Algebraic results for structured operational semantics

Vashti Galpin

vashti@cs.wits.ac.za

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

Department of Computer Science

University of the Witwatersrand

Introduction

- process algebras
  - many different variants
  - CCS (Calculus of Communicating Systems) and extensions
  - three components
    - syntax – description of processes
    - structured operational semantics – behaviour of processes as labelled transition system
    - semantic equivalences – bisimulation
- what is the relationship between different process algebras?
- can use extended *tyft/tyxt* format to compare
- how conditions for comparison results relate to algebras used to represent process algebra labels

Outline

- formats – metatheory of process algebras
- extended *tyft/tyxt* format
- using the format to express process algebras
- comparison results for this format
- summing congruences and algebras
- ensuring conditions for results

Formats

- metatheory of process algebra
  - consider form of operational semantics rules
  - prove general results that hold when rules have that form
- congruence, conservative extension, axiomatisation, etc.
- extended *tyft/tyxt* format
  - treats labels of transitions syntactically, not schematically
  - comparison of process algebra semantic equivalences

Notation and definitions

- many-sorted signature $\Sigma = (S \cup \{\mathsf{P}\}, F)$
  - S – set of sorts
  - P – sort of processes
  - F – set of operators, $f : s_1, \ldots, s_n \to s$
  - suitable – only operators with range P take arguments of sort P
- terms over Σ – open $\mathbb{T}(\Sigma)$, closed $\mathbf{T}(\Sigma)$
- extended transition system specification (eTSS) – $\mathcal{E} = (\Sigma, R)$
  - R – set of rules with specific form

$$\frac{\{p_i \xrightarrow{\lambda_i} p'_i \mid i \in I\}}{p \xrightarrow{\lambda} p'}$$

I an index set, $p_i, p'_i, p, p' \in \mathbb{T}(\Sigma)_{\mathsf{P}}$, and $\lambda_i, \lambda \in \mathbb{T}(\Sigma)_S$ for $i \in I$.

Extended *tyft/tyxt* format

- additional conditions on form of rules
- bisimulation
  - use congruence over label terms to match in terms of meaning
  - informally, two terms from $\mathcal{E}$ are bisimilar up to $\equiv$ ($t \sim^{\mathcal{E}}_{\equiv} u$) if
    1. whenever $t \xrightarrow{\alpha} t'$ there exists u' and β such that $u \xrightarrow{\beta} u'$, $\alpha \equiv \beta$ and $t' \sim^{\mathcal{E}}_{\equiv} u'$
    2. whenever $u \xrightarrow{\alpha} u'$ there exists t' and β such that $t \xrightarrow{\beta} t'$, $\alpha \equiv \beta$ and $t' \sim^{\mathcal{E}}_{\equiv} u'$

    where $t, t', u, u' \in \mathbf{T}(\Sigma)_{\mathsf{P}}$ and $\alpha, \beta \in \mathbf{T}(\Sigma)_S$

Expressing process algebras in extended *tyft/tyxt* format

- Σ-algebra
  - non-empty carrier sets for each sort in S
  - function for each operator in F, mapping from the appropriate carrier sets to the appropriate carrier set
- unique homomorphism $i_{\mathcal{A}}$ from $\mathbf{T}(\Sigma)$ to $\mathcal{A}$
- $i_{\mathcal{A}}$ induces congruence $\equiv_{\mathcal{A}}$ over $\mathbf{T}(\Sigma)_S$
- choose Σ-algebra $\mathcal{A}$ to represent labels
- use congruence $\equiv_{\mathcal{A}}$ for bisimulation

Results for extended *tyft/tyxt* format

- congruence – bisimulation is a congruence for all operators defined in the format
- sums of eTSSs – $\mathcal{E}_0 \oplus \mathcal{E}_1$
  - sums of signatures – $\Sigma_0 \oplus \Sigma_1$
  - union of rule sets – $R_0 \cup R_1$
  - sum of congruences – $\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}$
- what is the relationship between

$$\sim^{\mathcal{E}_0}_{\equiv_{\mathcal{A}_0}} \qquad \text{and} \qquad \sim^{\mathcal{E}_0 \oplus \mathcal{E}_1}_{\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}} \qquad ?$$

Results for extended *tyft/tyxt* format (cont.)

