Time-indexed Types for Contracts

Patrick Bahr   Jost Berthold   Martin Elsman

DIKU
paba@diku.dk

17th July, 2015

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 (American 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 (American 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 (American 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!

Example: American Option

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: American Option

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

$$\begin{array}{l}\textbf{if } \mathit{obs}(X \text{ exercises option}) \textbf{ within } 90 \\ \textbf{then } 100 \times (\mathrm{USD}(Y \to X) \,\&\, r \times \mathrm{DKK}(X \to Y)) \\ \textbf{else } \emptyset\end{array}$$

Overview

- Denotational semantics based on cash-flows
- Type system $\rightsquigarrow$ causality
- Reduction semantics
- Contract specialisation
- Formalised in the Coq theorem prover
- Certified implementation via code extraction

An Overview of the Contract Language

$\emptyset$	empty contract with no obligations
$a(p_1 \to p_2)$	p_1 has to transfer one unit of a to p_2
$c_1 \,\&\, c_2$	conjunction of c_1 and c_2
$e \times c$	multiply all obligations in c by e
$d \uparrow c$	shift c into the future by d days
let $x = e$ **in** c	observe today's value of e at any time (via x)

if e **within** d **then** c_1 **else** c_2

- behave like c_1 as soon as e becomes true
- if e does not become true within d days behave like c_2

An Overview of the Contract Language

$\emptyset$	empty contract with no obligations
$a(p_1 \to p_2)$	p_1 has to transfer one unit of a to p_2
$c_1 \mathbin{\&} c_2$	conjunction of c_1 and c_2
$e \times c$	multiply all obligations in c by e
$d \uparrow c$	shift c into the future by d days
let $x = e$ **in** c	observe today's value of e at any time (via x)

if e **within** d **then** c_1 **else** c_2

- behave like c_1 as soon as e becomes true
- if e does not become true within d days behave like c_2

Expression Language

Real-valued and Boolean-valued expressions, extended by

$obs(l, d)$	observe the value of l at time d
$acc(f, d, e)$	accumulation over the last d days

Example: Asian Option

$$90 \uparrow \textbf{if } \mathit{obs}(X \text{ exercises option}) \textbf{ within } 0$$
$$\textbf{then } 100 \times (\mathrm{USD}(Y \to X) \,\&\, (\mathit{rate} \times \mathrm{DKK}(X \to Y)))$$
$$\textbf{else } \emptyset$$

where

$$\mathit{rate} = \frac{1}{30} \cdot \mathit{acc}(\lambda r.r + \mathit{obs}(\mathrm{FX}(\mathrm{USD}, \mathrm{DKK})), 30, 0)$$

Denotational Semantics

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

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

Denotational Semantics

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

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

$$\begin{aligned}\mathsf{CashFlow} &= \mathbb{N} \to \mathsf{Transactions}\\ \mathsf{Transactions} &= \mathsf{Party} \times \mathsf{Party} \times \mathsf{Asset} \to \mathbb{R}\end{aligned}$$

Denotational Semantics

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

$$\mathcal{C}\llbracket\cdot\rrbracket_{\cdot} : \mathsf{Contr} \times \mathsf{Env} \rightarrow \mathsf{CashFlow}$$
$$\mathsf{Env} = \mathsf{Label} \times \mathbb{Z} \rightarrow \mathbb{B} \cup \mathbb{R}$$

$$\mathsf{CashFlow} = \mathbb{N} \rightarrow \mathsf{Transactions}$$
$$\mathsf{Transactions} = \mathsf{Party} \times \mathsf{Party} \times \mathsf{Asset} \rightarrow \mathbb{R}$$

Denotational Semantics

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

$$\mathcal{C}\llbracket\cdot\rrbracket_{\cdot} : \mathsf{Contr} \times \mathsf{Env} \to \mathsf{CashFlow}$$
$$\mathsf{Env} = \mathsf{Label}_\alpha \times \mathbb{Z} \to \alpha$$

