lecture 14, syntax of MPL

phil1012 introductory logic

overview

this lecture

- the syntax of MPL
- formal specification of the vocabulary and syntax of MPL
- how formulas of MPL are constructed
- the notions of logical operators, the scope of a quantifier, free and bound variables, open and closed formulas

learning outcomes

- after doing the relevant reading for this lecture, listening to the lecture, and attending the relevant tutorial, you will be able to:
    - identify well-formed formulas of MPL
    - construct construction tables for formulas of MPL
    - identify the main operator of a formula of MPL
    - identify the scope of a quantifier
    - identify free and bound variables and open and closed formulas of MPL

required reading

- section 8.4 of chapter 8

syntax of MPL

syntax of MPL

- we can give a precise specification of the syntax of MPL as we did for PL.

- the **symbols** of MPL are:
    - names:
        - a,b,c,…,ta, b, c, \ldots, t
    - variables:
        - x,y,z,u,v,wx, y, z, u, v, w
    - predicates:
        - A,B,C,…,X,Y,ZA, B, C, \ldots, X, Y, Z
    - five connectives:
        - ¬∧∨→↔\lnot \hspace{6pt} \land \hspace{6pt} \lor \hspace{6pt} \rightarrow \hspace{6pt} \leftrightarrow

---

- the **symbols** of MPL (continued) are:
    - two quantifier symbols:
        - ∀∃\forall \hspace{12pt} \exists
    - two punctuation symbols (parentheses):
        - ()(\hspace{12pt})

- **terms** are defined as follows:
    - a name is a term
    - a variable is a term
    - nothing else is a term

- **wffs** of MPL are defined as follows:
    - (3i) where P_\underline{P} is a predicate and t_\underline{t} is a term, the following is a wff:
        - P_t_\underline{P}\underline{t}
    - (3ii) where α\alpha and β\beta are wffs and x_\underline{x} is a variable, the following are wffs:
        - ¬α\lnot \alpha, (α∧β)(\alpha \land \beta), (α∨β)(\alpha \lor \beta), (α→β)(\alpha \rightarrow \beta), (α↔β)(\alpha \leftrightarrow \beta), ∀x_α\forall \underline{x} \alpha, ∃x_α\exists \underline{x} \alpha
    - nothing else is a wff

constructing wffs

constructing wffs

- given this syntax for MPL we can show how any well formed formula of MPL is constructed.
- suppose we want to construct
    - (∀xRx→∃xPx)(\forall x Rx \rightarrow \exists x Px)
- we might construct it as follows

Step	Wff constructed at this step	From steps/by clause
1	RxRx	/(3i)
2	PxPx	/(3i)

Step	Wff constructed at this step	From steps/by clause
1	RxRx	/(3i)
2	PxPx	/(3i)
3	∀xRx\forall x Rx	1, /(3ii ∀\forall)
4	∃xPx\exists x Px	2, /(3ii ∃\exists)

Step	Wff constructed at this step	From steps/by clause
1	RxRx	/(3i)
2	PxPx	/(3i)
3	∀xRx\forall x Rx	1, /(3ii ∀\forall)
4	∃xPx\exists x Px	2, /(3ii ∃\exists)
5	(∀xRx→∃xPx)(\forall x Rx \rightarrow \exists x Px)	3,4 /(3ii →\rightarrow)

- a **logical operator** is a connective or a quantifier
- the main operator (c.f. main connective) is the last operator added in the construction of the wff
- any wff constructed along the way is a subformula

- suppose we want to construct
    - ∀x(Rx→∃xPx)\forall x (Rx \rightarrow \exists x Px).
- we might construct it as follows

Step	Wff constructed at this step	From steps/by clause
1	RxRx	/(3i)
2	PxPx	/(3i)

Step	Wff constructed at this step	From steps/by clause
1	RxRx	/(3i)
2	PxPx	/(3i)
3	∃xPx\exists x Px	2, /(3ii ∃\exists)

Step	Wff constructed at this step	From steps/by clause
1	RxRx	/(3i)
2	PxPx	/(3i)
3	∃xPx\exists x Px	2, /(3ii ∃\exists)
4	(Rx→∃xPx)(Rx \rightarrow \exists x Px)	1,3 /(3ii →\rightarrow)

