Formal Methods for Communication Protocols

Harnessing distributed software design with behavioural contracts

Dragiša Žunić @ GSSI & Emilio Tuosto @ GSSI

– Lecture 5 –

3 - 12 March, 2026 - Novi Sad

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
DESK Developing Shared Knowledge

– Formalising market models –

Today's menu

- Market model design is tricky
  - CLOB issues
  - FBA solution
  - FBA issues

Today's menu

- Market model design is tricky
  - CLOB issues
  - FBA solution
  - FBA issues
- **C**oncurrent e**X**change (**CX**)
  - A new concurrent model
    - concurrency: what can be executed independently?

Today's menu

- Market model design is tricky
  - CLOB issues
  - FBA solution
  - FBA issues
- **C**oncurrent e**X**change (**CX**)
  - A new concurrent model
    - concurrency: what can be executed independently?
    - parallelism: what can be executed at the same time?

Today's menu

- Market model design is tricky
  - CLOB issues
  - FBA solution
  - FBA issues
- **C**oncurrent e**X**change (**CX**)
  - A new concurrent model
    - concurrency: what can be executed independently?
    - parallelism: what can be executed at the same time?

    The latter requires the former, not viceversa!
  - A formal definition of **CX** ... with a lot of hand-weaving ☺
  - Supported by a prototype tool for simulations

Take-away message

CX: defined in a model of concurrency

Take-away message

CX: defined in a model of concurrency

- is more "natural" than CLOB (& FBA)

Take-away message

CX: defined in a model of concurrency

- is more "natural" than CLOB (& FBA)
- is white-box: rules precisely defined and applied

Take-away message

CX: defined in a model of concurrency

- is more "natural" than CLOB (& FBA)
- is white-box: rules precisely defined and applied
- can certify desirable properties (or show absence of undesirable ones)

Take-away message

CX: defined in a model of concurrency

- is more "natural" than CLOB (& FBA)
- is white-box: rules precisely defined and applied
- can certify desirable properties (or show absence of undesirable ones)

There is a prototype ... which can also be shown to be correct

Is our approach worthwhile?

Take-away question

Is our approach worthwhile?

JEL Codes: D47, G10, G12, G14

"The market is rigged." —Michael Lewis, *Flash Boys* (Lewis 2014)

"Widespread latency arbitrage is a myth." —Bill Harts, CEO of the Modern Markets Initiative, a high-frequency trading (HFT) lobbyist (Michaels 2016)

Trading floor and open outcry

CBOT "The Pit" in 1908

The New York stock exchange trading floor in September 1963, before the introduction of electronic readouts and computer screens

Open outcry "pit" at the Chicago Board of Trade (CBOT) in 1993

- Human-mediated market interaction (open outcry)
- . . . humans performed order-matching and agreed on trades

Towards electronic systems

- 1971: NASDAQ introduced elements of electronic infrastructure
- 1977: Toronto Stock Exchange
  - First operational electronic limit order book system in the world
  - Initially used only for less liquid stocks, not the entire main market
- 1986: Paris Bourse: starts using CLOB as primary mechanism
- 1990s–2000s: NYSE, NASDAQ, and CME transitioned to electronic matching
- Today: market matching is automated, but fundamentally sequential (CLOB)

1992–1994: Borsa Italiana completed the transition from open outcry to electronic trading (`https://www.borsaitaliana.it/borsaitaliana/storia/storia/telematizzazione-scambi.en.htm`)

The big question

CME trading pits, Chicago

(2025)

Open outcry, Japan (circa 1960)

- *What is the real nature of the "computation" taking place on a trading venue?*

The big question

CME trading pits, Chicago

(2025)

Open outcry, Japan (circa 1960)

- *What is the real nature of the "computation" taking place on a trading venue?*
- Concurrency / Parallelism?
- What is the "right" model?

The big question

CME trading pits, Chicago

(2025)

Open outcry, Japan (circa 1960)

- *What is the real nature of the "computation" taking place on a trading venue?*
- Concurrency / Parallelism?
- What is the "right" model?
- We must start from a blank paper

Overview: where are we?

- **Fundamental** market design

- When humans were removed, markets converged to the CLOB model
- Is that solution optimal?
- *Is it optimally designed as a computational model at the fundamental level?*
- Not quite ...

Overview: where are we?

- **Fundamental** market design

- When humans were removed, markets converged to the CLOB model
- Is that solution optimal?
- *Is it optimally designed as a computational model at the fundamental level?*
- Not quite ...

- Motivation for Frequent Batch Auctions (Budish et al, 2015.)

Volume 130, Issue 4
November 2015

JOURNAL ARTICLE EDITOR'S CHOICE

The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response *

Eric Budish, Peter Cramton, John Shim

The Quarterly Journal of Economics, Volume 130, Issue 4, November 2015, Pages 1547–1621, https://doi.org/10.1093/qje/qjv027

Published: 23 July 2015

The issue with CLOB

The problem arises from the **interaction** of two **fundamental** design choices:

1. Treating time as continuous
2. Sequential processing of orders

The issue with CLOB

The problem arises from the **interaction** of two **fundamental** design choices:

1. Treating time as continuous
2. Sequential processing of orders

Both at the level of fundamental design

Manifest later at different levels

FBA's solution

Key idea: address both in the following way:

1. "Put time into discrete units"
   (discretize time into small uniform intervals)
2. "Process incoming orders in *batches* using auctions"
   (match orders simultaneously at the end of each interval; single market clearing price)

Frequent Batch Auctions

Budish, Cramton, Shim. *The high-frequency trading arms race: frequent batch auctions as a market design response.* QJE, 2015.

Limitation of FBA

Remark: FBAs work aimed to reveal the structural issues of markets, rather than a quest for definitive computational model(s).

FBA removes continuous-time priority ✓

However, regarding order matching: (?)

- Concurrency is not explicit in the model
- Parallelism is not native to the computation

Moreover, at the structural level: (?)

