VERSYDIS

Vérification des systèmes distribués

- LIAFA, Université Paris 7, CNRS (UMR 7089).

A. Muscholl, W. Zielonka, B. Genest, H. Gimbert, F. Horn, B. Lerman,

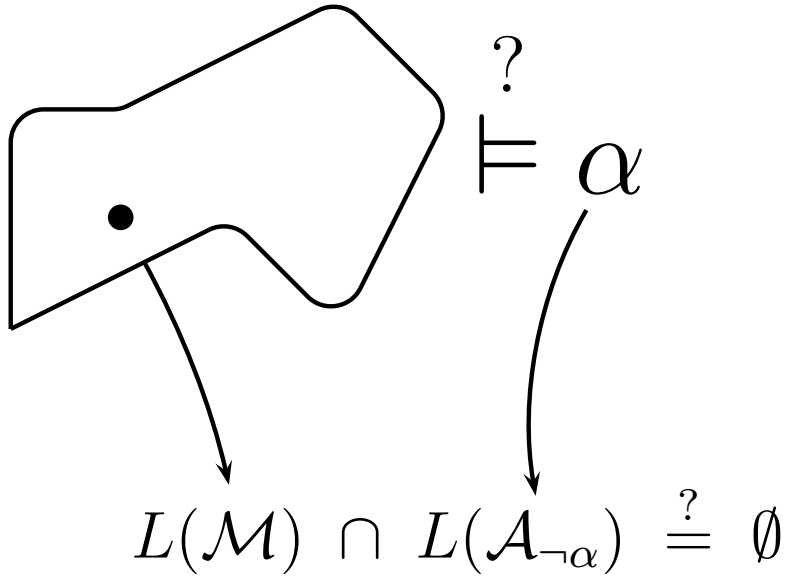
- LaBRI, Université Bordeaux 1, CNRS (UMR 5800).

D. Janin, O. Ly, I. Walukiewicz, P. Weil, M. Zeitoun, J. Bernet, A. Bouquet

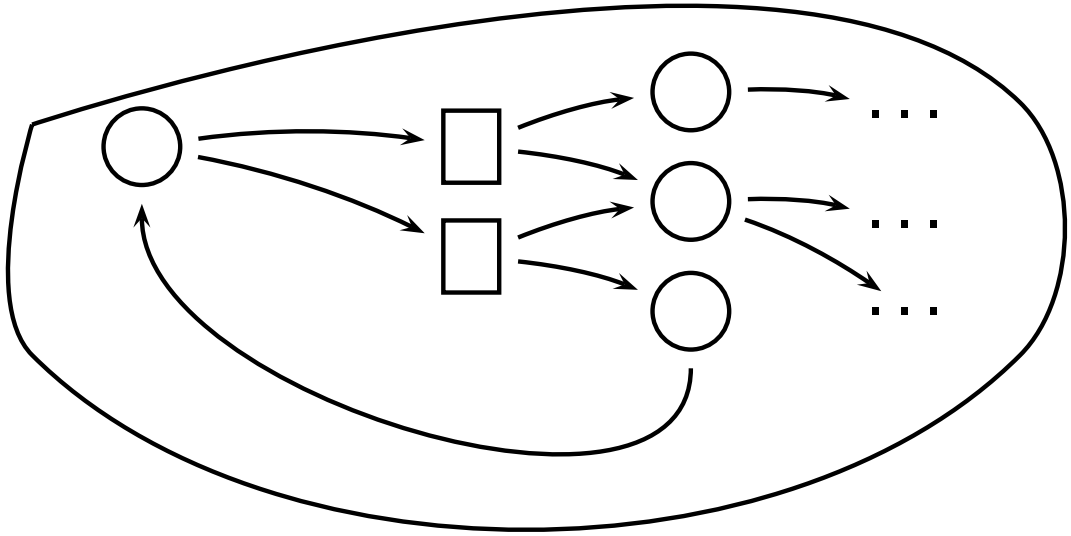
- LSV, ENS Cachan CNRS (UMR 8643)