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

Contract Equivalences

$$e_1 \times (e_2 \times c) \simeq (e_1 \cdot e_2) \times c$$
$$d_1 \uparrow (d_2 \uparrow c) \simeq (d_1 + d_2) \uparrow c$$
$$d \uparrow (c_1 \,\&\, c_2) \simeq (d \uparrow c_1) \,\&\, (d \uparrow c_2)$$
$$e \times (c_1 \,\&\, c_2) \simeq (e \times c_1) \,\&\, (e \times c_2)$$
$$d \uparrow (e \times c) \simeq (d \Uparrow e) \times (d \uparrow c)$$

$$d \uparrow \emptyset \simeq \emptyset$$
$$r \times \emptyset \simeq \emptyset$$
$$0 \times c \simeq \emptyset$$
$$c \,\&\, \emptyset \simeq c$$
$$c_1 \,\&\, c_2 \simeq c_2 \,\&\, c_1$$

$$d \uparrow \textbf{if } b \textbf{ within } e \textbf{ then } c_1 \textbf{ else } c_2 \simeq \textbf{if } d \Uparrow b \textbf{ within } e \textbf{ then } d \uparrow c_1 \textbf{ else } d \uparrow c_2$$

$$(e_1 \times a(p_1 \to p_2)) \,\&\, (e_2 \times a(p_1 \to p_2)) \simeq (e_1 + e_2) \times a(p_1 \to p_2)$$

Causality

Definition

A closed contract c is causal iff

$$\rho_1 =_t \rho_2 \implies \mathcal{C}\left[\!\left[c \right]\!\right]_{\rho_1}(t) = \mathcal{C}\left[\!\left[c \right]\!\right]_{\rho_2}(t) \qquad \text{for all } t, \rho_1, \rho_2$$

Causality

Definition

A closed contract c is causal iff

$$\rho_1 =_t \rho_2 \implies \mathcal{C}\left[\!\left[c\right]\!\right]_{\rho_1}(t) = \mathcal{C}\left[\!\left[c\right]\!\right]_{\rho_2}(t) \qquad \text{for all } t, \rho_1, \rho_2$$

Example

$$\textbf{obs}(\mathrm{FX}(\mathrm{USD}, \mathrm{DKK}), 1) \times \mathrm{DKK}(X \to Y)$$

Type System – Expressions

$\boxed{\Gamma \Vdash e : \tau^t}$ where $t \in \mathbb{Z}_{-\infty}$

$$\frac{}{\Gamma \Vdash r : \mathrm{Real}^t} \qquad \frac{}{\Gamma \Vdash r : \mathrm{Bool}^t} \qquad \frac{l \in \mathrm{Label}_\tau \quad t \leq t'}{\Gamma \Vdash \mathbf{obs}(l, t) : \tau^{t'}}$$

$$\frac{x : \tau^t \in \Gamma \quad t \leq t'}{\Gamma \Vdash x : \tau^{t'}} \qquad \frac{\vdash op : \tau_1 \times \cdots \times \tau_n \to \tau \quad \Gamma \Vdash e_i : \tau_i^t}{\Gamma \Vdash op(e_1, \ldots, e_n) : \tau^t}$$

$$\frac{\Gamma, x : \tau^{-\infty} \Vdash e_1 : \tau^t \quad \Gamma^{+d} \Vdash e_2 : \tau^{t+d}}{\Gamma \Vdash \mathbf{acc}(\lambda x.\ e_1, d, e_2) : \tau^t}$$

Type System – Contracts

$\boxed{\Gamma \Vdash c : \mathsf{Contr}^t}$ where $t \in \mathbb{Z}_{-\infty}$

$$\frac{\Gamma^{-d} \Vdash c : \mathsf{Contr}^{t-d}}{\Gamma \Vdash d \uparrow c : \mathsf{Contr}^t} \qquad \frac{t \leq 0}{\Gamma \Vdash a(p \to q) : \mathsf{Contr}^t}$$

$$\frac{}{\Gamma \Vdash \emptyset : \mathsf{Contr}^t} \qquad \frac{\Gamma \Vdash e : \mathsf{Real}^{t'} \quad \Gamma \Vdash c : \mathsf{Contr}^{t'} \quad t \leq t'}{\Gamma \Vdash e \times c : \mathsf{Contr}^t}$$

$$\frac{\Gamma \Vdash c_i : \mathsf{Contr}^t}{\Gamma \Vdash c_1 \,\&\, c_2 : \mathsf{Contr}^t} \qquad \frac{\Gamma \Vdash e : \tau^s \quad \Gamma, x : \tau^s \Vdash c : \mathsf{Contr}^t}{\Gamma \Vdash \textbf{let } x = e \textbf{ in } c : \mathsf{Contr}^t}$$

$$\frac{\Gamma \Vdash e : \mathsf{Bool}^0 \quad \Gamma \Vdash c_1 : \mathsf{Contr}^t \quad \Gamma^{-d} \Vdash c_2 : \mathsf{Contr}^{t-d}}{\Gamma \Vdash \textbf{if } e \textbf{ within } d \textbf{ then } c_1 \textbf{ else } c_2 : \mathsf{Contr}^t}$$

