LexiFi

*AUTOMATING THE DERIVATIVES MARKET: THE NEED OF A **FORMAL, EXHAUSTIVE AND COMPOSITIONAL ALGEBRA** ALLOWING A UNIFORM SHAREABLE DESCRIPTION OF THE PAYOFF OF ALL KIND OF FINANCIAL CONTRACTS*

Jean-Marc Eber
Founder, CEO of LexiFi

Bloomberg
RiskTech Talks
November 2019, London

Main driver of LexiFi's vision and developments

Question:	has the financial industry adopted a method for rigorously describing financial contract payoffs in a computer system?
Answer:	no
The finance industry	✓ is "highly technological" ✓ imposes/needs precise information exchanges between many different entities
Compare to	car makers, aeronautic industry, building sector, that all use CAD files

THE FINANCIAL INDUSTRY HAS NOT DEVELOPED ANY STANDARD GENERIC DATA FORMAT

A language for describing derivatives?

"One of the challenges that we need to address [. . .] is to have **a common language** to describe derivatives. Every firm uses a different set of terminologies, a different set of representations to describe their derivatives portfolios." *Kenneth Griffin*

Kenneth Griffin, Founder and CEO, Citadel Investment Group, testifying before the U.S. House Committee on Oversight and Government Reform on November 13, 2008.

Image source: C-Span
For more information, please refer to the disclaimer at the last page.

What is (conceptually) easy / difficult to describe or implement

Easy

What can be mapped immediately into a relational database system:

- ✓ Counterparty
- ✓ Trade
- ✓ Underlying references
- ✓ …

Difficult

What appears to have "infinite variability“:

- ✓ Contract "logic“
- ✓ Contract life-cycle
- ✓ Contract payoff

Importance of a contract payoff description

Payoff	is a fundamental part of a financial contract description
“Normal" rights and obligations	typically receive or deliver money or physical goods, take decisions,...
Temporal and logical evolution	depending on "observables" and contract participants decisions
Market usage	informal, verbose description: mechanical treatment impossible, error prone, no industrialization further than "in-house” systems

A GENERIC AND RIGOROUS APPROACH IS NEEDED

Limitations of a contract payoff description

Doesn't describe "everything":

- ✓ What happens when legal situation changes dramatically (example: Brexit)?
- ✓ What happens if a currency or equity appears or disappears (Euro introduction)?
- ✓ There may be even rounding disputes/errors

WE SHOULD MITIGATE PRECISELY WHAT IS PART OF THE SPECIFICATION, WHAT ISN'T

Many stakeholders, many use cases

A financial contract payoff needs to be

- ✓ priced, and its risk managed accordingly (apply or "map" mathematical models, numerical procedures,...)
- ✓ explained, documented to a potential buyer
- ✓ executed over time, when uncertainty resolves (life-cycle management)
- ✓ accessible to regulators or law enforcement entities
- ✓ accessible to all kind of analytical tools: statistics, data analysis, AI,...

Fully expose payoff semantics!

Industrial fragmentation and specialization (cloud, exchange of documents, APIs, Blockchain, regulation,...) makes a shareable rigorous description necessary

DIVERGENT NEEDS MAKE THE DESIGN OF A PAYOFF SPECIFICATION FORMALISM SURPRISINGLY DIFFICULT

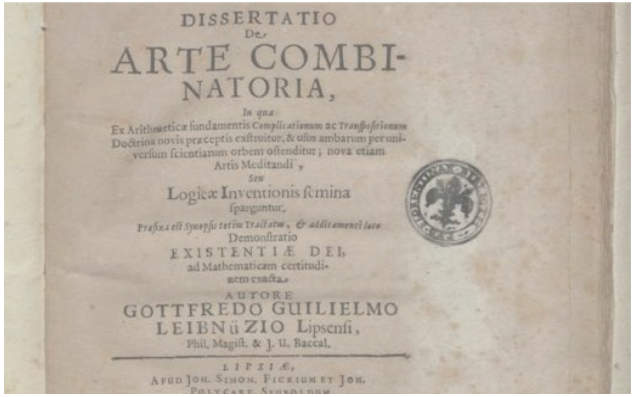
Goal

A contract payoff definition that can be read by a human being, efficiently processed by a computer, exchanged between market participants, and that satisfies three main goals:

- ✓ Describe the rights and obligations of the parties both precisely and exhaustively avoiding future disputes
- ✓ Lend itself to manipulations of various sorts, for example, for the purpose of pricing the contract and its credit risk, managing its clauses automatically, provide interactive simulation tools or producing cashflow forecasts
- ✓ Reflect the evolution of the contract through time (life-cycle management)

Avoiding future disputes: an old idea

"[...] if controversies were to arise, there would be no more need of disputation between two philosophers than between two calculators. For it would suffice for them to take their pencils in their hands and to sit down at the abacus, and say to each other (and if they so wish also to a friend called to help): Let us calculate."


