TECHNISCHE UNIVERSITÄT WIEN

GRAN SASSO SCIENCE INSTITUTE
SCHOOL OF ADVANCED STUDIES
Scuola Universitaria Superiore

Emerging Synchrony in Applauding Audiences

Formal Analysis and Specification

Luca Di Stefano, Omar Inverso

REoCAS Colloquium, ISoLA, 29 Oct 2024

The Story So Far

From this... (2019)

PHOTO: AMIRHOSSEIN KHEDRI, UNSPLASH. ANIMATION: DE NICOLA, DI STEFANO, INVERSO (LINK)

The Story So Far

To this… (2022)

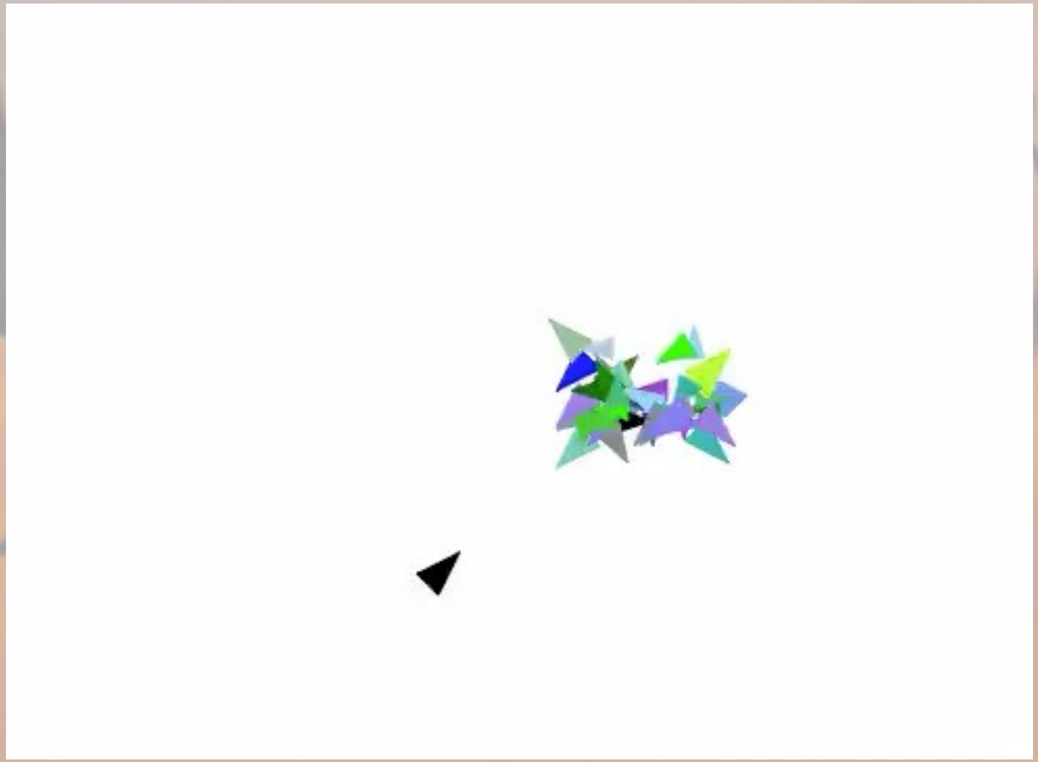


PHOTO: AMIRHOSSEIN KHEDRI, UNSPLASH. ANIMATION: DE NICOLA, DI STEFANO, INVERSO, VALIANI (LINK)

The Story So Far

…And this (2023)

