ELEMENTS OF DEDUCTIVE LOGIC

12. Predicate Logic: subsentential structure

J. Chandler

KUL 2012

Introduction

- Briefly mentioned a few lectures ago: the validity of *many* arguments isn't down to their sentential form.

Validity without valid sentential form

All hippos are bad-tempered.
Harry is a hippo. (Most descriptive s. form:
Therefore Harry is bad-tempered. p, q, therefore r)

- They are valid because of their sub-sentential forms: we need to look 'inside' the atomic sentences.

Introduction

- To talk about subsentential forms: we move from $\mathcal{L}_S$ to the language $\mathcal{L}_P$.
- We have some leftovers from $\mathcal{L}_P$.
- We keep our connectives: $\sim, \&, \vee, \supset, \equiv$
- And the relevant syntactic rule:

  If φ and ψ are wfs's, so too are $\sim\varphi$, $(\varphi\&\psi)$, $(\varphi\vee\psi)$, $(\varphi\supset\psi)$ and $(\varphi\equiv\psi)$.
- But we no longer have the structureless atomic wfs's of $\mathcal{L}_S$: all our wfs's will now have an internal structure.
- Note that Restall has $\mathcal{L}_P$ as an *extension* of $\mathcal{L}_S$: his language of PL also includes the atomic wfs's of $\mathcal{L}_S$.

Introduction

- The second premise of our introductory argument:
  (1) 'Harry is a hippo.'
- It involves:
  - a name, aka 'designator' ('Harry'), referring to a particular thing, here a person
  - a predicate ('. . . is a hippo'), to saying something about that thing
- Predicates are a bit like connectives in one respect:
  - Connectives make sentences out of sentences
  - Predicates make sentences out of names

Arity

- Like connectives, predicates comes in different 'arities' (= take different numbers of elements as 'inputs').

Arities

Unary: '…is short'

Binary: '…loves…'

Ternary: '…is between…and …'

- Unary predicates ⇒ claims regarding whether or not certain individuals have certain properties.
- n-ary predicates ($n > 1$) ⇒ claims regarding whether or not certain individuals stand in a certain relations to one another.

Names and predicates in $\mathcal{L}_P$

- Two new kinds of symbols in our expanded language $\mathcal{L}_P$ of subsentential forms:
  - For names, aka 'constants': $a, b, c, \ldots, a_1, b_2, c_3, \ldots$
  - For predicates: $F, G, H, \ldots, F_1, G_2, H_3, \ldots$
    (Note: these are sometimes superscripted to indicate arity, e.g. F^2 for a binary predicate.)
- And a new wfs formation rule:

  If F is a predicate of arity n and $a_1, \ldots, a_n$ are names, then $Fa_1 \ldots a_n$ is a wfs.
- Stylistic variants:
  - $F(a_1 \ldots a_n)$,
  - aFb (for binary predicates).
- Note: order matters! ($Lab \neq Lba$)

Translation

- Remember, for propositional logic:
  - To get a sentential form:

    different atomic NL sentences → different atomic $\mathcal{L}_S$ sentences
  - To get the *most descriptive* s. form: we also need

    same atomic NL sentences → same atomic $\mathcal{L}_S$ sentences
- Similar principle here:
  - To get a subsentential form:

    different NL names/predicates → different $\mathcal{L}_P$ names/predicates
  - To get the *most descriptive* subs. form: we also need

    same NL names/predicates → same $\mathcal{L}_P$ names/predicates

Translation (ctd.)

Subsentential form

'John liked Samantha, and Karl liked Trisha.'

Non-forms (diff n/p → same n/p):

- $Ljj\&Lkt$,
- $Ljs\&Lks$, etc.

Forms (diff n/p → diff n/p):

- $Ljs\&Mkt$,
- $Ljs\&Lkt$, etc.

Most descriptive form (same n/p → same n/p):

- $Ljs\&Lkt$

Translation (ctd.)

- Sometimes, the logical form of a sentence can't just be straightforwardly 'read off' :

  Before formalising, try first to *paraphrase* the sentence in English, using constructions that have straightforward translations.