DISSERTATIO
De
ARTE COMBI-
NATORIA,
In qua
Ex Arithmeticæ fundamentis Complicationum ac Transpositionum
Doctrina novis præceptis exstruitur, & usus ambarum per uni-
versum scientiarum orbem ostenditur; nova etiam
Artis Meditandi,
Seu
Logicæ Inventionis semina
sparguntur,
Præsixa est Synopsis tertiæ Tractatus, & additamenti loco
Demonstratio
EXISTENTIÆ DEI,
ad Mathematicam certitudi-
nem exacta.
AUTORE
GOTTFREDO GUILIELMO
LEIBNÜ ZIO Lipsensi,
Phil. Magist. & J. U. Baccal.
LIPSIÆ,
APUD JOH. SIMON. FICKIUM ET JOH.

"quando orientur controversiae, non magis disputatione opus erit inter duos philosophus, quam inter duos computistas. Sufficiet enim calamos in manus sumere sedereque ad abacos, et sibi mutuo (accito si placet amico) dicere: calculemus"

Gottfried Wilhelm von Leibniz,
"Dissertatio de Arte Combinatoria", 1666

For more information, please refer to the disclaimer at the last page.

Implementation cost amortization by genericity and global coverage

Support, once for all, all kind of payoffs, all underlyings, all markets etc.;
therefore, amortization over:

Functionalities	benefits to many processes (front-office, back-office, regulation, marketing,...)
Time	designed to last for the foreseeable future (still usable/valid in 20 years?)
Space	potentially world-wide covering (Asia's Accumulators, Europe's Autocalls, etc.)

Preferred methodology when suggesting a payoff description formalism

- ✓ **Self-contained**: don't depend on other documents or rules
- ✓ **Small:** "minimalist"
- ✓ **Precise:** to avoid divergent interpretations
- ✓ **Fundamental:** focus on concepts (difficult), not on syntax (easy)
- ✓ **Implementable**: exhibit an existing implementation
- ✓ **Generic:** not tied to any specific use
- ✓ **Extensible:** by design

