ELEMENTS OF DEDUCTIVE LOGIC

13. Predicate Logic: subsentential structure (ctd.) & semantics

J. Chandler

KUL 2012

Introduction

- Last time:
  - Validity by virtue of subsentential form
  - A language $\mathcal{L}_P$ of subsentential forms
    - Names and predicates
    - Universal and existential quantification
- This time:
  - Leftovers on quantification
  - Semantics for $\mathcal{L}_P$

Scope & binding

- An important bit of terminology:

  Where φ is a wfs, or the result of substituting variables for some of the names in a wfs, the scope of the quantifiers $(\forall x)$ and $(\exists x)$ in, respectively $(\forall x)\varphi$ and $(\exists x)\varphi$, is φ.
- Furthermore:

  An occurence of a variable x is bound by a given quantifier $(\forall x)$ or $(\exists x)$ iff it occurs within its scope.
- Finally:

  An occurence of a variable x that is *not* bound by a quantifier is called 'free'.
- Note that *no $\mathcal{L}_P$ wfs's have free variables*.
- Some examples…

Scope & binding (ctd.)

Scope & binding

$(\exists x)Dx\&Hx$ (This is different from $(\exists x)(Dx\&Hx)$!)

First occurence of x is bound: within scope Dx of $\exists$.

Second occurence is free.

$(\forall y)(Py \supset (\exists z)(Lzy))$

Single occurence of z bound by $(\exists z)$

Both occurences of y bound by $(\forall y)$

Translation

- Translation of quantified sentences into $\mathcal{L}_P$ can be *tricky*: it takes *practice*.
- Peter Suber gives some useful bits of advice here:

  www.cse.buffalo.edu/~rapaport/191/S09/transtip-pnllogic.html

  (Note: he uses '→' for '⊃', '↔' for '≡' and '∧' for '&')
- Also included: tips for translation into $\mathcal{L}_P$.
- We'll have a whole exercise session devoted to this.
- In the meanwhile, some quick comments and examples.

Translation (ctd.)

- The advice about paraphrase is particularly important here.

Paraphrasing first

'Apples and pears are fruit' ⇒ 'If anything is either an apple or a pear, then it is a fruit.' ⇒ $(\forall x)((Ax \vee Px) \supset Fx)$

($\not\Rightarrow$ $(\forall x)((Ax \& Px) \supset Fx)$, in spite of what some might be tempted to write!)

'All is not resolved yet.' ⇒ 'Something is not resolved yet.' ⇒ $(\exists x) \sim Rx$

($\not\Rightarrow$ $(\forall x) \sim Rx$, in spite of what the surface form might suggest!)

Translation (ctd.)

Some useful translation schemas

All A's are B's ⇒ $(\forall x)(Ax \supset Bx)$

No A's are B's ⇒ $(\forall x)(Ax \supset \sim Bx)$

Note: This is equivalent to 'All A's are not B's'

Only A's are B's ⇒ $(\forall x)(Bx \supset Ax)$

Some A's are B's ⇒ $(\exists x)(Ax \& Bx)$

Note #1: this is equivalent to 'Not all A's are not B's'

Note #2: we *don't* translate as $(\exists x)(Ax \supset Bx)$

Only some A's are B's ⇒ $(\exists x)(Ax \& Bx) \& (\exists y)(Ay \& \sim By)$

All and only A's are B's ⇒ $(\forall x)(Ax \equiv Bx)$

Translation (ctd.)

- Remarks regarding 'some':
  - Here, we assume that 'some A's are B's' = 'some A *is* B'.
  - 'some' ≠ 'some... but not all':

    'I took some money' conversationally *implies*, but *does not entail*, 'I did not take all the money'.

Translation (ctd.)

- Very often we will need multiple quantifiers. In this case, it is helpful to proceed in stages.

Formalising multiple quantification

'Every cloud has a silver lining.'

$\Rightarrow$ 'It is true of any thing, call it x, that if x is a cloud, then x has a silver lining'

$\Rightarrow$ 'It is true of any thing, call it x, that if x is a cloud, then there is a thing, call it y, such that y is a silver lining and x has y'

$\Rightarrow (\forall x)(Cx \supset (\exists y)(Sy \& Hxy))$

- Use new variables when you introduce new quantifiers. Not *always* necessary, but better safe than sorry!

