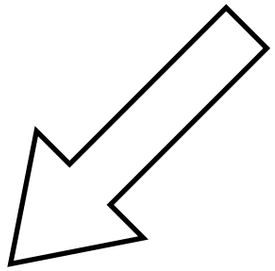
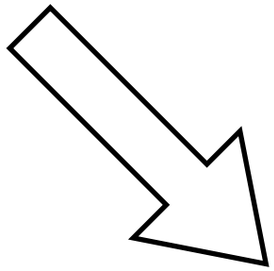
Candidature au concours CNRS 06/01

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

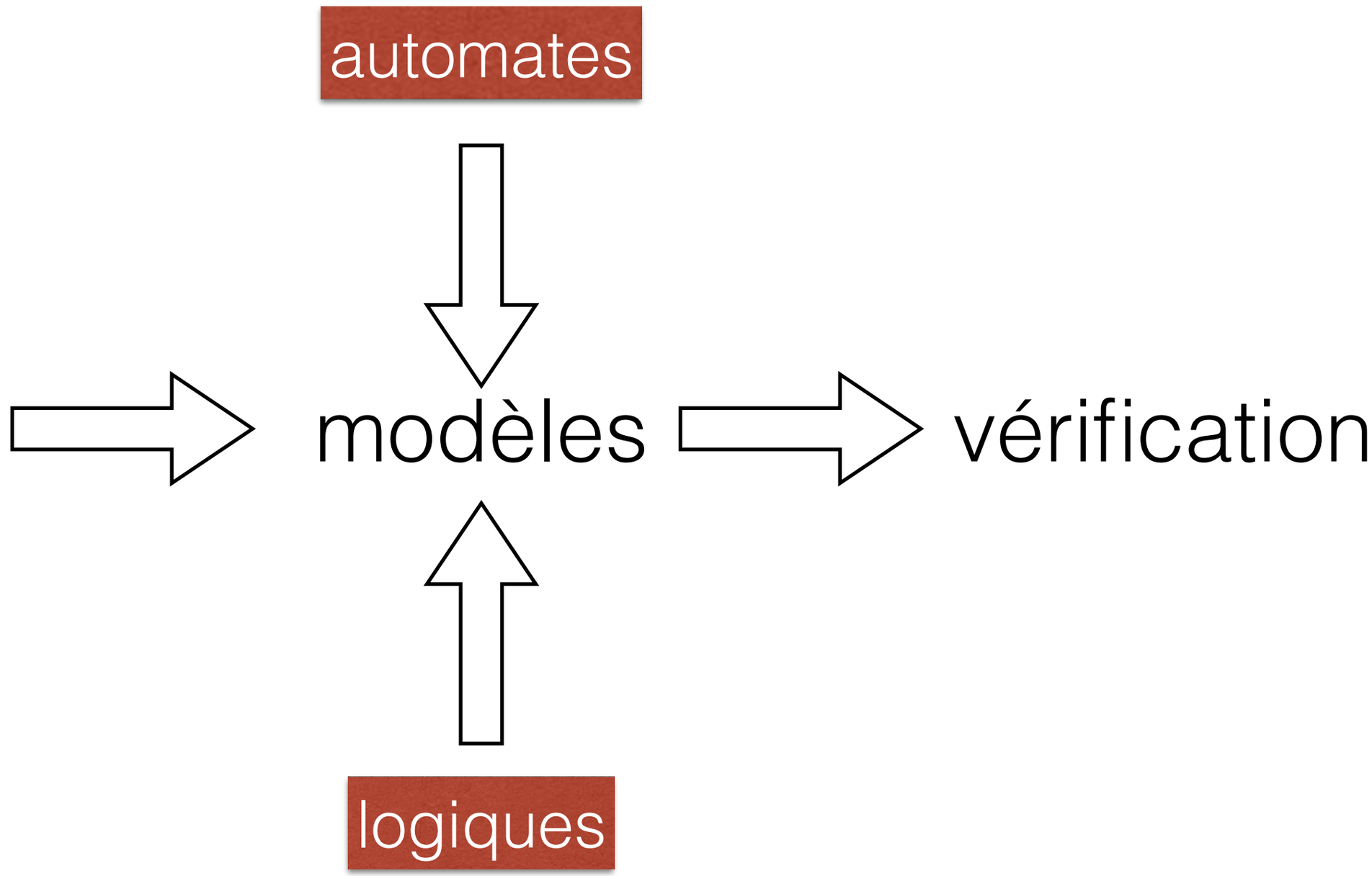
- pas de déréférencement de pointeur nul
- pas de fuites de mémoire

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;
```



Travaux précédents (2002-2020)

Extraction de modèles

- automates finis (model-checking)
- automates étendus (machines à compteurs)
- logiques (vecteurs d’entiers, tas de mémoire)

Vérification de modèles

- machines à compteurs (accélération)
- automates sur alphabets infinis (semi-algorithmes)
