

Candidature au concours CNRS 06/01

Radu IOSIF (VERIMAG)
CNRS / Université Grenoble Alpes

Vérification de programmes

```
struct node_t {  
    int data;  
    struct node_t *next;  
} *p;  
  
...  
while (p->data > 0)  
    p = p->next;
```



propriétés de sûreté

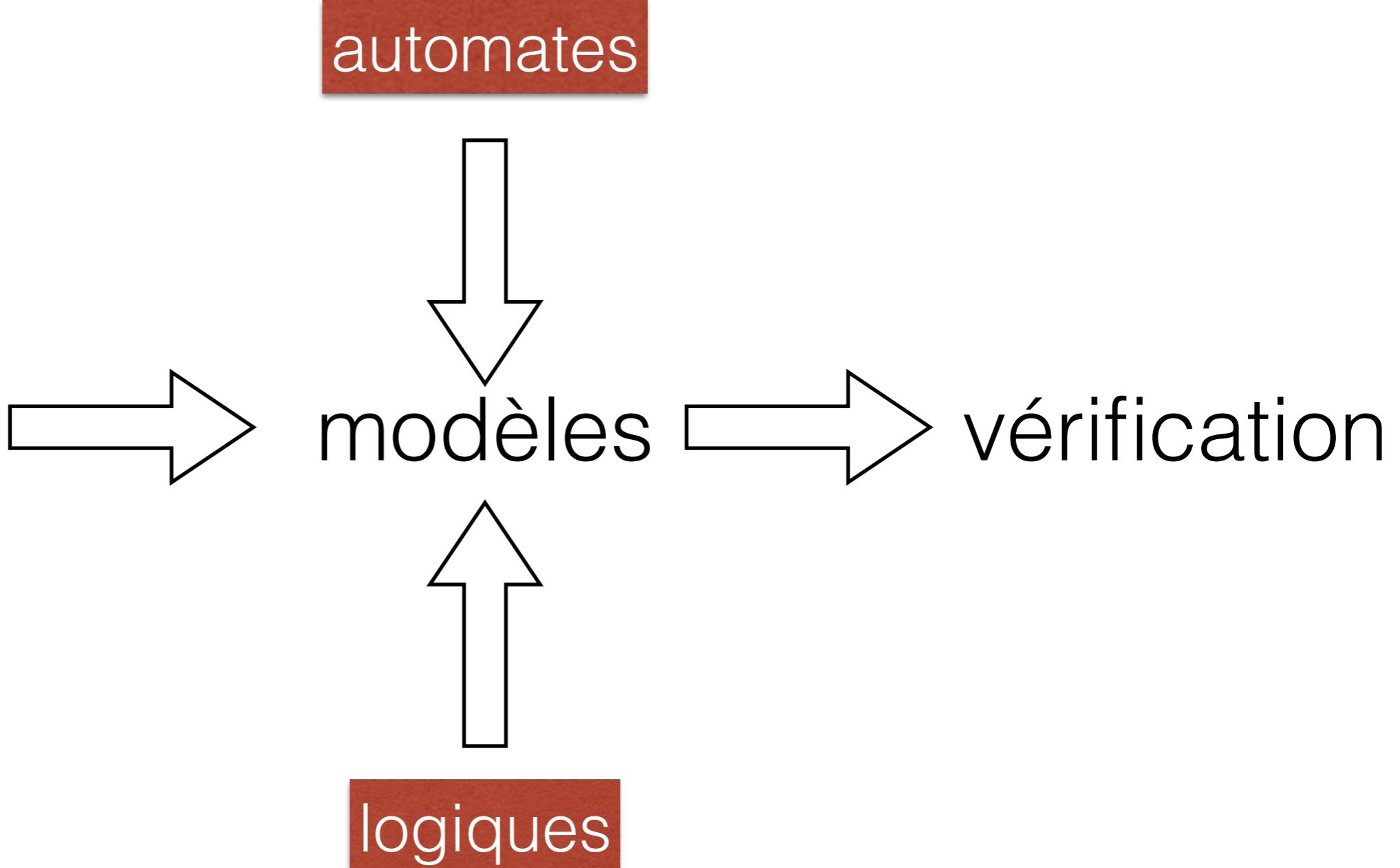
- pas de déréférencement de pointeur nul
- pas de fuites de mémoire

propriétés de vivacité

- terminaison
- chaque requête a une réponse

Vérification de programmes

```
struct node_t {  
    int data;  
    struct node_t *next;  
} *p;  
...  
while (p->data > 0)  
    p = p->next;
```



Extraction de modèles

programmes
avec listes

[Bouajjani, Bozga, Habermehl, I, Moro, Vojnar CAV'06, FMSD'10]
[Bozga, I, VMCAI'07]

programmes
avec arbres

[I, Rogalewicz CIAA'09]

