19 mai 2015

Stabilité des systèmes communicants

Introduction à la Recherche en Laboratoire

DELANAUX Rémy
ENSIMAG 2A - 2015

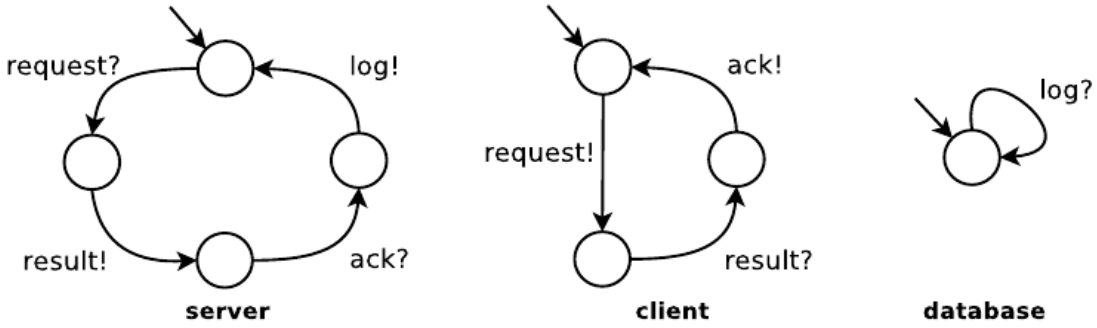
CONTEXTE DE RECHERCHE

- Systèmes communicants
  - Omniprésents (réseaux, cloud, …)
  - Séparés en briques fonctionnelles : **« peers »**

- Différents modes de communication
  - Synchrone (téléphone, …)
  - Asynchrone (e-mail, …)

CONTEXTE DE RECHERCHE

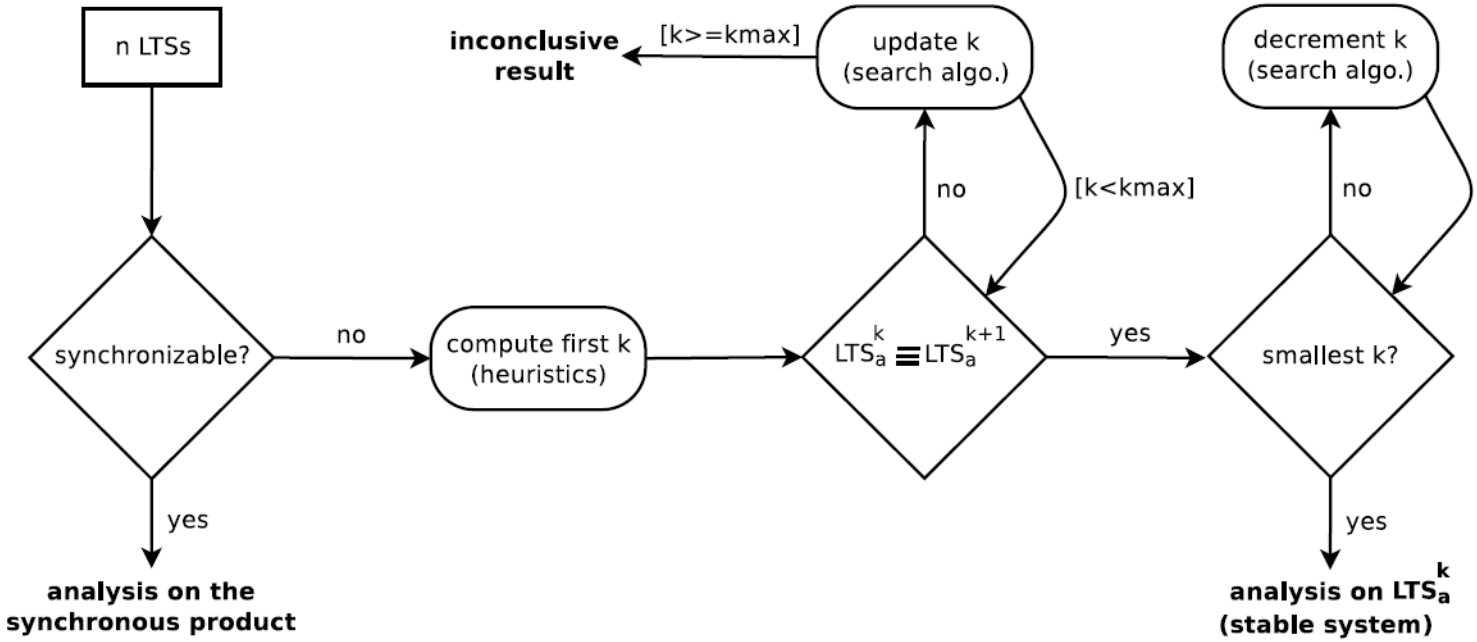
- Etudes réalisées sur des **LTS**
- Modélise des pairs communicant de façon **asynchrone**
  - Utilisation de buffers (en FIFO) infinis
  - Possibilité d'espaces d'états infinis
- Prouver leur bon comportement : **indécidable** [J.ACM 1983]



