Behavioural Theory for Session-Oriented Calculi

António Ravara[1]

Dep of Informatics, FCT, New Univ of Lisbon

Milan, November 23, 2009

[1] with Ivan Lanese and Hugo T. Vieira

Contextual equivalences for session-based core calculi

Aim

- Congruence relations, supporting
- equational reasoning, and enjoying of
- co-inductive characterisations, providing proof techniques.

The usual program

- Behavioural contextual equivalence: barbed congruence
- Co-inductive characterisation: full bisimilarity
  Substitution-closed ground bisimilarity over an early LTS
- Axioms for algebraic reasoning

Contextual equivalences for session-based core calculi

Aim

- Congruence relations, supporting
- equational reasoning, and enjoying of
- co-inductive characterisations, providing proof techniques.

The usual program

- Behavioural contextual equivalence: barbed congruence
- Co-inductive characterisation: full bisimilarity
  Substitution-closed ground bisimilarity over an early LTS
- Axioms for algebraic reasoning

Contextual equivalences for session-based core calculi

Aim

- Congruence relations, supporting
- equational reasoning, and enjoying of
- co-inductive characterisations, providing proof techniques.

The usual program

- Behavioural contextual equivalence: barbed congruence
- Co-inductive characterisation: full bisimilarity
  Substitution-closed ground bisimilarity over an early LTS
- Axioms for algebraic reasoning

Contextual equivalences for session-based core calculi

Aim

- Congruence relations, supporting
- equational reasoning, and enjoying of
- co-inductive characterisations, providing proof techniques.

The usual program

- Behavioural contextual equivalence: barbed congruence
- Co-inductive characterisation: full bisimilarity
  Substitution-closed ground bisimilarity over an early LTS
- Axioms for algebraic reasoning

Contextual equivalences for session-based core calculi

Aim

- Congruence relations, supporting
- equational reasoning, and enjoying of
- co-inductive characterisations, providing proof techniques.

The usual program

- Behavioural contextual equivalence: barbed congruence
- Co-inductive characterisation: full bisimilarity
  Substitution-closed ground bisimilarity over an early LTS
- Axioms for algebraic reasoning

Contextual equivalences for session-based core calculi

Aim

- Congruence relations, supporting
- equational reasoning, and enjoying of
- co-inductive characterisations, providing proof techniques.

The usual program

- Behavioural contextual equivalence: barbed congruence
- Co-inductive characterisation: full bisimilarity
  Substitution-closed ground bisimilarity over an early LTS
- Axioms for algebraic reasoning

Contextual equivalences for session-based core calculi

Aim

- Congruence relations, supporting
- equational reasoning, and enjoying of
- co-inductive characterisations, providing proof techniques.

The usual program

- Behavioural contextual equivalence: barbed congruence
- Co-inductive characterisation: full bisimilarity
  Substitution-closed ground bisimilarity over an early LTS
- Axioms for algebraic reasoning

Contextual equivalences for session-based core calculi

Aim

- Congruence relations, supporting
- equational reasoning, and enjoying of
- co-inductive characterisations, providing proof techniques.

The usual program

- Behavioural contextual equivalence: barbed congruence
- Co-inductive characterisation: full bisimilarity
  Substitution-closed ground bisimilarity over an early LTS
- Axioms for algebraic reasoning

Contextual equivalences for session-based core calculi

Aim

- Congruence relations, supporting
- equational reasoning, and enjoying of
- co-inductive characterisations, providing proof techniques.

The usual program

- Behavioural contextual equivalence: barbed congruence
- Co-inductive characterisation: full bisimilarity
  Substitution-closed ground bisimilarity over an early LTS
- Axioms for algebraic reasoning

Achievements

Results obtained

- Congruence relations:
  constructs of the calculi as compositional semantic operators on bisimilarity equivalence classes
- Axioms supporting equational reasoning, useful for proving:
  - Service compliance to an abstract behaviour
  - Correctness of program optimizations

Achievements

Results obtained

- Congruence relations:
  constructs of the calculi as compositional semantic operators on bisimilarity equivalence classes
- Axioms supporting equational reasoning, useful for proving:
  - Service compliance to an abstract behaviour
  - Correctness of program optimizations

Achievements

Results obtained

- Congruence relations:
  constructs of the calculi as compositional semantic operators on bisimilarity equivalence classes
- Axioms supporting equational reasoning, useful for proving:
  - Service compliance to an abstract behaviour
  - Correctness of program optimizations