- systèmes parallels paramètrés (invariants)
- logique de séparation (procédures de décision)

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]

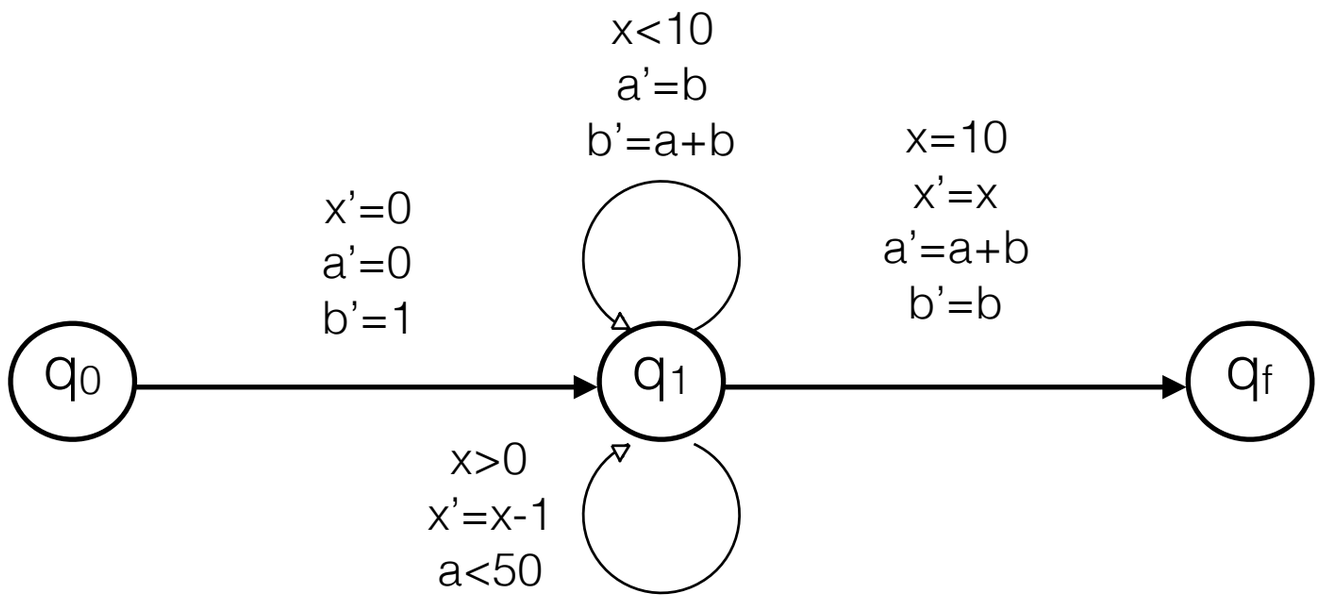
[Habermehl, I, Rogalewicz, Vojnar ATVA'07]

machines à
compteurs

programmes
avec vecteurs

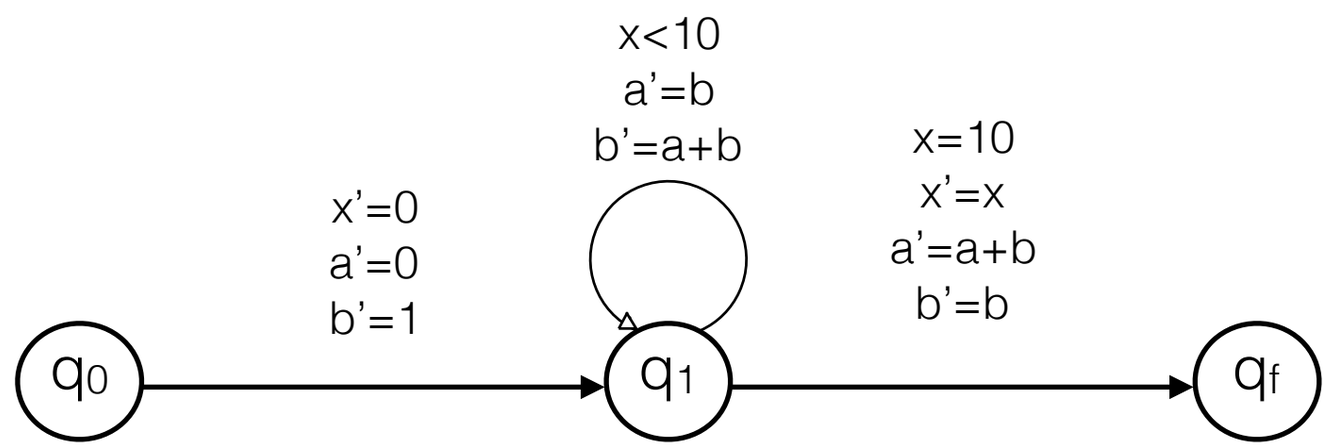
[Habermehl, I, Vojnar, FOSSACS'08, LPAR'08]
[Bozga, Habermehl, I, Konecny, Vojnar CAV'09]

Machines à compteurs

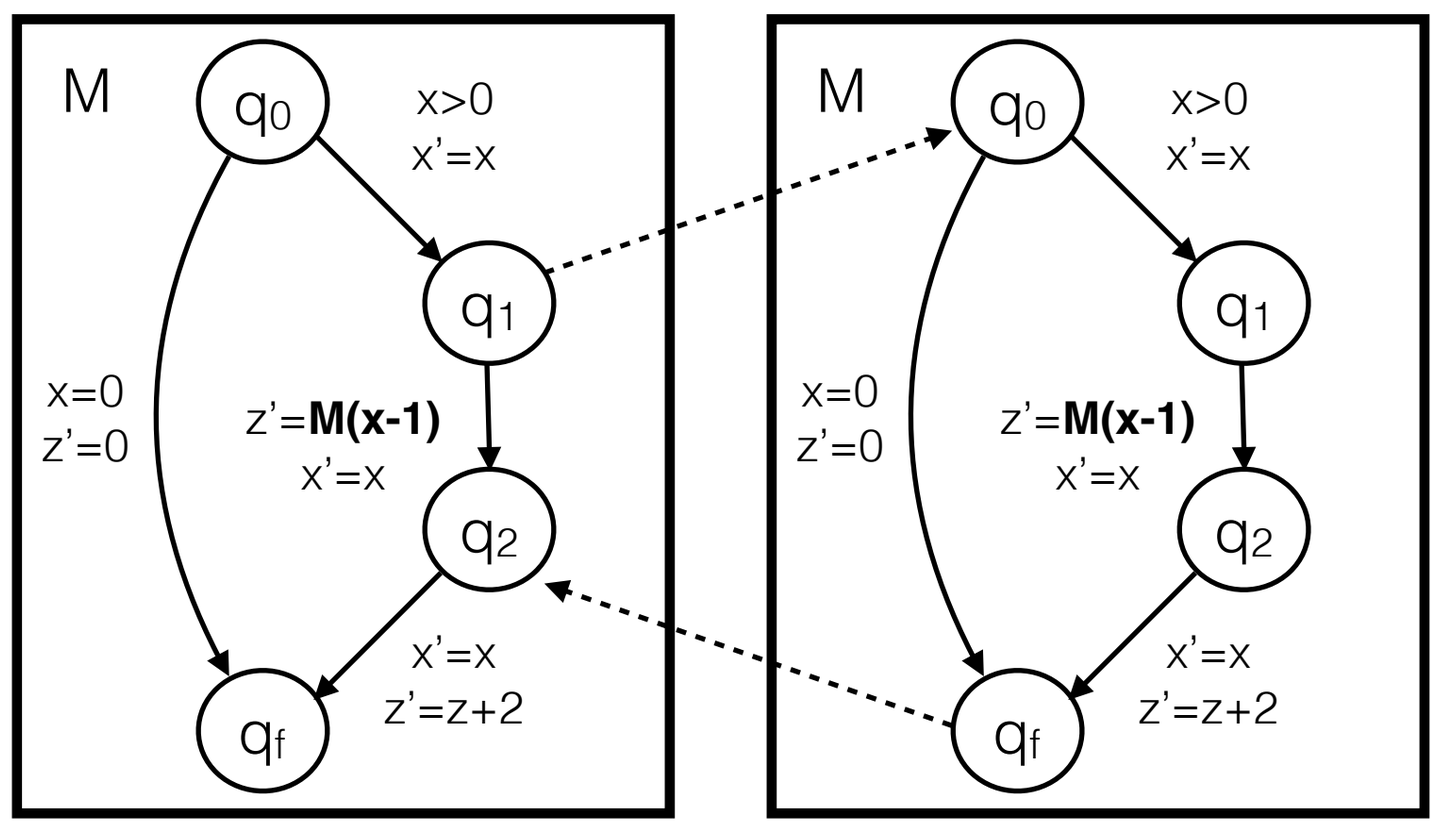


Machines à compteurs

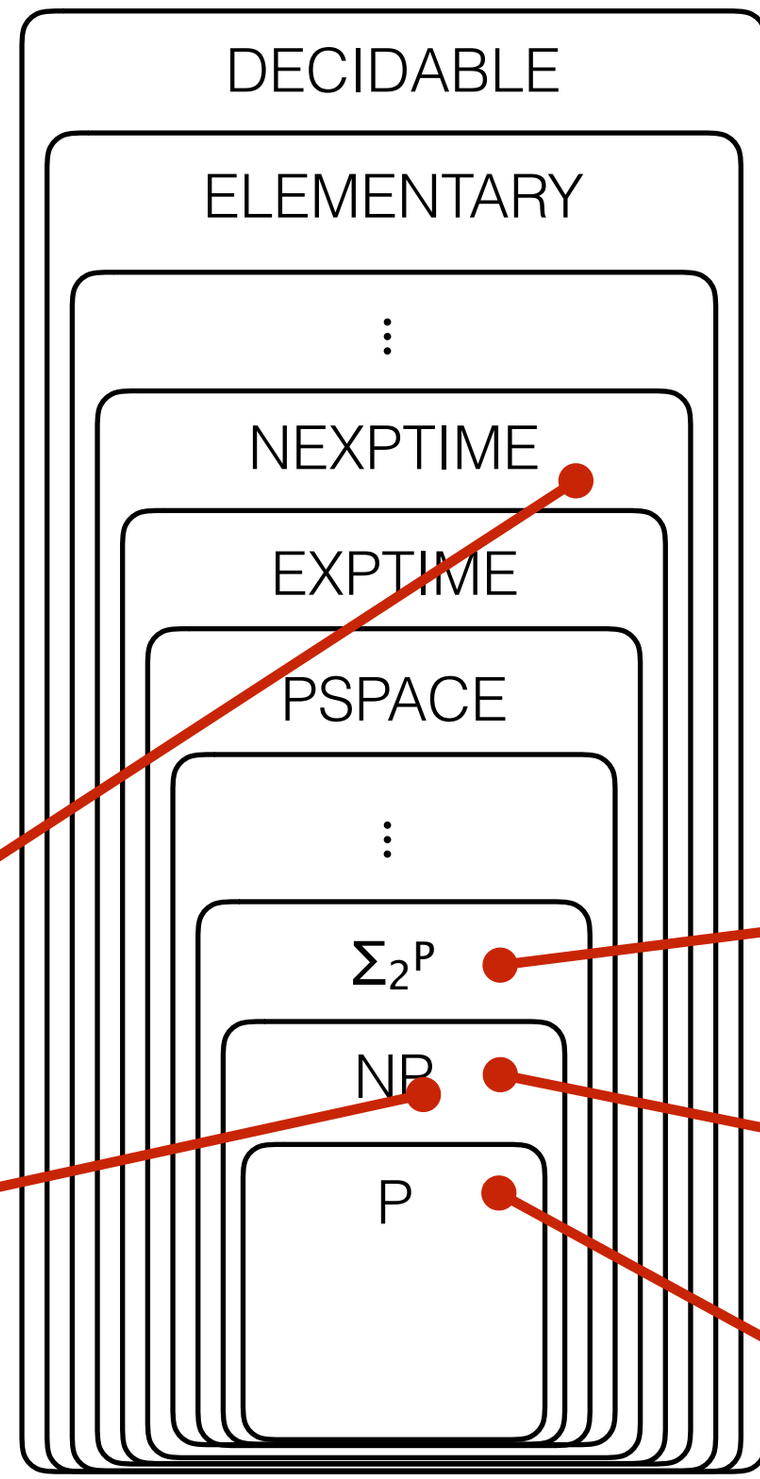
plates



récursives



Algorithmes de vérification



