Towards a theory of type structure

John C. Reynolds
presented by Hyun-Goo Kang

10/09/2000

□ Abstract

- Basic principle over type structure of PL has been the subject of an active development and still controversy

- Formalize basic principle over type structure of PL
  - introduce an extension of **typed lambda calculus** which permit **user defined types** and **polymorphic function** : System F?
  - show the semantics of this language and show that it satisfies a representation theorem which embodies our notion of "correct" type structure

□ Introduction

What we want show/suggest in this paper

- the meaning of a syntactically valid program in a type "correct" language should never depend on a particular representations used to implement its primitive / user-defined / polymorphic function type (**representation independence**)

  Example)

  if every <u>integer</u> input to M represents the corresponding <u>integer</u> input to M', and M interprets every primitive operation (involving integers) in a way which represents the interpretation of M', we expect output of M should represent the corresponding output of M'

- we'd like to to define polymorphic function like *sort*, and to have each call of the function syntactically checked to insure that it's type correct for some t , which was impossible
  $\rightarrow$ suggest a solution

□ An Illustrative Language(F)

Kwang's 520 course slide

□ Notational Preliminaries

- $S \times S'$: cartesian product of S and S'
- $S \Rightarrow S'$: set of functions from S to S'
- $S \rightarrow S'$: set of continuous functions from S to S' (D.Scott)
- $\Pi_{x \in S} F(x)$: set of functions f such that $Dom(f)$ is S, and for each $x \in S$, $f(x) \in F(x)$
- [f|x|x'] : λy∈S.if y=x then x' else f(y) (ex. appending a map to an environment)

□ Syntax

An Illustrative Language

$$e \;:=\; x \mid \lambda x \in t.e \mid (e_1 \; e_2) \mid \Lambda t.e \mid e[w]$$

Syntax of type expression

consider countably infinite sets : the set T of type variables and the set V of normal variables, then W, the set of type expressions is the minimal set satisfying

1. if $t \in T$ then $t \in W$
2. if $w_1, w_2 \in W$ then $(w_1 \rightarrow w_2) \in W$
3. if $t \in T$ and $w \in W$ then $(\Delta t.w) \in W$

Preliminaries

- $w = w'$ if they are α-convertible (unifiable)
- $w_1|_t^{w_2}$: substitution

Every normal expression has an explicit type

if we have types for all free variables, then r should be uniquely typable like followings

R_{Qw} is minimal family of sets satisfying

- if $Q \in V \Rightarrow W$ and $x \in V$ then,

  $x \in R_{QQ(x)}$

- if $Q \in V \Rightarrow W$, $w_1, w_1', w_2 \in W$, $w_1 = w_1'$, $r_1 \in R_{Q(w_1 \to w_2)}$ and $r_2 \in R_{Qw_1'}$ then,

  $(r_1\ r_2) \in R_{Qw_2}$

- if $Q \in V \Rightarrow W$, $w_1, w_2 \in W$, $x \in V$, and $r \in R_{[Q|x|w_1]}$ then,

  $(\lambda x \in w_1.r) \in R_{Q(w_1 \to w_2)}$

- if $Q \in V \Rightarrow W$, $w_1, w_2 \in W$, $t \in T$, and $r \in R_{Q(\Delta.w_1)}$ then,

  $(r\ [w_2]) \in R_{Q(w_1|_t^{w_2})}$

- if $Q \in V \Rightarrow W$, $w \in W$, $t \in T$, $r \in R_{QW}$, and t doesn't occur free in $Q(x)$ for any x which occur free in r then,

  $(\Lambda t.r) \in R_{Q(\Delta t.w)}$

Result

Now we have seen how to classify given normal expression by its type(assign a type to given normal expression using our type expression), then what's the meaning(semantics) of type expression? is it correct?

□ Semantics

Effect of a type expression

produce a Scott domain (given an assignment of a domain to each free type variable occurring in the type expression)

thus the meaning of type expression B is given by a function

$$B \ \in \ W \Rightarrow D^T \Rightarrow \mathbf{D}$$

Specification of B

- $B[t](\overline{D}) = \overline{D}(t)$
- $B[w_1 \to w_2](\overline{D}) = arrow(B[w_1](\overline{D}), B[w_2](\overline{D}))$
  where $arrow \in (\mathbf{D} \times \mathbf{D}) \Rightarrow \mathbf{D}$ satisfies
  $arrow(D_1, D_2) = D_1 \to D_2$
- $B[\Delta t.w](\overline{D}) = delta(\lambda D \in \mathbf{D}.B[w][\overline{D}|t|D])$
  where $delta \ \in \ (\mathbf{D} \Rightarrow \mathbf{D}) \Rightarrow \mathbf{D}$ satisfies
  $delta(\theta) \subseteq \Pi_{D \in \mathbf{D}} \theta(D)$

