Formale Grundlagen (Logik)
Modul 04-006-1001

Predicate Logic II

Leipzig University

January 25^{th}, 2024

Fabian Heck
(Slides by Imke Driemel & Sandhya Sundaresan,
based on Partee, ter Meulen und Wall 1990
"Mathematical Methods in Linguistics")

Recap: Predicate logic

- in predicate logic, the atoms come in two parts: **predicates** and **terms**
- a term is either a **constant** or a **variable**
  - a constant refers to a fixed entity, e.g. an individual like John
  - a variable does not refer to a fixed entity
- **predicates** express properties of terms, they can be understood as functions

(1) Max is small. $\underbrace{SMALL}_{predicate}(\underbrace{max}_{constant})$

- we represent predicates with capital letters: $S, P, Q, R, \ldots$ and terms with lower-case letters, where variables are often $x, y, z, \ldots$ and constants are either written out ($max, peter, \ldots$) or abbreviated ($m, p, \ldots$)

(2) Max is small $= S(max) = S(m)$

Recap: Predicate logic

- when predicates combine with constants, they form statements
- predicates have different arities, also called valency
- a predicate may be ...
  - *one-place*: Susan is human $= H(susan)$
  - *two-place*: John loves Peter $= L(john, peter)$
  - *three-place*: Lee introduces Jane to Mo. $= I(lee, mo, jane)$
- ... or more, depending on the number of terms it combines with to form a statement
- like in statement logic, we can form complex expressions by using connectives

(3) a. Jumbo ist größer als Max oder Max ist größer als Jumbo.
$= G(jumbo, max) \vee G(max, jumbo)$

b. Hans und Peter sind Studenten. $= S(h) \wedge S(p)$

c. Felix ist ein gelber Papagei. $= G(f) \wedge P(f)$

d. Wenn Max Hans sieht, dann lacht Hans immer. $= S(m, h) \rightarrow L(h)$

Recap: Quantifiers

- when a predicate combines with a variable, the result is not a statement but an an open statement
- we can convert an open statement into a statement by adding something extra to the beginning of the expression: a logical primitive called a **quantifier**
  1. the **universal quantifier** = $\forall \approx$ *all, each*, and *every*
  2. the **existential quantifier** = $\exists \approx$ *some, at least one, . . .*
- *SLEEP*(x) can be converted into a statement by adding a quantifier

(4) a. $(\forall x)S(x) =$ Everyone sleeps.

b. $(\exists x)S(x) =$ Someone sleeps.

- statements that involve complex quantifiers (i.e. quantifiers that come with a explicit **restriction**) make use of connectives

(5) a. Every woman ran.
$= (\forall x)(W(x) \rightarrow R(x)) =$ For every x, if x is female, then x ran.

b. Some woman ran.
$= (\exists x)(W(x) \wedge R(x)) =$ There is at least one x, such that x is female and x ran.

Exercise: Predicate logic

- Translate the following statements into prediacte logic!

(6) a. Einige Lehrer sind freundlich.
 b. Alle Lehrer sind unfreundlich.
 c. Kein Lehrer is freundlich.
 d. Alle fahren nach Rom.
 e. Nicht jeder fährt nach Rom.
 f. Jemand fährt nicht nach Rom.
 g. Kein Kind fährt nicht nach Rom.

Exercise: Predicate logic

- Translate the following statements into prediacte logic!

(6) a. Einige Lehrer sind freundlich. $(\exists x)(L(x) \wedge F(x))$

b. Alle Lehrer sind unfreundlich. $(\forall x)(L(x) \rightarrow \neg F(x))$

c. Kein Lehrer is freundlich. $\neg(\exists x)(L(x) \wedge F(x))$

d. Alle fahren nach Rom. $(\forall x)F(x, rom)$

e. Nicht jeder fährt nach Rom. $\neg(\forall x)F(x, rom)$

f. Jemand fährt nicht nach Rom. $(\exists x)\neg F(x, rom)$

g. Kein Kind fährt nicht nach Rom. $\neg(\exists x)(K(x) \wedge \neg F(x, rom))$

The vocabulary of predicate logic

- in statement logic, we only used statements as logical expressions and connectives
- predicate logic has more components:
  1. individual constants: $\{dan, norbert, m, j, \ldots\} \rightarrow$ *terms*
  2. individual variables: $\{x, y, z, \ldots\} \rightarrow$ *terms*
  3. predicate constants: $\{C, D, L, \ldots\}$
  4. connectives: $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$
  5. auxiliary symbols: (,), [,]
  6. quantifiers: $\forall$ universal, $\exists$ existential
- these various pieces are combined to make up expressions (atomic or complex) and (open) statements
- recall that within formal languages we separate between form and content
  - **syntax**:
    properties of expressions of the system itself, such as its primitives, axioms, rules of inference
  - **semantics**:
    relations between the system and its models or interpretations

