Retractable and Speculative Contracts

Ivan Lanese
Computer Science Department
University of Bologna/INRIA
Italy

Joint work with Franco Barbanera
and Ugo de'Liguoro

Map of the talk

- What retractable/speculative contracts are?
- Motivating example
- Results
- Conclusion

Map of the talk

- What retractable/speculative contracts are?
- Motivating example
- Results
- Conclusion

Contracts

- A contract is the abstract description of the behavior of either 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 without succeeding
- Compliance is statically decidable

Beyond classical contracts

- Contracts describing basic client/server interactions have been introduced in 2006 by Carpineti, Castagna, Laneve and Padovani
- We want to consider two features of (some) interacting systems not covered by classic contracts:
  - rollback, enabling one to go back to past states of the interaction till a successful path is found
  - speculation, enabling one to try different paths concurrently till a successful path is found

Why retractable contracts?

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

Why speculative contracts?

- Speculation is used for performance reasons in many contexts
  - Simulation, thread-level optimizations, web services
- Do these optimizations preserve correctness?
  - Not trivial, think to all the issues related to weak memory models
- Also speculation activities must be disciplined

Retractable/speculative contracts: syntax

- Retractable and speculative contracts have very different origin and aim
- Yet we describe both of them with the same syntax (but different semantics)

$\sigma ::=$

1	success	Standard contracts
$\bigoplus_{i \in I} \overline{a}_i.\sigma_i$	internal output choice	
$\sum_{i \in I} a_i.\sigma_i$	external input choice	
X	variable	
$\mathrm{rec}\ X.\sigma$	recursion	
$\bigoplus_{i \in I} a_i.\sigma_i$	internal input choice	
$\sum_{i \in I} \overline{a}_i.\sigma_i$	retractable/speculative output choice	

Retractable/speculative contracts: main idea

- The peculiar operator is retractable/speculative output choice:
  $\Sigma_{i \in I} \overline{a}_i.\sigma_i$
- In the retractable semantics it behaves as follows:
  - we perform an output, but other options are stored
  - if the computation gets stuck, the choice is undone and we try another option
- In the speculative semantics it behaves as follows:
  - we perform an output, but other options are not discarded and can be activated in parallel threads

Map of the talk

- What retractable/speculative contracts are?
- Motivating example
- Results
- Conclusion

Retractable contracts: history information

- To give semantics to retractable contracts we need history information
- We add $\circ$ (no alternatives left) 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 = $\overline{\text{bag}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) \oplus \overline{\text{belt}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$
- The seller accepts cards only for bags, not for belts
- Seller = $\text{bag}.\overline{\text{price}}.(\text{card} + \text{cash}) + \text{belt}.\overline{\text{price}}.\text{cash}$
- Buyer and seller are not compliant

Reversibility to the rescue

- Buyer = $\overline{\text{bag}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) \oplus \overline{\text{belt}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$
- Seller = $\text{bag}.\overline{\text{price}}.(\text{card} + \text{cash}) + \text{belt}.\overline{\text{price}}.\text{cash}$
- They become compliant if we make the buyer choice between bag and belt retractable
  - Or the one between card and cash (for belt)
- The buyer is still able to pay a belt with card if interacting with a seller allowing this
- Retractable choice “facilitates” compliance

Reversibility to the rescue

- Buyer = $\overline{\text{bag}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) + \overline{\text{belt}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$
- Seller = $\text{bag}.\overline{\text{price}}.(\text{card} + \text{cash}) + \text{belt}.\overline{\text{price}}.\text{cash}$
- They become compliant if we make the buyer choice between bag and belt retractable
  - Or the one between card and cash (for belt)
- The buyer is still able to pay a belt with card if interacting with a seller allowing this
- Retractable choice “facilitates” compliance

Sample retractable computation

- Buyer' =
  $[] \prec \overline{\text{bag}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) + \overline{\text{belt}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$

- Seller =
  $[] \prec \text{bag}.\overline{\text{price}}.(\text{card} + \text{cash}) + \text{belt}.\overline{\text{price}}.\text{cash}$

