Introduction to Logic

Jake Chandler

The Language of Predicate Logic

Last week

- The general idea behind a faster, more efficient proof method known as tableaux
- Some examples of the method in action
- The different “resolving rules”
- The concepts of open and closed branches / tableaux
- The concepts of soundness and completeness and their application to tableaux for sentential form

Introduction

Further validities

- The validity of *many* arguments isn’t due to sentential form

Hippos

(P_1) All hippos are bad-tempered.

(P_2) Andrew is a hippo.

(C) Andrew is bad-tempered.

Sentential form:

(P_1) p

(P_2) q

(C) r

Further validities (ctd.)

Cats

(P_1) Some cats are not friendly.

(P_2) All cats are mammals.

(C) Some mammals are not friendly.

Sentential form:

(P_1) p

(P_2) q

(C) r

Sub-sentential form

- But in the cases we've just seen, this validity is *still* due to form

Hippos

(P_1) All hippos are bad-tempered.

(P_2) Andrew is a hippo.

(C) Andrew is bad-tempered.

Corners

(P_1) All square have four corners.

(P_2) This shape is a square.

(C) This shape has four corners.

A language of sub-sentential form

- These validities are due to sub-sentential form: we need to look 'inside' the atomic sentences
- To do this, we move to $\mathscr{L}_P$: the "language of predicate logic".
- Inherited from $\mathscr{L}_P$.
  - Connectives: $\sim, \&, \vee, \supset, \equiv$
  - 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 $\mathscr{L}_S$: all wfs's now have internal structure

Names and Predicates

Names and predicates

- The second premise of the 'Hippo' argument:

  'Harry is a hippo.'

- It involves:
  - a name, aka 'designator' ('Harry'), referring to a particular thing
  - a predicate ('…is a hippo'), to saying something about what that thing is like

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

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

- Two new kinds of symbols in $\mathscr{L}_P$:
  - For names, aka 'constants': $a, b, c, \ldots, a_1, b_2, c_3, \ldots$
  - For predicates: $F, G, H, \ldots, F_1, G_2, H_3, \ldots$
- And a syntactic 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)
- In CS, words are often used, e.g.: $\mathsf{Impala}(\mathsf{Carmen})$
- Note: order matters! ($Lab \neq Lba$)

Translation

- As with sentential logic, we create a "dictionary" giving each name/predicate a lower/upper case letter

Subsentential form

'John liked Samantha, and Karl liked Trisha.'

Dictionary:

j = John

s =Samantha

k = Karl

t = Trisha

L = …liked …

Translation: $Ljs\&Lkt$

Translation (ctd)

- Again, 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.' $\Rightarrow$ 'John loves John and Karl loves Karl.' $\Rightarrow$ $Ljj\&Lkk$

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

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

Translation (ctd.)

- Use predicates of the right arity.

The right arity

'Mary hated John.' $\Rightarrow$ 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 one single predicate for both forms

Mixed passive/active

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

Introduction to Logic

Jake Chandler

The Language of Predicate Logic

LA TROBE UNIVERSITY

Quantification

Quantifiers

- The first premise of our introductory argument:

  '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*'

Quantifiers (ctd)

- In $\mathscr{L}_P$, we will just have translations for sentences involving:
  - 'all' (aka universal quantification)
  - 'at least one' (aka existential quantification)

Universal quantifiers

- What exactly do we *mean* by

  '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.

- The representation of the premise in predicate logic has that structure:

  $(\forall x)(Hx \supset Bx)$ (common variant: $(x)(Hx \supset Bx)$)

- Note: x is known as a "variable" and $\forall$ as a "quantifier symbol" (the "universal quantifier")

Existential quantifiers

- Similarly, consider:

  'Some cats are not friendly.'

- Gloss:

  It is true of *at least one* thing, call it 'x', that it is a cat and that it is not friendly.

