Type-Theoretical Semantics with Coercive Subtyping

Zhaohui Luo
Dept of Computer Science
Royal Holloway, Univ of London

- What is type-theoretical semantics?
  - Formal semantics of NLs, in the Montagovian tradition, but in modern (dependent) type theories
- What is coercive subtyping?
  - $A \leq_c B$: subtyping as abbreviation
  - Subtyping framework adequate for modern TTs
- This talk:
  - Use of coercive subtyping in type-theoretical semantics

Type theories: types v.s. sets

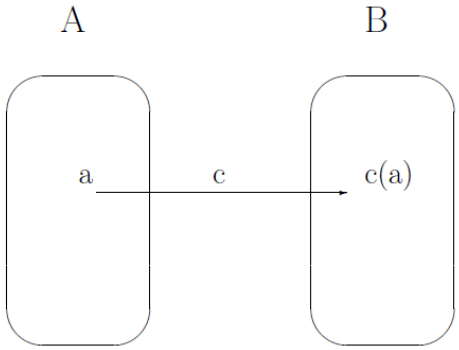
- Church's simple type theory (Montague semantics)
  - Types: e, t, e→t, (e→t)→t, …
  - Sets: s : A→t is a set over type A (a∈s iff s(a))
- Modern type theories (eg, Martin-Löf's type theory)
  - Many types – "many-sorted":
    - Propositional types
    - Inductive types (Nat, AxB, List(A), …)
    - Universes (eg, type Prop of all logical propositions)
  - But: few operations on types (much fewer than set oprns)
- Differences between types and sets
  - Eg, decidability: "a : A" is decidable, while "a ∈ s" is not.
  - Eg, logical consistency via "propositions-as-types" principle

Type-theoretical semantics (cf, Ranta 1994)

- Common nouns are interpreted as types (not as sets of type e→t)
  - Man =〔man〕: Type, Book =〔book〕: Type, …
- Verbs are interpreted as predicates
  - 〔walk〕: Man → Prop
  - 〔John walks〕=〔walk〕(j) : Prop
- Adjectives are interpreted as predicates
  - Handsome =〔handsome〕: Man → Prop
  - 〔handsome man〕= ∑(Man, Handsome)
  - Note: ∑-types for modified CNs.

Coercive subtyping: basic idea

- A≤B if there is a coercion c from A to B:



Eg. Even ≤ Nat; Man ≤ Human; ∑(Man, handsome) ≤ Man; …

- Subtyping as abbreviations:

  a : A $\leq_c$ B

  ➔ "a" can be regarded as an object of type B

  ➔ $\mathbf{C}_B[a] = \mathbf{C}_B[c(a)]$, ie, "a" stands for "c(a)"

Coercive subtyping: adequacy etc.

- Inadequacy of subsumptive subtyping
  - Canonical objects
  - Canonicity: key for TTs with canonical objects
  - Subsumptive subtyping violates canonicity.
- Adequacy of coercive subtyping
  - Coercive subtyping preserves canonicity & other properties.
  - Conservativity (Luo & Soloviev 2002; Xue, Luo & Adams 2011)
- Historical development and applications in CS
  - Formal presentation (Luo 1997/1999)
  - Implementations in proof assistants: Coq, Lego, Plastic, Matita

Coercive subtyping in TT semantics

1. Need for subtyping
2. Sense enumeration/selection via. overloading
3. Coercion contexts and local coercions
4. Dot-types and copredication
5. Structured lexical entries as Σ-types

Notes:

- Focus on representation mechanisms, rather than NL semantic treatments.
- However, linguistic examples, rather than formal details.

I. Subtyping: basic need in TT semantics

- What about, eg,
  - "A handsome man is a man" ?
  - "Paul walks", with p = 〔Paul〕 : 〔handsome man〕 ?
    (cf, "multiple categorisation of verbs" in (Ranta 1994))
- Solution: coercive subtyping
  - 〔handsome man〕 = $\sum$(Man, Handsome) $\leq_{\pi_1}$ Man
  - 〔Paul walks〕 = 〔walk〕(p) : Prop
    because
    〔walk〕 : Man$\rightarrow$Prop and p : 〔handsome man〕 $\leq_{\pi_1}$ Man

II. Sense selection via overloading

- Sense enumeration (cf, Pustejovsky 1995 and others)
  - Homonymy
  - Automated selection
- For example,

  1. John runs quickly.
  2. John runs a bank.

  with homonymous meanings

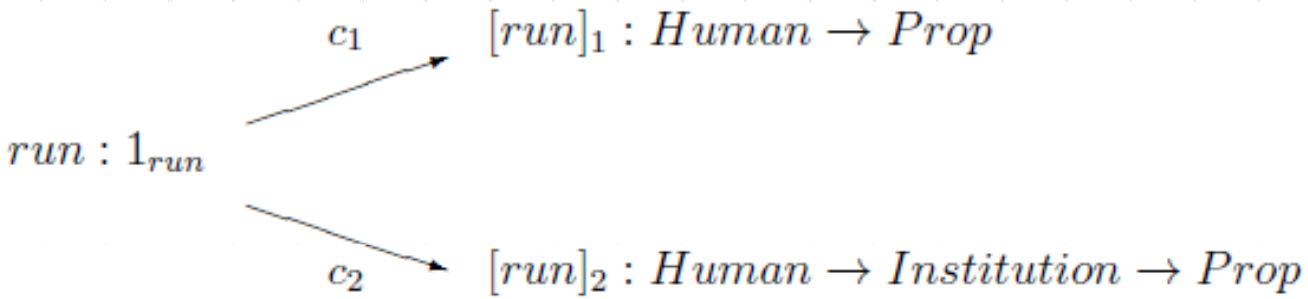
  1. $[\text{run}]_1$: Human→Prop
  2. $[\text{run}]_2$: Human→Institution→Prop

  "run" is overloaded – how to disambiguate?

