Lecture III. Indefinites and Anaphora

- (Recap) MTT-semantics for adjectival modification
  - Left from Lecture I
  - Σ-types for the following Lecture III

Adjectival modification of CNs – case study

- A traditional classification
  - Kamp 1975, Parsons 1970, Clark 1970, Montague 1970

classification	**property**	**example**
Intersective	Adj(N) ➔ Adj & N	handsome man
Subsectional	Adj(N) ➔ N	large mouse
Privative	Adj(N) ➔ ¬N	fake gun
Non-committal	Adj(N) ➔ ?	alleged criminal

Intersective adjectives

- Example: handsome man (see next page for Σ-types)

	Montague	**MTT-semantics**
man	man : **e→t**	Man : Type
handsome	handsome : **e→t**	Man→Prop
handsome man	λx. man(x) & handsome(x)	Σ(Man,handsome)

- In general:

	Montague	**MTT-semantics**
CNs	predicates	types
Adjectives	predicates	simple predicates
CNs modified by intersective adj	Predicate by conjunction	Σ-type

Σ-types

- An extension of the <u>product types</u> A x B of pairs
- Σ-types of "dependent pairs"
  - Σ(A,B) of (a,b) for a:A & b:B(a)
- Rules for Σ-types:
  - Σ(A,B) also written as Σx:A.B(x)
- Examples:
  - Σ(Human,dog)

    with dog(j)={d}, dog(m)=∅, …
  - Σ(Man,handsome)

$$(\Sigma)\quad \frac{\Gamma \vdash A\ type \quad \Gamma,\ x : A \vdash B\ type}{\Gamma \vdash \Sigma x : A.B\ type}$$

$$(pair)\quad \frac{\Gamma \vdash a : A \quad \Gamma \vdash b : [a/x]B \quad \Gamma, x : A \vdash B\ type}{\Gamma \vdash (a,b) : \Sigma x : A.B}$$

$$(\pi_1)\quad \frac{\Gamma \vdash p : \Sigma x : A.B}{\Gamma \vdash \pi_1(p) : A}$$

$$(\pi_2)\quad \frac{\Gamma \vdash p : \Sigma x : A.B}{\Gamma \vdash \pi_2(p) : [\pi_1(p)/x]B}$$

$$(proj_1)\quad \frac{\Gamma \vdash a : A \quad \Gamma \vdash b : [a/x]B \quad \Gamma, x : A \vdash B\ type}{\Gamma \vdash \pi_1(a,b) = a : A}$$

$$(proj_2)\quad \frac{\Gamma \vdash a : A \quad \Gamma \vdash b : [a/x]B \quad \Gamma, x : A \vdash B\ type}{\Gamma \vdash \pi_2(a,b) = b : [a/x]B}$$

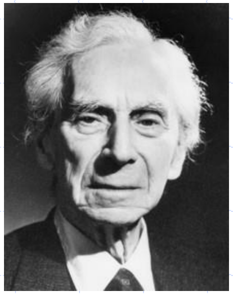
- An adjective maps CNs to CNs:
  - In MG, predicates to predicates.
  - In MTT-semantics, types to types.
- MTT-semantics (Chatzikyriakidis & Luo 2020, Luo 2023)

classification	**example**	**types employed**
Intersective	handsome man	Σ-types with simple predicates
Subsectional	large mouse	Π-polymorphic predicates and Σ-types
Privative	fake gun	Disjoint union types with Π/Σ-types
Non-committal	alleged criminal	special predicates

This lecture

1. Indefinites and the Russellian ∃-view
2. Dynamic semantics
3. Type-theoretical approach
4. Problem with the type-theoretic approach and solution with both strong/weak sums (possibly in Lecture IV)

