Linearity & TLLA 2018

Formalization of Automated Trading Systems in Concurrent Linear Framework (CLF)

Iliano Cervesato, Sharjeel Khan, Giselle Reis, Dragisa Zunic

Carnegie Mellon University

Automated Trading Systems (ATS)

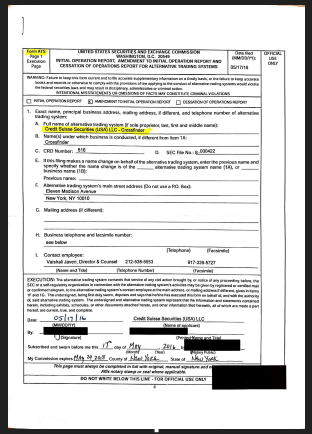
A system with rules for trading securities like stocks or bonds that are executed automatically by a computer.

- Investopedia

Examples:

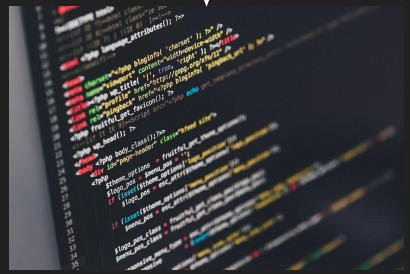
- Public Stock Markets (Nasdaq, NYSE)
- Private Exchanges (Dark pools)

How it works



"Form ATS is designed to enable the Commission to determine whether an alternative trading system subject to Regulation ATS is in compliance with Regulation ATS and other federal securities laws."

ORDER BOOK			RECENT TRADES		
PRICE	QTY	TOTAL	TIME	PRICE	QTY
1215.6	10	120	18:01:03	1214.9	3
1215.5	10	110	11:23:19	1220.7	3
1215.4	10	100	11:23:19	1220.6	37
1215.3	10	90	22:00:41	1219.5	1
1215.2	10	80	22:00:40	1219.5	1
1215.1	10	70	22:00:40	1219.5	1
1215.0	10	60	18:43:19	1223.8	10
1214.9	10	50	18:42:19	1223.8	10
1214.8	10	40	18:42:11	1223.7	10
1214.7	10	30	18:42:11	1223.8	10
1214.6	10	20	18:41:10	1223.0	40
1214.5	10	10	18:28:35	1213.1	10
1214.9 ↓			18:12:22	1212.0	10
3 days @ 1211.0			18:12:22	1212.0	10
1201.0 ❮ 1211.0 ❯ 1221.0			18:12:17	1212.0	10
1213.3	10	10	18:05:47	1203.4	10
1213.2	10	20	18:05:47	1203.3	10
1213.1	10	30	18:05:46	1203.2	10
1213.0	10	40	17:59:20	1203.1	10
1212.9	10	50	17:59:20	1203.1	10
1212.8	10	60	17:58:03	1203.1	10
1212.7	10	70	17:55:35	1199.9	10
1212.6	10	80	17:55:34	1199.8	10
1212.5	10	90	17:55:34	1199.7	10
1212.4	10	100	17:55:34	1199.6	10
1212.3	10	110	17:55:33	1199.5	10
1212.2	10	120	16:43:34	1194.0	10
1212.1	10	130	16:40:04	1193.9	10