- Incoming orders are not mandated to always interact first with the resident market[1]

[1] In CLOB, incoming orders always interact with the resident market first. By necessity.

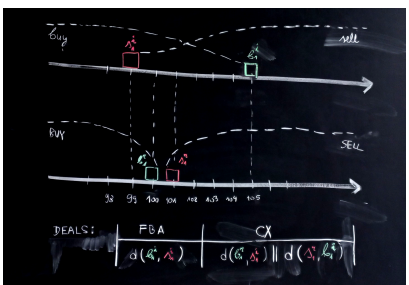
Our solution **CX**: beyond FBA

1. Put time into discrete units ✓ (as in FBA)
2. Enable natural concurrency and parallelism in order matching ✓ ✓
   (while preserving desirable computational and economic properties)

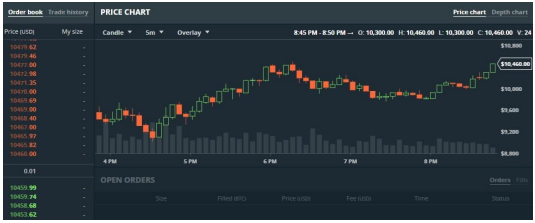
- Incoming order flow is mandated to always interact first with the resident market ✓

CX:

- enabling native concurrency and parallelism in electronic markets' computation
- retaining the structural consistency with CLOB, rathen that departing from it



Clarification: main views of the market



Price chart (more often used)

Depth chart (more holistic view)

From CLOB to FBA to **CX**

Key **starting insight** that **did not exist** and had to be recognized:

- incoming set of orders is a **market in its own right**[2], and therefore
- incoming and resident are two interacting markets

CLOB FBA **CX**

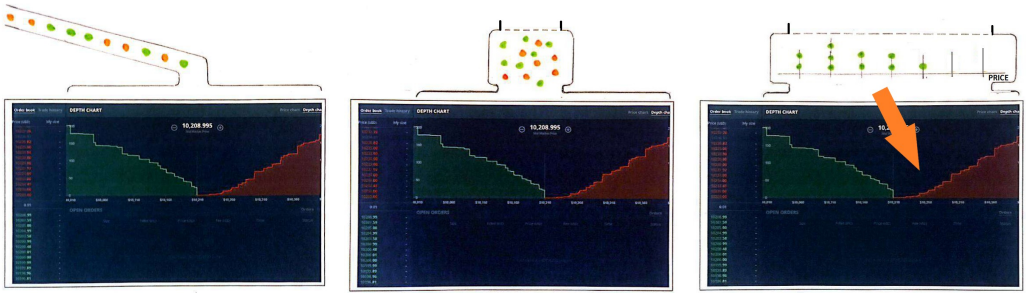
From **sequential**, to **batch**, to **parallel and concurrent processing**

[2] Pseudomarket, to be more precise.

From CLOB to FBA to **CX**

Key **starting insight** that **did not exist** and had to be recognized:

- incoming set of orders is a **market in its own right**[3], and therefore
- incoming and resident are two interacting markets



CLOB FBA **CX**

From **sequential**, to **batch**, to **parallel and concurrent processing**

[3] Pseudomarket, to be more precise.

From CLOB to FBA to **CX**

Key **starting insight** that **did not exist** and had to be recognized:

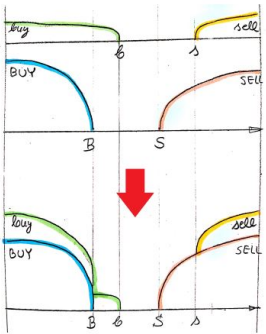
- incoming set of orders is a **market in its own right**[4], and therefore
- incoming and resident are two interacting markets

CLOB FBA **CX**

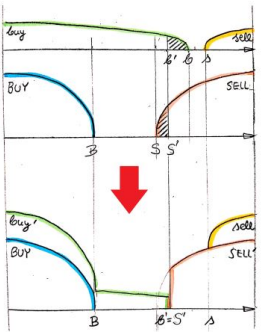
From **sequential**, to **batch**, to **parallel and concurrent processing**

[4] Pseudomarket, to be more precise.