III.1. Indefinites and Russellian ∃-view

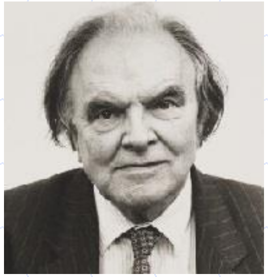


- We'll discuss indefinites like "a man". Are they
  - Quantifier phrases (as Russell suggests)?
  - Referring expressions?
- Russell (1919): the ∃-view
  - A man came in. ➔ $\exists x{:}\mathbf{e}.\ \text{man}(x) \wedge \text{come\_in}(x)$
- Arguments/examples in favour of the ∃-view
  - John saw <u>a dog</u> and Mary saw <u>a dog</u>, too.
    [Could be different dogs. Russell's ∃-view predicts it.]
    [Different "a dog" could refer to different things. c.f., He likes him.]
  - It is not the case that <u>a man</u> came in.
    Every child owns <u>a dog</u>.
    [Not a particular man/different dogs. Russell's ∃-view predicts it.]

- But what about, for example,
  - (1) A man came in. He lit a cigarette.

    (#) $[\exists x{:}\mathbf{e}.\text{man}(x)\wedge\text{come\_in}(x)]\ \ [\exists y{:}\mathbf{e}.\text{cigarette}(y)\wedge\text{light}(x_?,y)]$

    Geach's proposed solution (1962): put the latter into the scope of $\exists x$. But, this is non-compositional …



  - (2) Every farmer who owns a donkey beats it.

    (#) $\forall x{:}\mathbf{e}.\ \text{farmer}(x) \wedge \exists y{:}\mathbf{e}.(\text{donkey}(y) \wedge \text{own}(x,y) \rightarrow \text{beat}(x,y)$
  - (3) Every person who buys a TV and has a credit card uses it to pay for it.
- In the above sentences, "it" seems to refer to something
  - Variable? E.g., $x_?$ for (1) and the last "y" for (2)
  - But they are outside their scopes!

III.2. Dynamic semantics

- Dynamic approaches (widely accepted for anaphora treatment)
  - Discourse Representation Theory (Kamp 1981, Heim 1982)
  - Dynamic Predicate Logic (Groenendijk and Stokhof 1991)

$$\boxed{(\exists x \varphi); \psi} \Leftrightarrow ([x]; \varphi); \psi \Leftrightarrow [x]; (\varphi; \psi) \Leftrightarrow \boxed{\exists x(\varphi; \psi)}$$

$$\boxed{(\exists x \varphi) \to \psi} \Leftrightarrow ([x]; \varphi) \to \psi \Leftrightarrow [x] \to (\varphi \to \psi) \Leftrightarrow \boxed{\forall x(\varphi \to \psi)}$$

where ";" is the dynamic conjunction and ψ may have free x !

  - So, if we replace $\wedge$ by ; then $x_?$ and "y" in previous interpretations would be OK (because of the above equivalences)!

    ∀x:**e**. [farmer(x) ; ∃y:**e**.(donkey(y) ; own(x,y)] → beat(x,y)

    ⇔ ∀x:**e**∀y:**e**. [farmer(x) ; donkey(y) ; own(x,y)] → beat(x,y)

    This equivalence is true because of the above 2nd equivalence.

- However, logics in dynamic semantics are rather non-standard.
  - Dynamic Predicate Logic (DPL, Groenendijk and Stokhof 1991) is non-monotonic, has irreflexive/intransitive entailment, …
  - Substantial changes required for underlying logic(s) in semantics
- Two "extremes"? Anything "in the middle"?

  Russell (∃) |------------?------------| Dynamic

- Σ-types in MTTs may provide such a "middle" solution!

III.3. Type-theoretical approach

- Using dependent types (Mönnich 1985, Sundholm 1986)
- Every farmer who owns a donkey beats it.
  - (#) $\forall x$:**e**. farmer(x) $\wedge$ $\exists y$:**e**.(donkey(y) $\wedge$ own(x,y) → beat(x,y)
- In type theory, we could give semantics as follows:
  - $\forall z$: [Σx:Farmer Σy:Donkey. Own(x,y)]. Beat(π_1(z), π_1(π_2(z)))
  - Σ is the "strong sum" with two projections π_1 and π_2 .
  - Therefore, "it" refers to "a donkey" – by means of π_i, as π_1(π_2(z))
- This gives a compromise – something "in the middle" – see below.

