

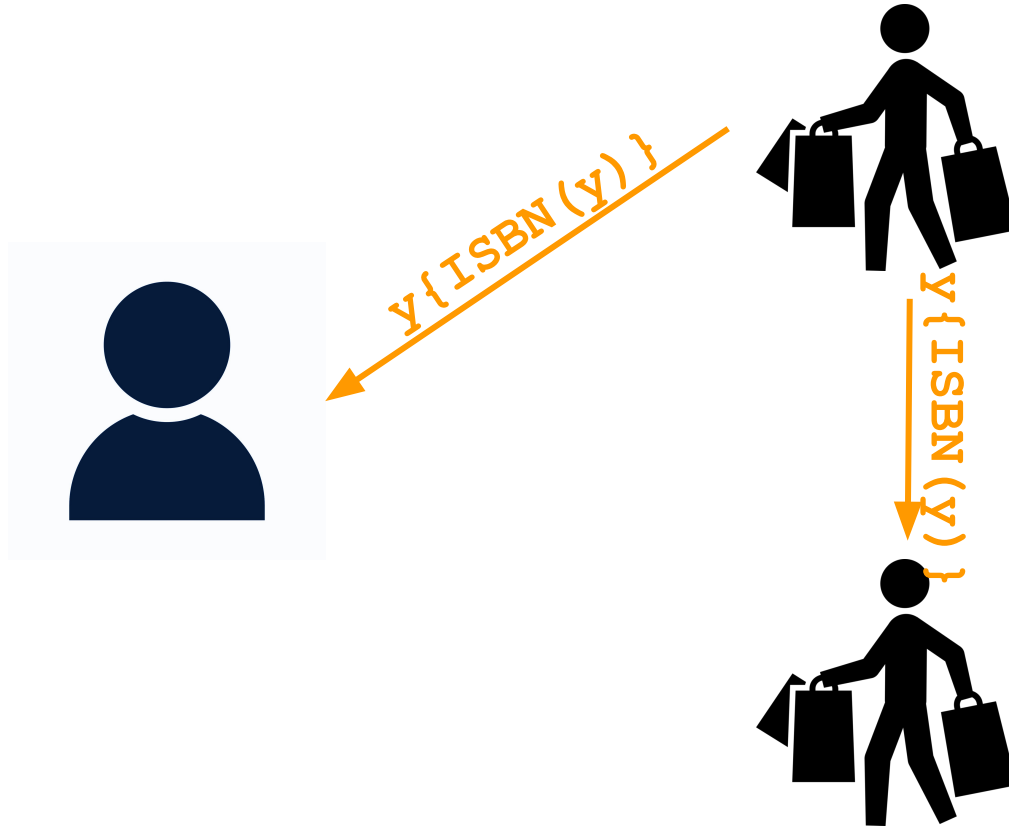
Certified Implementability of Global Multiparty Protocols