STABILITE D'UN SYSTEME

- Classes de systèmes à analyser plus facilement
  - **Synchronisabilité** [POPL 2012]
    - Même comportement entre système synchrone et asynchrone de borne de buffer 1
    - Très fort et rare
- Notion de stabilité [Salaün-Ye, 2014]
  - Stable à une borne k => stable **pour toutes les bornes supérieures**
  - Propriétés conservées pour bornes supérieures et buffers non-bornés

METHODOLOGIE



PROBLEMES OUVERTS

- ➔ <u>But :</u> améliorer la détection de la stabilité d'un système : performances, méthodologie, …
  - Traitement d'un système : jusqu'à une heure !

- Méthode de **calcul des bornes**

- **Uniformité des bornes**

- **Analyses formelles** : heuristiques structurelles

ANALYSE STRUCTURELLE

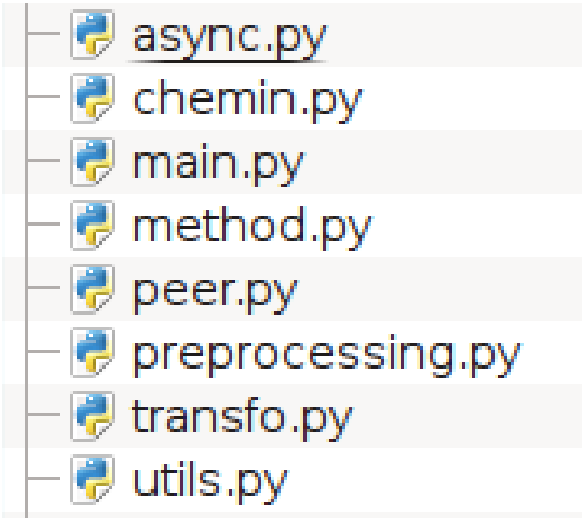
- Détection de **sous-classes de systèmes** qu'on sait traiter : heuristiques étudiées
  - Systèmes séquentiels
    - ➔ Stable
  - Système à cycle unique
    - ➔ Instable si cycle d'émissions, stable sinon
  - Systèmes à cycles complémentaires
  - Systèmes plus complexes