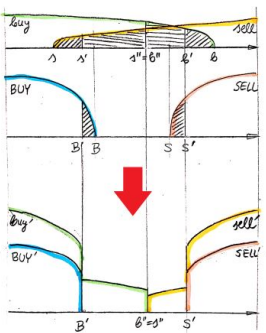
A closer look into **CX** computation



no successful order-matching

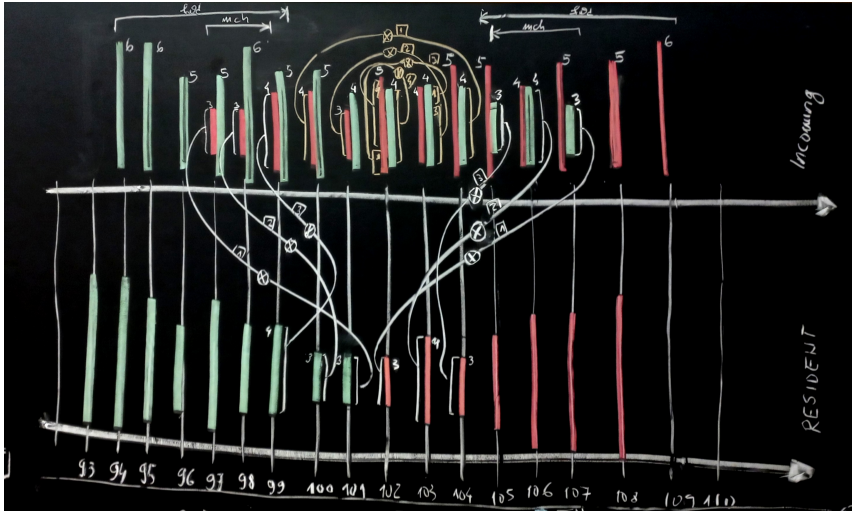


matching on one side



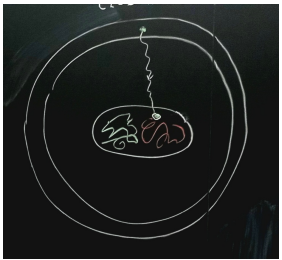
matching in all 3 segments

Illustrating an example computation

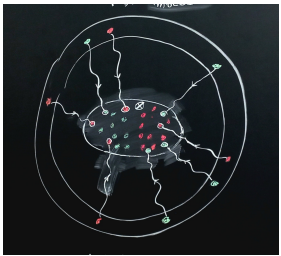


Another view on the 3 market models

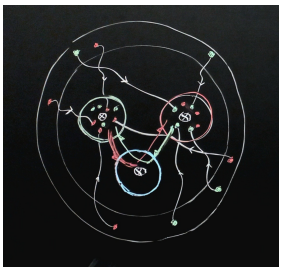
A higher level of abstraction: we emphasize the flow of orders, bringing the model closer to the **Reaction Systems** view of interaction.



CLOB

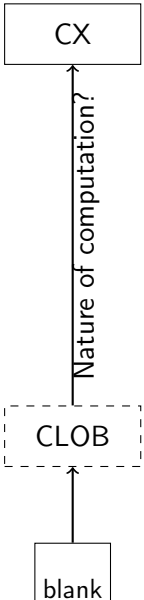


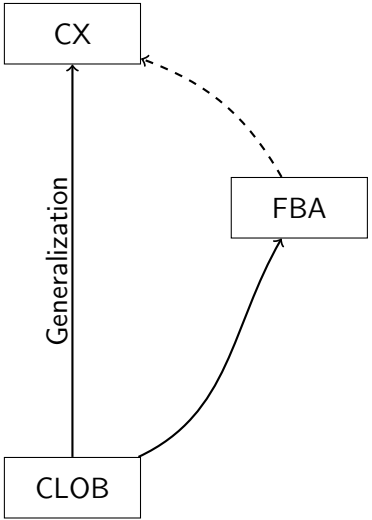
FBA



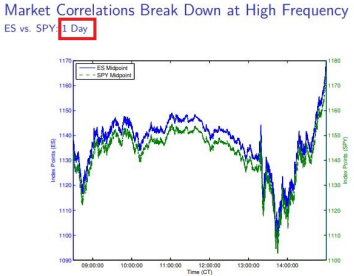
CX

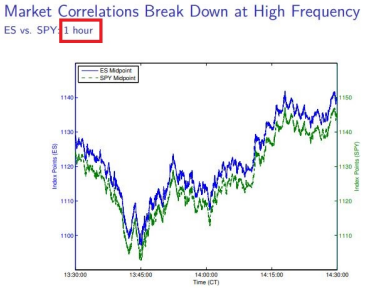
The big picture

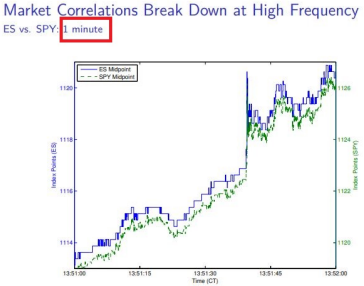


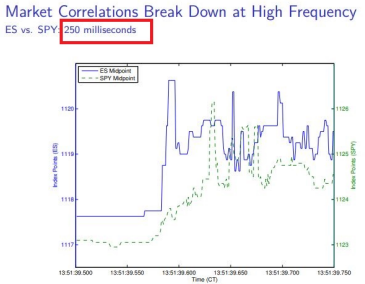


The HFT issue: latency arbitrage in correlated assets, an example







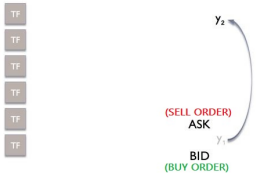


Primarily related to speed; rooted in continuous time

The sniping issue (related to front-running)

"Sniping"

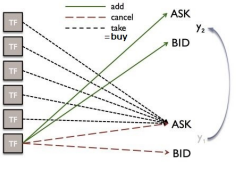
Fundamental value and bid-ask spread



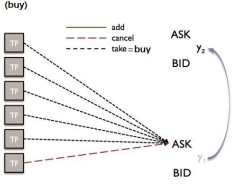
Fundamental value jumps

TFs providing liquidity send messages to cancel old quotes and add new quotes

TFs providing liquidity send messages to cancel old quotes and add new quotes



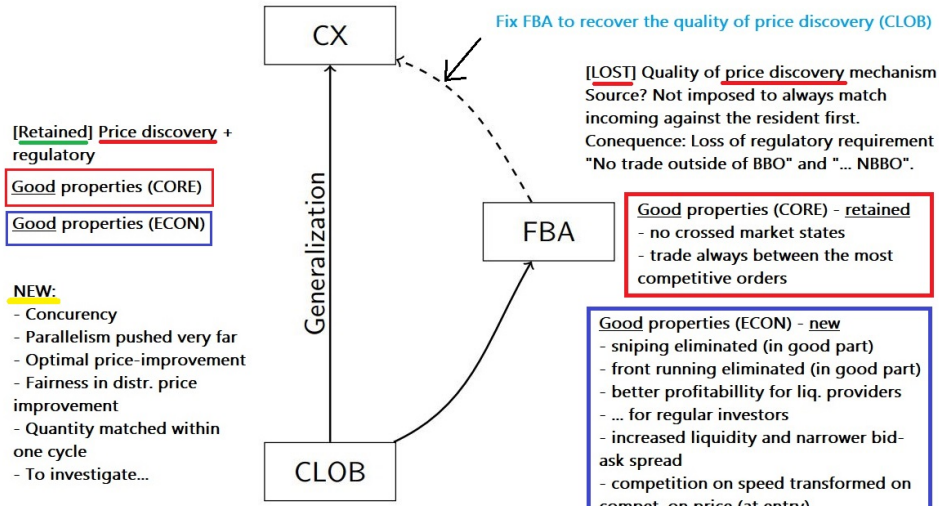
At same time, *other* TFs send messages to "snipe" the stale quotes



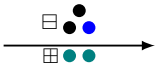
Because the market design processes messages in *serial*, liquidity providers get sniped with probability $\frac{N-1}{N}$ ... even though the information was public and all TFs have the exact same technology

- Even assuming the same speed; primarily rooted in sequential order processing.