Elaine Li

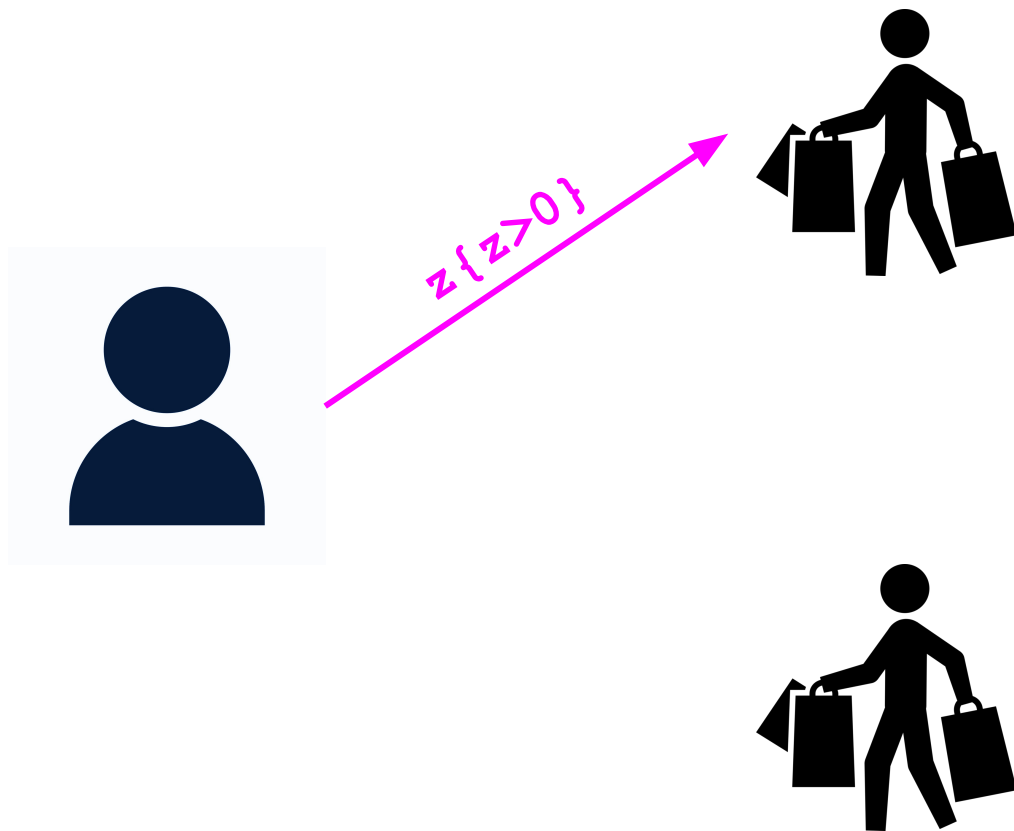
Thomas Wies



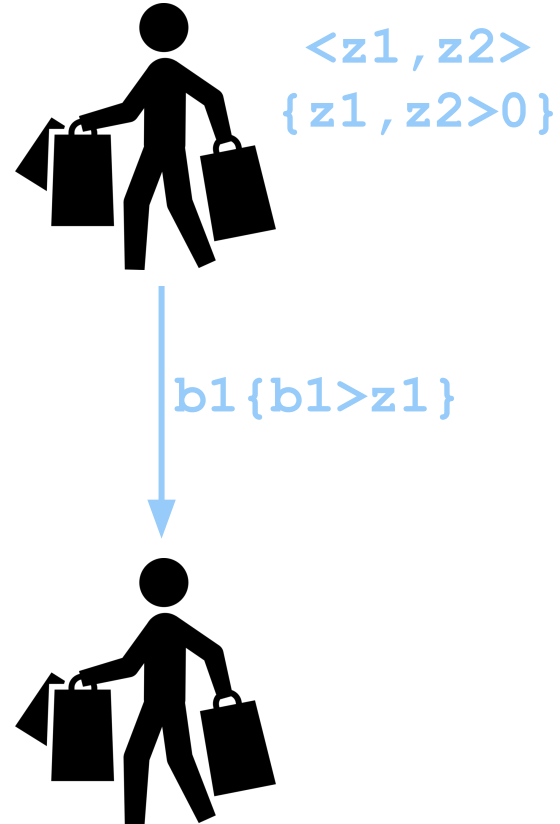
Global protocols: two-bidder protocol



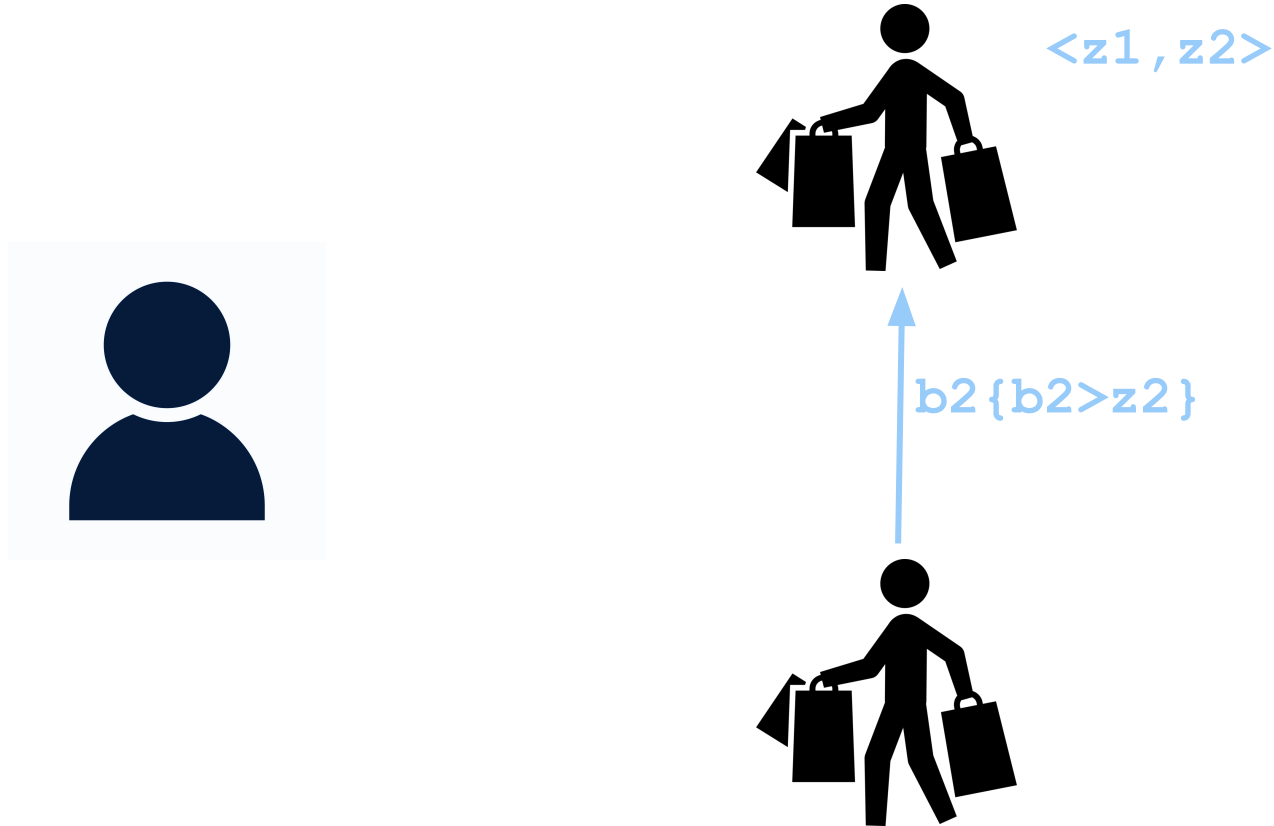
Global protocols: two-bidder protocol



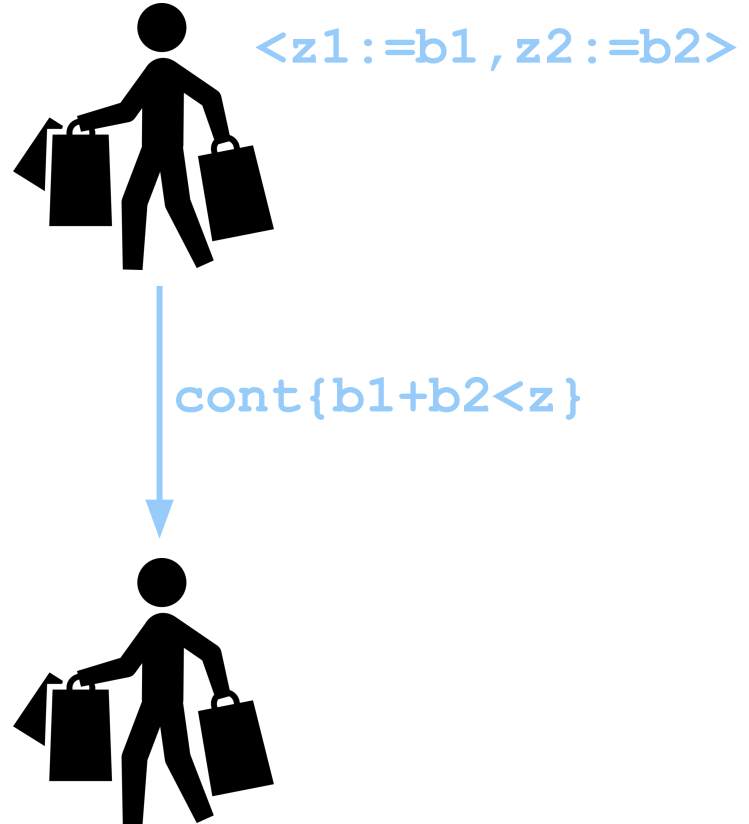
Global protocols: two-bidder protocol



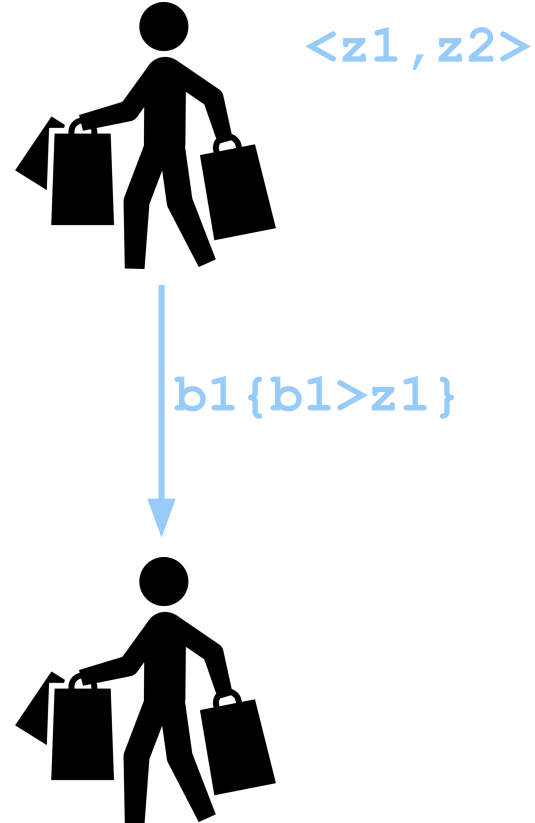
Global protocols: two-bidder protocol



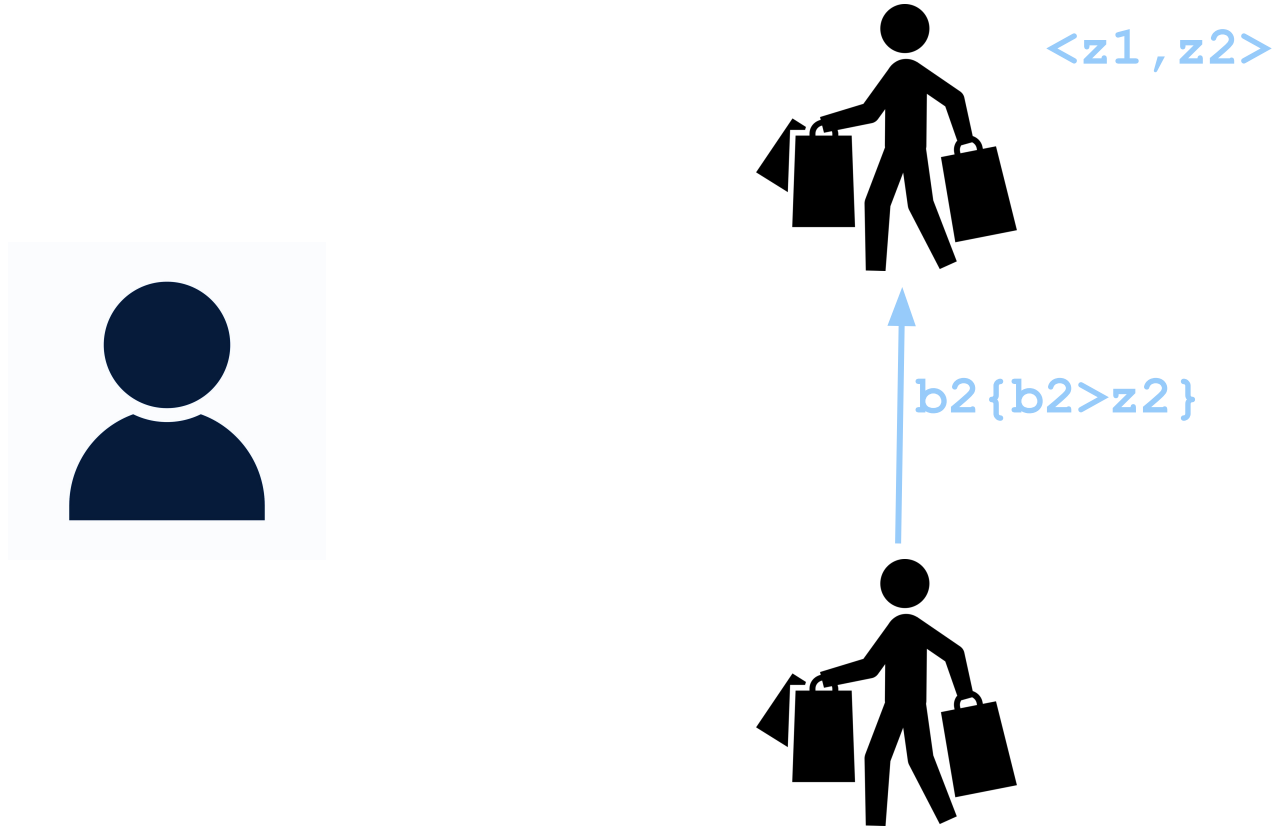