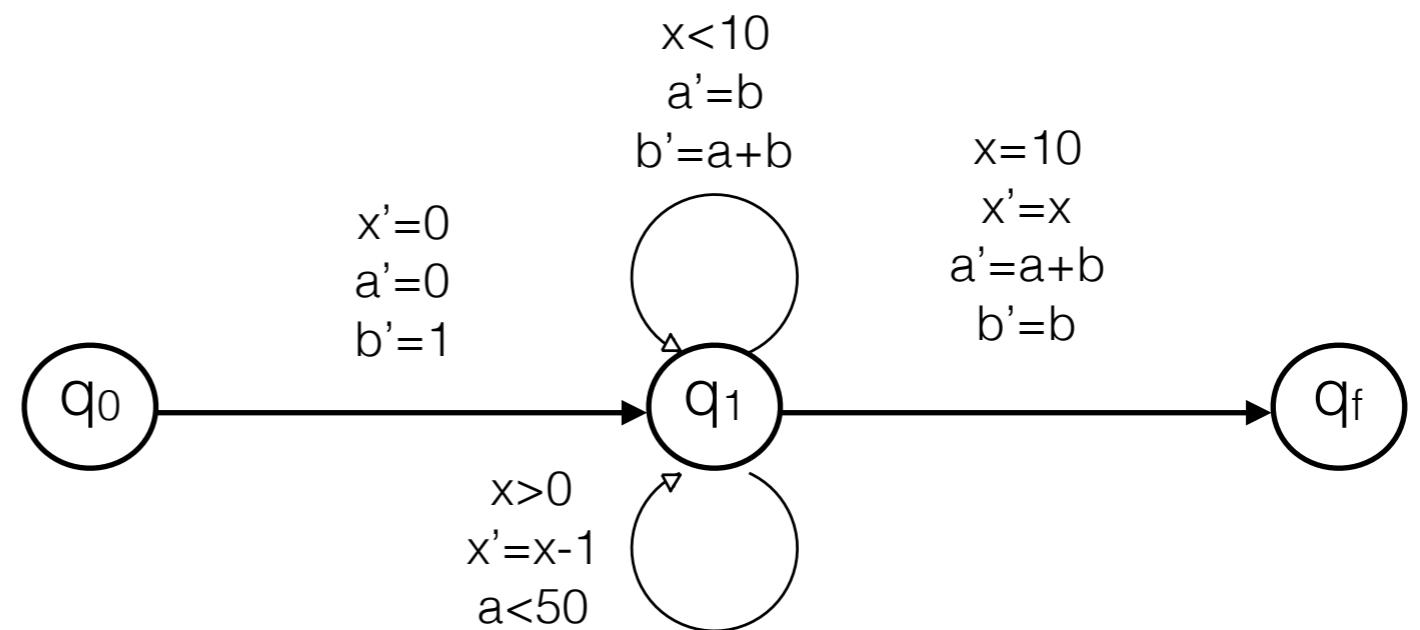
[Habermehl, I, Rogalewicz, Vojnar ATVA'07]

machines à
compteurs

programmes
avec vecteurs

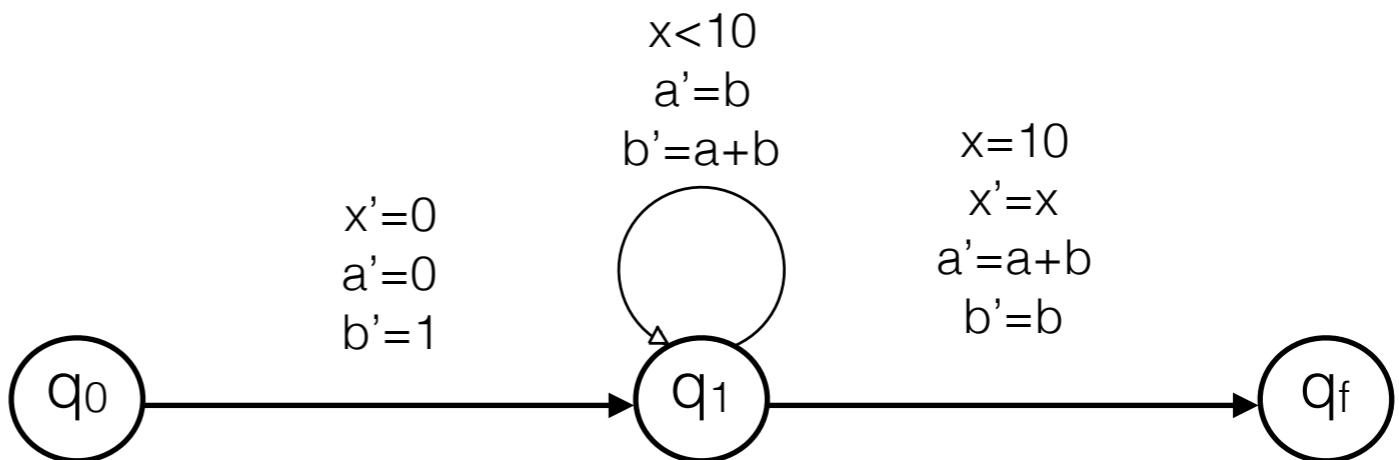
[Habermehl, I, Vojnar, FOSSACS'08, LPAR'08]
[Bozga, Habermehl, I, Konecny, Vojnar CAV'09]

