Contracts for Mobile Processes

Giuseppe Castagna <u>Luca Padovani</u>

Laboratoire PPS, CNRS, Université Paris Diderot

Istituto di Scienze e Tecnologie dell'Informazione, Università di Urbino

Parsec meeting, January 26th, 2009

Outline

- Motivation
 Protocols and processes
 Contracts and mobile systems
- Contracts
 Syntax
 Semantics
- Results
- 4 Concluding remarks

Protocols and processes

Session types

prescriptions on the use of channels

$$u: \sigma, v: \tau, \cdots \vdash P$$

Contracts

overall process behavior

$$u: Ch, v: Ch, \cdots \vdash P: T$$

Summary

- both are behavioral types
- σ = projection of T on u



What session types and contracts are for

Characterizing well-formed systems

- the system eventually terminates
- the system never deadlocks

Characterizing well-typed processes

- sent messages have the correct/expected type
- messages sent/delivered in the right order

Reasoning about processes by means of their type

- refactoring processes
- searching for services

A problem of abstraction

Session types	Contracts	
$? \texttt{Int.?Int.} \big(! \texttt{Real} \oplus ! \texttt{Error} \big)$	$a.a.(\overline{b}\oplus \overline{c})$	
?(!Bool.!Bool)	a	?

A natural candidate

Contracts without channel passing \Rightarrow ccs

Contracts with channel passing $\Rightarrow \pi$ -calculus

A problem of abstraction

Session types	Contracts
$? {\tt Int.?Int.} (!{\tt Real} \oplus !{\tt Error})$	$a.a.(\overline{b}\oplus \overline{c})$
?(!Bool.!Bool)	a

A natural candidate

Contracts without channel passing \Rightarrow ccs

Contracts with channel passing $\Rightarrow \pi$ -calculus

```
process store?(x).x?(y: Item).

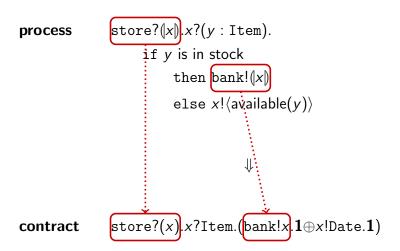
if y is in stock

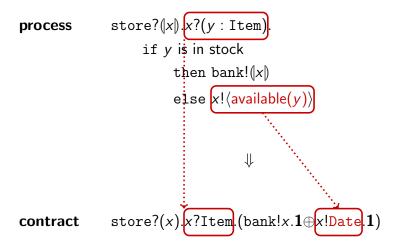
then bank!(x)

else x!\langle available(y)\rangle
```

 \downarrow

contract store?(x).x?Item.(bank!x. $\mathbf{1} \oplus x$!Date. $\mathbf{1}$)





```
store?(x).x?(y: Item).
process
                           v is in stock
                             then bank! (x)
                             else x! \langle available(y) \rangle
                 store?(x).x?Item.(bank!x.\mathbf{1} \oplus x!Date.\mathbf{1})
contract
```

Some typing rules

$$\frac{\Gamma \vdash e : t \qquad \Gamma \vdash P : T}{\Gamma \vdash \alpha! e.P : \alpha! t.T}$$

$$\frac{\Gamma \vdash P : T}{\Gamma \vdash \alpha! (\beta) . P : \alpha! \beta . T}$$

$$\frac{\Gamma, x : t \vdash P : T}{\Gamma \vdash \alpha?(x : t).P : \alpha?t.T}$$

$$\frac{\Gamma, x : \text{Ch} \vdash P : T}{\Gamma \vdash \alpha?(x) \cdot P : \alpha?(x) \cdot T}$$