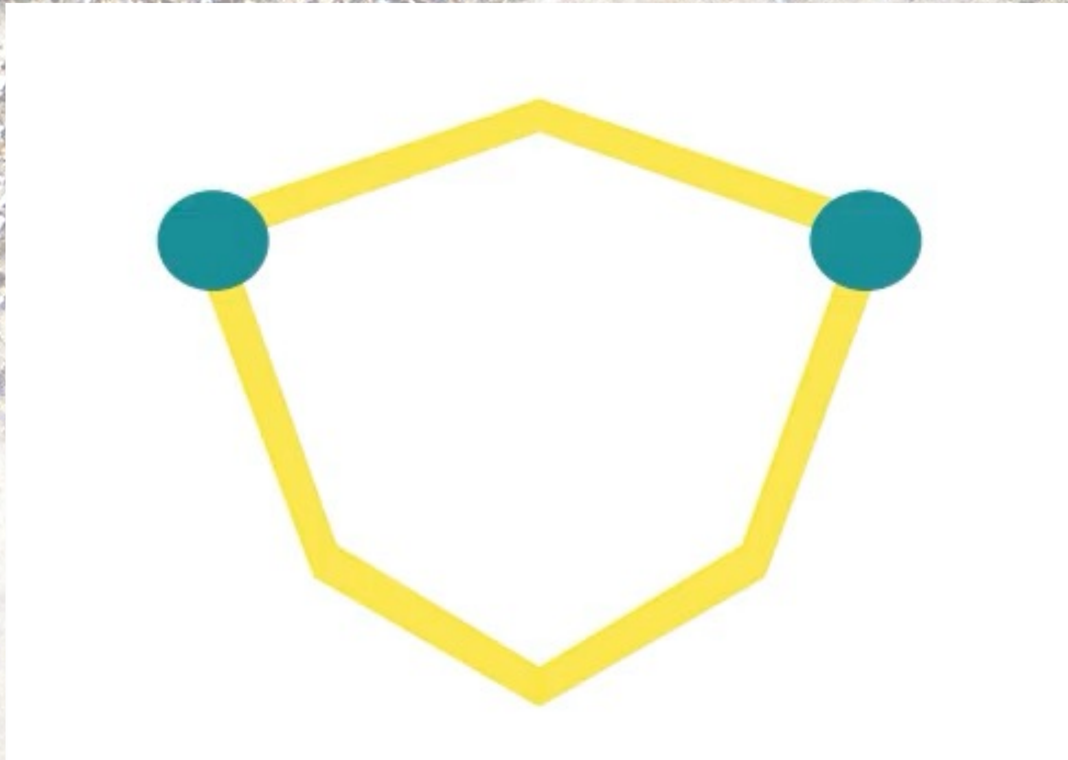


PHOTO: AMIRHOSSEIN KHEDRI, UNSPLASH. ANIMATION: DE NICOLA, DI STEFANO, INVERSO, VALIANI (LINK)

Another Example of Collective Behaviour

SOURCE: HTTPS://WWW.YOUTUBE.COM/WATCH?V=AU5TGPPCPUS

Our Contributions

- Formal specification of a clapping audience
- (Minor extensions to the formal language we used)
- Simulation through an automated workflow
- Verification that convergence is stable (in our model)

Our Specification in a Nutshell

- Each agent claps at its own frequency
- Agents can listen to audience
- When many agents clap at the same moment…
- …Other agents can sense it…
- …and try to synchronise with that collective rhythm

LAbS

- Specification of collective adaptive systems
- Original focus:[1]
  - Virtual stigmergies (replicated key-value stores)
  - Attribute-based Communication
  - Shared memory was also allowed
- More recently: agents observe and react to exposed features (attributes)[3,4]

[1]De Nicola, Di Stefano, Inverso. Multi-agent systems with virtual stigmergy. Sci. Comput. Program. 20202

[2]De Nicola et al. Modelling flocks of birds and colonies of ants from the bottom up. STTT, 2023

[3]De Nicola et al. Intuitive Modelling and Formal Analysis of Collective Behaviour in Foraging Ants. In CMSB’23

Modelling Challenges

- LAbS assumes one action per time step, but we have to model agents that act at the same time (similar to cellular automata)
  1. Restrict interleaving (rounds with 1 action per agent)
  2. Store “intermediate” state updates separately

