Monday, August 30, 2021

SD 206

Logic and Knowledge representation

Predicate logic

Jean-Louis Dessalles

Dep. InfRes

"All human beings are mortal"

$<h1\_human, h2\_human, h3\_human, h4\_human,$
$h5\_human, h6\_human, h7\_human, h8\_human, \ldots>$

$$(\forall h)\ (human(h) \supset mortal(h))$$

Quantification

$(\forall x)\ (\exists y)$


Gottlob Frege

- « 4 is preceded by 2 »
  « 2 precedes 4 »

- "every instant is preceded by an instant"
  "an instant precedes every instant"

- "no human being is immortal"

"If there is an interpreter, then any two individuals communicate"

$$(\forall x)\ (\forall y)\ (\forall l_1)\ (\forall l_2)\ ((p(x, l_1) \wedge (p(y, l_2) \wedge (\exists z)\ (p(z, l_1) \wedge p(z, l_2)))) \supset c(x, y))$$

$$(\exists x)\ (\exists y)\ (\exists l_1)\ (\exists l_2)\ ((p(x, l_1) \wedge (p(y, l_2) \wedge (\exists z)\ (p(z, l_1) \wedge p(z, l_2)))) \wedge \neg c(x, y))$$

"There are two individuals who do not communicate despite the existence of an interpreer"

Link with description logics

$\exists r.C$ means $(\exists y)\ (r(x,y) \wedge C(y))$

$\forall r.C$ means $(\forall y)\ (r(x,y) \supset C(y))$

$\text{Mother} = \text{Woman} \sqcap \exists \text{hasChild.Person}$

$\text{Mother}(x) \equiv \text{Woman}(x) \wedge (\exists y)\ (\text{hasChild}(x,y) \wedge \text{Person}(y))$

First_order language

The syntax of predicate logic is defined by the following recursive constructs:

- Terms are
  - variables x_1, x_2, ...
  - constants c_1, c_2...
  - functors $f(t_1,..., ...x_n)$, where n is the arity of the functor and t_i are terms.

  A closed term is a term without variables. The sets $\{xi\}$, $\{ci\}$, $\{f\}$, $\{p\}$ are given once, and form the language signature.
- An atomic formula has the form $p(t_1, ...t_n)$, where p is a predicate of arity n and t_i are terms.
- A formula is:
  - An atomic formula
  - $\neg F$, where F is a formula
  - $(F\ o\ G)$, where F and G are formulas and o a binary connective of propositional logic $(\forall x)\ F$ or $(\exists x)\ F$, where x is a variable and F is a formula.

Syntax

A term is *closed* iff it contains no variable.

A variable v is *free* in an atomic formula F iff v has an occurrence in F.
If ◊ is a quantifier, a variable v is free in (◊v')F iff v is free in F
and v is different from v'.
A variable v is free in $\neg F$ iff v is free in F.
If o is a 2-place connective, a variable v is free in F_1 o F_2 iff
v is free in F_1 or in F_2.

A variable v is *bound* in a formula F iff it has no free occurrence.
A formula is *closed* iff it contains no free variable.

Semantics: model

- We are given a non-empty set D called **domain**.
- An **interpretation** I associates
  - each constant c of the language with an element c^I of D,
  - each functor f of arity n to a function f^I from D^n to D;
  - each predicate P of arity n to a n-ary relation P^I in D.
- An **assignment** A instantiates each variable v by giving it a value v^A taken from D.
- Terms are interpreted recursively from the interpretation of their elements:

  for each term t, $t^{I,A}$ is defined as:

  c^I for a constant c,

  v^A for a variable v,

  $f^I(t_1^{I,A}, t_2^{I,A}, \dots t_n^{I,A})$ for a functional term $f(t_1, , \dots t_n)$.
- A **model** $M(D, I)$ is defined by the domain D and the interpretation I.

Attitudes (truth conditions)

$P(t_1, , \ldots t_n)^{I,A} = \boldsymbol{T}$ if and only if $(t_1^{I,A}, t_2^{I,A}, \ldots\ t_n^{I,A}) \in P^I$

$\top^{I,A} = \boldsymbol{T}$; $\bot^{I,A} = \boldsymbol{F}$

$(\neg X^{I,A}) = \neg; X^{I,A}$

$(XoY)^{I,A} = X^{I,A} \bullet Y^{I,A}$ for coupled operators o and • (cf. propositional logic)

$((\forall x)\ F)^{I,A} = \boldsymbol{T}$ if and only if $F^{I,B} = \boldsymbol{T}$ for all assignations B equal to A save for x.

$((\exists x)\ F)^{I,A} = \boldsymbol{T}$ if and only if $F^{I,B} = \boldsymbol{T}$ for (at least) one assignation B

equal to A save for x.

Satisfiability, Truth, Validity

- A formula F is *true* in $M(D,I)$ if $F^{I,A} = \boldsymbol{T}$ for all assignments A.
- A formula F is *valid* if F is true in any model of the language.
- A set S of formulas is *satisfiable* in $M(D, I)$
  if there is (at least) an assignment A
  such that $F^{I,A} = \boldsymbol{T}$ for all F belonging to S.
- S is satisfiable if S it is satisfiable in a model.