Achievements

Results obtained

- Congruence relations:
  constructs of the calculi as compositional semantic operators on bisimilarity equivalence classes
- Axioms supporting equational reasoning, useful for proving:
  - Service compliance to an abstract behaviour
  - Correctness of program optimizations

Contents of the Chapter

Work developed

- Labelled transition systems,
- Behavioural equivalences, and
- Some useful axioms for
  - SSCC: Stream-based Service Centered Calculus
  - μse: dynamic multiparty session-based calculus
  - CC: Conversation Calculus

Contents of the Chapter

Work developed

- Labelled transition systems,
- Behavioural equivalences, and
- Some useful axioms for
  - SSCC: Stream-based Service Centered Calculus
  - μse: dynamic multiparty session-based calculus
  - CC: Conversation Calculus

Contents of the Chapter

Work developed

- Labelled transition systems,
- Behavioural equivalences, and
- Some useful axioms for
  - SSCC: Stream-based Service Centered Calculus
  - μse: dynamic multiparty session-based calculus
  - CC: Conversation Calculus

Contents of the Chapter

Work developed

- Labelled transition systems,
- Behavioural equivalences, and
- Some useful axioms for
  - SSCC: Stream-based Service Centered Calculus
  - μse: dynamic multiparty session-based calculus
  - CC: Conversation Calculus

Contents of the Chapter

Work developed

- Labelled transition systems,
- Behavioural equivalences, and
- Some useful axioms for
  - SSCC: Stream-based Service Centered Calculus
  - μse: dynamic multiparty session-based calculus
  - CC: Conversation Calculus

Contents of the Chapter

Work developed

- Labelled transition systems,
- Behavioural equivalences, and
- Some useful axioms for
  - SSCC: Stream-based Service Centered Calculus
  - μse: dynamic multiparty session-based calculus
  - CC: Conversation Calculus

Contents of the Chapter

Sections

2 SSCC's behavioural theory

- Barbed congruence coincides with full bisimilarity, which is a congruence
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

3 μse's behavioural theory

- Weak full bisimilarity
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

4 CC's behavioural theory

- Strong and weak bisimilarity notions, which are congruences
- Behavioral identities: illuminate on the spatial nature of processes; and pave the way for establishing a normal form result

Contents of the Chapter

Sections

2 SSCC's behavioural theory
- Barbed congruence coincides with full bisimilarity, which is a congruence
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

3 μse's behavioural theory
- Weak full bisimilarity
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

4 CC's behavioural theory
- Strong and weak bisimilarity notions, which are congruences
- Behavioral identities: illuminate on the spatial nature of processes; and pave the way for establishing a normal form result

Contents of the Chapter

Sections

2 SSCC's behavioural theory

- Barbed congruence coincides with full bisimilarity, which is a congruence
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

3 μse's behavioural theory

- Weak full bisimilarity
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

4 CC's behavioural theory

- Strong and weak bisimilarity notions, which are congruences
- Behavioral identities: illuminate on the spatial nature of processes; and pave the way for establishing a normal form result

Contents of the Chapter

Sections

2 SSCC's behavioural theory

- Barbed congruence coincides with full bisimilarity, which is a congruence
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

3 μse's behavioural theory

- Weak full bisimilarity
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

4 CC's behavioural theory

- Strong and weak bisimilarity notions, which are congruences
- Behavioral identities: illuminate on the spatial nature of processes; and pave the way for establishing a normal form result

Contents of the Chapter

Sections

2 SSCC's behavioural theory

- Barbed congruence coincides with full bisimilarity, which is a congruence
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

3 μse's behavioural theory

- Weak full bisimilarity
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

4 CC's behavioural theory

- Strong and weak bisimilarity notions, which are congruences
- Behavioral identities: illuminate on the spatial nature of processes; and pave the way for establishing a normal form result

Contents of the Chapter

Sections

2 SSCC's behavioural theory
- Barbed congruence coincides with full bisimilarity, which is a congruence
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

3 μse's behavioural theory
- Weak full bisimilarity
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

4 CC's behavioural theory
- Strong and weak bisimilarity notions, which are congruences
- Behavioral identities: illuminate on the spatial nature of processes; and pave the way for establishing a normal form result