Machines à compteurs

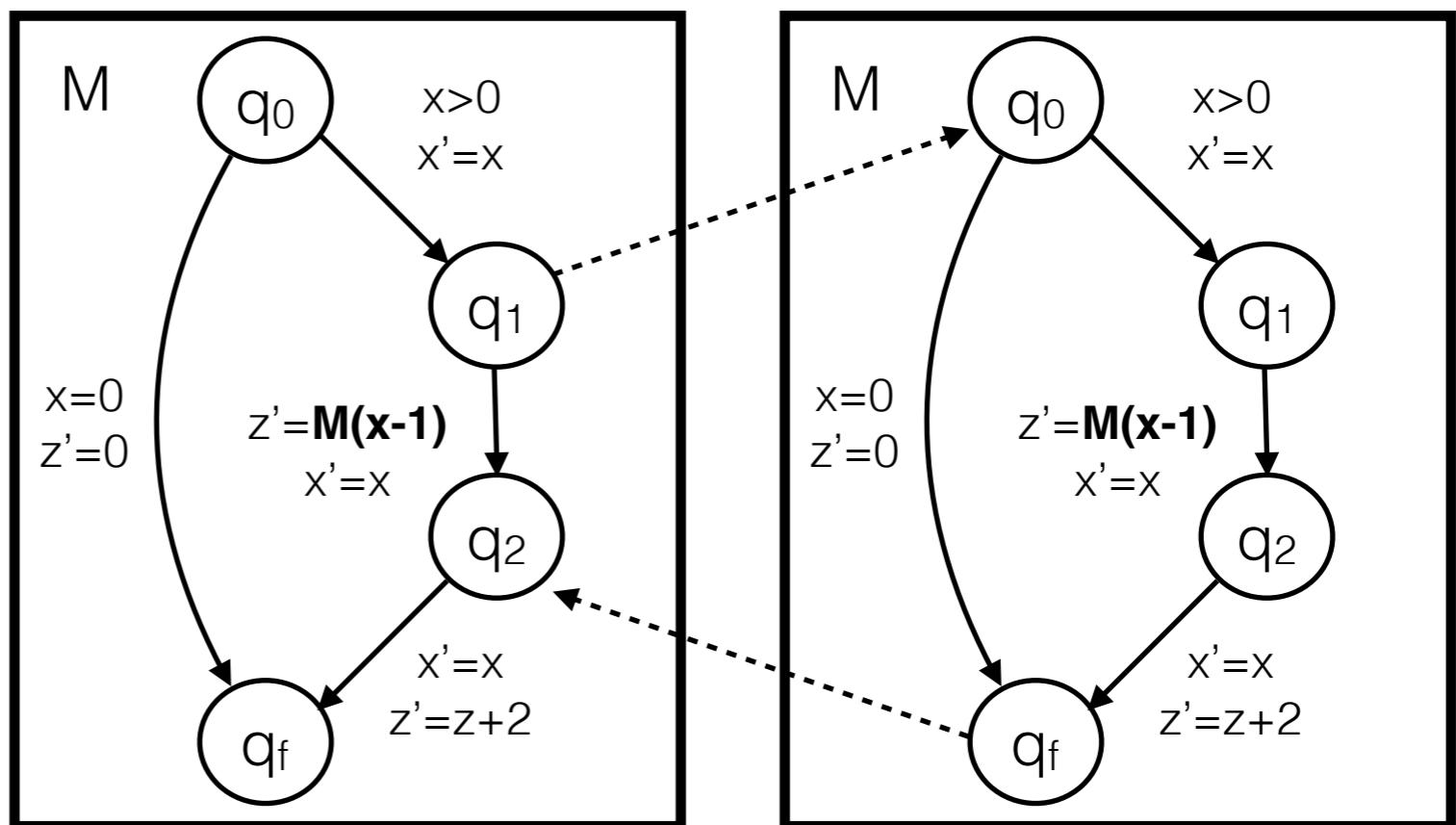


Machines à compteurs

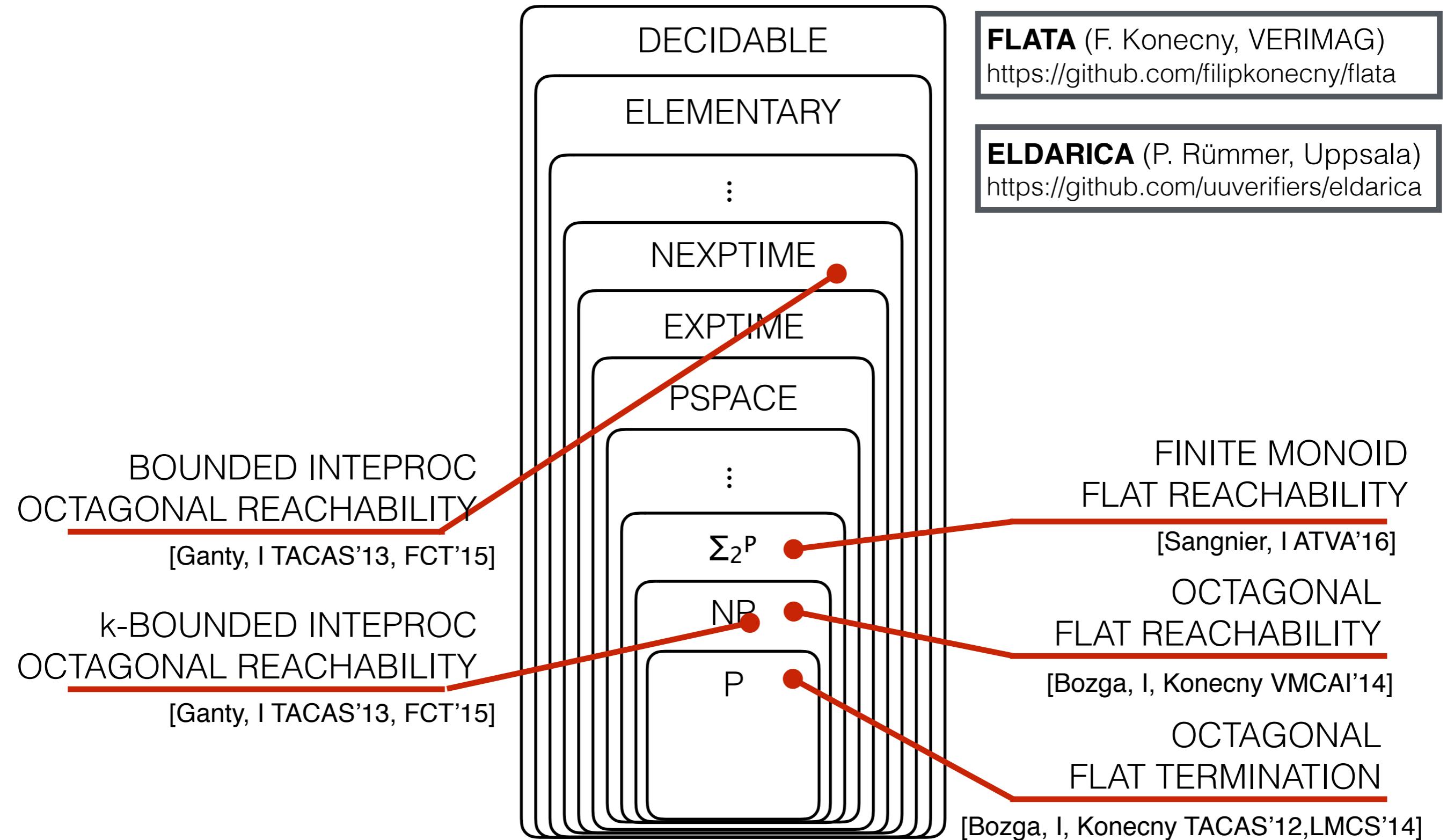
plates



récursives



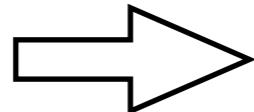
Vérification par acceleration



Automates avec alphabets infinis

Traces d'exécution = suites de valeurs des variables (alphabet infini)

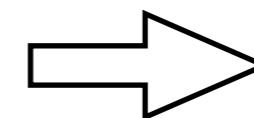
Inclusion entre
langages de traces



[I, Holik, Rogalewicz, Vojnar
TACAS'16, FMSD'20]

automate alternant
booléen

[I, Xu TACAS'18]



automate alternant
de premier ordre

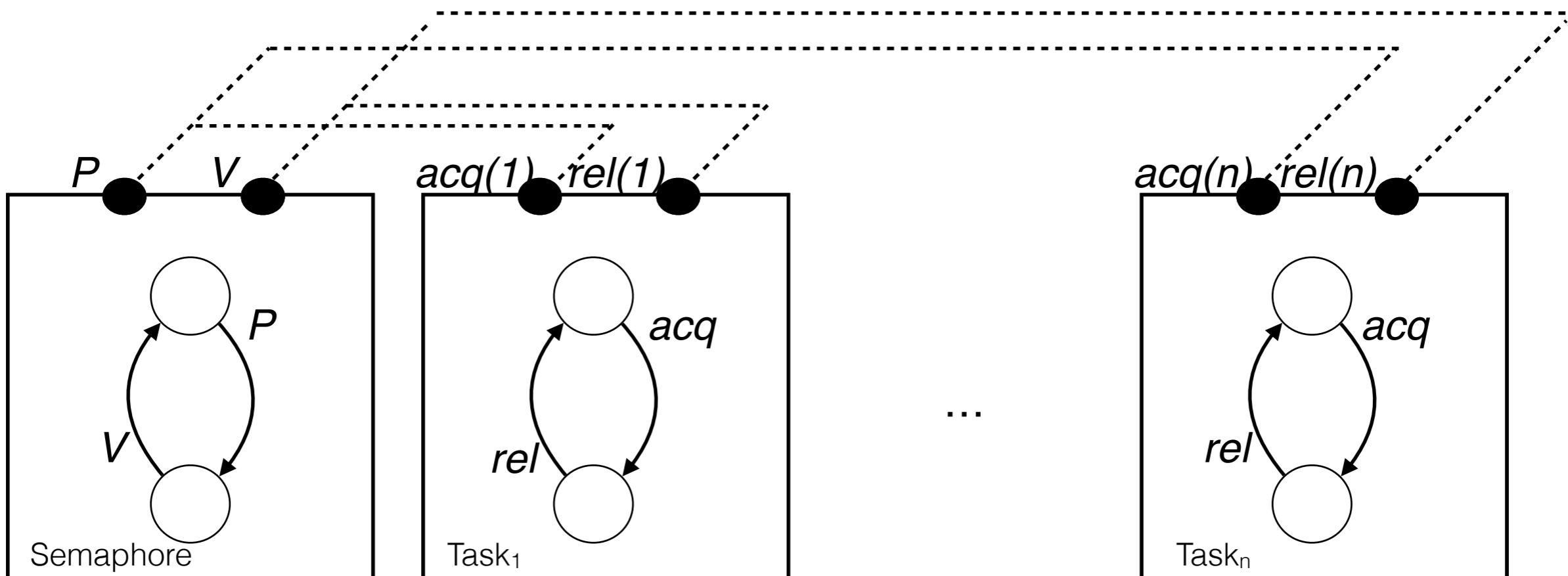
[I, Xu CAV'19]

Raffinement d'abstraction par interpolants (arithmétique linéaire + fonctions)

INCLUDER (A. Rogalewicz, Univ. Tech. Brno)
<https://www.fit.vutbr.cz/research/groups/verifit/tools/includer/>

FOADA (X. Xu, VERIMAG)
<https://github.com/cathiec/FOADA>

Systèmes parallels paramétrés



$N = \#$ non borné de composants (machines d'états finis) et d'interactions

Preuves de sûreté pour tout $N \geq 2$ par invariants structurels (trapes, mutex)

- ▶ architectures symétriques (cliques) [Bozga, I, Sifakis, TACAS'19, JLAMP'21]
- ▶ architectures linéaires/arborescentes [Bozga, Esparza, I, Sifakis, Welzel, TACAS'20]
- ▶ architectures récursives [Bozga, I, ARXIV'21]

Logique de séparation

Logique sous-structurelle de premier ordre pour décrire le tas de mémoire [Reynolds'99]