Some typing rules

$$\frac{\Gamma \vdash e : t \qquad \Gamma \vdash P : T}{\Gamma \vdash \alpha! e.P : \alpha! t.T}$$

$$\frac{\Gamma \vdash P : T}{\Gamma \vdash \alpha! (\beta) . P : \alpha! \beta . T}$$

$$\frac{\Gamma, x : t \vdash P : T}{\Gamma \vdash \alpha?(x : t).P : \alpha?t.T}$$

$$\frac{\Gamma, \mathbf{x} : \mathsf{Ch} \vdash P : T}{\Gamma \vdash \alpha?(|\mathbf{x}|).P : \alpha?(\mathbf{x}).T}$$

undecidable → decidable

Outline

- Motivation
 Protocols and processes

 Contracts and mobile systems
- Contracts
 Syntax
 Semantics
- Results
- 4 Concluding remarks

failure, success

$$T ::= \mathbf{0} \mid \mathbf{1} \qquad \pi.T \mid T+T \mid T \oplus T \qquad T \mid T \mid (\nu a)T$$

$$\pi ::= \alpha?f \mid \alpha!f \mid \alpha!(a)$$

$$f ::= x \mid (x) \mid a \mid \text{Int} \mid \text{Bool} \mid \cdots$$

${\sf Infinite\ behaviors} = {\sf infinite\ terms}$

- regularity
- boundedness

$$X = c?Int.X$$

$$X = a?(x).(c!x.\mathbf{1} \mid X)$$

dynamic operators
$$T ::= \mathbf{0} \mid \mathbf{1} \qquad \pi.T \mid T+T \mid T \oplus T \qquad T \mid T \mid (\nu a)T$$

$$\pi ::= \alpha?f \mid \alpha!f \mid \alpha!(a)$$

$$f ::= x \mid (x) \mid a \mid \text{Int} \mid \text{Bool} \mid \cdots$$

Infinite behaviors = infinite terms

- regularity
- boundedness

$$X = c?Int.X$$
$$X = a?(x).(c!x.1 | X)$$

$$T ::= \mathbf{0} \mid \mathbf{1} \qquad \pi.T \mid T+T \mid T \oplus T \qquad T \mid T \mid (\boldsymbol{\nu}a)T$$

$$\pi ::= \alpha?f \mid \alpha!f \mid \alpha!(a)$$

$$f ::= x \mid (x) \mid a \mid \text{Int} \mid \text{Bool} \mid \cdots$$

${\sf Infinite\ behaviors} = {\sf infinite\ terms}$

- regularity
- boundedness

$$X = c?Int.X$$
$$= a?(x) (c|x 1|X)$$

$$T ::= \mathbf{0} \mid \mathbf{1} \qquad \pi.T \mid T+T \mid T \oplus T \qquad T \mid T \mid (\nu a)T$$

$$\pi ::= \alpha?f \mid \alpha!f \mid \alpha!(a) \longleftarrow \text{prefixes}$$

$$f ::= x \mid (x) \mid a \mid \text{Int} \mid \text{Bool} \mid \cdots$$

${\sf Infinite\ behaviors} = {\sf infinite\ terms}$

- regularity
- boundedness

$$X = c?Int.X$$

$$X = a?(x).(c!x.1 \mid X)$$

$$T ::= \mathbf{0} \mid \mathbf{1} \qquad \pi.T \mid T+T \mid T \oplus T \qquad T \mid T \mid (\boldsymbol{\nu}a)T$$

$$\pi ::= \alpha?f \mid \alpha!f \mid \alpha!(a)$$

$$f ::= x \mid (x) \mid a \mid \text{Int} \mid \text{Bool} \mid \cdots$$

$$\vdots$$
Infin patterns = sets of values and names + binders

- regularity
- boundedness

X = c?Int.X

$$X = a?(x).(c!x.\mathbf{1} \mid X)$$



$$T ::= \mathbf{0} \mid \mathbf{1} \qquad \pi.T \mid T + T \mid T \oplus T \qquad T \mid T \mid (\nu a)T$$

$$\pi ::= \alpha?f \mid \alpha!f \mid \alpha!(a)$$

$$f ::= x \mid (x) \mid a \mid \text{Int} \mid \text{Bool} \mid \cdots$$

Infinite behaviors = infinite terms

- regularity
- boundedness

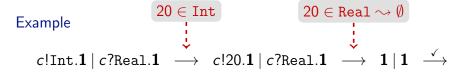
$$X = c?Int.X$$

$$X = a?(x).(c!x.\mathbf{1} \mid X)$$

Labeled operational semantics

$$1 \stackrel{\checkmark}{\longrightarrow} 1$$

$$\frac{m \in f \leadsto \sigma}{c?f.T \overset{c?m}{\longrightarrow} T\sigma} \qquad \frac{m \in f}{c!f.T \longrightarrow c!m.T} \qquad c!m.T \overset{c!m}{\longrightarrow} T$$



