Mathematics for linguists

WS 2009/2010

University of Tübingen

January 28, 2010

Gerhard Jäger

Examples

Side remark: if the truth value of a formula in a model does not depend on the assignment function, the assignment function index can be omitted. Instead of $[\varphi]^M_g$ we simply write $[\varphi]^M$.

- $[\exists x \textit{Animal}(x)]^M$
- $[\exists x(\textit{Animal}(x) \wedge \textit{Run}(x))]^M$
- $[\exists x(\textit{Animal}(x) \rightarrow \textit{Run}(x))]^M$
- $[\forall x(\textit{Animal}(x) \rightarrow \textit{Run}(x))]^M$
- $[\exists x \textit{Scream}(x)]^M$

Examples

Side remark: if the truth value of a formula in a model does not depend on the assignment function, the assignment function index can be omitted. Instead of $[\varphi]_g^M$ we simply write $[\varphi]^M$.

- $[\exists x \textit{Animal}(x)]^M = 1$
- $[\exists x(\textit{Animal}(x) \wedge \textit{Run}(x))]^M = 1$
- $[\exists x(\textit{Animal}(x) \rightarrow \textit{Run}(x))]^M = 1$
- $[\forall x(\textit{Animal}(x) \rightarrow \textit{Run}(x))]^M = 0$
- $[\exists x \textit{Scream}(x)]^M = 0$

Undecidability

- for finite models the truth value can always be determined
- in infinite models, it is not always possibel to determine the truth value of a formula
  - example: prime twins
  - model: system of natural numbers
  - truth value of the following formula (with the intended interpretation of the predicates) is unknown:

$$\forall x \exists y \exists z (x < y \wedge \textit{Prime}(y) \wedge \textit{Prime}(z) \wedge \textit{Plus}(y, 2, z))$$

Inference

- central notion for logic is **inference**
- truth is actually an auxiliary notion
- how can inference in predicate logic be determined?

Logical inference

Definition 2 (Logical inference) *From the premises* $\varphi_1, \ldots, \varphi_n$ *the conclusion* ψ *follows logically – formally written as*

$$\varphi_1 \ldots, \varphi_n \Rightarrow \psi$$

if and only if for all models M *and all assignment functions* g *it holds that: if* $[\varphi_i]_g^M = 1$ *for all* $1 \leq i \leq n$*, then also* $[\psi]_g^M = 1$*.*

- the definitions from statement logic for the other logical properties and relations can directly be applied to predicate logic as well:

Tautologies

Definition 4 (Tautology) *A formula* φ *is a predicate logical* ***tautology****, formally written as*

$$\Rightarrow \varphi$$

if and only if for all models M *and all assignment function* g *it holds:*

$$[\varphi]_g^M = 1$$

Contradictions

Definition 6 (Contradiction) *A formula φ is a predicate logical **Contradiction** if and only if for all models M and all assignment functions g it holds:*

$$[\varphi]_g^M = 0$$

Logical equivalence

Definition 8 (Logical equivalence) *Two formulas* φ *and* ψ *are* ***logically equivalent*** *— formally written as*

$$\varphi \Leftrightarrow \psi$$

if and onl if for all model M *and all assignment functions* g *it holds that:*

$$[\varphi]_g^M = [\psi]_g^M$$

- the meta-logical theorems of statement logic (cf. slides from December 15) hold for predicate logic as well
- How do we show that for instance a formula is a tautology?
- Example:

$$\stackrel{?}{\Rightarrow} \forall x \neg P(x) \rightarrow \neg \exists y P(y)$$

- two *semantic Methods*:
  - reformulate as a set-theoretical statement
  - try to construct a falsifying model

Reduction to set theory

to be proven:

- for all M and g: $[\forall x \neg P(x) \rightarrow \neg \exists y P(y)]_g^M = 1$

step-wise reformulation (successive application of the semantic definitions)

1. for all M and g: $\max([1 - [\forall x \neg P(x)]_g^M, [\neg \exists y P(y)]_g^M) = 1$
2. for all M and g: $\max([1 - \min_{a \in E}([\neg P(x)]_{g[a/x]}^M), 1 - [\exists y P(y)]_g^M) = 1$
3. for all M and g: $\max([1 - \min_{a \in E}(1 - [P(x)]_{g[a/x]}^M), 1 - [\max_{b \in E}([P(y)]_{g[b/y]}^M)) = 1$
4. for all M and g: $\max([\max_{a \in E}([P(x)]_{g[a/x]}^M), 1 - [\max_{b \in E}([P(y)]_{g[b/y]}^M)) = 1$

