Gradual Typing from a Categorical Perspective

Harley Eades III and Michael Townsend
Augusta University
{heades,mitownsend}@augusta.edu

Categories and Subject Descriptors D.3.3 [*??*]: ??—??

General Terms TODO

Keywords TODO

Abstract

TODO

1. Introduction

(Scott 1980) showed how to model the untyped λ-calculus within a cartesian closed category, $\mathcal{C}$, with a distinguished object we will call ? – read as the type of untyped terms – such that the object (type) $? \to ?$ is a retract of ?. That is, there are morphisms $\mathbf{squash} : (? \to ?) \longrightarrow ?$ and $\mathbf{split} : ? \longrightarrow (? \to ?)$ where $\mathbf{squash}; \mathbf{split} = \mathsf{id} : (? \to ?) \longrightarrow (? \to ?)$[1].

In the same volume as Scott (Lambek 1980) showed that cartesian closed categories also model the typed λ-calculus. Suppose we want to model the typed λ-calculus with pairs and natural numbers. That is, given two types A_1 and A_2 there is a type $A_1 \times A_2$, and there is a type $\mathbf{Nat}$. Furthermore, we have first and second projections, and zero and successor functions. This situation can easily be modeled by a cartesian closed category $\mathcal{C}$ – see Section 2 for the details – but also add to $\mathcal{C}$ the type of untyped terms ?, $\mathbf{squash}$, and $\mathbf{split}$. At this point $\mathcal{C}$ is a model of both the typed and the untyped λ-calculus. However, the two theories are really just sitting side by side in $\mathcal{C}$ and cannot really interact much.

Suppose $\mathcal{T}$ is a discrete category with the objects $\mathbf{Nat}$ and 1 (the terminal object or empty product) and $\mathsf{T} : \mathcal{T} \longrightarrow \mathcal{C}$ is a full and faithful functor. This implies that $\mathcal{T}$ is a subcategory of $\mathcal{C}$, and that $\mathcal{T}$ is the category of atomic types. Then for any type A of $\mathcal{T}$ we add to $\mathcal{C}$ the morphisms $\mathsf{box} : \mathsf{T}A \longrightarrow ?$ and $\mathsf{unbox} : ? \longrightarrow \mathsf{T}A$ such that $\mathsf{box}; \mathsf{unbox} = \mathsf{id} : \mathsf{T}A \longrightarrow \mathsf{T}A$ making $\mathsf{T}A$ a retract of ?. This is the bridge allowing the typed world to interact with the untyped one. What we have just built up is a categorical model that offers a new perspective of gradual typing.

(Siek and Taha 2006)

[1] We denote composition of morphisms by $f; g : A \longrightarrow C$ given morphisms $f : A \longrightarrow B$ and $g : B \longrightarrow C$.

2. Categorical Model

Definition 1. *Suppose $\mathcal{C}$ is a category. Then an object A is a **retract** of an object B if there are morphisms $i : A \longrightarrow B$ and $r : B \longrightarrow A$ such that the following diagram commutes:*

$$\begin{tikzcd} A \arrow[r, "i"] \arrow[dr, equal] & B \arrow[d, "r"] \\ & A \end{tikzcd}$$

Definition 2. *An **untyped λ-model**, $(\mathcal{C}, ?, \mathsf{split}, \mathsf{squash})$, is a cartesian closed category $\mathcal{C}$ with a distinguished object ? and two morphisms $\mathsf{squash} : (? \to ?) \longrightarrow ?$ and $\mathsf{split} : ? \longrightarrow (? \to ?)$ making the object $? \to ?$ a retract of ?.*

Theorem 3 (Scott (1980))**.** *An untyped λ-model is a sound and complete model of the untyped λ-calculus.*

Definition 4. *An object $\mathbb{N}$ of a category $\mathcal{C}$ with a terminal object 1 is a **natural number object (NNO)** if and only if there are morphisms $z : 1 \longrightarrow \mathbb{N}$ and $s : \mathbb{N} \longrightarrow \mathbb{N}$ such that for any other object A and morphisms $z' : 1 \longrightarrow A$ and $s' : A \longrightarrow A$ there is a unique morphism $u : \mathbb{N} \longrightarrow A$ making the following diagram commute:*

$$\begin{tikzcd} 1 \arrow[r, "z"] \arrow[dr, "z'"'] & \mathbb{N} \arrow[r, "s"] \arrow[d, "u"] & \mathbb{N} \arrow[d, "u"] \\ & A \arrow[r, "s'"] & A \end{tikzcd}$$