- Allow an agent to count how many agents of type T satisfy φ (e.g., how many are clapping right now):

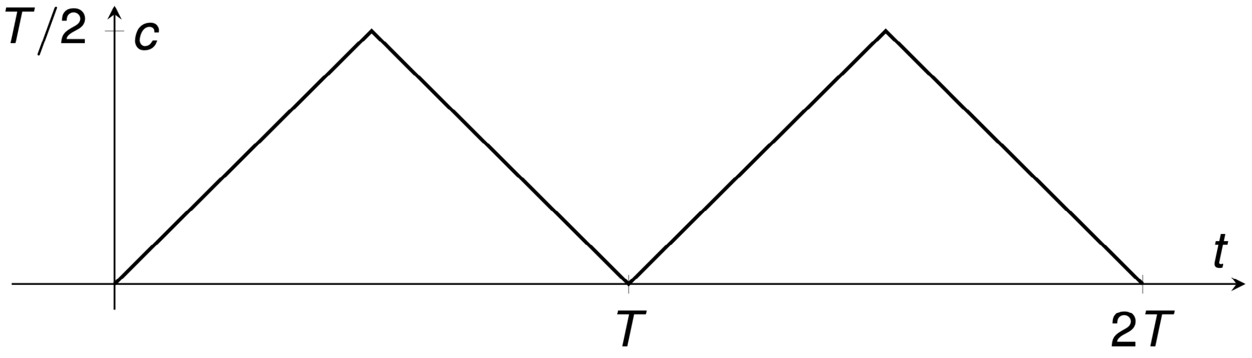
$$result := \mathbf{count}\ \mathrm{T}\ x, \varphi(x)$$

Individual Clapping

- T clapping period
- c counter variable: when 0, the agent is clapping
- $sign$ whether *c* should increase or decrease

Repeat forever:

$$sign := \textbf{if } c = 0 \textbf{ then } 1 \textbf{ else}$$
$$\textbf{if } c = T/2 \textbf{ then } -1 \textbf{ else } sign$$
$$c := c + sign$$



Listening

- Check how many agents are clapping
- Use a threshold value to distinguish loud moments
- Track the time interval θ between loud moments

$$
\begin{aligned}
&audienceClap := \textbf{count}\ \mathrm{Agent}\ i, c_i = 0 \\
&\theta := \ \textbf{if}\ audienceClap \geq loud\ \textbf{then}\ 0\ \textbf{else}\ \theta + 1
\end{aligned}
$$

Adaptation

- After 2 loud moments, agents can compare their own T with the time interval θ between them
- New T = average of old T and θ
- (Bounded by parameters *Tmin*, *Tmax*)

$$T := (T + \theta) / 2$$
$$T := \min(\max(Tmin, T), Tmax)$$

Other adaptation mechanisms (see our paper!)
• Lower loudness threshold if loud moments are too few • Increase loudness threshold if loud moments are many • If $T = \theta$ and agent not in sync, adjust phase

Experimental Setting

	Name	Meaning	Initial value(s)
Variables	T	Clapping period	$Tmin, \dots, Tmax$
	c	Clapping counter	$1, \dots\ T/2$
	$sign$	How c should be updated	1
	$loud$	Loudness threshold	$loud^{(0)}$
	θ	Time interval between loud moments	$-\infty$
Parameters	N	Number of agents	16
	$Tmin$	Minimum value for T	8
	$Tmax$	Maximum value for T	20
	$loud^{(0)}$	Initial loudness threshold	4

SLiVER[1] for SAT-based Simulation

- LAbS specification ⇒ sequential C program[2]
- Use off-the-shelf verification tools for C
- SAT-based BMC + Nondet heuristics + assertion that is violated after *B* steps = random execution traces

