Comparison of process algebra equivalences using formats

Vashti Galpin

vashti@cs.wits.ac.za

Programme for Highly Dependable Systems
Department of Computer Science
University of the Witwatersrand
South Africa

http://www.cs.wits.ac.za/~vashti

Outline

- introduction
- background and motivation
  - comparison
  - existing approaches
- formats
- new format
  - congruence result
  - extension results
  - applications

Introduction

- process algebras
  - structured operational semantics – SOS
  - labelled transition systems
  - CCS, ACP, CSP
  - semantic equivalences – bisimulation etc.
  - extensions for different aspects of concurrent behaviour
- comparison of semantic equivalences
- develop new format
- prove congruence
- prove comparison results for format

Motivation

- understanding relationships between equivalences
- some related work
  - *ad hoc* approaches
  - interleaving semantics – Van Glabbeek
  - observation trees – Degano *et al*
- deal with bisimulation based on relations
  - pomset bisimulation – $p \xrightarrow{a|b} p'$
  - location bisimulation – $p \xrightarrow[u]{a} p'$
- take syntactic approach to labels instead of schematic

Formats

- metatheory of process algebra
- description of process algebra rules
- De Simone, GSOS, *tyft/tyxt*, *ntyft/ntyxt*, *path*, *panth*
  - single-sorted signature and schematic labels
  - negative premises, predicates
- example – *tyft/tyxt* format

$$\frac{\{t_i \xrightarrow{a_i} y_i \mid i \in I\}}{f(x_1, \ldots, x_n) \xrightarrow{a} t} \qquad \frac{\{t_i \xrightarrow{a_i} y_i \mid i \in I\}}{x \xrightarrow{a} t}$$

- open terms, closed terms, substitutions
- proof

Formats (cont.)

- congruence results
- conservative extension results
- related work
  - Algebraic De Simone format – Ferrari and Montanari
  - *promoted tyft/tyxt* format – Bernstein
  - Fokkink and Verhoef

A new format

- many-sorted signature Σ
- distinguished process sort P, many label sorts
- no operator of label sort can have process sort as argument

$$\mathbf{op} : s_1, \ldots, s_n \rightarrow s, \;\; s \neq \mathsf{P} \Rightarrow s_i \neq \mathsf{P} \;\; \forall 1 \leqslant i \leqslant n$$

- extended *tyft/tyxt* format

$$\frac{\{p_i \xrightarrow{\lambda_i} y_i \mid i \in I\}}{f(\eta_1, \ldots, \eta_m, x_1, \ldots, x_n) \xrightarrow{\lambda} p} \quad \text{or} \quad \frac{\{p_i \xrightarrow{\lambda_i} y_i \mid i \in I\}}{x \xrightarrow{\lambda} p}$$

A new format (cont.)

- for a specific process algebra
  - use Σ-algebra for labels
  - induce congruence over term algebra
  - relation over labels
- bisimulation wrt congruence $\equiv$

  assume $\equiv$ is a congruence over closed terms with sort other than P, then $P \sim_{\equiv} Q$ iff for all closed terms λ

  1. whenever $P \xrightarrow{\lambda} P'$, there exists Q' and λ' such that $Q \xrightarrow{\lambda'} Q'$, $\lambda \equiv \lambda'$ and $P' \sim_{\equiv} Q'$
  2. whenever $Q \xrightarrow{\lambda} Q'$, there exists P' and λ' such that $P \xrightarrow{\lambda'} P'$, $\lambda \equiv \lambda'$ and $P' \sim_{\equiv} Q'$
- can work with congruence over label terms