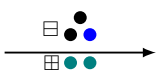
The big picture: properties

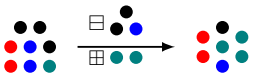


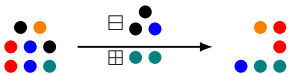
Reaction systems in a nutshell



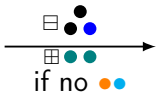
Reaction systems in a nutshell

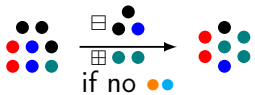


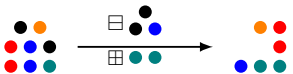




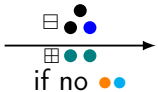
Reaction systems in a nutshell

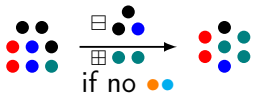


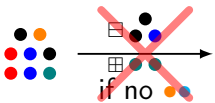




Reaction systems in a nutshell

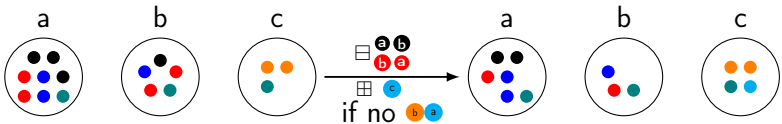






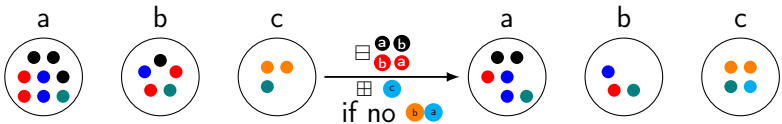
Reaction systems in a nutshell

Locations:

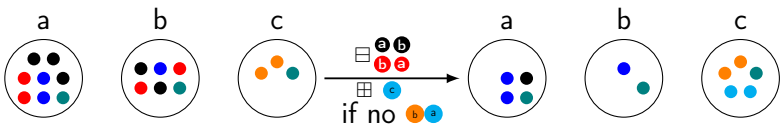


Reaction systems in a nutshell

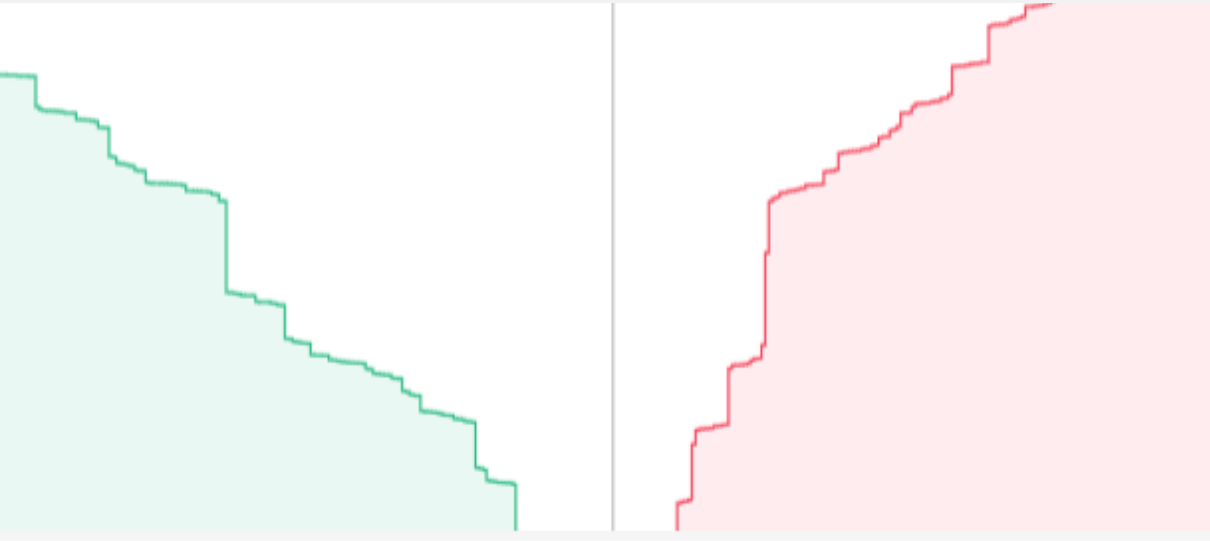
Locations:



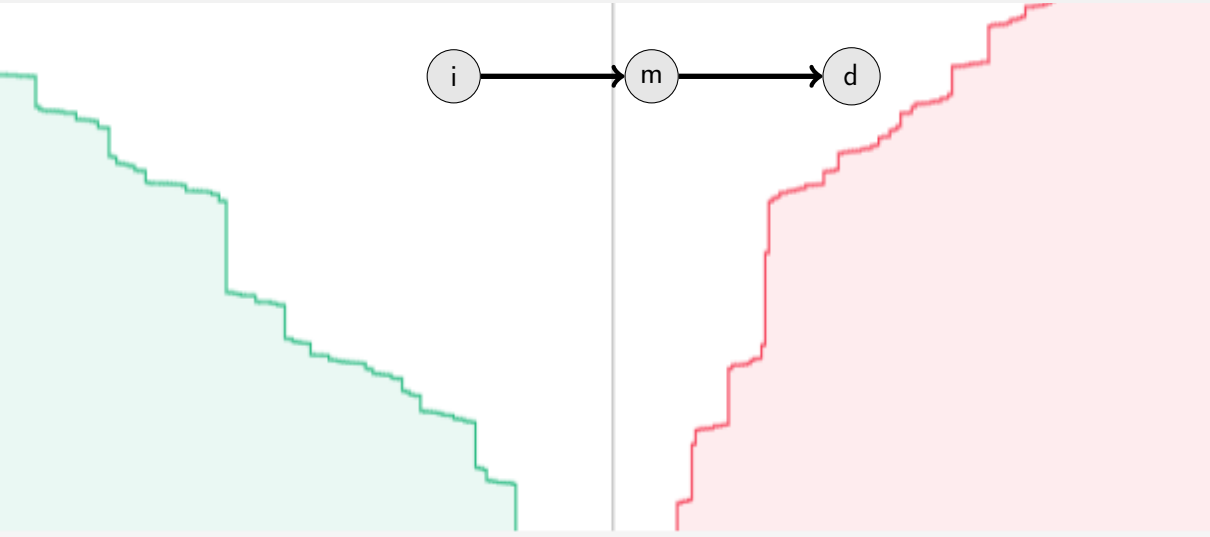
Concurrency:



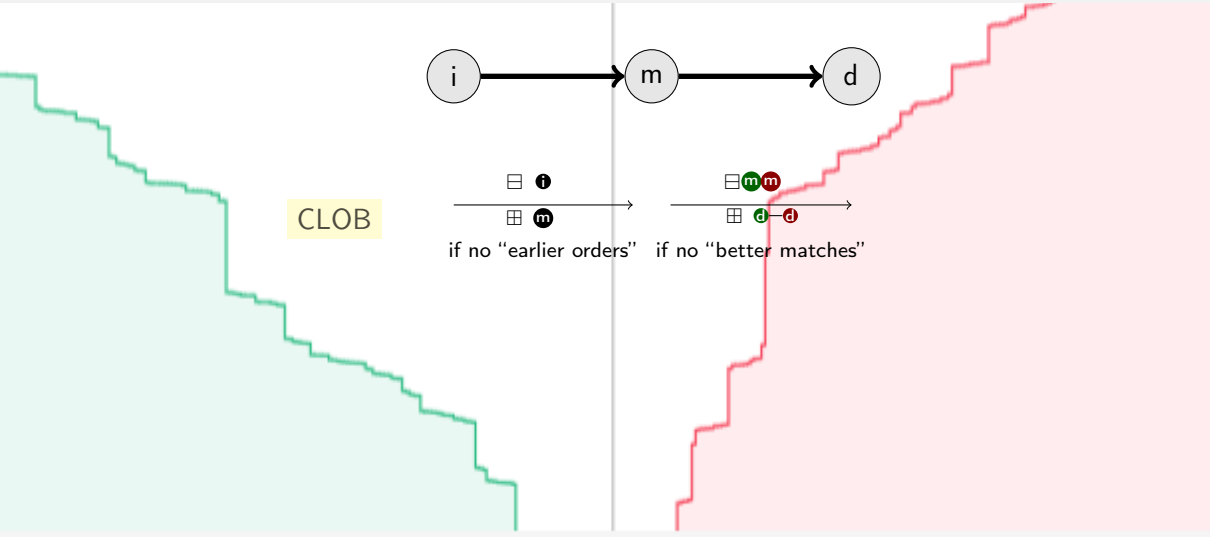
Formalising markets' with reaction systems