Σ-types (recap)

- An extension of the product types A x B of pairs
- Σ-types of "dependent pairs"
  - Σ(A,B) of (a,b) for a:A & b:B(a)
- Rules for Σ-types:
  - Σ(A,B) also written as Σx:A.B(x)
- Examples:
  - Σ(Human,dog)

    with dog(j)={d}, dog(m)=∅, …
  - Σ(Man,handsome)

$$(\Sigma)\quad \frac{\Gamma \vdash A\ type \quad \Gamma,\ x : A \vdash B\ type}{\Gamma \vdash \Sigma x : A.B\ type}$$

$$(pair)\quad \frac{\Gamma \vdash a : A \quad \Gamma \vdash b : [a/x]B \quad \Gamma, x : A \vdash B\ type}{\Gamma \vdash (a,b) : \Sigma x : A.B}$$

$$(\pi_1)\quad \frac{\Gamma \vdash p : \Sigma x : A.B}{\Gamma \vdash \pi_1(p) : A}$$

$$(\pi_2)\quad \frac{\Gamma \vdash p : \Sigma x : A.B}{\Gamma \vdash \pi_2(p) : [\pi_1(p)/x]B}$$

$$(proj_1)\quad \frac{\Gamma \vdash a : A \quad \Gamma \vdash b : [a/x]B \quad \Gamma, x : A \vdash B\ type}{\Gamma \vdash \pi_1(a,b) = a : A}$$

$$(proj_2)\quad \frac{\Gamma \vdash a : A \quad \Gamma \vdash b : [a/x]B \quad \Gamma, x : A \vdash B\ type}{\Gamma \vdash \pi_2(a,b) = b : [a/x]B}$$

So, in more details:

- Σ is a quantifier – Σx:A.P(x)
  - Quantifying over x in the scope P(x).
  - Σx:Man.handsome(x)
  - Σy:Donkey.own(j,y)
- Σ is like the existential quantifier ∃
  - ∃y:Donkey.own(j,y)

  but different: it has the first projection π_1:

  $$(a,b) : \Sigma x{:}A.P(x) \;\rightarrow\; \pi_1(a,b) = a$$

- This first projection does not exist for ∃. That's why Σ is also called the "strong sum", while ∃ the "weak sum".

- Two "extremes"? Anything "in the middle"?

  Russell (∃) |------------?------------| Dynamic

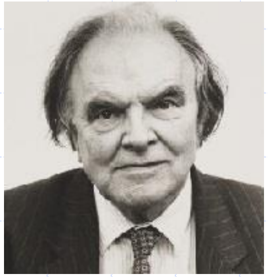
- Σ-types in MTTs may provide such a "middle" solution!
  - Σ is "strong" so that witnesses can be referred to outside its scope (by means of π_1 and π_2).
  - The change for the underlying logic is much less substantial in the sense that we just use Σ instead of ∃.
- However, still a (minor?) problem – see below.

A problem

- Σ has played two related but different roles.
  - "Subset":
    - Σx:Farmer. P(x) for "the farmers such that P holds"
  - Existential:
    - Σx:Farmer Σy:Donkey.own(x,y) for "the farmers who own a donkey"
- This is problematic → counting problem.
  - Satisfactory solution with both strong/weak sums (Luo 2021)
  - We'll use donkey anaphora as a case study.

- (III.4 is moved to Lecture IV)

III.4. Donkey anaphora: problem and solution (Luo 2021)



- Examples (Geach 1962, …)
  - (1) Every farmer who owns <u>a donkey</u> beats <u>it</u>.
  - (2) Every person who buys <u>a TV</u> and has <u>a credit card</u> uses <u>it</u> to pay for <u>it</u>.
- Strong/weak readings (Chierchia 1990):
  - Strong reading of (1):

    Every farmer who owns a donkey beats <u>every</u> donkey s/he owns.
  - Weak reading of (1):

    Every farmer who owns a donkey beats <u>some</u> donkeys s/he owns.