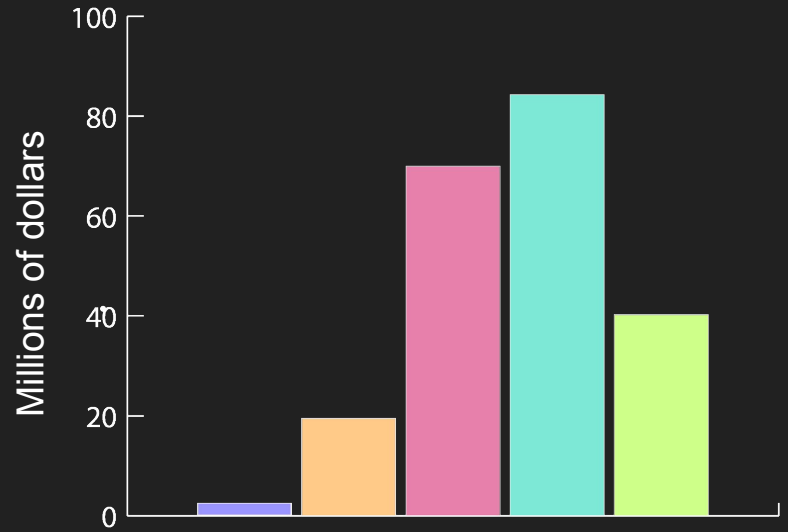


Order queue

Implementation

Automated Trading System

Violations are expensive



Examples of Properties

- Exchanges occur at the **highest of buy** (*bid*) or **lowest of sell** (*ask*) prices [1]
- **No locked/crossed market:** The bid price (maximum buy price) is strictly less than the ask price (minimum sell price)
- **Price-time priority satisfied:** Orders are exchanged first based on price then based on time the order entered the market
- **Order priority is transitive**

[1] Code of Federal Regulations, Title 17, Chapter II, Part 242, Section 242.301, paragraph (b)(3)(iii)(B)
https://www.law.cornell.edu/cfr/text/17/242.301

Motivation

- Hard to reason about the exchange rules when described in natural language
- Properties may be violated because an unforeseen combination of rules reaches a violating state

Can we leverage formal methods to provide better guarantees?

GOAL

Formalize the rules of an archetypal ATS and provide formal proofs for desired properties.

Concurrent Logical Framework (CLF)

Specification of object systems as a set of terms (types) in a fragment of (intuitionistic) linear logic[1]:

$$
\begin{array}{lll}
N, M ::= a \multimap N \mid a \rightarrow N \mid \{P\} \mid \forall x.N \mid a & & \text{(Negative)} \\
P, Q ::= P \otimes Q \mid 1 \mid !a \mid a & & \text{(Positive)}
\end{array}
$$

- Positive formulas are encapsulated in a monad {} (focusing)
- Specifications are executable
- Context (of linear facts) represents the state of the world

1 This is the fragment needed for this work, not the full CLF.

Concurrent Logical Framework (CLF)

Linear implication (⊸) is multiset rewriting (rewrites part of the context)

```
r: coffee ⊗ milk ⊸ {latte}.
```

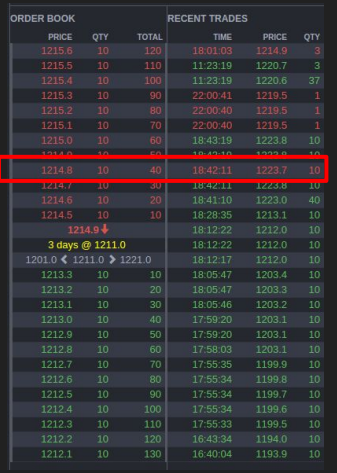
Intuitionistic implication (→) is the typical backward chaining:

```
plus/s: plus M N P → plus (s M) N (s P).
```

Concurrency can be modelled via ⊗

```
fork: proc (par P Q) ⊸ { proc P ⊗ proc Q }.
```

