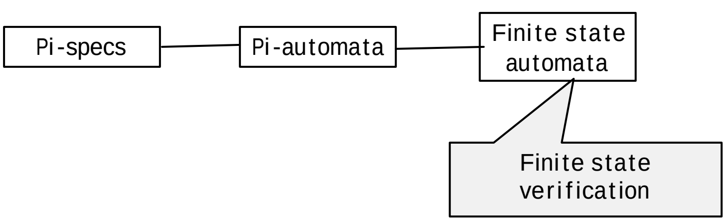
From co-algebraic specification to verification environment

G. Ferrari, U Montanari, M. Pistore
R. Raggi, E. Tuosto
Dipartimento di Informatica
Università di Pisa

Motivations

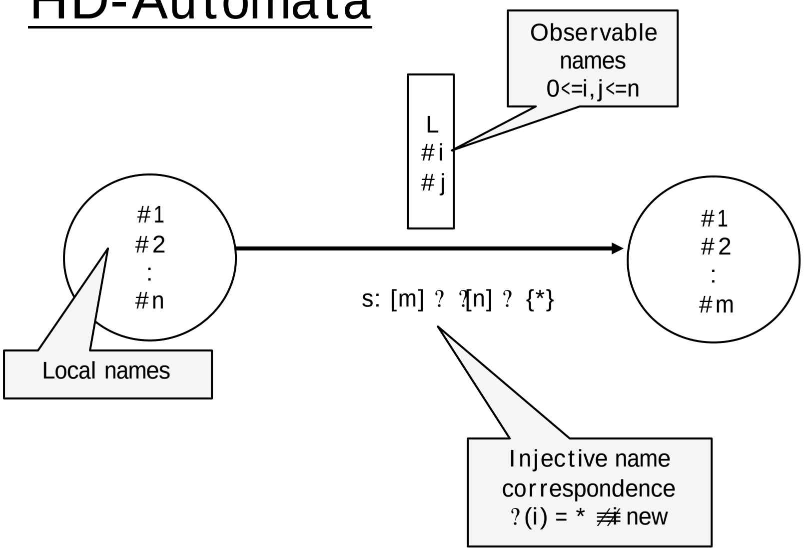
- The pi-calculus specification of the Handover protocol in HAL [CAV98]
  - 37199 States -- 47958 Transitions
  - Verification takes 15 Minutes
- The problem



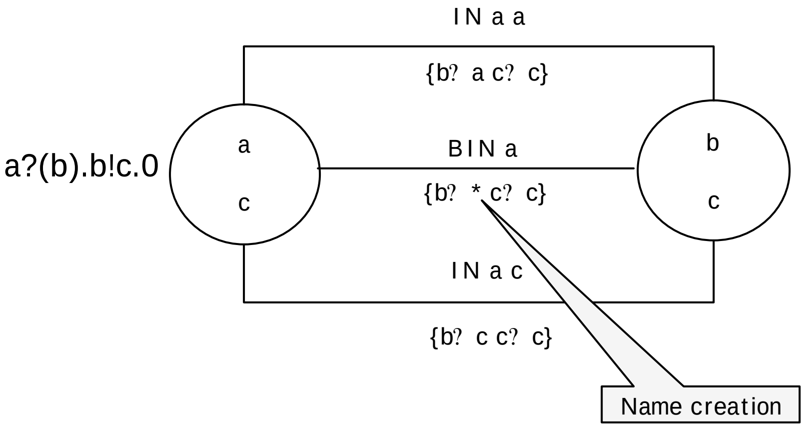
The approach

- Automata Model for Name Passing Process Calculi: History-Dependent (HD)-automata [Montanari&Pistore] specifically designed for verification purposes
  - Dynamic name allocation
  - Garbage collection of non-active names
  - Name symmetries
  - Finite state representation of finite control pi calculus agents
- Extend Automata-like Verification Techniques to HD-automata:
  Semantic Minimization via Partition Refinement

