Realisability methods of proof and semantics with application to expansion

First Year Examination

Supervisors : Professor Fairouz Kamareddine and Doctor Joe B. Wells
Student : Vincent Rahli

ULTRA group, MACS, Heriot Watt University

August 2, 2007

History

- Existence of paradoxes.
- Functions can be applied to any function.
- Formalization of the concept of type by Russel [Rus08] to restrict application of functions.
- Formalization of Mathematics: design of logical systems.

λ-calculus

- Design of the λ-calculus by Church in 1932 [Chu32].
- System to investigate functions.
- Syntax: $M \in \Lambda ::= x \mid \lambda x.M \mid M_1 M_2$
  such that $x \in \mathcal{V}$ (term variables).
- Rule of conversion:
  - $(\lambda x.M)N \to_\beta M[x := N]$
  - $\lambda x.Mx \to_\eta M$ where $x \notin FV(M)$
- Consistency: Church-Rosser theorem
- Church's thesis: "Effectively calculable functions from positive integers to positive integers are just those definable in the λ-calculus".
- Models: model D_∞ of Scott; λ-models of Hindley and Longo.

Typed λ-calculus

- Simply typed λ-calculus.

$$\sigma \in \mathsf{Type} ::= \alpha \mid \sigma_1 \to \sigma_2$$

  where $(\alpha \in)\mathcal{A}$ is a set of type variables.
- Polymorphism: intersection types.

$$\sigma \in \mathsf{Type} ::= \alpha \mid \sigma_1 \to \sigma_2 \mid \sigma_1 \cap \sigma_2$$

- λ-cube of Barendregt: 8 type systems with $\neq$ expressiveness.

Example of a type systems: $\lambda\cap^{\Omega}$ and D

(*ref*)	$\sigma \leq \sigma$
(*tr*)	$\sigma \leq \tau \wedge \tau \leq \rho \Rightarrow \sigma \leq \rho$
(in_L)	$\sigma \cap \tau \leq \sigma$
(in_R)	$\sigma \cap \tau \leq \tau$
($\rightarrow$-$\cap$)	$(\sigma \rightarrow \tau) \cap (\sigma \rightarrow \rho) \leq \sigma \rightarrow (\tau \cap \rho)$
(mon')	$\sigma \leq \tau \wedge \sigma \leq \rho \Rightarrow \sigma \leq \tau \cap \rho$
(*mon*)	$\sigma \leq \sigma' \wedge \tau \leq \tau' \Rightarrow \sigma \cap \tau \leq \sigma' \cap \tau'$
($\rightarrow$-η)	$\sigma \leq \sigma' \wedge \tau' \leq \tau \Rightarrow \sigma' \rightarrow \tau' \leq \sigma \rightarrow \tau$
(Ω)	$\sigma \leq \Omega$
(Ω'-*lazy*)	$\sigma \rightarrow \Omega \leq \Omega \rightarrow \Omega$

Figure: Ordering axioms on types

Example of a type system: $\lambda\cap^{\Omega}$ and D

$$\begin{array}{lcl} \Gamma \vdash M : \sigma & : & \text{typing judgment} \\ (\Gamma, \sigma) & : & (\text{typing of } M) \end{array}$$

$$\frac{}{\Gamma, x : \sigma \vdash x : \sigma}\ (ax) \qquad \frac{}{\Gamma \vdash M : \Omega}\ (\Omega)$$

$$\frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}\ (\to_E) \qquad \frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau}\ (\to_I)$$

$$\frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash M : \tau}{\Gamma \vdash M : \sigma \cap \tau}\ (\cap_I) \qquad \frac{\Gamma \vdash M : \sigma \quad \sigma \leq^{\nabla} \tau}{\Gamma \vdash M : \tau}\ (\leq^{\nabla})$$