Congruence theorem

- $\mathcal{E}$ in extended *tyft/tyxt* format
- congruence over label terms $\equiv$
- $\mathcal{E}$ well-founded
- $\equiv$ compatible with $\mathcal{E}$
- **then** bisimulation with respect to $\equiv$ is a congruence for all operators

$$\forall 1 \leqslant k \leqslant m,\ \mu_k \equiv \nu_k, \quad \forall 1 \leqslant j \leqslant n,\ u_j \sim_\equiv v_j \Rightarrow$$
$$\mathbf{op}(\mu_1, \ldots, \mu_m, u_1, \ldots, u_n) \sim_\equiv \mathbf{op}(\nu_1, \ldots, \nu_m, v_1, \ldots, v_n)$$

- proof sketch
  - similar to but simpler than abstracting extension theorem

Extensions

- form sums – $\mathcal{E}_0 \oplus\!\!> \mathcal{E}_1$, $\equiv_0 \oplus \equiv_1$

$$p \sim^{\mathcal{E}_0}_{\equiv_0} q \qquad\qquad p \sim^{\mathcal{E}_0 \oplus\!> \mathcal{E}_1}_{\equiv_0 \oplus \equiv_1} q$$

- abstracting extension up to bisimulation wrt to a congruence
- refining extension up to bisimulation wrt a congruence
- type-1 sum
  - restriction on sorts on label of conclusion for rules in $\mathcal{E}_1$
- type-0 sum
  - restriction on sorts on label of conclusion for rules in $\mathcal{E}_1$
  - restriction on function in source of conclusion for rules in $\mathcal{E}_1$

Abstracting extension theorem

- $\mathcal{E}_0$, $\mathcal{E}_1$ in extended *tyft/tyxt* format
- $\mathcal{E}_0$ pure, label-pure
- $\mathcal{E}_1$ well-founded
- $\mathcal{E}_0 \oplus\!\!> \mathcal{E}_1$ type-0
- $\equiv_0 \oplus \equiv_1$ compatible with $\mathcal{E}_0 \oplus\!\!> \mathcal{E}_1$
- **then** $\mathcal{E}_0 \oplus\!\!> \mathcal{E}_1$ is an abstracting extension
- proof sketch
  - define a relation and prove it is a bisimulation
  - consider transitions from each process and use induction on the depth of the proof of transitions
  - find a new substitution to generate a proof that a matching transition exists

Refining extension theorem

- $\mathcal{E}_0$, $\mathcal{E}_1$ in extended *tyft/tyxt* format
- $\mathcal{E}_0$ pure, label-pure
- $\mathcal{E}_0 \oplus\!\!> \mathcal{E}_1$ type-1
- $\equiv_0 \oplus \equiv_1$ conservative with respect to $\equiv_0$
- **then** $\mathcal{E}_0 \oplus\!\!> \mathcal{E}_1$ is a refining extension
- proof sketch
  - work with the contrapositive and show two terms not related by the original bisimulation cannot be related by the new bisimulation
  - show that any new transition cannot 'fix'
  - use lemma about last rule used

Applications

- using the new format to express process algebras and prove congruence
  - CCS – Milner
  - CCS with locations – Boudol *et al*
  - multiprocessor CCS – Krishan
  - pomset CCS – Castellani

Applications (cont.)

- using the new format for comparison of semantic equivalences
  - pomset bisimulation is a subset of n multiprocessor bisimulation
  - proof sketch
    - introduce intermediate process algebra
    - prove refining extension of n multiprocessor bisimulation
    - prove intermediate process algebra bisimulation and pomset process algebra bisimulation the same
  - proper subset

$$\sum_{i=1}^{n} \sum_{j=1\, j\neq i}^{n} a_i.a_j \;\mid\; \prod_{k=1\, k\neq i\, k\neq j}^{n} a_k$$

$$\Big(\sum_{i=1}^{n} \sum_{j=1\, j\neq i}^{n} a_i.a_j \;\mid\; \prod_{k=1\, k\neq i\, k\neq j}^{n} a_k\Big) \;+\; \prod_{k=1}^{n} a_k$$

Conclusions

- new format
- congruence result
- definition of extensions
- new extension results
- use of new format