Retractable Contracts And Beyond

Ivan Lanese
Computer Science Department
University of Bologna/INRIA
Italy

Joint work with Franco Barbanera,
Mariangiola Dezani-Ciancaglini
and Ugo de'Liguoro

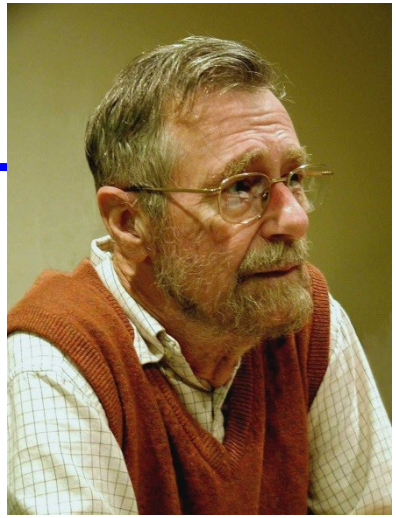
Map of the talk

- Why retractable contracts?
- What is a retractable contract?
- What is a speculative contract?
- What is beyond?

Map of the talk

- Why retractable contracts?
- What is a retractable contract?
- What is a speculative contract?
- What is beyond?

Undoing things considered harmful



- Undo operations are useful and widespread
  - Undo command in your favorite editor
  - Back button in your favorite browser
  - Restore a past backup
- In interactions (unilateral) undo may lead to unpredictable or undesired results
  - What happens if you press the back button of the browser while reserving a flight?
  - You don’t want a client to be able to undo her payment after a purchase
- Undo activities must be disciplined

Contracts

- A (binary) contract is the abstract description of the behaviour of a client or a server
- A client complies with a server if all her requirements are fulfilled
  - by reaching a distinguished satisfaction state or
  - by running an infinite interaction without ever getting stuck
- A client that does not comply with its server may get stuck

Retractable contracts

- We start from binary contracts
- Getting stuck may depend on wrong choices taken during the interaction
- Going back to past choices and trying different paths may solve the problem
- This will “facilitate” compliance
- We explore a notion of contracts where past decisions are stored and can be undone

Map of the talk

- Why retractable contracts?
- What is a retractable contract?
- What is a speculative contract?
- What is beyond?

Retractable contracts: syntax

- $\sigma, \rho ::= 1$ — success
  - $\sum_{i \in I} a_i . \sigma_i$ — external input choice
  - $\oplus_{i \in I} \overline{a}_i . \sigma_i$ — internal output choice
  - $\sum_{i \in I} \overline{a}_i . \sigma_i$ — retractable output choice
  - $\oplus_{i \in I} a_i . \sigma_i$ — internal input choice
  - x — variable
  - $rec\, x . \sigma$ — recursion

- We add $\circ$ (no more alternatives) to contracts σ
- Histories are stacks of contracts $h ::= [\,] \mid h : \sigma$
- Contracts with history: $h \prec \sigma$

Motivating problem

- A buyer wants to buy either a bag or a belt
- She will decide whether to pay by card or cash after knowing the price

$$Buyer = \overline{bag}.price.(\overline{card} \oplus \overline{cash}) \oplus \overline{belt}.price.(\overline{card} \oplus \overline{cash})$$

- The seller accepts cards only for bags, not for belts

$$Seller = bag.\overline{price}.(card + cash) + belt.\overline{price}.cash$$

- Buyer and seller are not compliant
- They become compliant if we make, e.g., the buyer choice between bag and belt retractable
- The buyer is still able to pay a belt with card if interacting with a seller allowing this

Motivating problem

- A buyer wants to buy either a bag or a belt
- She will decide whether to pay by card or cash after knowing the price

$$Buyer = \overline{bag}\,.\,price\,.\left(\overline{card} \oplus \overline{cash}\right) + \overline{belt}\,.\,price\,.\left(\overline{card} \oplus \overline{cash}\right)$$