- abstracting extension

  $$\sim^{\mathcal{E}_0}_{\equiv_{\mathcal{A}_0}} \quad \subseteq \quad \sim^{\mathcal{E}_0 \oplus \mathcal{E}_1}_{\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}}$$

  whenever
  - $\mathcal{E}_0$ pure, label-pure; $\mathcal{E}_1$ well-founded; $\mathcal{E}_0 \oplus \mathcal{E}_1$ type-0
  - $\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}$ is **compatible** with respect to $\mathcal{E}_0 \oplus \mathcal{E}_1$

- refining extension

  $$\sim^{\mathcal{E}_0}_{\equiv_{\mathcal{A}_0}} \quad \supseteq \quad \sim^{\mathcal{E}_0 \oplus \mathcal{E}_1}_{\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}}$$

  whenever
  - $\mathcal{E}_0$ pure, label-pure; $\mathcal{E}_0 \oplus \mathcal{E}_1$ type-1
  - $\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}$ is **conservative** with respect to $\equiv_{\mathcal{A}_0}$

More definitions

- $\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}$ **compatible** with respect to $\mathcal{E}_0 \oplus \mathcal{E}_1$
  for certain label terms that appear in the rules, it is possible to find a substitution with certain properties

- $\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}$ **conservative** with respect to $\equiv_{\mathcal{A}_0}$
  on the closed terms $\mathbf{T}(\Sigma_0)$, $\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}$ identifies the same terms as $\equiv_{\mathcal{A}_0}$

- $\mathcal{A}_0 \oplus \mathcal{A}_1$ – sum of algebras
  take sorted union of $\mathcal{A}_0$ and $\mathcal{A}_1$ when
  1. the carrier sets are identical for sorts in both $\Sigma_0 \cap \Sigma_1$
  2. the functions representing operators in $F_0 \cap F_1$ are equal

Questions

1. Is $\mathcal{A}_0 \oplus \mathcal{A}_1$ a $(\Sigma_0 \oplus \Sigma_1)$-algebra?
2. Is $\equiv_{\mathcal{A}_0 \oplus \mathcal{A}_1}$ the same as $\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}$?
3. Is $\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}$ conservative with respect to $\equiv_{\mathcal{A}_0}$?
4. Is $\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}$ compatible with $\mathcal{E}_0 \oplus \mathcal{E}_1$?

5. Are there general conditions that ensure compatibility?

Answers

1. $\mathcal{A}_0 \oplus \mathcal{A}_1$ is a $(\Sigma_0 \oplus \Sigma_1)$-algebra always
2. Under condition of sort-similarity
3. Under condition of sort-similarity
4. Under condition of sort-similarity

5. Under conditions on functions representing the operators that appears in the terms for which compatibility is required

Sort-similarity

- $\Sigma_0 \oplus \Sigma_1$ is **sort-similar** if for each $s \in S_0 \cap S_1$, $f \in F_0 \cup F_1$ with $f : s_1, \ldots, s_n \rightarrow s$ implies $f \in F_0 \cap F_1$
- this implies that the closed terms $\mathbf{T}(\Sigma_0 \oplus \Sigma_1) = \mathbf{T}(\Sigma_0) \cup \mathbf{T}(\Sigma_1)$, namely no new terms are formed by summing the eTSSs
- this also implies that any closed term with a sort from $S_0 \cap S_1$ must be in $\mathbf{T}(\Sigma_0) \cap \mathbf{T}(\Sigma_1)$

Proofs

- $\equiv_{\mathcal{A}_0 \oplus \mathcal{A}_1} = \equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}$
  - use sort-similarity to show that $i_{\mathcal{A}_0 \oplus \mathcal{A}_1} = i_{\mathcal{A}_0}$
  - $\Rightarrow$: straightforward
  - $\Leftarrow$: induction on the definition of $\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}$
- $\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}$ is conservative with respect to $\equiv_{\mathcal{A}_0}$
  - use the fact that $i_{\mathcal{A}_0 \oplus \mathcal{A}_1} = i_{\mathcal{A}_0}$
- $\equiv_{\mathcal{A}_0} \oplus \equiv_{\mathcal{A}_1}$ is compatible with $\mathcal{E}_0 \oplus \mathcal{E}_1$
  - $s \in (S_0 \cup S_1) - (S_0 \cap S_1)$ – by conservativity
  - $s \in (S_0 \cap S_1)$ – use the fact that $i_{\mathcal{A}_0 \oplus \mathcal{A}_1} = i_{\mathcal{A}_0}$

Conclusion

- existing results for extended *tyft/tyxt* format for semantic equivalence comparison
- conditions on algebras used to represent process algebra labels
- under condition of sort-similarity, can work with equivalence induced by sum of algebras
- general conditions under which compatibility can be achieved