Overloading via coercive subtyping

- Overloading can be represented by coercions

  Eg



- Homonymous meanings can be represented.
- Automated selection according to typings

Question: What if typings cannot disambiguate (eg, bank)?

A solution: Local coercions

III. Coercion contexts and local coercions

- Coercion contexts

  $x{:}C, \ldots, A \leq_c B, \ldots \vdash \ldots \ldots$

- Useful in representing context-sensitivity
  - Eg, reference transfer

    The ham sandwich shouts.

    This can be interpreted in a context that contains, eg,

    [sandwich] < [human]

    that coerces sandwich into the person who has ordered a sandwich.

Remark: "coherent contexts" needed, not just valid contexts.

(Formal details omitted.)

- Local coercions (in terms/judgements)

  coercion $A \leq_c B$ in t
- Useful in disambiguation
  - Eg, "bank" has different meanings in

    (1) the bank of the river

    (2) the richest bank in the city

    Rather than two coercions for "bank" in the same context, (which may be problematic), we can use

    coercion $[\text{bank}]_1 \leq [\text{riverside}]$ in [(1)]

    coercion $[\text{bank}]_2 \leq [\text{financial institution}]$ in [(2)]

Remark: Both coercion contexts and local coercions require further extensions of TT. (Formal details omitted.)

IV. Dot-types and copredication

- Dot-types in Pustejovsky's GL theory
  - Example: PHY•INFO
  - PHY•INFO ≤ PHY and PHY•INFO ≤ INFO
- Copredication

  "John picked up and mastered the book."

  ⟦pick up⟧ : Human → PHY → Prop
  ≤ Human → PHY•INFO → Prop
  ≤ Human → ⟦book⟧ → Prop

  ⟦master⟧ : Human → INFO → Prop
  ≤ Human → PHY•INFO → Prop
  ≤ Human → ⟦book⟧ → Prop

Remark: CNs as types in type-theoretical semantics – so things work.

Modelling dot-types in type theory

- What is A•B?
  - Inadequate accounts (cf, (Asher 08)):
    - Intersection type
    - Product type
- Proposal (SALT20, 2010)
  - A•B as type of pairs that do not share components
  - Both projections as coercions

$$\frac{A : Type \quad B : Type \quad \mathscr{C}(A) \cap \mathscr{C}(B) = \emptyset}{A \bullet B : Type}$$

$$\frac{a : A \quad b : B}{\langle a, b \rangle : A \bullet B} \qquad \frac{c : A \bullet B}{p_1(c) : A} \qquad \frac{c : A \bullet B}{p_2(c) : B} \qquad \frac{a : A \quad b : B}{p_1(\langle a, b \rangle) = a : A} \qquad \frac{a : A \quad b : B}{p_2(\langle a, b \rangle) = b : B}$$

$$\frac{A \bullet B : Type}{A \bullet B <_{p_1} A : Type} \qquad \frac{A \bullet B : Type}{A \bullet B <_{p_2} B : Type}$$

V. Structured lexical entries

- Proposal (1998, 2011):
  Basic CNs represented by Σ-types, eg,

$$[\![book]\!] = \sum \left\{ \begin{array}{lcl} Arg & : & \textsc{Phy} \bullet \textsc{Info} \\ Qualia & : & \sum \left\{ \begin{array}{lcl} Formal & : & Hold(p_1(Arg), p_2(Arg)) \\ Telic & : & R(Arg) \\ Agent & : & \exists h{:}Human.W(h, Arg) \end{array} \right\} \end{array} \right\}$$

- Remarks
  - Should lexicon be complex/structured/generative?
  - Non-CN lexical entries: a general structure (A, φ)?
  - Cf, Cooper's work on record types (2005, 2007)

Future work

- How far may a type-theoretical semantics go?
  - Further development: eg, predicate-modifying adverbs can be given type

    $$\prod A : CN.\ (A \to Prop) \to (A \to Prop).$$

    where CN is the universe of interpretations of CNs.
- Implementation
  - Extending mathematical vernacular
  - Exploiting the existing TT-based proof technology
- Technical issues include
  - Recent proof of conservativity of coercive subtyping (Xue et al 2011)
  - Meta-theory for coercion contexts and local coercions

References

- A. Ranta. *Type-Theoretical Grammar*. Oxford University Press. 1994.
- Z. Luo. Coercive subtyping. J. of Logic and Computation, 9(1). 1999.
- Z. Luo. Type-theoretical semantics with coercive subtyping. SALT20. 2010.
- Z. Luo. Contextual analysis of word meanings in type-theoretical semantics. LACL 2011, LNAI 6736. 2011.