◆ロト ◆母 ト ◆ 豊 ト ◆ 豊 ト 多 へ で ...

Contracts as behavioral types

Systems

$$S \stackrel{\mathrm{def}}{=} T_1 \mid T_2 \mid \cdots \mid T_n$$

- when is a system well-formed?
- 2 when is a process well-typed?
- **3** when are two types equal?

Participant satisfaction

Definition

 $T \triangleleft S$ if $T \mid S \Longrightarrow T' \mid S'$ and $T' \longrightarrow$ implies

- $T' \xrightarrow{\mu_1}$ and $S' \stackrel{\mu_2}{\Longrightarrow}$
- $\mu_1 \# \mu_2$

 $(c!m \# c?m, \checkmark \# \checkmark)$

for some μ_1 and μ_2

Examples

- c!Int. $\mathbf{1} \triangleleft c?$ Real. $\mathbf{1}$
- c!Real. $\mathbf{1} \not < c?$ Int. $\mathbf{1}$ c!Real. $\mathbf{1} \mid c?$ Int. $\mathbf{1} \longrightarrow c!$ $\sqrt{2}.\mathbf{1} \mid c?$ Int. $\mathbf{1}$

stuck

Well-formed systems

$$S \stackrel{\mathrm{def}}{=} T_1 \mid T_2 \mid \cdots \mid T_n$$

Definition

S is well formed if

$$T_k \triangleleft \prod_{i \in \{1,\ldots,n\} \setminus \{k\}} T_i$$

for every $1 \le k \le n$

Examples

- c!Int. $1 \mid c?$ Real.1 is well formed
- c!Real.**1** | c?Int.**1** is ill formed

Well-typed participant

Definition

T is *viable* if $T \mid S$ is well formed for some S

Example

$$T \stackrel{\text{def}}{=} c? \text{Int.} \mathbf{1} + c? \text{Bool.} \mathbf{0}$$

 $S \stackrel{\text{def}}{=} c? \text{Int.} \mathbf{0} + c? \text{Bool.} \mathbf{1}$

- T is viable
- *S* is viable
- $T \oplus S$ is not viable

Example: global order on channels

$$P \stackrel{\text{def}}{=} a?(x).b?(y).x!3.x?(z:\text{Int}).y!\text{true}.0$$

$$P' \stackrel{\text{def}}{=} a?(x).b?(y).x!3.y!\text{true}.x?(z.\text{Int}).0$$

$$Q \stackrel{\text{def}}{=} a!(c).b!(d).c?(z:\text{Int}).d?(z:\text{Bool}).c!5.0$$

$$Q' \stackrel{\text{def}}{=} a!(c).b!(d).c?(z:\text{Int}).c!5.d?(z':\text{Bool}).0$$

- deadlock because of cyclic dependency
- T_P | T_Q ill-formed (not viable!)



Example: global order on channels

$$P \stackrel{\text{def}}{=} a?(x).b?(y).x!3.x?(z:\text{Int}).y!\text{true}.0$$

$$P' \stackrel{\text{def}}{=} a?(x).b?(y).x!3.y!\text{true}.x?(z:\text{Int}).0$$

$$Q \stackrel{\text{def}}{=} a!(c).b!(d).c?(z:\text{Int}).\psi?(z:\text{Bool}).c!5.0$$

$$Q' \stackrel{\text{def}}{=} a!(c).b!(d).c?(z:\text{Int}).c!5.d?(z':\text{Bool}).0$$