Definition 5. *A **gradual λ-model**, $(\mathcal{T}, \mathcal{C}, ?, \mathsf{T}, \mathsf{split}, \mathsf{squash}, \mathsf{box}, \mathsf{unbox})$, where $\mathcal{T}$ and $\mathcal{C}$ are cartesian closed categories with NNOS, $(\mathcal{C}, ?, \mathsf{split}, \mathsf{squash})$ is an untyped λ-model, $\mathsf{T} : \mathcal{T} \longrightarrow \mathcal{C}$ is a cartesian closed embedding – a full and faithful cartesian closed functor that is injective on objects and preserves the NNO – and for every object, A, of $\mathcal{T}$ there are morphisms $\mathsf{box}_A : TA \longrightarrow ?$ and $\mathsf{unbox}_A : ? \longrightarrow TA$ making TA a retract of ?.*

3. Grady

References

Joachim Lambek. From lambda calculus to cartesian closed categories. *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 376–402, 1980.

Dana Scott. Relating theories of the lambda-calculus. In *To H.B. Curry: Essays on Combinatory Logic, Lambda-Calculus and Formalism (eds. Hindley and Seldin)*, pages 403–450. Academic Press, 1980.

Jeremy G Siek and Walid Taha. Gradual typing for functional languages. In *Scheme and Functional Programming Workshop*, volume 6, pages 81–92, 2006.

A. The Complete Spec of Grady

$termvar,\ x,\ z$

$index,\ k$

t	$::=$			term
	$\mid$	x		variable
	$\mid$	triv		unit
	$\mid$	$\mathbf{squash}_S$		injection of the retract
	$\mid$	$\mathbf{split}_S$		surjection of the retract
	$\mid$	box_T		generalize to the untyped universe
	$\mid$	unbox_T		specialize the untyped universe to a specifi
	$\mid$	$\lambda x : A.t$		λ-abstraction
	$\mid$	$t_1\ t_2$		function application
	$\mid$	(t_1, t_2)		pair constructor
	$\mid$	$\mathsf{fst}\ t$		first projection
	$\mid$	$\mathsf{snd}\ t$		second projection
	$\mid$	$\mathsf{succ}\ t$		successor function
	$\mid$	0		zero
	$\mid$	(t)	S	

h	$::=$			head-normal forms
	$\mid$	triv		
	$\mid$	$\mathbf{split}_S$		
	$\mid$	$\mathbf{squash}_S$		
	$\mid$	box_T		
	$\mid$	unbox_T		
	$\mid$	$\lambda x : A.t$		
	$\mid$	(t_1, t_2)		
	$\mid$	$\mathsf{fst}\ t$		
	$\mid$	$\mathsf{snd}\ t$		
	$\mid$	$\mathsf{succ}\ t$		
	$\mid$	0		

T	$::=$			terminating types
	$\mid$	1		unit type
	$\mid$	$\mathbf{Nat}$		natural number type
	$\mid$	$T_1 \to T_2$		function type
	$\mid$	$T_1 \times T_2$		cartesian product type
	$\mid$	(T)	S	

S	$::=$			
	$\mid$	$? \to ?$		
	$\mid$	$?\times?$		

A	$::=$			type
	$\mid$	1		unit type
	$\mid$	$\mathbf{Nat}$		natural number type
	$\mid$	$?$		untyped universe
	$\mid$	$A_1 \to A_2$		function type
	$\mid$	$A_1 \times A_2$		cartesian product type
	$\mid$	(A)	S	

Γ	$::=$			typing context
	$\mid$	$\cdot$		empty context
	$\mid$	$\Gamma, x : A$		cons

vd	$::=$			
	$\mid$	$\vdash$		
	$\mid$	$\not\vdash$		

$\boxed{\Gamma vd t : A}$ t has type A in context Γ

$$\frac{x : A \in \Gamma}{\Gamma \vdash x : A}\ \text{VAR}$$

$$\frac{}{\Gamma \vdash \mathsf{box}_T : T \to ?}\ \textsc{Box}$$

$$\frac{}{\Gamma \vdash \mathsf{unbox}_T : ? \to T}\ \textsc{Unbox}$$

$$\frac{}{\Gamma \vdash \mathbf{squash}_S : S \to ?}\ \textsc{squash}$$

$$\frac{}{\Gamma \vdash \mathbf{split}_S : ? \to S}\ \textsc{split}$$

$$\frac{}{\Gamma \vdash \mathsf{triv} : 1}\ \textsc{unit}$$

$$\frac{}{\Gamma \vdash 0 : \mathbf{Nat}}\ \textsc{zero}$$

$$\frac{\Gamma \vdash t : \mathbf{Nat}}{\Gamma \vdash \mathsf{succ}\ t : \mathbf{Nat}}\ \textsc{succ}$$

