An Asynchronous Game on Distributed Petri Nets

Federica Adobbati Luca Bernardinello Lucia Pomello

Dipartimento di Informatica, Sistemistica e Comunicazione Università degli Studi di Milano-Bicocca viale Sarca 336-U14, I-20126, Milano, Italy

PNSE 2019, Aachen, 24 June 2019

Distributed system

User Controlled Environment

Uncontrolled

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ



◆□ > ◆□ > ◆臣 > ◆臣 > ─ 臣 ─ のへで



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 三臣 - のへで



DEADLOCK

▲□▶ ▲圖▶ ▲臣▶ ▲臣▶ ―臣 … のへで



◆□▶ ◆□▶ ◆臣▶ ◆臣▶ ─臣 ─のへで



◆□ > ◆□ > ◆豆 > ◆豆 > ̄豆 = のへで

Features of the system

- Full observability
- Distributed nets (only local conflicts)



Unfolding

UNF
$$(\Sigma) = (B, E, F, \lambda)$$
,



 $E = E_c \cup E_{nc}$ $\lambda(E_c) = T_c$ $\lambda(E_{nc}) = T_{nc}$

◆□ > ◆□ > ◆豆 > ◆豆 > ̄豆 = のへで



A play is a pair (ρ, π) ρ : NC-weakly fair run π : sequence of B-cuts

Winning condition: a set of runs

Examples

- Liveness conditions:
 - weak observable liveness of a target transition,
 - reachability of a target transition.
- Safeness conditions:
 - avoidance of a target place.

Strategy

 $BCuts(\Sigma)$: B-cuts of the unfolding of Σ

$$\alpha: BCuts(\Sigma) \rightarrow 2^{E_c}$$

 $\mathsf{RM}(\Sigma)$: Reachable markings of Σ

 $\alpha: RM(\Sigma) \rightarrow 2^{T_c}.$

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ

$\alpha-\mathsf{play}$

Let α be a strategy. A play (ρ, π) is an α -play iff:

- Every controllable event in ρ is chosen by the strategy in the cut immediately before its occurrence.
- There is no finally eligible and postponed controllable transition.

Winning strategy

A strategy is winning if the winning condition is satisfied in every $\alpha-{\rm play.}$

Example





Strategy

$$\begin{aligned} &\alpha(\{p_1, p_6\}) = \{t_9\}, \\ &\alpha(\{p_1, p_2\}) = \emptyset, \\ &\alpha(\{p_7, p_9\}) = \{t_7\}, \\ &\alpha(\{p_7, p_{10}\}) = \{t_7\}, \\ &\alpha(\{p_1, p_5\}) = \{t_8\}. \end{aligned}$$

・ロト・西ト・西ト・日・ 日・ シック

- Restriction: no concurrency within components.
- Goal: reachability of a target transition.
- "Lazy" strategy. User waits to observe the Environment moves before taking its own decisions.
- Recursive algorithm on the unfolding, with backtracking.

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ

ExploreNC

From B-cut γ , add uncontrollable events until

- **(**) Deadlock reached (no winning strategy from γ)
- 2 Target fires (Environment tries to backtrack)
- Only controllable events enabled (go to ExploreC)
- Ocycle detected (either no winning strategy or go to ExploreC)

ExploreC

From B-cut $\gamma,$ add a controllable event

• Restart with ExploreNC















Partial observability



◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ

There is no winning strategy for the user.

Implementing the strategy



Adding control places from observable to controllable transitions.

◆□ > ◆□ > ◆臣 > ◆臣 > ─ 臣 ─ のへで

- Partial observability
- Implementing the strategy
- Extension to concurrent Environment and User

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 のへぐ

- Different goals
- Complexity issues

THANK YOU!