Sample retractable computation

- Buyer' =
  $[] \prec \overline{\text{bag}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) + \overline{\text{belt}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$
- $\overline{\text{bag}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) \prec \text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$

- Seller =
  $[] \prec \text{bag}.\overline{\text{price}}.(\text{card} + \text{cash}) + \text{belt}.\overline{\text{price}}.\text{cash}$
- $\text{bag}.\overline{\text{price}}.(\text{card} + \text{cash}) \prec \overline{\text{price}}.\text{cash}$

Sample retractable computation

- Buyer' =
  $[] \prec \overline{bag}.price.(\overline{card} \oplus \overline{cash}) + \overline{belt}.price.(\overline{card} \oplus \overline{cash})$
- $\overline{bag}.price.(\overline{card} \oplus \overline{cash}) \prec price.(\overline{card} \oplus \overline{cash})$
- $\overline{bag}.price.(\overline{card} \oplus \overline{cash}) : \circ \prec \overline{card} \oplus \overline{cash}$

- Seller =
  $[] \prec bag.\overline{price}.(card + cash) + belt.\overline{price}.cash$
- $bag.\overline{price}.(card + cash) \prec \overline{price}.cash$
- $bag.\overline{price}.(card + cash) : \circ \prec cash$

Sample retractable computation

- Buyer' =
  $[] \prec \overline{bag}.price.(\overline{card} \oplus \overline{cash}) + \overline{belt}.price.(\overline{card} \oplus \overline{cash})$
- $\overline{bag}.price.(\overline{card} \oplus \overline{cash}) \prec price.(\overline{card} \oplus \overline{cash})$
- $\overline{bag}.price.(\overline{card} \oplus \overline{cash}) : \circ \prec \overline{card} \oplus \overline{cash}$
- $\overline{bag}.price.(\overline{card} \oplus \overline{cash}) : \circ \prec \overline{card}$

- Seller =
  $[] \prec bag.\overline{price}.(card + cash) + belt.\overline{price}.cash$
- $bag.\overline{price}.(card + cash) \prec \overline{price}.cash$
- $bag.\overline{price}.(card + cash) : \circ \prec cash$

Sample retractable computation

- Buyer' =
  $\overline{\text{bag}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) : \circ \prec \overline{\text{card}}$

- Seller =
  $\text{bag}.\overline{\text{price}}.(\text{card} + \text{cash}) : \circ \prec \text{cash}$

Sample retractable computation

- Buyer' =
  $\overline{\text{bag}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) : \circ \prec \overline{\text{card}}$
- $\overline{\text{bag}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) \prec \circ$

- Seller =
  $\text{bag}.\overline{\text{price}}.(\text{card} + \text{cash}) : \circ \prec \text{cash}$
- $\text{bag}.\overline{\text{price}}.(\text{card} + \text{cash}) \prec \circ$

Sample retractable computation

- Buyer' =
  $\overline{bag}.price.(\overline{card} \oplus \overline{cash}) : \circ \prec \overline{card}$
- $\overline{bag}.price.(\overline{card} \oplus \overline{cash}) \prec \circ$
- $[] \prec \overline{bag}.price.(\overline{card} \oplus \overline{cash})$

- Seller =
  $bag.\overline{price}.(card + cash) : \circ \prec cash$
- $bag.\overline{price}.(card + cash) \prec \circ$
- $[] \prec bag.\overline{price}.(card + cash)$

Sample retractable computation

- Buyer' =
  $[] \prec \overline{\text{bag}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$

- Seller =
  $[] \prec \text{bag}.\overline{\text{price}}.(\text{card} + \text{cash})$

Sample retractable computation

- Buyer' =
  $[] \prec \overline{bag}.price.(\overline{card} \oplus \overline{cash})$
- $\circ \prec price.(\overline{card} \oplus \overline{cash})$

- Seller =
  $[] \prec bag.\overline{price}.(card + cash)$
- $\circ \prec \overline{price}.(card + cash)$

Sample retractable computation

- Buyer' =
  $[] \prec \overline{bag}.price.(\overline{card} \oplus \overline{cash})$
- $\circ \prec price.(\overline{card} \oplus \overline{cash})$
- $\circ : \circ \prec \overline{card} \oplus \overline{cash}$