- Again, same structure in the predicate logic representation:

  $(\exists x)(Cx \& \sim Fx)$

- $\exists$ is also a quantifier symbol (the "existential quantifier")
- So we have some new symbols:
  - Two symbols for quantification: $\forall$ and $\exists$
  - A set of variables: $x, y, z, \ldots, x_1, y_2, z_3, \ldots$

Substitution and wfs's

- But we also need syntactic rules for our new wfs's...
- For this, we need the following piece of notation:

  Where φ is a wfs, a a name, and x a variable, '$\varphi(a := x)$' denotes the result of replacing all occurrences of a in φ by x.

- Note: $\varphi(a := x)$ is typically not itself a wfs!!!

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

Substitution and wfs's (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
- $(\forall y)(Py \supset Hyb)(b := x) = (\forall y)(Py \supset Hyx)$

Translation

- Translation of quantified sentences into $\mathscr{L}_P$ can be *tricky*: it takes *practice*.
- Peter Suber gives advice here
- The advice about paraphrase is particularly important here.

Paraphrasing first

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

$(\not\Rightarrow (\forall x)((Ax\&Px) \supset Fx)!)$

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

$(\not\Rightarrow (\forall x) \sim Rx!)$

Translation (ctd.)

Some useful translation schemas

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

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

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

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

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

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 $\Rightarrow (\exists x)(Ax\&Bx)\&(\exists y)(Ay\& \sim By)$

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

Translation (ctd.)

- Remarks regarding 'some':
  - Here, we assume that 'some A's are B's' = 'some A *is* B'.
  - 'some' $\neq$ '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!

$\mathscr{L}_P$ at a glance

- 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:

(1) 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

(2) If P is a wfs, a is a name, and x is a variable, then $(\forall x)P(a := x)$ and $(\exists x)P(a := x)$ are both wfs's

(3) If P and Q are wfs's, so too are $\sim P$, $(P \& Q)$, $(P \vee Q)$, $(P \supset Q)$ and $(P \equiv Q)$

(4) Nothing else is a wfs

Introduction to Logic

Jake Chandler

The Language of Predicate Logic

Truth in $\mathscr{L}_P$

Validity and models

- As previously, we define validity in terms of possible assignments of truth values to sentences:

  > An argument is valid iff there is no possible valuation that assigns "true" to all premises but assigns "false" to the conclusion.

- But previously, these assignments were *only* constrained by the t. tables for connectives
- In particular: the truth values of wfs's with no shared atomic sentences were independent

Independence and $\mathscr{L}_S$

For any atomic p, q in $\mathscr{L}_S$, there exist valuations v_1 and v_2, such that $v_1(p) = 1$ and $v_1(q) = 0$, but $v_2(p) = 0$ and $v_2(q) = 1$

Validity and models (ctd.)

- Because we pay attention to subsentential structure here, this no longer holds.

Independence and $\mathscr{L}_P$

There does not exist a valuation v such that $v((\forall x)Fx) = 1$ 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 $M = \langle D, \cdot^I \rangle$ consisting of
  - a domain D of discourse, which is a non-empty set of individuals. ('the world')
  - an interpretation function $\cdot^I$, which gives us the meanings of the different names and predicates in terms of elements of D ('how the language connects to the world')

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: $\cdot^I$ tells us the meanings of the names, namely what or who the names refer to
- Remarks:
  (1) Two different names can be mapped onto the same thing: synonymy is OK (e.g. 'Clark Kent' and 'Superman')
  (2) All names are mapped onto some thing: no failure of denotation
  (3) All names are only mapped onto one thing: no ambiguity
  (4) Every thing has a name: no anonymity

Interpretations: predicates

- n-ary predicates are mapped onto sets of n-tuples of members of D, e.g.:
  - Unary predicates are mapped onto the individuals that have the relevant property
  - Binary predicates are mapped onto the pairs of individuals that stand in the relevant binary relation
- Again $\cdot^I$ tells us the meanings of the predicates, conceived of as referring to sets of tuples
- Our previous assumptions (1)–(3) hold again ((4) isn't required)

