Blame Assignment for Higher-Order Contracts with Intersection and Union

UNI FREIBURG

Albert-Ludwigs-Universität Freiburg

Matthias Keil Peter Thiemann

University of Freiburg

23 Mar 2015

Flat Contracts

UNI FREIBURG

Examples

$$Pos = \{x \mid x > 0\}$$

$$Even = \{x \mid x\%2 = 0\}$$

Flat Contracts

UNI FREIBURG

Examples

$$Pos = \{x \mid x > 0\}$$

$$Even = \{x \mid x\%2 = 0\}$$

Assertion

- $M@C$: assert contract C to term M

Flat Contracts

UNI FREIBURG

Examples

$$Pos = \{x \mid x > 0\}$$

$$Even = \{x \mid x\%2 = 0\}$$

Assertion

- $M@C$: assert contract C to term M
- $10@Pos \longrightarrow 10$

Flat Contracts

Examples

$$Pos = \{x \mid x > 0\}$$

$$Even = \{x \mid x\%2 = 0\}$$

Assertion

- $M@C$: assert contract C to term M
- $10@Pos \longrightarrow 10$
- $10@Even \longrightarrow 10$

Flat Contracts

Examples

$$Pos = \{x \mid x > 0\}$$

$$Even = \{x \mid x\%2 = 0\}$$

Assertion

- $M@C$: assert contract C to term M
- $10@Pos \longrightarrow 10$
- $10@Even \longrightarrow 10$
- $(2-4)@Pos$ contract violation (by subject -2)

Flat Contracts

Examples

$$Pos = \{x \mid x > 0\}$$

$$Even = \{x \mid x\%2 = 0\}$$

Assertion

- $M@C$: assert contract C to term M
- $10@Pos \longrightarrow 10$
- $10@Even \longrightarrow 10$
- $(2-4)@Pos$ contract violation (by subject -2)
- $(2+3)@Even$ contract violation (by subject 5)

Higher-Order Contracts [Findler, Felleisen 2002]

UNI FREIBURG

Examples

$$Pos \to Pos \qquad Even \to Even \qquad Pos \to (Even \to Even)$$

Higher-Order Contracts [Findler, Felleisen 2002]

UNI FREIBURG

Examples

$$Pos \to Pos \qquad Even \to Even \qquad Pos \to (Even \to Even)$$

Assertion (first-order function)

- Let $f = \lambda x.x - 10$

Higher-Order Contracts [Findler, Felleisen 2002]

UNI FREIBURG

Examples

$$Pos \to Pos \qquad Even \to Even \qquad Pos \to (Even \to Even)$$

Assertion (first-order function)

- Let $f = \lambda x.x - 10$
- $(f @ (Pos \to Pos))\, 100 \longrightarrow 90$

Higher-Order Contracts [Findler, Felleisen 2002]

UNI FREIBURG

Examples

$$Pos \to Pos \qquad Even \to Even \qquad Pos \to (Even \to Even)$$

Assertion (first-order function)

- Let $f = \lambda x.x - 10$
- $(f\,@(Pos \to Pos))\,100 \longrightarrow 90$
- $(f\,@(Pos \to Pos))\,10$ blame subject f

Higher-Order Contracts [Findler, Felleisen 2002]

UNI FREIBURG

Examples

$$Pos \to Pos \qquad Even \to Even \qquad Pos \to (Even \to Even)$$

Assertion (first-order function)

- Let $f = \lambda x.x - 10$
- $(f\,@(Pos \to Pos))\,100 \longrightarrow 90$
- $(f\,@(Pos \to Pos))\,10$ blame subject f
- $(f\,@(\mathit{Pos} \to Pos))\,0$ blame context $\square\,0$

Higher-Order Contracts II

UNI FREIBURG

Assertion

- Let $add = \lambda x.\lambda y.x + y$ and $C = Pos \to (Even \to Even)$

Higher-Order Contracts II

UNI FREIBURG

Assertion

- Let $add = \lambda x.\lambda y.x + y$ and $C = Pos \to (Even \to Even)$
- $(add@C)\,0$ blame context $\square\,0$

Higher-Order Contracts II

UNI FREIBURG

Assertion

- Let $add = \lambda x.\lambda y.x + y$ and $C = Pos \to (Even \to Even)$
- $(add@C)\,0$ blame context $\square\,0$
- $((add@C)\,1)\,1$ blame context $(\square\,1)\,1$

Higher-Order Contracts II

UNI FREIBURG

Assertion

- Let $add = \lambda x.\lambda y.x + y$ and $C = Pos \to (Even \to Even)$
- $(add @ C)\,0$ blame context $\square\,0$
- $((add @ C)\,1)\,1$ blame context $(\square\,1)\,1$
- $((add @ C)\,1)\,0$ blame subject *add*

Contracts vs. Types

UNI FREIBURG

Contracts ≈ dynamically checked types

- flat contracts ≈ subset types
- function contracts ≈ function types

Contracts vs. Types

UNI FREIBURG

Contracts ≈ dynamically checked types

- flat contracts ≈ subset types
- function contracts ≈ function types

Contract work driven by types . . .

- pair contracts [Hinze, Löh]
- sum contracts
- polymorphic contracts [Ahmed, Findler, Guha, Krishnamurthi, Matthews, Wadler]

This Work

UNI FREIBURG

This Work

UNI FREIBURG

What about intersection and union types?

UNI FREIBURG

What about intersection and union types?

Intersection types

- modeling overloading
- multiple inheritance

UNI FREIBURG

What about intersection and union types?

Intersection types

- modeling overloading
- multiple inheritance

Union types

- dual of intersection type
- domain of overloaded function
- XML typing and dynamic typing

Intersection Types

Intuition of intersection type

- If a term has both type S and T, then it also possesses the intersection type $S \cap T$.
- A context for $M : S \cap T$ can choose to treat M as S or T.

