UTP Theories of Undefinedness

Jim Woodcock & Victor Bandur

University of York

UTP Symposium, Paris: 27 August 2012

Outline

Outline

Introduction

- unifying theory for monotonic partial logics
- Victor Bandur's PhD work
- based on original ideas due to Mark Saaltink
- underpinnings for Z/Eves
- joint papers at Marktoberdorf and ICECCS 2007
- new motivation for unifying theory
- COMPASS Modelling Language CML
  - member of the *Circus* family
  - VDM + CSPM + time + OO + pointers + mobility
- what do we do about undefined expressions?
  - Jones's VDM: Logic of Partial Functions (LPF)
  - Larsen's VDM: left-to-right evaluation (Overture)
  - CSPM: arithetic overflow, boolean short circuit
  - *Circus*: classical logic, arbitrary undefined values
- does it matter?
- CML is for SoS with heterogeneous constituents
- there maybe many different formalisms in use

Roadmap for Talk

- augment alphabetised relational calculus with undefined values
- use as meta-language for five theories
  - strict, McCarthy, Kleene, classical, and semi-classical theories
- theorem: strict $\Rightarrow$ McCarthy $\Rightarrow$ Kleene
- factual ordering also relates all theories
- guard: $G \rightsquigarrow_{\boldsymbol{T}} P$
  - main theorem: you can prove G in stronger system
  - . . . and guarantee it holds in your chosen logic
- healthiness conditions for guards in general
- choose McCarthy guards for CML
- choose Isabelle for proving CML theorems

Outline

Basic Sets and Constructors

- $\mathbb{B} = \{true, false\}$: set of boolean values
- $\mathbb{U}$: universe of values, disjoint from $\mathbb{B}$
- $\perp$: specific semantic value for undefined
- $X^\perp = S \cup \{\perp\}$, for $\perp \notin X$
- $\perp$ is not a tuple or a function
- $\perp$ is not in $\mathbb{B}$ or $\mathbb{U}$
- for k, a natural number, X^k is the set of k-tuples over X
- X^0 has one element: the 0-tuple $()$
- X^* is the union of all X^ks
- function spaces:
  - $X \to Y$: set of total functions
  - $X \nrightarrow Y$: set of partial functions
- $\mathrm{dom}\, f$: domain of f
- $\mathrm{ran}\, f$: range of f

3-Valued Logic in UTP

- inspiration:
  - Roses's standard encoding (c.1950)
  - Hoare & He's UTP designs
- model 3 logical values using a pair: (P, Q)
- intuitive meaning:
  - P is the region where (P, Q) is true
  - Q is the region where (P, Q) is defined
- observational variable: def
- combines two predicates into single predicate
- model: $(def \Rightarrow P) \wedge (Q = def)$
- interpretation:

$$\begin{array}{rcccc} R = true & = & (true, true) & = & def \\ R = false & = & (false, true) & = & false \\ R = \bot & = & \left\{ \begin{array}{c} (true, false) \\ (false, false) \end{array} \right\} & = & \neg\, def \end{array}$$

3-Valued Logic in UTP

- $(P \wedge Q, Q) = (P, Q)$

$$
\begin{aligned}
&(P \wedge Q, Q) \\
&= (def \Rightarrow P \wedge Q) \wedge (Q = def) \\
&= (def \Rightarrow P) \wedge (Q = def) \\
&= (P, Q)
\end{aligned}
$$

- $R = (R^t, \neg\, R^f), \qquad \text{where} R^b = R[b/def]$

$$
\begin{aligned}
&((P, Q)^t, \neg\, (P, Q)^f) \\
&= (((def \Rightarrow P) \wedge (Q = def))^t, \neg\, ((def \Rightarrow P) \wedge (Q = def))^f) \\
&= ((true \Rightarrow P) \wedge (Q = true), \neg\, ((false \Rightarrow P) \wedge (Q = false))) \\
&= (P \wedge Q, \neg\, (true \wedge \neg\, Q)) \\
&= (P \wedge Q, Q)
\end{aligned}
$$

Example

- $(z = x/y)$ is defined exactly when $(y \neq 0)$:

  $((x = y * z), (y \neq 0))$

- three example predicates:

$$
\begin{aligned}
&(3 = 6/2) \\
&= ((6 = 2 * 3), (2 \neq 0)) \\
&= (true, true) \\
&= def
\end{aligned}
$$

$$
\begin{aligned}
&(2 = 6/2) \\
&= ((6 = 2 * 2), (2 \neq 0)) \\
&= (false, true) \\
&= false
\end{aligned}
$$

$$
\begin{aligned}
&(2 = 6/0) \\
&= ((6 = 0 * 2), (0 \neq 0)) \\
&= (false, false) \\
&= \neg\, def
\end{aligned}
$$

Conjunction

- truth tables: $3^9 \simeq 20,000$ combinations
- we choose a strict interpretation

