# Certified Management of Financial Contracts

**Patrick Bahr** Jost Berthold Martin Elsman

University of Copenhagen, Department of Computer Science
(DIKU)

26th Nordic Workshop on Programming Theory, 2014

# Introduction

## What are financial contracts?

- stipulate future transactions between different parties
- have time constraints
- may depend on stock prices, exchange rates etc.

# Introduction

## What are financial contracts?

- stipulate future transactions between different parties
- have time constraints
- may depend on stock prices, exchange rates etc.

### Example (Foreign Exchange Option)

At any time within the next 90 days, party X may decide to buy USD 100 from party Y, for a fixed rate $r$ of Danish Kroner.

# Introduction

## What are financial contracts?

- stipulate future transactions between different parties
- have time constraints
- may depend on stock prices, exchange rates etc.

### Example (Foreign Exchange Option)

At any time within the next 90 days, party X may decide to buy USD 100 from party Y, for a fixed rate $r$ of Danish Kroner.

## Goals

- Express such contracts in a formal language
- Symbolic manipulation and analysis of such contracts.

# Introduction

## What are financial contracts?

- stipulate future transactions between different parties
- have time constraints
- may depend on stock prices, exchange rates etc.

### Example (Foreign Exchange Option)

At any time within the next 90 days, party X may decide to buy USD 100 from party Y, for a fixed rate $r$ of Danish Kroner.

## Goals

- Express such contracts in a formal language
- Symbolic manipulation and analysis of such contracts.
- Formally verified!

# Contract Language Goals in Detail

- **Compositionality**.
  Contracts are time-relative $\Rightarrow$ straightforward compositionality
- **Multi-party**.
  Specify obligations and opportunities for multiple parties,
  (which opens up the possibility for specifying portfolios)
- **Contract management**.
  Contracts can be managed and symbolically evolved;
  a contract gradually reduces to the empty contract.
- **Contract utilities (symbolic)**.
  Contracts can be analysed in a variety of ways
- **Contract pricing (numerical, staged)**.
  Code for payoff can be generated from contracts
  (input to a stochastic pricing engine)

# Example

## Contract in natural language

- At any time within the next 90 days,
- party X may decide to
- buy USD 100 from party Y,
- for a fixed rate $r$ of Danish Kroner.

# Example

## Contract in natural language

- At any time within the next 90 days,
- party X may decide to
- buy USD 100 from party Y,
- for a fixed rate *r* of Danish Kroner.

## Translation into contract language

$$
\begin{aligned}
&\mathit{if}(\mathit{obs}_{\mathbb{B}}(X,0), 90, \mathit{trade}, \mathit{zero}) \\
&\quad \text{where} \quad \mathit{trade} = \mathit{scale}(100, \mathit{both}(\mathit{transfer}(Y,X,\mathrm{USD}), \mathit{pay})) \\
&\quad \phantom{\text{where} \quad \mathit{trade}} \mathllap{\mathit{pay}} = \mathit{scale}(r, \mathit{transfer}(X,Y,\mathrm{DKK}))
\end{aligned}
$$

# Contributions

- Denotational semantics based on cash-flows
- Reduction semantics (sound and complete)
- Correctness proofs for common contract analyses and transformations
- Formalised in the Coq theorem prover
- Certified implementation via code extraction

# An Overview of the Contract Language

## Core Calculus of Contracts

$$\begin{aligned}
\mathit{zero} &: \mathsf{Contr} \\
\mathit{transfer} &: \mathsf{Party} \times \mathsf{Party} \times \mathsf{Currency} \to \mathsf{Contr} \\
\mathit{both} &: \mathsf{Contr} \times \mathsf{Contr} \to \mathsf{Contr} \\
\mathit{scale} &: \mathsf{Expr}_{\mathbb{R}} \times \mathsf{Contr} \to \mathsf{Contr} \\
\mathit{translate} &: \mathbb{N} \times \mathsf{Contr} \to \mathsf{Contr} \\
\mathit{if} &: \mathsf{Expr}_{\mathbb{B}} \times \mathbb{N} \times \mathsf{Contr} \times \mathsf{Contr} \to \mathsf{Contr}
\end{aligned}$$