FLATA (F. Konecny, VERIMAG)
https://github.com/filipkonecny/flata

ELDARICA (P. Rümmer, Uppsala)
https://github.com/uuverifiers/eldarica

BOUNDED INTEPROC
OCTAGONAL REACHABILITY
[Ganty, I TACAS'13, FCT'15]

k-BOUNDED INTEPROC
OCTAGONAL REACHABILITY
[Ganty, I TACAS'13, FCT'15]

FINITE MONOID
FLAT REACHABILITY
[Sangnier, I ATVA'16]

OCTAGONAL
FLAT REACHABILITY
[Bozga, I, Konecny VMCAI'14]

OCTAGONAL
FLAT TERMINATION
[Bozga, I, Konecny TACAS'12,LMCS'14]

Automates avec alphabets infinis

Traces d'execution = 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]

⇨

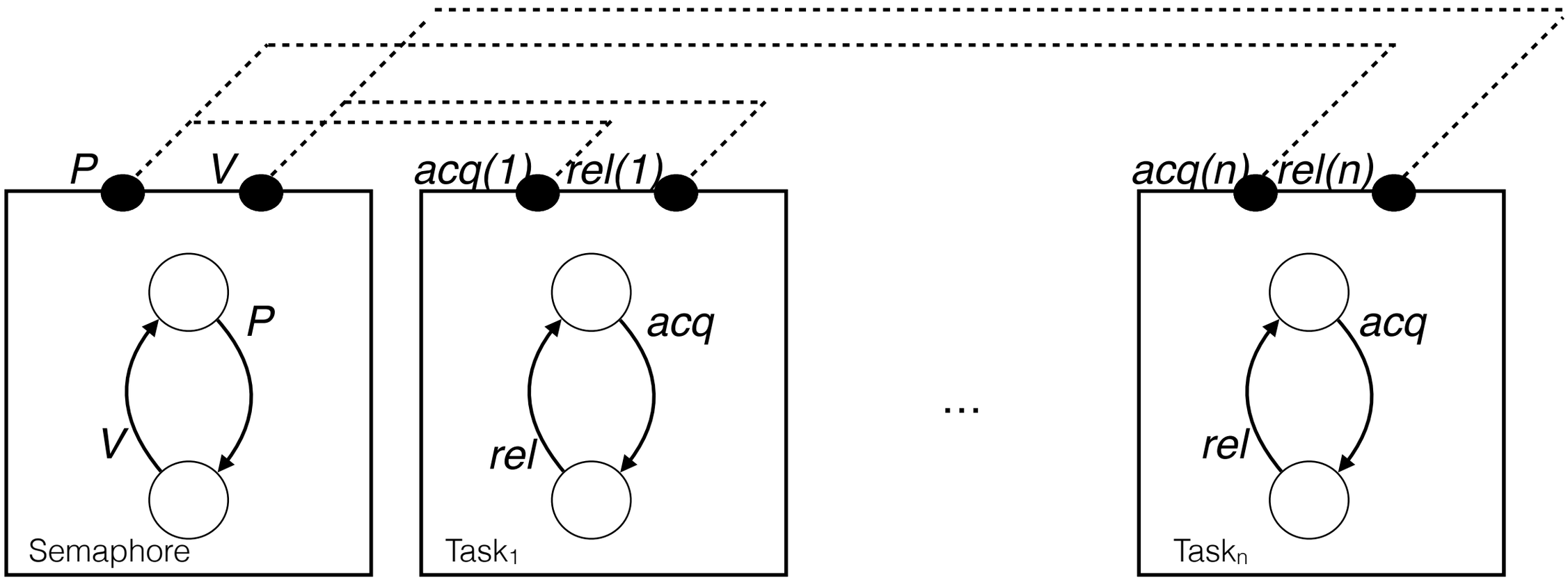