Reduction to set theory

- the last line essentially says: for a specific truth value α:

$$\max(\alpha, 1 - \alpha) = 1$$

- this is of course always true
- hence the original formula is a tautology
- *method is tedious and sometimes not very illuminating*

Constructing a counter model

- alternative method: construct a falsifying model
- basic idea: indirect proof
  - suppose the formula is not a tautology
  - this means that there is a model and an assignment function that make the formula false
  - we try to construct such a model (and an appropriate assignment function)
  - if this attempt fails, the formula must be a tautology

Constructing a counter model

- Suppose: there are M and g such that $[\forall x \neg P(x) \rightarrow \neg \exists y P(y)]_g^M = 0$
- Hence: $[\forall x \neg P(x)]_g^M = 1$ and $\neg \exists y P(y)]_g^M = 0$
- Hence: $[\forall x \neg P(x)]_g^M = 1$ and $[\exists y P(y)]_g^M = 1$
- Hence: $\min_{a \in E}([\neg P(x)]_{g[a/x]}^M) = 1$ and $\max_{b \in E}([P(y)]_{g[b/y]}^M) = 1$
- Hence: $\min_{a \in E}(1 - [P(x)]_{g[a/x]}^M) = 1$ and $\max_{b \in E}([P(y)]_{g[b/y]}^M) = 1$
- Hence: $\max_{a \in E}([P(x)]_{g[a/x]}^M) = 0$ and $\max_{b \in E}([P(y)]_{g[b/y]}^M) = 1$: **Contradiction**

Constructing a counter model

- Example for a non-tautology:

$$\forall x \exists y Rxy$$

- Assumption: there is a (counter) model M and an assignment g such that:
  - $[\forall x \exists y Rxy]^M_g = 0$
  - hence: $\min_{a \in E}[\exists y Rxy]^M_{g[a/x]}] = 0$
  - hence: for some $a \in E$: $[\exists y Rxy]^M_{g[a/x]}] = 0$
  - hence: $\max_{b \in E}[Rxy]^M_{g[a/x][b/y]} = 0$
  - hence: for all $b \in E$: $[Rxy]^M_{g[a/x][b/y]} = 0$
  - hence: for all $b \in E$: $\langle a, b \rangle \notin F(R)$

Constructing a counter model

- simplest model with these properties:
  - $M = \langle E, F \rangle$
  - $E = \{a\}$
  - $F(R) = \emptyset$
- counter model method can be automatized to a certain degree:
- **truth tree method for predicate logic**

Truth tree calculus for predicate logic

- all rules of the truth tree calculus for statement logic remain valid
- there are four new rules, two per quantifier

Rules

- universal quantifier

$$(\forall) \quad \frac{\forall x \varphi}{[c/x]\varphi}$$

where c is an arbitrary constant that **does occur** within the same branch. If no constant occurs in this branch so far, c can be freely chosen.

- existential quanifier

$$(\exists) \quad \frac{\exists x \varphi}{[c/x]\varphi}$$

where c is an arbitrary constant that **does not occur** within the same branch.

Rules

- negation + universal quantifier

$$(Neg + \forall) \qquad \begin{array}{c} \neg\forall x\varphi \\ \hline [c/x]\neg\varphi \end{array}$$

where c is an arbitrary constant that **does not occur** within the same branch.

- negation + existential quantifier

$$(Neg + \exists) \qquad \begin{array}{c} \neg\exists x\varphi \\ \hline [c/x]\neg\varphi \end{array}$$

where c is an arbitrary constant that **does occur** within the same branch. If no constant occurs in this branch so far, c can be freely chosen.