- Seller =
  $[] \prec bag.\overline{price}.(card + cash)$
- $\circ \prec \overline{price}.(card + cash)$
- $\circ : \circ \prec card + cash$

Sample retractable computation

- Buyer' =
  $\circ : \circ \prec \overline{\text{card}} \oplus \overline{\text{cash}}$

- Seller =
  $\circ : \circ \prec \text{card} + \text{cash}$

Sample retractable computation

- Buyer' =
  - $\circ : \circ \prec \overline{\text{card}} \oplus \overline{\text{cash}}$
- $\circ : \circ \prec \overline{\text{card}}$

- Seller =
  - $\circ : \circ \prec \text{card} + \text{cash}$

Sample retractable computation

- Buyer' =
  - $\circ : \circ \prec \overline{\text{card}} \oplus \overline{\text{cash}}$
- $\circ : \circ \prec \overline{\text{card}}$
- $\circ : \circ : \circ \prec 1$

- Seller =
  - $\circ : \circ \prec \text{card} + \text{cash}$
- $\circ : \circ : \text{cash} \prec 1$

Example under the speculative semantics

- At runtime contracts are composed by multiple threads
  - We use parallel composition |
- Each thread is identified by a unique prefix obtained by composing past actions $a_1@...@a_2@\sigma$
- Only threads with dual prefix can interact

Sample speculative computation

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

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

Sample speculative computation

- Buyer' =
  $\overline{\text{bag}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) + \overline{\text{belt}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$
- $\overline{\text{bag}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) \mid \overline{\text{belt}}@\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$

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

Sample speculative computation

- Buyer' =
  $\overline{\text{bag}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) + \overline{\text{belt}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$
- $\overline{\text{bag}}.\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) \mid \overline{\text{belt}}@\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$
- $\overline{\text{bag}}@\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) \mid \overline{\text{belt}}@\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$

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

Sample speculative computation

- Buyer' =
  $\overline{\text{bag}}@\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) \mid \overline{\text{belt}}@\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$

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

Sample speculative computation

- Buyer' =
  $\overline{\text{bag}}@\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}}) \mid \overline{\text{belt}}@\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$
- $\overline{\text{bag}}@\text{price}@(\overline{\text{card}} \oplus \overline{\text{cash}}) \mid \overline{\text{belt}}@\text{price}.(\overline{\text{card}} \oplus \overline{\text{cash}})$

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

Sample speculative computation

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

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

Sample speculative computation

- Buyer' =
  $\overline{\text{bag}}@\text{price}@(\overline{\text{card}} \oplus \overline{\text{cash}}) \mid \overline{\text{belt}}@\text{price}@(\overline{\text{card}} \oplus \overline{\text{cash}})$

- Seller =
  $\text{bag}@\overline{\text{price}}@(\text{card} + \text{cash}) \mid \text{belt}@\overline{\text{price}@\text{cash}}$

Sample speculative computation

- Buyer' =
  $\overline{\text{bag}}@\text{price}@(\overline{\text{card}} \oplus \overline{\text{cash}}) \mid \overline{\text{belt}}@\text{price}@(\overline{\text{card}} \oplus \overline{\text{cash}})$
- $\overline{\text{bag}}@\text{price}@\overline{\text{card}} \mid \overline{\text{belt}}@\text{price}@(\overline{\text{card}} \oplus \overline{\text{cash}})$

- Seller =
  $\text{bag}@\overline{\text{price}}@(\text{card} + \text{cash}) \mid \text{belt}@\overline{\text{price}}@\text{cash}$

Sample speculative computation

- Buyer' =
  $\overline{\text{bag}}@\text{price}@(\overline{\text{card}} \oplus \overline{\text{cash}}) \mid \overline{\text{belt}}@\text{price}@(\overline{\text{card}} \oplus \overline{\text{cash}})$
  - $\overline{\text{bag}}@\text{price}@\overline{\text{card}} \mid \overline{\text{belt}}@\text{price}@(\overline{\text{card}} \oplus \overline{\text{cash}})$
  - $\overline{\text{bag}}@\text{price}@\overline{\text{card}} \mid \overline{\text{belt}}@\text{price}@\overline{\text{card}}$