- imposing global order
- $T_P \mid T_{Q'}$ well-formed

Example: global order on channels

$$P \stackrel{\text{def}}{=} a?(x).b?(y).x!3.x?(z : Int).y!true.0$$

$$P' \stackrel{\text{def}}{=} a?(x).b?(y).x!3.y!true.x?(z : Int).0$$

$$Q \stackrel{\text{def}}{=} a!(c).b!(d).c?(z : Int).d?(z : Bool).c!5.0$$

$$Q' \stackrel{\text{def}}{=} a!(c).b!(d).c?(z : Int).c!5.d?(z' : Bool).0$$

- global order is not necessary
- $T_{P'} \mid T_Q$ well-formed

Example: linearity

Subcontract

Definition

 $T \leq S$ if $T \mid R$ well formed implies $S \mid R$ well formed for every R

Examples

- $T \oplus S \prec T$
- $\pi.T + \pi.S \approx \pi.(T \oplus S)$
 - ...very much like the *must* preorder ...
- **0** ≤ *T*



$$\leq$$
 is **not** a precongruence

$$\mathbf{0} \preceq T$$

Definition (strong subcontract)

Let \sqsubseteq be the largest precongruence included in \preceq

Theorem

If T is viable, then $T \leq S$ iff $T \sqsubseteq S$

- $T \sqsubseteq \mathbf{0}$ iff T is not viable
- if $\mathbf{1} + T \sqsubseteq T$, then T is well formed
- π.0 □ π.T

Outline

- Motivation
 Protocols and processes
 Contracts and mobile systems
- Contracts
 Syntax
 Semantics
- Results
- 4 Concluding remarks

On progress

Theorem

If $\vdash P : T$ and T w.f. and $P \stackrel{\tau}{\Longrightarrow} Q \stackrel{\tau}{\leadsto}$, then Q has succeeded

success = "no pending actions"

On decidability

Proposition

- well-formedness
- viability
- subcontract

are decidable provided that c!f matches finitely many names

If a name is sent:

- either it is fresh
- or it is a public name
- or it was received earlier

c!(a)

Outline

- Motivation
 Protocols and processes
 Contracts and mobile systems
- 2 Contracts
 Syntax
 Semantics
- Results
- 4 Concluding remarks

Session types and contracts: a comparison

- optimistic vs conservative
- global vs compositional

	Session types	Contracts
structuring	++	
analysis		++

Concluding remarks

Contributions

- contracts for processes with channel mobility
- straightforward solution to global progress (of bounded systems)

Our wish list

- algorithms (almost done)
- choreographic specifications
- expressiveness

Concluding remarks

Contributions

- contracts for processes with channel mobility
- straightforward solution to global progress (of bounded systems)

Our wish list

- algorithms (almost done)
- choreographic specifications
- expressiveness

Thank you.



Regular does not mean finite-state

- unbounded participants
- unbounded buffers
- state encoded within processes

$$P(x: Int) = deposit?(y: Int).P(x + y) + withdraw?(y: Int).P(max{0, x - y})$$

$$P(0)$$

$$P = c?(x: Int).(deposit?(y: Int).c!\langle x + y \rangle.P + withdraw?(y: Int).c!\langle max{0, x - y} \rangle.P)$$

$$Q = c?(x: Int).c!\langle x \rangle.Q$$

 $(\nu c)(P \mid c!\langle 0 \rangle.Q)$

Simulating asynchrony

INPUT
$$\frac{\Gamma \vdash \alpha : \text{Ch} \qquad \Gamma, x : t \vdash P : T}{\Gamma \vdash \alpha?(x : t).P : \alpha?t.T + \alpha? \neg t.\mathbf{0}}$$

$$\frac{\text{C-RECV}}{\Gamma \vdash \alpha : \text{Ch}} \qquad \Gamma, x : \text{Ch} \vdash P : T}{\Gamma \vdash \alpha?(x).P : \alpha?(x).T + \alpha? \neg \text{Ch}.\mathbf{0}}$$