Note that a formula F is valid iff $\{\neg F\}$ is not satisfiable.

$$((\exists x) (A \wedge \neg P(x)) \supset \neg(\forall x) P(x))$$

$$\neg ((\exists x) (A \wedge \neg P(x)) \supset \neg(\forall x) P(x))$$

$$(\exists x) (A \wedge \neg P(x))$$

$$(\forall x) P(x)$$

[ ]

$$((\exists x) (A \wedge \neg P(x)) \supset \neg(\forall x) P(x))$$

$$\neg ((\exists x) (A \wedge \neg P(x)) \supset \neg(\forall x) P(x))$$

$$(\exists x) (A \wedge \neg P(x))$$

Skolemization

$$(\forall x) P(x)$$

$$(\mathrm{A} \wedge \neg P(c))$$

A

$$\neg P(c)$$

unification

$$P(x)$$

[ ]

Proof by resolution of:

$$((\forall x)\,(P(x) \vee Q(x)) \supset ((\exists x)\,P(x) \vee (\forall x)\,Q(x)))$$

1. $[\, \neg((\forall x)\,(P(x) \vee Q(x)) \supset ((\exists x)\,P(x) \vee (\forall x)\,Q(x)))\,]$
2. $[\, (\forall x)\,(P(x) \vee Q(x))\,]$
3. $[\, \neg((\exists y)\,P(y) \vee (\forall z)\,Q(z))\,]$
4. $[\, \neg(\exists y)\,P(y)\,]$
5. $[\, \neg(\forall z)\,Q(z)\,]$
6. $[\, \neg\, Q(c)\,]$ Skolemization
7. $[\, \neg P(y)\,]$ from 4.
8. $[\, P(x), Q(x)\,]$ from 2.
9. $[\, Q(y)\,]$ resolution 7. & 8.
10. $[\ ]$ resolution with unification 6. & 9.

Prenex form

In the following formulas, the variable x is assumed to have no occurrence in the part of the formula that is outside the scope of the quantifier; this can be achieved by renaming the variables. The following formulas are valid:

$$\neg (\exists v)\, A \equiv (\forall v) \neg A$$
$$\neg (\forall v)\, A \equiv (\exists v) \neg A$$
$$((\forall v)\, A \wedge B) \equiv (\forall v)\, (A \wedge B)$$
$$(A \wedge (\forall v)\, B) \equiv (\forall v)\, (A \wedge B)$$
$$((\exists v)\, A \wedge B) \equiv (\exists v)\, (A \wedge B)$$
$$(A \wedge (\exists v)\, B) \equiv (\exists v)\, (A \wedge B)$$
$$((\forall v)\, A \supset B) \equiv (\exists v)\, (A \supset B)$$
$$(A \supset (\forall v)\, B) \equiv (\forall v)\, (A \supset B)$$
$$((\exists v)\, A \supset B) \equiv (\forall v)\, (A \supset B)$$
$$(A \supset (\exists v)\, B) \equiv (\exists v)\, (A \supset B)$$

Put in prenex form:

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

Skolemization

A formula in prenex form $(Q_1x_1)\ (Q_2x_2)\ \ldots(\exists\, x_k)\ldots(Q_nx_n)\ F$
is transformed into $(Q_1x_1)\ (Q_2x_2)\ \ldots\ \ldots(Q_nx_n)\ F(x_k/\, f(x_1,\, x_2\, \ldots x_{k\text{-}1}))$
where *f* is a new functor that does not belong to the language.
f is called a *Skolem* function.

The two formulas have the same truth conditions.

The use of Skolem functions ensures that any formula *F* with no free variables (i.e. without not quantified variables) can be mapped to a formula *G* in prenex form with only universal quantifiers such that *F* is satisfiable if and only if *G* is satisfiable.

Put this formula in prenex skolemized form:
$(\forall x)\ (\forall y)\ (p(x) \wedge p(y)) \supset (\forall x)\ (\forall y)\ (p(x) \vee p(y))$

Proof method

To prove the validity of formula F:

1. **Transform $\neg F$ in prenex form, renaming variables if necessary**
2. **Remove existential quantifiers through skolemization**
3. **Remove universal quantifiers**
   (this amounts to transforming the associated quantified variables into free variables)
4. **Transform the formula into conjunctive normal form (CNF)**
5. **Use the resolution algorithm with unification.**
6. **If the empty clause is obtained through this process starting from $\neg F$, then {¬F} is not satisfiable, and F is valid.**

Check the validity of: $(\forall x)\ (\exists y)\ (\forall z)\ (\exists w)\ (\mathrm{R}(x,y) \vee \neg R(w,z))$

Soundness, completeness, decidability

- There are correct et complete proof methods for predicate logic (Gödel 1930)
  - Axiomatic systems
  - Tableaux
  - Resolution
  - Natural deduction
  - ...
- There are only semi-decidable algorithms
  (the set of theorems is recursively enumerable, but is not recursive (in general))