Extension and Intension

Kenneth Lai

Brandeis University

October 18, 2022

Announcements

- For tomorrow
  - Read Lipovača Chapter 8.9, 11.1, 11.2, if you can
  - HW2 due
  - Paper Presentation Ideas due

Paper Presentation Ideas

- If you are still looking for a group, please let me know by the end of the day today
  - If you have any paper(s) in mind, or any possible topics, that would be good to know

Today's Plan

- Implementing Semantic Interpretation: NP and VP
- Extension and Intension

Adjectives

```
friendly :: (Entity -> Bool) -> (Entity -> Bool)
friendly p | filter p entities == filter wizard entities
                       = list2OnePlacePred [W]
           | filter p entities == filter girl entities
                       = list2OnePlacePred [S,A,D,G]
           | otherwise = list2OnePlacePred []
```

Implementing Semantic Interpretation

- Interpretation of noun phrases: two kinds
  - Proper names: “Atreyu”
  - Determiners plus a common noun: “every wizard”

Implementing Semantic Interpretation

- Interpretation of noun phrases: two kinds
  - Proper names: “Atreyu”
  - Determiners plus a common noun: “every wizard”

  - (Also noun phrases with relative clauses: “the girl that laughed”, “some boy that Snow White loved”, etc., which we won’t talk about here)

Implementing Semantic Interpretation

- Interpretation of noun phrases: two kinds
  - Proper names: "Atreyu"
  - Determiners plus a common noun: "every wizard"

  - (Also noun phrases with relative clauses: "the girl that laughed", "some boy that Snow White loved", etc., which we won't talk about here)

- Problem: "Atreyu" is an `Entity`, while "every wizard" is an `(Entity -> Bool) -> Bool`

Solution: 'Generalization to the worst case'

All NPs denote expressions of type $(e \to t) \to t$.

$$
\begin{aligned}
[\![\textit{some princess}]\!] &= \lambda P.\exists x.(\textit{princess}\ x) \wedge (P\ x) \\
[\![\textit{every wizard}]\!] &= \lambda P.\forall x.(\textit{wizard}\ x) \to (P\ x) \\
[\![\textit{Atreyu}]\!] &= \lambda P.(P\ a) \\
[\![\textit{Dorothy}]\!] &= \lambda P.(P\ d)
\end{aligned}
$$

Individuals as generalized quantifiers

$$[\![Atreyu]\!] = \lambda P.(P\ a)$$

- *Atreyu* denotes a function that takes a predicate and hands it the argument a.
- So it tells us, which properties are true of Atreyu.
- Technically, *Atreyu* denotes the characteristic function of the set of all sets that contain the individual Atreyu.
  In other words: *Atreyu* denotes the set of all properties of Atreyu.

Implementing Semantic Interpretation

- Interpretation of proper names in Haskell

```
intNP SnowWhite     = \ p -> p snowWhite
intNP Alice         = \ p -> p alice
intNP Dorothy       = \ p -> p dorothy
intNP Goldilocks    = \ p -> p goldilocks
intNP LittleMook    = \ p -> p littleMook
intNP Atreyu        = \ p -> p atreyu
```

Implementing Semantic Interpretation

- Interpretation of verb phrases: three kinds
  - Intransitive verbs: "cheered"
  - Transitive verbs plus a direct object: "admired Alice"
  - Ditransitive verbs plus an indirect object and a direct object: "gave every princess some sword"

Implementing Semantic Interpretation

- Interpretation of verb phrases: three kinds
  - Intransitive verbs: "cheered"
  - Transitive verbs plus a direct object: "admired Alice"
  - Ditransitive verbs plus an indirect object and a direct object: "gave every princess some sword"
- Problem: transitive verbs have type `Entity -> Entity -> Bool`, while direct objects (noun phrases) have type `(Entity -> Bool) -> Bool`
  - Similar problem for ditransitive verbs

Implementing Semantic Interpretation

- Solution: quantifier raising

```
intVP (VP1 tv np) =
  \ subj -> intNP np (\ obj -> intTV tv subj obj)
intVP (VP2 dv np1 np2) =
  \ subj -> intNP np1 (\ iobj -> intNP np2 (\ dobj ->
                           intDV dv subj iobj dobj))
```

Implementing Semantic Interpretation

- Solution: quantifier raising

```
intVP (VP1 tv np) =
  \ subj -> intNP np (\ obj -> intTV tv subj obj)
intVP (VP2 dv np1 np2) =
  \ subj -> intNP np1 (\ iobj -> intNP np2 (\ dobj ->
                              intDV dv subj iobj dobj))
```

- See handout!

Computational Semantics
Day 4: Extensionality and intensionality

Jan van Eijck[1] & Christina Unger[2]

[1]CWI, Amsterdam, and UiL-OTS, Utrecht, The Netherlands
[2]CITEC, Bielefeld University, Germany

ESSLLI 2011, Ljubljana

Recapitulation: Meaning as reference

Until now we identified meaning with reference.