P. Gastin, N. Sznajder

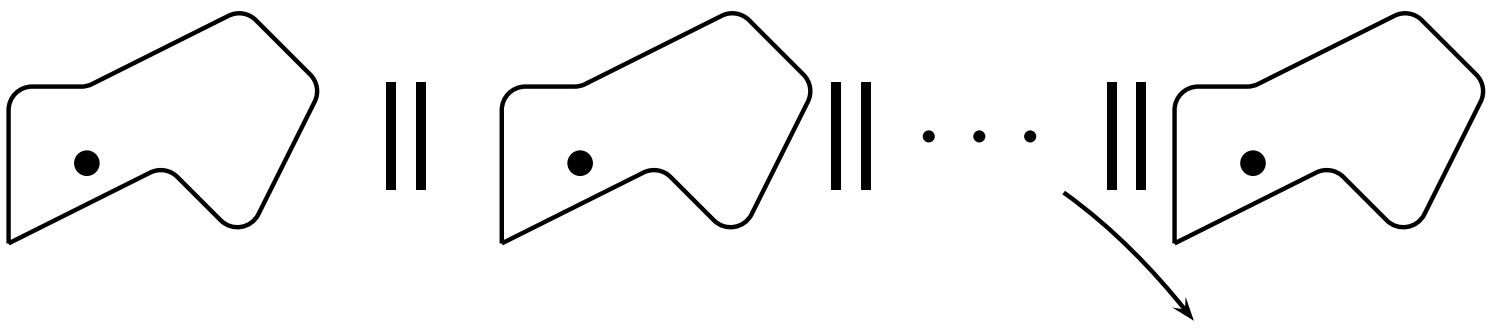
General context



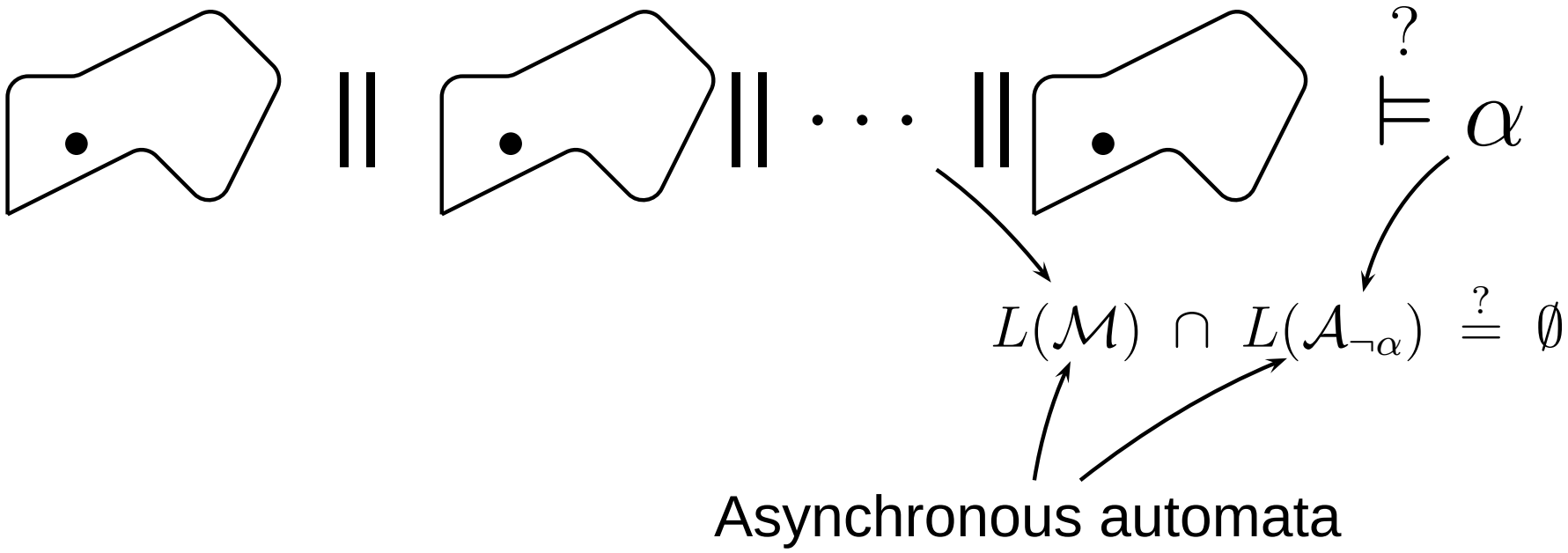
- Emptiness test → solving two player games.