Original problem and use of dependent types

- Every farmer who owns a donkey beats it.
- In traditional logics:
  - $(\#)\ \forall x.\ [farmer(x)\ \&\ \exists y.(donkey(y)\ \&\ own(x,y))] \Rightarrow beat(x,y)$
    where $\exists$ is a "<u>weak sum</u>" and the last y is outside its scope.
- Using dependent types (Mönnich 85, Sundholm 86)
  - $\forall z : F_\Sigma.\ beat(\pi_1(z), \pi_1(\pi_2(z)))$ with $F_\Sigma = \Sigma x{:}F\ \Sigma y{:}D.\ own(x,y)$
    where Σ is the "<u>strong sum</u>" with two projections π_1 and π_2 .
  - Note: the interpretation only conforms to the strong reading.
- Σ plays a <u>double role</u>:
  - subset constructor (1^{st} Σ) and existential quantifier (2^{nd} Σ).
  - But this is problematic ➔ counting problem.

Problem of counting (Sundholm 89, Tanaka 15)

- Cardinality of finite types
  - $|A| = n$ if $A \cong Fin(n)$ (i.e., it has exactly n objects.)
- Consider the donkey sentence with “most”:
  - Most farmers who own a donkey beat it.
  - $Most_S\ z : F_\Sigma.\ beat(\pi_1(z), \pi_1(\pi_2(z)))$ with $F_\Sigma = \Sigma x{:}F\ \Sigma y{:}D.\ own(x, y)$
- But, this is inadequate – failing to “count” correctly:
  - $|F_\Sigma|$ = the number of (x,y,p) $\neq$ #(donkey-owning farmers)
    - E.g., 10 farmers:
    - 1 owns 20 donkeys and beats all of them, and
    - the other 9 own 1 donkey each and do not beat them.
    - The above sentence with “most” could be true – incorrect semantics.
  - C.f., the “proportion problem” in using DRT to do this.

Why and …?

- "Double role" by Σ in $F_\Sigma = \Sigma x{:}Farmer\,\Sigma y{:}Donkey.own(x,y)$
  - First Σ: representing the collection of farmers such that …
  - Second Σ: representing the existential quantifier (!)
- But, unlike traditional ∃, Σ is strong:
  - $|\Sigma x{:}A.B(x)|$ is the number of pairs (a,b), not just the number of a's such that B(a) is true. So, the 2nd Σ is problematic.
- Can we somehow replace the 2nd Σ by ∃?
  - Yes, although not directly (c.f., the original scope problem), by considering different readings of donkey sentences <u>AND IF</u> we have both Σ and ∃ in the type theory.
  - Note: ∃ in Montague's simple TT and Σ in Martin-Löf's TT, but <u>not both</u>.

UTT (Luo 1994): a type theory with both Σ/∃

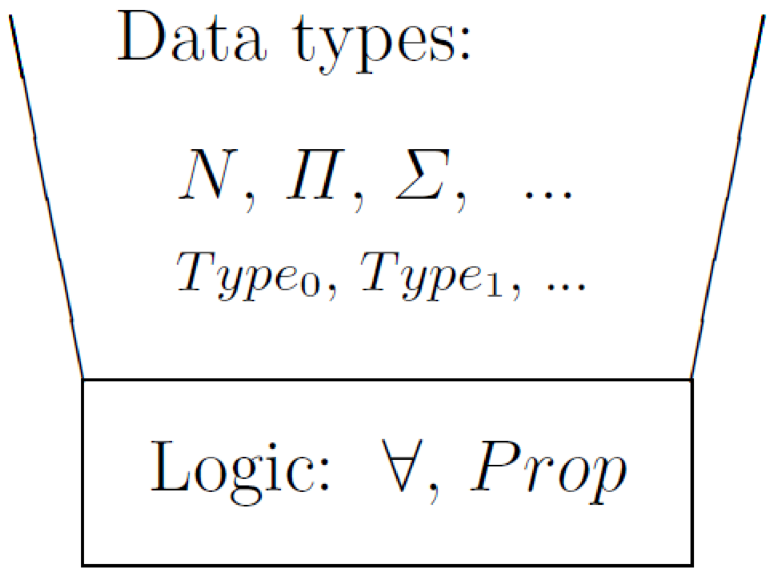


Fig. 1. The type structure in UTT.