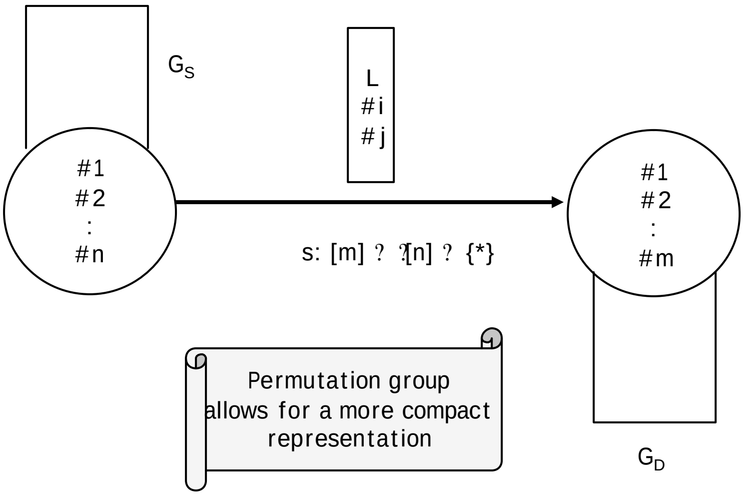
HD-Automata



Example



HD-Automata (Cont.)



Co-algebraic semantics

- Labelled Transition Systems = Co-algebras: endofunctor F over a suitable category

  K: Q ? P(LxQ)
- HD-automata are co-algebras defined on top of a permutation algebra [Montanari&Pistore MFCS200]

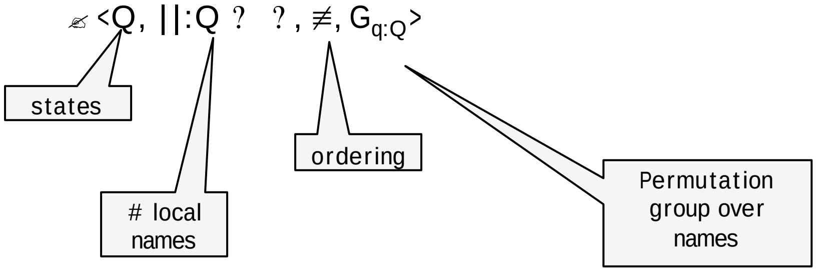
  General results: Minimal HD-automaton exists and equivalent pi-calculus processes have isomorphic minimal realizations

From co-algebras to verification environments

- Investigate the relationships between semantical structures and implementation data structures (next talk by Emilio Tuosto)
- Investigate a concrete representation of the underlying category

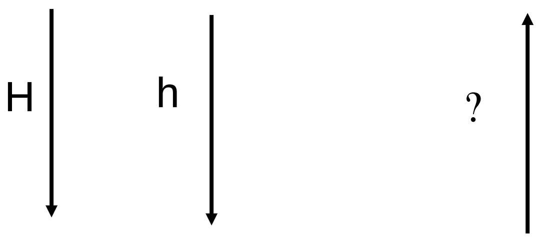
Named Sets

- A named set is a set of states equipped with a mechanism to give local meanings to names occurring in states
  - $\langle Q, ||:Q \; ? \quad ?, \not\equiv, G_{q:Q} \rangle$



Named Functions

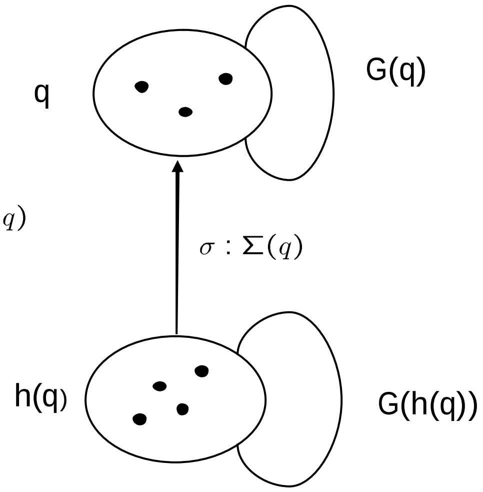
$$S = \langle Q_S, ||_S, \leq_S, G_S \rangle$$



$$\Sigma(q) : \{h(q)\} \xrightarrow{inj} \{q\}$$

$$D = \langle Q_D, ||_D, \leq_D, G_D \rangle$$

Named Functions (cont.)