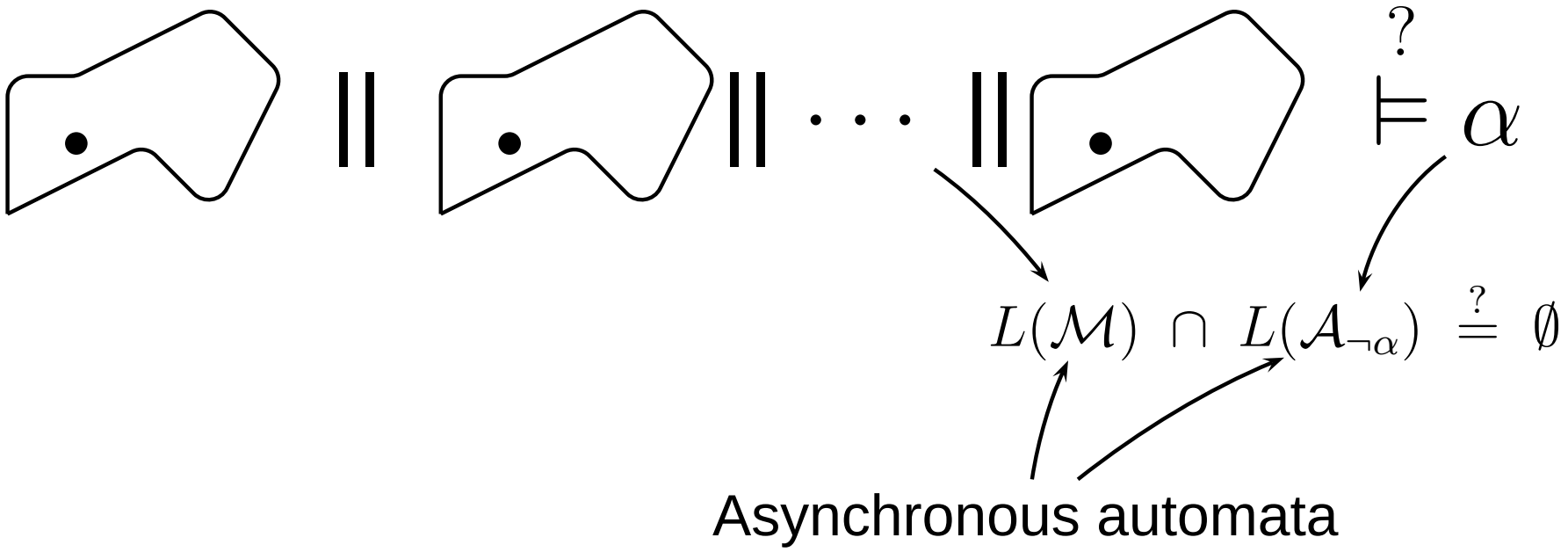
General context



General context



General context

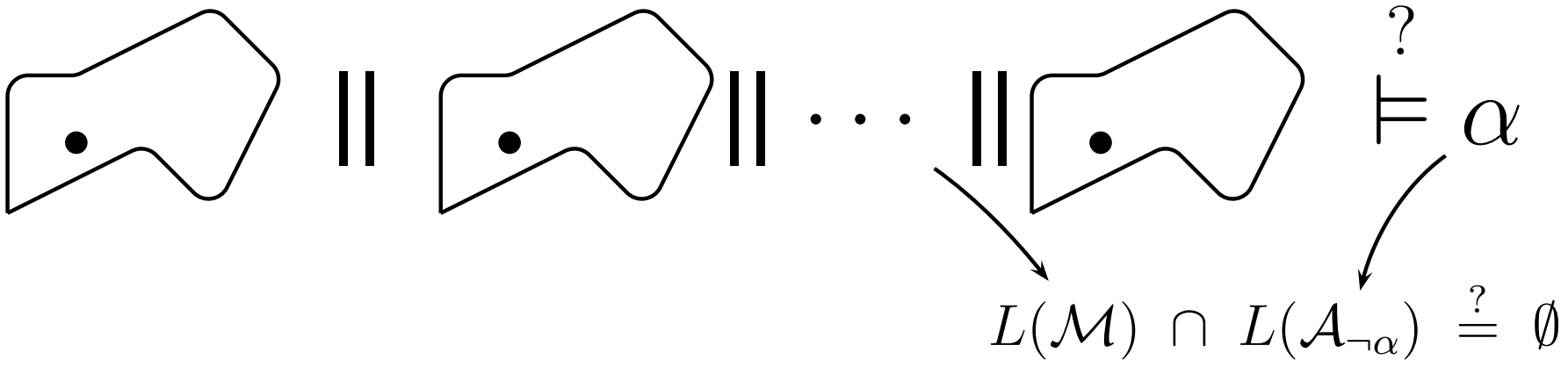


- Size polynomial in the size of the system.
- Problem : emptiness check more difficult.

Thèmes:

- Formalismes de spécification pour les systèmes concurrents.
- Algorithmes et vérification des systèmes concurrents.
- Synthèse de contrôleurs pour les systèmes concurrents
- Jeux distribués.

Two results:



- Understanding of program logics for synchronous communication.

- Understanding of models of asynchronous communication.