Paraphrasing first

'John and Karl love themselves.' ⇒ 'John loves John and Karl loves Karl.' ⇒ $Ljj\&Lkk$

'John smokes Camels and so does Karl.' ⇒ 'John smokes Camels and Karl smokes Camels.' ⇒ $Cj\&Ck$

'John used to be either a policeman or a fireman.' ⇒ 'John used to be a policeman or John used to be a fireman.' ⇒ $Pj \vee Fj$

Translation (ctd.)

- Use predicates of the right arity.

The right arity

'Mary hated John.' ⇒ Hmj

We use a binary predicate standing for '…hated…', not a unary predicate standing for '…hated John'.

- Beware of mixed passive/active variants in a given argument or sentence: use a single predicate for both forms.

Mixed passive/active

'Either Mary was hated by John, or she hated him.' ⇒ $Hjm \vee Hmj$

Quantifiers

- The first premise of our introductory argument:
  (2) 'All hippos are bad-tempered.'
- This isn't about a *particular* individual: no proper names here…
- Instead: quantified sentence.
- Quantified sentences are statement about *quantities* of individuals.
  - 'all'
  - 'some'
  - 'most'
  - 'few'
  - 'at least n'
- In $\mathcal{L}_P$, we will just have translations for sentences involving:
  - 'all' (aka universal quantification)
  - 'at least one' (aka existential quantification)

Quantification in $\mathcal{L}_P$

- Ok, so what exactly do we *mean* by
  (3) 'All hippos are bad-tempered.'
- Plausible gloss:

  It is true of *any* thing, call it 'x', that if x is a hippo, then x is bad tempered.
- Similarly, consider:
  (4) 'Adam has lost something.'
- Gloss:

  It is true of *at least one* thing, call it 'x', that Adam has lost x.
- The translations of (3) and (4) in $\mathcal{L}_P$ are very close to these glosses, in structural terms…

Quantification in $\mathcal{L}_P$ (ctd.)

- Regarding (3):
  $(\forall x)(Hx \supset Bx)$
  Stylistic variant: $(x)(Hx \supset Bx)$
- This is read: 'For all x, if Hx, then Bx.'
- Regarding (4):
  $(\exists x)Lax$
- This is read:'There exists an x, such that Lax'
- We call $(\forall x)$ and $(\exists x)$ in our $\mathcal{L}_P$ formulae quantifiers.
- So we have some new symbols:
  - Two symbols for quantification: $\forall$ and $\exists$
  - A set of variables: $x, y, z, \ldots, x_1, y_2, z_3, \ldots$
- But we also need syntactic rules for our new wfs's...

Quantification in $\mathcal{L}_P$ (ctd.)

- For this, we need the following piece of notation:
  Where φ is a wfs, a is a name, and x is a variable, we write '$\varphi(a := x)$' to denote the result of replacing all occurences of a in φ by x.

Substitution of variables

$Lab(a := x) = Lxb$

$((Fc \& Gd) \supset Fd)(d := y) = ((Fc \& Gy) \supset Fy)$

- We can now offer:
  If φ is a wfs, a is a name, and x is a variable, then $(\forall x)\varphi(a := x)$ and $(\exists x)\varphi(a := x)$ are both wfs's.

Quantification in $\mathcal{L}_P$ (ctd.)

Derivation of some wfs's

$(\forall x)(Hx \supset Bx)$ is a wfs:

- $(Ha \supset Ba)$ is a wfs
- $(Ha \supset Ba)(a := x) = (Hx \supset Bx)$

$(\exists x)(\forall y)(Py \supset Hyx)$ is a wfs:

- $(Pa \supset Hab)$ is a wfs
- $(Pa \supset Hab)(a := y) = (Py \supset Hyb)$
- So $(\forall y)(Py \supset Hyb)$ is a wfs
- $(\forall y)(Py \supset Hyb)(b := x) = (\forall y)(Py \supset Hyx)$

Next session

- Topic: wrapping up quantifiers and first part of semantics for $\mathcal{L}_P$.
- Reading: Restall, Ch. 8, from 'Translation' onwards + Ch. 9, up to, but excluding, 'Quantifiers'.