Global protocols: two-bidder protocol



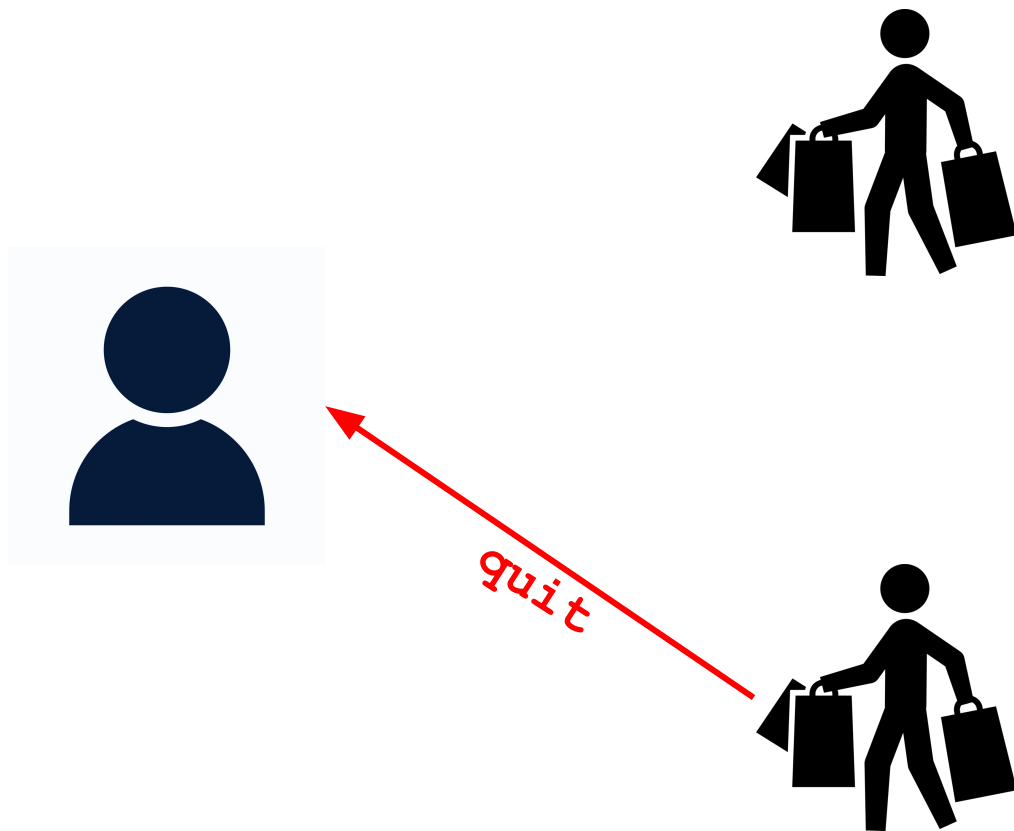
Global protocols: two-bidder protocol



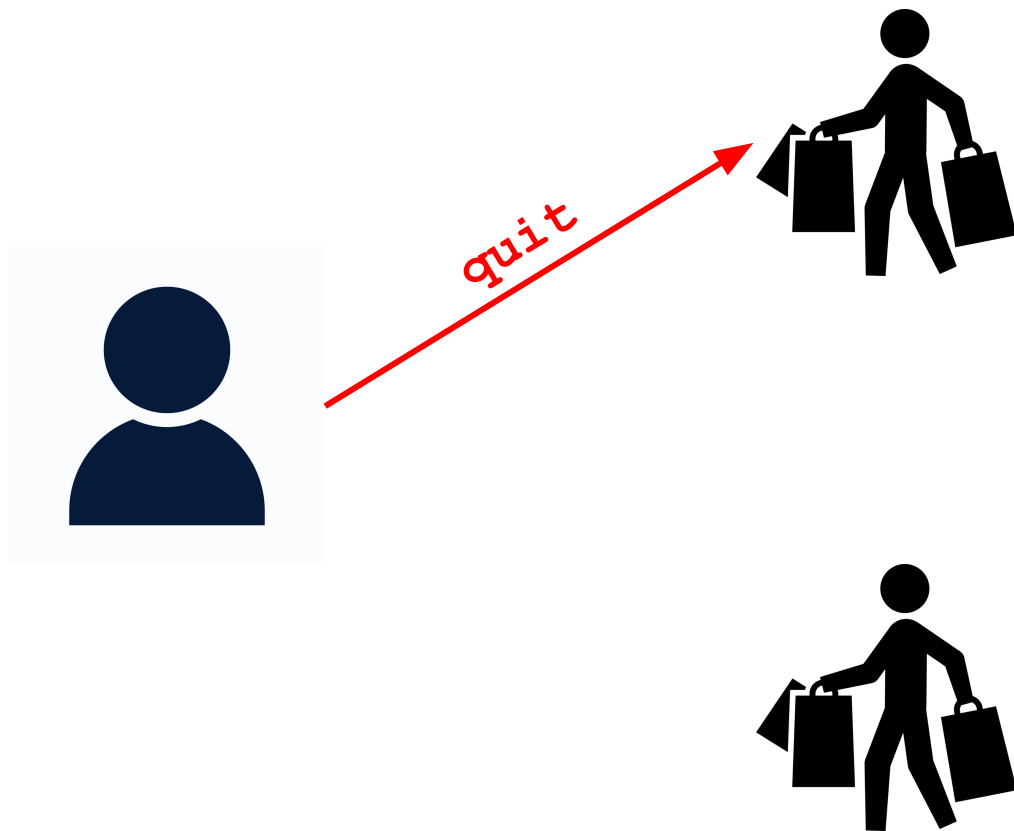
Global protocols: two-bidder protocol



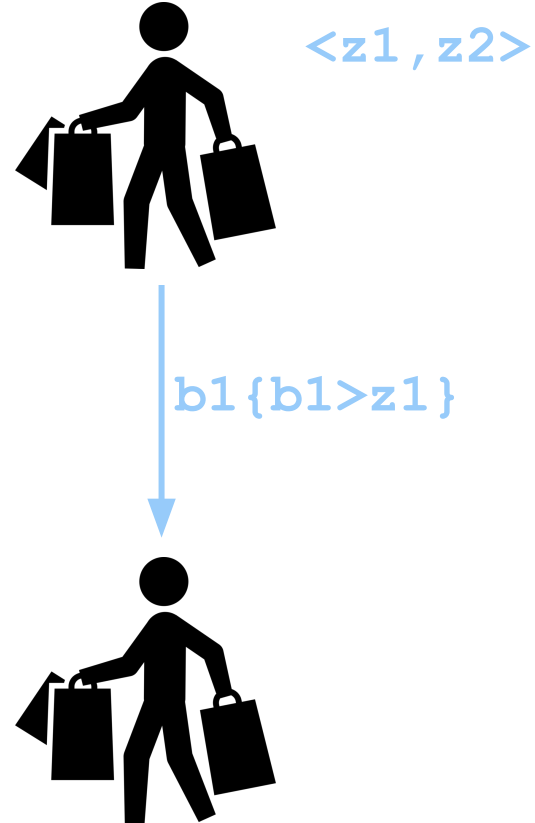
Global protocols: two-bidder protocol



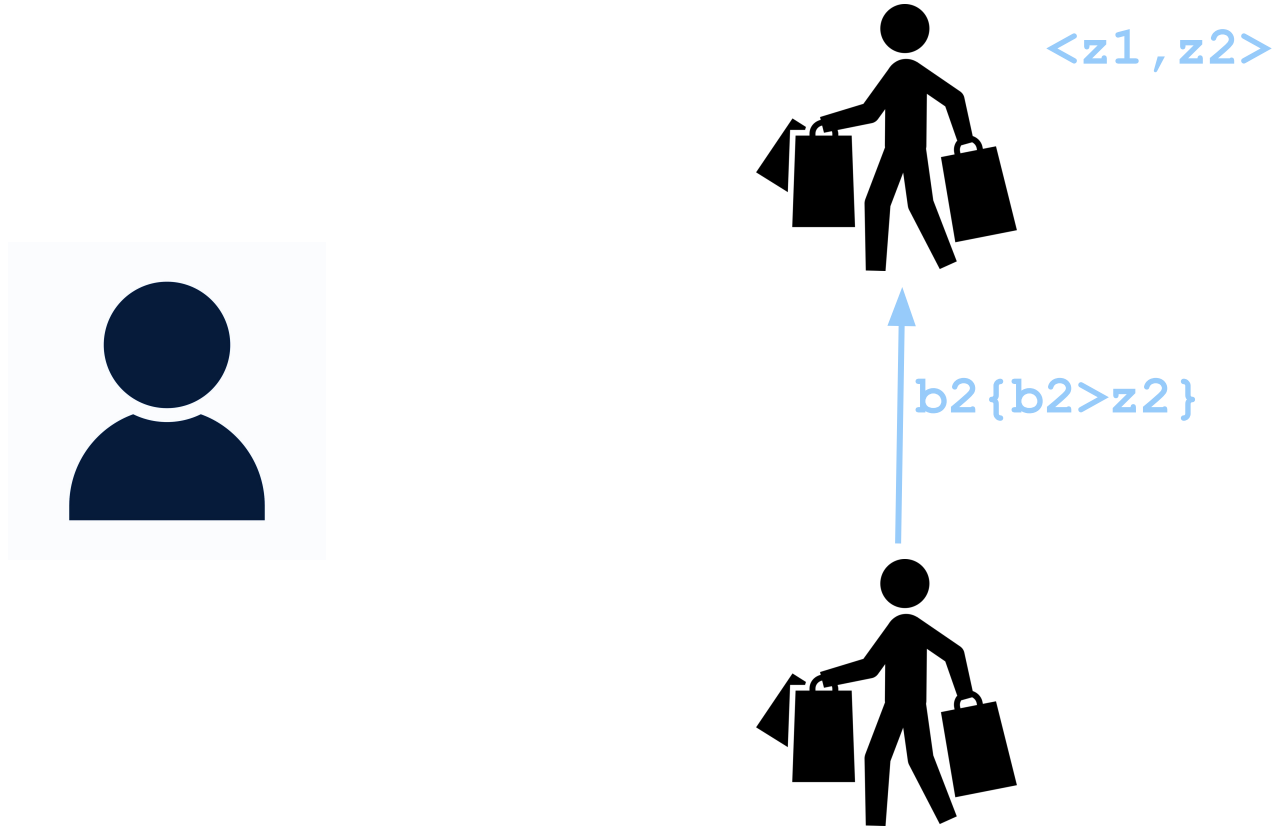
Global protocols: two-bidder protocol



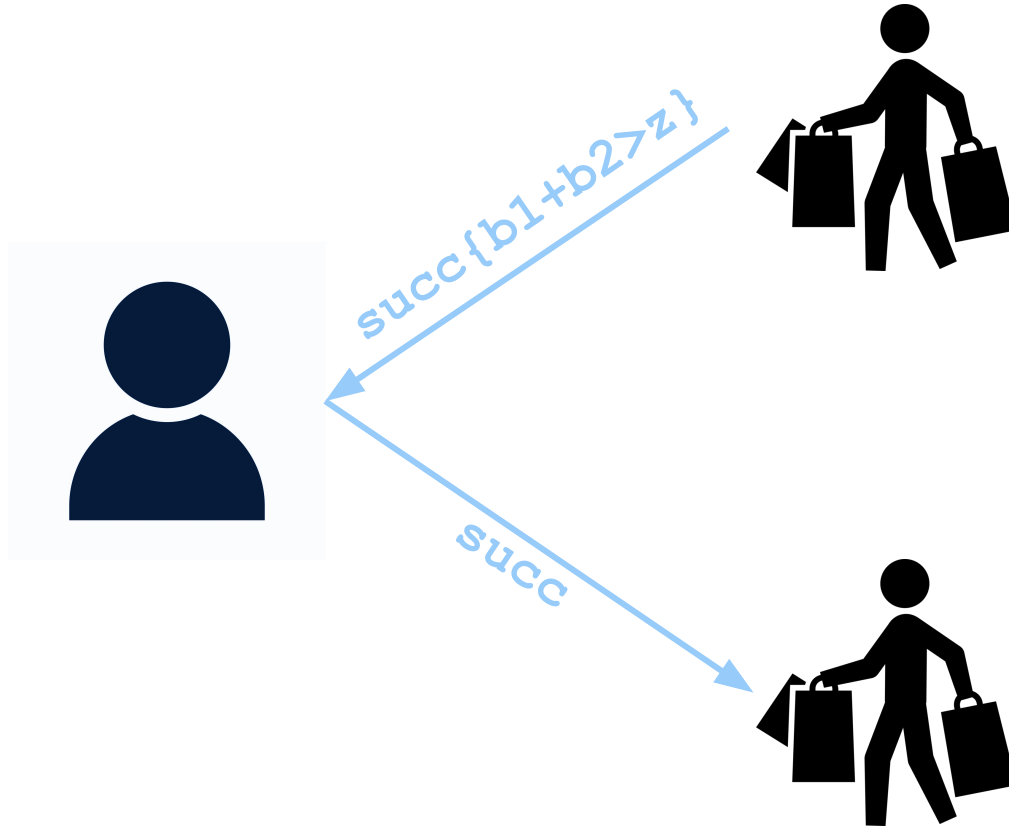
Global protocols: two-bidder protocol