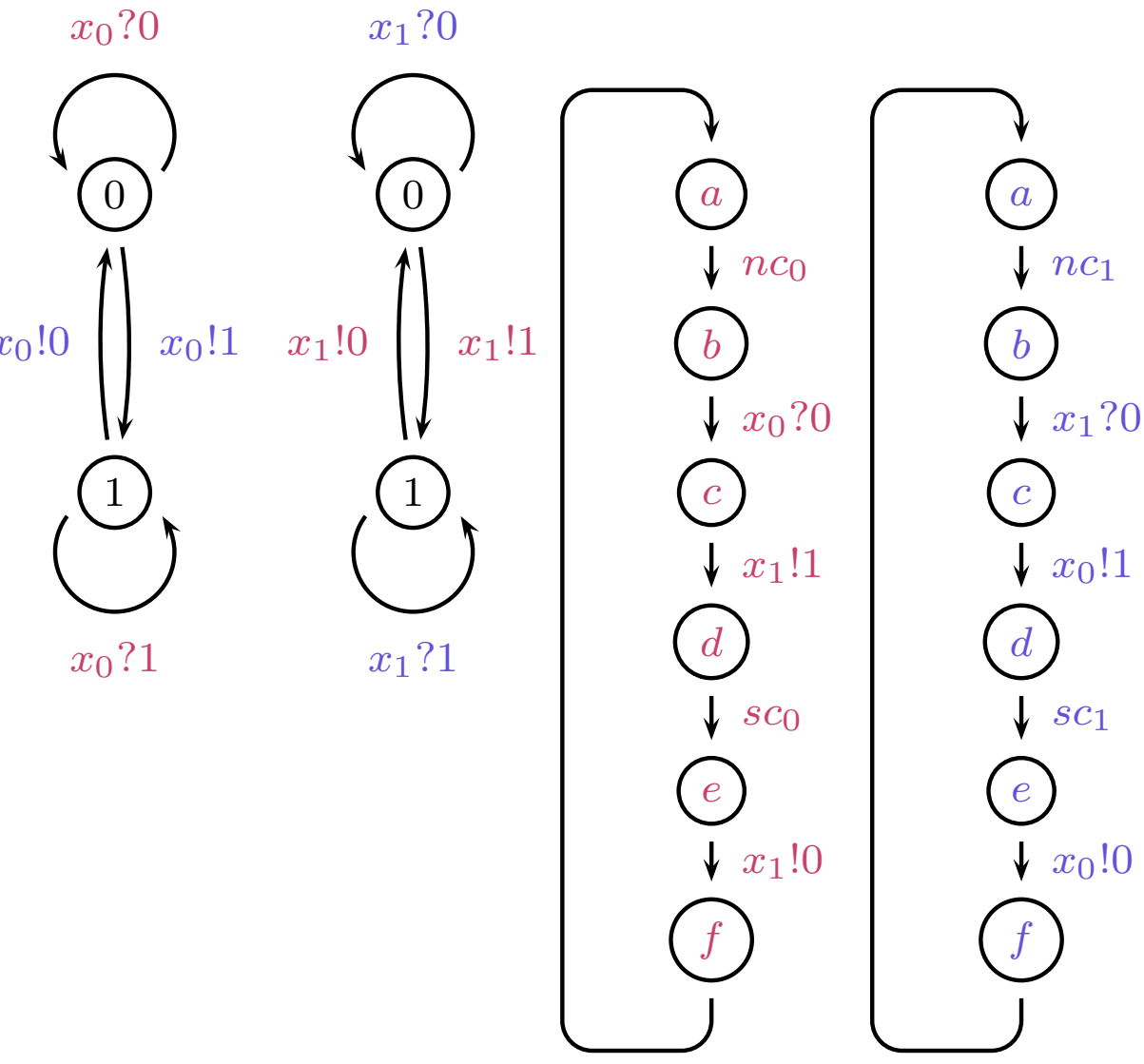
Mutual exclusion

- x0,x1 global variables initialized to 0

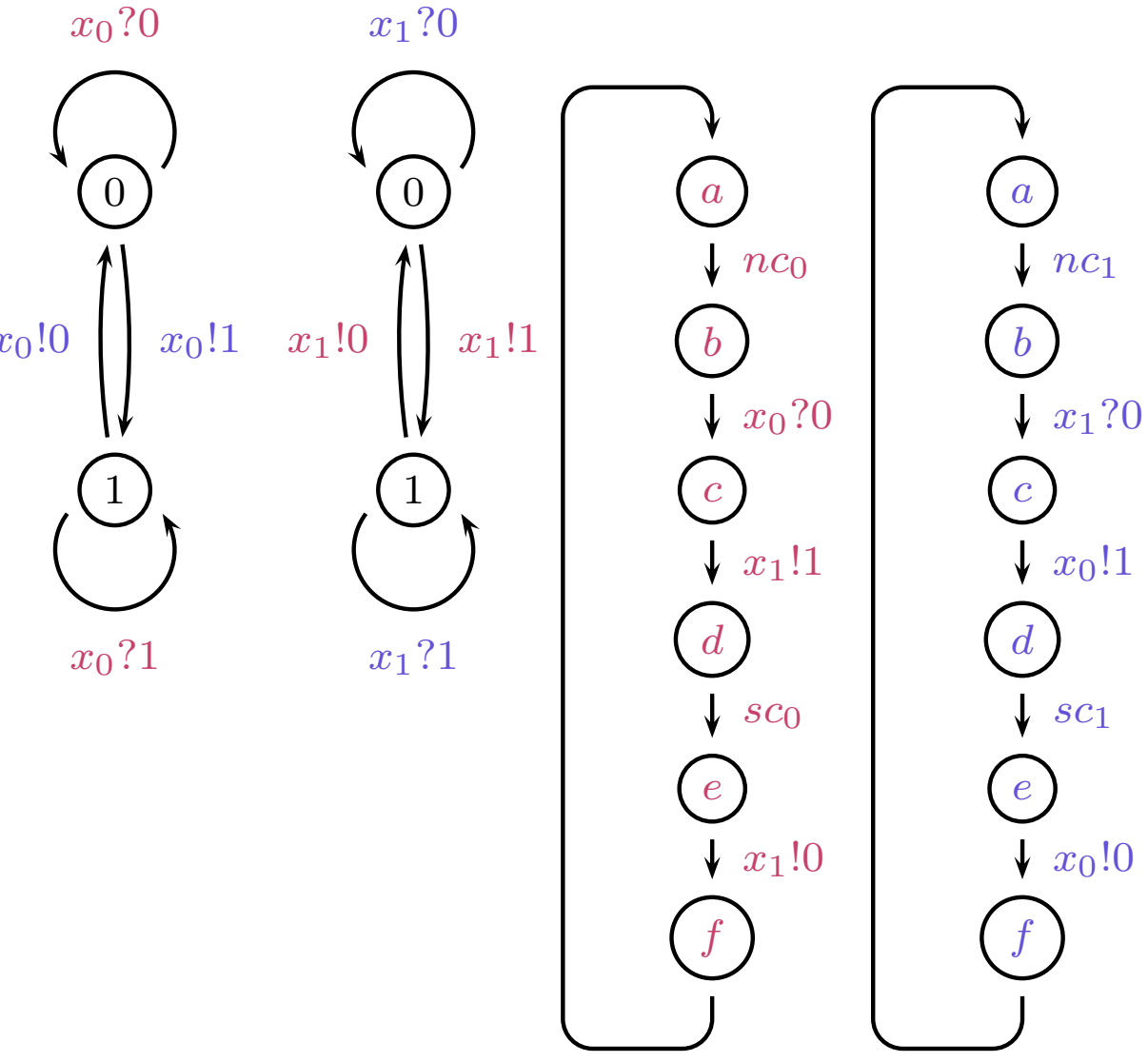
```
  Proc (0) :
  while (true) do
(1)   non-critical section(0) ;          //  nc
(2)   while (  x1 == 1 ) do skip;    // test
(3)    x0 = 1;                        //  req
(4)   critical section(0);           //  cs
(5)    x1 = 0;                        //  nreq
```

