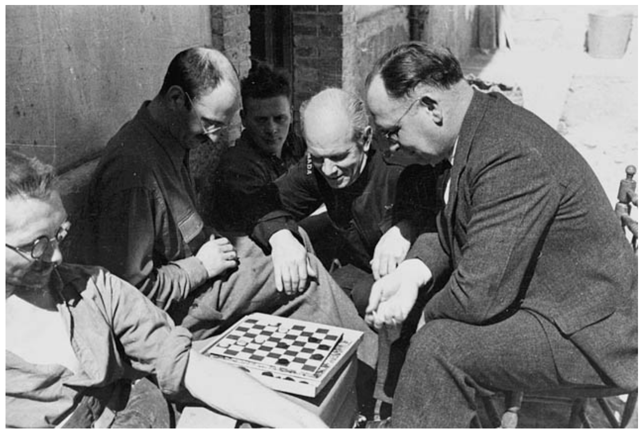
Asynchronous Distributed Stochastic Games

Hugo Gimbert

Tutorial and Workshop on Stochastic Games, IMS, NUS, Singapour

Trading memory for concurrency



Theorem [Zermelo 13]: either white or black has a winning strategy in checkers, or both players can enforce a draw.

Solving checkers using (almost) no memory?English/french/Türing version?

First try

Take two perfectly rational players with unlimited computational power

Let them play on a checker board and check the result.

Memory required? Quadratic number of bits

Second try: Jérôme keeps track of the board.

Actions $A = \{1, \ldots, 8, W, B, \perp, \sharp, +, -, =\}$

-  outputs the initial configuration

$$11W12W13W \cdots 27W28W31\perp 32\perp \ldots 88B\sharp$$

-  moves a pawn:

$$2837\sharp$$

-  $11W12W13W \cdots 28\perp\perp \ldots 37W \ldots 88B\sharp$
-  $7553\sharp$
-  $11W12W13W \cdots 53B \cdots 75\perp \cdots 88B\sharp$
- ...
-  configuration with no B, followed by "$+$" to announce that

How to detect Jérôme cheats?

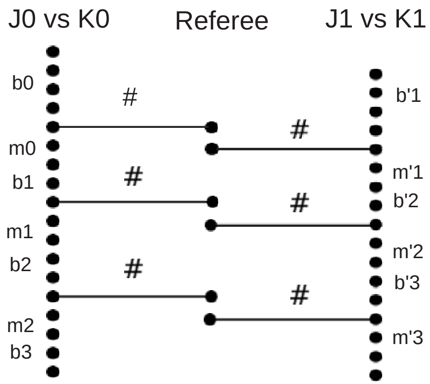
Use a copy-cat

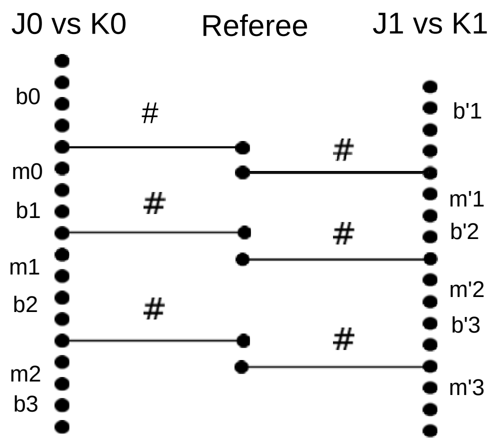
Two teams {J0,J1} vs {K0,K1}

Two deterministic perfect-information stochastic games

J0 vs K0	**J1 vs K1**
	
…	…

boards $(b_0, b_1, \ldots)$ with $b_i \in \{1, \ldots, n, W, B\}^*$
moves $(m_0, m_1, \ldots)$ with $m_i \in \{1, \ldots, n, W, B\}^*$
local actions $A = \{1, \ldots, n, W, B, \perp, +, -, =\}$
shared action $\sharp$

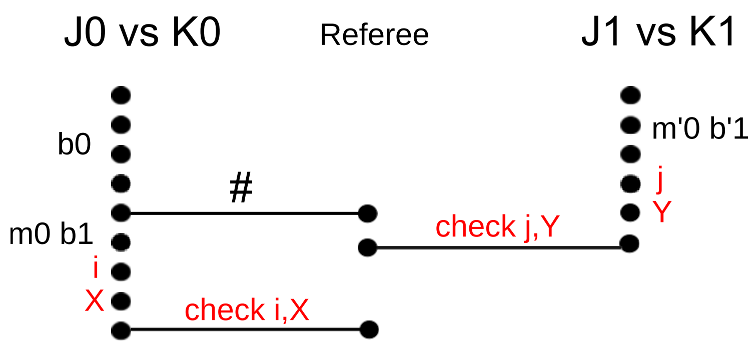




Referee checks that

$$(\forall i, m_i = m'_i) \implies (\forall i, (b_i = b'_i) \wedge (b_i * m_i = b_{i+1}))$$

- The two teams should play the same strategy in both games. Whenever $m_i \neq m'_i$, the faulty team immediately loses.



$$(\forall i, m_i = m'_i) \implies (\forall i, (b_i = b'_i) \wedge (b_i * m_i = b_{i+1}))$$

Referee checks

If $(i = j) \wedge (X \neq Y)$ K-team wins otherwise J-team wins

Only way for the J-team to win for sure $\forall i, b_i = b'_i$

Similar trick to enforce $\forall i, b_i * m_i = b'_{i+1}$

Referee implementable by a finite-state machine with

Defining and solving asynchronous games

Traces

Mazurkiewicz traces [70], Zielonka theorem [87]