Figure: Typing rules

Example of a type system: $\lambda\cap^\Omega$ and D

- $\sigma \in \text{Type}^{\lambda\cap^\Omega} ::= \alpha \mid \sigma_1 \to \sigma_2 \mid \sigma_1 \cap \sigma_2 \mid \Omega$.
- $\mathcal{B}^{\lambda\cap^\Omega} = \{\Gamma = \{x : \sigma \mid x \in \mathcal{V}, \sigma \in \text{Type}^{\lambda\cap^\Omega}\} \mid \forall x : \sigma, y : \tau \in \Gamma,\ \text{if } \sigma \neq \tau \text{ then } x \neq y\}$.
- $\nabla = \{(\mathit{ref}), (\mathit{tr}), (\mathit{in}_L), (\mathit{in}_R), (\to\text{-}\cap), (\mathit{mon}'), (\mathit{mon}), (\to\text{-}\eta), (\Omega), (\Omega' - \mathit{lazy})\}$.
- The relation $\leq^\nabla$ is defined on types $\text{Type}^{\lambda\cap^\Omega}$ and the set of axioms ∇. The equivalence relation is defined by: $\sigma \sim^\nabla \tau \iff \sigma \leq^\nabla \tau \wedge \tau \leq^\nabla \sigma$.
- $\lambda\cap^\Omega = \langle \Lambda, \text{Type}^{\lambda\cap^\Omega}, \vdash \rangle$ such that $\vdash$ is type derivability relation on $\mathcal{B}^{\lambda\cap^\Omega}$, Λ and $\text{Type}^{\lambda\cap^\Omega}$ generated using the typing rules of Figure 2.

- $\sigma \in \text{Type}^D ::= \alpha \mid \sigma_1 \to \sigma_2 \mid \sigma_1 \cap \sigma_2$.
- $\mathcal{B}^D = \{\Gamma = \{x : \sigma \mid x \in \mathcal{V}, \sigma \in \text{Type}^D\} \mid \forall x : \sigma, y : \tau \in \Gamma,\ \text{if } \sigma \neq \tau \text{ then } x \neq y\}$.
- $\nabla = \{(\mathit{ref}), (\mathit{tr}), (\mathit{in}_L), (\mathit{in}_R)\}$.
- The relation $\leq^{\nabla}$ is defined on types Type^D and the set of axioms ∇. The equivalence relation is defined by:
  $\sigma \sim^{\nabla} \tau \iff \sigma \leq^{\nabla} \tau \wedge \tau \leq^{\nabla} \sigma$.
- $\lambda\cap^{\Omega} = \langle \Lambda, \text{Type}^D, \vdash \rangle$ such that $\vdash$ is type derivability relation on $\mathcal{B}^D$, Λ and Type^D generated using the typing rules of Figure 2 except (Ω).

Properties of the λ-calculus

- Church-Rosser property:
  R: binary relation on Λ.
  R is Church-Rosser iff $MRM_1 \wedge MRM_2 \Rightarrow \exists M_3, M_1 R M_3 \wedge M_2 R M_3$.
  - method of parallel reductions
  - method of finiteness of developments

  Let L be a set of terms and $\rightarrow$ a reduction relation:

  $$\mathrm{CR}^L_{\rightarrow} = \{M \in L \mid M \rightarrow M_1 \wedge M \rightarrow M_2 \Rightarrow \exists M_3, M_1 \rightarrow M_3 \wedge M_2 \rightarrow M_3\}$$

  $\mathrm{CR}^L = \mathrm{CR}^L_{\rightarrow_\beta}$ and $\mathrm{CR}_{\rightarrow} = \mathrm{CR}^{\Lambda}_{\rightarrow}$.
- Normalization properties.
  - Standardization
  - Developments

  $\mathrm{SN} = \{M \in \Lambda \mid$ each β-reduction from M is finite$\}$.
  $\mathrm{WN} = \{M \in \Lambda \mid \exists$ aβ-reduction from M which is finite$\}$.