$\wedge$	def	$\neg\ def$	$false$
def	def	$\neg\ def$	$false$
$\neg\ def$	$\neg\ def$	$\neg\ def$	$\neg\ def$
$false$	$false$	$\neg\ def$	$false$

$\wedge$	$true$	$\bot$	$false$
$true$	$true$	$\bot$	$false$
$\bot$	$\bot$	$\bot$	$\bot$
$false$	$false$	$\bot$	$false$

- model: $(P, Q) \wedge (R, S) = (P \wedge R, Q \wedge S)$
- example:

$$
\begin{aligned}
&(y = 3) \wedge (z = x/y) \\
&= ((y = 3), true) \wedge ((x = y * z), (y \neq 0)) \\
&= ((y = 3) \wedge (x = y * z), true \wedge (y \neq 0)) \\
&= ((y = 3) \wedge (x = 3 * z), (y \neq 0))
\end{aligned}
$$

Negation

- truth table:

$\neg$	
def	$false$
$\neg\ def$	$\neg\ def$
$false$	def

$\neg$	
$true$	$false$
$\bot$	$\bot$
$false$	$true$

- model:

$$\neg\ (P, Q) = (\neg\ P, Q)$$

- example:

$$\begin{aligned}
&\neg\ (z = x/y) \\
&= \neg\ ((x = y * z), (y \neq 0)) \\
&= ((x \neq y * z), (y \neq 0))
\end{aligned}$$

Disjunction

- truth table:

$\vee$	def	$\neg\ def$	$false$
def	def	$\neg\ def$	def
$\neg\ def$	$\neg\ def$	$\neg\ def$	$\neg\ def$
$false$	def	$\neg\ def$	$false$

$\vee$	$true$	$\bot$	$false$
$true$	$true$	$\bot$	$true$
$\bot$	$\bot$	$\bot$	$\bot$
$false$	$true$	$\bot$	$false$

- model:

$$(P, Q) \vee (R, S) = (P \vee Q, R \wedge S)$$

Example

- define $P \Rightarrow Q$ as $\neg\, P \vee Q$
- suppose f is a partial function symbol, such that

$$(y = f(x)) = ((y = f(x)), x \in \operatorname{dom} f)$$

$$\begin{aligned}
& x \in \operatorname{dom} f \Rightarrow (y = f(x)) \\
&= \neg\, (x \in \operatorname{dom} f) \vee (y = f(x)) \\
&= \neg\, (x \in \operatorname{dom} f, \mathit{true}) \vee (y = f(x)) \\
&= (x \notin \operatorname{dom} f, \mathit{true}) \vee (y = f(x)) \\
&= (x \notin \operatorname{dom} f, \mathit{true}) \vee ((y = f(x)), x \in \operatorname{dom} f) \\
&= (x \notin \operatorname{dom} f \vee (y = f(x)), \mathit{true} \wedge x \in \operatorname{dom} f) \\
&= (x \in \operatorname{dom} f \Rightarrow (y = f(x)), x \in \operatorname{dom} f) \\
&= ((y = f(x)), x \in \operatorname{dom} f)
\end{aligned}$$

Equality

- equality with undefined values:

$$\begin{array}{rcl@{\qquad}rcl} (def = \neg\, def) & = & false & (true{=}\bot) & = & false \\ (def = false) & = & false & (true{=}false) & = & false \\ (\neg\, def = false) & = & false & (\bot{=}false) & = & false \end{array}$$

- together with the symmetric closure
- other words: it's just classical equality
- set membership and comprehension analogous

Example

- one of our definitions:

$$(f(x,y) = \bot) \lhd (x = \bot) \lor (y = \bot) \rhd (f(x,y) = (x = y))$$

- UTP equality contains the use of 3-valued logic
- each equality is either $true$ or $false$, not $\bot$
- we restrict our use of quantifiers to 2-valued predicates

Lemma

1. $Q \Rightarrow (\neg\,(P, Q) = \neg\, P)$
2. $Q \wedge S \Rightarrow ((P, Q) \wedge (R, S) = P \wedge Q)$
3. $Q \wedge S \Rightarrow ((P, Q) \vee (R, S) = P \vee Q)$

- this justifies UTP with 3-valued logic
- we won't use definite description or partial functions...
- ...but we will *discuss* them
- our meta-logic cannot manufacture undefined values...
- ...but it can *discuss* them