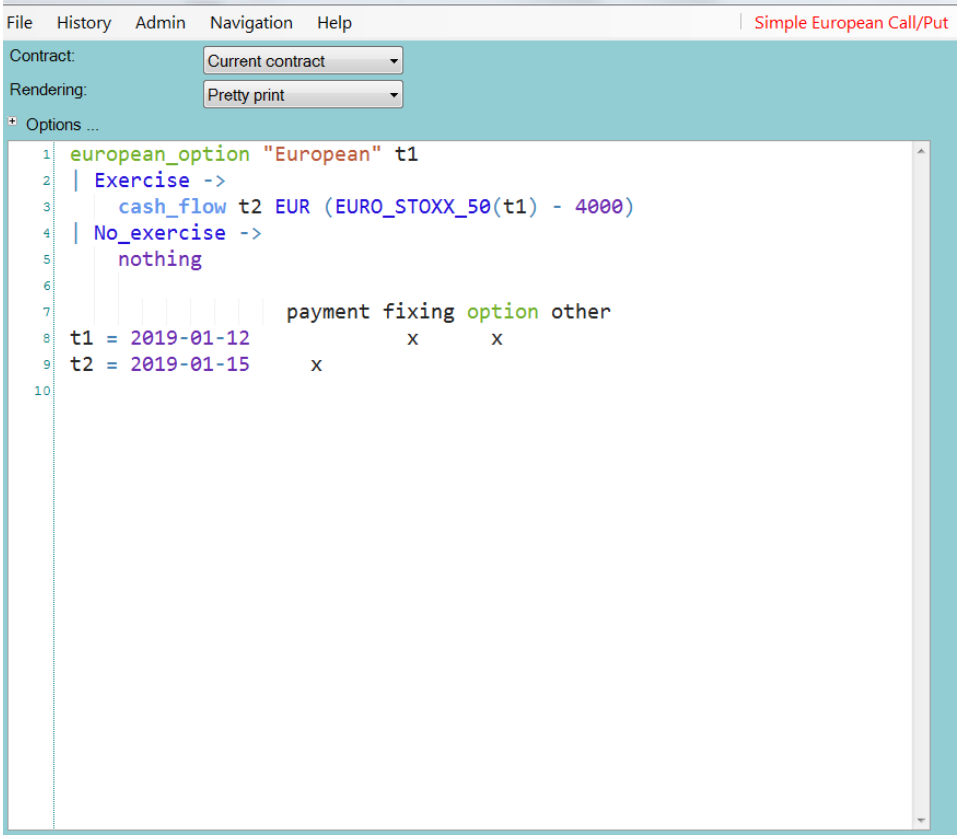
Our approach

For fulfilling all these requirements, we suggest that a payoff formalism should:

- ✓ Be a compositional algebra, defined with a limited number of basic combinators
- ✓ Include lessons learned from theoretical computer science
- ✓ Be as small as possible wrt. expressivity
- ✓ Have a compositional semantics (only "understanding" all sub-expressions of an expression is needed for "understanding" an expression)
- ✓ Not be considered as a "program" or "script", but as a value, that can easily be analyzed, or even transformed
- ✓ Be itself potentially subject to formal analysis (axiomatization, rewriting systems, machine-checked proofs,...)

ANALOGY WITH ALGEBRA: $\max(10\,X + Y, 0) \times 2$

Representation of a call (simplified pretty-print for readability)



Analogy with Algebra

Term sheet: ". . . one should obtain unknown values X and Y, take the sum of 10 times X and Y, replace it with zero if it is negative and multiply by two. . . "

Formal representation: $R = \max(10\,X + Y, 0) \times 2$; **possible uses:**

Pricing	calculate R's expectation under a hypothesis of joint probability distribution of x and y
Lifecycle calendar	find that there are two unknowns, x and y ("calendar of future needed fixings")
Capital protection?	yes, by proving that R is always positive

BUT WAIT: EXPRESSIONS CAN MOREOVER BE TRANSFORMED…

Life-cycle Management "for free"

Contract description transformation, similar to usual algebra:

State 0:	**max(10 X + Y, 0) × 2**
Fixing:	X = 5
State 1:	**max(50 + Y, 0) × 2**
Fixing:	Y = 3
State 2:	**106**

- ✓ Payoff description simplifies as uncertainty resolves
- ✓ This approach formalizes (and allows for implementation) life-cycle management.

max(10 X + Y, 0) × 2	is the initial contract
max(50 + Y, 0) × 2	current (“simplified”) contract
[X = 5; Y = 3]	an audit trail of past observations etc.

BECOMES A STATE-TRANSITION SYSTEM, STATE BEING THE "CURRENT" PAYOFF DESCRIPTION, TRANSITIONS FIRED BY EXTERNAL OBSERVATIONS OR EVENTS, TRANSITIONS MAY HAVE SIDE EFFECTS (TYPICALLY PAYMENTS)

No slide-ware, but field proven technology

Used and enhanced since nearly two decades in demanding industrial software platforms and services

Used for all types of financial contracts: from simplest to most complex payoffs

Solves "once for all" many challenging problems as it allows to:

- ✓ generate highly efficient pricing code
- ✓ score potential applicable pricing models to point out the best model to use (given a balance between speed and preciseness)
- ✓ provide interactive simulation tools
- ✓ implement contract life-cycle management with perfect pricing synchronization
- ✓ generate regulatory documents (KID)
- ✓ …

STANDARD PROPOSITION TO BE PUBLISHED BY LEXIFI Q1/Q2 2020

Disclaimer

This presentation is for informational purposes only. LEXIFI SAS MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, IN THIS SUMMARY. MLFi includes all or parts of the OCaml system developed by INRIA and its contributors. LexiFi, LexiFi Apropos, Instrument Box and MLFi are either trademarks or registered trademarks of LexiFi SAS in France and/or other countries. **ALL OTHER COMPANY AND PRODUCT NAMES REFERENCED IN THIS DOCUMENT ARE USED FOR IDENTIFICATION PURPOSES ONLY AND MAY BE TRADE NAMES OR TRADEMARKS OF THEIR RESPECTIVE OWNERS.**