automate alternant de premier ordre

[I, Xu CAV'19]

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 paramètrés



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

Preuves de sûreté pour tout N ≥ 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]

OSTRICH (C. Welzel, TU Munich)
https://gitlab.lrz.de/i7/ostrich/

Logique de séparation

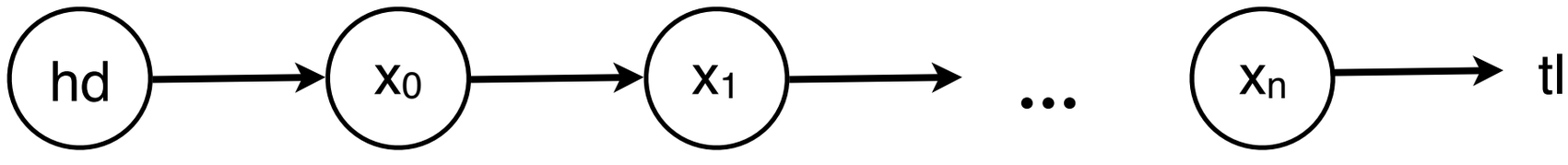
Logique sous-structurelle de premier ordre pour décrire le tas de mémoire [Reynolds'99]

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

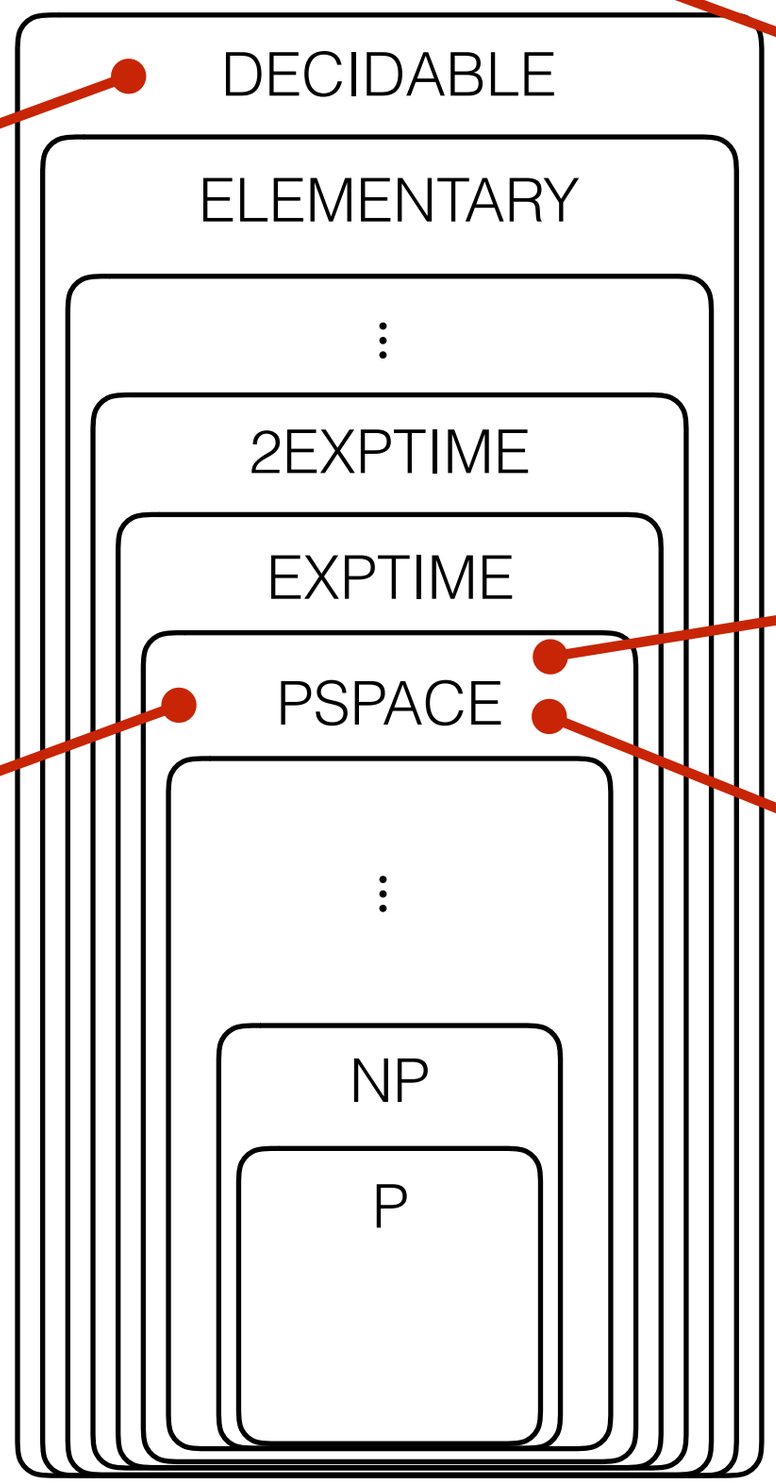
Logique de Hoare basée sur du **raisonnement local** (frame rule)