Formalization: main elements

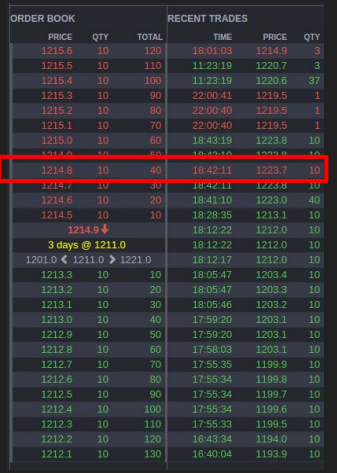


ORDER BOOK			RECENT TRADES		
PRICE	QTY	TOTAL	TIME	PRICE	QTY
1215.6	10	120	18:01:03	1214.9	3
1215.5	10	110	11:23:19	1220.7	3
1215.4	10	100	11:23:19	1220.6	37
1215.3	10	90	22:00:41	1219.5	1
1215.2	10	80	22:00:40	1219.5	1
1215.1	10	70	22:00:40	1219.5	1
1215.0	10	60	18:43:19	1223.8	10
1214.8	10	40	18:42:11	1223.7	10
1214.7	10	30	18:42:11	1223.8	10
1214.6	10	20	18:41:10	1223.0	40
1214.5	10	10	18:28:35	1213.1	10
1214.9 ↓			18:12:22	1212.0	10
3 days @ 1211.0			18:12:22	1212.0	10
1201.0 ❮ 1211.0 ❯ 1221.0			18:12:17	1212.0	10
1213.3	10	10	18:05:47	1203.4	10
1213.2	10	20	18:05:47	1203.3	10
1213.1	10	30	18:05:46	1203.2	10
1213.0	10	40	17:59:20	1203.1	10
1212.9	10	50	17:59:20	1203.1	10
1212.8	10	60	17:58:03	1203.1	10
1212.7	10	70	17:55:35	1199.9	10
1212.6	10	80	17:55:34	1199.8	10
1212.5	10	90	17:55:34	1199.7	10
1212.4	10	100	17:55:34	1199.6	10
1212.3	10	110	17:55:33	1199.5	10
1212.2	10	120	16:43:34	1194.0	10
1212.1	10	130	16:40:04	1193.9	10

order*

* Not facts *per se*.

Formalization: main elements



order*

orderQ

actPrices

* Not facts *per se*.

Formalization: rule

"A limit order is an order to buy or sell a stock at a specific price or better. A buy limit order can only be executed at the limit price or lower, and a sell limit order can only be executed at the limit price or higher."

```
orderQ(front((limit,A,P,ID,N,T),Q)) ⊗
dual(A,A') ⊗
actPrices(A',LP) ⊗
store(A,LP,P) ⊗
priceQ(A,P,L) ⊗
extendP(L,ID,N,T,L')

 ⊸ {priceQ(A',P,L') ⊗ actPrices(A',LP) ⊗ orderQ(Q)}.
```

No exchange can take place

New facts

+ rules for exchanging

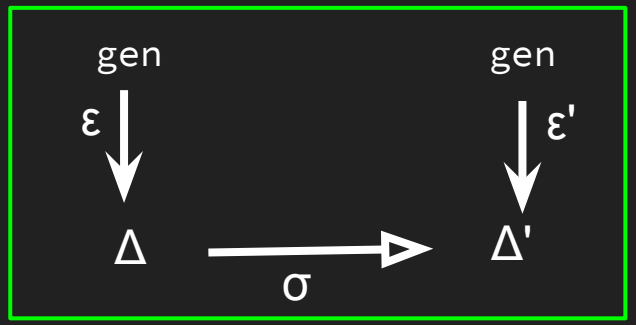
Formalization: in numbers

- Three exchange order types: limit, market, and immediate-or-cancel
- Cancel orders
- ~25 exchange rules in total
- Infrastructure (lists, nats, queues): ~250 lines of code*
- Actual ATS: ~450 loc*

* With comments and empty lines.

Reasoning about CLF specifications

- Ongoing work
- Requires reasoning on states (contexts) and execution traces
- Current proposal: generative grammars



gen is the start symbol of a grammar that only generates context which satisfy the desired property

σ is one step in the execution of the CLF specification

ε and ε' are derivations in the grammar

No locked/crossed market

Property: The bid price (maximum buy price) is strictly less than the ask price (minimum sell price).