- Seller =
  $\text{bag}@\overline{\text{price}}@(\text{card} + \text{cash}) \mid \text{belt}@\overline{\text{price}}@\text{cash}$

Sample speculative computation

- Buyer' = $\overline{\text{bag}}@\text{price}@(\overline{\text{card}} \oplus \overline{\text{cash}}) \mid \overline{\text{belt}}@\text{price}@(\overline{\text{card}} \oplus \overline{\text{cash}})$
  - $\overline{\text{bag}}@\text{price}@\overline{\text{card}} \mid \overline{\text{belt}}@\text{price}@(\overline{\text{card}} \oplus \overline{\text{cash}})$
  - $\overline{\text{bag}}@\text{price}@\overline{\text{card}} \mid \overline{\text{belt}}@\text{price}@\overline{\text{card}}$
  - $\overline{\text{bag}}@\text{price}@\overline{\text{card}}@1 \mid \overline{\text{belt}}@\text{price}@\overline{\text{card}}$

- Seller = $\text{bag}@\overline{\text{price}}@(\text{card} + \text{cash}) \mid \text{belt}@\overline{\text{price}}@\text{cash}$
  - $\text{bag}@\overline{\text{price}}@\text{card}@1 \mid \text{bag}@\overline{\text{price}}@\text{cash} \mid \text{belt}@\overline{\text{price}}@\text{cash}$

Map of the talk

- What retractable/speculative contracts are?
- Motivating example
- Results
- Conclusion

Compliance

- The retractable compliance relation $h \prec \sigma \dashv\!\!\| \; k \prec \rho$ holds iff

  $h \prec \sigma \,\|\, k \prec \rho \rightarrow^{*} h' \prec \sigma' \,\|\, k' \prec \rho' \not\mapsto$ implies $\sigma' = 1$

  - If the computation stops then the client is satisfied

- The retractable compliance relation on contracts is obtained by executing them with an empty history

- The speculative compliance relation holds iff if the computation stops then at least one of the threads of the client is in the success state 1

Main result

- The retractable compliance and the speculative compliance do coincide
- At first sight surprising, since they have different definitions and work on different semantics
- Intuition: both require the existence of a successful path
  - Two implementations of angelic nondeterminism
- Whether alternatives are explored sequentially or in parallel does not make a difference
- Consequence: all the results we derive from the compliance hold on both the settings

Compliance: decidability

- Compliance is decidable even for contracts with recursion
- We use judgments of the form $\Gamma \rhd \rho \sim| \sigma$

$$\frac{}{\Gamma \rhd \mathbf{1} \backsim| \sigma}\ (\textsc{Ax}) \qquad \frac{}{\Gamma, \rho \backsim| \sigma \rhd \rho \backsim| \sigma}\ (\textsc{Hyp}) \qquad \frac{\Gamma, \alpha.\rho + \rho' \backsim| \overline{\alpha}.\sigma + \sigma' \rhd \rho \backsim| \sigma}{\Gamma \rhd \alpha.\rho + \rho' \backsim| \overline{\alpha}.\sigma + \sigma'}\ (+ \cdot +)$$

$$\frac{\forall h \in I.\ \Gamma, \bigoplus_{i \in I} \overline{\alpha}_i.\rho_i \backsim| \sum_{j \in I \cup J} \alpha_j.\sigma_j \rhd \rho_h \backsim| \sigma_h}{\Gamma \rhd \bigoplus_{i \in I} \overline{\alpha}_i.\rho_i \backsim| \sum_{j \in I \cup J} \alpha_j.\sigma_j}\ (\oplus \cdot +)$$

$$\frac{\forall h \in I.\ \Gamma, \sum_{j \in I \cup J} \overline{\alpha}_j.\rho_j \backsim| \bigoplus_{i \in I} \alpha_i.\sigma_i \rhd \rho_h \backsim| \sigma_h}{\Gamma \rhd \sum_{j \in I \cup J} \overline{\alpha}_j.\rho_j \backsim| \bigoplus_{i \in I} \alpha_i.\sigma_i}\ (+ \cdot \oplus)$$