Définitions inductives pour décrire des structures de données récursives

$$\mathrm{lseg}(hd,tl) \equiv emp \wedge hd=tl \vee \exists x\ .\ hd \rightarrow x * \mathrm{lseg}(x,tl)$$



Algorithmes (premier ordre)



SAT PRENEX(SL^1)

[Echenim, I, Peltier, TABLEAUX'19]

SAT BSR(SL^k)

[Echenim, I, Peltier, FOSSACS'19, TOCL'20]
[I, Reynolds, Serban VMCAI'17]

SAT BSR^{inf}(SL^k)
SAT BSR^{fin}(SL^k)

[Echenim, I, Peltier, FOSSACS'19, TOCL'20]

SAT BSR(SL^1)

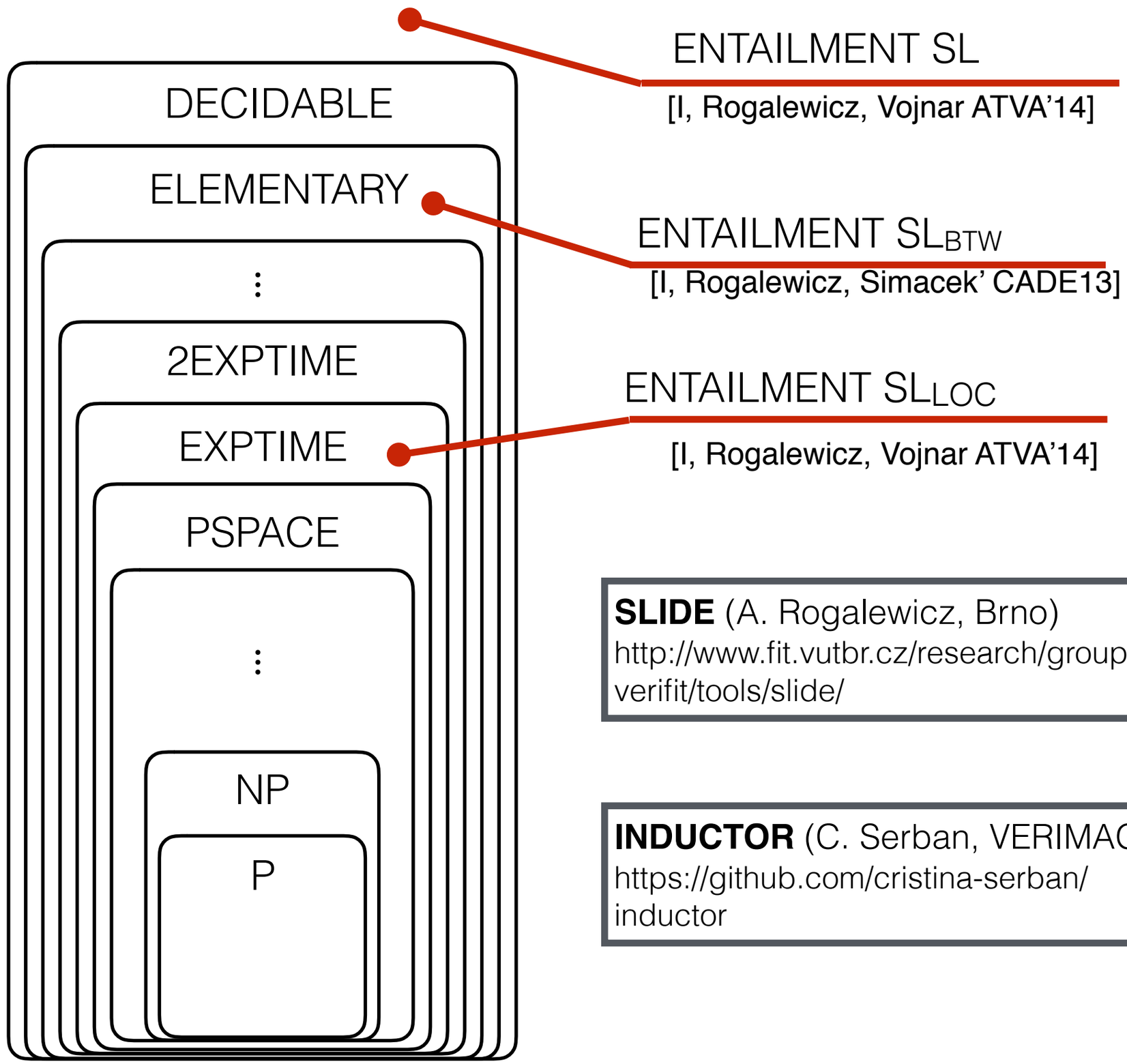
[Echenim, I, Peltier, TABLEAUX'19]

SAT QF(SL)

[I, Reynolds, Serban, King, ATVA'17]

