History-Dependent Automata

Ugo Montanari
Dipartimento di Informatica
Università di Pisa

in collaboration with

Marzia Buscemi, GianLuigi Ferrari, Marco Pistore,
Roberto Raggi, Emilio Tuosto

Outline

- History Dependence
- Finite-State Verification of HD-Systems
- History-Dependent Automata
- The Zoo of HD-Automata
- The HAL Environment
- Coalgebras
- History-Dependent Automata with Symmetries
- Bialgebras
- A Bialgebraic Theory of HD-Automata
- Permutation Algebras
- A new sufficient condition for lifting from <u>Set</u> to <u>Alg(Σ)</u>
- Conclusions & Future Work

History Dependence

- Ability of declaring new names (variables, locations, resources) while computing and of referring to them later
- Examples:
  - declarations in block structured languages
  - mobile systems (e.g. π-calculus): extrusion of new names
  - causal systems (e.g. CCS with causality, Petri nets)
    - » every transition generates a new name (event)
    - » causally dependent transitions refer to it
  - located systems: new localities are new names
  - combination of the above
- Equivalence/congruence defined up to bisimilarity
- Ordinary definition with infinite supply of ordered fresh names

Finite-State Verification of HD-Systems

- Useful for model checking causality properties
- Several security properties expressed as semantic equivalence
- Need of deallocation/reallocation of unused, old names
- Similar to memory allocation/deallocation in block-structured languages
- Finiteness condition often fulfilled for protocols, coordinators
- Fresh names cannot be chosen from a different set
- Equivalent systems can have different free names (deadlocked, unusable)
- Difficult to agree on the choice of new names
- Formal definition uses any fresh name => infinite branching transition system
- Algorithms just assume that fresh, corresponding names are the same
- No coalgebraic representation of LTS, no minimal representatives