Goal: generalization and simplification of proof methods.

Properties of Type Systems

Properties that we might want to be held by a type system:

- Decidability of the type inference.
- Decidability of the type checking.
- Principal typing.
- Subject Reduction/Expansion.
- Strong Normalization.

Proof Methods

There exists different methods to prove properties of the λ-calculus or of type systems but not an universal one.

- $\neq$ properties $\Rightarrow$ $\neq$ proof methods.
- changes of framework $\Rightarrow$ all the proofs need to be reproved.
- A method may work in a framework but not in another one $\Rightarrow$ Introduction of new methods, new concept.
- Expansion: new concept to calculate typing from a principal one in intersection type systems
- Reducibility: general proof methods to prove properties of the λ-calculus.

Expansion

- Calculate a type of a term from its principal one, in a intersection type system, need more than substitution [CDCV80].
- Introduction of the mechanism of expansion.
- Development of the mechanism of expansion [KW99]: expansion variables.
- Goal: find a semantics of an intersection type system with expansion mechanism.

Reducibility

Let $\mathcal{P} \subseteq \Lambda$.

The reducibility method is based on realisability semantics.

- Interpretation of types by sets of terms such that they (mostly) turn to be subsets of $\mathcal{P}$.
- Proof of a soundness result.

Semantics of Type Systems

- Interpretation of the logical contents of a type system (Curry-Howard isomorphism).
- Study and characterization of legal types.
- Verification of the intended behavior of a type system.

A semantics is complete w.r.t. a type system if: a typing judgment is true in the semantics if and only if it is derivable in the type system [Hin83] (soundness: if direction).

Realisability semantics: types are interpreted by realizers (functions).

Reducibility in [GL02]

Let $\mathcal{P} \subseteq \Lambda$.

- Definition of an interpretation of the types in $\mathrm{Type}^{\lambda\cap^{\Omega}}$:
  - $[\![\alpha]\!] = \mathcal{P}$,
  - $[\![\sigma \to \tau]\!] = \{M \in \mathcal{P} \mid \forall N \in [\![\sigma]\!], MN \in [\![\tau]\!]\}$,
  - $[\![\sigma \cap \tau]\!] = [\![\sigma]\!] \cap [\![\tau]\!]$,
  - $[\![\Omega]\!] = \Lambda$
- Definition of closure properties:
  - $\mathrm{VAR}(\mathcal{P}, \mathcal{X})$: $\forall x \in \mathcal{V}, \forall n \geq 0, \forall M_1, \ldots, M_n \in \mathcal{P}, xM_1 \ldots M_n \in \mathcal{X}$.
  - $\mathrm{SAT}(\mathcal{P}, \mathcal{X})$: $\forall M, N \in \Lambda, \forall n \geq 0, \forall M_1, \ldots, M_n \in \mathcal{P}$, $M[x := N]M_1 \ldots M_n \in \mathcal{X} \Rightarrow (\lambda x.M)NM_1 \ldots M_n \in \mathcal{X}$.
  - $\mathrm{CLO}(\mathcal{P}, \mathcal{X})$: $\forall M \in \mathcal{X}, \lambda x.M \in \mathcal{P}$.
  - $\mathrm{INV}(\mathcal{P})$: $\forall M \in \Lambda, M \in \mathcal{P} \iff \lambda x.M \in \mathcal{P}$.