Theorem: For every reachable state, if `actPrices(buy, LB)`, `actPrices(sell, LS)`, `maxP(LB, B)`, and `minP(LS,S)` then B < S.

No locked/crossed market

Generating contexts satisfying the properties:

```
gen/00 : gen ⊸ {actPrices(buy, nil) ⊗ actPrices(sell, nil)}.

gen/01 : gen ⊗ (LB ≠ nil) ⊸ {actPrices(buy, LB) ⊗ actPrices(sell, nil)}.

gen/10 : gen ⊸ {actPrices(buy, nil) ⊗ actPrices(sell, LS)}.

gen/11 : gen ⊗ (LB ≠ nil) ⊗ (LS ≠ nil) ⊗ maxP(LB, B) ⊗ minP(LS,S) ⊗ B < S ⊸
         {actPrices(buy, LB) ⊗ actPrices(buy, LS)}.
```

CLF type:

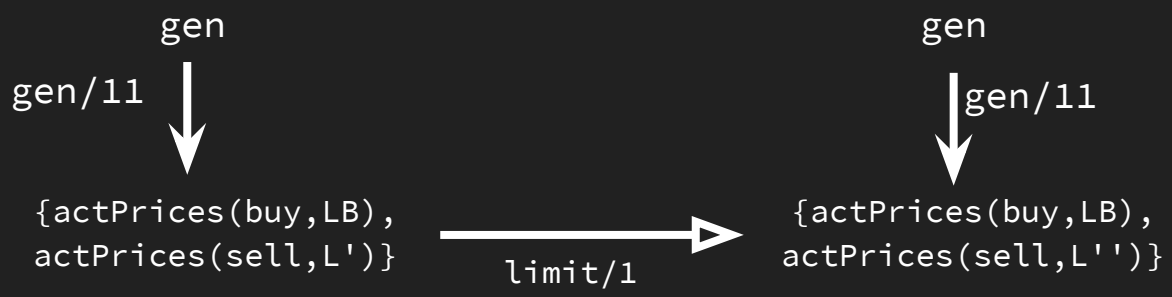
$$\forall_{G(\Sigma)}\delta.\,\forall_{G(\Sigma)}\delta'.\,\Pi\,\epsilon : [\mathbf{gen} \rightsquigarrow^*_{G(\Sigma)} \delta].\,\Pi\,\sigma : [\delta \rightsquigarrow_{\Sigma} \delta'].\,\Pi\,\epsilon' : [\mathbf{gen} \rightsquigarrow^*_{G(\Sigma)} \delta'].\,mtype$$

Proof (Case: the order is not exchanged, A is buy, A' is sell)

```
limit/store :
  orderQ(front((limit,A,P,ID,N,T),Q)) ⊗
  dual(A,A') ⊗
  actPrices(A',LP) ⊗
  store(A,LP,P) ⊗
  priceQ(A,P,L) ⊗
  extendP(L,ID,N,T,L')
  ⊸ {priceQ(A',P,L') ⊗ actPrices(A',LP) ⊗ orderQ(Q)}.
```

Proof (Case: the order is exchanged, A is buy, A' is sell)

```
limit/1:
  orderQ(front((limit,A,P,ID,N,T ),Q)) ⊗
  dual(A,A') ⊗
  actPrices(A',L') ⊗
  exchange(A,L',P,X) ⊗
  priceQ(A',X,consP(ID',N,T',nilP)) ⊗
  remove(L',X,L'')
  ⊸ {orderQ(Q) ⊗ actPrices(A',L'')}
```



Conclusion

- (Modular) executable specification of archetypal orders
- Proofs using the generative grammar method:
  - No locked/crossed market
  - Exchange price is always bid or ask
- More generally: we should use more formalizations and less natural language for regulated systems (but here this is preaching to the choir).

Future Work

- Development of a meta-logic for automating proofs
- Extend specification to more complicated exchange systems
- Prove more properties

Thank you!