Formalising markets' with reaction systems

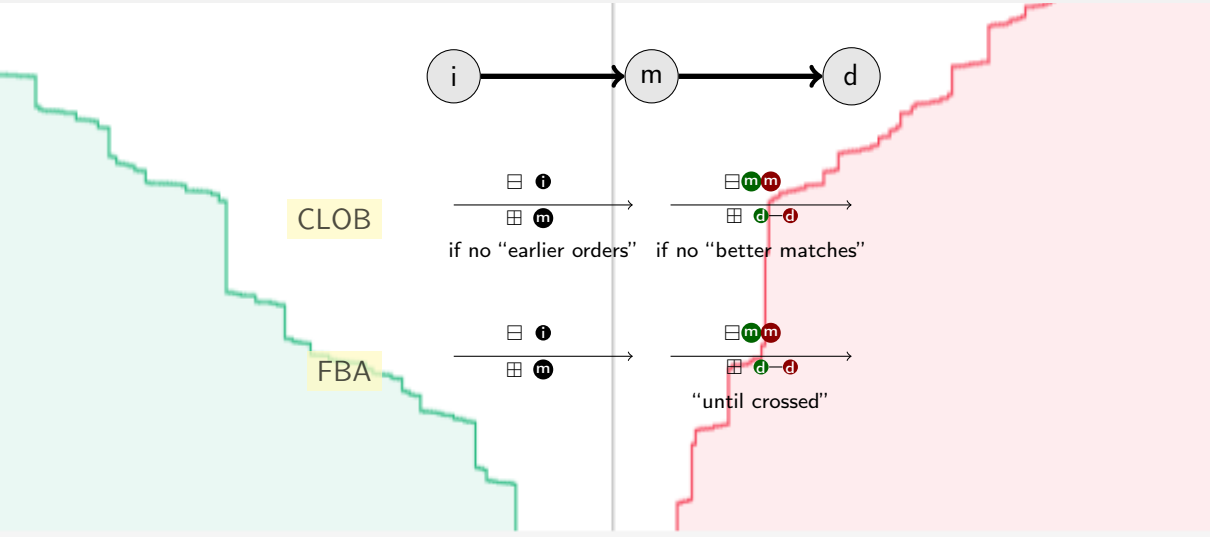


Formalising markets' with reaction systems



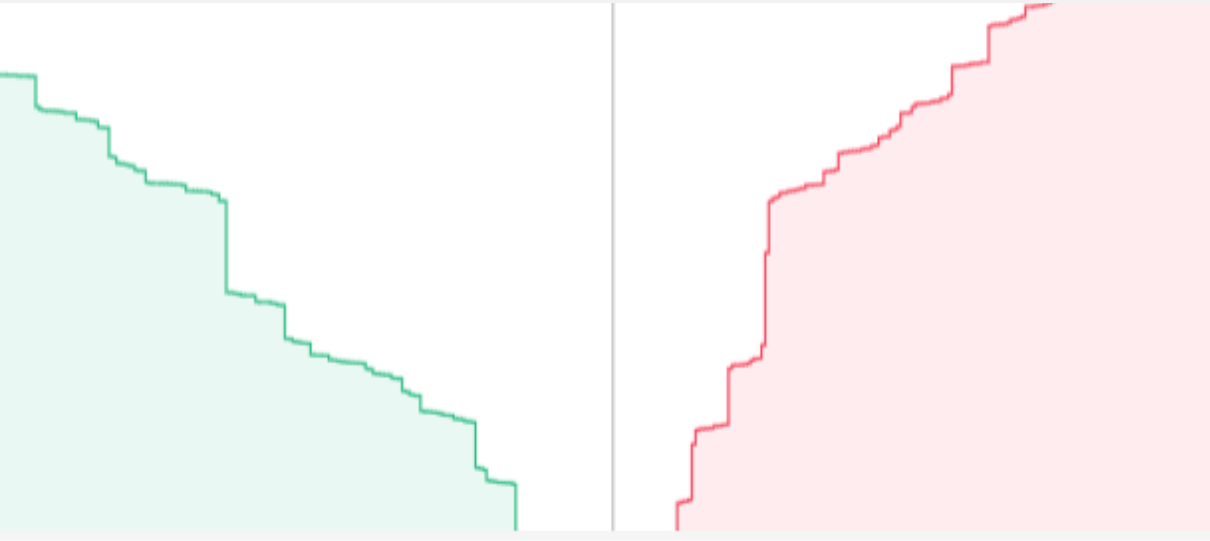
NIT/USDT +87.82% 0.1326 PAXG/USDT -0.46% 5,014.27 AGLD/USDT -0.86% 0.231 SCR