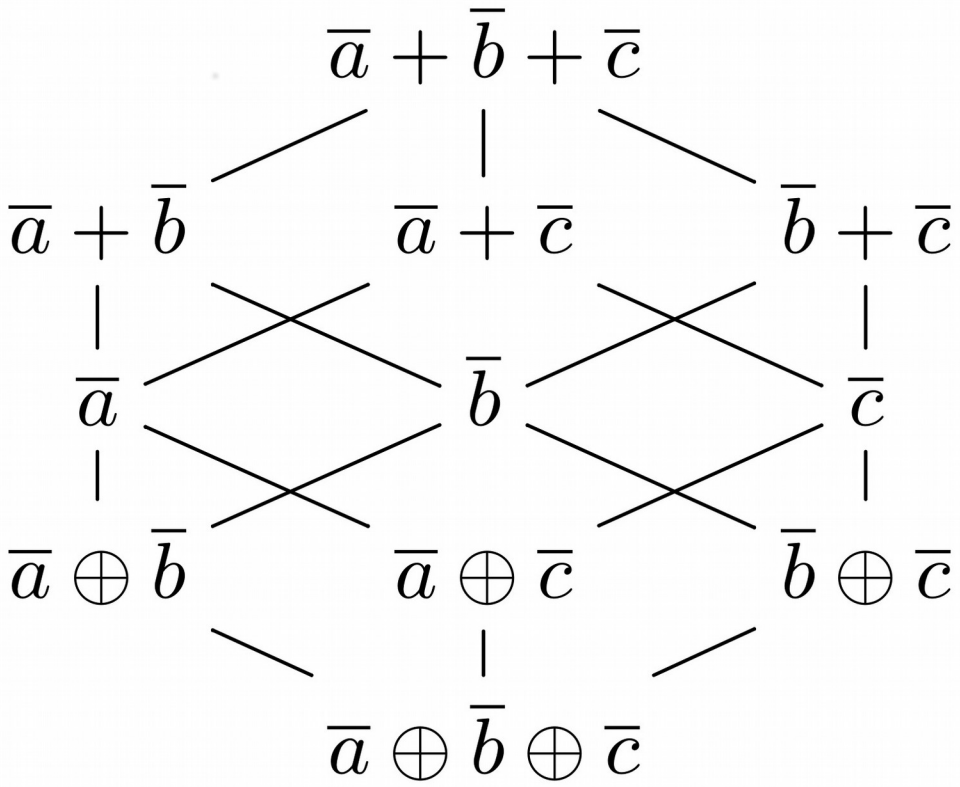
Compliance: complexity

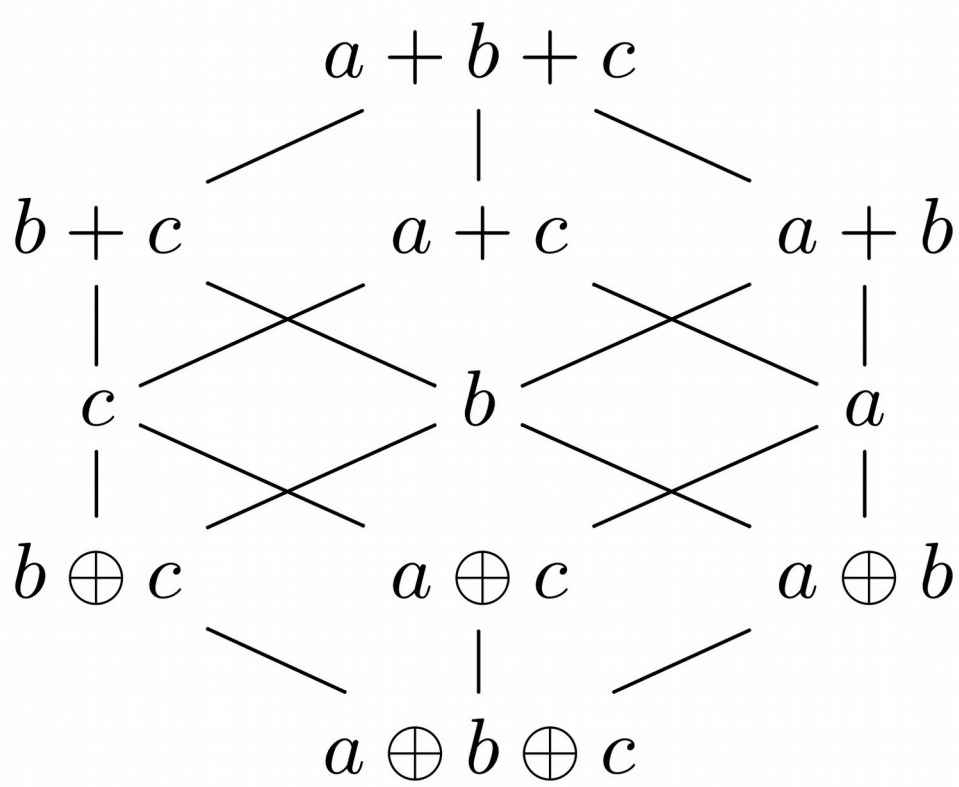
- One can define a recursive proof-search algorithm reading bottom-up the rules
- The complexity is exponential
- Better solution: extend the algorithm for subtyping of recursive arrow and product types from Pierce
  - not a trivial extension
  - keep trace not only of past successes, but also of past failures
- The complexity is $O(n^5)$
  - Pierce's algorithm has complexity $O(n^2)$

Subcontract relation

- Subcontract relation for servers:
  $\rho \preccurlyeq_s \rho'$ iff for each client σ. $\sigma \dashv\Vdash \rho$ implies $\sigma \dashv\Vdash \rho'$
  - ρ' has more clients than ρ
- Subcontract relation for clients is dual:
  $\sigma \preccurlyeq_c \sigma'$ iff for each server ρ. $\sigma \dashv\Vdash \rho$ implies $\sigma' \dashv\Vdash \rho$
- The two subcontract relations are partial orders

Subcontract relation: example





Duality

- We define the dual $\overline{\sigma}$ of a client contract σ as the minimum server compliant with σ
- Duality enjoys the classic simple syntactic definition
  - Swap inputs with outputs ($a \leftrightarrow \overline{a}$) and internal choice with external choice ($\oplus \leftrightarrow \Sigma$)

Subcontract relation: results

- Subcontract relation for servers and for clients are related:
  $\rho \preccurlyeq_s \rho'$ iff $\overline{\rho'} \preccurlyeq_c \overline{\rho}$
- Subcontract relation and compliance are related:
  $\rho \preccurlyeq_s \rho'$ iff $\overline{\rho} \dashv\!\!\| \rho'$