Logic in UTT and proof irrelevance

- Formulas/propositions: $\forall x{:}A.P$, $\exists x{:}A.P$, $P \Rightarrow Q$, …
  - For example: $\exists x : A.P(x) = \forall X : Prop.\ (\forall x : A.(P(x) \Rightarrow X)) \Rightarrow X$
- Proof irrelevance:
  - Every two proofs of the same proposition are the same.
  - In UTT, this can be enforced by the following rule:

$$\frac{P : Prop \quad p : P \quad q : P}{p = q : P}$$

  - Note: This wouldn't be possible for Martin-Löf's type theory.
- As a consequence, we have, for example:
  - $|P| \leq 1$, if $P : Prop$ (e.g., $|\exists x{:}A.R| \leq 1$)
  - $|\Sigma x{:}A.Q(x)| \leq |A|$, if A is a finite type and $Q : A \rightarrow Prop$

Donkey sentences in UTT

- Most farmers who own a donkey beat it.
  - Most farmers who own a donkey beat every donkey they own.
  - Most farmers who own a donkey beat some donkeys they own.
- "Most" in UTT (formal details next page)
  - Definition similar to (Sundholm 89), but with $\exists$ as existential quantifier, instead of Σ.
- Interpretations

$$F_\exists = \Sigma x{:}F.\ \exists y{:}D.own(x,y)$$

$$Most\ \ z : F_\exists.\ \forall y' : \Sigma y{:}D.own(\pi_1(z),y).\ beat(\pi_1(z),\pi_1(y'))$$

$$Most\ \ z : F_\exists.\ \exists y' : \Sigma y{:}D.own(\pi_1(z),y).\ beat(\pi_1(z),\pi_1(y'))$$

D *Most* in UTT

Let A be a finite type with $|A| = n_A$, $P : A \to Prop$ a predicate over A, and $Fin(n)$ the types with n objects defined in Appendix B. Then, in UTT, the logical proposition $Most\ x{:}A.P(x)$ of type $Prop$ is defined as follows, where $inj(f)$ is a proposition expressing that f is an injective function:

$$Most\ x{:}A.P(x) = \exists k : N.\ (k \geq \lfloor n_A/2 \rfloor + 1) \\ \wedge \exists f{:}Fin(k) \to A.\ inj(f) \wedge \forall x{:}Fin(k).P(f(x))$$

The type $Fin(n)$, indexed by $n : N$ with N being the type of natural numbers, consists of exactly n objects and can be specified by means of the following introduction rules (we omit their elimination and computation rules):

$$\frac{n : N}{zero(n) : Fin(n+1)}$$

$$\frac{n : N \quad i : Fin(n)}{succ(n,i) : Fin(n+1)}$$

Another example

- Every person who buys a TV and has a credit card uses it to pay for it.
  - where "a TV" obtains a strong $\forall$-reading and "a credit card" a weak $\exists$-reading.

$$\forall z : \Sigma x{:}Person.\ \exists y_1{:}TV.\ buy(x, y_1) \wedge \exists y_2{:}Card.\ own(x, y_2)$$
$$\forall y : \Sigma y_1{:}TV.\ buy(\pi_1(z), y_1)$$
$$\exists y' : \Sigma y_2{:}Card.\ own(\pi_1(z), y_2).$$
$$\quad pay(\pi_1(z), \pi_1(y), \pi_1(y'))$$

- Note: It would be impossible to do this in MLTT.