Reducibility in [GL02]

- Proof of a soundness result:
  $(\forall \sigma \in \mathsf{Type}^{\lambda\cap^{\Omega}}, \mathsf{VAR}(\mathcal{P}, [\![\sigma]\!]) \wedge \mathsf{SAT}(\mathcal{P}, [\![\sigma]\!]) \wedge \mathsf{CLO}(\mathcal{P}, [\![\sigma]\!])) \Rightarrow (\forall \sigma \in \mathsf{Type}^{\lambda\cap^{\Omega}}, \Gamma \vdash M : \sigma \wedge \sigma \not\sim^{\nabla} \Omega \Rightarrow M \in [\![\sigma]\!] \subseteq \mathcal{P})$.
- "Generalization" of hypotheses:
  $(\mathsf{VAR}(\mathcal{P}, \mathcal{P}) \wedge \mathsf{SAT}(\mathcal{P}, \mathcal{P}) \wedge \mathsf{CLO}(\mathcal{P}, \mathcal{P})) \Rightarrow (\forall \sigma \in \mathsf{Type}^{\lambda\cap^{\Omega}}, \Gamma \vdash M : \sigma \wedge \sigma \not\sim^{\nabla} \Omega \wedge (\forall \tau \in \mathsf{Type}, \tau \not\sim^{\nabla} \Omega \Rightarrow \sigma \not\sim^{\nabla} \Omega \rightarrow \tau) \Rightarrow M \in \mathcal{P})$.
- Proof of the proof method:
  $\mathsf{VAR}(\mathcal{P}, \mathcal{P}) \wedge \mathsf{SAT}(\mathcal{P}, \mathcal{P}) \wedge \mathsf{INV}(\mathcal{P}) \Rightarrow \mathcal{P} = \Lambda$.
- Proof of one of the principal result: $\mathsf{VAR}(\mathsf{CR}, \mathsf{CR})$, $\mathsf{SAT}(\mathsf{CR}, \mathsf{CR})$ and $\mathsf{INV}(\mathsf{CR})$ are true. Hence, $\Lambda = \mathsf{CR}$.

We establish that reducibility in [GL02] fails

Our counter example that [GL02] fails:

- VAR(WN, WN), SAT(WN, WN) and INV(WN) are satisfied,
- but $\Lambda \neq$ WN.

Our solution to repair [GL02]:

-
  - $\sigma^+ \in \mathrm{Type}^{\lambda\cap^\Omega +} = \{\sigma \in \mathrm{Type}^{\lambda\cap^\Omega} \mid \sigma \sim^\nabla \Omega\}$.
  - $\sigma^- \in \mathrm{Type}^{\lambda\cap^\Omega -} = \{\sigma \in \mathrm{Type}^{\lambda\cap^\Omega} \mid \sigma \not\sim^\nabla \Omega\}$.
  - $\sigma \in S_1 ::= \alpha \mid \sigma_1^+ \to \sigma_2^+ \mid \sigma^- \to \sigma \mid \sigma_1 \cap \sigma_2$.
- $(\mathrm{VAR}(\mathcal{P},\mathcal{P}) \wedge \mathrm{SAT}(\mathcal{P},\mathcal{P}) \wedge \mathrm{CLO}(\mathcal{P},\mathcal{P})) \Rightarrow (\forall \sigma \in \mathrm{Type}^{\lambda\cap^\Omega}, \Gamma \vdash M : \sigma \wedge \sigma \not\sim^\nabla \Omega \wedge (\sigma = \tau \to \rho \Rightarrow \tau, \rho \in S_1 \wedge \rho \not\sim^\nabla \Omega) \Rightarrow M \in \mathcal{P})$.

Reducibility in [KS07]

- Definition of an interpretation of the types in Type^D:
  - $[\![\alpha]\!] = \mathsf{CR}$,
  - $[\![\sigma \to \tau]\!] = \{M \in \mathsf{CR} \mid \forall N \in [\![\sigma]\!], MN \in [\![\tau]\!]\}$,
  - $[\![\sigma \cap \tau]\!] = [\![\sigma]\!] \cap [\![\tau]\!]$.
- Proof of a soundness result:
  $\forall \sigma \in \mathsf{Type}^D, \Gamma \vdash M : \sigma \Rightarrow M \in [\![\sigma]\!] \subseteq \mathsf{CR}$.