Introduction and elimination for intersection [Pierce 1991]

$$\textsc{Inter-I}\quad \frac{A \vdash V : S \qquad A \vdash V : T}{A \vdash V : S \cap T}$$

$$\textsc{Sub-Inter-L}\quad S \cap T <: S \qquad\qquad \textsc{Sub-Inter-R}\quad S \cap T <: T$$

Intersection for Overloading

UNI FREIBURG

An overloaded + operator

- $+ : \mathit{Num} \times \mathit{Num} \to \mathit{Num}$
- $+ : \mathit{Str} \times \mathit{Str} \to \mathit{Str}$

Intersection for Overloading

An overloaded + operator

- $+ : Num \times Num \to Num$
- $+ : Str \times Str \to Str$
- hence $+ : (Num \times Num \to Num) \cap (Str \times Str \to Str)$

If we had intersection contracts . . .

Let $p = +@(Num \times Num \to Num) \cap (Str \times Str \to Str)$

Intersection for Overloading

An overloaded + operator

- $+ : Num \times Num \to Num$
- $+ : Str \times Str \to Str$
- hence $+ : (Num \times Num \to Num) \cap (Str \times Str \to Str)$

If we had intersection contracts . . .

Let $p = +@(Num \times Num \to Num) \cap (Str \times Str \to Str)$

- $p(17, 4) \longrightarrow 21$

Intersection for Overloading

An overloaded + operator

- $+ : Num \times Num \to Num$
- $+ : Str \times Str \to Str$
- hence $+ : (Num \times Num \to Num) \cap (Str \times Str \to Str)$

If we had intersection contracts . . .

Let $p = +@(Num \times Num \to Num) \cap (Str \times Str \to Str)$

- $p(17, 4) \longrightarrow 21$
- $p(\text{"}foo\text{"}, \text{"}bar\text{"}) \longrightarrow \text{"}foobar\text{"}$

Intersection for Overloading

An overloaded + operator

- $+ : Num \times Num \to Num$
- $+ : Str \times Str \to Str$
- hence $+ : (Num \times Num \to Num) \cap (Str \times Str \to Str)$

If we had intersection contracts . . .

Let $p = +@(Num \times Num \to Num) \cap (Str \times Str \to Str)$

- $p(17, 4) \longrightarrow 21$
- $p(\text{"}foo\text{"}, \text{"}bar\text{"}) \longrightarrow \text{"}foobar\text{"}$
- $p(17, \text{"}bar\text{"})$ blame context $\square\,(17, \text{"}bar\text{"})$

Intersection for Overloading

An overloaded + operator

- $+ : Num \times Num \to Num$
- $+ : Str \times Str \to Str$
- hence $+ : (Num \times Num \to Num) \cap (Str \times Str \to Str)$

If we had intersection contracts . . .

Let $p = +@(Num \times Num \to Num) \cap (Str \times Str \to Str)$

- $p(17, 4) \longrightarrow 21$
- $p(\text{"}foo\text{"}, \text{"}bar\text{"}) \longrightarrow \text{"}foobar\text{"}$
- $p(17, \text{"}bar\text{"})$ blame context $\square\,(17, \text{"}bar\text{"})$

No subject blame because + fulfills the intersection contract

Blaming Intersection

Let $C = (Pos \to Pos) \cap (Even \to Even)$
Let $f = \lambda x.\texttt{if}\ x > 2\ \texttt{then}\ x - 10\ \texttt{else}\ x + 1$

Blaming Intersection

UNI FREIBURG

Let $C = (\mathit{Pos} \to \mathit{Pos}) \cap (\mathit{Even} \to \mathit{Even})$
Let $f = \lambda x.\texttt{if}\ x > 2\ \texttt{then}\ x - 10\ \texttt{else}\ x + 1$

- $(f\,@\,C)\,(-1)$ blame context

Blaming Intersection

Let $C = (\mathit{Pos} \to \mathit{Pos}) \cap (\mathit{Even} \to \mathit{Even})$
Let $f = \lambda x.\texttt{if}\ x > 2\ \texttt{then}\ x - 10\ \texttt{else}\ x + 1$

- $(f\,@\,C)\,(-1)$ blame context
- $(f\,@\,C)\,0$ blame subject

Blaming Intersection

Let $C = (\mathit{Pos} \to \mathit{Pos}) \cap (\mathit{Even} \to \mathit{Even})$
Let $f = \lambda x.\texttt{if}\ x > 2\ \texttt{then}\ x - 10\ \texttt{else}\ x + 1$

- $(f \texttt{@} C)\,(-1)$ blame context
- $(f \texttt{@} C)\,0$ blame subject
- $(f \texttt{@} C)\,1 \longrightarrow 2$

Blaming Intersection

Let $C = (\mathit{Pos} \to \mathit{Pos}) \cap (\mathit{Even} \to \mathit{Even})$
Let $f = \lambda x.\texttt{if}\ x > 2\ \texttt{then}\ x - 10\ \texttt{else}\ x + 1$

- $(f \texttt{@} C)\,(-1)$ blame context
- $(f \texttt{@} C)\,0$ blame subject
- $(f \texttt{@} C)\,1 \longrightarrow 2$
- $(f \texttt{@} C)\,2$ blame subject

Blaming Intersection

Let $C = (\mathit{Pos} \to \mathit{Pos}) \cap (\mathit{Even} \to \mathit{Even})$
Let $f = \lambda x.\texttt{if}\ x > 2\ \texttt{then}\ x - 10\ \texttt{else}\ x + 1$

- $(f\,@\,C)\,(-1)$ blame context
- $(f\,@\,C)\,0$ blame subject
- $(f\,@\,C)\,1 \longrightarrow 2$
- $(f\,@\,C)\,2$ blame subject
- $(f\,@\,C)\,3$ blame subject