Global protocols: two-bidder protocol



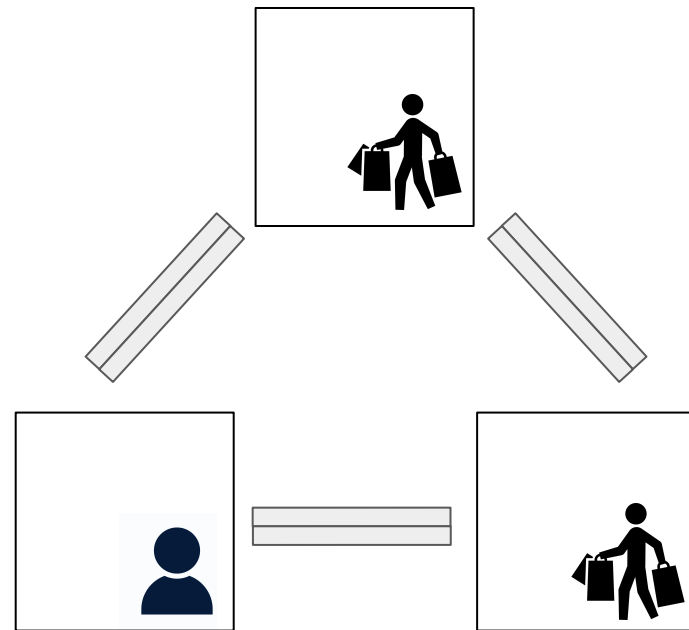
Global protocols: two-bidder protocol



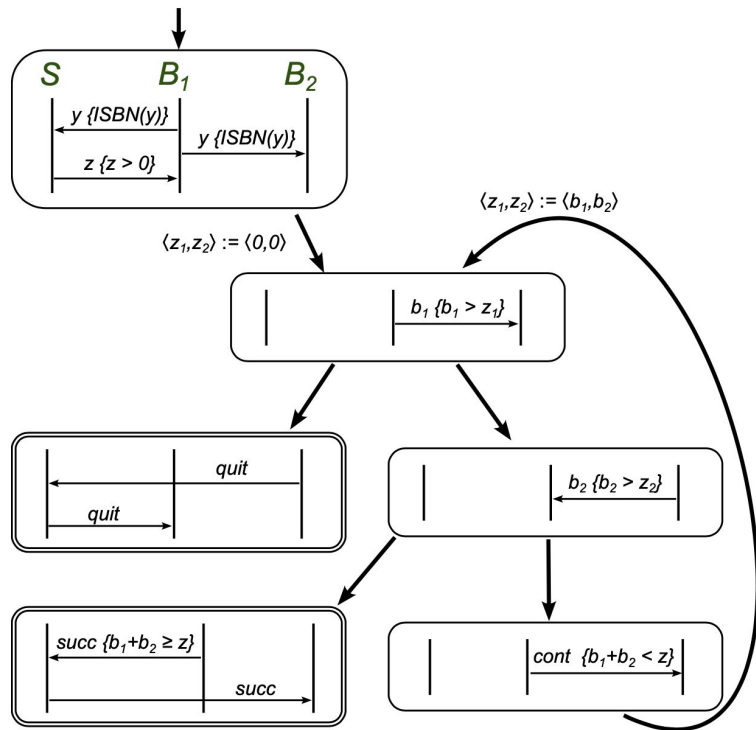
Overview

Asynchronous, message-passing programs are challenging to implement individually

- Communication errors e.g. orphan messages, unspecified receptions
- Deadlocks



Overview

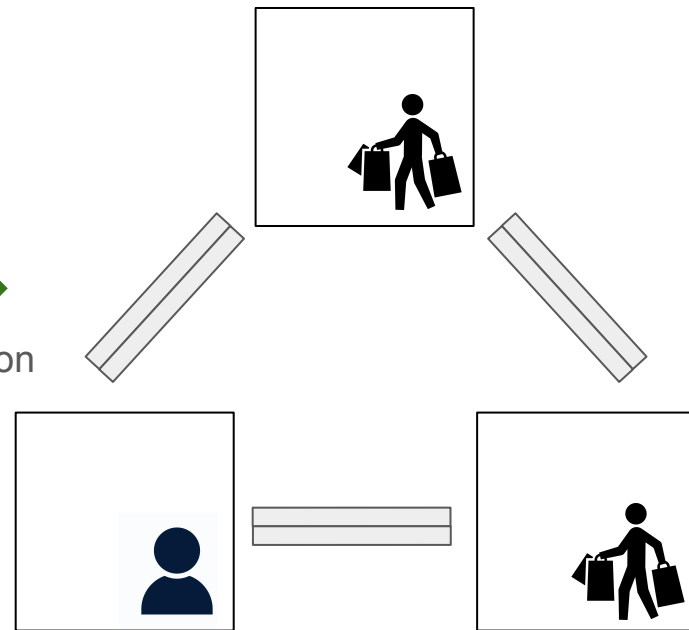
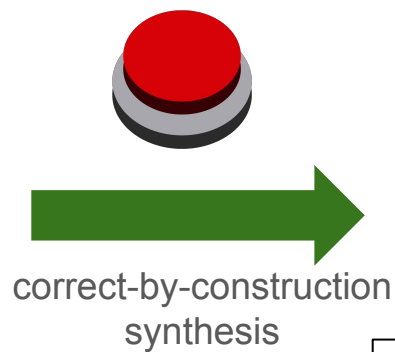
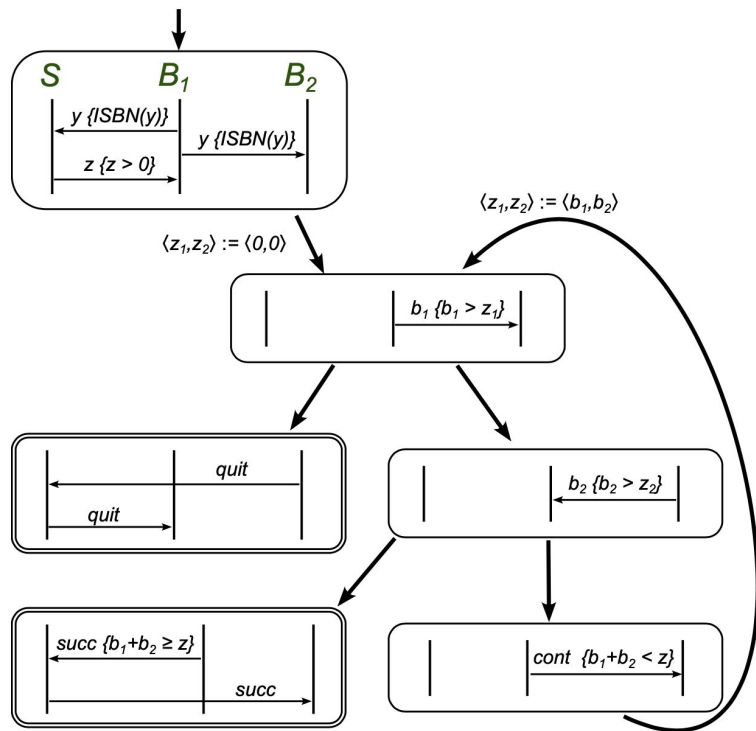