Formalising markets' with reaction systems

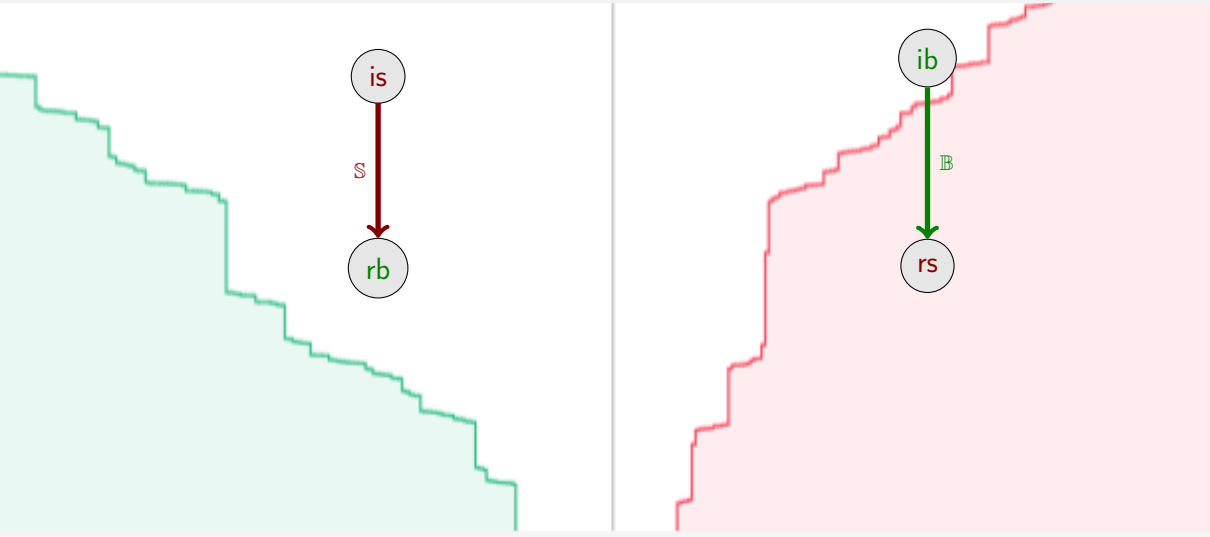


NIT/USDT +87.82% 0.1326 PAXG/USDT -0.46% 5,014.27 AGLD/USDT -0.86% 0.231 SCR

CX: A new market model

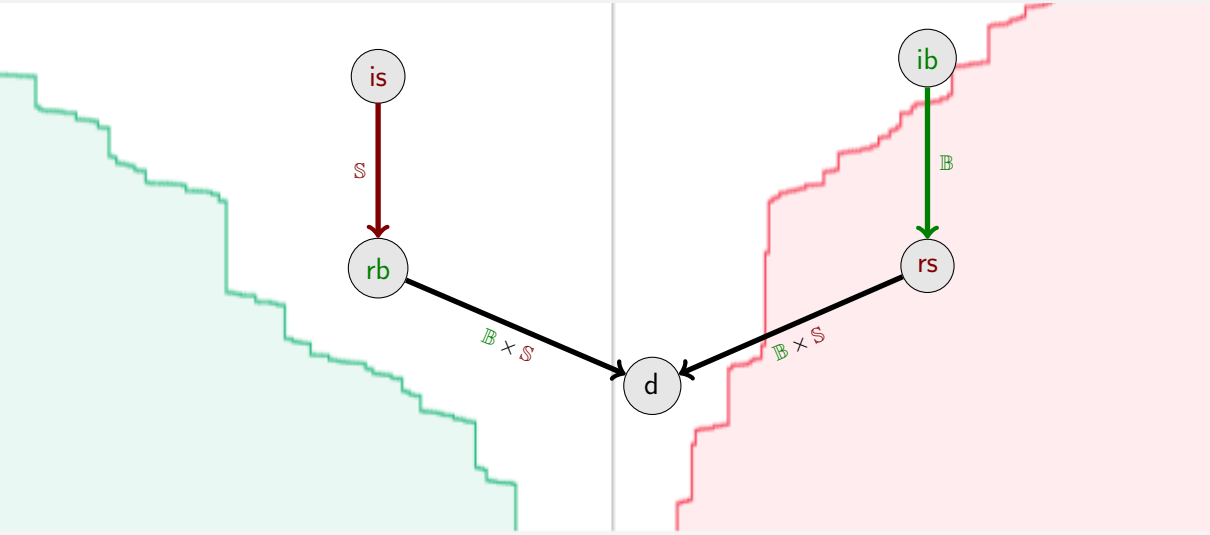


CX: A new market model



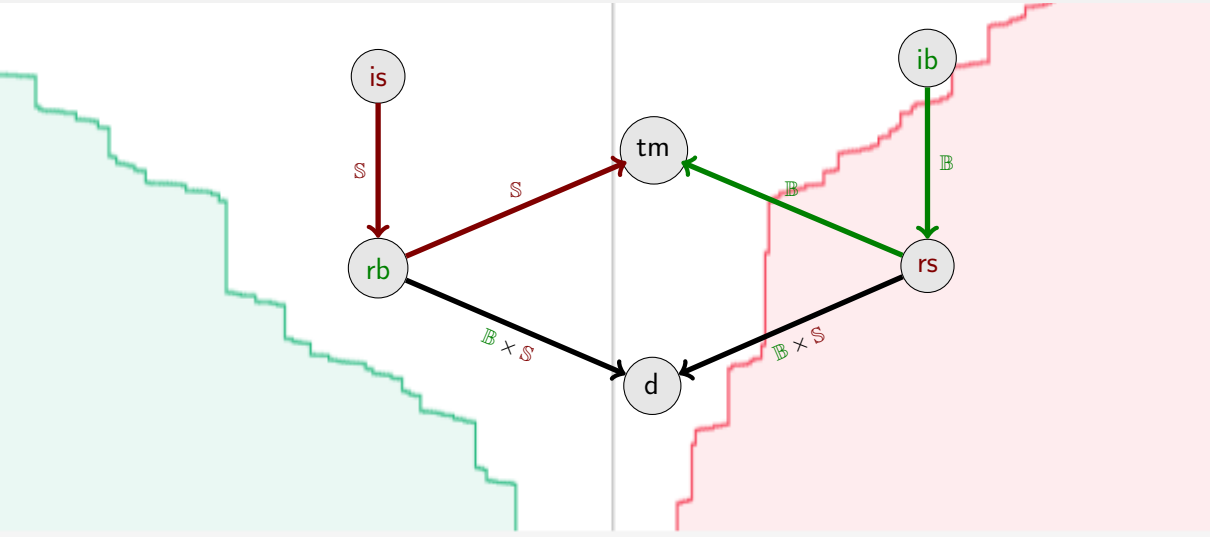
NIT/USDT +87.82% 0.1326 PAXG/USDT -0.46% 5,014.27 AGLD/USDT -0.86% 0.231 SCR

CX: A new market model



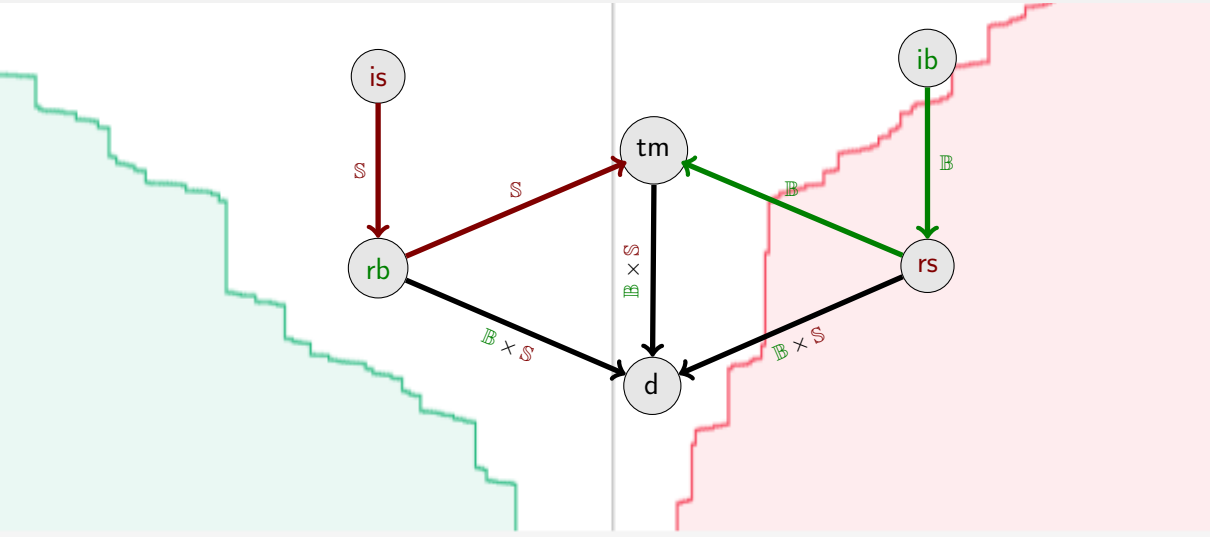
NIT/USDT +87.82% 0.1326 PAXG/USDT -0.46% 5,014.27 AGLD/USDT -0.86% 0.231 SCR