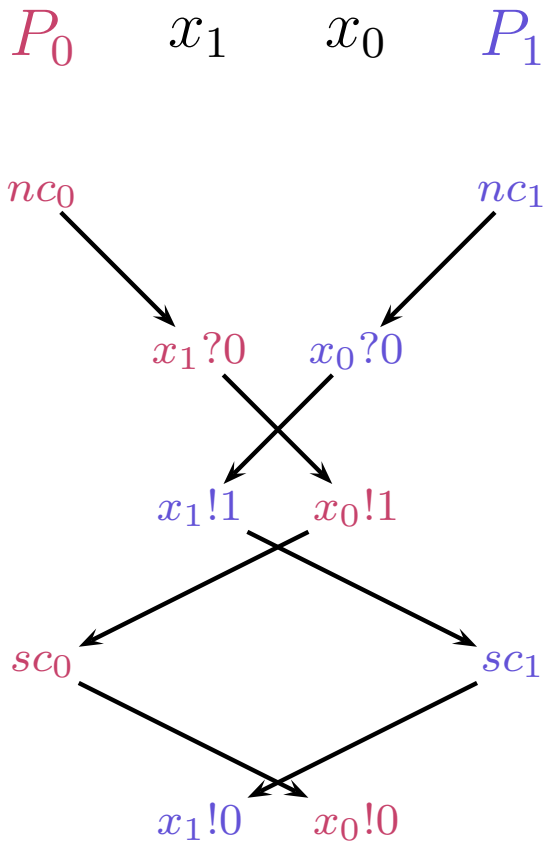
- Question :
  Is mutual exclusion guaranteed in `Proc(0)||Proc(1)`?

Formal model : partial order



Formal model : partial order





- Synchronous product of four automata.

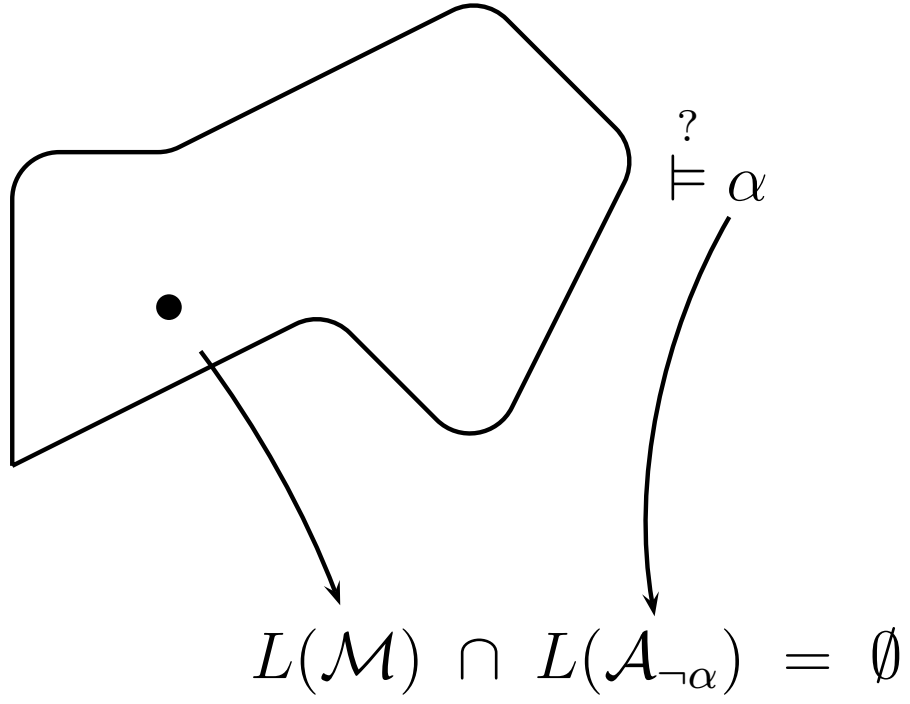
Words and Traces

- Fix a finite alphabet Σ.
- A word is a sequence of letters over Σ, i.e, a linear order with elements labelled by elements of Σ.
- Fix a reflexive and symmetric dependency relation $D \subseteq \Sigma \times \Sigma$.
- A Mazurkiewicz trace is a partial order such that every two elements labelled by dependent letters are comparable.
- In traces we explicitly represent the concurrency information.