$$\frac{\Gamma \vdash t_1 : A_1 \quad \Gamma \vdash t_2 : A_2}{\Gamma \vdash (t_1, t_2) : A_1 \times A_2}\ \textsc{pair}$$

$$\frac{\Gamma \vdash t : A_1 \times A_2}{\Gamma \vdash \mathsf{fst}\ t : A_1}\ \textsc{fst}$$

$$\frac{\Gamma \vdash t : A_1 \times A_2}{\Gamma \vdash \mathsf{snd}\ t : A_2}\ \textsc{snd}$$

$$\frac{\Gamma, x : A_1 \vdash t : A_2}{\Gamma \vdash \lambda x : A_1.t : A_1 \to A_2}\ \textsc{lam}$$

$$\frac{\Gamma \vdash t_1 : A_1 \to A_2 \quad \Gamma \vdash t_2 : A_1}{\Gamma \vdash t_1\ t_2 : A_2}\ \textsc{app}$$

$\boxed{\Gamma \vdash t_1 \rightsquigarrow t_2 : A}$ t_1 reduces to t_2 with type A in context Γ

$$\frac{\Gamma \vdash t : T}{\Gamma \vdash \mathsf{unbox}_T\ (\mathsf{box}_T\ t) \rightsquigarrow t : T}\ \textsc{rd\_retract}T$$

$$\frac{\Gamma \not\vdash t : T}{\Gamma \vdash \mathsf{unbox}_T\ (\mathsf{box}_{T'}\ t) \rightsquigarrow \mathsf{wrong} : \mathsf{TypeError}}\ \textsc{rd\_t}W\textsc{rong}$$

$$\frac{h \neq \mathsf{box}_T\ t}{\Gamma \vdash \mathsf{unbox}_T\ h \rightsquigarrow \mathsf{wrong} : \mathsf{TypeError}}\ \textsc{rd\_h}W\textsc{rong}$$

$$\frac{\Gamma \vdash t : S}{\Gamma \vdash \mathbf{split}_S\ (\mathbf{squash}_S\ t) \rightsquigarrow t : S}\ \textsc{rd\_retract}U$$

$$\frac{\Gamma, x : A_1 \vdash t_2 : A_2 \quad \Gamma \vdash t_1 : A_1}{\Gamma \vdash (\lambda x : A_1.t_2)\ t_1 \rightsquigarrow [t_1/x]t_2 : A_2}\ \textsc{rd\_beta}$$

$$\frac{\Gamma \vdash t : A_1 \to A_2 \quad x \notin \mathsf{FV}(t)}{\Gamma \vdash \lambda x : A_1.t\ x \rightsquigarrow t : A_1 \to A_2}\ \textsc{rd\_eta}$$

$$\frac{\Gamma \vdash t_1 : A_1 \quad \Gamma \vdash t_2 : A_2}{\Gamma \vdash \mathsf{fst}\ (t_1, t_2) \rightsquigarrow t_1 : A_1}\ \textsc{rd\_proj}1$$

$$\frac{\Gamma \vdash t_1 : A_1 \quad \Gamma \vdash t_2 : A_2}{\Gamma \vdash \mathsf{snd}\ (t_1, t_2) \rightsquigarrow t_2 : A_2}\ \textsc{rd\_proj}2$$

$$\frac{\Gamma \vdash t : A_1 \times A_2}{\Gamma \vdash (\mathsf{fst}\ t, \mathsf{snd}\ t) \rightsquigarrow t : A_1 \times A_2}\ \textsc{rd\_eta}P$$

$$\frac{\Gamma, x : A_1 \vdash t \rightsquigarrow t' : A_2}{\Gamma \vdash \lambda x : A_1.t \rightsquigarrow \lambda x : A_1.t' : A_1 \to A_2}\ \textsc{rd\_lam}$$

$$\frac{\Gamma \vdash t_1 \rightsquigarrow t_1' : A_1 \to A_2 \quad \Gamma \vdash t_2 : A_1}{\Gamma \vdash t_1\ t_2 \rightsquigarrow t_1'\ t_2 : A_2}\ \textsc{rd\_app}1$$

$$\frac{\Gamma \vdash t_1 : A_1 \to A_2 \quad \Gamma \vdash t_2 \rightsquigarrow t_2' : A_1}{\Gamma \vdash t_1\ t_2 \rightsquigarrow t_1\ t_2' : A_2}\ \textsc{rd\_app}2$$