The syntax of predicate logic

- the syntactic rules below generate the set of well-formed formulas by combining the vocabulary pieces appropriately (statements are a proper subset of the list below)
- the first rule creates atomic formulas (formulas without connectives or quantifiers)
- the remaining rules create complex (possibly open) statements

(7) *Syntax of predicate logic*

a. If δ is an n-place predicate and $t_1, \ldots, t_n$ are terms, then $\delta(t_1, \ldots, t_n)$ is a well-formed formula.

b. If ϕ is a formula, then $(\neg\phi)$ is a well-formed formula.

c. If ϕ and ψ are formulas, then $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, and $(\phi \leftrightarrow \psi)$ are well-formed formulas.

d. If ϕ is a formula and x is an individual variable, then $(\forall x)\phi$ and $(\exists x)\phi$ are well-formed formulas.

e. Nothing else is a formula.

- note that the fourth rule allows for quantifiers being prefixed to expressions that do not contain a variable or that do not contain the variable the quantifier refers to, e.g. $(\forall x)P(y)$ and $(\forall x)L(mary, john)$ are wff (well-formed formulas): vacuous quantification

Exercise

- Determine the well-formedness of the following formulas

(8) a. $C(john)$

b. $(\forall x)D(n, d)$

c. $C(j \wedge d)$

d. $(L(n, joe) \vee C(x))$

e. $((C(d) \wedge L(n, j)) \rightarrow H(n))$

(9) a. $D(C(j), m)$

b. $D(n, d)$

c. $(C(j) \wedge H(\neg d))$

d. $(L(n, j) \vee C(j))$

e. $(\forall y)((C(d) \wedge D(y)) \rightarrow H(n))$

Exercise

- Determine the well-formedness of the following formulas

(8) a. $C(john)$ **wff**

b. $(\forall x)D(n, d)$ **wff**

c. $C(j \wedge d)$ **not wff**

d. $(L(n, joe) \vee C(x))$ **wff**

e. $((C(d) \wedge L(n, j)) \rightarrow H(n))$ **wff**

(9) a. $D(C(j), m)$ **not wff**

b. $D(n, d)$ **wff**

c. $(C(j) \wedge H(\neg d))$ **not wff**

d. $(L(n, j) \vee C(j))$ **wff**

e. $(\forall y)((C(d) \wedge D(y)) \rightarrow H(n))$ **wff**

The semantics of predicate logic

- as with statement logic, a statement in predicate logic can have one of two binary values: True or False (relative to a model M)
- unlike in statement logic, truth values are not the only possible semantic values
- this is because predicate logic is more complex than statement logic: it doesn't just consist of statements, it also consists of terms and predicates, constants for example have entities as values
- except for the first rule, the semantic rules are identical to those of statement logic

(10) *Semantics of predicate logic*

a. If δ is an n-place predicate and $t_1, \ldots, t_n$ are terms, then $[\![\delta(t_1, \ldots, t_n)]\!]^M = 1$ iff $\langle [\![t_1]\!]^M, \ldots, [\![t_n]\!]^M \rangle \in [\![\delta]\!]^M$.

b. If ϕ is a formula, then $[\![(\neg\phi)]\!]^M = 1$ iff $[\![\phi]\!]^M = 0$.

c. If ϕ and ψ are formulas, then $[\![(\phi \wedge \psi)]\!]^M = 1$ iff both $[\![\phi]\!]^M = 1$ and $[\![\psi]\!]^M = 1$.

d. If ϕ and ψ are formulas, then $[\![(\phi \vee \psi)]\!]^M = 1$ iff at least one of $[\![\phi]\!]^M, [\![\psi]\!]^M = 1$.

e. If ϕ and ψ are formulas, then $[\![(\phi \rightarrow \psi)]\!]^M = 1$ iff either $[\![\phi]\!]^M = 0$ or $[\![\psi]\!]^M = 1$.

f. If ϕ and ψ are formulas, then $[\![(\phi \leftrightarrow \psi)]\!]^M = 1$ iff $[\![\phi]\!]^M = [\![\psi]\!]^M$.

The semantics of predicate logic

- the following example constitutes a statement with a truth value

(11) $H(s)$ = Susan is happy

- s denotes an individual, chosen from a domain of discourse D
- D is the set whose members we agree on in advance
- H, a one-place predicate, denotes a set: specifically, the set of all happy people
- the statement $H(s)$ is True iff the individual corresponding to s happens to be member of H
- otherwise, $H(s)$ is False
- to make this more clear, let us consider two scenarios (models M and M'):

(12)
a. $[\![s]\!]^M = \text{Susan}; [\![s]\!]^{M'} = \text{Susan}$
b. $[\![H]\!]^M = \{\text{Bill, Mary, Jack}\}$
c. $[\![H]\!]^{M'} = \{\text{Bill, Susan, Jack}\}$