$G(h(q);\sigma = \Sigma(q)$

$\sigma; G(q) \subseteq \Sigma(q)$

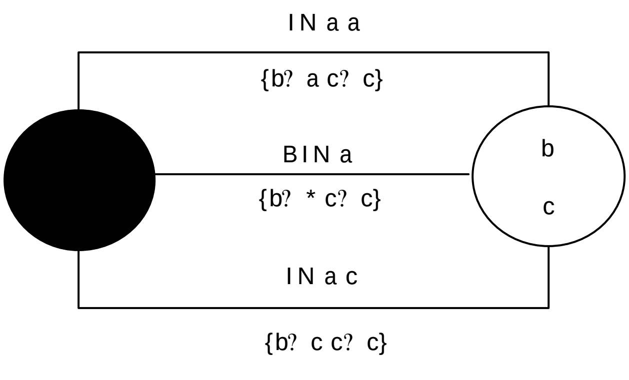
Minimization: partition refinement

- Basic step: splitting of blocks to create a new partition
- Basic operation: compute all the labelled transitions out of a given state

Bundles over ? actions

- ? = <D: NSet, Step>
- Step = {<l, ?, q, ?>} where
  - l : pi-calculus label
  - ?: function yielding the observable names
  - q: destination state
  - ?: injection relating the names of the destination state with the names of the original state such that G(q) ; Sq = Sq
    where Sq = {<l, ?, q, ? >} and
    ?; <l, ?, q, ? >} = <l, ?, q, ??? >}

Bundle (Example)



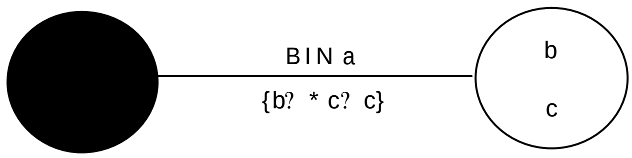
Bundle normalization

- Elimination of redundant transitions
  - <IN, xy, q, {#i ? y}>
  - <BIN, x, q, {#i ? *}>
- The first tuple is redundant: the second tuple represents its behaviour
- Red1(?) is the bundle obtained by removing redundant input transitions

Bundle Normalization (Cont)

- Compute the set of active names (an) of Red1(?).
  - Active names: names which appear in a destination state or in a label of a non redundant transition
- Compute Red2(?) by removing all the input transitions which refers to non-active names
- Construct the canonical permutation for the bundle

Bundle Normalization (Example)



The endofunctor T

The action over named sets

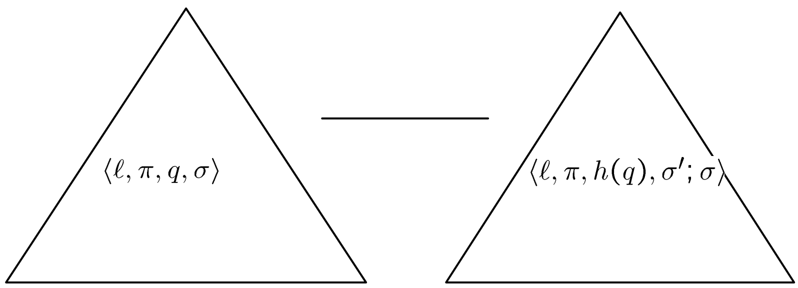
- $Q_{T(A)} = \{\beta : Bundle \mid \mathcal{D}_\beta = A, \beta$ normalized$\}$,
- $\mid \beta \mid_{T(A)} = \mid \beta \mid$,
- $G_{T(A)}(\beta) = Gr\ \beta$,
- $\beta_1 \leq_{T(A)} \beta_2$ iff $Step_{\beta_1} \sqsubseteq Step_{\beta_2}$,

The endofunctor T (cont)

- $S_{T(H)} = T(S_H)$,
- $D_{T(H)} = T(D_H)$,
- $h_{T(H)}(\beta : Q_{T(S_H)}) : Q_{T(D_H)} = norm(\beta')$,
- $\Sigma_{T(H)}(\beta : Q_{T(S_H)}) = Gr(norm(\beta')); (perm(\beta'))^{-1}; inj : \{\!| norm(\beta') |\!\} \longrightarrow \{\beta\}_{T(S_H)}$

  $\beta' = \langle D_H, \{\langle\ \ell, \pi, h_H(q), \sigma'; \sigma\ \rangle \mid \langle\ \ell, \pi, q, \sigma\ \rangle : Step_\beta, \sigma' : \Sigma_H(q)\}\rangle$.

