ELEMENTS OF DEDUCTIVE LOGIC

14. Predicate Logic: semantics (ctd.) + tableau methods

J. Chandler

KUL 2012

Introduction

- Last time:
  - Translation of quantified NL sentences
  - Models, domains and interpretations
- This time: wrapping up the semantics for $\mathcal{L}_P$ + introducing tableaux
  - Truth in a model: the general case
  - Truth in *finite* models
  - Testing for validity: a special case
  - Testing for validity: the general case (tbc)

Truth in a model

- As I said earlier, 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 $\mathcal{L}_P$ in some model M.
- We denote by v_M the function that returns the truth values of the sentences in $\mathcal{L}_P$ in M.
- Ok, so what makes a sentence true or false (aka 'satisfied' or 'not satistied') in a model?
- We start with atomic sentences, i.e. sentences without quantifiers or connectives:

  Where F is an n-ary predicate and $a_1,\ldots,a_n$ are names:
  $v_M(Fa_1\ldots a_n) = 1$ iff $I(F)(\langle I(a_1),\ldots,I(a_n)\rangle) = 1$.

Truth in a model (ctd.)

Values of atomic sentences

Domain: $\{d,k,e\}$; Names: $\{a,b\}$; Predicates: $\{P,R\}$. Let $I(a) = d$ and $I(b) = k$ and, as before:

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

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

$v_M(Pa) = 1$, since $I(P)(\langle I(a)\rangle) = I(P)(\langle d\rangle) = 1$

$v_M(Rab) = 0$, since $I(R)(\langle I(a),I(b)\rangle) = I(R)(\langle d,k\rangle) = 0$

Truth in a model (ctd.)

- 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.

Universal sentences

Universal sentences:

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

Not a universal sentence:

$((\forall x)Fx \& (\forall x)Gx)$
(Since $Fa \& (\forall x)Gx$ is not a wfs: we need outer brackets)

Truth in a model (ctd.)

- 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.

Truth in a model (ctd.)

Instances

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

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

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

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

$Fb \& \sim (\forall x)Gx$ 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 in a model (ctd.)

- Given this, we can now offer:
  (i) $v_M((\forall x)A) = 1$ iff for every name a, $v_M(A(x := a)) = 1$.
  (ii) $v_M((\exists x)A) = 1$ iff for at least one name a, $v_M(A(x := a)) = 1$.
- But what if we have fewer names than domain elements? Nothing we have said so far precludes this (see our first example).
- This would intuitively land us in trouble:
  - By (i), in our first example, $v_M((\forall x)Px) = 1$: all the instances, Pa and Pb, are true.
  - But $I(P)(\langle e \rangle) = 0$: Einstein is no philosopher. So surely we should have $v_M((\forall x)Px) = 0$.
- So we now simply stipulate that this does not happen: in every model, every domain element is named.

Truth in a model (ctd.)

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

Truth in a model (ctd.)

Values of complex sentences

M as in our first example, except for extra name c for element e.

We show that $v((\forall x)(Rxa \supset (\exists y)(Py\&Ryx))) = 0$:

- Instances:

  (1) $(Raa \supset (\exists y)(Py\&Rya))$; (2) $(Rba \supset (\exists y)(Py\&Ryb))$; (3) $(Rca \supset (\exists y)(Py\&Ryc))$

- One of these instances, namely (3), is false:

  Its antecedent Rca is true, since $I(R)(\langle e,d\rangle) = 1$.

  Its consequent $(\exists y)(Py\&Ryc)$ is false, since all its instances are false:

  (4) $(Pa\&Rac)$; (5) $(Pb\&Rbc)$; (6) $(Pc\&Rcc)$

Truth in finite models

- Two interesting observations:
  - (i) In finite models, a universal sentence is true iff the conjunction of its instances is.
  - (ii) In finite models, an existential sentence is is true iff the disjunction of its instances is.
- Upshot: in finite models, any sentence can ultimately be 'rewritten' without any quantifiers whatsoever.
- This is probably not entirely surprising. Consider (i):
  - A universal sentence is true iff all its instances are.
  - A conjunction is true iff all its conjuncts are.
  - If the model is finite, the number of instances of a universal sentence will be finite.
  - If the number of instances is finite, they can be listed as conjuncts in a finite conjunction.

Truth in finite models (ctd.)

- Why doesn't this hold for non-finite models?
- Simply because:
  - In a non-finite model, assuming that every member of the domain receives a different name, as we do here, there will be an infinite number of instances
  - So we would need an infinite conjunction/disjunction
  - But this isn't allowed by the syntax of our language $\mathcal{L}_P$.

Testing for validity: a special case

- Definition of validity in (2-valued) predicate logic:

  An argument is valid iff there is no model M such that v_M assigns to all premises a value of 1 but assigns to the conclusion a value of 0.

- Checking for validity using the method that we used–building a model and considering v_M–isn't feasible in general.
- In *many* cases, we would need to check *all* models of *all* sizes.
- But as Restall notes, this isn't true of *all* cases.
- Interesting observation (see Restall pp. 144-145 for proof):

  An argument *containing only n monadic predicates* is valid iff there is no model *M with a domain of size* 2^n such that v_M assigns to all premises a value of 1 but assigns to the conclusion a value of 0.

Testing for validity: a special case (ctd.)

- Furthermore, the previous observation wrt truth in finite models suggests a quick and familiar way of testing for validity in models with domains of size 2^n, since these are finite.
- Here's how:
  (1) Translate any universal or existential sentences into quantifier-free sentences.
  (2) Use the tableau rules to check for validity.
  (3) Read countermodels off any open branch.
- So we have a quick and faniliar way of testing for validity of arguments containing only n monadic predicates.
- See Restall pp. 136-143 for more on this.

General comments

- What about testing for validity in the general case?
- Here is a new, sound and complete (see Restall pp. 161-164), tableau method.
- Similar principle to before, given our classical (2-valued) semantics:

  Try to find a model M such that its associated valuation v_M assigns the value 1 to each of the premises but the value 0 to the conclusion.

- Same (a) way of starting the tree, (b) closing rules and (c) rules for the connectives (propositional/sentential rules) as in classical sentential logic.
- But we add some new rules for universal and existential sentences, both negated and non-negated (instantiation rules).

Existential sentences

- The rule:

$$\begin{array}{c}(\exists x)A\\ \hline A(x := a)\\ (\text{new } a)\end{array}$$

- Rationale:
  - If $(\exists x)A$ is true, then one of its instances is true.
  - This might not be an instance that involves a name already used on the branch. So we introduce a new name, just in case.

Negated existential sentences

- The rule:

$$\begin{array}{c} \sim (\exists x)A \\ | \\ \sim A(x := a) \end{array}$$

(any a already on branch)

- Rationale:
  - If $\sim (\exists x)A$ is true, then $(\exists x)A$ is false, so all the instances of the latter are false and hence their negations are all true.
  - This includes all those negations involving names already used on the branch.

Next session

- Topic: tableau methods (ctd.)
- Reading: Restall, Ch. 10.