Contents of the Chapter

Sections

2 SSCC's behavioural theory
- Barbed congruence coincides with full bisimilarity, which is a congruence
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

3 μse's behavioural theory
- Weak full bisimilarity
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

4 CC's behavioural theory
- Strong and weak bisimilarity notions, which are congruences
- Behavioral identities: illuminate on the spatial nature of processes; and pave the way for establishing a normal form result

Contents of the Chapter

Sections

2 SSCC's behavioural theory

- Barbed congruence coincides with full bisimilarity, which is a congruence
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

3 μse's behavioural theory

- Weak full bisimilarity
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

4 CC's behavioural theory

- Strong and weak bisimilarity notions, which are congruences
- Behavioral identities: illuminate on the spatial nature of processes; and pave the way for establishing a normal form result

Contents of the Chapter

Sections

2 SSCC's behavioural theory
- Barbed congruence coincides with full bisimilarity, which is a congruence
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

3 μse's behavioural theory
- Weak full bisimilarity
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

4 CC's behavioural theory
- Strong and weak bisimilarity notions, which are congruences
- Behavioral identities: illuminate on the spatial nature of processes; and pave the way for establishing a normal form result

Contents of the Chapter

Sections

2 SSCC's behavioural theory
- Barbed congruence coincides with full bisimilarity, which is a congruence
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

3 μse's behavioural theory
- Weak full bisimilarity
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

4 CC's behavioural theory
- Strong and weak bisimilarity notions, which are congruences
- Behavioral identities: illuminate on the spatial nature of processes; and pave the way for establishing a normal form result

Contents of the Chapter

Sections

2 SSCC's behavioural theory

- Barbed congruence coincides with full bisimilarity, which is a congruence
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

3 μse's behavioural theory

- Weak full bisimilarity
- Axioms that clarify the relationships between some constructs, and allow the proof of program transformations

4 CC's behavioural theory

- Strong and weak bisimilarity notions, which are congruences
- Behavioral identities: illuminate on the spatial nature of processes; and pave the way for establishing a normal form result

SSCC Behavioural identities

Garbage Collection Laws

Session Garbage Collection Consider that $\mathcal{D}$ does not bind r

$$(\nu \mathsf{r})\mathcal{D}[\![\mathsf{r} \rhd \mathbf{0}, \mathsf{r} \lhd \mathbf{0}]\!] \sim_{\mathsf{f}} \mathcal{D}[\![\mathbf{0}, \mathbf{0}]\!] \tag{1}$$

Stream Garbage Collection if f does not occur in P,

$$\text{stream } \mathbf{0} \text{ as } f \text{ in } P \sim_{\mathsf{f}} P \tag{2}$$

SSCC Behavioural identities

Independence Laws

Session Independence if $s \neq r$,

$$r \bowtie \mathsf{Q} \mid s \bowtie \mathsf{P} \sim_{\mathsf{f}} r \bowtie (s \bowtie \mathsf{P} \mid \mathsf{Q}) \tag{3}$$

Stream Independence if $f \neq g$,

$$\begin{aligned} &\text{stream } P \text{ as } f \text{ in stream } P' \text{ as } g \text{ in } Q \sim_{\mathsf{f}} \\ &\qquad \text{stream } P' \text{ as } g \text{ in stream } P \text{ as } f \text{ in } Q \end{aligned} \tag{4}$$

Streams are Orthogonal to Sessions

$$r \bowtie (\text{feed } v \mid P) \sim_{\mathsf{f}} \text{feed } v \mid r \bowtie P \tag{5}$$

SSCC Behavioural identities

Streams Laws

Stream Locality if $f \notin \mathrm{fn}(Q')$,

$$\mathsf{stream}\ P\ \mathsf{as}\ f\ \mathsf{in}\ (Q \mid Q') \sim_{\mathsf{f}} (\mathsf{stream}\ P\ \mathsf{as}\ f\ \mathsf{in}\ Q) \mid Q' \tag{6}$$

Unused Stream

$$\mathsf{stream}\ P\ \mathsf{as}\ f\ \mathsf{in}\ 0 \approx_{\mathsf{f}} P\{\mathsf{feed}\ v.Q \to Q\} \tag{7}$$

Parallel Composition Versus Streams
if $f \notin \mathrm{fn}(Q)$ and P does not contain feed ,

$$\mathsf{stream}\ P\ \mathsf{as}\ f\ \mathsf{in}\ Q \sim_{\mathsf{f}} P|Q \tag{8}$$