E-type Anaphora (Evans 77, …)

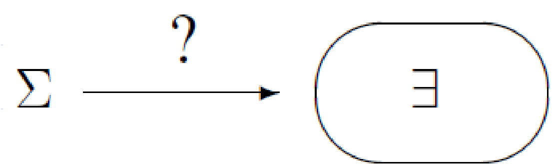
- Evans' example:
  - Few congressmen admire Kennedy, and they are very junior.
  - Few congressmen admire Kennedy, and the congressmen who do admire Kennedy are very junior.
- Note: "they" cannot be "bound" by "Few congressmen" for, otherwise, the meaning is different.
  - It would mean: Few congressmen are such that they admire Kennedy and are very junior (at the same time).
- Interpretations in type theory:

$$Few\ x{:}C.admire(x,K) \wedge \forall z{:}[\Sigma x{:}C.admire(x,K)].junior(\pi_1(z))$$

  - Link of Σ with descriptions (Martin-Löf, Carlström, Mineshima)

Combining strong and weak sums

- How to add Σ to an impredicative type theory with ∃-propositions?



- Three possibilities:
  - UTT (seen before): Σ-types + ∃-propositions
  - "Large" Σ-propositions
    - ➔ logical inconsistency

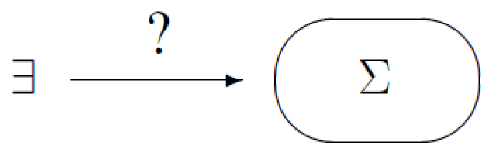
$$\frac{A\ type \quad P : A \to Prop}{\Sigma x{:}A.P(x) : Prop}$$

  - "Small" Σ-propositions
    - ➔ weak ∃ becoming strong

$$\frac{A : Prop \quad P : A \to Prop}{\Sigma x{:}A.P(x) : Prop}$$

Conclusion: Only the UTT's approach is OK.

- How to add ∃ to a predicative type theory with Σ-types?



- Not clear how to do this (but see next page for $MLTT_h$)
  - One might define ∃ by Π in predicative universes U_i:

$$\exists_i x{:}A.B(x) = \Pi X{:}U_i.\ (\Pi x{:}A.(B(x) \to X)) \to X$$

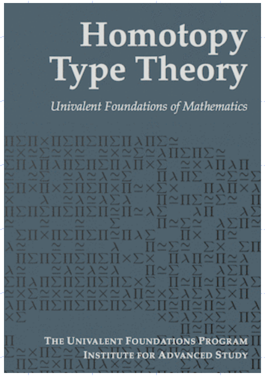
  - But, thus defined, $\exists_i$ is the same as the strong sum Σ!

    We can define $p : \exists_i x{:}A.B(x) \to \Sigma x{:}A.B(x)$ such that $\exists_i$-projections exist:

$$p_1 = \pi_1 \circ p \quad \text{and} \quad p_2 = \pi_2 \circ p$$

    (Bad side effect!)

$MLTT_h$: Extension of MLTT with H-logic



- H-logic (in Homotopy Type Theory; HoTT book 2013)
  - A proposition is a type with at most one object.
  - Logical operators (examples):
    - $P \supset Q = P \to Q$ and $\forall x{:}A.P = \prod x{:}A.P$
    - $P \vee Q = |P+Q|$ and $\exists x{:}A.P = |\Sigma x{:}A.P|$

    where $|A|$ is <u>propositional truncation</u>, a proper extension.
- $MLTT_h$ = MLTT + h-logic (subsystem of HoTT) [Luo 2019]
  - Proof irrelevance is “built-in” in h-logic (by definition).
  - $\exists$ defined by truncating Σ is a weak sum and can be used to give adequate semantics of donkey sentences as proposed.
  - Note: $MLTT_h$ is a <u>proper extension</u> of MLTT.

Concluding remarks

- Summary
  - Donkey sentences – old topic, but still intriguing.
  - Type theories – with "standard" logics embedded.
  - We have studied this completely proof-theoretically.
- Dynamics in semantics
  - "Dynamic type theory"? (Not a way forward, even IF possible)
    - Cf, Dynamic Predicate Logic that extends FOL.
    - But, DPL is non-standard (eg, non-monotonic …) and proof-theoretically difficult [Veltman 2000] (probably problematic).
  - Formal semantics based on such is too big a price to pay.