- The seller accepts cards only for bags, not for belts

$$Seller = bag\,.\,\overline{price}\,.\left(card + cash\right) + belt\,.\,\overline{price}\,.\,cash$$

- Buyer and seller are not compliant
- They become compliant if we make, e.g., the buyer choice between bag and belt retractable
- The buyer is still able to pay a belt with card if interacting with a seller allowing this

Retractable contracts: semantics

- Contracts are executed as usual but…
- … branches in external choices which are not selected are stored in the history
- When the interaction is stuck, both client and server can pop from the history the last state

Sample computation

- Buyer =

  $$[\,] \prec \overline{bag}\,.\,price\,.\left(\overline{card} \oplus \overline{cash}\right) + \overline{belt}\,.\,price\,.\left(\overline{card} \oplus \overline{cash}\right)$$

- Seller =

  $$[\,] \prec bag\,.\,\overline{price}\,.\left(card + cash\right) + belt\,.\,\overline{price}\,.\,cash$$

Sample computation

- Buyer =

$$[\,] \prec \overline{bag}.price.(\overline{card} \oplus \overline{cash}) + \overline{belt}.price.(\overline{card} \oplus \overline{cash})$$

$$\rightarrow [\,]:\overline{bag}.price.(\overline{card} \oplus \overline{cash}) \prec price.(\overline{card} \oplus \overline{cash})$$

- Seller =

$$[\,] \prec bag.\overline{price}.(card + cash) + belt.\overline{price}.cash$$

$$\rightarrow [\,]:bag.\overline{price}.(card + cash) \prec \overline{price}.cash$$

Sample computation

- Buyer =

$$[\,] \prec \overline{bag}.\,price.(\overline{card} \oplus \overline{cash}) + \overline{belt}.\,price.(\overline{card} \oplus \overline{cash})$$

$$\rightarrow [\,]:\overline{bag}.\,price.(\overline{card} \oplus \overline{cash}) \prec price.(\overline{card} \oplus \overline{cash})$$

$$\rightarrow [\,]:\overline{bag}.\,price.(\overline{card} \oplus \overline{cash}):\circ \prec \overline{card} \oplus \overline{cash}$$

- Seller =

$$[\,] \prec bag.\,\overline{price}.(card + cash) + belt.\,\overline{price}.\,cash$$

$$\rightarrow [\,]:bag.\,\overline{price}.(card + cash) \prec \overline{price}.\,cash$$

$$\rightarrow [\,]:bag.\,\overline{price}.(card + cash):\circ \prec cash$$

Sample computation

- Buyer =

  $[\,]:\overline{bag}\,.\,price\,.\left(\overline{card}\oplus\overline{cash}\right):\circ \;\prec\; \overline{card}\oplus\overline{cash}$

- Seller =

  $[\,]:bag\,.\,\overline{price}\,.\left(card+cash\right):\circ \;\prec\; cash$

Sample computation

- Buyer =

  $[\,]:\overline{bag}\,.\,price\,.\,(\overline{card}\oplus\overline{cash}):\circ \;\prec\; \overline{card}\oplus\overline{cash}$

→ $[\,]:\overline{bag}\,.\,price\,.\,(\overline{card}\oplus\overline{cash}):\circ \;\prec\; \overline{card}$

- Seller =

  $[\,]:bag\,.\,\overline{price}\,.\,(card+cash):\circ \;\prec\; cash$

Sample computation

- Buyer =

  $[\,]:\overline{bag}\,.\,price\,.\,(\overline{card}\oplus\overline{cash}):\circ \;\prec\; \overline{card}\oplus\overline{cash}$

→ $[\,]:\overline{bag}\,.\,price\,.\,(\overline{card}\oplus\overline{cash}):\circ \;\prec\; \overline{card}$

Interaction is stuck

- Seller =

  $[\,]:bag\,.\,\overline{price}\,.\,(card+cash):\circ \;\prec\; cash$

Sample computation

- Buyer =

  $[\,]:\overline{bag}\,.\,price\,.\left(\overline{card}\oplus\overline{cash}\right):\circ \;\prec\; \overline{card}$