Outline

Contexts for First-order Theories

- we introduce a context ***CXT*** for our first-order theories
- all our theories are subtheories of this context

signature

$$
\begin{aligned}
PShape &: \mathbb{P}((\mathbb{U}^\perp)^* \pfun \mathbb{B}^\perp) \\
FShape &: \mathbb{P}((\mathbb{U}^\perp)^* \pfun \mathbb{U}^\perp) \\
= &: \mathbb{U}^\perp \times \mathbb{U}^\perp \to \mathbb{B}^\perp \\
\neg &: \mathbb{B}^\perp \to \mathbb{B}^\perp \\
\vee &: \mathbb{B}^\perp \times \mathbb{B}^\perp \to \mathbb{B}^\perp \\
\forall &: (\mathbb{U} \pfun \mathbb{B}^\perp) \to \mathbb{B}^\perp \\
\iota &: (\mathbb{U} \pfun \mathbb{B}^\perp) \to \mathbb{U}^\perp
\end{aligned}
$$

healthiness condition

$$
\mathbf{CTX}(P) =
$$

$$
P \wedge (\forall f : \mathbb{U} \pfun \mathbb{B}^\perp \bullet f \neq \emptyset \Rightarrow \iota(f) \in \operatorname{dom} f^\perp)
$$

Example

- context with no predicate symbols
- only monadic and dyadic function symbols

$$\textbf{\textit{X1}}(P) = P \wedge (PShape = \emptyset) \wedge (FShape = (\mathbb{U}^{\perp} \cup (\mathbb{U}^{\perp})^2 \nrightarrow \mathbb{U}^{\perp}))$$

- $PShape$ and $FShape$ are used to add type information
- we use them to restrict how undefined expressions are treated
- *e.g.*, predicate and function symbols may be
  - strict
  - definite
  - monotonic

First-Order Theory

- richer theory acts as model for context
- domain, variables, predicate & function symbols
- ***FOT*** extends ***CXT***:

alphabet

A partition $\langle Var, Pred, Fun \rangle$
$Dom : \mathbb{P}\,\mathbb{U}$
$\rho : Pred \cup Fun \to \mathbb{N}$

healthiness conditions

$$\mathbf{DV}(P) = P \land (\forall\, v : Var \bullet v \in Dom)$$
$$\mathbf{DP}(P) = P \land (\forall\, p : Pred \bullet p \in ((Dom^{\perp})^{\rho(p)} \to \mathbb{B}^{\perp}) \cap PShape)$$
$$\mathbf{DF}(P) = P \land (\forall\, f : Fun \bullet f \in ((Dom^{\perp})^{\rho(f)} \to Dom^{\perp}) \cap FShape)$$

Example

- consider a theory ***T1*** with context ***X1***
- just a single function symbol for integer division:

$$
\begin{aligned}
&\textbf{\textit{T1}}(P) = \\
&\quad \textbf{\textit{X1}}(P) \\
&\quad \wedge\ Var = \emptyset \\
&\quad \wedge\ Pred = \emptyset \\
&\quad \wedge\ Fun = \{\_/\_\} \\
&\quad \wedge\ Dom = \mathbb{N} \\
&\quad \wedge\ \rho = \{\_/\_ \mapsto 2\} \\
&\quad \wedge\ \_/\_ \in (\mathbb{N}^{\perp} \times \mathbb{N}^{\perp} \to \mathbb{N}^{\perp}) \cap FShape
\end{aligned}
$$

Ordering

- elements: for any set X, $a, b \in X$

  $$a \sqsubseteq b \;=\; (a \neq \bot) \Rightarrow (a = b)$$

- pointwise extension to tuples: for $x, y \in X^k$

  $$x \sqsubseteq y \;=\; \forall\, i : 1 \,..\, k \bullet x_i \sqsubseteq y_i$$

- pointwise extension to functions: for $f, g \in X \to Y$

  $$f \sqsubseteq g \;=\; (\operatorname{dom} f = \operatorname{dom} g) \wedge (\forall\, x : \operatorname{dom} f \bullet f(x) \sqsubseteq g(x))$$

- comparing sets of functions: for $A, B : \mathbb{P}\, X$
  - Hoare preorder: $A \sqsubseteq_H B \;=\; \forall\, a : A \bullet \exists\, b : B \bullet a \sqsubseteq b$
  - Smyth preorder: $A \sqsubseteq_S B \;=\; \forall\, b : B \bullet \exists\, a : A \bullet a \sqsubseteq b$
- irrelevant which ordering we choose (consistently)

Examples

1. elements:

$$\begin{aligned} \bot &\sqsubseteq 1 \\ 1 &\sqsubseteq 1 \\ \neg\,(1 &\sqsubseteq 2) \end{aligned}$$

2. tuples:

$$\begin{aligned} (0, \bot, 2) &\sqsubseteq (0, 1, 2) \\ () &\sqsubseteq () \\ (1, 2) &\sqsubseteq (1, 2) \\ \neg\,((1, 2) &\sqsubseteq (2, 2)) \end{aligned}$$

Examples

3. functions:

$$(\lambda\, x, y : \mathbb{N} \bullet \bot \lhd (y = 0) \rhd x/y)$$
$$\sqsubseteq (\lambda\, x, y : \mathbb{N} \bullet 0 \lhd (y = 0) \rhd x/y)$$
$$(\lambda\, n : \mathbb{N} \bullet \bot \lhd (n \bmod 2 = 0) \rhd n) \sqsubseteq (\lambda\, n : \mathbb{N} \bullet n)$$