A useful extension

$B[w_1|_t^{w_2}](\overline{D}) = B[w_1][\overline{D}|t|B[w_2](\overline{D})]$

Effect of a normal expression

produce a value (given an assignment of domains to its free type variables and an assignment of values to its free normal variables(environment))

$r \in R_{Qw}$ must only accept environments which map each variable x into a member of the domain $B[Q(x)](\overline{D})$, and r must produce a member of the domain $B[w](\overline{D})$ $\rightarrow$ "no typing error"

then, the meaning of normal expressions in R_{Qw} is given by a function

$$M_{Qw} \;\in\; R_{Qw} \Rightarrow \Pi_{\overline{D} \in \mathbf{D}^{T}}(Env_Q(\overline{D}) \rightarrow B[w](\overline{D}))$$

where

$$Env_Q(\overline{D}) = \Pi_{x \in V} B[Q(x)](\overline{D})$$

Specification of M_{Qw}

- $M_{QQ(x)}[x](\overline{D})(e) = e(x)$
- $M_{Q(w_1 \rightarrow w_2)}[\lambda x \in w_1.r](\overline{D})(e)$
  $= \lambda a \in B[w_1](\overline{D}).M_{[Q|x|w_1]w_2}[r](\overline{D})[e|x|a]$
- $M_{Qw_2}[r_1\ r_2](\overline{D})(e)$
  $= (M_{Q(w_1 \rightarrow w_2)}[r_1](\overline{D})(e))(M_{Qw'_1}[r_2](\overline{D})(e))$
- $M_{Q(w_1|_t^{w_2})}[r[w_2]](\overline{D})(e)$
  $= M_{Q(\Delta t.w_1)}[r](\overline{D})(e)(B[w_2](\overline{D}))$
- $M_{Q(\Delta t.w)}[\Lambda t.r](\overline{D})(e)$
  $= \lambda D \in \mathbf{D}.M_{Qw}[r][\overline{D}|t|D](e)$

□ Representation

What we mean by representation

- representation between D and D' : $rep(D, D')$

  $$rep(D, D') = \{\langle \phi, \psi \rangle \mid \begin{array}{l} \phi \in D \to D',\ \psi \in D' \to D, \\ \psi \cdot \phi \sqsupseteq I_D,\ \phi \cdot \psi \sqsubseteq I_{D'} \end{array} \}$$

- For $x \in D$, $x' \in D'$, and $\rho = \langle \phi, \psi \rangle \in rep(D, D')$, we write

  $$\rho \ : \ x \ \mapsto \ x'$$

  and say that x represents x' according to ρ, iff

  $$x \sqsubseteq \psi(x') \text{ or equivalently } \phi(x) \sqsubseteq x'$$

An obvious and useful extension of definition

For $\overline{D}, \overline{D'} \in \mathbf{D}^T$, we define

$$rep(\overline{D}, \overline{D'}) = \Pi_{t \in T}\ rep(\overline{D}(t), \overline{D'}(t))$$

□ Representation Theorem

A preliminary version of representation theorem

Consider set of normal expression R_{Qw}, and suppose $\overline{D}, \overline{D'} \in \mathbf{D}^T$, $\overline{\rho} \in rep(\overline{D}, \overline{D'})$

Moreover, suppose that e and e' are env such that $(\forall x \in Dom(env))e(x)$ represent $e'(x)$ (i.e. $\overline{\rho}(Q(x))$)

Then, we expect that the value of any $r \in R_{Qw}$ (when evaluated with respect to $\overline{D}$ and e) should represent the value of the same normal expression (when evaluated with respect to $\overline{D'}$ and e'), according to the relevant representation. (i.e. $\overline{\rho}(w)$)

More Formally

Let $Q \in V \Rightarrow W$, $w \in W$, $\overline{D}, \overline{D'} \in \mathbf{D}^T$, $\overline{\rho} \in rep(\overline{D}, \overline{D'})$, $e \in Env_Q(\overline{D})$ and $e' \in Env_Q(\overline{D'})$.