Step	Wff constructed at this step	From steps/by clause
1	RxRx	/(3i)
2	PxPx	/(3i)
3	∃xPx\exists x Px	2, /(3ii ∃\exists)
4	(Rx→∃xPx)(Rx \rightarrow \exists x Px)	1,3, /(3ii →\rightarrow)
5	∀x(Rx→∃xPx)\forall x (Rx \rightarrow \exists x Px)	4/(3ii ∀\forall)

quantifier scope

quantifier scope

- if a wff has a quantifier in it, then it must have got there by being stuck on the front of some sub-formula α\alpha (by 3ii ∀\forall or 3ii ∃\exists) at some stage in the construction
- for any quantifier appearing in a wff, we call this subformula α\alpha the **scope** of the quantifier

---

- in (∀xRx→∃xPx)(\forall x Rx \rightarrow \exists x Px), the scope of the quantifier ∀x\forall x is the wff RxRx
- to see this, consider how we would construct (∀xRx→∃xPx)(\forall x Rx \rightarrow \exists x Px)

Step	Wff constructed at this step	From steps/by clause
1	RxRx	/(3i)
2	PxPx	/(3i)

Step	Wff constructed at this step	From steps/by clause
1	RxRx	/(3i)
2	PxPx	/(3i)
3	∀xRx\forall x Rx	1, /(3ii ∀\forall)
4	∃xPx\exists x Px	2, /(3ii ∃\exists)

- in ∀x(Rx→∃xPx)\forall x (Rx \rightarrow \exists x Px), the scope of the quantifier ∀x\forall x is the wff (Rx→∃xPx)(Rx \rightarrow \exists x Px).

- to see this, consider the construction table for ∀x(Rx→∃xPx)\forall x (Rx \rightarrow \exists x Px)

Step	Wff constructed at this step	From steps/by clause
1	RxRx	/(3i)
2	PxPx	/(3i)
3	∃xPx\exists x Px	2, /(3ii ∃\exists)
4	(Rx→∃xPx)(Rx \rightarrow \exists x Px)	1,3, /(3ii →\rightarrow)

Step	Wff constructed at this step	From steps/by clause
1	RxRx	/(3i)
2	PxPx	/(3i)
3	∃xPx\exists x Px	2, /(3ii ∃\exists)
4	(Rx→∃xPx)(Rx \rightarrow \exists x Px)	1,3, /(3ii →\rightarrow)
5	∀x(Rx→∃xPx)\forall x (Rx \rightarrow \exists x Px)	4, /(3ii ∀\forall)

free and bound variables

free and bound variables

- an occurrence of a variable in a wff is **bound** if it is in the scope of a quantifier that contains that variable.
- an occurrence of a variable that is *not* bound in a wff is **free**.

- consider:
  - (Fx→∃xGx)(Fx \rightarrow \exists x Gx)
- the first occurrence of xx is free.
- the second—the one in the quantifier—is bound.
- the third is bound.

- if a variable falls within the scope of multiple quantifiers containing that variable, it is bound by the one added first (in the construction of the wff).
- consider:
  - ∀x(Fx→∃xGx)\forall x(Fx \rightarrow \exists x Gx)
- the second occurrence of xx is bound by ∀x\forall x.
- the fourth occurrence of xx is bound by ∃x\exists x.

- an occurrence of a quantifier is **vacuous** if the variable in the quantifier does not occur free within the scope of the quantifier.
- consider:
  - ∃xPy\exists x P y
- this occurrence of ∃x\exists x is vacuous.

- we distinguish between the quantifier symbols ∀\forall and ∃\exists and quantifiers ∀x\forall x and ∃x\exists x.
- a quantifier consists of a quantifier symbol and a variable.
- the variable which is a constituent of a quantifier is not free. (is it bound?)
- this variable is sometimes called the **operator variable**.

open and closed wffs

open and closed wffs

- corresponding to the notion of free and bound variables is the notion of open and closed wffs
- a wff with no free occurrences of variables is a **closed** wff.
    - e.g. $\forall x(Fx \rightarrow \exists x Gx)$
- a wff with one or more free occurrences of variables is an **open** wff.
    - e.g. $(F\underline{x} \rightarrow \exists x Gx)$

- open and closed wffs are equally well-formed.
- but open wffs do not express propositions—they cannot themselves be true or false.
- consider: Rx
    - this doesn’t express a proposition
    - $\exists x R x$ does, however

wrapping up

this lecture

- the syntax of MPL
- formal specification of the vocabulary and syntax of MPL
- how formulas of MPL are constructed
- the notions of logical operators, the scope of a quantifier, free and bound variables, open and closed formulas

next lecture

- lecture 15, the semantics of MPL, part 1