Reducibility in [KS07]

Confluence of developments:

- Let $x \in \mathcal{V}' = \mathcal{V} \setminus \{c\}$, then
  $M \in \Lambda_c ::= x \mid \lambda x.M \mid cM_1M_2 \mid (\lambda x.M_1)M_2$
- $\forall M \in \Lambda_c$, M is typable in the system D.
- A one step development of $(M, \mathcal{F})$ is a pair $(M', \mathcal{F}')$ such that $M \to_\beta M'$, $\mathcal{F}$ is a set of redexes in M and $\mathcal{F}'$ is the set of residuals of $\mathcal{F}$ in M'.
- Development $=$ reflexive and transitive closure of a one step development.
- Proof of the confluence of developments using the confluence of Λ_c.

Parallel reduction:

- $M \to_1 M' \iff \exists \mathcal{F}, \mathcal{F}', (M, \mathcal{F}) \to_\beta (M', \mathcal{F}')$.
- $\to_\beta^* = \to_1^*$.
- $\Lambda = \mathrm{CR}$.

Our extension of reducibility of [KS07]

We have:

- Adapted the method to βI: $\Lambda_I = \mathsf{CR}^{\Lambda_I}$.
- Extended and generalized the method to $\beta\eta$: $\Lambda = \mathsf{CR}_{\beta\eta}$.
- Formalized the various steps of the proof so it can be further extended.

Semantics of an intersection type system with expansion

Syntax:

- a set of expansion variables $e \in$ E-var.
- a rule for introduction of e:

$$\frac{\Gamma \vdash M : \sigma}{e\Gamma \vdash M : e\sigma}\ e_I$$

Goal: Find a complete semantics for an intersection type system with expansion mechanism.

Intermediate Goal: Find a complete semantics for an intersection type system with expansion variables.

Semantics of an intersection type system with expansion

We have 2 results so far:

- [KNRW06] Complete semantics for an intersection type system
  - with only one expansion variable
  - without the expansion mechanism
- [KNRW07] Complete semantics for an intersection type system
  - with an infinity of expansion variables
  - without the expansion mechanism

Use of the λ-calculus with some decorations.

Future work

- Find a complete semantics for an intersection type system with the expansion mechanism. (September 2007 – September 2008).
  - Realisability semantics
  - Denotational semantics
  - Operational semantics
- Generalization of proof methods of properties of the λ-calculus using reducibility. (September 2008 – July 2009).
  - In the framework of the λ-cube
  - In the framework of expansion
- Generalization and extension of PTSs. (September 2007 – July 2009).
  - Capture of the Strong Normalization property.
  - Extension with: Unified binder, Type inclusion, Π-application and abbreviations.

Mario Coppo, Mariangiola Dezani-Ciancaglini, and Betti Venneri.
Principal type schemes and λ-calculus semantic.
1980.

Alonzo Church.
A set of postulates for the foundation of logic.
The Annals of Mathematics, 33(2):346–366, 1932.

Silvia Ghilezan and Silvia Likavec.
Reducibility: A ubiquitous method in lambda calculus with intersection types.
Electr. Notes Theor. Comput. Sci., 70(1), 2002.

Roger Hindley.
The completeness theorem for typing lambda-terms.
Theor. Comput. Sci., 22:1–17, 1983.

Fairouz Kamareddine, Karim Nour, Vincent Rahli, and Joe B. Wells.
Developing realisability semantics for intersection types and expansion variables.
2006.

Fairouz Kamareddine, Karim Nour, Vincent Rahli, and Joe B. Wells.
Realisability semantics for an intersection typing system with expansion variables.
2007.

George Koletsos and G. Stravinos.
Church-rosser property and intersection types.
2007.

Assaf J. Kfoury and Joe B. Wells.
Principality and decidability type inference for finite-rank intersection types.
pages 161–174, 1999.

Bertrand Russel.
Mathematical logic as based on the theory of types.
American Journal of Mathematics, 30(3):222–262, 1908.