The language $\mathcal{L}_P$: overview

- Symbols:
  - Names: $a, b, c, \ldots, a_1, b_2, c_3, \ldots$
  - Variables: $x, y, z, \ldots, x_1, y_2, z_3, \ldots$
  - Predicates: $F, G, H, \ldots, F_1, G_2, H_3, \ldots$
  - Quantification symbols: $\forall, \exists$
  - Connectives: $\sim, \&, \vee, \supset, \equiv$
- Rules for wfs's:
  - 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.
  - 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.
  - If φ and ψ are wfs's, so too are $\sim \varphi$, $(\varphi \& \psi)$, $(\varphi \vee \psi)$, $(\varphi \supset \psi)$ and $(\varphi \equiv \psi)$.
  - Nothing else is a wfs.

Validity and models

- In prop. logic, we had (semantic) validity defined in terms of possible assignments of truth values to sentences:

  > An argument is valid iff there is no possible valuation that assigns to all premises a designated value but assigns to the conclusion a non-designated value.

- The same is true here.
- But in prop. logic: assignments were *only* constrained by truth tables for connectives.
- In particular: the truth values of distinct connective-free wfs's were independent.
- So we could have, for any atomic wfs's p and q, any combinations of truth values: $v(p) = 1$ and $v(q) = 0$, or $v(p) = 0$ and $v(q) = 1$, etc.

Validity and models (ctd.)

- Because we pay attention to subsentential structure here, this no longer holds.
- We can't have, say: $v((\forall x)Fx) = 1$ and $v((\exists x) \sim Fx) = 1$.
- To capture these extra constraints, we appeal to something called a 'model'.

Validity and models (ctd.)

- A model for predicate logic is a pair $M = \langle D, I \rangle$ consisting of
  - a domain D of discourse, which is a non-empty set of individuals. ('the world')
  - an interpretation function I, which gives us the meanings of the different names and predicates. ('how the language connects to the world')
- Note: talk of *pair* (aka '2-tuple'), rather than *set*, using angle brackets $\langle\rangle$ rather than curly brackets $\{\}$ to list members.
- What is the difference?
  - A set: an *unordered* collection of objects. Two sets are the same iff they have the same members: $\{a,b\} = \{b,a\}$.
  - A tuple: an *ordered* collection, or sequence, of objects. Having the same members isn't sufficient for identity: $\langle a,b\rangle \neq \langle b,a\rangle$.

Interpretations: names

- The interpretation function maps names and predicates onto two *different* respective kinds of things.
- Names are mapped onto members of D: their 'denotations'.
- Intuitively: I tells us the meanings of the names, namely what or who the names refer to.
- *Note #1*: two different names can be mapped onto the same individual; we allow for synonymy (// 'Clark Kent' and 'Superman').
- *Note #2*: all names are mapped onto something; we do not allow for 'failure of denotation'.
- *Note #3*: all names are only mapped onto one thing; we do not allow for ambiguous names.

Interpretations: names (ctd.)

Names and domains

Let $D = \{d,k,e\}$

Let the set of names be $\{a,b,c\}$

We could have:

I	
a	d
b	k
c	e

So, for instance, we have $I(a) = d$.

Interpretations: predicates

- n-ary predicates are mapped onto functions from n-tuples of members of D to $V = \{0,1\}$.
- The basic idea: the meaning of a predicate is captured by the truth values of the sentences that it yields when it takes different names as inputs.
- So, the meaning of '... is a philosopher' is captured by the truth values of:

  'Descartes was a philosopher.' (true)
  'Kant was a philosopher.' (true)
  'Einstein was a philosopher.' (false)
  etc.
- Again, we have the assumptions of possibility of synonymy, success of denotation and lack of ambiguity.

Interpretations: predicates (ctd.)

Predicates

D and set of names as in previous example.
Set of predicates: $\{P, R\}$, with P unary and R binary.
We could have:

$I(P)$	
d	1
k	1
e	0

$I(R)$	d	k	e
d	1	0	0
k	1	1	0
e	1	0	1

Think: d = Descartes, k = Kant, e = Einstein, P = '... is a *p*hilosopher', R ='... has *r*ead the works of...'.
So, for instance, we have $I(P)(\langle d \rangle) = 1$ and $I(R)(\langle k, e \rangle) = 0$.

Next session

- Translation exercises: check Toledo.
- Session after that: Second part of semantics for $\mathcal{L}_P$.
- Reading: Restall Ch. 9, from 'Quantifiers' onwards.