- Also the subcontract relation can be decided in $O(n^5)$

Conservative extension

- Retractable/speculative contracts are conservative extensions of classic contracts
  - Syntactically
  - Semantically
  - From the point of view of compliance, subcontract relation and duality

Map of the talk

- What retractable/speculative contracts are?
- Motivating example
- Results
- Conclusion

Summary

- We presented a model of contracts with retractable/speculative choice
- Using retractable/speculative choice instead of normal choice ensures compliance with a larger set of partners
- Retractable/speculative contracts are a conservative extension of classic contracts, yet they preserve most of the good properties of contracts:
  - decidability of compliance and subcontract relation
  - efficient decidability algorithm
  - easy syntactic characterization of duality

Future work

- Explore the notion of retractable/speculative contracts in multiparty sessions
- How can we extract a contract from a reversible/speculative application?
  - See ICE paper on retraction in session types: Session types for orchestrated interactions, by Barbanera and de'Liguoro
- Which is the relation between retractable contracts and process calculi for reversible computation?
  - Preliminary result: retractable contracts can be seen as a controlled form of reversibility on classic 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	COORDINATION 2017
free rollback	vs	rollback only when stuck
explicit checkpoint	vs	implicit checkpoint
one checkpoint	vs	stack of checkpoints
compliance harder	vs	compliance easier