CVC4 (A. Reynolds, Iowa)
https://github.com/CVC4

Algorithmes (ordre supérieur)



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

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

Algorithmes (ordre supérieur)

ENTAILMENT $SL_{PC\text{-}NOT\text{-}E}$

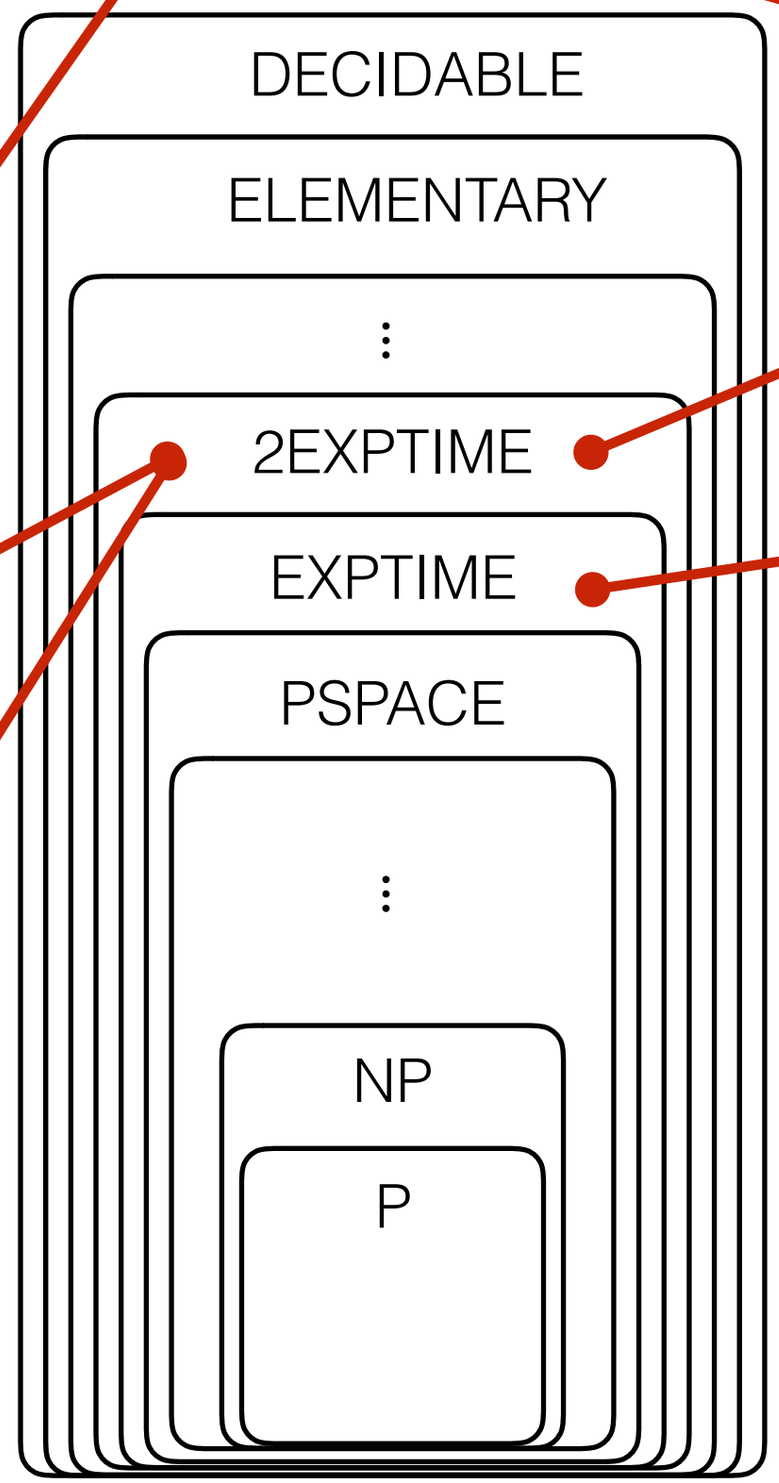
[Echenim, I, Peltier, IPL'21]

ENTAILMENT SL_{SAFE}

[Echenim, I, Peltier, CADE'21]

ENTAILMENT SL_{RESTR}

[Echenim, I, Peltier, CSL'21]
[Echenim, I, Peltier, LPAR'20]



ENTAILMENT SL

[I, Rogalewicz, Vojnar ATVA'14]

ENTAILMENT SL_{BTW}

[I, Rogalewicz, Simacek' CADE13]
[Zuleger, Pagel, LPAR'20]

ENTAILMENT SL_{LOC}