Blaming Intersection

Let $C = (\mathit{Pos} \to \mathit{Pos}) \cap (\mathit{Even} \to \mathit{Even})$
Let $f = \lambda x.\texttt{if}\ x > 2\ \texttt{then}\ x - 10\ \texttt{else}\ x + 1$

- $(f \texttt{@} C)\,(-1)$ blame context
- $(f \texttt{@} C)\,0$ blame subject
- $(f \texttt{@} C)\,1 \longrightarrow 2$
- $(f \texttt{@} C)\,2$ blame subject
- $(f \texttt{@} C)\,3$ blame subject
- $(f \texttt{@} C)\,4$ blame subject

Blaming Rules for Intersection

Subject blame

$\mathcal{L}[M@(C \cap D)]$ blames the subject iff
$\mathcal{L}[M@C]$ blames the subject **or** $\mathcal{L}[M@D]$ blames the subject.

$$\frac{A \vdash V : S \qquad A \vdash V : T}{A \vdash V : S \cap T}\ \textsc{Inter-I}$$

Blaming Rules for Intersection

Subject blame

$\mathcal{L}[M@(C \cap D)]$ blames the subject iff
$\mathcal{L}[M@C]$ blames the subject **or** $\mathcal{L}[M@D]$ blames the subject.

$$\frac{A \vdash V : S \qquad A \vdash V : T}{A \vdash V : S \cap T}\ \textsc{Inter-I}$$

Context blame

$\mathcal{F}[M@(C \cap D)]$ blames the context iff
$\mathcal{F}[M@C]$ blames the context **and** $\mathcal{F}[M@D]$ blames the context.

$\mathcal{F} ::= E[\Box\, V] \mid \ldots$ an **elimination context**

$$S \cap T <: S\ \textsc{Sub-Inter-L} \qquad S \cap T <: T\ \textsc{Sub-Inter-R}$$

Union Types

UNI FREIBURG

Union Types

UNI FREIBURG

Intuition of union type

- If a term has type S **or** T, then it also possesses the union type $S \cup T$.
- A context for $M : S \cup T$ must be able to deal with S **and** T.

Union Types

Intuition of union type

- If a term has type S **or** T, then it also possesses the union type $S \cup T$.
- A context for $M : S \cup T$ must be able to deal with S **and** T.

Introduction and elimination for union [Pierce 1991]

$$\textsc{Union-E}\quad \frac{A \vdash M : S \cup T \qquad A, x : S \vdash N : R \qquad A, x : T \vdash N : R}{A \vdash \mathit{let}\, x = M\, \mathit{in}\, N : R}$$

$$\textsc{Sub-Union-L}\quad S <: S \cup T \qquad\qquad \textsc{Sub-Union-R}\quad T <: S \cup T$$

Blaming Rules for Union

UNI FREIBURG

Dualize rules for intersection . . .

Blaming Rules for Union

UNI FREIBURG

Dualize rules for intersection . . .

Subject blame

$E[M@(C \cup D)]$ blames the subject iff
$E[M@C]$ blames the subject **and** $E[M@D]$ blames the subject.
Viz. introduction rule.

Blaming Rules for Union

UNI FREIBURG

Dualize rules for intersection ...

Subject blame

$E[M@(C \cup D)]$ blames the subject iff
$E[M@C]$ blames the subject **and** $E[M@D]$ blames the subject.
Viz. introduction rule.

Context blame

$\mathcal{L}[M@(C \cup D)]$ blames the context iff
$\mathcal{L}[M@C]$ blames the context **or** $\mathcal{L}[M@D]$ blames the context.
Viz. elimination rules.

Blaming Union

UNI FREIBURG

Let $C = (Pos \to Pos) \cup (Even \to Even)$
Let $g = \lambda x.x - 4$ (fulfills C because it fulfills $Even \to Even$)
$\Rightarrow$ no subject blame arises!

Blaming Union

UNI FREIBURG

Let $C = (\mathit{Pos} \to \mathit{Pos}) \cup (\mathit{Even} \to \mathit{Even})$
Let $g = \lambda x.x - 4$ (fulfills C because it fulfills $\mathit{Even} \to \mathit{Even}$)
$\Rightarrow$ no subject blame arises!

- $(g @ C)\, 0$ blame context

Blaming Union

UNI FREIBURG

Let $C = (\mathit{Pos} \to \mathit{Pos}) \cup (\mathit{Even} \to \mathit{Even})$
Let $g = \lambda x.x - 4$ (fulfills C because it fulfills $\mathit{Even} \to \mathit{Even}$)
$\Rightarrow$ no subject blame arises!

- $(g @ C)\,0$ blame context
- $(g @ C)\,1$ blame context

Blaming Union

Let $C = (\mathit{Pos} \to \mathit{Pos}) \cup (\mathit{Even} \to \mathit{Even})$
Let $g = \lambda x.x - 4$ (fulfills C because it fulfills $\mathit{Even} \to \mathit{Even}$)
$\Rightarrow$ no subject blame arises!

- $(g \mathbin{@} C)\, 0$ blame context
- $(g \mathbin{@} C)\, 1$ blame context
- $(g \mathbin{@} C)\, (-1)$ blame context

Blaming Union

Let $C = (\mathit{Pos} \to \mathit{Pos}) \cup (\mathit{Even} \to \mathit{Even})$
Let $g = \lambda x.x - 4$ (fulfills C because it fulfills $\mathit{Even} \to \mathit{Even}$)
$\Rightarrow$ no subject blame arises!

- $(g \mathbin{@} C)\, 0$ blame context
- $(g \mathbin{@} C)\, 1$ blame context
- $(g \mathbin{@} C)\, (-1)$ blame context
- $(g \mathbin{@} C)\, 2$ $\longrightarrow -2$ because of $\mathit{Even} \to \mathit{Even}$

But there is a further twist ...