- opérateurs adjoints booléens (\wedge, \rightarrow) et **spatiaux** ($*$, $-*$)

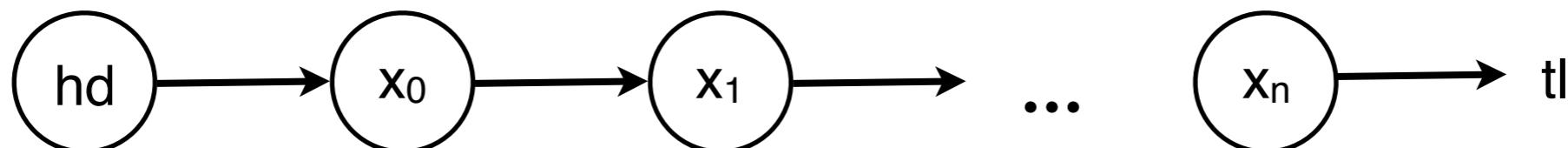
$$\begin{array}{c} \exists \\ \boxed{\phi \quad \psi} \\ \phi * \psi \end{array}$$

$$\frac{\{\phi\} \vdash \{\psi\}}{\{\phi * F\} \vdash \{\psi * F\}} \text{ (frame rule)}$$

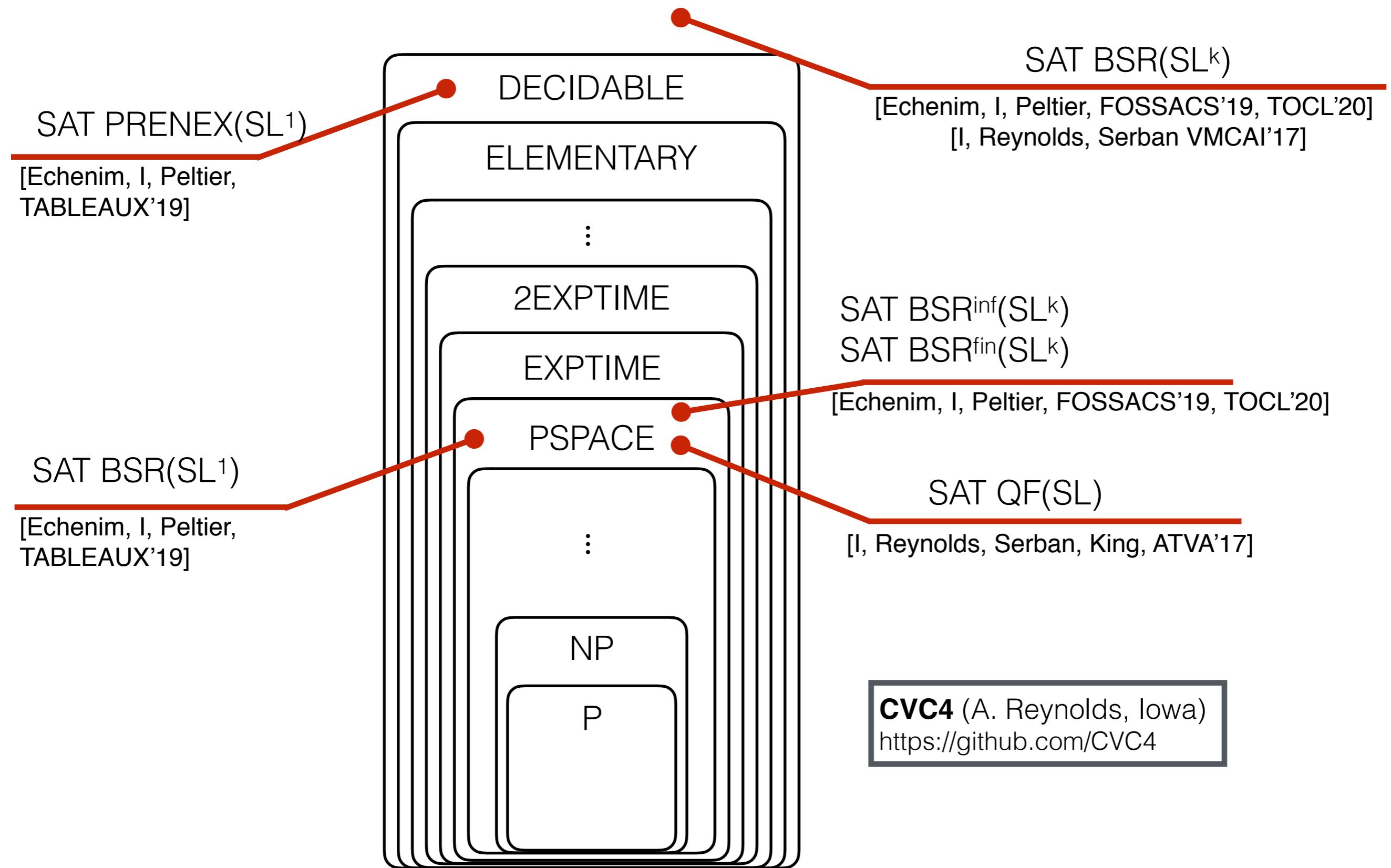
Logique de Hoare basée sur du **raisonnement local**