$$\frac{\Gamma \vdash t \rightsquigarrow t' : A_1 \times A_2}{\Gamma \vdash \mathsf{fst}\ t \rightsquigarrow \mathsf{fst}\ t' : A_1}\ \textsc{rd\_fst}$$

$$\frac{\Gamma \vdash t \rightsquigarrow t' : A_1 \times A_2}{\Gamma \vdash \mathsf{snd}\ t \rightsquigarrow \mathsf{snd}\ t' : A_2}\ \textsc{rd\_snd}$$

$$\frac{\Gamma \vdash t_1 \rightsquigarrow t_1' : A_1 \quad \Gamma \vdash t_2 : A_2}{\Gamma \vdash (t_1, t_2) \rightsquigarrow (t_1', t_2) : A_1 \times A_2}\ \textsc{rd\_pair}1$$

$$\frac{\Gamma \vdash t_1 : A_1 \quad \Gamma \vdash t_2 \rightsquigarrow t_2' : A_2}{\Gamma \vdash (t_1, t_2) \rightsquigarrow (t_1, t_2') : A_1 \times A_2}\ \textsc{rd\_pair}2$$

B. The Complete Spec of $\lambda^{\langle A \rangle}_{\to}$-Calculus

$termvar,\ x,\ z$
$index,\ k$

t	::=		term
	|	x	variable
	|	triv	unit
	|	$\langle A\rangle t$	type cast
	|	$\lambda x : A.t$	λ-abstraction
	|	$t_1\ t_2$	function application
	|	(t_1, t_2)	pair constructor
	|	$\mathsf{fst}\ t$	first projection
	|	$\mathsf{snd}\ t$	second projection
	|	$\mathsf{succ}\ t$	successor function
	|	0	zero
	|	(t) S	

$A,\ B,\ C$	::=		type
	|	1	unit type
	|	$\mathbb{N}$	natural number type
	|	$?$	untyped universe
	|	$A_1 \to A_2$	function type
	|	$A_1 \times A_2$	cartesian product type
	|	(A) S	

Γ	::=		typing context
	|	$\cdot$	empty context
	|	$x : A$	cons

vd	::=	
	|	$\vdash$
	|	$\nvdash$

$\boxed{A \sim B}$ A is consistent with B

$$\frac{}{A \sim A}\ \text{REFL}$$

$$\frac{}{A \sim ?}\ \text{BOX}$$

$$\frac{}{? \sim A}\ \text{UNBOX}$$

$$\frac{A_1 \sim A_2 \quad B_1 \sim B_2}{A_1 \to B_1 \sim A_2 \to B_2}\ \text{ARROW}$$

$$\frac{A_1 \sim A_2 \quad B_1 \sim B_2}{A_1 \times B_1 \sim A_2 \times B_2}\ \text{PROD}$$

$\boxed{\Gamma vd\, t : A}$ t has type A in context Γ

$$\frac{x : A \in \Gamma}{\Gamma \vdash x : A}\ \text{VAR}$$

$$\frac{}{\Gamma \vdash \mathsf{triv} : 1}\ \text{UNIT}$$

$$\frac{}{\Gamma \vdash 0 : \mathbb{N}}\ \text{ZERO}$$

$$\frac{\Gamma \vdash t : \mathbb{N}}{\Gamma \vdash \mathsf{succ}\ t : \mathbb{N}}\ \text{SUCC}$$

$$\frac{\Gamma \vdash t_1 : A_1 \quad \Gamma \vdash t_2 : A_2}{\Gamma \vdash (t_1, t_2) : A_1 \times A_2}\ \text{PAIR}$$

$$\frac{\Gamma \vdash t : A_1 \times A_2}{\Gamma \vdash \mathsf{fst}\ t : A_1}\ \text{FST}$$

$$\frac{\Gamma \vdash t : A_1 \times A_2}{\Gamma \vdash \mathsf{snd}\ t : A_2}\ \text{SND}$$

$$\frac{\Gamma, x : A_1 \vdash t : A_2}{\Gamma \vdash \lambda x : A_1.t : A_1 \to A_2}\ \text{LAM}$$

$$\frac{\Gamma \vdash t_1 : A_1 \to A_2 \quad \Gamma \vdash t_2 : A_1}{\Gamma \vdash t_1\ t_2 : A_2}\ \text{APP}$$

$$\frac{\Gamma \vdash t : A \quad A \sim B}{\Gamma \vdash \langle B\rangle t : B}\ \text{CAST}$$