4. sets of functions:

$$\{(\lambda\, x, y : \mathbb{N} \bullet \bot \lhd (y = 0) \rhd x/y),$$
$$(\lambda\, n : \mathbb{N} \bullet \bot \lhd (n \bmod 2 = 0) \rhd n),$$
$$(\lambda\, n : \mathbb{N} \bullet n)\}$$

$$\sqsubseteq_H$$

$$\{(\lambda\, x, y : \mathbb{N} \bullet 0 \lhd (y = 0) \rhd x/y),$$
$$(\lambda\, n : \mathbb{N} \bullet n)\}$$

Comparing Contexts

$$\boldsymbol{S} \sqsubseteq_H \boldsymbol{T} = \forall\, P : \boldsymbol{S};\ Q : \boldsymbol{T} \bullet P \sqsubseteq_H Q$$

where

$$P \sqsubseteq_H Q =$$

$$\begin{aligned}
&PShape_{\boldsymbol{S}} \sqsubseteq_H PShape_{\boldsymbol{T}} \\
&\land\ FShape_{\boldsymbol{S}} \sqsubseteq_H FShape_{\boldsymbol{T}} \\
&\land\ (=_{\boldsymbol{S}}) \sqsubseteq (=_{\boldsymbol{T}}) \\
&\land\ (\neg_{\boldsymbol{S}}) \sqsubseteq (\neg_{\boldsymbol{T}}) \\
&\land\ (\vee_{\boldsymbol{S}}) \sqsubseteq (\vee_{\boldsymbol{T}}) \\
&\land\ (\forall_{\boldsymbol{S}}) \sqsubseteq (\forall_{\boldsymbol{T}}) \\
&\land\ (\iota_{\boldsymbol{S}}) \sqsubseteq (\iota_{\boldsymbol{T}})
\end{aligned}$$

Example

- consider ***X2***, a subtheory of ***X1***:

  $\forall f : FShape_{\mathbf{X1}} \bullet zero \circ f \in FShape_{\mathbf{X2}}$

- where the total function $zero$ is defined:

  $zero(x) = (0 \lhd (x = \bot) \rhd x)$

- all other components unchanged
- then $P_{\mathbf{X1}} \sqsubseteq_H P_{\mathbf{X2}}$, since

  $f \sqsubseteq zero \circ f$
  $= (\mathrm{dom}\, f = \mathrm{dom}(zero \circ f)) \wedge \forall x : \mathrm{dom}\, f \bullet f(x) \sqsubseteq zero \circ f(x)$

  $FShape_{X1} \sqsubseteq_H FShape_{X2} =$
  $\forall f : FShape_{\mathbf{X1}} \bullet \exists g : FShape_{\mathbf{X2}} \bullet f \sqsubseteq g$

Strictness

- function $f : (X^\perp)^{\rho(f)} \to Y^\perp$ is strict:

  $$\textbf{\textit{strict}}(f) = \forall\, x : (X^\perp)^{\rho(f)} \bullet (\exists\, i : 1 \,..\, \rho(f) \bullet (x_i = \perp)) \Rightarrow (f(x) = \perp)$$

- example:
  - suppose that $\_ * \_$ is standard multiplication operator on natural numbers:

    $$\_ * \_ : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

  - define strict version of the operator:

    $$\_ * \_ : \mathbb{N}^\perp \times \mathbb{N}^\perp \to \mathbb{N}^\perp$$
    $$x * y = \perp \lhd (x = \perp) \lor (y = \perp) \rhd x * y$$

Strict Contexts

- making a context $\boldsymbol{T}$ strict:

$$
\begin{array}{l}
\boldsymbol{strict}(\boldsymbol{T}) = \{\, P : \boldsymbol{T} \bullet \boldsymbol{strict}(P) \,\} \\
\boldsymbol{where} \\
\boldsymbol{strict}(P) = \\
\quad \exists\, PShape_0, FShape_0 \bullet \\
\quad\quad PShape = \{\, p : PShape_0 \mid \boldsymbol{strict}(p) \,\} \\
\quad\quad \wedge\ FShape = \{\, f : FShape_0 \mid \boldsymbol{strict}(f) \,\} \\
\quad\quad \wedge\ \boldsymbol{T}[PShape_0, FShape_0 / PShape, FShape]
\end{array}
$$

Definite

- function $f : (X^\perp)^{\rho(f)} \to Y^\perp$ is definite:

  $$\textbf{\textit{definite}}(f) = \forall\, x : (X^\perp)^{\rho(f)} \bullet (f(x) = \perp) \Rightarrow (\exists\, i : 1 \,..\, \rho(f) \bullet (x_i = \perp))$$

