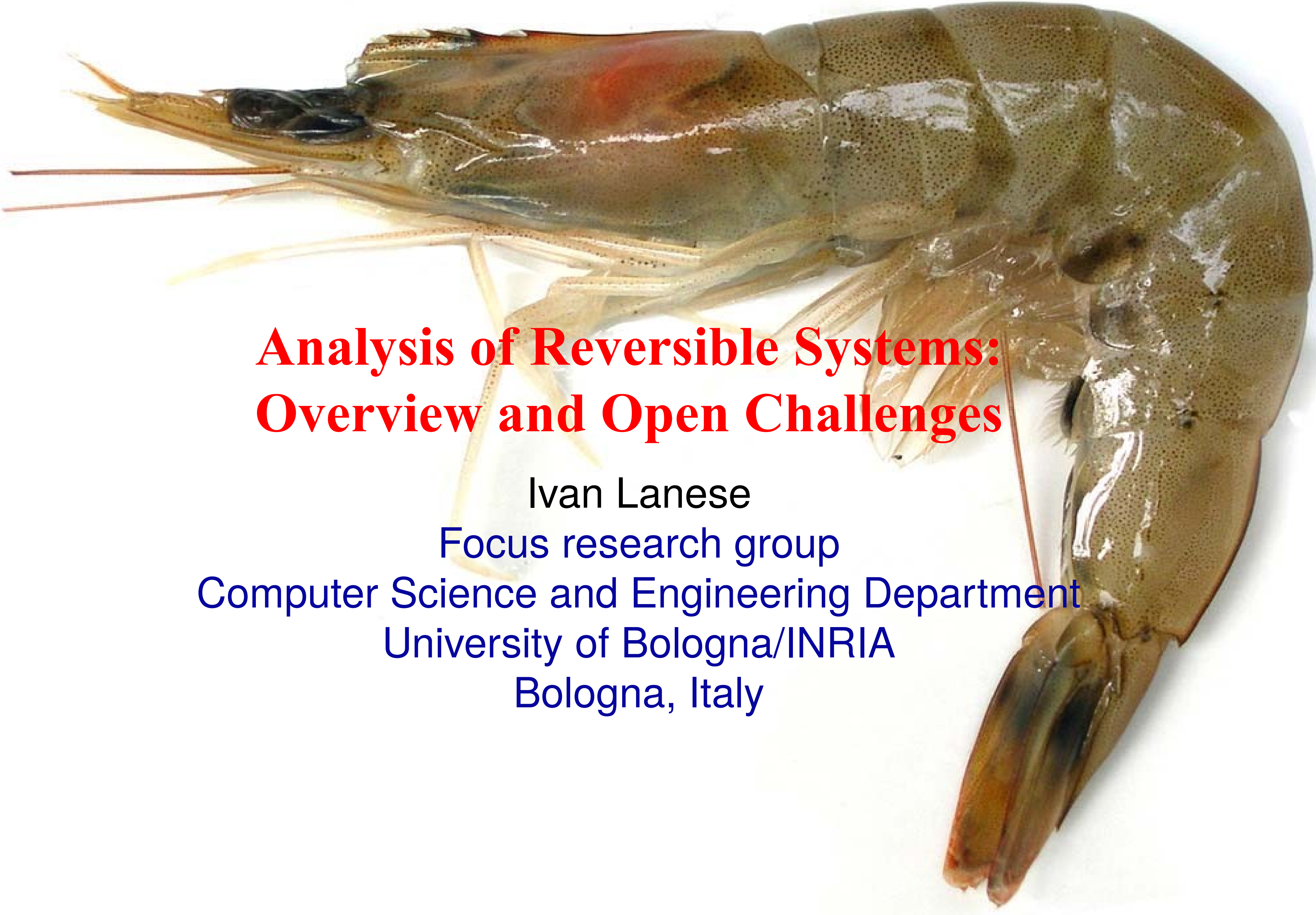

Analysis of Reversible Systems: Overview and Open Challenges

Ivan Lanese
Focus research group
Computer Science and Engineering Department
University of Bologna/INRIA
Bologna, Italy

Disclaimers

- Reversibility comes in different flavors and has different forms
- I will speak about reversibility for concurrent interacting systems, mainly causal-consistent reversibility
- I am not omniscient: please interrupt me if I suggest as future work something you have already done

Analysis

- Given a system, how to (statically) ensure that a given property is satisfied
- A plethora of approaches
  - Behavioral types
  - Behavioral equivalences
  - Model-checking
  - Abstract interpretation
  - ...
- Related to monitoring, debugging and testing
- Relies on logic: if an analysis technique can verify different properties, we need a logic to express them
  - E.g., in model checking

Relevance of analysis

- A main topic of research in computer science
  - At last POPL 42 out of 66 papers were on analysis
- Only a limited amount of works in causal-consistent reversibility
  - Mostly recent
  - A 2014 survey on causal-consistent reversibility contained only 4 papers on analysis

Analysis and reversibility: basic questions

- Can I take my favorite technique and apply it to reversible systems directly?
- Do I need to update the technique to make it applicable to reversible systems?
- Can I exploit reversibility to make the technique simpler/more efficient/more powerful?