(13)
a. $[\![H(s)]\!]^M = \text{False}$
b. $[\![H(s)]\!]^{M'} = \text{True}$

The semantics of predicate logic

- set theory is also useful when we talk about two-place predicates

(14) $L(m, j) =$ Mary loves Jane

- m and j denote individuals, chosen from a domain of discourse D
- L also denotes a set, but this time a set of ordered pairs of individuals chosen from D, where the first individual of the ordered pair loves the second
- L is a subset of the Cartesian Product $D \times D$
- the statement $L(m, j)$ is True iff the pair $\langle m, j \rangle$ is a member of L
- otherwise, $L(m, j)$ is False
- to make this more clear, let us again consider two scenarios:

(15)
a. $[\![m]\!]^{M,M'} =$ Mary
b. $[\![j]\!]^{M,M'} =$ Jane
c. $[\![L]\!]^{M} = \{\langle m, j \rangle, \langle b, a \rangle, \langle m, k \rangle\}$
d. $[\![L]\!]^{M'} = \{\langle b, a \rangle, \langle m, k \rangle\}$

(16)
a. $[\![L(m, j)]\!]^{M'} =$ False
b. $[\![L(m, j)]\!]^{M} =$ True

The semantics of predicate logic

- so we formulate semantic rules of predicate logic with notions of set theory

(17) *Semantics of predicate logic*

a. If δ is a one-place predicate and α is a term, then $[\![\delta(\alpha)]\!]^M = 1$ iff $[\![\alpha]\!]^M \in [\![\delta]\!]^M$.

b. If γ is a two-place predicate and α and β are terms, then $[\![\gamma(\alpha, \beta)]\!]^M = 1$ iff $\langle [\![\alpha]\!]^M, [\![\beta]\!]^M \rangle \in [\![\gamma]\!]^M$.

c. ...

- as with statement logic, we relativize truth values to a model *M* (which shows up as a superscript sometimes if we want to make it explicit)
- also, similar to statement logic, there are certain statements whose truth value is fixed, regardless of the model

(18) a. $(P(s) \vee \neg P(s))$ is always True (logical tautology)

b. $(P(s) \wedge \neg P(s))$ is always False (logical contradiction)

- in contrast to statement logic, we now also consider values which are not truth values

The semantics of predicate logic

- we are now going to be more precise about the nature of logical models
- truth-values are logically contingent on the membership of the domain of discourse and the choice of semantic values (i.e. the denotations) of constants and predicates involved
- a model M is a pair $\langle D, I \rangle$, where D is the domain, a set of individuals, and I is an interpretation function: as assignment of semantic values to every basic expression in the language
- models are distinguished both by the objects in their domains and by the values assigned to the expressions of the language by I – by the particular way that the words of the language are "linked" to the things in the world

(19) $M = \langle D, I \rangle$, where:

a. $D = \{Dee, Nat, Jean, Mo\}$

b. I determines the following mapping between terms and predicate terms and objects in D:

term	value	predicate	value
d	Dee	H	$\{Nat, Mo\}$
n	Nat	C	$\{Nat, Mo, Dee\}$
j	Jean	R	$\{\langle Mo, Dee \rangle, \langle Nat, Dee \rangle\}$
m	Mo	L	$\{\langle Nat, Jean \rangle, \langle Dee, Jean \rangle, \langle Mo, Jean \rangle\}$

Exercise

Determine the truth values of the (well-formed) formulas!

(20) a. $[\![C(j)]\!]^M$

b. $[\![R(n, d)]\!]^M$

c. $[\![C(j \wedge d)]\!]^M$

d. $[\![(L(n, j) \vee C(j))]\!]^M$

e. $[\![(C(d) \wedge L(n, j)) \rightarrow H(n))]\!]^M$

term	value	pred	value
d	Dee	H	$\{Nat, Mo\}$
n	Nat	C	$\{Nat, Mo, Dee\}$
j	Jean	R	$\{\langle Mo, Dee\rangle, \langle Nat, Dee\rangle\}$
m	Mo	L	$\{\langle Nat, Jean\rangle, \langle Dee, Jean\rangle, \langle Mo, Jean\rangle\}$

Table: Model M

Exercise

Determine the truth values of the (well-formed) formulas!

(20)
a. $[\![C(j)]\!]^M$ **false**

b. $[\![R(n,d)]\!]^M$ **true**

c. $[\![C(j \wedge d)]\!]^M$ **not well-formed**

d. $[\![(L(n,j) \vee C(j))]\!]^M$ **true**

e. $[\![(C(d) \wedge L(n,j)) \rightarrow H(n))]\!]^M$ **true**

term	value	pred	value
d	Dee	H	$\{Nat, Mo\}$
n	Nat	C	$\{Nat, Mo, Dee\}$
j	Jean	R	$\{\langle Mo, Dee\rangle, \langle Nat, Dee\rangle\}$
m	Mo	L	$\{\langle Nat, Jean\rangle, \langle Dee, Jean\rangle, \langle Mo, Jean\rangle\}$

Table: Model M