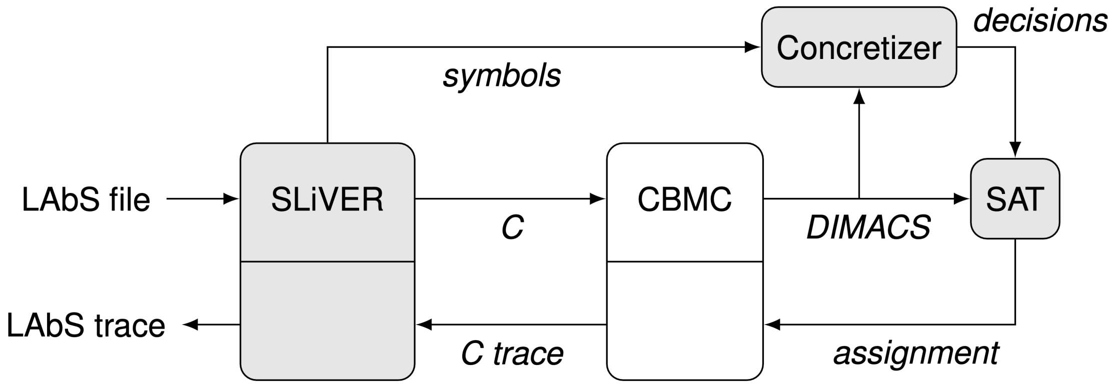
Problem
• Tool encodes nondeterministic state by **symbolic variables** (for verification purposes) • Decision **not efficient**, esp. with nondet heuristics

[1]https://github.com/labs-lang/sliver

[2]Di Stefano, De Nicola, Inverso. Verification of Distributed Systems via Sequential Emulation. TOSEM, 2022