if

$$(\forall x \in V)\overline{\rho}(Q(x)) : e(x) \mapsto e'(x)$$

then

$$(\forall r \in R_{Qw})\overline{\rho}(w) : M_{Qw}[r](\overline{D})(e) \mapsto M_{Qw}[r](\overline{D'})(e')$$

Problem

representation $\overline{\rho}(Q(x))$ and $\overline{\rho}(w)$ are not fully defined.

once we have representation for integer and real, then we expect that this choice would determine a representation for any other type expression constructed from integer and real

In brief, not only must $B[w]$ map an assignment of domains to type variables into a domain, but it must also map an assignment of representation into a representation. More formally

$$\rho \in rep(\overline{D}, \overline{D'})$$

should imply

$$B[w](\overline{\rho}) \in rep(B[w](\overline{D}), B[w](\overline{D'}))$$

A correct formulation of the representation theorem

Let $Q \in V \Rightarrow W$, $w \in W$, $\overline{D}, \overline{D'} \in \mathbf{D}^T$, $\overline{\rho} \in rep(\overline{D}, \overline{D'})$, $e \in Env_Q(\overline{D})$ and $e' \in Env_Q(\overline{D'})$.

if

$$(\forall x \in V) B[Q(x)](\overline{\rho}) : e(x) \mapsto e'(x)$$

then

$$(\forall r \in R_{Qw}) B[w](\overline{\rho}) : M_{Qw}[r](\overline{D})(e) \mapsto M_{Qw}[r](\overline{D'})(e')$$

now, it's time to extend the semantics of type expression...

□ Full Semantics of type-exp

Note1

combination of domains and representations forms a category(category of types : C)

1. the set of objects is **D**

2. the set of morphisms from D to D' is $rep(D, D')$

3. composition is given by
   $\langle \phi', \psi' \rangle \cdot \langle \phi, \psi \rangle = \langle \phi' \cdot \phi, \psi \cdot \psi' \rangle$

4. identity for $\overline{D}$ is $I_D = \langle I_D, I_D \rangle$

Note2

two further categories by standard constructions of category theory

- category C^T
  1. the set of objects is $\mathbf{D}^T$
  2. the set of morphisms from $\overline{D}$ to $\overline{D'}$ is $rep(\overline{D}, \overline{D'})$
  3. composition is given by
     $(\overline{\rho'} \cdot \overline{\rho})(t) = \overline{\rho'}(t) \cdot \overline{\rho}(t)$
  4. identity for $\overline{D}$ is $I_D(t) = I_{\overline{D}(t)}$

- category $Funct(C, C)$
  1. objects are functors from C to C
  2. morphisms from θ to θ' are natural transformations from θ to θ'
     i.e. the functions $\eta \in \Pi_{D \in \mathbf{D}} rep(\theta(D), \theta'(D))$ such that, for all $D, D' \in \mathbf{D}$ and $\rho \in rep(D, D')$,
     $$\theta'(\rho) \cdot \eta(D) = \eta(D') \cdot \theta(\rho)$$
  3. composition is given by
     $(\eta' \cdot \eta)(D) = \eta(D') \cdot \eta(D)$
  4. identity for $\overline{\theta}$ is $I_\theta(D) = I_{\theta(D)}$

The meaning $B[w]$ of a type expression w

1. must map the objects of C^T into the objects of C and the morphisms of C^T into the morphisms of C
2. must have

   $\rho \in rep(\overline{D}, \overline{D'})$ implies $B[w](\overline{\rho}) \in rep(B[w](\overline{D}), B[w](\overline{D'}))$

Indeed, by pursuing the analogy of categories with sets and functors with functions, we can induce the main structure of the definition of $B[w]$

The syntactic definition of W

- $B[t](\overline{D}) = \overline{D}(t)$
  $B[t](\overline{\rho}) = \overline{\rho}(t)$

- $B[w_1 \to w_2](\overline{D}) = arrow(B[w_1](\overline{D}), B[w_2](\overline{D}))$
  $B[w_1 \to w_2](\overline{\rho}) = arrow(B[w_1](\overline{\rho}), B[w_2](\overline{\rho}))$
  where $arrow$ is a bifunctor from $C \times C$ into C

- $B[\Delta t.w] = delta \cdot abstract$
  where $abstract$ is the functor from C^T into $Funct(C, C)$ such that

$$abstract(\overline{D})(D) = B[w][\overline{D}|t|D]$$
$$abstract(\overline{D})(\rho) = B[w][I_{\overline{D}}|t|\rho]$$
$$abstract(\overline{\rho})(D) = B[w][\overline{\rho}|t|I_D]$$

  and $delta$ is a functor from $Funct(C, C)$ into C

now, it's time to define the functors arrow and delta...