Blaming Union II

UNI FREIBURG

Let $C = (Pos \to Pos) \cup (Even \to Even)$
Let $h = \lambda x.\texttt{if } x\%3 = 0 \texttt{ then } -x \texttt{ else } x + 1$

Blaming Union II

UNI FREIBURG

Let $C = (Pos \to Pos) \cup (Even \to Even)$
Let $h = \lambda x.\texttt{if } x\%3 = 0 \texttt{ then } -x \texttt{ else } x + 1$

Observation: h does not fulfill C

- h is not $Even \to Even$ because $h(2) = 3$
- h is not $Pos \to Pos$ because $h(3) = -3$

Blaming Union II

Let $C = (\mathit{Pos} \to \mathit{Pos}) \cup (\mathit{Even} \to \mathit{Even})$
Let $h = \lambda x.\texttt{if } x\%3 = 0 \texttt{ then } -x \texttt{ else } x + 1$

Observation: h does not fulfill C

- h is not $\mathit{Even} \to \mathit{Even}$ because $h(2) = 3$
- h is not $\mathit{Pos} \to \mathit{Pos}$ because $h(3) = -3$

Blaming

- $(h@C)\,2 \longrightarrow 3$ because of $\mathit{Pos} \to \mathit{Pos}$

Blaming Union II

Let $C = (\mathit{Pos} \to \mathit{Pos}) \cup (\mathit{Even} \to \mathit{Even})$
Let $h = \lambda x.\texttt{if } x\%3 = 0 \texttt{ then } -x \texttt{ else } x + 1$

Observation: h does not fulfill C

- h is not $\mathit{Even} \to \mathit{Even}$ because $h(2) = 3$
- h is not $\mathit{Pos} \to \mathit{Pos}$ because $h(3) = -3$

Blaming

- $(h@C)\,2 \longrightarrow 3$ because of $\mathit{Pos} \to \mathit{Pos}$
- $(h@C)\,3$ blame context: Even violated

Blaming Union II

Let $C = (Pos \to Pos) \cup (Even \to Even)$
Let $h = \lambda x.\texttt{if } x\%3 = 0 \texttt{ then } -x \texttt{ else } x + 1$

Observation: h does not fulfill C

- h is not $Even \to Even$ because $h(2) = 3$
- h is not $Pos \to Pos$ because $h(3) = -3$

Blaming

- $(h@C)\,2 \longrightarrow 3$ because of $Pos \to Pos$
- $(h@C)\,3$ blame context: $Even$ violated
- $(h@C)\,0$ blame context: Pos violated

Blaming Union II

Let $C = (\mathit{Pos} \to \mathit{Pos}) \cup (\mathit{Even} \to \mathit{Even})$
Let $h = \lambda x.\texttt{if } x\%3 = 0 \texttt{ then } -x \texttt{ else } x + 1$

Observation: h does not fulfill C

- h is not $\mathit{Even} \to \mathit{Even}$ because $h(2) = 3$
- h is not $\mathit{Pos} \to \mathit{Pos}$ because $h(3) = -3$

Blaming

- $(h@C)\,2 \longrightarrow 3$ because of $\mathit{Pos} \to \mathit{Pos}$
- $(h@C)\,3$ blame context: Even violated
- $(h@C)\,0$ blame context: Pos violated
- $(h@C)\,6 \longrightarrow -6$ because of $\mathit{Even} \to \mathit{Even}$

Blaming Union II

UNI FREIBURG

Let $C = (Pos \to Pos) \cup (Even \to Even)$
Let $h = \lambda x.\texttt{if } x\%3 = 0 \texttt{ then } -x \texttt{ else } x + 1$

Observation: h does not fulfill C

- h is not $Even \to Even$ because $h(2) = 3$
- h is not $Pos \to Pos$ because $h(3) = -3$

Blaming

- $(h@C)\,2 \longrightarrow 3$ because of $Pos \to Pos$
- $(h@C)\,3$ blame context: $Even$ violated
- $(h@C)\,0$ blame context: Pos violated
- $(h@C)\,6 \longrightarrow -6$ because of $Even \to Even$

Problem: no single test can detect the violation

Blaming Union II

Let $C = (Pos \to Pos) \cup (Even \to Even)$
Let $h = \lambda x.\texttt{if } x\%3 = 0 \texttt{ then } -x \texttt{ else } x + 1$

Observation: h does not fulfill C

- h is not $Even \to Even$ because $h(2) = 3$
- h is not $Pos \to Pos$ because $h(3) = -3$

Blaming

- $(h@C)\,2 \longrightarrow 3$ because of $Pos \to Pos$
- $(h@C)\,3$ blame context: $Even$ violated
- $(h@C)\,0$ blame context: Pos violated
- $(h@C)\,6 \longrightarrow -6$ because of $Even \to Even$

$\Rightarrow$ Union must stick to one alternative

A Denotational Model of Contracts

Denotational Model of Contracts

UNI FREIBURG

Intuition

The semantics of a contract C comprises

1. a set of terms $[\![C]\!]^+$ that fulfill the contract and
2. a set of contexts $[\![C]\!]^-$ that respect the contract.

Denotational Model

Flat contracts

1. $[\![\{x \mid P\}]\!]^+ = \{M \mid (\lambda x.P)\, M \not\longrightarrow^* \mathit{false}\}$
2. $[\![\{x \mid P\}]\!]^- = \{\mathcal{L} \mid \mathcal{L} \text{ is a context}\}$

Denotational Model

Flat contracts

1. $[\![\{x \mid P\}]\!]^+ = \{M \mid (\lambda x.P)\, M \not\longrightarrow^* \mathit{false}\}$
2. $[\![\{x \mid P\}]\!]^- = \{\mathcal{L} \mid \mathcal{L} \text{ is a context}\}$

Example

$$
\begin{array}{rcl}
[\![\mathit{Pos}]\!]^+ & = & \{M \mid M > 0 \not\longrightarrow^* \mathit{false}\} \\
& = & \{1, 2, 3, \ldots\} \cup \\
& & \{M \mid M \text{ diverges}\} \cup \\
& & \{M \mid (M > 0) \text{ gets stuck}\}
\end{array}
$$

