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

Local names

Observable names

Injective name correspondence

G.Ferrari: From co-algebraic …
Example

```
Example

IN a a
{b? a c? c}

a? b).blc.0
a b c

IN a c
{b? c c? c}

Name creation

```

HD-Automata (Cont.)

```
Permutation group allows for a more compact representation

```

```
G.Ferrari: From co-algebraic …

G.Ferrari: From co-algebraic …

```

```
```

G. Ferrari: From co-algebraic …

G. Ferrari: From co-algebraic …

```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
```

```
Co-algebraic semantics

.Labelled Transition Systems = Co-algebras: endofunctor $F$ over a suitable category $K: Q \rightarrow P(L \times Q)$

.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.

\[ \langle Q, \| : Q \|, \leq, G_{Q^Q} \rangle \]

states

ordering

Permutation group over names

# local names

Named Functions

\[ S = \langle Q_S, \|_S, \leq_S, G_S \rangle \]

\[ H \quad h \quad ? \]

\[ D = \langle Q_D, \|_D, \leq_D, G_D \rangle \]

\[ \Sigma(q) : \{h(q)\} \xrightarrow{\text{inj}} \{q\} \]
**Named Functions (cont.)**

\[ G(h(q); \sigma = \Sigma(q)) \]
\[ \sigma : \Sigma(q) \]
\[ \sigma; G(q) \subseteq \Sigma(q) \]

\[ G(q) \]
\[ h(q) \]
\[ G(h(q)) \]

**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

- \( \mathcal{D} = \langle D: \text{NSet}, \text{Step} \rangle \)
- \( \text{Step} = \{ \langle l, ?, q, ? \rangle \} \) 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 \( G(q) ; S_q = S_q \)
    where \( S_q = \{ \langle l, ?, q, ? \rangle \} \) and
    \( ? : \langle l, ?, q, ? \rangle = \langle l, ?, q, ?? \rangle \)

Bundle (Example)
Bundle normalization

- Elimination of redundant transitions
  - \( <\text{IN}, xy, q, \{\#i \ ? \ y\}> \)
  - \( <\text{BIN}, x, q, \{\#i \ ? \ *}\> \)
- 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

- $Q_{T(A)} = \{ \beta : Bundle \mid D_{\beta} = A, \beta \text{ normalized} \}$,
- $| \beta |_{T(A)} = | \beta |$,
- $G_{T(A)}(\beta) = Gr \beta$,
- $\beta_1 \leq_{T(A)} \beta_2$ iff $Step_{\beta_1} \subseteq Step_{\beta_2}$. 
The endofunctor $T$ (cont)

- $S_{T(H)} = T(S_H)$.

- $D_{T(H)} = T(D_H)$.

- $h_{T(H)}(\beta : QT(S_H)) : QT(D_H) = \text{norm}(\beta')$.

- $\Sigma_{T(H)}(\beta : QT(S_H)) = \text{Gr}(\text{norm}(\beta') ; (\text{perm}(\beta'))^{-1}; \text{inj} : \{\text{norm}(\beta')\} \rightarrow \{\beta\}_T(S_H)$

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

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 \( \pi \)-agents \( p(v_1..v_n) \) ordered lexicographically: \( p_1 \leq_A p_2 \) iff \( p_1 \leq_{lex} p_2 \)

- \( |p(v_1..v_n)|_A = n \)

- \( G_Aq = \{id : \{q\}_A \rightarrow \{q\}_A\} \), where \( id \) denotes the identity function,

- \( h : Q_A \rightarrow \{\beta \mid D_\beta = A\} \) is such that \( \langle k, \pi, q', \sigma \rangle \in \text{Step}_h(q) \) represent the \( \pi \)-calculus transitions from agent \( q \).

HD-automata as Named Functions

- \( S_K = A \)

- \( h_K(q) = \text{norm}(h(q)) \)

- \( \Sigma_K(q) = \text{Gr}(h_K(q)); (\text{perm}(h(q)))^{-1}; \text{inj} : \{h(q)\} \rightarrow \{q\}_A \)
The initial approximation

Initial approximation $H$: all pi-calculus processes are in the same block

$S_{H_0} = S_K \Delta_{H_0} = \text{unit } = \{\ast\}$

$G_{\text{unit}} \ast = ?$

$h_{H_0}(q) = \ast$

$\pi_{H_0}(q) = \{?\}$

The iterative construction

Computation along the terminal sequence

$H_{n+1} = K; T(H_n)$
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 automaton is the homomorphic image along the terminal sequence

Splitting blocks

There are $q$ and $q'$ such that

$$h_{H_n} q = h_{H_n} q' \text{ and } h_{H_{n+1}} q = h_{H_{n+1}} q'$$
The iteration step

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

where \( q \xrightarrow{\ell} \pi \sigma q', \sigma' : \Sigma_{H_n}(q') \} \} \)

---

The iteration step

\[ h_{H_n}(q) = x \]

\[ h_{H_n} q = x \]

Block at step n
### The new approximation

\[ H_{n+1} = K \cdot T(H_n) \]
\[ h_{H_{n+1}} p = \text{norm}(?) \]
\[ \text{Step}_n = \{<l, x, ?, ?; q, ?> \text{ and } p \rightarrow l, ?, ?, ?>q \} \]
\[ ?H_{n+1} p = G' ; ?_p ; ?>_q = G' ; ?_q \]

### 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 bisimilarity)
- Finite state Ambient Calculus