- example:

  $\_ * \_$ is definite

Definite Contexts

- making a context definite:

$$\mathbf{definite}(\mathbf{T}) = \{\, P : \mathbf{T} \bullet \mathbf{definite}(P) \,\}$$

where

$$\mathbf{definite}(P) =$$
$$\exists\, PShape_0, FShape_0 \bullet$$
$$PShape = \{\, p : PShape_0 \mid \mathbf{definite}(p) \,\}$$
$$\land\ FShape = \{\, f : FShape_0 \mid \mathbf{definite}(f) \,\}$$
$$\land\ \mathbf{T}[PShape_0, FShape_0 / PShape, FShape]$$

Monotonic

- function $f : (X^{\perp})^{\rho(f)} \to Y^{\perp}$ is monotonic:

$$\textbf{\textit{monotonic}}(f) = \\ \forall\, x_1, x_2 : (X^{\perp})^{\rho(f)} \bullet \\ x_1 \sqsubseteq x_2 \Rightarrow f(x_1) \sqsubseteq f(x_2)$$

- example: $\neg$ is monotonic

$\neg$	
$true$	$false$
$\perp$	$\perp$
$false$	$true$

Monotonic Contexts

- $\boldsymbol{T}$ is a monotonic context:

$$\begin{array}{l}
\mathbf{monotonic}(\boldsymbol{T}) = \forall\, P : \boldsymbol{T} \bullet \mathbf{monotonic}(P) \\
\mathbf{where} \\
\quad \mathbf{monotonic}(P) = \\
\quad\quad (\forall\, p : Pred_{\boldsymbol{T}} \bullet \mathbf{monotonic}(p)) \\
\quad\quad \wedge\ (\forall\, f : Fun_{\boldsymbol{T}} \bullet \mathbf{monotonic}(f)) \\
\quad\quad \wedge\ \mathbf{monotonic}(=_{\boldsymbol{T}}) \\
\quad\quad \wedge\ \mathbf{monotonic}(\neg_{\boldsymbol{T}}) \\
\quad\quad \wedge\ \mathbf{monotonic}(\vee_{\boldsymbol{T}}) \\
\quad\quad \wedge\ \mathbf{monotonic}(\forall_{\boldsymbol{T}}) \\
\quad\quad \wedge\ \mathbf{monotonic}(\iota_{\boldsymbol{T}})
\end{array}$$

Comparing FOTs

- comparing FOTs $\boldsymbol{U}$ and $\boldsymbol{V}$: for $P : \boldsymbol{U}$ and $Q : \boldsymbol{V}$

$$
\begin{aligned}
P \sqsubseteq_H Q = \\
& Dom_{\boldsymbol{U}} = Dom_{\boldsymbol{V}} \\
& \wedge\ Pred_{\boldsymbol{U}} \sqsubseteq_H Pred_{\boldsymbol{V}} \\
& \wedge\ Fun_{\boldsymbol{U}} \sqsubseteq_H Fun_{\boldsymbol{V}}
\end{aligned}
$$

Two Lemmas

Models

- suppose that we have two CXTs $\boldsymbol{S}$ and $\boldsymbol{T}$
- where $\boldsymbol{S} \sqsubseteq_H \boldsymbol{T}$
- suppose that $\boldsymbol{U}$ is a FOT extending $\boldsymbol{S}$
- then there is a FOT $\boldsymbol{V}$ extending $\boldsymbol{T}$
- such that $\boldsymbol{U} \sqsubseteq \boldsymbol{V}$

Expression Consistency

- suppose that e is an expression over a FOT $\boldsymbol{U}$
- then every $\boldsymbol{U}$-healthy predicate P ensures:

$$P \Rightarrow e \in D_{\boldsymbol{U}}^{\perp}$$

Example: Application of Model Lemma

	S	T
$PShape$	$\emptyset$	$\emptyset$
$FShape$	$\mathbf{strict}(\mathbb{U}^{\perp} \nrightarrow \mathbb{U}^{\perp})$	$\mathbb{U}^{\perp} \nrightarrow \mathbb{U}$
	U	V
Dom	$\{0,1\}$	$\{0,1\}$
ρ	$\{inc \mapsto 1\}$	$\{inc \mapsto 1\}$
A	$inc(\perp) = \perp$	$inc(\perp) = 0$
	$inc(0) = 1$	$inc(0) = 1$
	$inc(1) = \perp$	$inc(1) = 1$

Theorem: Construct Monotonicity

- suppose $S \sqsubseteq_H T$
- U extends S and V extends T
- either S or T is monotonic
- then, for any construct c, we have