Denotational Model

Function contracts

1. $[\![C \to D]\!]^+ = \{M \mid \forall N \in [\![C]\!]^+ . M\,N \in [\![D]\!]^+ \\ \qquad \wedge \forall \mathcal{N} \in [\![D]\!]^- . \mathcal{N}[M\,\square] \in [\![C]\!]^-\}$
2. $[\![C \to D]\!]^- = \ldots$

Denotational Model

Function contracts

1. $[\![C \to D]\!]^+ = \{M \mid \forall N \in [\![C]\!]^+ . M\, N \in [\![D]\!]^+ \\ \qquad \wedge \forall \mathcal{N} \in [\![D]\!]^- . \mathcal{N}[M\,\square] \in [\![C]\!]^-\}$
2. $[\![C \to D]\!]^- = \ldots$

- Standard function interpretation

Denotational Model

Function contracts

1. $[\![C \to D]\!]^+ = \{M \mid \forall N \in [\![C]\!]^+ . M\,N \in [\![D]\!]^+ \\ \qquad \wedge \forall \mathcal{N} \in [\![D]\!]^- . \mathcal{N}[M\,\square] \in [\![C]\!]^-\}$
2. $[\![C \to D]\!]^- = \ldots$

- Standard function interpretation
- NEW M acts on contexts by transforming a context that respects D into a context that respects C

Denotational Model

Function contracts

1. $[\![C \to D]\!]^+ = \{M \mid \forall N \in [\![C]\!]^+ . M\,N \in [\![D]\!]^+ \\ \qquad \wedge \forall \mathcal{N} \in [\![D]\!]^- . \mathcal{N}[M\,\square] \in [\![C]\!]^-\}$
2. $[\![C \to D]\!]^- = \ldots$

- Standard function interpretation
- NEW M acts on contexts by transforming a context that respects D into a context that respects C
- Set of contexts that respect $C \to D$
  - promise to only pass argument that fulfill C
  - promise to put result in context respecting D

Denotational Model

Function contracts

1. $[\![C \to D]\!]^+ = \{M \mid \forall N \in [\![C]\!]^+ . M\,N \in [\![D]\!]^+ \\ \qquad \wedge \forall \mathcal{N} \in [\![D]\!]^- . \mathcal{N}[M\,\square] \in [\![C]\!]^-\}$
2. $[\![C \to D]\!]^- = \ldots$

- Standard function interpretation
- NEW M acts on contexts by transforming a context that respects D into a context that respects C
- Set of contexts that respect $C \to D$
  - promise to only pass argument that fulfill C
  - promise to put result in context respecting D
- Defined by coinductive inference rules

Contexts for Function Contracts

UNI FREIBURG

Selected rules for $P = [\![C \to D]\!]^-$

$$\textsc{P-Apply}\quad \frac{N \in [\![C]\!]^+ \qquad E \in [\![D]\!]^-}{E[\square\, N] \in P}$$

Contexts for Function Contracts

UNI FREIBURG

Selected rules for $P = [\![C \to D]\!]^-$

$$\textsc{P-Apply}\quad \frac{N \in [\![C]\!]^+ \qquad E \in [\![D]\!]^-}{E[\Box\, N] \in P}$$

$$\textsc{P-Reduce}\quad \frac{\mathcal{N} \in P \qquad \mathcal{M} \longrightarrow \mathcal{N}}{\mathcal{M} \in P}$$

Contexts for Function Contracts

Selected rules for $P = [\![C \to D]\!]^-$

$$\textsc{P-Apply}\quad \frac{N \in [\![C]\!]^+ \qquad E \in [\![D]\!]^-}{E[\Box\, N] \in P}$$

$$\textsc{P-Reduce}\quad \frac{\mathcal{N} \in P \qquad \mathcal{M} \longrightarrow \mathcal{N}}{\mathcal{M} \in P}$$

$\mathcal{M} \longrightarrow \mathcal{N}$ is context reduction

Contexts for Function Contracts

Selected rules for $P = [\![C \to D]\!]^-$

$$\textsc{P-Apply}\quad \frac{N \in [\![C]\!]^+ \qquad E \in [\![D]\!]^-}{E[\Box\, N] \in P}$$

$$\textsc{P-Reduce}\quad \frac{\mathcal{N} \in P \qquad \mathcal{M} \longrightarrow \mathcal{N}}{\mathcal{M} \in P}$$

$$\textsc{P-Irred}\quad \frac{\mathcal{M} \not\longrightarrow \qquad \mathcal{M} \notin \{E[(\lambda x.N)\,\Box], E[\Box\, N]\}}{\mathcal{M} \in P}$$

Contexts for Function Contracts II

Beta redexes involving holes

$$\textsc{P-Delete}\ \frac{x \notin \mathit{free}(M)}{E[(\lambda x.M)\,\square] \in P}$$

Contexts for Function Contracts II

Beta redexes involving holes

$$\textsc{P-Delete}\quad \frac{x \notin \mathit{free}(M)}{E[(\lambda x.M)\,\square] \in P}$$

$$\textsc{P-Linear}\quad \frac{M = \mathcal{F}[x] \qquad E[\mathcal{F}] \in P}{E[(\lambda x.M)\,\square] \in P}$$

Contexts for Function Contracts II

Beta redexes involving holes

$$\textsc{P-Delete}\quad \frac{x \notin \mathit{free}(M)}{E[(\lambda x.M)\,\Box] \in P}$$

$$\textsc{P-Linear}\quad \frac{M = \mathcal{F}[x] \qquad E[\mathcal{F}] \in P}{E[(\lambda x.M)\,\Box] \in P}$$

$$\textsc{P-Expand}\quad \frac{\forall \mathcal{F}, V.\, \lambda x.M = \lambda x.\mathcal{F}[x] \Rightarrow E[\mathcal{F}\{x \mapsto V\}] \in P}{E[(\lambda x.M)\,\Box] \in P}$$