Définitions inductives pour décrire des structures de données récursives

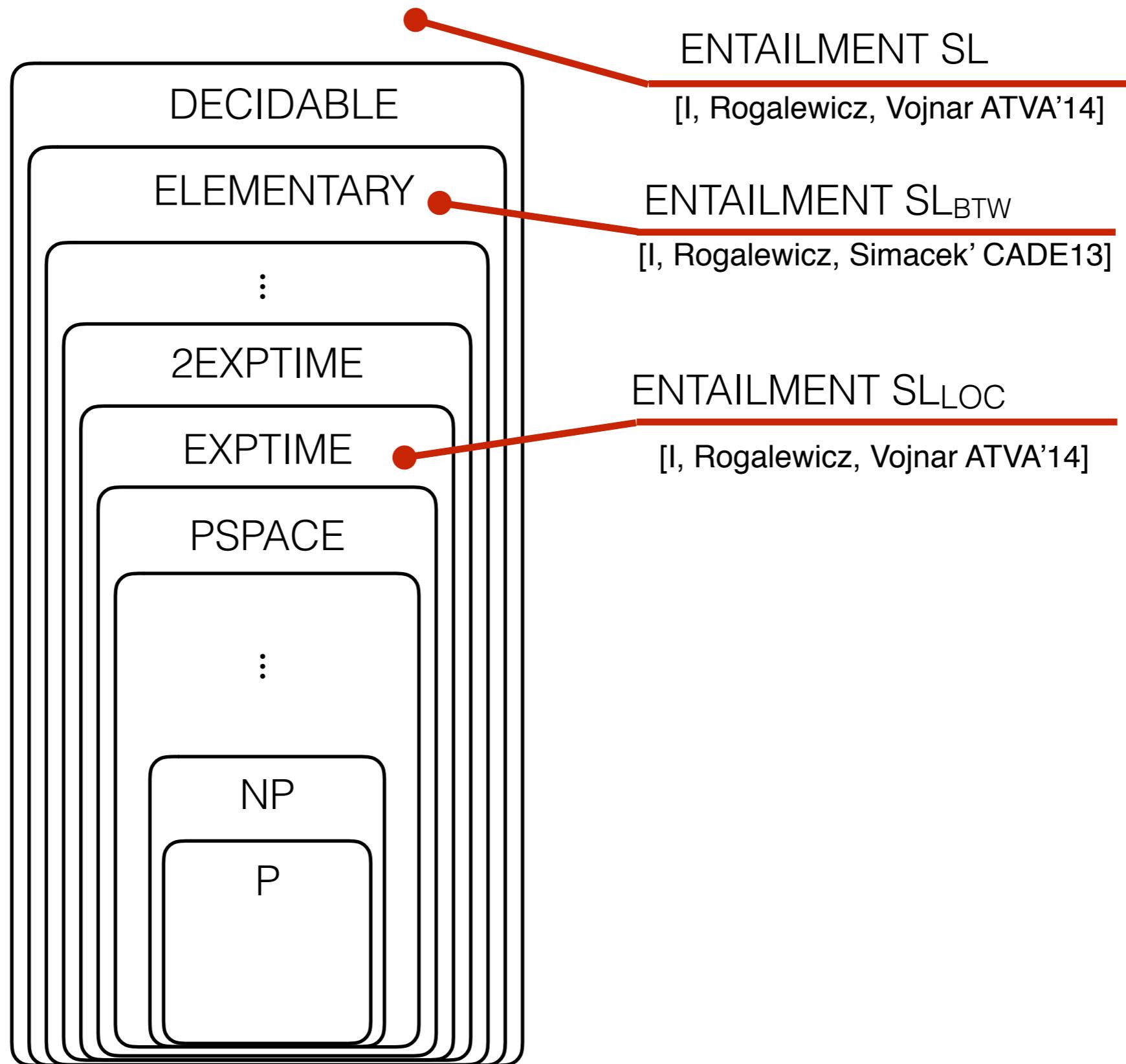
$$lseg(hd, tl) \Leftarrow emp \wedge hd = tl \mid \exists x . hd \rightarrow x * lseg(x, tl)$$