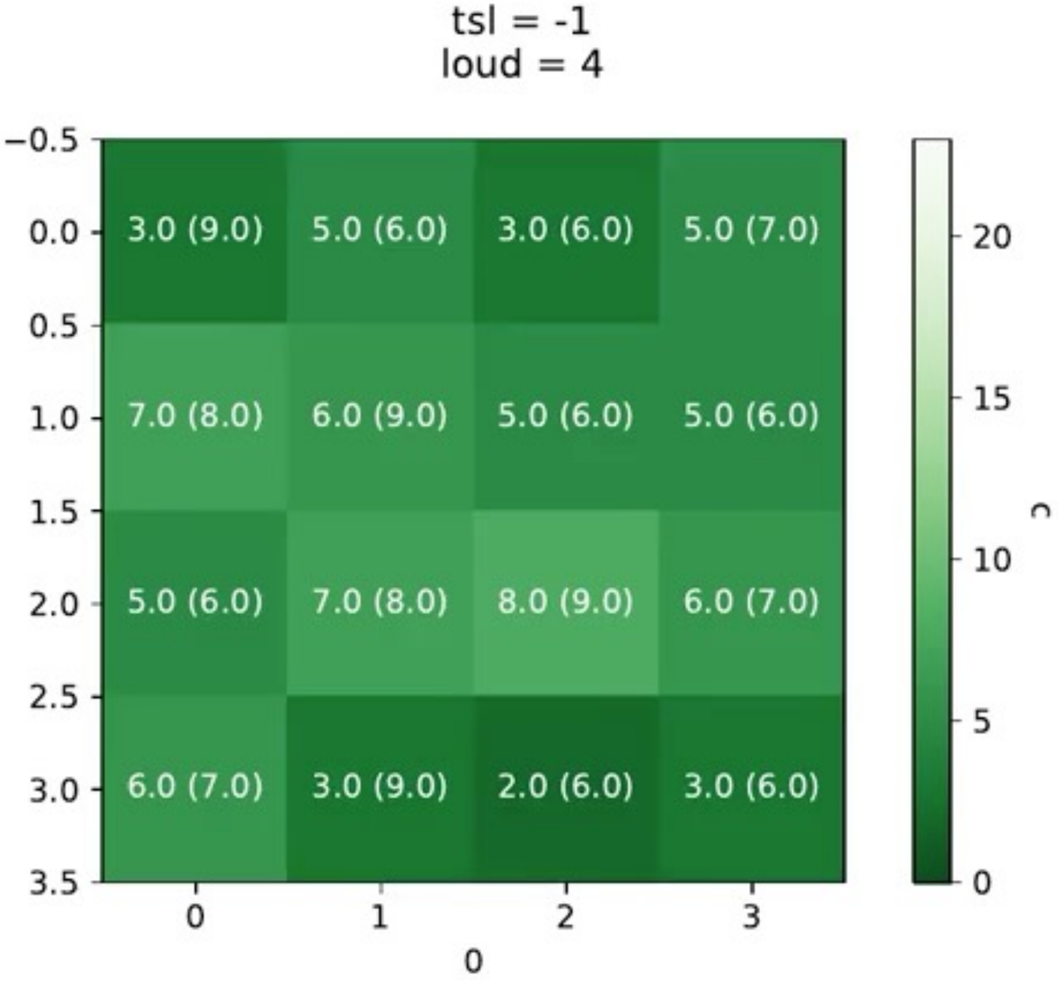
Concretization

- Simulation: no need to explore multiple initial states
- Tell SAT to use one (random) value for each variable
- Still allow SAT to backtrack
- (Applicable to any nondet variable that can be guessed statically)



(Gray=components we built or modified)