Letters $A = \{a, b, c\}$

Independent letters: $a \mathbb{I} b$ can commute

Equivalence relation: $ccabbcabba \equiv ccbabcabba$

Finite Traces: $\{a, b, c\}^*_{/\equiv}$

Infinite traces: $\{a, b, c\}^\omega_{/\equiv}$

$(ab)^\omega = ababab a \cdots$

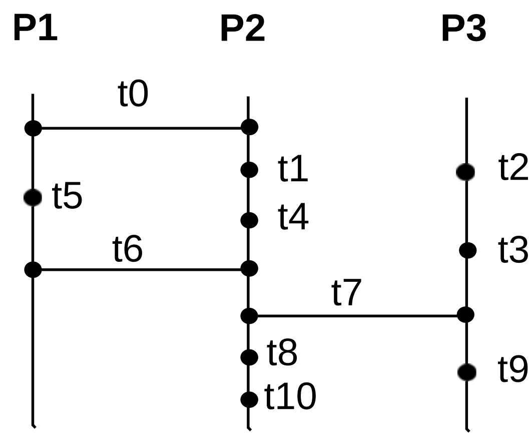
Under some conditions, infinite traces are concatenable

Players $P = \{p_1, p_2, \ldots, p_n\}$
Asynchronous Game $(S_p, i_p, A_p, F_p, \mathcal{T})_{p \in P}$
with states S_p, initial $i_p \in S_p$, target $F_p \subseteq S_p$, actions A_p and $\mathcal{T}$ is the set of transitions
Transition t of domain $Q \subseteq P$ with $Q \neq \emptyset$

$$t \in \Pi_{q \in Q}(S_q \times A_q \times S_q)$$

Transition rule (non-deterministic) t can be executed whenever $\forall q \in Q$, q is in state s_q and plays action a_q
Commutation

$$(t_1 \mathbb{I} t_2) \iff (dom(t_1) \cap dom(t_2) = \emptyset)$$

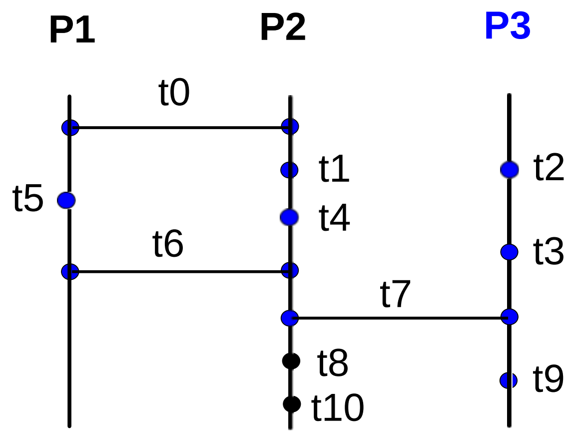


Play $t_0 t_1 \cdots t_9 t_{10}$

$dom(t_0) = \{p_1, p_2\}$ and $dom(t_1) = \{p_2\}$

No global clock, only causality

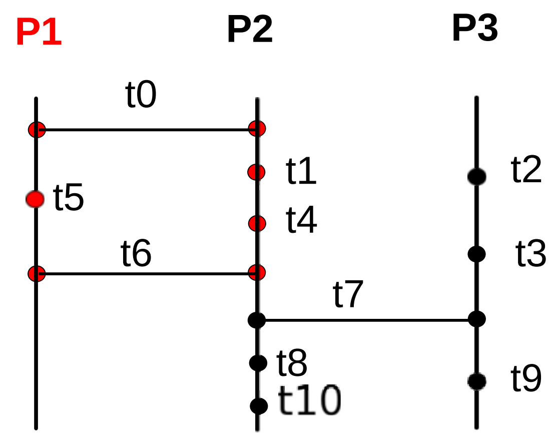
Players **dont communicate with each other** except when playing a shared transition, then all players in the domain of



The view of process p_3

p_3 has learned about t_4 and t_5 from P_2 when t_7 was played

p_3 has no clue (for the moment) about t_8 and t_{10}



The decision of p_1 is indepedent of $\{t_2, t_3, t_7, t_8, t_9\}$

Strategy for p is $\sigma_p : Plays \rightarrow A_p$ such that

$$(view_p(u_1) = view_p(u_2)) \implies (\sigma_p(u_1) = \sigma_p(u_2))$$

Winning strategy:

$(\sigma_p)_{p \in P}$ is winning if all consistent maximal plays

- are infinite (no global deadlock)
- contains infinitely many p-transitions, for each proces p (no local deadlock)
- contains at least one final state F_p for each process p (reachability condition)

Algorithmic problem: given the description of the game, decide if the team of players has a winning strategy.

Determinacy problem: if the answer is no, what can be said? What if players are allowed to randomize (and transitions are

Acyclic games the graph of players connected by their shared actions is acyclic.

Decidability [2013,Genest, G., Muscholl, Walukiewicz] the existence of a winning strategy is decidable in the acyclic case.
Complexity is a tower of exponential
Reduction from halting problem for alternating Turing machines with bounded space

Determinacy[2014, Cheval, Gimbert] two-player asynchronous games have a (computable) value. Extendable to acyclic.

General case: Decidability is an open question. Sufficient