Interpretations: names and predicates

Vocabulary of *African Wildlife*

Names:

$\{\mathsf{Leo}, \mathsf{Immy}, \mathsf{Moses}, \mathsf{Patrick}\}$

Unary predicates:

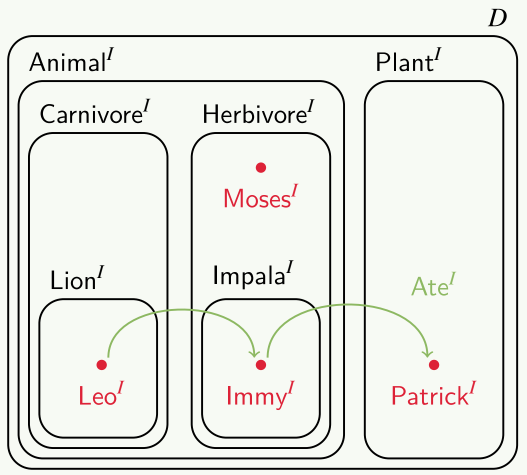
$\{\mathsf{Animal}, \mathsf{Plant}, \mathsf{Carnivore}, \mathsf{Herbivore}, \mathsf{Lion}, \mathsf{Impala}\}$

Binary predicate:

$\{\mathsf{Ate}\}$

Interpretations: names and predicates

Model of *African Wildlife* (diagrammatically)



Interpretations: names and predicates (ctd.)

Model of *African Wildlife* (in writing)

$D = \{\mathsf{Leo}^I, \mathsf{Moses}^I, \mathsf{Immy}^I, \mathsf{Patrick}^I\}$
$\mathsf{Animal}^I = \{\mathsf{Leo}^I, \mathsf{Moses}^I, \mathsf{Immy}^I\}$
$\mathsf{Plant}^I = \{\mathsf{Patrick}^I\}$
$\mathsf{Carnivore}^I = \{\mathsf{Leo}^I\}$
$\mathsf{Herbivore}^I = \{\mathsf{Moses}^I, \mathsf{Immy}^I\}$
$\mathsf{Lion}^I = \{\mathsf{Leo}^I\}$
$\mathsf{Impala}^I = \{\mathsf{Immy}^I\}$
$\mathsf{Ate}^I = \{\langle \mathsf{Leo}^I, \mathsf{Immy}^I\rangle, \langle \mathsf{Immy}^I, \mathsf{Patrick}^I\rangle\}$

Truth in a model

- We said models are invoked to place constraints on possible truth value assignments.
- More precisely: a truth value assignment is possible iff it gives us the truth values of the sentences in $\mathscr{L}_P$ in some model M.
- We denote by v_M the function that returns the truth values of the sentences in $\mathscr{L}_P$ in M (when model clear from context: just "v")
- When $v_M(\varphi) = 1$ we say that M satisfies or is a model of φ (common notation: $M \Vdash \varphi$)
- What makes a sentence true or false in a model?

Truth of simple sentences

- For simple sentences, i.e. sentences without quantifiers or connectives:

  Where F is an n-ary predicate and $a_1, \ldots, a_n$ are names:
  $v(Fa_1 \ldots a_n) = 1$ if $\langle a_1^I, \ldots, a_n^I\rangle \in F^I$, and $v(Fa_1 \ldots a_n) = 0$ otherwise

Values of simple sentences in *African Wildlife*

$v(\mathsf{Lion}(\mathsf{Leo})) = 1$, since $\mathsf{Leo}^I \in \mathsf{Lion}^I$
$v(\mathsf{Plant}(\mathsf{Immy})) = 0$, since $\mathsf{Immy}^I \notin \mathsf{Plant}^I$
$v(\mathsf{Ate}(\mathsf{Immy}, \mathsf{Patrick})) = 1$, since $\langle \mathsf{Immy}^I, \mathsf{Patrick}^I\rangle \in \mathsf{Ate}^I$