From an object-centred to a session-centred view

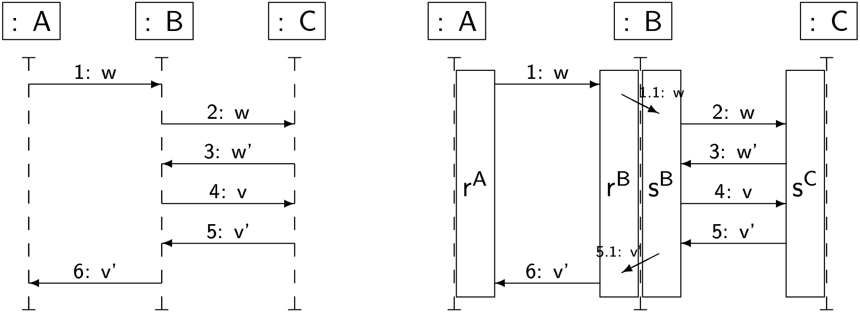

Figure: Sequence diagram communication pattern: object-centred and session-centred view.

Breaking sessions into request-response patterns

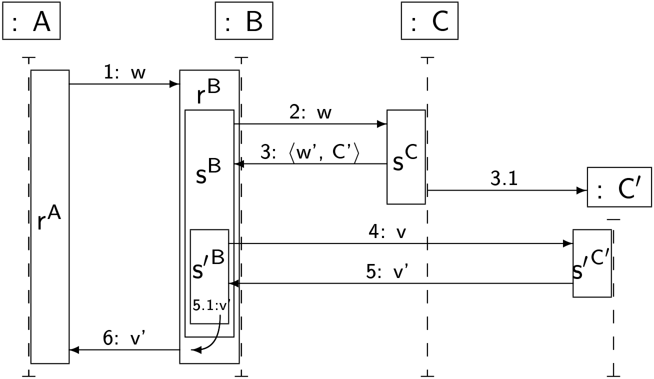

Figure: Sequence diagram communication pattern: using subsessions and continuations.

Muse Behavioural identities

Garbage Collection Laws

Session Garbage Collection $r \triangleright \mathbf{0} \sim_{\mathsf{f}} \mathbf{0}$

Location Garbage Collection $l :: \mathbf{0} \sim_{\mathsf{f}} \mathbf{0}$

Sessions Laws

Session Independence $r \triangleright Q \mid s \triangleright P \sim_{\mathsf{f}} r \triangleright (s \triangleright P \mid Q)$

Intra-Session Communication is Orthogonal w.r.t. Locations

$$l :: \overline{x}w \sim_{\mathsf{f}} m :: \overline{x}w$$

Intra-Location Communication is Orthogonal w.r.t. Sessions

$$r \triangleright x!w \sim_{\mathsf{f}} s \triangleright x!w$$

Muse Behavioural identities

Garbage Collection Laws

Session Garbage Collection $r \rhd \mathbf{0} \sim_{\mathsf{f}} \mathbf{0}$

Location Garbage Collection $l :: \mathbf{0} \sim_{\mathsf{f}} \mathbf{0}$

Sessions Laws

Session Independence $r \rhd Q \mid s \rhd P \sim_{\mathsf{f}} r \rhd (s \rhd P \mid Q)$

Intra-Session Communication is Orthogonal w.r.t. Locations

$$l :: \overline{x}w \sim_{\mathsf{f}} m :: \overline{x}w$$

Intra-Location Communication is Orthogonal w.r.t. Sessions

$$r \rhd x!w \sim_{\mathsf{f}} s \rhd x!w$$

Implementation correct with respect to a specification

Credit request scenario: specification

$I :: *CalculateRating \Rightarrow P$ with
$P = data(user).some\_comp.\overline{ret}\ rating$

Credit request scenario: implementation

$$I :: (\nu Calc_1 \ldots Calc_n)((\nu av)(\prod_{i=1}^{n} \mathrm{rec}\ X.av!Calc_i.X \mid$$
$$* CalculateRating \Rightarrow av?(u).\mathrm{invoke}\ u) \mid \prod_{i=1}^{n} *Calc_i \Rightarrow P)$$