  $c_U \sqsubseteq c_V$

- Proof:
  - by induction on structure of c

Construct Monotonicity: Induction Case 2

application of function symbol to actual parameters

induction hypothesis $x_S \sqsubseteq x_T$

case 2.1: S is monotonic

$(f(x))_S$

$= \{\textit{interpretation}\}$

$f_S(x_S)$

$= \{\textit{hypothesis + } S \textit{ monotonic}\}$

$f_S(x_T)$

$\sqsubseteq \{\textit{assumption: } P_S \sqsubseteq_H Q_T\}$

$f_T(x_T)$

$= \{\textit{interpretation}\}$

$(f(x))_T$

case 2.2: T is monotonic

$(f(x))_S$

$= \{\textit{interpretation}\}$

$f_S(x_S)$

$\sqsubseteq \{\textit{assumption: } P_S \sqsubseteq_H P_T\}$

$f_T(x_S)$

$= \{\textit{hypothesis + } T \textit{ monotonic}\}$

$f_T(x_T)$

$= \{\textit{interpretation}\}$

$(f(x))_T$

Outline

Strict First-Order Theory

$$\textbf{\textit{S1}}(P) = \textbf{\textit{strict}}(P)$$

$$(=_{\textbf{\textit{s}}}(x, y) = \bot) \lhd (x = \bot) \vee (y = \bot) \rhd (=_{\textbf{\textit{s}}}(x, y) = (x = y))$$

$$(\iota_{\textbf{\textit{s}}}(f) = x) \lhd \bot \notin \operatorname{ran} f \wedge (\operatorname{dom}(f \rhd \{\mathit{true}\}) = \{x\}) \rhd (\iota_{\textbf{\textit{s}}}(f) = \bot)$$

$$(\forall_{\textbf{\textit{s}}}(f) = \bot) \lhd \bot \in \operatorname{ran} f \rhd (\forall_{\textbf{\textit{s}}}(f) = (\operatorname{ran} f = \{\mathit{true}\}))$$

$$\neg_{\textbf{\textit{s}}}(P) = \neg\, P$$

$$\vee_{\textbf{\textit{s}}}\,(P, Q) = P \vee Q$$

$\neg_{\textbf{\textit{s}}}$	
true	false
$\bot$	$\bot$
false	true

$\vee_{\textbf{\textit{s}}}$	true	$\bot$	false
true	true	$\bot$	true
$\bot$	$\bot$	$\bot$	$\bot$
false	true	$\bot$	false

Kleene System

$$\mathbf{K1}(P) = P \land (\forall f : Pred_{\mathbf{k}} \cup Fun_{\mathbf{k}} \bullet \mathbf{monotonic}(f))$$

$$(=_{\mathbf{k}}) = (=_{\mathbf{s}})$$

$$(\iota_{\mathbf{k}}) = (\iota_{\mathbf{s}})$$

$$((\forall_{\mathbf{k}}(f) = false) \lhd false \in \operatorname{ran} f \rhd$$
$$((\forall_{\mathbf{k}}(f) = true) \lhd (\operatorname{ran} f = \{true\}) \rhd (\forall_{\mathbf{k}}(f) = \bot)))$$

$$\neg_{\mathbf{k}} = \neg_{\mathbf{s}}$$

$$((\vee_{\mathbf{k}}(P, Q) = true) \lhd (P = true) \vee (Q = true) \rhd$$
$$((\vee_{\mathbf{k}}(P, Q) = false) \lhd (P = false) \land (Q = false) \rhd$$
$$(\vee_{\mathbf{k}}(P, Q) = \bot)))$$

$\vee_{\mathbf{k}}$	$true$	$\bot$	$false$
$true$	$true$	$true$	$true$
$\bot$	$true$	$\bot$	$\bot$
$false$	$true$	$\bot$	$false$

McCarthy System

$$\boldsymbol{M1 = K1}$$
$$(=_{\boldsymbol{m}}) = (=_{\boldsymbol{k}})$$
$$\iota_{\boldsymbol{m}} = \iota_{\boldsymbol{k}}$$
$$\forall_{\boldsymbol{m}} = \forall_{\boldsymbol{k}}$$
$$\neg_{\boldsymbol{m}} = \neg_{\boldsymbol{k}}$$
$$((\vee_{\boldsymbol{m}}(P, Q) = true) \lhd (P = true) \vee ((P = false) \wedge (Q = true)) \rhd ((\vee_{\boldsymbol{m}}(P, Q) = \bot) \lhd (P = \bot) \vee (Q = \bot) \rhd (\vee_{\boldsymbol{m}}(P, Q) = false)))$$

$\vee_{\boldsymbol{m}}$	$true$	$\bot$	$false$
$true$	$true$	$true$	$true$
$\bot$	$\bot$	$\bot$	$\bot$
$false$	$true$	$\bot$	$false$