# An Overview of the Contract Language

## Core Calculus of Contracts

$$
\begin{aligned}
\mathit{zero} &: \mathsf{Contr} \\
\mathit{transfer} &: \mathsf{Party} \times \mathsf{Party} \times \mathsf{Currency} \to \mathsf{Contr} \\
\mathit{both} &: \mathsf{Contr} \times \mathsf{Contr} \to \mathsf{Contr} \\
\mathit{scale} &: \mathsf{Expr}_{\mathbb{R}} \times \mathsf{Contr} \to \mathsf{Contr} \\
\mathit{translate} &: \mathbb{N} \times \mathsf{Contr} \to \mathsf{Contr} \\
\mathit{if} &: \mathsf{Expr}_{\mathbb{B}} \times \mathbb{N} \times \mathsf{Contr} \times \mathsf{Contr} \to \mathsf{Contr}
\end{aligned}
$$

## Expression Language

$\mathsf{Expr}_{\mathbb{R}}$, $\mathsf{Expr}_{\mathbb{B}}$: real-valued resp. Boolean-valued expressions.

# An Overview of the Contract Language

## Core Calculus of Contracts

$$
\begin{aligned}
\mathit{zero} &: \mathsf{Contr} \\
\mathit{transfer} &: \mathsf{Party} \times \mathsf{Party} \times \mathsf{Currency} \to \mathsf{Contr} \\
\mathit{both} &: \mathsf{Contr} \times \mathsf{Contr} \to \mathsf{Contr} \\
\mathit{scale} &: \mathsf{Expr}_{\mathbb{R}} \times \mathsf{Contr} \to \mathsf{Contr} \\
\mathit{translate} &: \mathbb{N} \times \mathsf{Contr} \to \mathsf{Contr} \\
\mathit{if} &: \mathsf{Expr}_{\mathbb{B}} \times \mathbb{N} \times \mathsf{Contr} \times \mathsf{Contr} \to \mathsf{Contr}
\end{aligned}
$$

## Expression Language

$\mathsf{Expr}_{\mathbb{R}}$, $\mathsf{Expr}_{\mathbb{B}}$: real-valued resp. Boolean-valued expressions.

$$
\begin{aligned}
\mathit{obs}_\alpha &: \mathsf{Label}_\alpha \times \mathbb{Z} \to \mathsf{Expr}_\alpha \\
\mathit{acc}_\alpha &: (\mathsf{Expr}_\alpha \to \mathsf{Expr}_\alpha) \times \mathbb{N} \times \mathsf{Expr}_\alpha \to \mathsf{Expr}_\alpha
\end{aligned}
$$

# Example: Asian Option

$$
\mathit{translate}(90, \mathit{if}(\mathit{obs}_{\mathbb{B}}(X, 0), 0, \mathit{trade}, \mathit{zero}))
$$

$$
\begin{aligned}
\text{where} \quad \mathit{trade} &= \mathit{scale}(100, \mathit{both}(\mathit{transfer}(Y, X, \mathrm{USD}), \mathit{pay})) \\
\mathit{pay} &= \mathit{scale}(\mathit{rate}, \mathit{transfer}(X, Y, \mathrm{DKK})) \\
\mathit{rate} &= \frac{1}{30} \cdot \mathit{acc}(\lambda r.r + \mathit{obs}_{\mathbb{R}}(\mathrm{FX\ USD/DKK}, 0), 30, 0)
\end{aligned}
$$

# Denotational Semantics

The semantics of a contract is given by the cash-flow it stipulates.