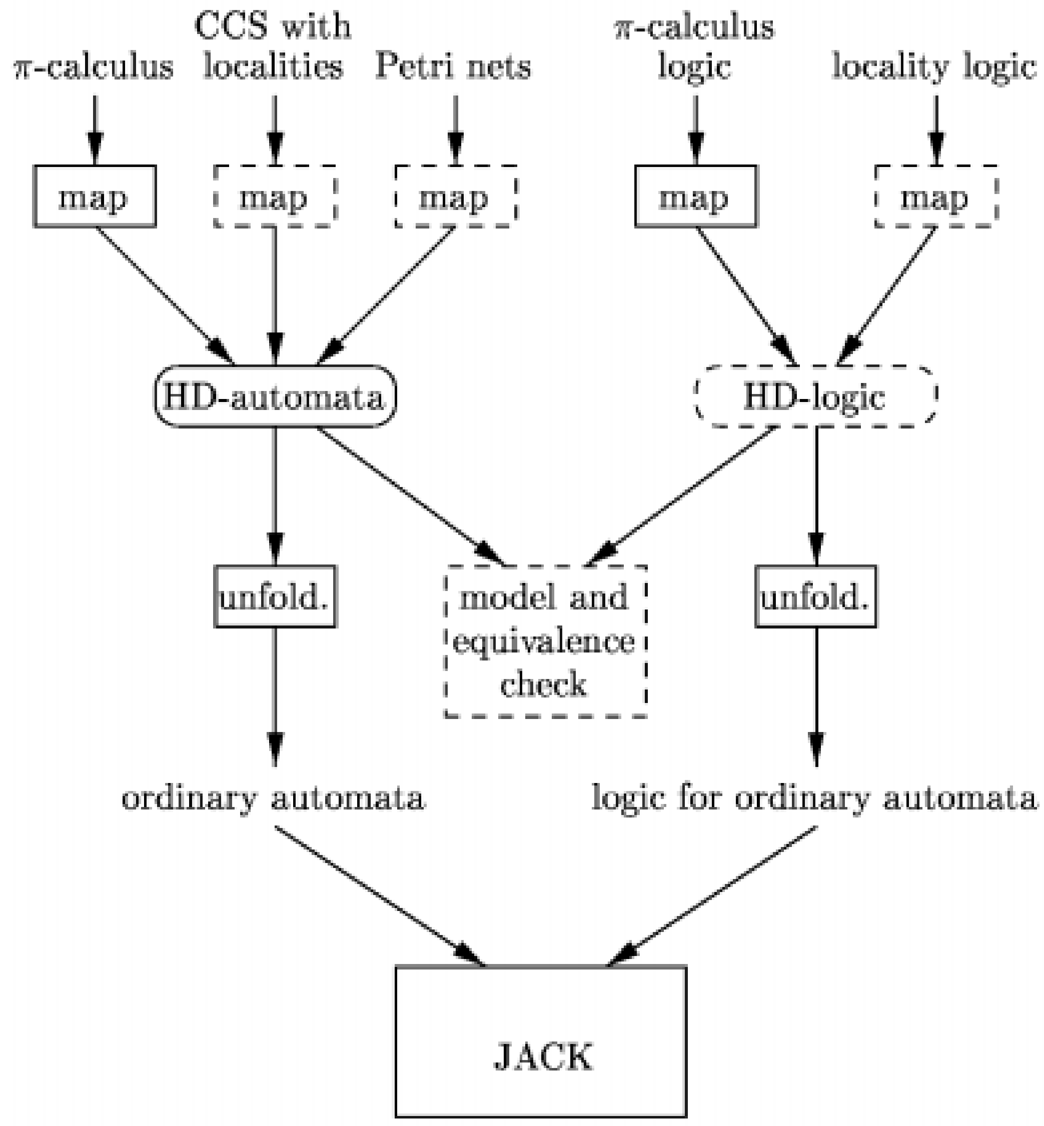
History-Dependent Automata

- States are equipped with local names
- A transition is decorated by
  - a label referring to names in the source state
  - an injective function defining the names of the target state in terms of
    - (some of) the names of the source state
    - fresh names generated in the transition
- Every bisimilar pair is equipped with a partial bijective function defining name correspondence
- Bisimilarity checks for compatibility of transition labels and of correspondences in the target state.
- More complex definition for formal names in input transitions

The Zoo of HD-Automata

- HD-automata for
  - early/late π-calculus
  - CCS with localities, causality, P/T Petri nets
  - open π-calculus, asynchronous π-calculus
  - causal dependency on graph rewriting
- Finite when there is a bounded number of threads
- HAL verification environment developed at IEI-CNR
- HD-automata bisimilarity defined
  - via span of open maps
  - on the category of marked labeled graphs
  - internal on the category of named sets
- No minimal representatives

The HAL Environment



HD Automata with Symmetries, I

Definition 32 (HD-automata) *A* HD-automaton with Symmetries *(or simply* HD-automaton*)* $\mathcal{A}$ *is a tuple* $\langle \mathcal{S}, \mathrm{sym}, \mathcal{L}, \longmapsto \rangle$, *where:*

- $\mathcal{S}$ *is the set of* states*;*
- $\mathrm{sym} : \mathcal{S} \to \mathcal{S}ym$ *associates to each state a finite-support* symmetry*;*
- $\mathcal{L}$ *is the set of* labels*;*
- $\longmapsto \subseteq \{\langle Q, l, \zeta, Q' \rangle \mid Q, Q' \in \mathcal{S},\, l \in \mathcal{L},\, \zeta$ *is a finite-kernel permutation*$\}$ *is the* transition relation, *where:*
  - $\cdot$ Q *and* Q' *are, respectively, the source and the target states;*
  - $\cdot$ l *is the label of the transition, and*
  - $\cdot$ ζ *is a permutation, that describes how the names of the target state* Q' *correspond, along this transition, to the names of the source state* Q*.*

  Whenever $\langle Q, l, \zeta, Q' \rangle \in \longmapsto$ *then we write* $Q \overset{l}{\longmapsto}_{\zeta} Q'$.