Global protocols are **synchronous** specifications of **all** participants' behaviors

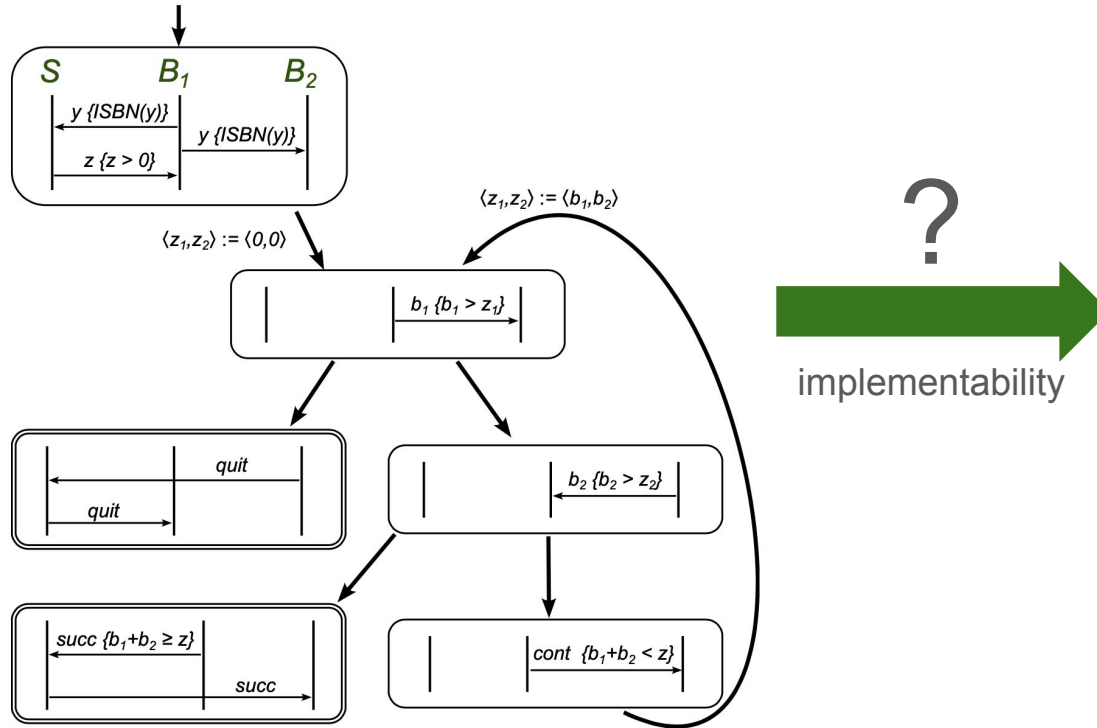
- High-level message sequence charts [Mauw and Reniers 97]
- Session types [Honda et al. 08]
- Choreographic programming [Carbone and Montesi 13]

Applications: ITU standard, UML, Web Service Choreography Description Language, cyber-physical systems etc.

Overview



Overview

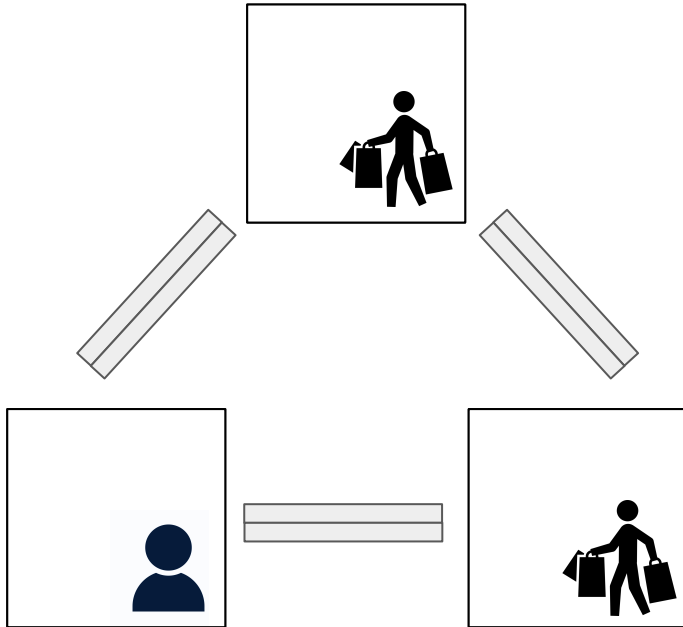


Implementation model

(controllable)

(non-controllable)

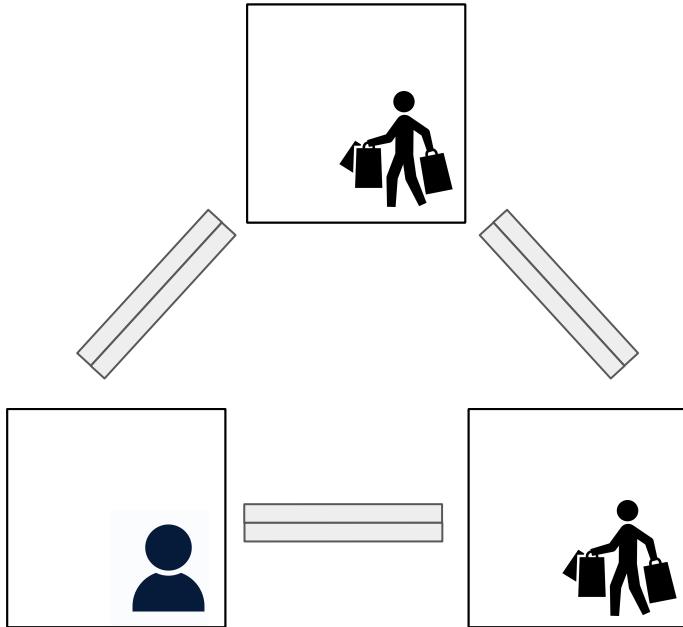
Communicating Labeled Transition System (CLTS) = per participant LTS + peer-to-peer FIFO channels



- Communicating state machines [Brand and Zafiropulo 83]
- Ubiquitous in theory and verification

Implementation model

Communicating Labeled Transition System (CLTS) = **per participant LTS** + peer-to-peer FIFO channels

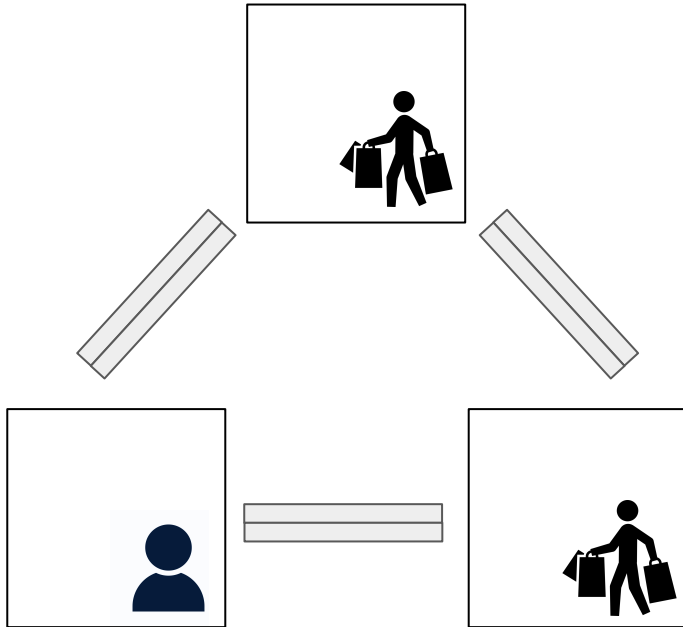


$b_1 ! s : y \cdot b_1 ! b_2 : y \cdot s ? b_1 : y \cdot s ! b_1 : z \cdot b_2 ? b_1 : y$ ✓

$b_1 ! s : y \cdot b_1 ! b_2 : y \cdot s ! b_1 : z \cdot s ? b_1 : y \cdot b_2 ? b_1 : y$ ✗

Implementation model

Communicating Labeled Transition System (CLTS) = per participant LTS + **peer-to-peer FIFO channels**



$b_1 ! s : y \cdot b_1 ! b_2 : y \cdot s ? b_1 : y \cdot b_2 ? b_1 : y \cdot s ! b_1 : z$ ✓

$b_1 ! s : y \cdot b_1 ! b_2 : y \cdot b_2 ? b_1 : y \cdot s ? b_1 : y \cdot s ! b_1 : z$ ✓

$b_1 ! s : y \cdot b_1 ! b_2 : y \cdot s ? b_1 : y \cdot s ! b_1 : z \cdot b_2 ? b_1 : y$ ✓

CLTS indistinguishability

A word is **executable** if it is a trace of some* CLTS

Two words are **indistinguishable** if any CLTS that executes one must execute the other

$b_1 ! s : y \cdot b_1 ! b_2 : y \cdot s ? b_1 : y \cdot b_2 ? b_1 : y \cdot s ! b_1 : z$ ✓

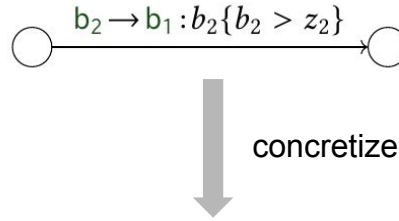
$b_1 ! s : y \cdot b_1 ! b_2 : y \cdot b_2 ? b_1 : y \cdot s ? b_1 : y \cdot s ! b_1 : z$ ✓

