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

CONCUR 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
$? {\tt Int.?Int.} \big(!{\tt Real} \oplus !{\tt Error} \big)$	$a.a.(\overline{b}\oplus\overline{c})$
?(!Bool.!Bool)	a

A natural candidate

Contracts without channel passing ⇒ ccs

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

A problem of abstraction

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

$$? (! \texttt{Bool.!Bool}) \qquad \textit{a}$$

A natural candidate

Contracts without channel passing \Rightarrow ccs

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

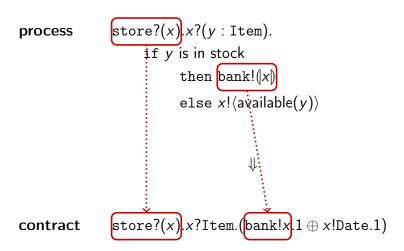
An example

```
process store?(x).x?(y:Item).  
if y is in stock  
then bank!(|x|)  
else x!\langle available(y)\rangle
```

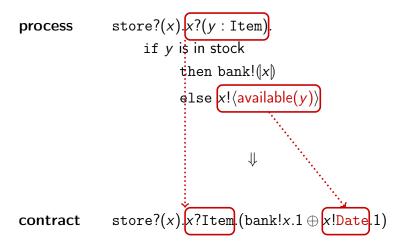
 \downarrow

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

An example



An example



Some typing rules

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

$$\frac{\text{c-send}}{\Gamma \vdash P : T} \frac{\Gamma \vdash \alpha! (|\beta|) \cdot P : \alpha! \beta \cdot T}{\Gamma \vdash \alpha! (|\beta|) \cdot P : \alpha! \beta \cdot T}$$

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

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

Some typing rules

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

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

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

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

undecidable → decidable

Outline

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

failure, success

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

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

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

- regularity
- boundedness

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

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

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

- regularity
- boundedness

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



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

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

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

- regularity
- boundedness

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



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

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

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

- regularity
- boundedness

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



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

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

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

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

- regularity
 - boundedness

X = c?Int.X

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



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

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

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

- regularity
- boundedness

$$X = c$$
?Int. X

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



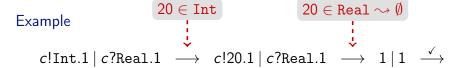
Labeled operational semantics

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

$$\frac{m \in f \leadsto \sigma}{c?f.T \xrightarrow{c?m} T\sigma}$$

$$\frac{m \in f}{c!f.T \longrightarrow c!m.T}$$

$$c!m.T \stackrel{c!m}{\longrightarrow} T$$



Contracts as behavioral types

Systems

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

- 1 when is a system well-formed?
- 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.1 \triangleleft c?Real.1
- c!Real.1 \checkmark c?Int.1 c!Real.1 | c?Int.1 \longrightarrow c! $\sqrt{2}$.1 | c?Int.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

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

for every $1 \le k \le n$

Examples

- c!Int.1 | 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

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

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

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

Example: global order on channels

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

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

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

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

- deadlock because of cyclic dependency
- $T_P \mid T_Q$ ill-formed (not viable!)

Example: global order on channels

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

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

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

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

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

Example: global order on channels

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

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

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

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

- global order is not necessary
- $T_{P'} \mid T_O$ 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

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



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

 $0 \leq T$

Definition (strong subcontract)

Let \sqsubseteq be the largest precongruence included in \preceq

Theorem

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

- $T \sqsubseteq 0$ iff T is not viable
- if $1 + T \sqsubseteq T$, 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
- 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

- 1 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

P(x: Int) = deposit?(y: Int).P(x + y)

- unbounded participants
- unbounded buffers
- state encoded within processes

$$P(0)$$

$$P(0)$$

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

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

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

Simulating asynchrony

input

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

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