HD Automata with Symmetries, II

Definition 34 (HD-bisimulation) *Let $\mathcal{A}$ be a HD-automaton. A* HD-simulation *for $\mathcal{A}$ is a set of triples*

$$\mathcal{R} \subseteq \{\langle Q_1, \delta, Q_2\rangle \mid Q_1, Q_2 \in \mathcal{Q},\ \delta \text{ is a finite-kernel permutation}\}$$

such that, whenever $\langle Q_1, \delta, Q_2\rangle \in \mathcal{R}$ then:

- *for each $\rho_1 \in \mathrm{sym}(Q_1)$ and each $Q_1 \xmapsto{l_1}_{\zeta_1} Q_1'$, there exist some $\rho_2 \in \mathrm{sym}(Q_2)$ and some $Q_2 \xmapsto{l_2}_{\zeta_2} Q_2'$, such that:*
  $\cdot$ $l_2 = \gamma(l_1)$, *where* $\gamma = \rho_2^{-1} \circ \delta \circ \rho_1$;
  $\cdot$ $\langle Q_1', \delta', Q_2'\rangle \in \mathcal{R}$, *where:* $\delta' = \begin{cases} \zeta_2^{-1} \circ \gamma \circ \zeta_1 & \text{if } l_1 \in \mathcal{L}_0 \\ \zeta_2^{-1} \circ \gamma_{+1} \circ \zeta_1 & \text{if } l_1 \in \mathcal{L}_1. \end{cases}$

A HD-bisimulation *for $\mathcal{A}$ is a set of triples $\mathcal{R}$ such that both $\mathcal{R}$ and $\mathcal{R}^{-1} = \{\langle Q_2, \delta^{-1}, Q_1\rangle \mid \langle Q_1, \delta, Q_2\rangle \in \mathcal{R}\}$ are HD-simulations for $\mathcal{A}$.*

Coalgebras

- Interactive systems as labeled transition systems
- Coalgebraic semantics of labeled transition systems
  - coalgebras dual to algebras
  - initial algebras vs. final coalgebras
  - the unique morphism identifies bisimilar states
  - the image via the unique morphism yields the minimal representative

Algebras vs. Coalgebras

Category **Set**
Functor F
Function f
Function g

$$\begin{tikzcd} X \arrow[r, "h"] & Y \\ F(X) \arrow[u, "f"] \arrow[r, "F(h)"] \arrow[ur, phantom, "="] & F(Y) \arrow[u, "g"'] \end{tikzcd}$$

$$A \xrightarrow{h} B$$

Algebra A
Algebra B
Category Alg(F)

Category **Set**
Functor F
Function f
Function g

$$\begin{tikzcd} X \arrow[r, "h"] \arrow[d, "f"'] \arrow[dr, phantom, "="] & Y \arrow[d, "g"] \\ F(X) \arrow[r, "F(h)"] & F(Y) \end{tikzcd}$$

$$A \xrightarrow{h} B$$

Coalgebra A
Coalgebra B
Category Coalg(F)

Iterative Algorithm

$$X \xrightarrow{h_0} 1 \qquad \begin{tikzcd} X \arrow[r, "h_n"] \arrow[d, "f"'] & F(1)^n \\ F(X) \arrow[r, "F(h_n)"] & F(1)^{n+1} \end{tikzcd} \qquad X \xrightarrow{h_{n+1}=f;F(h_n)} F(1)^{n+1}$$

In the finite case the algorithm terminates
when the kernel of h_n coincides with the kernel of h_{n+1}

Category of Named Sets

Objects

$A : (ns = \langle\, Q : Set, |\_| : Q \longrightarrow \omega, \leq : Q \times Q \longrightarrow Bool, G : \prod_{q:Q} .\mathcal{P}_f(\{v_1..v_{|q|}\} \xrightarrow{bij} \{v_1..v_{|q|}\})\,\rangle)$
with the constraint:
$\forall q : Q_A.G_A q$ permutation group and $\leq_A$ total ordering
$\{q : Q_A\}_A \stackrel{def}{=} \{v_1..v_{|q|_A}\}$

Arrows

$H : (nf = \langle\, s : ns, d : ns, h : Q_s \longrightarrow Q_d, \Sigma : \prod_{q:Q_s} .\mathcal{P}_f(\{hq\}_d \xrightarrow{inj} \{q\}_s)\,\rangle)$
with
$\forall q : Q_{s_H}.\forall \sigma : \Sigma_H q.\ G_{d_H}(h_H q); \sigma = \Sigma_H q$ and $\sigma; G_{s_H} q \subseteq \Sigma_H q$