CX: A new market model



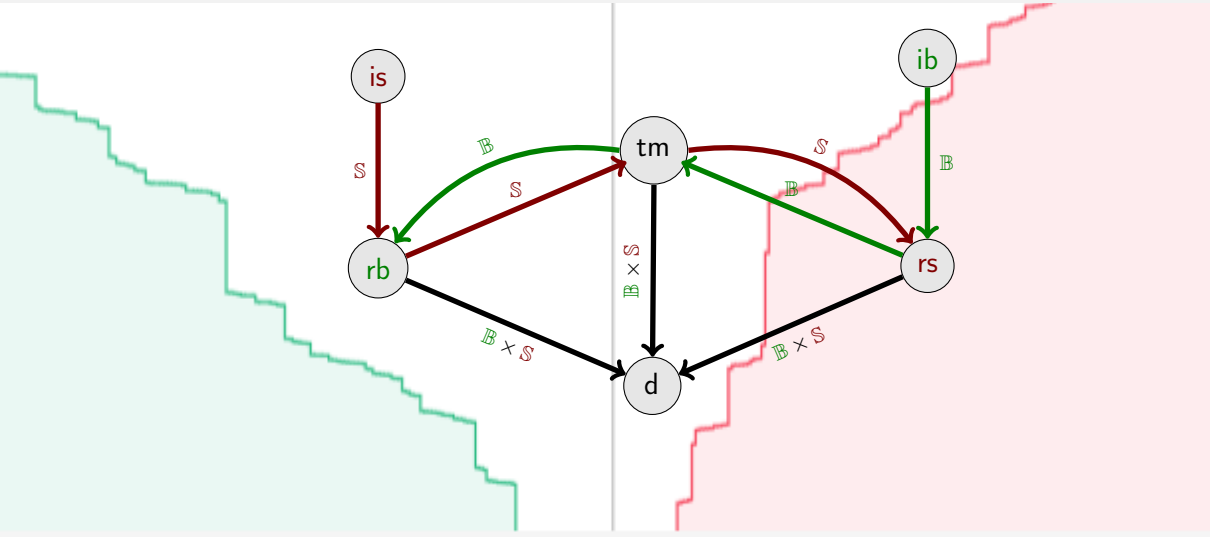
NIT/USDT +87.82% 0.1326 PAXG/USDT -0.46% 5,014.27 AGLD/USDT -0.86% 0.231 SCR

CX: A new market model



NIT/USDT +87.82% 0.1326 PAXG/USDT -0.46% 5,014.27 AGLD/USDT -0.86% 0.231 SCR

CX: A new market model



NIT/USDT +87.82% 0.1326 PAXG/USDT -0.46% 5,014.27 AGLD/USDT -0.86% 0.231 SCR

A glimpse of CX's formalisation

$$
\begin{aligned}
r_{\mathsf{deal}} &= \ell[(\mathsf{tok}(\ell), \mathsf{b}^{@i}, \mathsf{s}^{@j} \ ; \ \mathbb{B}^{>i} \cup \mathbb{S}^{<j} \ ; \ \mathsf{tok}(\ell), \mathsf{d}_{(\mathsf{b}^{@i}, \mathsf{s}^{@j})})] \\
r_{\mathsf{fwd}} &= \ell[(\mathsf{tok}(\ell), e \ ; \ \overline{e} \ ; \ \mathsf{tok}(\ell), \ell'[e])] \\
r_{\mathsf{ctl}} &= \ell[(\mathsf{tok}(\ell) \ ; \ I \ ; \ \ell'[\mathsf{tok}(\ell')])] \qquad \text{with} \qquad \ell \neq \mathsf{tm} \text{ and } \ell \xrightarrow{I} \ell' \text{ in } \mathcal{G} \\
r_{\mathsf{tms}} &= \mathsf{tm}_{(\beta,\sigma \ ; \ \emptyset \ ; \ \tau)} \\
r_{\mathsf{close}} &= \mathsf{tm}_{(\tau \ ; \ \mathbb{B} \cup \mathbb{S} \ ; \ \mathsf{ib}[\beta], \mathsf{is}[\sigma])}
\end{aligned}
$$

Added values

We can prove several properties:

- Market never crossed
- No trade outside current bid or ask
- Order priority respected

Added values

We can prove several properties:

- Market never crossed
- No trade outside current bid or ask
- Order priority respected

We can compare models and simulate them

Added values

We can prove several properties:

- Market never crossed
- No trade outside current bid or ask
- Order priority respected

We can compare models and simulate them

We can specify agents' strategies

- (not covered in this talk)

Added values

We can prove several properties:

- Market never crossed
- No trade outside current bid or ask
- Order priority respected

We can compare models and simulate them

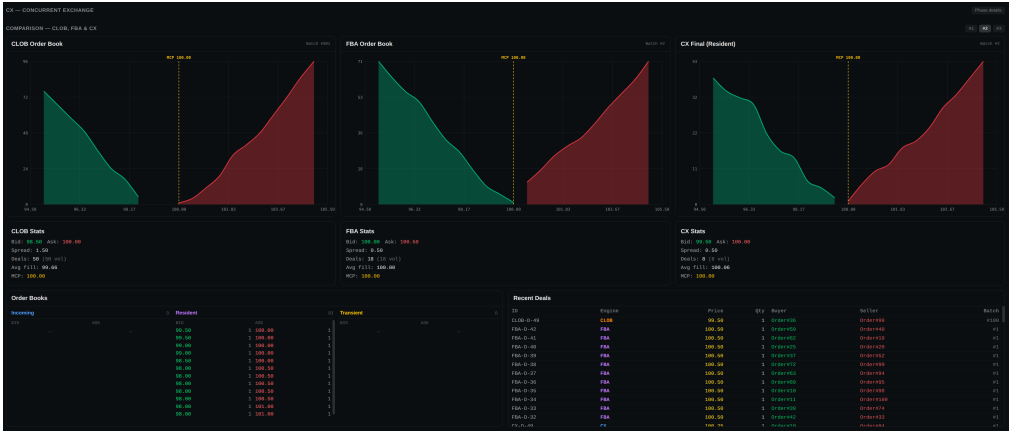
We can specify agents' strategies

- (not covered in this talk)

We conjecture several economic properties, e.g.,:

- Mitigation of front-running regular investors (non-HFT participants)
- Mitigation of sniping (i.e., the predatory removal of liquidity provider quotes)
- Better execution prices and increased profitability for regular investors
- Greater liquidity and narrower bid-ask spreads
- ...

Our prototype tool



References I

[1] Cedric Fournet and George Gonthier. The reflexive CHAM and the join-calculus.
In *Conference Record of POPL '96: The 23rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 372–385, St. Petersburg Beach, Florida, January 1996.

[2] John A. Trono. A new exercise in concurrency.
SIGCSE Bull., 26(3):8–10, September 1994.