ANALYSE STRUCTURELLE

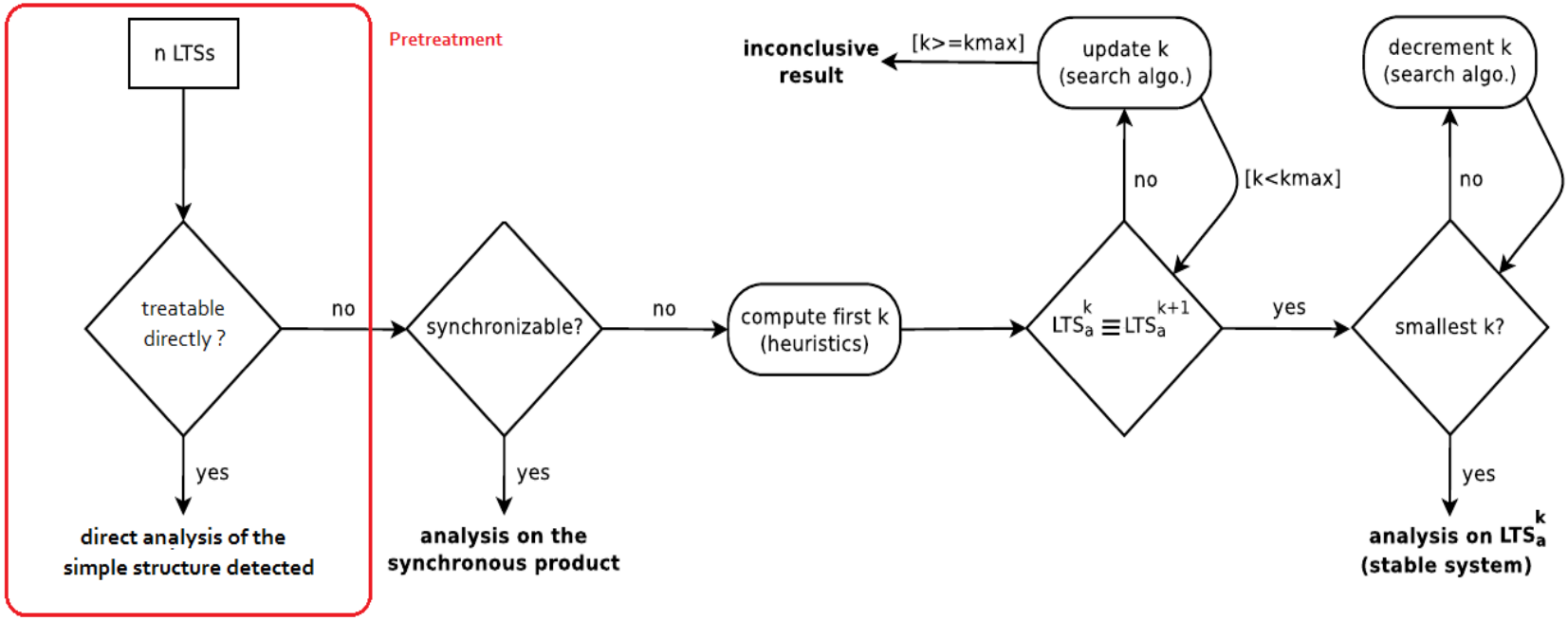
- Implémentation **d'algorithmes de détection**
  - Utilisés séquentiellement, et sur chaque pair
  - Détection de cycles :
    - Cycles d'émission
    - Cycles successifs
    - Cycles imbriqués, …
  - Vérifications formelles
    - Cycle initial d'un pair

REPRISE DU CODE

- Code Python conséquent (> 2000 lignes)
- **Refactoring** en unités sémantiques
- Ajouts
  - Méthodes de pré-traitement
  - Algorithmes de détection au niveau des systèmes *(async)* et pairs *(peer)*



METHODOLOGIE (2)



EXPERIMENTATION

- **Validation** sur banque de systèmes
  - Systèmes tirés d'exemples réels
  - Systèmes réalisés à la main

- Traitement possible pour **> 50% des systèmes**
  - Gain conséquent de temps !
    - < 1 minute pour traiter les 300 systèmes
  - Méthode simple, claire, et moins coûteuse

CONCLUSION

- Améliorations heuristiques
  - Implémentation d'un pré-traitement efficace
  - Etudes théoriques plus poussés à faire
- Bonne immersion dans la recherche
  - A la frontière du théorique et du concret
  - Découverte d'autres méthodologies de travail

Références :

- Brand, D., and Zafiropulo, P. *On communicating finite-state machines*, 1983.
- Basu, S., Bultan, T., and Ouederni, M. *Deciding choreography realizability*, 2012.
- Salaün, G., and Ye, L. *Stability of Asynchronously Communicating Systems*, 2014.

19 mai 2015

Des questions ?

Merci pour votre attention !