Arrow composition

$\_;\_ : nf \times nf \longrightarrow nf$ partial
H;K is defined only if $d_H = s_K$
$s_{H;K} = s_H$
$d_{H;K} = d_K$
$h_{H;K} : Q_{s_H} \longrightarrow Q_{d_K} = h_H; h_K$
$\Sigma_{H;K}(q : Q_{s_H}) = \Sigma_K(h_H q); \Sigma_H q$

Identity

$id : ns \longrightarrow nf$
$s_{id\ A} = d_{id\ A} = A$
$h_{id\ A} q = q$
$\Sigma_{id\ A} q = G_A q$

Algebras & Coalgebras

- Compositional systems represented as algebras
- Compose both states and whole transition systems:
  - in CCS p|q vs. the synchronization tree

    synch(p|q) = synch(p)|synch(q)
- Commuting pentagonal diagram by Turi & Plotkin on algebras/coalgebras => bialgebras
- Sufficient conditions by Corradini, Heckel, Montanari on algebraic specifications
- Simpler conditions by Buscemi, Montanari
- Bisimilarity is a congruence, existence of the minimal representatives

Bialgebras

Coalgebras for $P_L : \mathbf{Set} \to \mathbf{Set}$ with

$$S \mapsto \mathcal{P}_{countable}(L \times S)$$

are LTS's with countable degree.

Structured coalgebras: lift P_L to $P_L^{\mathcal{R}}$ on $\mathcal{A}lg(\Sigma, E)$

SOS rules in algebraic format define Σ-operations on

$$P_L(|A|) = \mathcal{P}_{countable}(L \times |A|)$$

E.g.

$$[\mathsf{com}]\ \frac{P \xrightarrow{\bar{x}y} P', Q \xrightarrow{xy} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

➡ $S_1|S_2 = \ldots \cup$
$\{\langle \tau, P'|Q' \rangle \mid \langle \bar{x}y, P' \rangle \in S_1, \langle xy, Q' \rangle \in S_2\} \cup$
$\{\langle \tau, P'|Q' \rangle \mid \langle xy, P' \rangle \in S_1, \langle \bar{x}y, Q' \rangle \in S_2\}$

A Bialgebraic Theory of HD-Automata

- Case study for the π-calculus
- Permutation algebras of states
- Labeled transition system essentially the same
- SOS rules in De Simone format propagating permutations through transitions
- Generation of new names obtained by shifting permutations forwards
- Conditions by Corradini, Heckel, Montanari satisfied
- Orbits $[p] = \{p' | p'=\rho(p), \rho$ a name permutation$\}$ of processes are HD-states
- Symmetries $\{p\} = \{\rho | \rho(p)= p\}$ associated to states => fewer transitions
- HD-automata defined as coalgebras in the category of named sets or as bialgebras on the category of permutation algebras coincide

Permutation Algebras

A permutation algebra A consists of:

- a set $|A|$ of states (the support)
- for every permutation $\rho : \mathbf{N} \to \mathbf{N}$, an operation

$$\rho^A : |A| \to |A|$$

The operations should satisfy the axioms of permutations:

$$\mathrm{id}^A(X) = X \qquad \text{and} \qquad \rho^A({\rho'}^A(X)) = (\rho \circ \rho')^A(X)$$

Lifting a Coalgebra from <u>Set</u> to <u>Alg(Σ)</u>

(Buscemi, Montanari)

<u>Set</u>

$$\begin{tikzcd} |A| \arrow[r, "h"] \arrow[d, "f"'] \arrow[dr, phantom, "="] & |B| \arrow[d, "g"] \\ P_L(|A|) \arrow[r, "P_L(h)"'] & P_L(|B|) \end{tikzcd}$$

<u>Alg(Σ)</u>

$$\begin{tikzcd} A \arrow[r, "h"] \arrow[d, "f"'] \arrow[dr, phantom, "="] & B \arrow[d, "g"] \\ P_\Delta(A) \arrow[r, "P_\Delta(h)"'] & P_\Delta(B) \end{tikzcd}$$