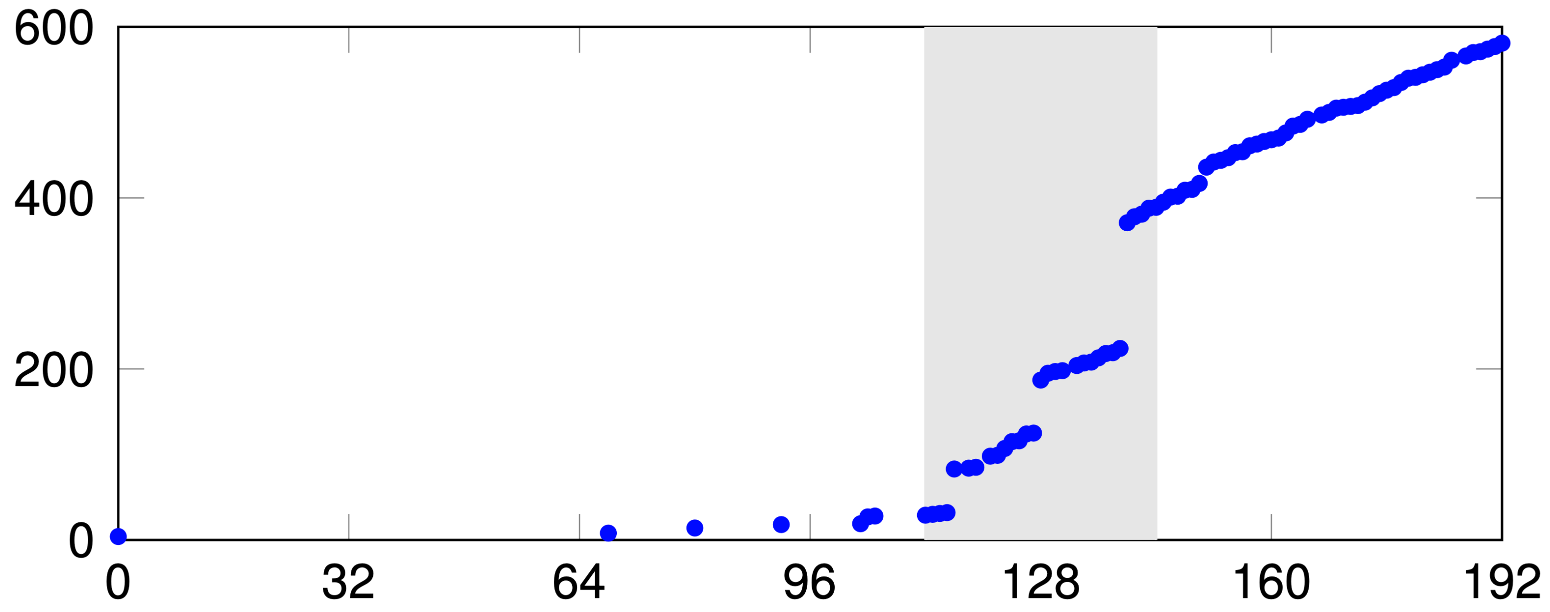
A Sample Simulation



AVAILABLE AT HTTPS://DOI.ORG/10.5281/ZENODO.11374963

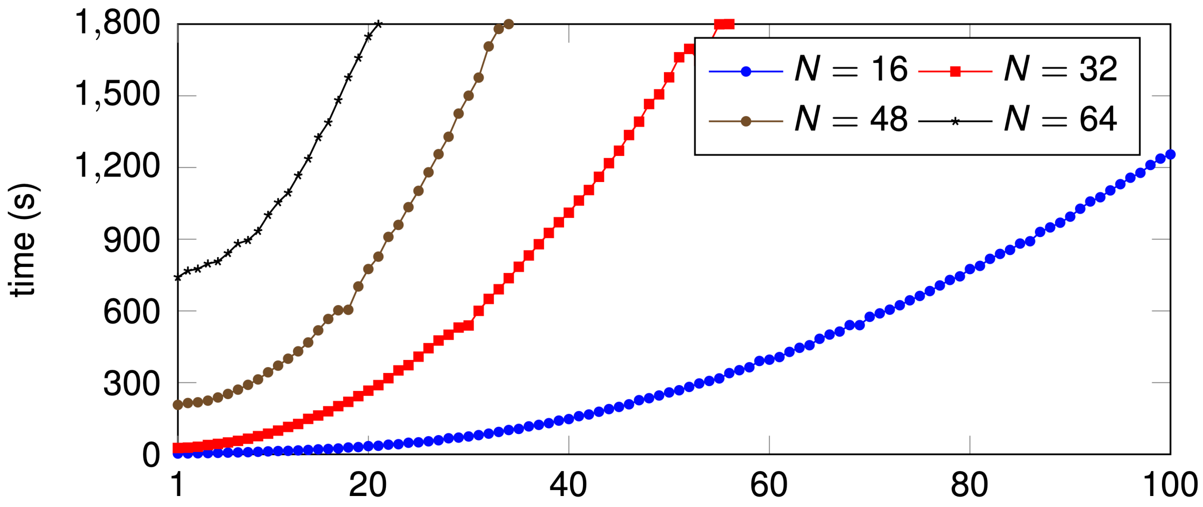
Emergence of Synchronous Applause

- 1000 simulations, 192 steps each
- 58% synchronise before the end of the trace
- “Phase transition” around 128 time steps



Stability of Synchronous Applause

- Can our agent break synchrony after reaching it?
- BMC, increasing bound, assuming audience in synchrony at time 0
- We stop at bound=100 or after timeout (30')
- No violations observed ⇒ stable synchrony



Conclusion

- We presented a first attempt at formal modelling of a clapping audience
- Mix of simulation and verification to analyze its emergent behaviour
- Formal approach to CAS modelling has several benefits
  - High-level formalisms
  - Intuitive specifications
  - Access to efficient analysis techniques

… As pointed out by Rocco in countless occasions ☺
AbC, CARMA, DReAM, KLAIM, SCEL, … (and LAbS!)

Future Work

- Improve simulation performance
- Improve range of supported properties for verification
- Data-driven approaches
  - Given one or more traces of a system S
  - Write a LAbS model M
  - Does M allow (all, most, some) of the traces from S?
- (Do you have interesting case studies? ☺)