Quantified sentences

- We now move on to universal and existential sentences, i.e. sentences of the form $(\forall x)A$ or $(\exists x)A$, where $A = \varphi(a := x)$ for some name a and wfs φ.
- A sentence can contain a universal/existential quantifier without being a universal/existential sentence!
- Compare: A sentence can contain "&" without being a conjunction

Universal sentences

Universal sentences:

$(\forall x)Fx$
$(\forall x)(\exists y)(Fx \supset Gy)$
$(\forall x)(Fx \& (\forall x)Gx)$

Not a universal sentence:

$((\forall x)Fx \& (\forall x)Gx)$ ($Fa \& (\forall x)Gx$ is not a wfs! Needs outer brackets)

Instances

- An important definition and piece of notation:

  For any sentence φ of the form $(\forall x)A$ or $(\exists x)A$, we denote by $A(x := a)$ the result of

  (i) substituting a for every *free* occurence of x in A

  (ii) subsequently deleting the leftmost quantifier $(\forall x)$ or $(\exists x)$.

  We say that $A(x := a)$ is an instance of φ.

Instances (ctd.)

Instances vs non-instances

Consider $(\forall x)(\exists y)(Fx \supset Gy\&Hx)$:

$(\exists y)(Fa \supset Gy\&Ha)$ is an instance.

$(\exists y)(Fa \supset Gy\&Hb)$ is *not* an instance (two occurrences replaced by different names).

Consider $(\exists x)(Fx\& \sim (\forall x)Gx)$:

$Fb\& \sim (\forall x)Gx$ is an instance.

$Fb\& \sim (\forall x)Gb$ is *not* an instance ('b' replaced an instance of 'x' that wasn't free in '$(Fx\& \sim (\forall x)Gx)$')

Truth of quantified sentences

- We can now offer:

  $v((\forall x)A) = 1$ iff for every name a, $v(A(x := a)) = 1$
  A universal sentence is true iff all its instances are

  $v((\exists x)A) = 1$ iff for at least one name a, $v(A(x := a)) = 1$
  An existential sentence is true iff it has at least one true instance

Values of quantified sentences in *African Wildlife*

$v((\exists x)(\mathsf{Herbivore}(x)\& \sim \mathsf{Impala}(x)) = 1$, since

$v(\mathsf{Herbivore}(\mathsf{Moses})\& \sim \mathsf{Impala}(\mathsf{Moses})) = 1$

$v((\forall x)(\mathsf{Lion}(x) \supset \mathsf{Carnivore}(x)) = 1$, since:

$v(\mathsf{Lion}(\mathsf{Leo}) \supset \mathsf{Carnivore}(\mathsf{Leo})) = 1$

$v(\mathsf{Lion}(\mathsf{Immy}) \supset \mathsf{Carnivore}(\mathsf{Immy})) = 1$

etc.

Truth of arbitrary sentences

- Finally, the usual constraints are imposed by the truth tables for the connectives.
- With this, we can now compute the truth value of any $\mathscr{L}_P$ sentence in any model.

Values of arbitrary sentences in *African Wildlife*

$v((\forall x)((\exists y)\mathsf{Ate}(y,x) \supset (\exists z)\mathsf{Ate}(x,z))) = 0$, since it has a false instance:

$v((\exists y)\mathsf{Ate}(y,\mathsf{Patrick}) \supset (\exists z)\mathsf{Ate}(\mathsf{Patrick},z)) = 0$

This instance is false because of the t. table for $\supset$ and the fact that we have

$v((\exists y)\mathsf{Ate}(y,\mathsf{Patrick})) = 1$ (has true instance)

$v((\exists z)\mathsf{Ate}(\mathsf{Patrick},z)) = 0$ (has no true instance)