Global protocol semantics must be indistinguishability-closed w.r.t. its target implementation model

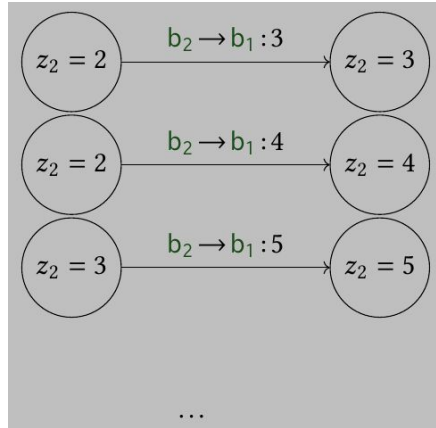
*non-deadlocking

Global protocol semantics

Symbolic transition

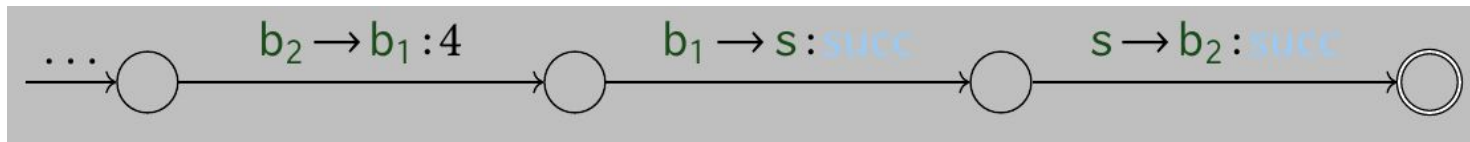


Concrete transition



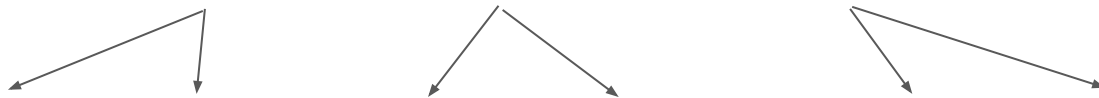
Global protocol semantics

Concrete run



Synchronous

$\dots b_2 \rightarrow b_1 : 4 \cdot b_1 \rightarrow s : \text{succ} \cdot s \rightarrow b_2 : \text{succ}$



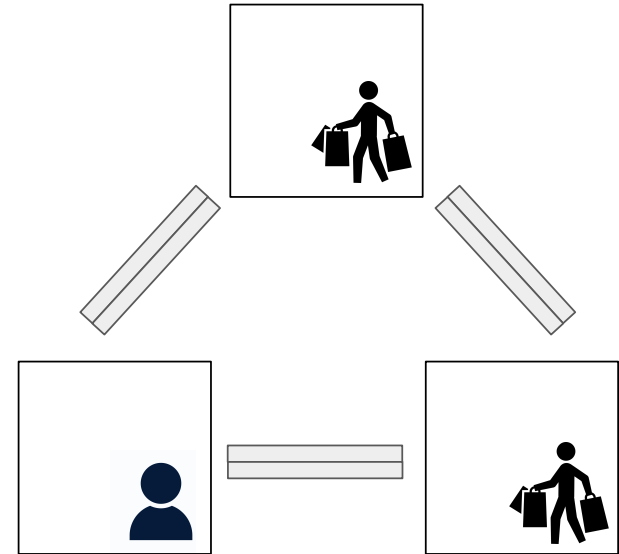
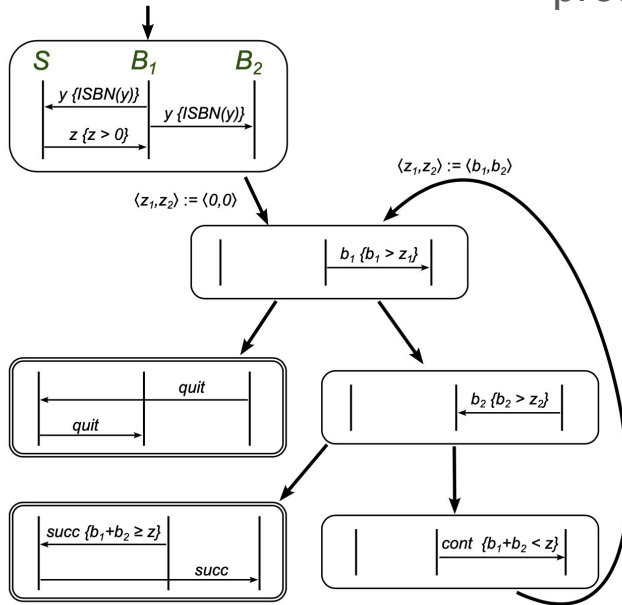
Asynchronous

$\dots b_2 ! b_1 : 4 \cdot b_1 ? b_2 : 4 \cdot b_2 ! s : \text{succ} \cdot s ? b_2 : \text{succ} \cdot s ! b_2 : \text{succ} \cdot b_2 ? s : \text{succ}$

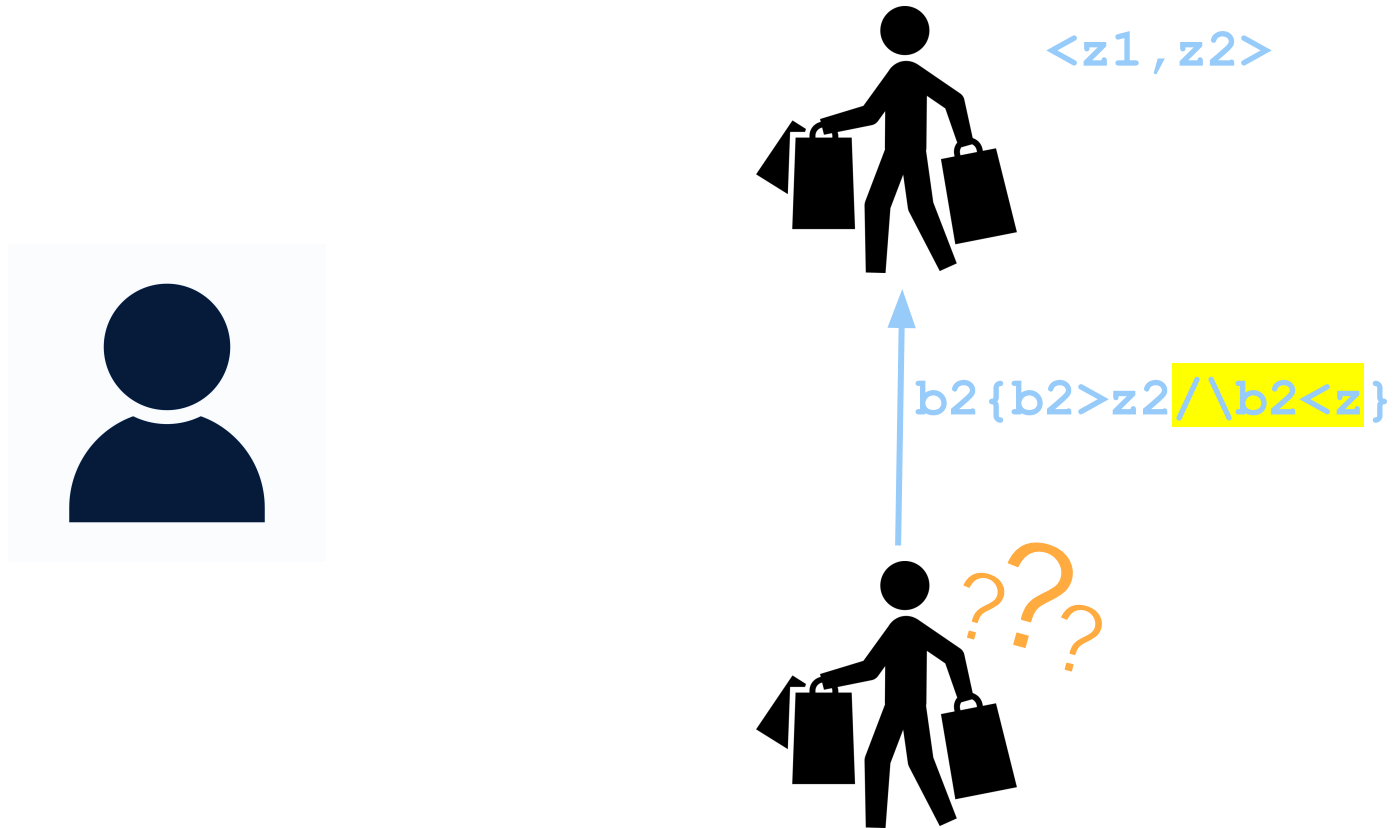
...and all words indistinguishable

Implementability

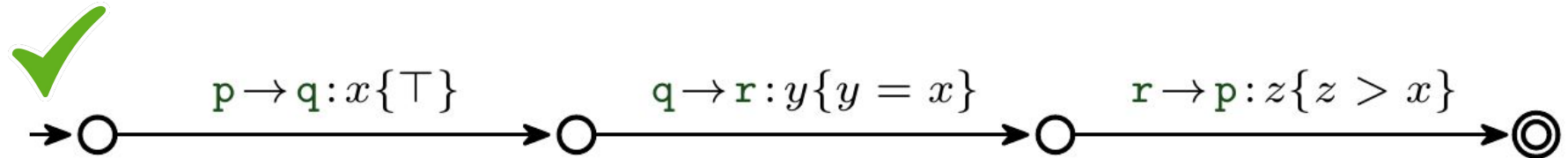
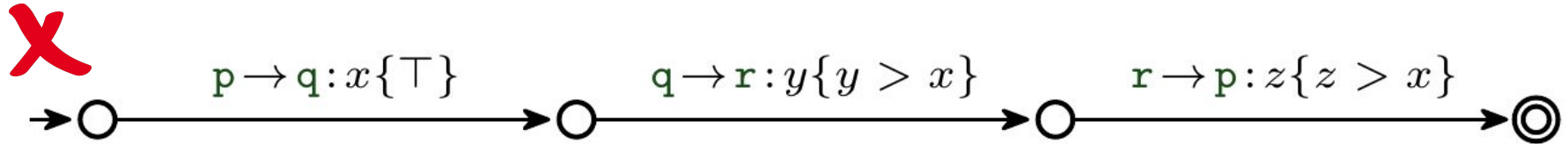
Implementability = exists a CLTS satisfying
protocol fidelity + deadlock freedom?