Contexts for Function Contracts II

Beta redexes involving holes

$$\textsc{P-Delete}\quad \frac{x \notin \mathit{free}(M)}{E[(\lambda x.M)\,\square] \in P} \qquad \textsc{P-Linear}\quad \frac{M = \mathcal{F}[x] \qquad E[\mathcal{F}] \in P}{E[(\lambda x.M)\,\square] \in P}$$

$$\textsc{P-Expand}\quad \frac{\forall \mathcal{F}, V.\, \lambda x.M = \lambda x.\mathcal{F}[x] \Rightarrow E[\mathcal{F}\{x \mapsto V\}] \in P}{E[(\lambda x.M)\,\square] \in P}$$

- P-Delete and P-Linear are special cases of P-Expand

Semantics of Intersection and Union Contracts

UNI FREIBURG

Intersection

1. $[\![C \cap D]\!]^+ = [\![C]\!]^+ \cap [\![D]\!]^+$
2. $[\![C \cap D]\!]^- = [\![C]\!]^- \cup [\![D]\!]^- \cup \ldots$
   (defined by closing under P-Expand etc)

Semantics of Intersection and Union Contracts

UNI FREIBURG

Intersection

1. $[\![C \cap D]\!]^+ = [\![C]\!]^+ \cap [\![D]\!]^+$
2. $[\![C \cap D]\!]^- = [\![C]\!]^- \cup [\![D]\!]^- \cup \ldots$
   (defined by closing under P-Expand etc)

Union

1. $[\![C \cup D]\!]^+ = [\![C]\!]^+ \cup [\![D]\!]^+$
2. $[\![C \cup D]\!]^- = [\![C]\!]^- \cap [\![D]\!]^-$

Semantics of Intersection and Union Contracts

UNI FREIBURG

Intersection

1. $[\![C \cap D]\!]^+ = [\![C]\!]^+ \cap [\![D]\!]^+$
2. $[\![C \cap D]\!]^- = [\![C]\!]^- \cup [\![D]\!]^- \cup \ldots$
   (defined by closing under P-Expand etc)

Union

1. $[\![C \cup D]\!]^+ = [\![C]\!]^+ \cup [\![D]\!]^+$
2. $[\![C \cup D]\!]^- = [\![C]\!]^- \cap [\![D]\!]^-$

- Cf. blaming rules and typing rules

Provable from Denotational Model

UNI FREIBURG

Intersection for flat contracts

$$\{x \mid P\} \cap \{x \mid Q\} = \{x \mid P \wedge Q\}$$

Union for flat contracts

$$\{x \mid P\} \cup \{x \mid Q\} = \{x \mid P \vee Q\}$$

- proof: simple calculation
- only subject blame
- context blame does not arise

Contract Monitoring

Contract Monitoring

UNI FREIBURG

Challenges

- Small-step operational semantics
  - nondeterministic specification
  - deterministic implementation (with simulation result)
- Gathering blame for intersection and union
- Gathering blame across different uses of same union

Operational Semantics

Reduction relation

$$\varrho, M \mapsto \varsigma, N$$

- M, N terms

Operational Semantics

Reduction relation

$$\varrho, M \mapsto \varsigma, N$$

- M, N terms
- ϱ, ς lists of constraints in order of generation

Operational Semantics

Reduction relation

$$\varrho, M \mapsto \varsigma, N$$

- M, N terms
- ϱ, ς lists of constraints in order of generation
- one constraint for each contract operator $\rightarrow$, $\cup$, $\cap$

Operational Semantics

Reduction relation

$$\varrho, M \mapsto \varsigma, N$$

- M, N terms
- ϱ, ς lists of constraints in order of generation
- one constraint for each contract operator $\rightarrow$, $\cup$, $\cap$
- one constraint for each evaluated flat contract

Operational Semantics

UNI FREIBURG

Reduction relation

$$\varrho, M \mapsto \varsigma, N$$

- M, N terms
- ϱ, ς lists of constraints in order of generation
- one constraint for each contract operator $\rightarrow$, $\cup$, $\cap$
- one constraint for each evaluated flat contract
- cannot blame immediately: flat contract may be nested in intersection or union

Operational Semantics

Reduction relation

$$\varrho, M \mapsto \varsigma, N$$

- M, N terms
- ϱ, ς lists of constraints in order of generation
- one constraint for each contract operator $\rightarrow$, $\cup$, $\cap$
- one constraint for each evaluated flat contract
- cannot blame immediately: flat contract may be nested in intersection or union
- instead: blame computed from list of constraints

Evaluation Rules

Flat contracts

I-Flat

$$\varsigma, E[V \mathbin{@^b} \mathit{flat}(M)] \longrightarrow \varsigma, E[V \mathbin{@^b} \mathit{eval}(M\ V)]$$

I-Unit

$$\varsigma, E[V \mathbin{@^b} \mathit{eval}(W)] \longrightarrow b \blacktriangleleft (W) : \varsigma, E[V]$$

Evaluation Rules

Flat contracts

I-Flat

$$\varsigma, E[V \mathbin{@^b} \mathit{flat}(M)] \longrightarrow \varsigma, E[V \mathbin{@^b} \mathit{eval}(M\ V)]$$

I-Unit

$$\varsigma, E[V \mathbin{@^b} \mathit{eval}(W)] \longrightarrow b \blacktriangleleft (W) : \varsigma, E[V]$$

Constraint Satisfaction

Solution of a constraint set

$$\mu \in ((\!|b|\!) \times \{\mathit{subject}, \mathit{context}\}) \to \mathbb{B}$$

- for each blame identifier b
- assign subject blame and context blame
- drawn from $\mathbb{B} = \{\mathsf{t}, \mathsf{f}\}$
- ordered by $\mathsf{t} \sqsubset \mathsf{f}$