□ The functor arrow

Definition of the functor arrow

For all $D_1, D_2 \in \mathbf{D}$,

$$arrow(D_1, D_2) = D_1 \to D_2$$

For all $\langle \phi_1, \psi_1 \rangle \in rep(D_1, D'_1)$ and $\langle \phi_2, \psi_2 \rangle \in rep(D_2, D'_2)$

$$\begin{aligned} &arrow(\langle \phi_1, \psi_1 \rangle, \langle \phi_2, \psi_2 \rangle) = \\ &\quad \langle \lambda f \in D_1 \to D_2 . \phi_2 \cdot f \cdot \psi_1, \lambda f \in D'_1 \to D'_2 . \psi_2 \cdot f \cdot \phi_1, \rangle \end{aligned}$$

Lemma 1

by the definition of arrow and the properties of representation, we get the following lemma

Let $f \in D_1 \to D_2, f' \in D'_1 \to D'_2$, $\rho_1 \in rep(D_1, D'_1)$, and $\rho_2 \in rep(D_2, D'_2)$, then,

$$arrow(\rho_1, \rho_2) : f \mapsto f'$$

iff, for all $x \in D_1$ and $x' \in D'_1$

$$\rho_1 : x \mapsto x' \text{ implies } \rho_2 : f(x) \mapsto f'(x')$$

Lemma 2

lemma 1, with the definition of B, gives the following lemma:

Let $w_1, w_2 \in W$, $\overline{\rho} \in rep(\overline{D}, \overline{D'})$, $f \in B[w_1 \to w_2](\overline{D})$, and $f' \in B[w_1 \to w_2](\overline{D'})$, then

$$B[w_1 \to w_2](\overline{\rho}) = f \mapsto f'$$

iff, for all $x \in B[w_1](\overline{D})$ and $x' \in B[w_1](\overline{D'})$,

$$B[w_1](\overline{\rho}) : x \mapsto x' \text{ implies } B[w_2](\overline{\rho}) : f(x) \mapsto f'(x')$$

$\square$ The functor delta

Definition of the functor delta

For all functors θ from C to C, $delta(\theta)$ is the complete lattice with elements

$\{f | f \in \Pi_{D \in \mathbf{D}}\ \theta(D)$ and $(\forall D, D' \in \mathbf{D})(\forall \rho \in rep(D, D'))\theta(\rho) : f(D) \mapsto f(D')\}$

with the partial ordering $f \sqsubseteq g$ iff $(\forall D \in \mathbf{D}) f(D) \sqsubseteq_{\theta(D)} g(D)$

For all natural transformation η from θ to θ'

$$\begin{aligned}\langle\ & \lambda f \in delta(\theta).\lambda D \in \mathbf{D}.[\eta(D)]_1(f(D)),\\ & \quad \lambda f \in delta(\theta').\lambda D \in \mathbf{D}.[\eta(D)]_2(f(D))\ \rangle\end{aligned}$$

Lemma 3

by the definition of delta and the properties of representation, we get the following lemma

Let η be a natural transformation from θ to θ', $f \in delta(\theta)$ and $f' \in delta(\theta')$. Then,

$$delta(\eta) : f \mapsto f'$$

iff, for all $D, D' \in \mathbf{D}$, and $\rho \in rep(D, D')$,

$$\eta(D') \cdot \theta(\rho) : f(D) \mapsto f'(D')$$

Lemma 4

lemma 3, with the definition of B, gives the following lemma:

Let $t \in T, w \in W$, $\overline{\rho} \in rep(\overline{D}, \overline{D'})$, $f \in B[\Delta t.w](\overline{D})$, and $f' \in B[\Delta t.w](\overline{D'})$. Then

$$B[\Delta t.w](\overline{\rho}) = f \mapsto f'$$

iff, for all $d, D' \in \mathbf{D}$ and $\rho \in rep(D, D')$,

$$B[w][\overline{\rho}|t|\rho] : f(D) \mapsto f'(D')$$

now, from the final lemmas(2,4) obtained about arrow and delta, the representation theorem can by proved by structural induction on r

□ Conlusion

- we have explored our illustrative language semantically rather than syntactically, i.e., we have provided it with a mathematical meaning instead of investigating the syntactic consequences of reducibility.