[I, Rogalewicz, Vojnar ATVA'14]

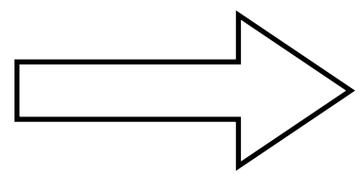
SLIDE (A. Rogalewicz, Brno)
http://www.fit.vutbr.cz/research/groups/verifit/tools/slide/

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

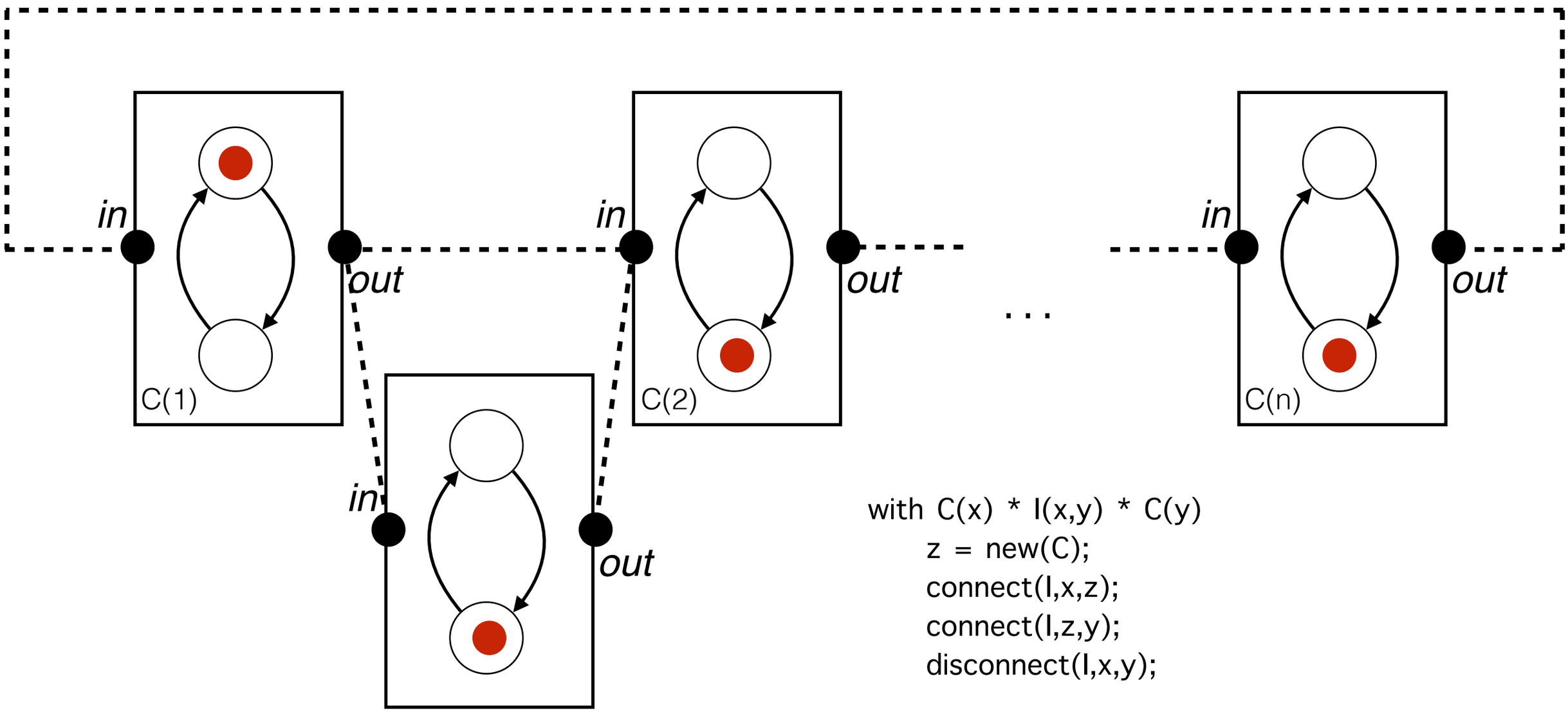
1970 — 2000

2000 — au jour

Modularité: distinction comportement / coordination

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

Systèmes parametrés reconfigurables




```
with C(x) * I(x,y) * C(y)
    z = new(C);
    connect(I,x,z);
    connect(I,z,y);
    disconnect(I,x,y);
```

Systèmes parametrés 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)