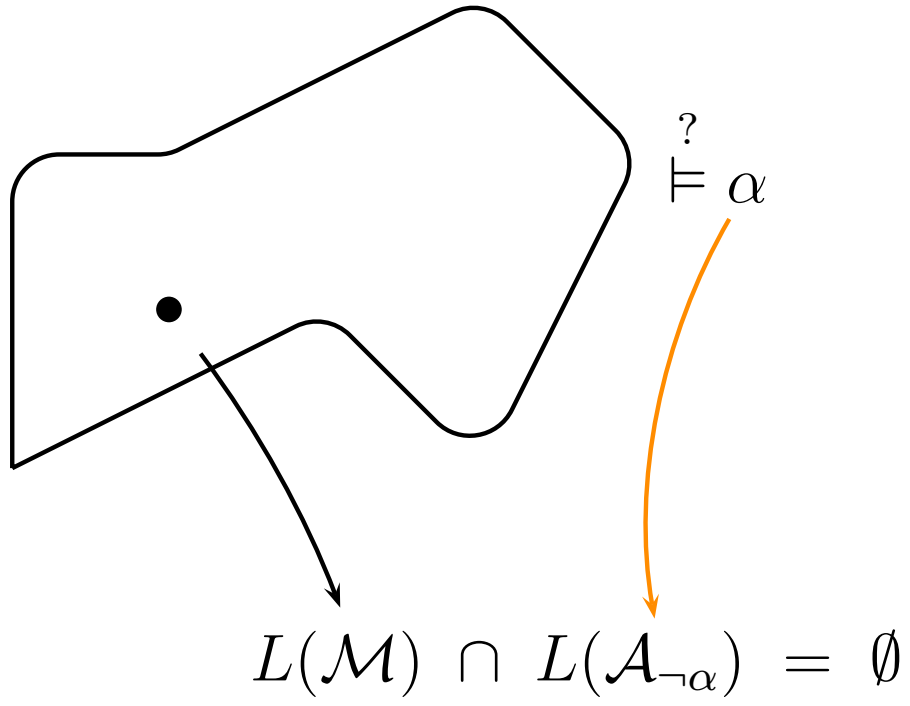
Verification with words

Thm[Büchi]: Regular $\equiv$ MSOL.
Languages definable by finite automata are exactly those definable in Monadic Second-Order Logic over words.

- This is the basis of automata based techniques to model-checking. Properties are translated to automata and verification is reduced to emptiness checking.



Program logics



- Translation MSOL $\rightarrow$ automata results in non-elementary blowup. That is why other logics are used : LTL, mu-calculus.

- First-order logic and monadic second-order logics, serve as expressivity benchmarks for all other logical formalisms.

Verification with traces

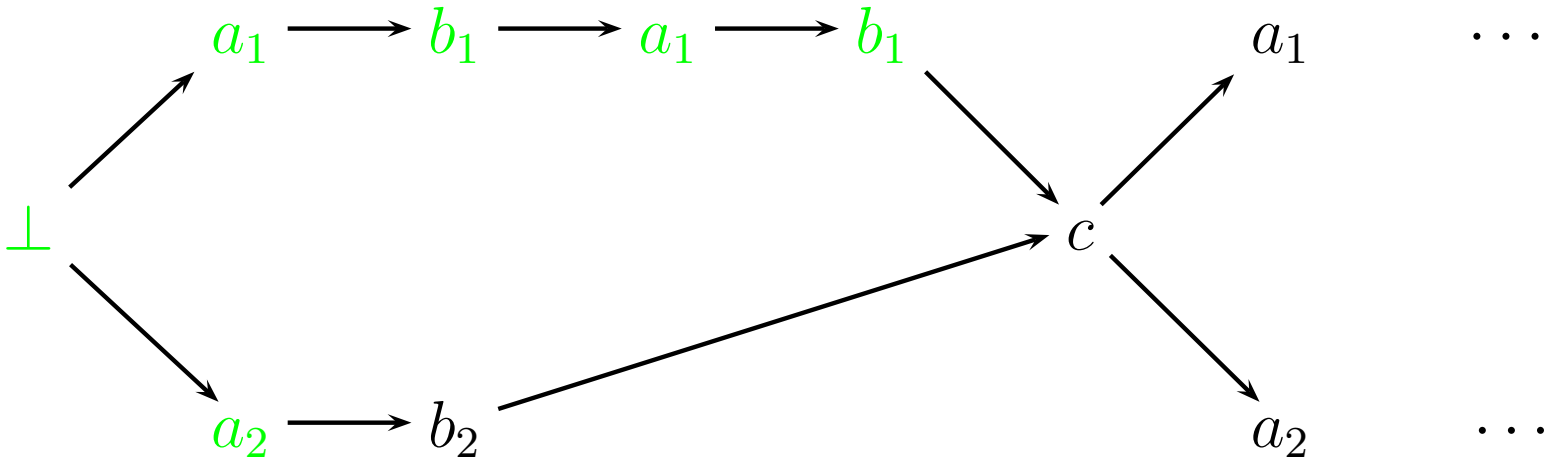
Thm[Zielonka,...]: Regular $\equiv$ MSOL.
Trace languages definable by asynchronous finite automata are exactly those definable in MSOL over traces.

- This theorem can serve as a basis for verification with traces.

- Wanted : logical formalisms with lower complexity than MSOL.

We have made within VERSYDIS an important step in understanding ways of constructing such formalism.

Global vs. Local logics



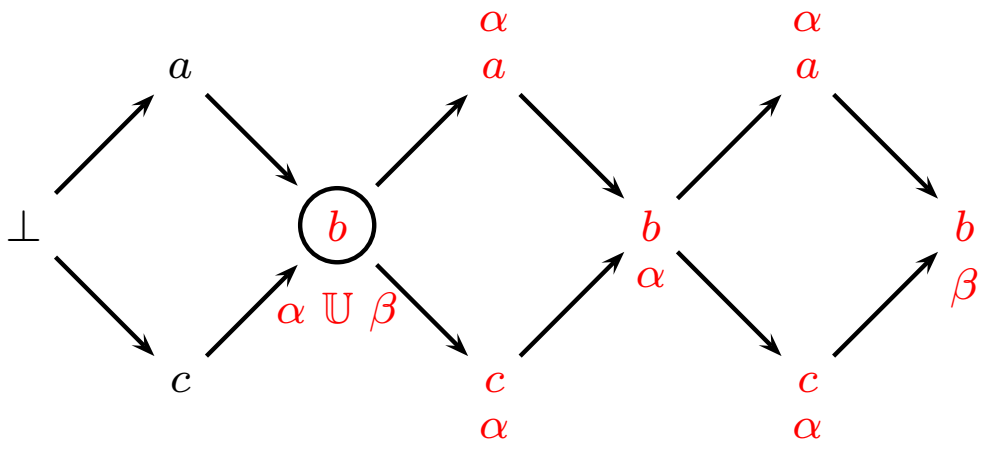
- Configuration in a trace is a downward closed set of events.
- Global logic is evaluated in configurations : $T, C \models \alpha$.
- Local logic is evaluated in events : $T, e \models \alpha$.