Rules

- The rules $(\exists)$ and $(\neg\forall)$ may only be applied once per formula.
- The rules $(\forall)$ and $(\neg\exists)$ can be applied with every constant that occurs in this branch.
- Rule of thumb: if you have the choice, first apply $(\exists)$ and $(\neg\forall)$, and apply $(\forall)$ and $(\neg\exists)$ later

Examples

1.	$\neg(\forall x \neg Px \rightarrow \neg\exists x Px)$	(A)
2.	$\forall x \neg Px$	(1)
3.	$\neg\neg\exists x Px$	(1)
4.	$\exists x Px$	(3)
5.	Pa	(4)
6.	$\neg Pa$	(2)
7.	x	$(5, 6)$

The assumption that $\forall x \neg Px \rightarrow \neg\exists x Px$ is false in a model, i.e. that the negation $\neg(\forall x \neg Px \rightarrow \neg\exists x Px)$ is true leads to a contradiction. Hence the original formula is a tautology.

Examples

$$\begin{array}{lcr} 1. & \neg\forall x \exists y Rxy & (A) \\ 2. & \neg\exists y Ray & (1) \\ 3. & Raa & (2) \end{array}$$

The branch remains open, even though no further rules can be applied. The formula $\forall x \exists y Rxy$ is thus not a tautology.

Inference and truth trees

- logical inferences can be proved using the truth tree calculus as well
- similary as in statement logic, for indirect proof we assume that
  - all premises are true, and
  - the conclusion is false
- hence a truth tree for an inference starts with the premises and the negation of the conclusion

Examples

$$\forall x P(x) \Rightarrow \forall y P(y)$$

1.	$\forall x P(x)$	(A)
2.	$\neg \forall y P(y)$	(A)
3.	$\neg P(a)$	(2)
4.	$P(a)$	(1)
5.	x	$(3, 4)$

Examples

$$\forall x(P(x) \rightarrow Q(x)) \Rightarrow \forall x P(x) \rightarrow \forall x Q(x)$$

1.	$\forall x(P(x) \rightarrow Q(x))$	(A)
2.	$\neg(\forall x P(x) \rightarrow \forall x Q(x))$	(A)
3.	$\forall x P(x)$	(2)
4.	$\neg \forall x Q(x)$	(2)
5.	$\neg Q(a)$	(4)
6.	$P(a)$	(3)
7.	$P(a) \rightarrow Q(a)$	(1)

8.	$\neg P(a)$	(7)	9.	$Q(a)$	(7)
	x	$(6, 8)$		x	$(5, 9)$

Examples

$$\exists x P(x) \not\Rightarrow P(a)$$

1.	$\exists x P(x)$	(A)
2.	$\neg P(a)$	(A)
3.	$P(a)$	(1)
	x	$(2, 3)$

WRONG!!

Examples

$$\exists x P(x) \not\Rightarrow P(a)$$

1.	$\exists x P(x)$	(A)
2.	$\neg P(a)$	(A)
3.	$P(b)$	(1)

CORRECT

Examples

$$\exists x \forall y R(x,y) \Rightarrow \forall y \exists x R(x,y)$$

1.	$\exists x \forall y R(x,y)$	(A)
2.	$\neg \forall y \exists x R(x,y)$	(A)
3.	$\forall y R(a,y)$	(1)
4.	$\neg \exists x R(x,b)$	(2)
5.	$R(a,b)$	(3)
6.	$\neg R(a,b)$	(4)
	x	$(5,6)$

Undecidability

$$\stackrel{?}{\Rightarrow} \exists x \forall y R(x,y)$$

$$\begin{array}{lcr}
1. & \neg\exists x \forall y R(x,y) & (A) \\
2. & \neg\forall y R(a,y) & (1) \\
3. & \neg R(a,b) & (2) \\
4. & \neg\forall y R(b,y) & (1) \\
5. & \neg R(b,c) & (2) \\
6. & \neg\forall y R(c,y) & (1) \\
7. & \neg R(c,d) & (2) \\
 & \vdots &
\end{array}$$

Undecidability

- branch can be extended arbitrarily often without ever encountering a contradiction
- it generally holds:
  - only logical inferences can be proved with this method (i.e. the calculus is **sound**)
  - for each logical inference there is a proof within the truth tree calculus (the calculus is **complete**)
  - there is no guarantee that a non-inference is recognized as such
  - procedure may enter infinite loops

Undecidability

- there are no other mechanical procedures either that always correctly distinguish inference from non-inferences within finite time
- inference in predicate logic is **undecidable**