- Seller =

  $[\,]:bag\,.\,\overline{price}\,.\left(card+cash\right):\circ \;\prec\; cash$

Sample computation

- Buyer =

$$[\,]:\overline{bag}\,.\,price\,.\left(\overline{card}\oplus\overline{cash}\right):\circ \;\prec\; \overline{card}$$

$$\rightarrow\; [\,]:\overline{bag}\,.\,price\,.\left(\overline{card}\oplus\overline{cash}\right) \;\prec\; \circ$$

- Seller =

$$[\,]:bag\,.\,\overline{price}\,.\left(card+cash\right):\circ \;\prec\; cash$$

$$\rightarrow\; [\,]:bag\,.\,\overline{price}\,.\left(card+cash\right) \;\prec\; \circ$$

Sample computation

- Buyer =

$$[\,]:\overline{bag}\,.\,price\,.\,(\overline{card}\oplus\overline{cash}):\circ \;\prec\; \overline{card}$$

$$\rightarrow\; [\,]:\overline{bag}\,.\,price\,.\,(\overline{card}\oplus\overline{cash}) \;\prec\; \circ$$

$$\rightarrow\; [\,] \;\prec\; \overline{bag}\,.\,price\,.\,(\overline{card}\oplus\overline{cash})$$

- Seller =

$$[\,]:bag\,.\,\overline{price}\,.\,(card+cash):\circ \;\prec\; cash$$

$$\rightarrow\; [\,]:bag\,.\,\overline{price}\,.\,(card+cash) \;\prec\; \circ$$

$$\rightarrow\; [\,] \;\prec\; bag\,.\,\overline{price}\,.\,(card+cash)$$

Sample computation

- Buyer =

  $$[\,] \prec \overline{bag}\,.\,price\,.\left(\overline{card} \oplus \overline{cash}\right)$$

- Seller =

  $$[\,] \prec bag\,.\,\overline{price}\,.\left(card + cash\right)$$

Sample computation

- Buyer =

$$[\,] \prec \overline{bag}.\,price.\left(\overline{card} \oplus \overline{cash}\right)$$

$$\rightarrow [\,]:\circ \prec price.\left(\overline{card} \oplus \overline{cash}\right)$$

- Seller =

$$[\,] \prec bag.\,\overline{price}.\left(card + cash\right)$$

$$\rightarrow [\,]:\circ \prec \overline{price}.\left(card + cash\right)$$

Sample computation

- Buyer =

$$[\,] \prec \overline{bag}.\,price.\left(\overline{card} \oplus \overline{cash}\right)$$

$$\rightarrow [\,]:\circ \prec price.\left(\overline{card} \oplus \overline{cash}\right)$$