- Names denote individual constants (type e), which represent the entities they refer to.
- Sentences denote truth-values (type t).
- Predicates denote functions from individuals to truth-values.

Extensionality

Extensionality

In a complex expression E, a sub-expression can be substituted by another expression that has the same meaning without changing the meaning of E.

If meaning is reference:
Expressions with the same reference should be interchangeable without changing the truth-value of the sentences they occur in.

Extensional contexts

- *Alonzo greeted the queen of the Netherlands.*
- *Alonzo greeted Beatrix.*

- *Eight is greater than seven.*
- *The number of planets in our solar system is greater than seven.*

Opaque contexts

- *Eight is necessarily greater than seven.*
- *The number of planets in our solar system is necessarily greater than seven.*

- *I believe that Alonzo greeted the queen of the Netherlands.*
- *I believe Alonzo greeted Beatrix.*

- *Alonzo is looking for the queen of the Netherlands.*
- *Alonzo is looking for Beatrix.*

Since those sentences mean different things, meaning seems to be more than reference.

Sense and reference

Frege proposed to distinguish:

- the conceptual content of an expression (*Sinn*, or *intension*)
- its actual reference (*Bedeutung*, or *extension*)

Sense and reference

Frege proposed to distinguish:

- the conceptual content of an expression (*Sinn*, or *intension*)
- its actual reference (*Bedeutung*, or *extension*)

Example:

- *the queen of the Netherlands*
  - Intension: royal head of the state of the Netherlands
  - Extension: Juliana (1949-1980), Beatrix (1980-present), . . .
- *Slovenian number category*
  - Intension: grammatical category that expresses count distinction in Slovene
  - Extension: {singular,dual,plural}
- For a sentence, the intension is its truth conditions and the extension is its actual truth-value.

Intension and extension

So usually the intension is fixed, while the extension varies from context to context.

But vice versa, the extension could be the same, while the intension differs.

Example: *morning star, evening star*

- *The morning star is the evening star.*
- *The morning star is the morning star.*

Intensions

When we want to determine the reference of an expression, we have to consider the context, i.e. reference is not absolute anymore but depends on the context (time, possible worlds, anaphoric potential,. . .).

Meaning as intension

Meaning as intension

The meaning of an expression is not its extension (reference) anymore but its intension, i.e. a function that determines the reference given a certain context.

Intensions are functions from contexts to extensions.

Context: Possible worlds

Hintikka, Kripke

A *possible world* is a state of affairs that can differ from the actual state of affairs in any point.

Possible worlds can be represented as sets of propositional constants, namely all the propositions that hold in that world.

Examples

Truth and reference depend on the actual as well as possible situation.

- *The lecturer's team might have won.*
- *If the argonauts had recognized the Dolions, they wouldn't have killed them.*

Examples

Truth and reference depend on the actual as well as possible situation.

- *The lecturer's team might have won.*
- *If the argonauts had recognized the Dolions, they wouldn't have killed them.*

To determine the truth of modal statements like

- *Possibly the Higgs boson exists.*
- *The Higgs boson necessarily exists.*

it is not important to know whether *The Higgs boson exists* is true in the actual world, but rather whether it is true in some world (so it is possible that it is true) or in all worlds (so there is no other way than for it to be true).

Extension and Intension

- Idea 1: If the interpretation of something in an extensional model has type α, then its intensional interpretation has type $s \to \alpha$, where s is the type of possible worlds (`World`)

Extension and Intension

- Idea 1: If the interpretation of something in an extensional model has type α, then its intensional interpretation has type $s \to \alpha$, where s is the type of possible worlds (`World`)
  - Introduce abbreviations for types `World -> Entity` and `World -> Bool`

```
type IEntity = World -> Entity
type IBool   = World -> Bool
```

Extension and Intension

```
iSnowWhite :: IEntity
iSnowWhite W1 = snowWhite
iSnowWhite W2 = snowWhite'
iSnowWhite W3 = snowWhite'

iGirl, iPrincess, iPerson :: World -> Entity -> Bool
iGirl     W1 = girl
iGirl     W2 = girl'
iGirl     W3 = girl'
iPrincess W1 = princess
iPrincess W2 = princess'
iPrincess W3 = girl'
iPerson   W1 = person
iPerson   W2 = person'
iPerson   W3 = person'
```

Extension and Intension

```
iLaugh, iShudder :: World -> Entity -> Bool
iLaugh W1 =  laugh
iLaugh W2 =  laugh'
iLaugh W3 =  laugh'
iShudder W1 =  shudder
iShudder W2 =  shudder'
iShudder W3 =  shudder'

iCatch :: World -> Entity -> Entity -> Bool
iCatch W1 = \ x y -> False
iCatch W2 = \ x y -> False
iCatch W3 = \ x y -> elem x [B,R,T] && girl' y
```