$$\mathcal{C}\left[\!\left[\cdot\right]\!\right]_{\cdot} : \mathsf{Contr} \qquad\qquad \to \mathsf{CashFlow}$$

# Denotational Semantics

The semantics of a contract is given by the cash-flow it stipulates.

$$\mathcal{C}\left[\!\left[\cdot\right]\!\right]_{\cdot} : \mathsf{Contr} \qquad \rightarrow \mathsf{CashFlow}$$

$$\begin{aligned}\mathsf{CashFlow} &= \mathbb{N} \rightharpoonup \mathsf{Transactions}\\ \mathsf{Transactions} &= \mathsf{Party} \times \mathsf{Party} \times \mathsf{Currency} \rightarrow \mathbb{R}\end{aligned}$$

# Denotational Semantics

The semantics of a contract is given by the cash-flow it stipulates.

$$\mathcal{C}[\![\cdot]\!]_{\cdot} : \mathsf{Contr} \times \mathsf{Env} \to \mathsf{CashFlow}$$
$$\mathsf{Env} = \mathsf{Label} \times \mathbb{Z} \rightharpoonup \mathbb{B} \cup \mathbb{R}$$

$$\mathsf{CashFlow} = \mathbb{N} \rightharpoonup \mathsf{Transactions}$$
$$\mathsf{Transactions} = \mathsf{Party} \times \mathsf{Party} \times \mathsf{Currency} \to \mathbb{R}$$

# Denotational Semantics

The semantics of a contract is given by the cash-flow it stipulates.

$$\mathcal{C}\left[\!\left[\cdot\right]\!\right]_{\cdot} : \mathsf{Contr} \times \mathsf{Env} \to \mathsf{CashFlow}$$
$$\mathsf{Env} = \mathsf{Label}_{\alpha} \times \mathbb{Z} \rightharpoonup \alpha$$

$$\mathsf{CashFlow} = \mathbb{N} \rightharpoonup \mathsf{Transactions}$$
$$\mathsf{Transactions} = \mathsf{Party} \times \mathsf{Party} \times \mathsf{Currency} \to \mathbb{R}$$

# Contract Analyses

## Examples

- contract dependencies
- contract causality
- contract horizon

# Contract Analyses

## Examples

- contract dependencies
- contract causality
- contract horizon

## Semantics vs. Syntax

- these analyses have precise semantic definition
- they cannot be effectively computed
- we provide sound approximations, e.g. type system

# Contract Transformations

### Contract equivalences

When can we replace a sub-contract with another one, without changing the semantics of the contract?

### Reduction semantics

What does the contract look like after $n$ days have passed?

### Contract Specialisation

What does the contract look like after we learned the actual value of some observables?

# Contract Equivalences

$$
\begin{aligned}
translate(d, zero) &\simeq zero \\
scale(r, zero) &\simeq zero \\
scale(0, c) &\simeq zero \\
both(c, zero) &\simeq c \\
scale(s_1, scale(s_2, c)) &\simeq scale(s_1 \cdot s_2, c) \\
translate(d_1, translate(d_2, c)) &\simeq translate(d_1 + d_2, c) \\
translate(d, both(c_1, c_2)) &\simeq both(translate(d, c_1), translate(d, c_2)) \\
scale(x, both(c_1, c_2)) &\simeq both(scale(x, c_1), scale(x, c_2)) \\
translate(d, scale(s, c)) &\simeq scale(s/d, translate(d, c)) \\
translate(d, if(b, e, c_1, c_2)) &\simeq \\
&\quad if(b/d, e, translate(d, c_1), translate(d, c_2)) \\
both(scale(x, transfer(a, b, c)), &scale(y, transfer(a, b, c))) \\
&\simeq scale(x + y, transfer(a, b, c))
\end{aligned}
$$