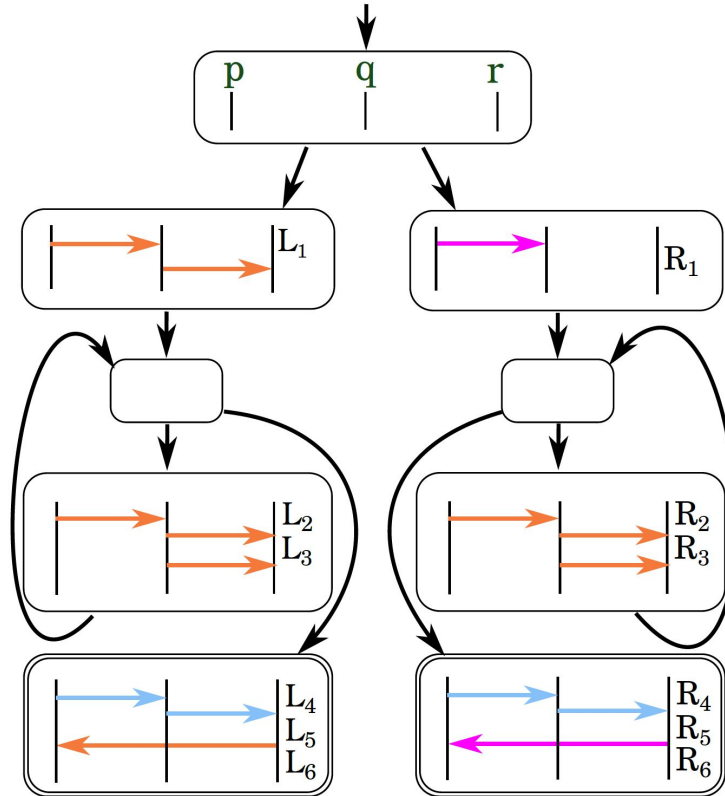
Non-implementability: two-bidder protocol



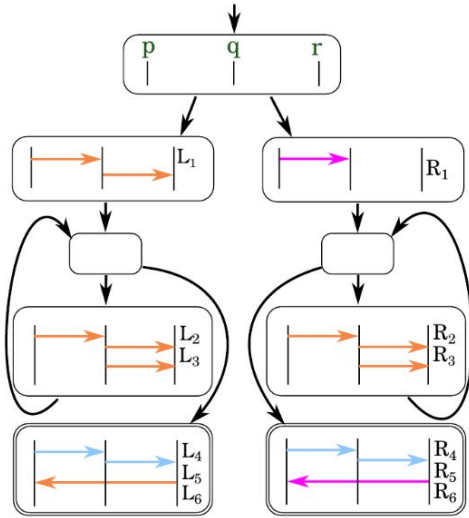
Non-implementability: protocol variables



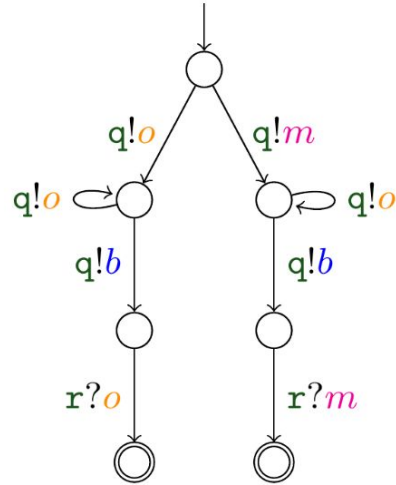
Non-implementability?