Lemmas

- strict system is monotonic
- Kleene system is monotonic
- McCarthy system is monotonic
- for $\rho_s = \rho_m = \rho_k$ and $Dom_s = Dom_m = Dom_k$

$$FOT_s \sqsubseteq FOT_m \sqsubseteq FOT_k$$

Semi-classical System

1. $\boldsymbol{CXT_s \sqsubseteq CXT_{sc}}$
2. $PShape_{\boldsymbol{sc}} \subseteq (\mathbb{U}^{\perp})^* \nrightarrow \mathbb{B}$
3. $=_{\boldsymbol{sc}} \in \mathbb{U}^{\perp} \times \mathbb{U}^{\perp} \to \mathbb{B}$
4. $\neg_{\boldsymbol{sc}} \in \mathbb{B}^{\perp} \to \mathbb{B}$
5. $\vee_{\boldsymbol{sc}} \in \mathbb{B}^{\perp} \times \mathbb{B}^{\perp} \to \mathbb{B}$
6. $\forall_{\boldsymbol{sc}} \in (\mathbb{U}^{\perp} \nrightarrow \mathbb{B}^{\perp}) \to \mathbb{B}$

Classical Theory

1. $\boldsymbol{CXT}_s \sqsubseteq \boldsymbol{CXT}_c$
2. $FShape$ contains only definite functions
3. $PShape$ contains only definite functions
4. $\forall f : \mathbb{U} \pfun \mathbb{B} \bullet (f \neq \bot) \Rightarrow (\iota_c(f) \neq \bot)$

Theorem

- suppose that $\boldsymbol{CXT_a}$ is such that $\boldsymbol{CXT_s} \sqsubseteq \boldsymbol{CXT_a}$
- then there is a semi-classical system $\boldsymbol{CXT_b}$ such that $\boldsymbol{CXT_a} \sqsubseteq \boldsymbol{CXT_b}$
- proof
  - define

    $$raise : (X^{\perp} \to \mathbb{B}^{\perp}) \to (X^{\perp} \to \mathbb{B})$$
    $$raise(f)(x) = (false \lhd (f(x) = \perp) \rhd f(x))$$

  - then construct

    $$PShape_{\boldsymbol{b}} = \{\, p : PShape_{\boldsymbol{a}} \bullet raise(p) \,\}$$
    $$(=_{\boldsymbol{b}}) = raise(=_{\boldsymbol{a}})$$

- suppose that $\boldsymbol{CXT_a}$ is such that $\boldsymbol{CXT_s} \sqsubseteq \boldsymbol{CXT_a}$
- then there is a classical system $\boldsymbol{CXT_b}$
- such that $\boldsymbol{CXT_a} \sqsubseteq \boldsymbol{CXT_b}$

Validity

- suppose $\boldsymbol{T}$ is a $\boldsymbol{CXT}$ and P is a predicate
- define P is valid in $\boldsymbol{T}$

$$\boldsymbol{T} \models P \;=\; \text{for all } \boldsymbol{U}, \boldsymbol{T} \sqsubseteq_H \boldsymbol{U} \text{ implies } P_{\boldsymbol{U}} = true$$

Guards

- suppose c is a construct
- predicate G is a guard for c in $\boldsymbol{CXT_T}$
- $G \rightsquigarrow_{\boldsymbol{T}} P$
- iff for every $\boldsymbol{FOT_V}$ that extends $\boldsymbol{CXT_T}$ we have
  1. $(G_{\boldsymbol{V}} \neq \bot)$
  2. $(G_{\boldsymbol{V}} = true) \Rightarrow (c_{\boldsymbol{V}} \neq \bot)$
- G is a tight guard if we also have
  3. $(G_{\boldsymbol{V}} = false) \Rightarrow (c_{\boldsymbol{V}} = \bot)$

Outline

Main Theorem (Saaltink)

- suppose $\boldsymbol{CXT_S} \sqsubseteq \boldsymbol{CXT_T}$
- either one is monotonic
- G is a guard for P in $\boldsymbol{CXT_S}$
- then if $(\boldsymbol{T} \models G)$ and $(\boldsymbol{T} \models P)$, then $(\boldsymbol{S} \models P)$

- significance: trading theorems between provers

- example
  - we want a proof of P in Larsen's VDM (Overture)
  - but the only theorem prover we have is for Jones's VDM
  - Overture $\sqsubseteq$ LPF
  - find a guard G for P in Overture (McCarthy logic)
  - carry out the proof of G in Jones's logic (Kleene)
  - carry out the proof of P in Jones's logic
  - then P is a theorem in Overture

