété de *sûreté* (safety) –  $S$   
pour toute trace n'appartenant pas à  $S$ , il existe un préfixe fini tel que toute trace prolongeant ce préfixe n'appartient pas à  $S$  :

$$\forall \beta \notin S . \exists \alpha \text{ finie } \leq \beta . \forall \gamma . \alpha \gamma \notin S$$

# Sûreté et vivacité – Une définition

(Alpern et Schneider - 1985)

- propriété décrite par un ensemble de traces (finies ou infinies) d'exécution
- propriété de *vivacité* (liveness) –  $V$   
toute trace finie peut être prolongée en une trace appartenant à  $V$  :

$$\forall \alpha \text{ finie} . \exists \beta . \alpha \beta \in V$$

# Sûreté et vivacité – Une définition

(Alpern et Schneider - 1985)

- propriété décrite par un ensemble de traces (finies ou infinies) d'exécution
- théorème :  
Toute propriété est égale à l'intersection d'une propriété de sûreté et d'une propriété de vivacité.

# Application à la disponibilité

Actions composant les traces

# Application à la disponibilité

Actions composant les traces

- $R_i ?$  : requête  $i$  de la ressource  $R$

# Application à la disponibilité

Actions composant les traces

- $R_i ?$  : requête  $i$  de la ressource  $R$
- $[R_i$  : début d'utilisation de la ressource  $R$  en réponse à la requête  $i$

# Application à la disponibilité

## Actions composant les traces

- $R_i ?$  : requête  $i$  de la ressource  $R$
- $[R_i$  : début d'utilisation de la ressource  $R$  en réponse à la requête  $i$
- $R_i]$  : fin d'utilisation de la ressource  $R$  en réponse à la requête  $i$

# Application à la disponibilité

Actions composant les traces

- $R_i ?$  : requête  $i$  de la ressource  $R$
- $[R_i$  : début d'utilisation de la ressource  $R$  en réponse à la requête  $i$
- $R_i]$  : fin d'utilisation de la ressource  $R$  en réponse à la requête  $i$

Propriété recherchée  $\stackrel{def}{=}$  ensemble de traces défini par un système de transitions

# Application à la disponibilité

Actions composant les traces

- $R_i ?$  : requête  $i$  de la ressource  $R$
- $[R_i$  : début d'utilisation de la ressource  $R$  en réponse à la requête  $i$
- $R_i]$  : fin d'utilisation de la ressource  $R$  en réponse à la requête  $i$

Propriété recherchée  $\stackrel{def}{=}$  ensemble de traces défini par un système de transitions

État  $\stackrel{def}{=}$  application qui à toute ressource  $R$  associe deux ensembles d'identifiants de requêtes :

- ensemble des requêtes pendantes, noté  $J$
- ensemble des requêtes en cours de satisfaction, noté  $K$

# Application à la disponibilité

Actions composant les traces

- $R_i ?$  : requête  $i$  de la ressource  $R$
- $[R_i$  : début d'utilisation de la ressource  $R$  en réponse à la requête  $i$
- $R_i]$  : fin d'utilisation de la ressource  $R$  en réponse à la requête  $i$

Propriété recherchée  $\stackrel{def}{=}$  ensemble de traces défini par un système de transitions

État  $\stackrel{def}{=}$  application qui à toute ressource  $R$  associe deux ensembles d'identifiants de requêtes :

- ensemble des requêtes pendantes, noté  $J$
- ensemble des requêtes en cours de satisfaction, noté  $K$

Étiquette d'une transition  $\stackrel{def}{=}$  une action

# Bon fonctionnement

Identification correcte des requêtes  
Utilisation après requête

$$(R \mapsto (J, K)) \xrightarrow{R_i?} (R \mapsto (J \cup \{i\}, K)) \quad (i \notin J)$$

$$(R \mapsto (J, K)) \xrightarrow{[R_i} (R \mapsto (J - \{i\}, K \cup \{i\})) \quad (i \in J, i \notin K)$$

$$(R \mapsto (J, K)) \xrightarrow{R_i]} (R \mapsto (J, K - \{i\})) \quad (i \in K)$$

→ Sûreté

# Contrôle de la gestion des ressources

limiter le nombre de requêtes pendantes

limiter le nombre de requêtes en cours de satisfaction

Améliorations possibles et simples :

- stratégie pour la satisfaction des requêtes (FIFO, etc.)
- limiter la consommation totale des ressources

→ Sûreté

# Disponibilité

Toute requête est satisfaite.

$$\forall \text{ trace } (\beta_n)_n . \forall \text{ requête } R_i ? . \forall n . \\ \beta_n = R_i ? \Rightarrow \exists p > n . \beta_p = [R_i$$

→ Vivacité

# Disponibilité

Toute requête est satisfaite.

$$\forall \text{ trace } (\beta_n)_n . \forall \text{ requête } R_i ? . \forall n . \\ \beta_n = R_i ? \Rightarrow \exists p > n . \beta_p = [R_i$$

→ Vivacité

Toute requête est satisfaite en un « temps » borné  $B$ .

$$\forall \text{ trace } (\beta_n)_n . \forall \text{ requête } R_i ? . \forall n . \\ \beta_n = R_i ? \Rightarrow \exists p > n . (\beta_p = [R_i) \wedge (p - n < B)$$

→ Sûreté

Toute utilisation s'arrête etc.