Odd-even example

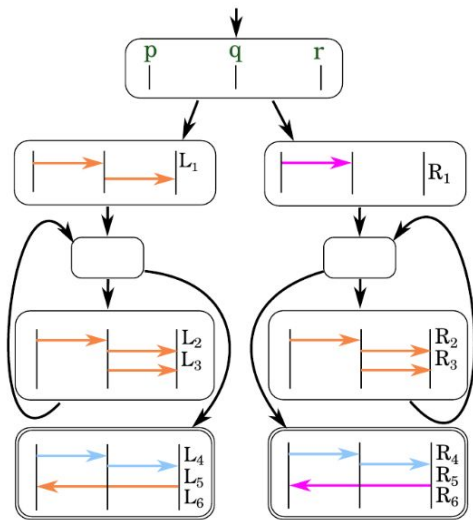


(a) Odd-even protocol

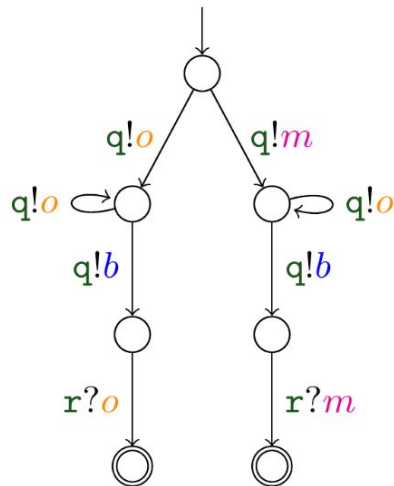


(b) Local impl.
for role p

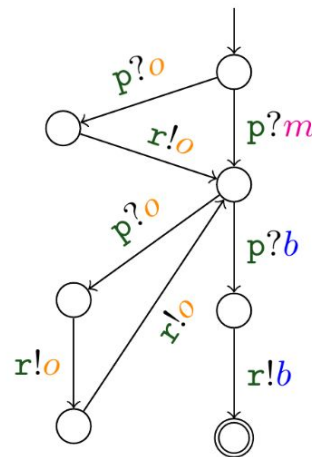
Odd-even example



(a) Odd-even protocol

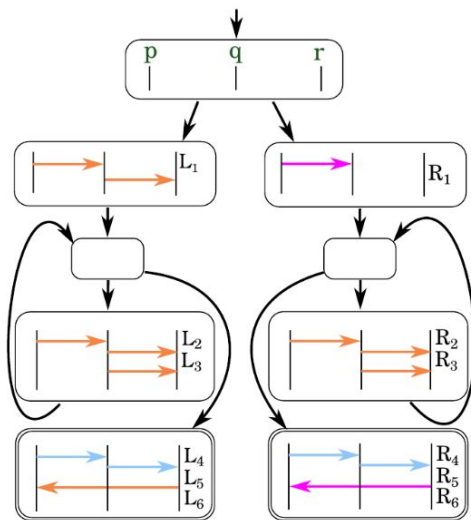


(b) Local impl.
for role p

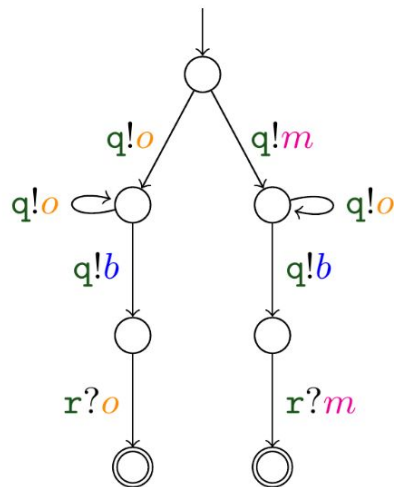


(c) Local impl.
for role q

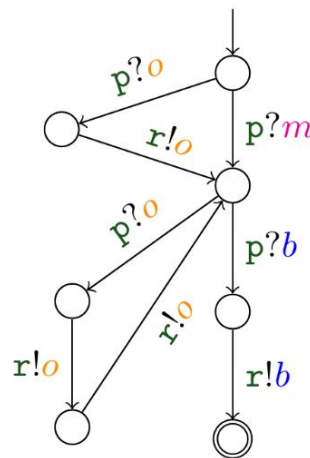
Odd-even example



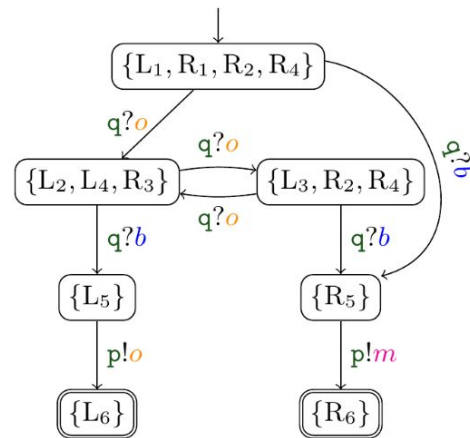
(a) Odd-even protocol



(b) Local impl.
for role **p**

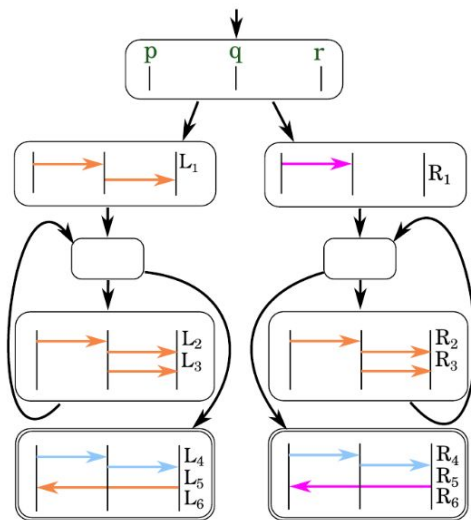


(c) Local impl.
for role **q**

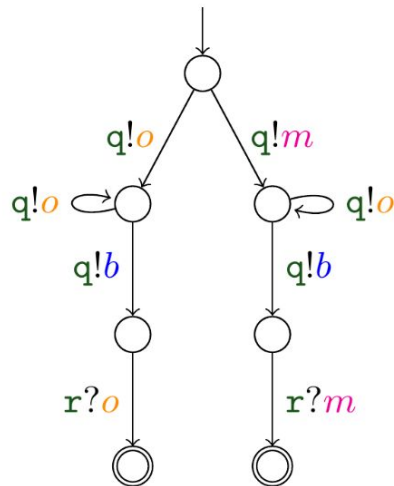


(d) Local impl.
for role **r**

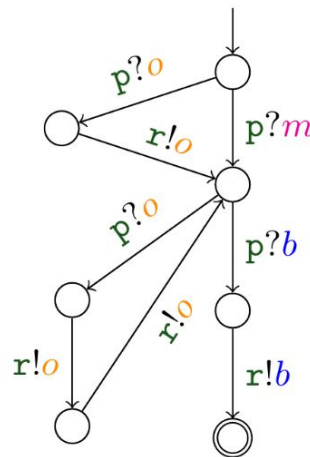
Odd-even example



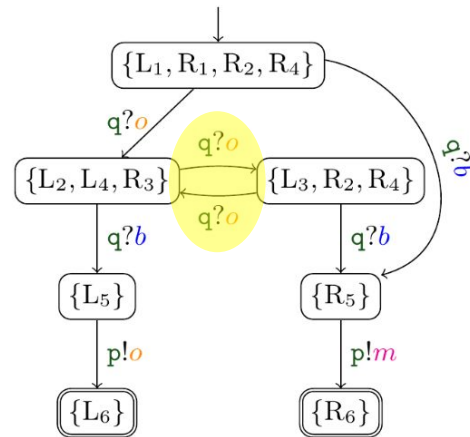
(a) Odd-even protocol



(b) Local impl.
for role **p**



(c) Local impl.
for role **q**



(d) Local impl.
for role **r**

Flawed results in the literature

FIFO asynchronous and when they are just rendez-vous synchronizations. This property was claimed to be decidable in several conference and journal papers [1, 4, 3, 2] for either mailboxes (*-1) or peer-to-peer (1-1) communications, thanks to a form of small model property. In this paper, we show that this small model property **does not hold neither for mailbox communications, nor for peer-to-peer communications**, therefore the decidability of synchronizability becomes an open question. We close this question for peer-to-peer communications, and we show that synchronizability is **actually undecidable**. We show that synchronizability is decidable if the topology

[Finkel and Lozes 17]

(b) Caires and Pérez [2016]; Chen [2015]; Deniélou et al. [2012]; Deniélou and Yoshida [2012]; Toninho and Yoshida [2016]	$\geq$ full	no	flawed	(C2)
---	-------------	----	--------	------

[Scalas and Yoshida 19]

Our reduction shows that deciding the $\text{avail}_{p,q,\{q\}}(m,s)$ predicate for global types is in co-NP, **which refutes the polynomial time upper bound claimed in [55]**. The proof of Lemma 5.9 can be

[Li et al. 25]



Rocq mechanization

```
Theorem preciseness :  
  ∀ {State : Type}  
    (S : @LTS SyncAlphabet State),  
    GCLTS S →  
    (∃ {LocalState : Type},  
      @implementable State LocalState S) ↔  
      @NMC State S ∧ @RCC State S ∧ @SCC State S.
```

Theorem [Li et al. 25]. A protocol is implementable if and only if it satisfies the Coherence Conditions.

In a nutshell, from two simultaneously reachable global states, a participant can:

- Send a message permissible from both states (SCC)
- Receive a message distinguishing the two states (RCC)

but cannot choose between sending or receiving a message (NMC).



Rocq mechanization

```
Theorem preciseness :  
  ∀ {State : Type}  
    (S : @LTS SyncAlphabet State),  
    GCLTS S →  
    (∃ {LocalState : Type},  
      @implementable State LocalState S) ↔  
      @NMC State S ∧ @RCC State S ∧ @SCC State S.
```

Design choices:

- Unified protocol representation
- Formal language-theoretic treatment

```
Record LTS {A : Type} {State : Type} :=  
  mkLTS {  
    transition : State → A → State → P;  
    s0 : State;  
    final : State → P;  
  }.
```



Rocq mechanization

```
Theorem preciseness :  
  ∀ {State : Type}  
    (S : @LTS SyncAlphabet State),  
    GCLTS S →  
    (∃ {LocalState : Type},  
      @implementable State LocalState S) ↔  
      @NMC State S ∧ @RCC State S ∧ @SCC State S.
```

Takeaways:

- Generalization to infinite participants
- Sink-finality GCLTS assumption can be conditionally removed
- Infinite word semantics bug



Rocq mechanization

```
Theorem preciseness :  
  ∀ {State : Type}  
    (S : @LTS SyncAlphabet State),  
    GCLTS S →  
    (∃ {LocalState : Type},  
      @implementable State LocalState S) ↔  
      @NMC State S ∧ @RCC State S ∧ @SCC State S.
```

Takeaways:

- Generalization to infinite participants
- Sink-finality GCLTS assumption can be conditionally removed
- **Infinite word semantics bug**

Global semantics must be indistinguishability-closed

Finite words:

$b_1 ! s : y \cdot b_1 ! b_2 : y \cdot s ? b_1 : y \cdot b_2 ? b_1 : y \cdot s ! b_1 : z$ ✓

$b_1 ! s : y \cdot b_1 ! b_2 : y \cdot b_2 ? b_1 : y \cdot s ? b_1 : y \cdot s ! b_1 : z$ ✓

Global semantics must be indistinguishability-closed

Infinite words:

$$\begin{array}{l} p!q:m^n \cdot p!q:m' \cdot r!s:m' \\ r!s:m' \cdot p!q:m^\omega \end{array} \quad \text{for all } n \in \mathbb{N}$$


Global semantics must be indistinguishability-closed

Infinite words:

$$p!q:m^n \cdot p!q:m' \cdot r!s:m' \quad \text{for all } n \in \mathbb{N} \quad \checkmark$$

$$r!s:m' \cdot p!q:m^\omega \quad \checkmark$$

"there exists a global run for every prefix of an infinite word" [Majumdar et al. 22]

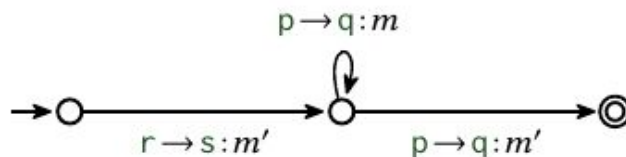
Global semantics must be indistinguishability-closed

Infinite words:

$p!q:m^n \cdot p!q:m' \cdot r!s:m'$ for all $n \in \mathbb{N}$ ✓

$r!s:m' \cdot p!q:m^\omega$ ✓

"there exists a global run for every prefix of an infinite word" [Majumdar et al. 22]



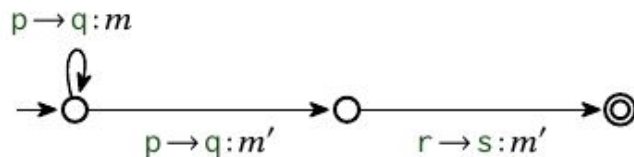
Global semantics must be indistinguishability-closed

Infinite words:

$p!q:m^n \cdot p!q:m' \cdot r!s:m'$ for all $n \in \mathbb{N}$ ✓

$r!s:m' \cdot p!q:m^\omega$ ✗

"there exists a global run for every prefix of an infinite word" [Majumdar et al. 22]



Global semantics must be indistinguishability-closed

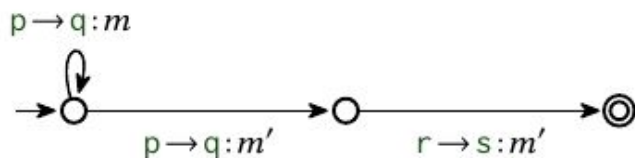
Infinite words:

$$p!q:m^n \cdot p!q:m' \cdot r!s:m' \quad \text{for all } n \in \mathbb{N} \quad \checkmark$$

$$r!s:m' \cdot p!q:m^\omega \quad \checkmark$$

"there exists a global run for every prefix of an infinite word" [Majumdar et al. 22]

"for every prefix of an infinite word there exists a global run"



Future work and extensions

- Modeling and proving implementability of existing protocol formalisms [Castro-Perez et al. 21, Hirsch and Garg 22]
- Synthesis of certified implementations
- Investigating different network models

