

# Behavioural Theory for Session-Oriented Calculi

António Ravara<sup>1</sup>

Dep of Informatics, FCT, New Univ of Lisbon

Milan, November 23, 2009

---

<sup>1</sup>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

- Labeled transition systems,
- Behavioural equivalences, and
- Some useful axioms for
  - SSCC: Stream-based Service Centered Calculus
  - $\mu se$ : 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
  - $\mu se$ : 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
  - $\mu se$ : 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
  - $\mu se$ : 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
  - $\mu$ se: 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
  - $\mu$ se: 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 $\mu$ se'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 $\mu$ se'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 $\mu$ se'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 $\mu$ se'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 $\mu$ se'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 $\mu$ se'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 $\mu$ se'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 $\mu$ se'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 $\mu$ se'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 $\mu$ se'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 $\mu$ se'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  $r$

$$(\nu r)\mathcal{D}[[r \triangleright \mathbf{0}, r \triangleleft \mathbf{0}]] \sim_f \mathcal{D}[[\mathbf{0}, \mathbf{0}]] \quad (1)$$

**Stream Garbage Collection** if  $f$  does not occur in  $P$ ,

$$\text{stream } \mathbf{0} \text{ as } f \text{ in } P \sim_f P \quad (2)$$

# SSCC Behavioural identities

## Independence Laws

**Session Independence** if  $s \neq r$ ,

$$r \bowtie Q \mid s \bowtie P \sim_f r \bowtie (s \bowtie P \mid Q) \quad (3)$$

**Stream Independence** if  $f \neq g$ ,

$$\begin{aligned} \text{stream } P \text{ as } f \text{ in stream } P' \text{ as } g \text{ in } Q &\sim_f \\ \text{stream } P' \text{ as } g \text{ in stream } P \text{ as } f \text{ in } Q &\quad (4) \end{aligned}$$

**Streams are Orthogonal to Sessions**

$$r \bowtie (\text{feed } v \mid P) \sim_f \text{feed } v \mid r \bowtie P \quad (5)$$

# SSCC Behavioural identities

## Streams Laws

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

$$\text{stream } P \text{ as } f \text{ in } (Q \mid Q') \sim_f (\text{stream } P \text{ as } f \text{ in } Q) \mid Q' \quad (6)$$

**Unused Stream**

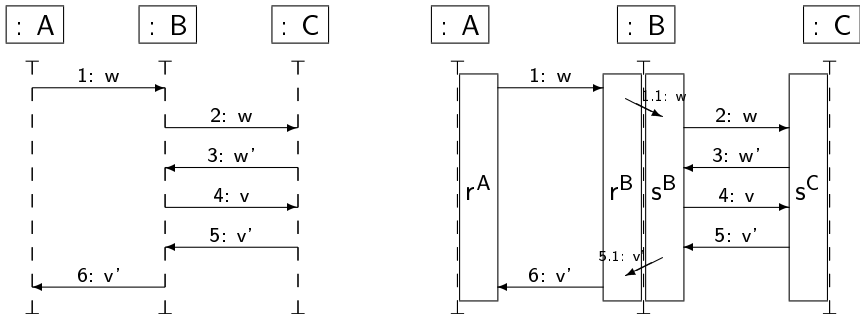
$$\text{stream } P \text{ as } f \text{ in } 0 \approx_f P\{\text{feed } v. Q \rightarrow Q\} \quad (7)$$

**Parallel Composition Versus Streams**

if  $f \notin \text{fn}(Q)$  and  $P$  does not contain feed ,

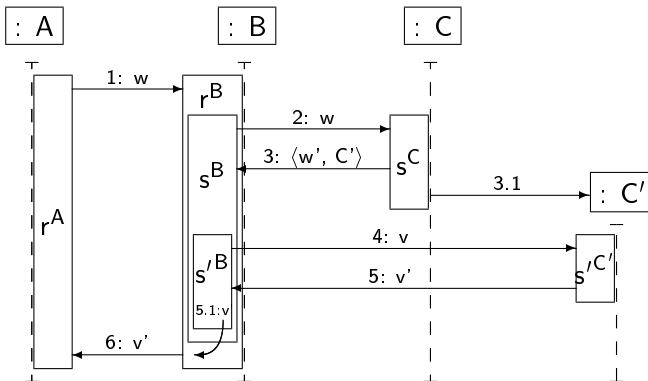
$$\text{stream } P \text{ as } f \text{ in } Q \sim_f P \mid Q \quad (8)$$