- Does reversibility enables/requires new analysis techniques?

Analysis and reversibility

- Reversibility may have a strong impact on analysis
  - In uncontrolled reversibility deadlocks are avoided by construction
  - Deadlock detection is useless
  - Not true with controlled reversibility
- The answer to the questions depends on the specific reversible model
  - Underlying language (CCS, pi-calculus, Erlang, ..,)
  - Control mechanism (no control, roll, ...)
- History and causality information are the main novelty
  - Pro: useful information that may help the analysis
  - Con: it needs to be analysed too

State of the art: bird view

- Logic: a few works exist
- Behavioral equivalences: a few works exist
- Behavioral types: many works exist, both for binary and multiparty interactions
- Program verification, model checking, abstract interpretation: nothing as far as I know
- Debugging: just one work
- Testing: nothing as far as I know
- New techniques: just one work

Logic

- Logics with reverse modalities exist
  [Rocco De Nicola, Frits W. Vaandrager: Three Logics for Branching Bisimulation. LICS 1990]
  [Iain Phillips, Irek Ulidowski: Event Identifier Logic. Mathematical Structures in Computer Science 24(2) (2014)]
- These logics have been mainly applied to forward-only systems
  - To explore their concurrent behavior
- Are them suitable to describe reversible systems?

Which observables?

- What we want to observe in reversible systems?
- Only forward moves
  - reversibility as a hidden mechanism
  - e.g., reversibility for reliability
- Both forward and backward moves
  - reversibility as first-class citizen
  - e.g., modelling of biological systems
- Both make sense, but they are very different
- Impact on the definition of logic, bisimulation, ...

Behavioral equivalences

- Allow one to find when systems A and B are equivalent
  - A should match B moves and viceversa
- Many different equivalences, mainly divided into
  - Bisimulations: consider internal and external moves
  - Barbed equivalences: consider internal moves + barbs
  - Testing equivalences: use a testing process
- Bisimulations and barbed may be strong or weak
  - Weak equivalences allow internal moves to be executed for free
- For reversible systems one mainly has to decide how to match backward moves

Bisimulations for reversible systems

- [R. De Nicola, U. Montanari, F. Vaandrager, Back and forth bisimulations, CONCUR 1990]
  - They force to go back along the same path
  - They say they do not see how allowing to reverse concurrent actions in any order would be useful
- [I. C. C. Phillips, I. Ulidowski: Reversing Algebraic Process Calculi. FoSSaCS 2006]
  - Based on causal-consistent reversibility
  - Distinguishes a.b+b.a from a|b
  - Equivalent to hereditary history-preserving bisimulation (with conditions)
- Both for strong bisimulation

Barbed for reversible systems

- First proposal (strong and weak) in [I. Lanese, C. A. Mezzina, J.-B. Stefani: Reversing Higher-Order Pi. CONCUR 2010]
  - No distinction between forward and backward actions
  - Limited distinguishing power
- Refined (strong) in [I. Lanese, C. A. Mezzina, J.-B. Stefani: Reversibility in the higher-order π-calculus. Theor. Comput. Sci. 625 (2016)]
  - Forward actions matched by forward actions
  - Backward actions matched by backward actions

Testing for reversible systems

- No published study as far as I know
  - Some ongoing work, see Irek’s talk
- Closest work uses total order of actions [C. A. Mezzina, V. Koutavas: A Safety and Liveness Theory for Total Reversibility. TASE 2017]
- Related work on interacting transactions in [E. de Vries, V. Koutavas, M. Hennessy: Liveness of Communicating Transactions. APLAS 2010]
- Both can be seen as special cases of causal-consistent reversibility

Open questions: axiomatizations

- Enable to use axiomatic reasoning to prove equivalences
- Complete axiomatizations are hard
- Partial axiomatization are also interesting
  - Tool to prove equivalences
  - Tool to compare equivalence relations
- For uncontrolled reversibility one may go through the characterization in terms of hereditary history-preserving bisimulation
  - (Not sure whether it has been axiomatized)
- What about controlled reversibility?