CC Behavioural identities

Laws

1. $n \blacktriangleleft [P] \mid n \blacktriangleleft [Q] \sim n \blacktriangleleft [P \mid Q]$
2. $m \blacktriangleleft [n \blacktriangleleft [o \blacktriangleleft [P]]] \sim n \blacktriangleleft [o \blacktriangleleft [P]]$
3. $n \blacktriangleleft [l^{\uparrow}!(\tilde{n}).P] \sim l^{\downarrow}!(\tilde{n}).n \blacktriangleleft [P]$
4. If $n \notin \tilde{x}$ then $n \blacktriangleleft [l^{\uparrow}?(\tilde{x}).P] \sim l^{\downarrow}?(\tilde{x}).n \blacktriangleleft [P]$
5. $m \blacktriangleleft [n \blacktriangleleft [l^{\downarrow}!(\tilde{n}).P]] \sim n \blacktriangleleft [l^{\downarrow}!(\tilde{n}).m \blacktriangleleft [n \blacktriangleleft [P]]]$
6. If $\{m, n\} \# \tilde{x}$ then
   $m \blacktriangleleft [n \blacktriangleleft [l^{\downarrow}?(\tilde{x}).P]] \sim n \blacktriangleleft [l^{\downarrow}?(\tilde{x}).m \blacktriangleleft [n \blacktriangleleft [P]]]$

Finance Portal

Implementation correct with respect to a specification

$$
\begin{array}{l}
\mathit{FinPort} \blacktriangleleft [\ \mathit{CreditChat} \blacktriangleleft [\ \mathtt{login}^{\downarrow}?(\mathit{uId}).\mathit{ServProt}\ ]\ ] \\
| \\
\mathit{Client} \blacktriangleleft [\ \mathit{CreditChat} \blacktriangleleft [\ \mathtt{login}^{\downarrow}!(\mathit{uId}).\mathit{ClientProt}\ ]\ ]
\end{array}
$$

$$\sim$$

$$
\begin{array}{l}
\mathit{CreditChat} \blacktriangleleft [\ \mathtt{login}^{\downarrow}?(\mathit{uId}).\mathit{FinPort} \blacktriangleleft [\ \mathit{CreditChat} \blacktriangleleft [\ \mathit{ServProt}\ ]\ ]\ ] \\
| \\
\mathit{Client} \blacktriangleleft [\ \mathit{CreditChat} \blacktriangleleft [\ \mathtt{login}^{\downarrow}!(\mathit{uId}).\mathit{ClientProt}\ ]\ ]
\end{array}
$$

$$\sim$$

$$
\begin{array}{l}
\mathit{CreditChat} \blacktriangleleft [\ \mathtt{login}^{\downarrow}?(\mathit{uId}).\mathit{FinPort} \blacktriangleleft [\ \mathit{CreditChat} \blacktriangleleft [\ \mathit{ServProt}\ ]\ ]\ ] \\
| \\
\mathit{CreditChat} \blacktriangleleft [\ \mathtt{login}^{\downarrow}!(\mathit{uId}).\mathit{Client} \blacktriangleleft [\ \mathit{CreditChat} \blacktriangleleft [\ \mathit{ClientProt}\ ]\ ]\ ]
\end{array}
$$

Finance Portal

Implementation correct with respect to a specification

$$\begin{array}{l} \mathit{CreditChat} \blacktriangleleft [\, \mathtt{login}^{\downarrow}?(\mathit{uId}).\mathit{FinPort} \blacktriangleleft [\, \mathit{CreditChat} \blacktriangleleft [\, \mathit{ServProt} \,]\,]\,] \\ \mid \\ \mathit{CreditChat} \blacktriangleleft [\, \mathtt{login}^{\downarrow}!(\mathit{uId}).\mathit{Client} \blacktriangleleft [\, \mathit{CreditChat} \blacktriangleleft [\, \mathit{ClientProt} \,]\,]\,] \end{array}$$

$$\sim$$

$$\mathit{CreditChat} \blacktriangleleft [\, \mathtt{login}^{\downarrow}?(\mathit{uId}).\mathit{FinPort} \blacktriangleleft [\, \mathit{CreditChat} \blacktriangleleft [\, \mathit{ServProt} \,]\,] \mid \mathtt{login}^{\downarrow}!(\mathit{uId}).\mathit{Client} \blacktriangleleft [\, \mathit{CreditChat} \blacktriangleleft [\, \mathit{ClientProt} \,]\,]\,]$$