Type System – Properties

Theorem

If $\Vdash c : \mathrm{Contr}^t$*, then* c *is causal.*

Type System – Properties

Theorem

If $\Vdash c : \mathrm{Contr}^t$*, then* c *is causal.*

Lemma

(i) *If* $\Gamma \Vdash e : \tau^t$*, then* $\Gamma \Vdash e : \tau^s$ *for all* $s \geq t$*.*

(ii) *If* $\Gamma \Vdash c : \mathrm{Contr}^t$*, then* $\Gamma \Vdash c : \mathrm{Contr}^s$ *for all* $s \leq t$*.*

Type System – Properties

Theorem

If $\Vdash c : \mathrm{Contr}^t$, *then* c *is causal.*

Lemma

(i) *If* $\Gamma \Vdash e : \tau^t$, *then* $\Gamma \Vdash e : \tau^s$ *for all* $s \geq t$.

(ii) *If* $\Gamma \Vdash c : \mathrm{Contr}^t$, *then* $\Gamma \Vdash c : \mathrm{Contr}^s$ *for all* $s \leq t$.

Theorem (Type inference is sound and complete)

(i) *If* $\Gamma \mapsto c : \mathrm{Contr}^t$, *then* $\Gamma \Vdash c : \mathrm{Contr}^s$ *for all* $s \leq t$.

(ii) *If* $\Gamma \Vdash c : \mathrm{Contr}^s$, *then* $\Gamma \mapsto c : \mathrm{Contr}^t$ *for a unique* $t \geq s$.

Reduction Semantics

$$c \xrightarrow{T}_{\rho} c'$$

Reduction Semantics

$$c \xrightarrow{T}_{\rho} c'$$

Theorem (Computational adequacy of $\xrightarrow{T}_{\rho}$)

Let $\Vdash c : \mathrm{Contr}^t$ *and* $\rho \in \mathrm{Env}_{\mathrm{P}}$*.*

(i) *If* $c \xrightarrow{T}_{\rho} c'$*, then the following holds for all* ρ' *that extend* ρ*:*

   (a) $\mathcal{C}\left[\!\left[c\right]\!\right]_{\rho'}(0) = T$*, and*
   (b) $\mathcal{C}\left[\!\left[c\right]\!\right]_{\rho'}(i+1) = \mathcal{C}\left[\!\left[c'\right]\!\right]_{\rho'/1}(i)$ *for all* $i \in \mathbb{N}$*,*

(ii) *If* $c \xrightarrow{T}_{\rho} c'$*, then* $\Vdash c' : \mathrm{Contr}^{t-1}$*.*

(iii) *If* ρ *is historically complete, then there is a unique* c' *such that* $c \xrightarrow{T}_{\rho} c'$ *and* $T = \mathcal{C}\left[\!\left[c\right]\!\right]_{\rho}(0)$*.*

Code Extraction

Coq formalisation

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

Code Extraction

Coq formalisation

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

Code Extraction

Coq formalisation

- Denotational & reduction semantics
- Meta-theory of contracts (causality, type system, ...)
- 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

Contracts in Haskell – Example

```
{-# LANGUAGE RebindableSyntax #-}

import RebindableEDSL

american :: Contr
american = if bObs (Decision X "exercise") 0 `within` 90
           then 100 # (transfer Y X USD &
                          (6.23 # transfer X Y DKK))
           else zero

asian :: Contr
asian = 90 ! if bObs (Decision X "exercise") 0
             then 100 # (transfer Y X USD &
                            (rate # transfer X Y DKK))
             else zero
    where rate = (acc (\r -> r +
                        rObs (FX USD DKK) 0) 30 0) / 30
```