Open questions: axiomatizations and history

- Frequently history and processes are partially redundant
  $[k:a\langle P\rangle | k':a(X) \triangleright (X | b\langle Q\rangle); k''] \| k'':(a\langle P\rangle | b\langle Q\rangle)$
- If we change the process without changing the history we get inconsistent processes
- Should we
  - live with inconsistent processes?
  - change history and processes in a coherent way?
  - avoid redundancy (as in CCSk)?

Open questions: weak equivalences

- We probably want
  - forward steps matched by forward steps
- We may want or not
  - backward steps matched by backward steps
- Should we allow auxiliary backward steps when matching forward steps/barbs and viceversa?
- Different possible combinations
  - Do they produce the same equivalences?
  - If not, which axioms distinguish them?

Behavioral types and contracts

- Behavioral types describe the flow of communication

  $(?\,\mathrm{Int}\,.\,!\,\mathit{String}\,.\,\mathit{end}) + (?\,\mathit{Bool}\,.\,?\,\mathrm{Int}\,.\,!\,\mathrm{Int}\,.\,\mathit{end})$
- Ensure absence of communication errors
- Binary communications: ensuring compliance between a client and a server
- Multiparty communications: avoiding deadlocks and communication races
- Behavioral contracts: essentially behavioral types without saying which is the typed program

Reversible behavioral types: binary case

- [F. Tiezzi, N. Yoshida: Reversible session-based pi-calculus. JLAMP 84(5), 2015]
  - Use standard binary behavioral types in a session-based pi-calculus
- [C. A. Mezzina, J. A. Pérez: Reversibility in session-based concurrency: A fresh look. JLAMP 90, 2017]
  - Session types used as monitors
  - Types include modalities limiting the number of times an action can be undone

Reversible behavioral types: multiparty case

- [I. Castellani, M. Dezani-Ciancaglini, P. Giannini: Concurrent Reversible Sessions. CONCUR 2017]
  - Choices work as checkpoints
  - Nondeterministic rollback
- [C. A. Mezzina, J. A. Pérez: Causally Consistent Reversible Choreographies: A Monitors-as-Memories Approach. PPDP 2017]
  - Monitors for each participant extracted from a choreography

Reversible binary behavioral contracts

- [F. Barbanera, M. Dezani-Ciancaglini, U. de'Liguoro: Reversible client/server interactions, Formal Asp. Comput. 28(4), 2016]
  - Ensuring compliance even with arbitrary reversibility
- [F. Barbanera, I. Lanese, U. de'Liguoro: Retractable and speculative contracts. COORDINATION 2017]
  - Using controlled backtracking to ensure compliance

Open question: types for reversibility control

- Most of the works above use standard types
- Very little about reversibility control
  - Which are the checkpoints
  - How many times an action can be reversed
- What about types for reversibility control?
  - Types describe which patterns of actions and undoing are legal

$$(?\,\mathrm{Int}\,.\,!^{-1}\,String\,.\,end) + (?\,Bool\,.\,?^{-1}\,\mathrm{Int}\,.\,!\,\mathrm{Int}\,.\,end)$$

  - Can be used to typecheck or monitor processes in languages with controlled reversibility
  - Will ensure coherent rollback in component-based systems

Model checking of reversible systems

- Checks whether a property in a given logic holds by exploring the state space
- Should properties describe full behavior or only forward behavior?
- History makes infinite state any recursive program
  - rec X. a. X
  - Abstractions are needed
- Causal consistency seems related to partial-order reduction
  - Can this be made formal?
  - How can this be exploited?

Abstract interpretations of reversible systems

- Extract abstract view of a system preserving given properties
- There is a Galois connection between a reversible calculus and the underlying forward-only calculus
  - Abstraction: removing history information
  - Concretion: adding empty history
- Weaker abstractions may be interesting
  - Part of the history relevant for some property
  - Cfr. program slicing

New analysis technique

- [J. Krivine: A Verification Technique for Reversible Process Algebra. RC 2012]
- Consider a program that needs backtracking (e.g., 8 queens problem)
- Can be implemented in a reversible calculus with irreversible actions
- In order to check its correctness it is enough to consider its causal traces
  - Much simpler than considering all traces
  - Can be efficiently computed

Summary

- Analysis techniques are a main topic in concurrency theory and programming language research
- Reversible concurrent calculi and languages are fast developing
- Time has arrived for considering analysis techniques for them
- Reversibility has a strong impact on analysis techniques

Finally

Thanks!

Questions?