Constraint Satisfaction

Solution of a constraint set

$$\mu \in ((\!|b|\!) \times \{\mathit{subject}, \mathit{context}\}) \to \mathbb{B}$$

- for each blame identifier b
- assign subject blame and context blame
- drawn from $\mathbb{B} = \{\mathsf{t}, \mathsf{f}\}$
- ordered by $\mathsf{t} \sqsubset \mathsf{f}$

- Ordering reflects gathering of information with each execution step

Constraint Satisfaction

Solution of a constraint set

$$\mu \in ((\!|b|\!) \times \{\mathit{subject}, \mathit{context}\}) \to \mathbb{B}$$

- for each blame identifier b
- assign subject blame and context blame
- drawn from $\mathbb{B} = \{\mathsf{t}, \mathsf{f}\}$
- ordered by $\mathsf{t} \sqsubset \mathsf{f}$

- Ordering reflects gathering of information with each execution step
- False has "more" information because it indicates a failing contract

Flat contracts

$$\textsc{CT-Flat}\ \frac{\mu(b.\mathit{subject}) \sqsupseteq \tau(W) \qquad \mu(b.\mathit{context}) \sqsupseteq \mathrm{t}}{\mu \models b \blacktriangleleft W}$$

Constraint Satisfaction II

Flat contracts

$$\textsc{CT-Flat}\ \frac{\mu(b.\mathit{subject}) \sqsupseteq \tau(W) \qquad \mu(b.\mathit{context}) \sqsupseteq \mathsf{t}}{\mu \models b \blacktriangleleft W}$$

- Raise blame if b is a blame label from the source program and either $\mu(b.\mathit{subject}) \sqsupseteq \mathsf{f}$ or $\mu(b.\mathit{context}) \sqsupseteq \mathsf{f}$

Evaluation Rules

Function contracts

D-Fun

$$\frac{\iota_1, \iota_2 \notin \varsigma}{\varsigma, E[(V \mathbin{@^b} (C \to D))\, W] \longrightarrow b \blacktriangleleft (\iota_1 \to \iota_2) : \varsigma, E[(V\, (W \mathbin{@^{\iota_1}} C)) \mathbin{@^{\iota_2}} D]}$$

Evaluation Rules

Function contracts

D-Fun

$$\frac{\iota_1, \iota_2 \notin \varsigma}{\varsigma, E[(V \mathbin{@^b} (C \to D))\, W] \longrightarrow b \blacktriangleleft (\iota_1 \to \iota_2) : \varsigma, E[(V\, (W \mathbin{@^{\iota_1}} C)) \mathbin{@^{\iota_2}} D]}$$

Evaluation Rules

Function contracts

D-Fun

$$\frac{\iota_1, \iota_2 \notin \varsigma}{\varsigma, E[(V \mathbin{@^b} (C \to D))\, W] \longrightarrow b \blacktriangleleft (\iota_1 \to \iota_2) : \varsigma, E[(V\, (W \mathbin{@^{\iota_1}} C)) \mathbin{@^{\iota_2}} D]}$$

Satisfaction for function constraints

CT-Function

$$\frac{\mu(b.subject) \sqsupseteq \mu(\iota_1.context \wedge (\iota_1.subject \Rightarrow \iota_2.subject)) \qquad \mu(b.context) \sqsupseteq \mu(\iota_1.subject \wedge \iota_2.context)}{\mu \models b \blacktriangleleft \iota_1 \to \iota_2}$$

Evaluation Rules

Intersection contracts

D-Inter

$$\frac{\iota_1, \iota_2 \notin \varsigma}{\varsigma, E[(V \mathbin{@}^{b} (Q \cap R))\, W] \longrightarrow b \blacktriangleleft (\iota_1 \cap \iota_2) : \varsigma, E[\langle (V \mathbin{@}^{\iota_1} Q)\, W \,[]\, (V \mathbin{@}^{\iota_2} R)\, W \rangle]}$$

Evaluation Rules

Intersection contracts

D-Inter

$$\frac{\iota_1, \iota_2 \notin \varsigma}{\varsigma, E[(V \mathbin{@^b} (Q \cap R))\ W] \longrightarrow b \blacktriangleleft (\iota_1 \cap \iota_2) : \varsigma, E[\langle (V \mathbin{@^{\iota_1}} Q)\ W \,[]\, (V \mathbin{@^{\iota_2}} R)\ W \rangle]}$$

Evaluation Rules

Intersection contracts

D-Inter

$$\frac{\iota_1, \iota_2 \notin \varsigma}{\varsigma, E[(V \mathbin{@}^b (Q \cap R))\, W] \longrightarrow b \blacktriangleleft (\iota_1 \cap \iota_2) : \varsigma, E[\langle (V \mathbin{@}^{\iota_1} Q)\, W \,[]\, (V \mathbin{@}^{\iota_2} R)\, W \rangle]}$$

- Reduction proceeds independently in pair components

Evaluation Rules

Intersection contracts

D-Inter

$$\frac{\iota_1, \iota_2 \notin \varsigma}{\varsigma, E[(V \mathbin{@^b} (Q \cap R))\, W] \longrightarrow b \blacktriangleleft (\iota_1 \cap \iota_2) : \varsigma, E[\langle (V \mathbin{@^{\iota_1}} Q)\, W \mathbin{[]} (V \mathbin{@^{\iota_2}} R)\, W \rangle]}$$

- Reduction proceeds independently in pair components
- Shares the constraint list

Evaluation Rules

Intersection contracts

D-Inter

$$\frac{\iota_1, \iota_2 \notin \varsigma}{\varsigma, E[(V \mathbin{@^b} (Q \cap R))\, W] \longrightarrow b \blacktriangleleft (\iota_1 \cap \iota_2) : \varsigma, E[\langle (V \mathbin{@^{\iota_1}} Q)\, W \mathbin{[]} (V \mathbin{@^{\iota_2}} R)\, W \rangle]}$$