Thm[Walukiewicz]: The satisfiability problem for a very simple global logic has non-elementary complexity.

Contribution of VERSYDIS

$$\text{LTL} ::= tt \mid \neg\alpha \mid \alpha \wedge \beta \mid \langle a \rangle \alpha \mid \alpha \ \mathbb{U} \ \beta$$

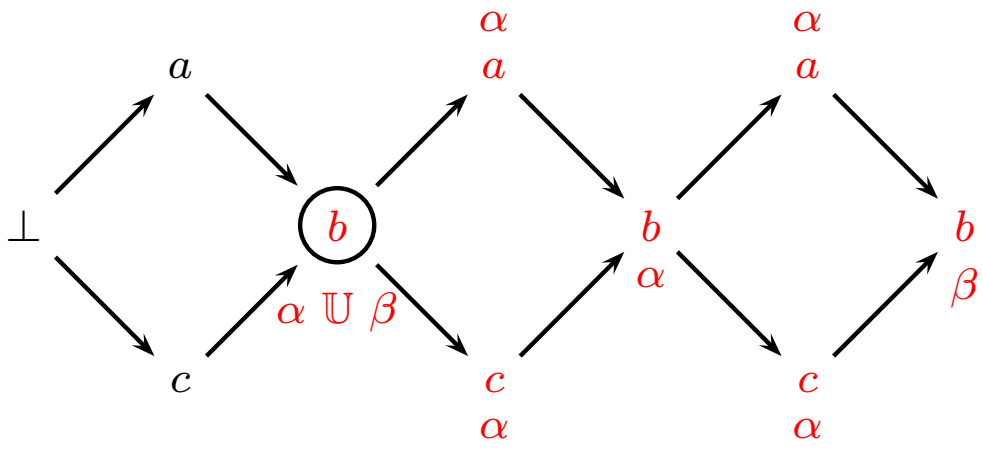
- Semantics of Until



Contribution of VERSYDIS

$$\text{LTL} ::= tt \mid \neg\alpha \mid \alpha \wedge \beta \mid \langle a \rangle \alpha \mid \alpha \; \mathbb{U} \; \beta$$

- Semantics of Until

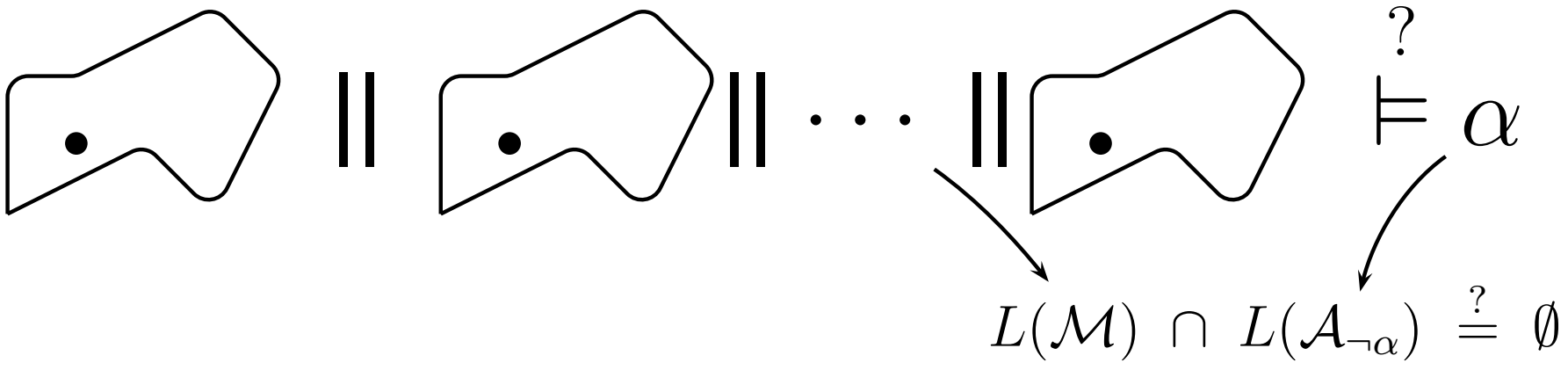


Thm[Diekert & Gastin]: LTL over traces is expressively complete with respect to first-order logic over traces.

Thm[Gastin & Kuske]: Any local trace logic with finite number of operators definable in MSOL is decidable in PSPACE.

- These two theorems give a lower and upper bounds for logical formalisms for traces.

Synchronous communication models



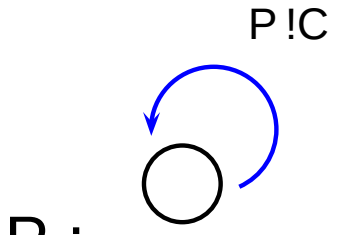
- Experiments with unwinding methods have started.