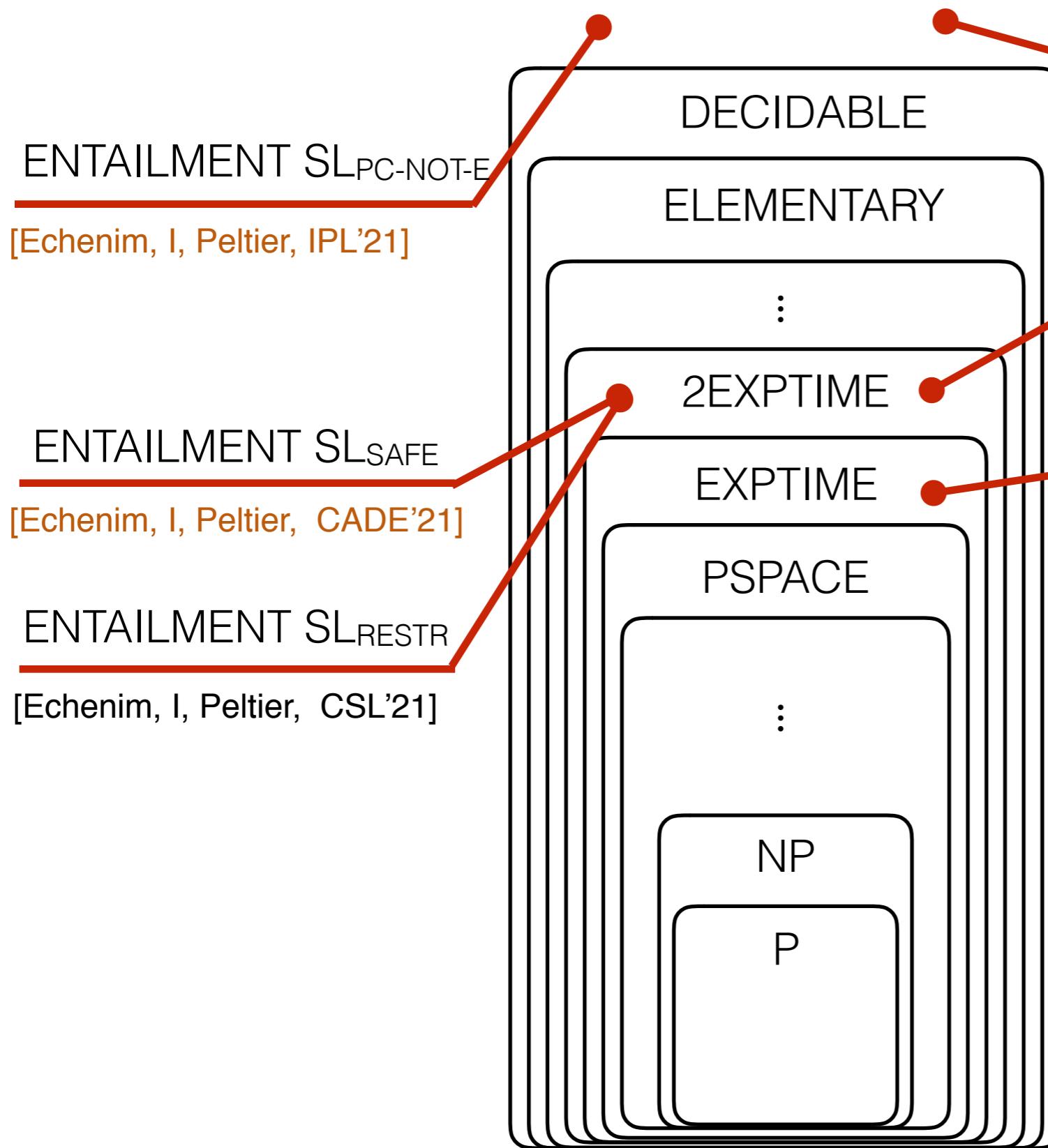
Satisfiabilité (premier ordre)



Entailment (ordre supérieur)



Entailment (ordre supérieur)



ENTAILMENT SL
[I, Rogalewicz, Vojnar ATVA'14]

ENTAILMENT SL_{BTW}
[I, Rogalewicz, Simacek' CADE13]
[Zuleger, Pagel, LPAR'20]
[Echenim, I, Peltier, LPAR'20]

ENTAILMENT SL_{LOC}
[I, Rogalewicz, Vojnar ATVA'14]

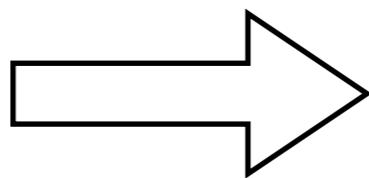
SLIDE (A. Rogalewicz, Brno)
<http://www.fit.vutbr.cz/research/groups/verifit/tools/slides/>