- Reduction proceeds independently in pair components
- Shares the constraint list

Intersection constraints

CT-Intersection

$$\frac{\mu(b.\mathit{subject}) \sqsupseteq \mu(\iota_1.\mathit{subject} \wedge \iota_2.\mathit{subject}) \quad \mu(b.\mathit{context}) \sqsupseteq \mu(\iota_1.\mathit{context} \vee \iota_2.\mathit{context})}{\mu \models b \blacktriangleleft \iota_1 \cap \iota_2}$$

Evaluation Rules

Union contracts

UNION

$$\frac{\iota_1, \iota_2 \notin \varsigma}{\varsigma, E[V \mathbin{@}^{b} (\mathcal{K}[C \cup D])] \longrightarrow b \blacktriangleleft (\iota_1 \cup \iota_2) : \varsigma, E[\langle V \mathbin{@}^{\iota_1} \mathcal{K}[C] \,[]\, V \mathbin{@}^{\iota_2} \mathcal{K}[D] \rangle]}$$

Evaluation Rules

Union contracts

UNION

$$\frac{\iota_1, \iota_2 \notin \varsigma}{\varsigma, E[V \mathbin{@}^{b} (\mathcal{K}[C \cup D])] \longrightarrow b \blacktriangleleft (\iota_1 \cup \iota_2) : \varsigma, E[\langle V \mathbin{@}^{\iota_1} \mathcal{K}[C] \,[]\, V \mathbin{@}^{\iota_2} \mathcal{K}[D] \rangle]}$$

Evaluation Rules

Union contracts

$$\textsc{Union}\quad \frac{\iota_1, \iota_2 \notin \varsigma}{\varsigma, E[V \mathbin{@}^{b} (\mathcal{K}[C \cup D])] \longrightarrow b \blacktriangleleft (\iota_1 \cup \iota_2) : \varsigma, E[\langle V \mathbin{@}^{\iota_1} \mathcal{K}[C] \,[]\, V \mathbin{@}^{\iota_2} \mathcal{K}[D] \rangle]}$$

- All uses of V refer to the same constraint on b

Evaluation Rules

Union contracts

$$\textsc{Union}\quad \frac{\iota_1, \iota_2 \notin \varsigma}{\varsigma, E[V \mathbin{@}^{b} (\mathcal{K}[C \cup D])] \longrightarrow b \blacktriangleleft (\iota_1 \cup \iota_2) : \varsigma, E[\langle V \mathbin{@}^{\iota_1} \mathcal{K}[C] \parallel V \mathbin{@}^{\iota_2} \mathcal{K}[D] \rangle]}$$

- All uses of V refer to the same constraint on b
- Inconsistent uses of the union are detected

Evaluation Rules

Union contracts

$$\textsc{Union}\quad \frac{\iota_1, \iota_2 \notin \varsigma}{\varsigma, E[V \mathbin{@}^{b} (\mathcal{K}[C \cup D])] \longrightarrow b \blacktriangleleft (\iota_1 \cup \iota_2) : \varsigma, E[\langle V \mathbin{@}^{\iota_1} \mathcal{K}[C] \parallel V \mathbin{@}^{\iota_2} \mathcal{K}[D] \rangle]}$$

- All uses of V refer to the same constraint on b
- Inconsistent uses of the union are detected

Union constraints

$$\textsc{CT-Union}\quad \frac{\mu(b.\mathit{subject}) \sqsupseteq \mu(\iota_1.\mathit{subject} \vee \iota_2.\mathit{subject}) \quad \mu(b.\mathit{context}) \sqsupseteq \mu(\iota_1.\mathit{context} \wedge \iota_2.\mathit{context})}{\mu \models b \blacktriangleleft \iota_1 \cup \iota_2}$$

Results

UNI FREIBURG

Contract soundness

1. $M \mathbin{@^{\flat}} C \in [\![C]\!]^{+}$.
2. $\mathcal{L}[\Box \mathbin{@^{\flat}} C] \in [\![C]\!]^{-}$.

Results

Contract soundness

1. $M \mathbin{@^{\flat}} C \in [\![C]\!]^{+}$.
2. $\mathcal{L}[\Box \mathbin{@^{\flat}} C] \in [\![C]\!]^{-}$.

Subject blame soundness (abridged)

Suppose that $M \in [\![C]\!]^{+}$.
If $\varrho, E[M \mathbin{@^{b}} C] \longmapsto^{*} \varsigma, N$, then $[\![\varsigma]\!](b, \mathit{subject}) \sqsubseteq \mathrm{t}$.

Results

Contract soundness

1. $M \mathbin{@^{\flat}} C \in [\![C]\!]^{+}$.
2. $\mathcal{L}[\square \mathbin{@^{\flat}} C] \in [\![C]\!]^{-}$.

Subject blame soundness (abridged)

Suppose that $M \in [\![C]\!]^{+}$.
If $\varrho, E[M \mathbin{@^{b}} C] \longmapsto^{*} \varsigma, N$, then $[\![\varsigma]\!](b, \mathit{subject}) \sqsubseteq \mathrm{t}$.

Context blame soundness (abridged)

Suppose that $\mathcal{L} \in [\![C]\!]^{-}$.
If $\varrho, \mathcal{L}[M \mathbin{@^{b}} C] \longmapsto^{*} \varsigma, N$, then $[\![\varsigma]\!](b, \mathit{context}) \sqsubseteq \mathrm{t}$.

Further Challenges Addressed

UNI FREIBURG

- Deal with $(A \cup B) \cap (C \cup D)$
- Solutions don't increase monotonically when new constraints are added
- Deterministic semantics and simulation
- Implementation

Conclusions

UNI FREIBURG

- First investigation of intersection and union contracts
- Novel semantics of contracts (subject, context)
- Implemented in TreatJS, a new contract system for JavaScript, which is available on the web
  `http://proglang.informatik.uni-freiburg.de/treatjs/`