- there are other interesting theorems relating these logics

Proof of Main Theorem

- models lemma
  - $\boldsymbol{CXT_S} \sqsubseteq \boldsymbol{CXT_T}$ and $\boldsymbol{FOT_U}$ extends $\boldsymbol{CXT_S}$
  - $\therefore$ there exists $\boldsymbol{FOT_V}$ that extends $\boldsymbol{CXT_T}$ and $\boldsymbol{CXT_U} \sqsubseteq \boldsymbol{CXT_V}$
- $G \rightsquigarrow_{\boldsymbol{S}} P$
  - $(G_{\boldsymbol{U}} \neq \bot) \wedge ((G_{\boldsymbol{U}} = true) \Rightarrow (P_{\boldsymbol{U}} \neq \bot))$
- construct monotonicity ($\boldsymbol{S}$ monotonic)
  - $G_{\boldsymbol{U}} \sqsubseteq G_{\boldsymbol{V}}, \quad \therefore (G_{\boldsymbol{U}} = G_{\boldsymbol{V}})$
  - but $\boldsymbol{T} \models G$, so $(G_{\boldsymbol{V}} = true)$
  - $\therefore (G_{\boldsymbol{U}} = true)$
  - $\therefore (P_{\boldsymbol{U}} \neq \bot)$
- construct monotonicity ($\boldsymbol{S}$ monotonic)
  - $P_{\boldsymbol{U}} \sqsubseteq P_{\boldsymbol{V}}, \quad \therefore (P_{\boldsymbol{U}} = P_{\boldsymbol{V}})$
  - but $\boldsymbol{T} \models P$, so $(P_{\boldsymbol{V}} = true)$
  - $\therefore (P_{\boldsymbol{U}} = true)$

Outline

Definedness Guards

- suppose e is an expression
- $\mathcal{D}e$: combination of definedness of alphabetic variables and additional constraints
- example
  - let $\alpha((x + y)/z) = \{w, x, y, z\}$
  - $\mathcal{D}((x + y)/z) = \mathcal{D}x \land \mathcal{D}y \land \mathcal{D}z \land z \neq 0$
- $\mathcal{D}x$ is a variable dependent on the name "x":
  - example: x can be subjected to substitution in $\mathcal{D}x$
    - $\mathcal{D}x[y/x] = \mathcal{D}y$
  - example: $\mathcal{D}w$ can be used as a bound variable
    - $\mathcal{D}((x + y)/z) \land \neg\, \mathcal{D}w \Rightarrow \forall\, w, \mathcal{D}w \bullet \mathcal{D}((x + y)/z)$
- logical theories differ in the precise definition of $\mathcal{D}$
- but they share much common ground
- definedness function is first order:

$$\textbf{\textit{D1}}(\mathcal{D}\Phi) \;=\; \mathcal{D}\Phi \land \mathcal{D}(\mathcal{D}\Phi)$$

Guards for Definite McCarthy System

$$
\begin{array}{rcl}
\mathcal{D}_{\boldsymbol{m}} x & = & true \\
\mathcal{D}_{\boldsymbol{m}}(p(e)) & = & \forall\, i : 1 \,..\, \rho(P) \bullet \mathcal{D}_{\boldsymbol{m}} e_i \\
\mathcal{D}_{\boldsymbol{m}}(f(e)) & = & \forall\, i : 1 \,..\, \rho(f) \bullet \mathcal{D}_{\boldsymbol{m}} e_i \\
\mathcal{D}_{\boldsymbol{m}}(e_1 = e_2) & = & \mathcal{D}_{\boldsymbol{m}} e_1 \wedge \mathcal{D}_{\boldsymbol{m}} e_2 \\
\mathcal{D}_{\boldsymbol{m}}(\neg\, P) & = & \mathcal{D}_{\boldsymbol{m}} P \\
\mathcal{D}_{\boldsymbol{m}}(P \vee Q) & = & \mathcal{D}_{\boldsymbol{m}} P \wedge (P \vee \mathcal{D}_{\boldsymbol{m}} Q) \\
\mathcal{D}_{\boldsymbol{m}}(\forall\, x \bullet P) & = & \forall\, x \bullet \mathcal{D}_{\boldsymbol{m}} P \\
\mathcal{D}_{\boldsymbol{m}}(\iota x \bullet P) & = & (\forall\, x \bullet \mathcal{D}_{\boldsymbol{m}} P) \wedge (\exists_1\, x \bullet P)
\end{array}
$$

theorem

- if c is a construct, $\mathcal{D}_{\boldsymbol{m}}(c)$ is a guard for c in ***definite***$(\boldsymbol{T})$, and a tight guard for c in ***strict***(***definite***$(\boldsymbol{T})$)

Conclusions

- Contribution
  - a theory for unifying theories with undefined expressions
  - precondition predicates for indefinite systems
  - theorems for guards for definite and indefinite systems
    - strict, McCarthy, Kleene, classical, semi-classical
  - semantics for monotonic, partial logics
  - meta-theorems for trading predicates
  - use classical logic for proving facts in all logics
  - use classical logic for refuting conjectures in all logics
- future work
  - comprehensive treatment of undefined expressions in CML
  - can we include every treatment of undefinedness?
  - what about:
    - Alloy paradigm: no function application
    - LCF: quantifiers also range over undefined values
    - 2nd-order undefinedness