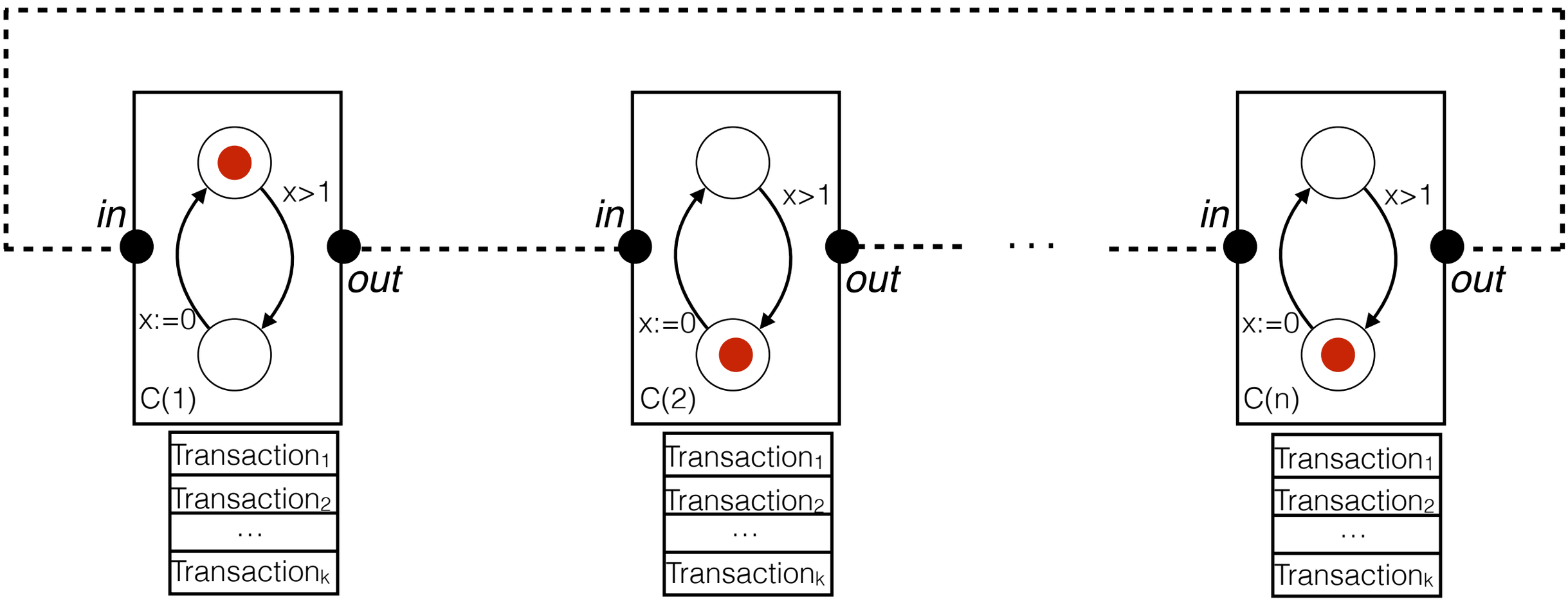
Raisonnement local automatisé:

- preuves en logique de Hoare des sequences de reconfiguration
- procedures de décision pour la nouvelle logique de ressources

ANR NARCO "Non-Aggregative Resource COmposition" (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

Axe VERIMAG/MOHYTOS “MOdular HYbrid and Timed cOmponent-based Systems”
dirigée par T. Dang et R. Iosif

Conclusions

Travaux précédents (2002-2020)

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

Travaux en cours et futurs

- systèmes distribués reconfigurables
- modèles de comportement etendus