# 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  $r \triangleright \mathbf{0} \sim_f \mathbf{0}$

Location Garbage Collection  $l :: \mathbf{0} \sim_f \mathbf{0}$

## Sessions Laws

Session Independence  $r \triangleright Q \mid s \triangleright P \sim_f r \triangleright (s \triangleright P \mid Q)$

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

$$l :: \bar{x}w \sim_f m :: \bar{x}w$$

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

$$r \triangleright x!w \sim_f s \triangleright x!w$$

# Muse Behavioural identities

## Garbage Collection Laws

Session Garbage Collection  $r \triangleright \mathbf{0} \sim_f \mathbf{0}$

Location Garbage Collection  $l :: \mathbf{0} \sim_f \mathbf{0}$

## Sessions Laws

Session Independence  $r \triangleright Q \mid s \triangleright P \sim_f r \triangleright (s \triangleright P \mid Q)$

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

$$l :: \bar{x}w \sim_f m :: \bar{x}w$$

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

$$r \triangleright x!w \sim_f s \triangleright x!w$$

# Implementation correct with respect to a specification

## Credit request scenario: specification

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

## Credit request scenario: implementation

$$I :: (\nu Calc_1 \dots Calc_n) ((\nu av) (\prod_{i=1}^n \text{rec } X. av! Calc_i. X \mid$$

$$* CalculateRating \Rightarrow av?(u).invoke u) \mid \prod_{i=1}^n * Calc_i \Rightarrow P)$$

# CC Behavioural identities

## Laws

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

## Finance Portal

## Implementation correct with respect to a specification

$$\begin{array}{l}
 \text{FinPort} \triangleleft [ \text{CreditChat} \triangleleft [ \text{login}^{\downarrow?}(uld).\text{ServProt} ] ] \\
 | \\
 \text{Client} \triangleleft [ \text{CreditChat} \triangleleft [ \text{login}^{\downarrow!}(uld).\text{ClientProt} ] ]
 \end{array}$$

~

$$\begin{array}{l}
 \text{CreditChat} \triangleleft [ \text{login}^{\downarrow?}(uld).\text{FinPort} \triangleleft [ \text{CreditChat} \triangleleft [ \text{ServProt} ] ] ] \\
 | \\
 \text{Client} \triangleleft [ \text{CreditChat} \triangleleft [ \text{login}^{\downarrow!}(uld).\text{ClientProt} ] ]
 \end{array}$$

~

$$\begin{array}{l}
 \text{CreditChat} \triangleleft [ \text{login}^{\downarrow?}(uld).\text{FinPort} \triangleleft [ \text{CreditChat} \triangleleft [ \text{ServProt} ] ] ] \\
 | \\
 \text{CreditChat} \triangleleft [ \text{login}^{\downarrow!}(uld).\text{Client} \triangleleft [ \text{CreditChat} \triangleleft [ \text{ClientProt} ] ] ]
 \end{array}$$

## Finance Portal

## Implementation correct with respect to a specification

$$\begin{array}{l}
 \textit{CreditChat} \triangleleft [ \textit{login}^{\downarrow?}(uld).\textit{FinPort} \triangleleft [ \textit{CreditChat} \triangleleft [ \textit{ServProt} ] ] ] \\
 | \\
 \textit{CreditChat} \triangleleft [ \textit{login}^{\downarrow!}(uld).\textit{Client} \triangleleft [ \textit{CreditChat} \triangleleft [ \textit{ClientProt} ] ] ] \\
 \qquad \qquad \qquad \sim \\
 \textit{CreditChat} \triangleleft [ \textit{login}^{\downarrow?}(uld).\textit{FinPort} \triangleleft [ \textit{CreditChat} \triangleleft [ \textit{ServProt} ] ] \\
 | \\
 \qquad \qquad \qquad \textit{login}^{\downarrow!}(uld).\textit{Client} \triangleleft [ \textit{CreditChat} \triangleleft [ \textit{ClientProt} ] ] ]
 \end{array}$$