# Reduction Semantics

$$c \overset{\tau}{\Longrightarrow}_{\rho} c'$$

# Reduction Semantics

$$c \overset{\tau}{\Longrightarrow}_{\rho} c'$$

$$\frac{}{\mathit{transfer}(p_1, p_2, c) \overset{\tau_{p_1,p_2,c}}{\Longrightarrow}_{\rho} \mathit{zero}}$$

# Reduction Semantics

$$c \stackrel{\tau}{\Longrightarrow}_\rho c'$$

$$\frac{}{\mathit{transfer}(p_1, p_2, c) \stackrel{\tau_{p_1,p_2,c}}{\Longrightarrow}_\rho \mathit{zero}} \qquad \frac{c \stackrel{\tau}{\Longrightarrow}_\rho c' \quad \mathcal{E}\,[\![e]\!]_\rho = v}{\mathit{scale}(e, c) \stackrel{v*\tau}{\Longrightarrow}_\rho \mathit{scale}(e/{-1}, c')}$$

# Reduction Semantics

$$c \stackrel{\tau}{\Longrightarrow}_{\rho} c'$$

$$\frac{}{transfer(p_1, p_2, c) \stackrel{\tau_{p_1, p_2, c}}{\Longrightarrow}_{\rho} zero} \qquad \frac{c \stackrel{\tau}{\Longrightarrow}_{\rho} c' \quad \mathcal{E}[\![e]\!]_{\rho} = v}{scale(e, c) \stackrel{v * \tau}{\Longrightarrow}_{\rho} scale(e/-1, c')}$$

$$\vdots$$

# Reduction Semantics

$$c \overset{\tau}{\Longrightarrow}_{\rho} c'$$

$$\frac{}{\mathit{transfer}(p_1, p_2, c) \overset{\tau_{p_1,p_2,c}}{\Longrightarrow}_{\rho} \mathit{zero}} \qquad \frac{c \overset{\tau}{\Longrightarrow}_{\rho} c' \quad \mathcal{E}\left[\!\left[ e \right]\!\right]_{\rho} = v}{\mathit{scale}(e, c) \overset{v * \tau}{\Longrightarrow}_{\rho} \mathit{scale}(e/-1, c')}$$

$$\vdots$$

## Theorem (Reduction semantics correctness)

(i) *If* $c \overset{\tau}{\Longrightarrow}_{\rho} c'$, *then*

   (a) $\mathcal{C}\left[\!\left[ c \right]\!\right]_{\rho}(0) = \tau$, *and*

   (b) $\mathcal{C}\left[\!\left[ c \right]\!\right]_{\rho}(i+1) = \mathcal{C}\left[\!\left[ c' \right]\!\right]_{\rho/1}(i) \quad$ *for all* $i \in \mathbb{N}$.

(ii) *If* $\mathcal{C}\left[\!\left[ c \right]\!\right]_{\rho}(0) = \tau$, *then there is a unique* $c'$ *with* $c \overset{\tau}{\Longrightarrow}_{\rho} c'$.

# Code Extraction

## Coq formalisation

- Denotational & reduction semantics
- Meta-theory of contracts (causality, monotonicity, ...)
- Definition of contract transformations and analyses
- Correctness proofs

# Code Extraction

## Coq formalisation

- Denotational & reduction semantics
- Meta-theory of contracts (causality, monotonicity, ...)
- Definition of contract transformations and analyses
- Correctness proofs

# Code Extraction

## Coq formalisation

- Denotational & reduction semantics
- Meta-theory of contracts (causality, monotonicity, ...)
- Definition of contract transformations and analyses
- Correctness proofs

## Extraction of executable Haskell code

- efficient Haskell implementation
- embedded domain-specific language for contracts
- contract analyses and contract management

# Future Work

- improve code extraction
- advanced analyses and transformations
  (e.g. scenario generation and “zooming”)
- combine this work with numerical methods

# Conclusion

The code is available from

`http://j.mp/contractDSL`

including

- full Coq proofs
- code extraction
- Prototype Haskell implementation
- example contracts
- technical report with all details