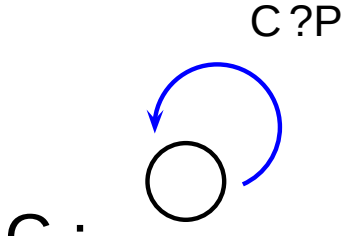
Asynchronous communication models

- Several peers exchange messages through P2P FIFO channels (unbounded).
- Each peer : Finite automaton with send/receive events.
- ITU norm Z 100 (SDL)

Communicating FSM and MSC

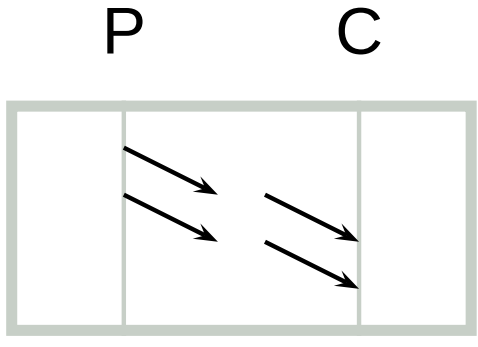
- CFM : Finite automata with send and receive actions :
  send $P!Q(m)$ and receive $P?Q(m)$
- Configuration of CFM : local states plus (unbounded) channel contents.





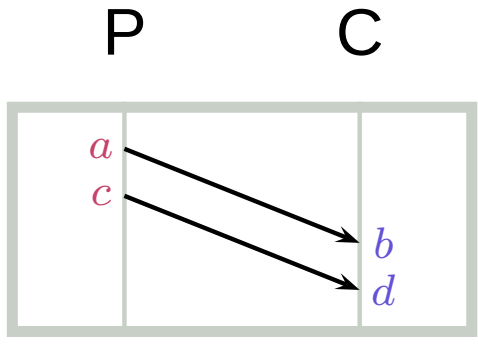
- FIFO channels
- Two equivalent linearizations :

P !C, C ?P, P !C, C ?P
P !C, P !C, C ?P, C ?P



- Message sequence chart (ITU norm Z120)

Message sequence charts (MSC)



- Partial order semantics :
  - Events : a, b, c, d
  - Partial order :
    process order : $a <_P c$, $b <_C d$
    message order $a < c$, $b < d$
- Logic over MSC : talks about process/message orders

Bounds on CFM channels

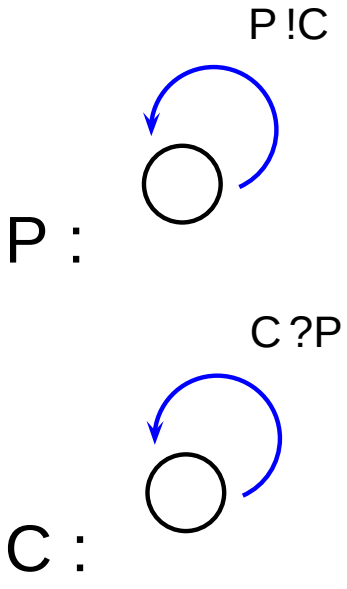
- CFM are Turing powerful. All nontrivial problems about CFMs are undecidable.

- Strong channel bound B :
  in no execution the channel exceeds size B.

- A CFM is strongly bounded iff there is a strong bound for all the channels.

Thm[Mukund et.al.]: Strongly bounded CFM = MSOL over bounded MSC.

Strong bounds : too restrictive



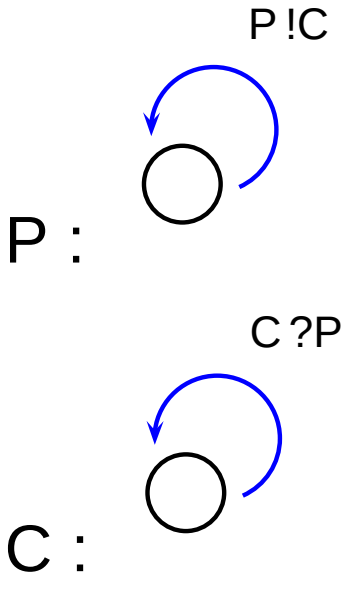
- Producer/consumer : a typical unbounded behaviour.

P !C, . . ., P !C, C ?P, . . ., C ?P

- Strongly bounded CFM are too restrictive :

state space is finite $\longrightarrow$ finite automata

VERSYDIS : Weakly bounded channels



$$\mathbf{R} = (P!C\ C?P)^*$$

is a regular set of representatives

- Every execution has an equivalent one in $\mathbf{R}$.

Def: A CFM is weakly B bounded if there is a bound B such that each execution is equivalent to a B-bounded execution.

- Receiving messages can be scheduled so that channels do not exceed some fixed size.
- Implementations of protocols are usually weakly bounded.

Contribution of VERSYDIS

- Definition of weakly bounded CFM and characterisation à la Büchi.

Thm[Genest, Kuske, Muscholl]: Weakly-bounded CFM $\equiv$ MSOL over w-bounded MSC.

Cor: Model checking for weakly-bounded CFM as easy as for strongly bounded CFM.

- Systems that are not weakly bounded are in some sense unreasonable (wrong use of concurrency).

- It is undecidable if a system is weakly bounded.

- The study of properties of weakly-bounded systems is presented in the thesis of B. Genest.

Other contributions

- Game theory for verification.
- Games for synthesis of distributed systems.
- Algebraic theory of tree languages.

- > 30 publications in journals and conferences.
- A number of invited lectures : LICS, CSL, LPAR, ...
- A number of presentations at summer schools : Ecole de printemps 2004 & 2006, Ecole de jeunes chercheurs, Infinite Games and Applications.

- 1 PhD thesis defended, 1 to be defended in November, 3 close to being finished.