$$\rightarrow [\,]:\circ:\circ \prec \overline{card} \oplus \overline{cash}$$

- Seller =

$$[\,] \prec bag.\,\overline{price}.\left(card + cash\right)$$

$$\rightarrow [\,]:\circ \prec \overline{price}.\left(card + cash\right)$$

$$\rightarrow [\,]:\circ:\circ \prec card + cash$$

Sample computation

- Buyer =

  $[\,]:\circ:\circ \;\prec\; \overline{card} \oplus \overline{cash}$

- Seller =

  $[\,]:\circ:\circ \;\prec\; card + cash$

Sample computation

- Buyer =

  $[\,]:\circ:\circ \;\prec\; \overline{card} \oplus \overline{cash}$

→ $[\,]:\circ:\circ \;\prec\; \overline{card}$

- Seller =

  $[\,]:\circ:\circ \;\prec\; card + cash$

Sample computation

- Buyer =

  $[\,]:\circ:\circ \;\prec\; \overline{card} \oplus \overline{cash}$

→ $[\,]:\circ:\circ \;\prec\; \overline{card}$

→ $[\,]:\circ:\circ:\circ \;\prec\; 1$

- Seller =

  $[\,]:\circ:\circ \;\prec\; card + cash$

  $[\,]:\circ:\circ:cash \;\prec\; 1$

Compliance

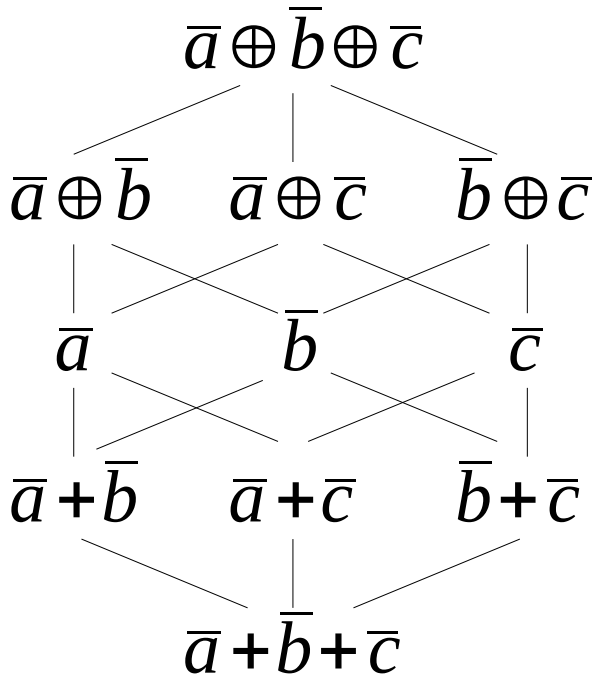
- We can use the standard notion of compliance
  - If the computation gets stuck, then the client is satisfied
- We can define a formal system to decide compliance
- The main novelty is that two external choices are compliant iff there exists a compatible branch

$$\frac{\Gamma, \alpha.\rho+\rho' \dashv \overline{\alpha}.\sigma+\sigma' \ \rhd \ \rho \dashv \sigma}{\Gamma \ \rhd \ \alpha.\rho+\rho' \dashv \overline{\alpha}.\sigma+\sigma'}$$

- The formal system is correct, complete and terminating, hence it can be transformed into a procedure
- The procedure requires $O(n^5)$
  - More than for standard contracts

Duality and subcontract relation

- From the notion of compliance we can define a notion of subcontract
  - Replacing a contract with a subcontract preserves compliance
- The syntactic dual is also a semantic dual
  - The more general compliant contract



Map of the talk

- Why retractable contracts?
- What is a retractable contract?
- What is a speculative contract?
- What is beyond?

Speculative contracts

- We start again from binary contracts
- For efficiency reasons one may want to try different options concurrently
- As soon as one of them succeeds, the whole computation is successful
- We use the same syntax that we used for retractable contracts
  - Now external choice among outputs has the speculative behaviour above
  - External since the environment can slow down undesired paths selecting the one he wants to succeed

Speculative contracts: results

- The decision procedure for retractable contracts and for speculative contracts coincide
- As a consequence, all the results about compliance, duality and subcontract apply

Map of the talk

- Why retractable contracts?
- What is a retractable contract?
- What is a speculative contract?
- What is beyond?

Summary

- We presented a model of contracts with retractable choice
  - Compliance
  - Subcontract relation
  - Duality
- Simple and neat extension of the theory of binary contracts
- Using retractable choice instead of normal choice extends the set of compliant contracts
- The same theory captures also speculative execution

What is beyond?

- Explore the notion of retractable contracts in multiparty sessions
- Are there other meaningful ways to exploit contracts/behavioural types to control reversibility?
- Are there other useful computational patterns that can be tamed using contracts?

End of talk

Thanks!

Questions?

Most related work

Franco Barbanera, Mariangiola Dezani-Ciancaglini, Ugo de'Liguoro: Compliance for reversible client/server interactions. BEAT 2014
also considered contracts with rollback

BEAT 2014	vs	PLACES 2015
Free rollback	vs	rollback only when stuck
Explicit checkpoint	vs	implicit checkpoint
One checkpoint	vs	stack of checkpoints
Compliance harder	vs	compliance easier