B: Σ-algebra of interest
A: Σ-algebra freely generated by B
h: surjective homomorphism
g: LTS of interest
f: LTS freely generated by g
E: a complete axiomatization of B

if all the axioms in E bisimulate
then the diagram commutes in <u>Set</u>

if the diagram commutes in <u>Set</u>
with h surjective
then g is a homomorphism
and the diagram commutes in Alg(Σ)

Conclusions & Future Work

- A coalgebraic semantics for HD-automata
- A first-order, coalgebraic denotational semantics for flat π-calculus
- Existence of minimal realizations
- Uniform minimization algorithms (list partitioning by Kanellakis & Smolka)
- Implementation of the minimization algorithm
- Non-flat π-calculus (parallel composition, restriction, etc., but not prefix)
- Extensions to π-calculus with prefix, fusion calculus
- Extension to term/process passing calculi, symbolic execution

Bibliography on HD-Automata and Coalgebras

- Montanari, U. and Pistore, M., Checking Bisimilarity for Finitary pi-calculus, in: Insup Lee, Scott A. Smolka, Eds., CONCUR'95: Concurrency Theory, Springer LNCS 962, pp. 42-56.
- Montanari, U., Pistore, M., and Yankelevich, D., Efficient Minimization up to Location Equivalence, in: Hanne Riis Nielson, Ed., Programming Languages and Systems - ESOP'96, Springer LNCS 1058, pp. 265-279.
- Montanari, U. and Pistore, M., Minimal Transition Systems for History-Preserving Bisimulation, in: Ruediger Reischuk, Michel Morvan, Eds., STACS 97, Springer LNCS 1200, 1997, pp. 413-425.
- Ferrari, G., Ferro, G., Gnesi, S., Montanari, U., Pistore, M. and Ristori, G., An Automata Based Verification Environment for Mobile Processes, in: Ed Brinksma, Ed., TACAS 1997, Springer LNCS 1217, pp. 275-289.
- Honsell, F., Lenisa, M., Montanari, U. and Pistore, M., Final Semantics for the Pi-Calculus, in: D. Gries and W-P. de Roever, Eds., PROCOMET'98, Chapman & Hall 1998, pp. 226-243.
- Ferrari, G., Gnesi, S., Montanari, U., Pistore, M. and Ristori, G., Verifying Mobile Processes in the HAL Environment, in: Alan J. Hu and Moshe Y. Vardi, Eds., CAV'98, Springer LNCS 1427, pp.511-515.
- Montanari, U. and Pistore, M., An Introduction to History Dependent Automata, in: Andrew Gordon, Andrew Pitts and Carolyn Talcott, Eds, Second Workshop on Higher-Order Operational Techniques in Semantics (HOOTS II), ENTCS, Vol. 10, 1998.
- Montanari, U. and Pistore, M., Finite State Verification for the Asynchronous Pi-Calculus, in: W. Rance Cleaveland, Ed., TACAS'99, Springer LNCS 1579, pp.255-269, 1999.
- Corradini, A., Heckel, R. and Montanari, U., Compositional SOS and Beyond: A Coalgebraic view of Open Systems, TCS, to appear.
- Montanari, U. and Pistore, M., Pi-Calculus, Structured Coalgebras and Minimal HD-Automata, in: Mogens Nielsen and Branislav Roman, Eds., Proc. MFCS 2000, Springer LNCS 1983.
- Ferrari, G., Montanari, U. and Pistore, M., Minimizing Transition Systems for Name Passing Calculi: A Co-algebraic Formulation, paper submitted for publication.
- Baldan, P., Corradini, A. and Montanari, U., Bisimulation Equivalences for Graph Grammars, in Wilfried Brauer, Juhani Karhumaeki, Arto Salomaa and Hartmut Ehrig, Festschrift in Honor of Grzegorz Rozeberg, Springer LNCS, 2002.
- Ferrari, G., Montanari, U. and Pistore, M., Minimizing Transition Systems for Name Passing Calculi: A Co-algebraic Formulation, Procs. ESOP 2002.
- Buscemi, M.G. and Montanari, U., A First Order Coalgebraic Model of Pi-Calculus Early Observational Equivalence, paper sumbitted for publication.