The endofunctor (intuition)



The quadruples of the new bundle are obtained by saturating names by exploiting the canonical permutation

HD-automata: the underlying named set

- the elements of the state Q_A are π-agents $p(v_1..v_n)$ ordered lexicographically: $p_1 \leq_A p_2$ iff $p_1 \leq_{lex} p_2$
- $|p(v_1..v_n)|_A = n$,
- $G_A q = \{id : \{q\}_A \longrightarrow \{q\}_A\}$, where id denotes the identity function,
- $h : Q_A \longrightarrow \{\beta \mid \mathcal{D}_\beta = A\}$ is such that $\langle\, \ell, \pi, q', \sigma \,\rangle \in Step_{h(q)}$ represent the π-calculus transitions from agent q.

HD-automata as Named Functions

- $S_K = A$,
- $h_K(q) = norm(h(q))$,
- $\Sigma_K(q) = Gr(h_K(q)); (perm(h(q)))^{-1}; inj : \{\!|\, h(q) \,|\!\} \longrightarrow \{q\}_A$

The initial approximation

- Initial approximation H: all pi-calculus processes are in the same block
  $S_{H0} = S_K$ D_{H0} = unit ={*}

  G_{unit} * = ?

  $h_{H0}(q) = *$

  $?_{H0}(q) = \{?\}$

G.Ferrari: From co-algebraic …

PROFUNDIS@SOPHIA 23

The iterative construction

- Computation along the terminal sequence

  $H_{n+1} = K; T(H_n)$

G.Ferrari: From co-algebraic …

PROFUNDIS@SOPHIA 24

Main Theorem

✍ Let K be a finite state HD-automaton

- ✍ The iteration along the terminal sequence converges in a finite number of steps
- ✍ The minimal automata is the homomorphic image along the terminal sequence

Splitting blocks

There are q and q' such that

$h_{Hn}\, q = h_{Hn}\, q'$ and $h_{Hn+1}\, q \; ? \; h_{Hn+1}\, q'$

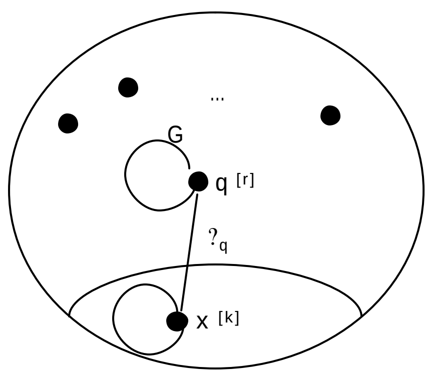
The iteration step

$$h_{H_{n+1}}(q) = norm\,\langle D_{H_n}, \{\langle\, \ell, \pi, h_{H_n}(q'), \sigma'; \sigma\,\rangle$$

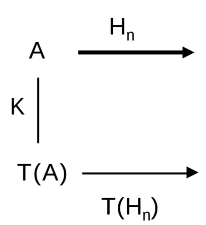
where $q \xrightarrow[\pi]{\ell}_{\sigma} q', \sigma' : \Sigma_{H_n}(q')\}\rangle$

G.Ferrari: From co-algebraic …

The iteration step



Block at step n

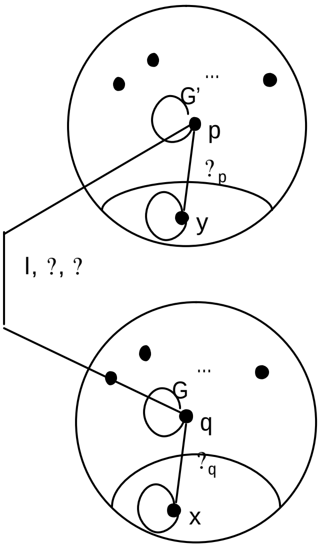


$h_{Hn}\ q = x$

$?_{Hn}\ q = G\ ;\ ?_q$

G.Ferrari: From co-algebraic …

The new approximation



- H_{n+1} = K ; T(H_n)
- h_{Hn+1} p = norm(?)
- $Step_?$ = {<l, x, ?, ?;$?_q$;?> and p –l, ?, ?->q}
- $?_{Hn+1}$ p = G' ; $?_p$; ? ?= G' ; $?_q$

Conclusions

- Tool engineering and more experimental results
- Applications: Security protocols (new name = nonces of sessions)
- Model Checking (logic for name allocation and deallocation Observational Semantics (Open bisimularity)
- Finite state Ambient Calculus