INDUCTOR (C. Serban, VERIMAG)
<https://github.com/cristina-serban/inductor>

Travaux en cours et futurs

Vérification de systèmes distribués

Programme sur
ordinateur isolé



Réseaux
interconnectés

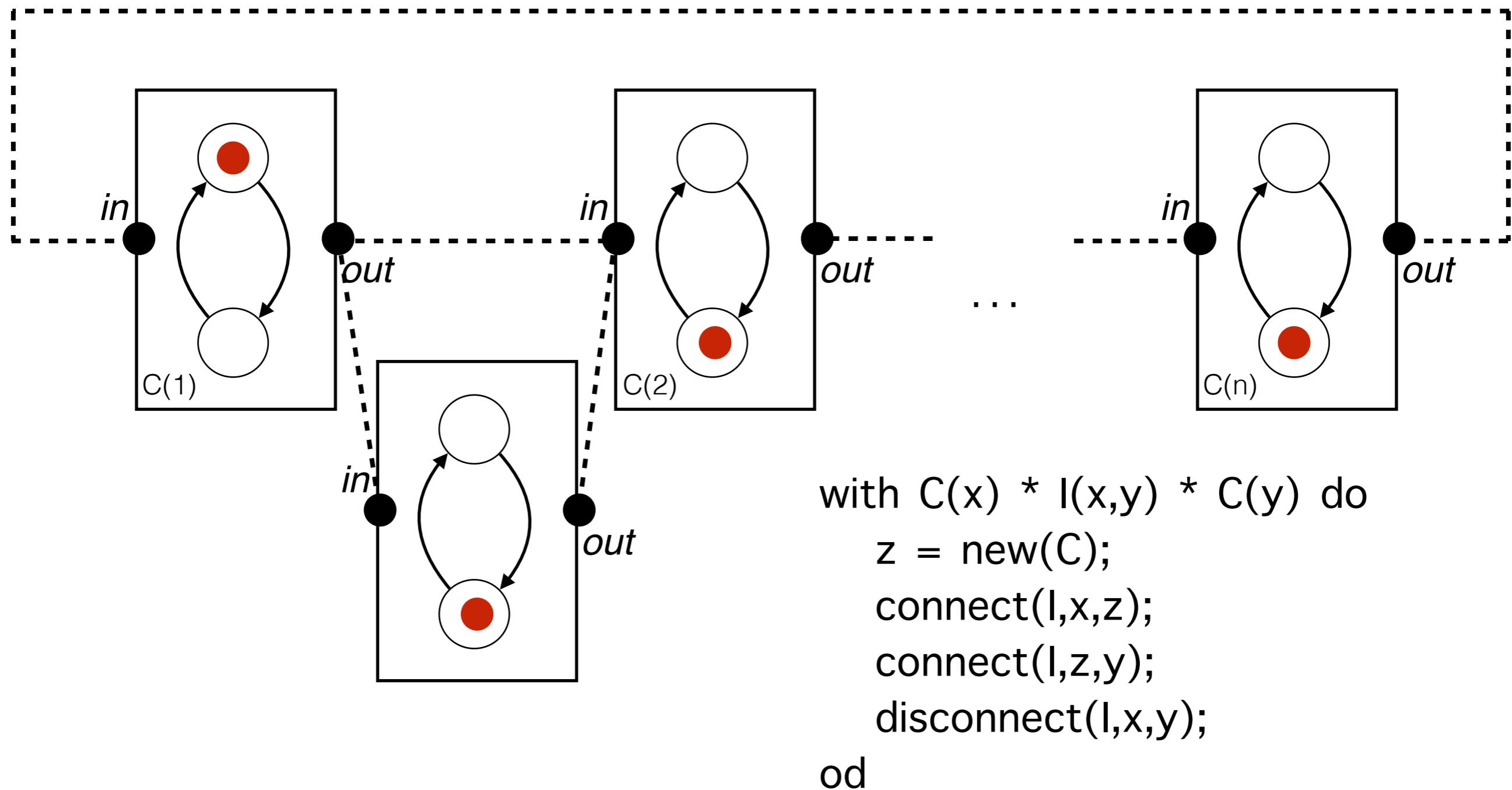
1970 — 2000

2000 — au jour

Modularité: distinction comportement / coordination

- ▶ Modèles de coordination (architectures) reconfigurables
- ▶ Modèles de comportement étendus (donnés, temps réel, hybrides)

Systèmes paramétrés reconfigurables



Systèmes paramétrés reconfigurables

Logique de ressources adaptée à la spécification:

- ▶ des familles d'architectures récursives (chaînes, anneaux, arbres)
- ▶ des actions de reconfiguration (axiomes locaux + frame rule)

E. Ahrens “Local Reasoning for Reconfigurable Distributed Systems” (RWTH Aachen)

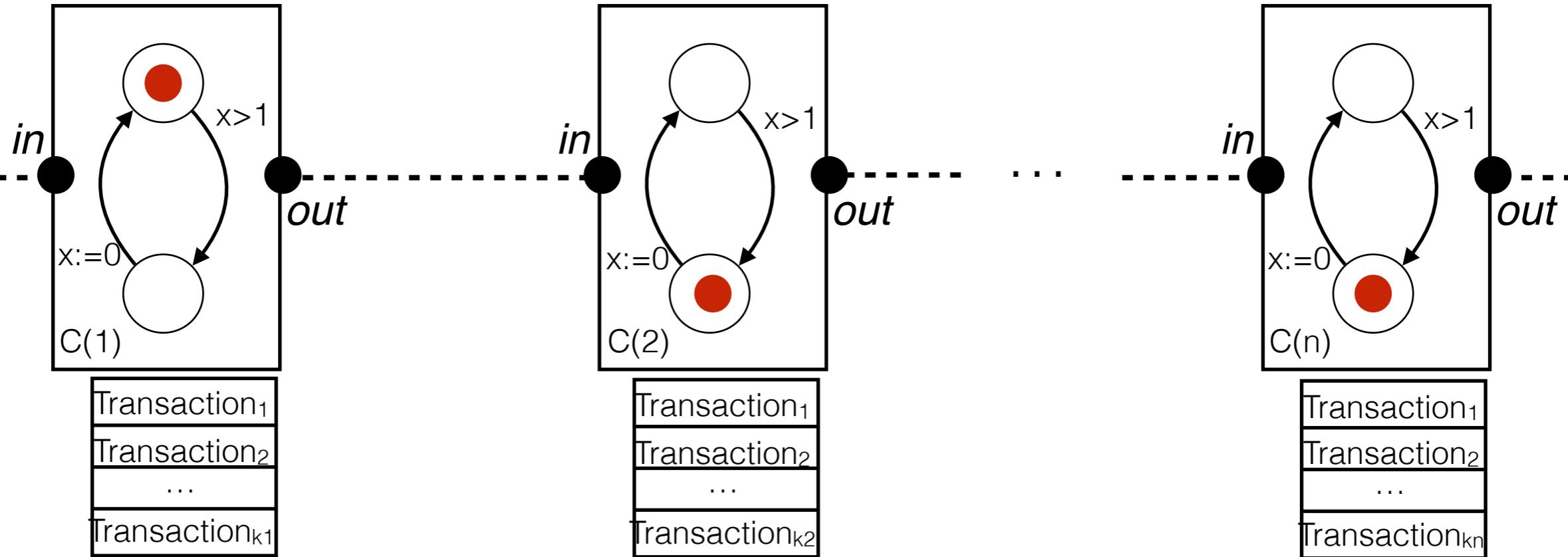
Raisonnement local automatisé:

- ▶ preuves en logique de Hoare des séquences de reconfiguration
- ▶ procédures de décision pour la nouvelle logique de ressources

ANR NARCO “Non-Aggregative Resource COnposition” (en cours)

ANR LORDS “LOcal Reasoning for Reconfigurable Distributed Systems” (en cours)

Comportements étendus



Extensions du modèle de comportement à états finis:

- ▶ structures de données non-bornées (journaux de transaction)
- ▶ horloges temps réel et variables continues

Conclusions

Travaux précédents (1998-2020)

- ▶ extraction de modèles
- ▶ machines à compteurs
- ▶ automates avec alphabets infinis
- ▶ systèmes parallels paramétrés
- ▶ logique de séparation

Travaux en cours et futurs

- ▶ systèmes distribués reconfigurables
- ▶ modèles de comportement étendus