PREDICATE CALCULUS

Predicate language is more expressive than propositional language.
It is used to express object properties and relations between objects.

A *relation* $\mathcal{R}$ on $D_1, D_2, \ldots, D_n$ is a subset of the cartesian product $D_1 \times D_2 \times \cdots \times D_n$; number n is the *arity* of the relation.

$$\mathcal{LESS}(x, y) = \{(x, y) \in (\mathbf{N} \times \mathbf{N}) \mid x < y\}$$

$$\begin{aligned} = \{&(0,1),(0,2),(0,3),\ldots,\\ &(1,2),(1,3),(1,4),\ldots,\\ &(2,3),(2,4),(2,5),\ldots,\\ &\vdots \qquad\qquad\qquad\qquad\qquad \} \end{aligned}$$

$$\mathcal{SQUARE}(x, y) = \{(x, y) \in (\mathbf{N} \times \mathbf{N}) \mid y = x^2\} = \{(0,0), \ldots, (3,9), \ldots\}$$

$$\mathcal{PR}(x) = \{x \in \mathbf{N} \mid x \text{ is a prime number}\} = \{2, 3, 5, 7, 11, 13, \ldots\}$$

$\mathcal{R}$ is an n-*ary relation* on domain D if $\mathcal{R}$ is a subset of D^n.

The *predicate* R associated with the n-ary relation $\mathcal{R}$ is defined by

$$R(d_1, \ldots, d_n) = T \quad \text{iff} \quad (d_1, \ldots, d_n) \in \mathcal{R}.$$

Examples. :
$LESS(0,1) = T \quad LESS(8,4) = F \quad LESS(3,6) = T \quad \cdots$
$SQUARE(0,0) = T \quad SQUARE(0,2) = F \quad SQUARE(2,4) = T$
$SQUARE(2,7) = F \quad \cdots$
$PR(3) = T \quad PR(8) = F \quad \cdots$

Proposition : property, true or false

Predicate : property, true for some elements of a domain.

Predicate logic

- quantification
- interpretation : domain D
  + predicate : relation on the domain, subset of D^n
  + function : $D^n \mapsto D$

Valid formulas can be enumerated but there is no decision procedure.

Complete methods :

- Semantic tableaux (Hintikka)
- Axiomatic systems (Hilbert)
- Canonical models (Herbrand)
- Resolution (Robinson)

PREDICATE CALCULUS (without functions) : THE SYNTAX

— $\mathcal{P} = \{p, q, r, \ldots\}$: a set of *predicate symbols* (with arity).

NB : propositions are 0-ary predicates.

— $\mathcal{A} = \{a, a_1, a_2, \ldots, b, c, \ldots\}$: a set of *(individual) constants*.

— $\mathcal{X} = \{x, x_1, x_2, x', \ldots, y, z, , \ldots\}$: a set of *(individual) variables*.

Atomic formulas involve terms.

We define terms, then atoms, then (general) formulas.

Terms and formulas

A *term* is a constant $a \in \mathcal{A}$ or a variable $x \in \mathcal{X}$.

An *atomic formula (atom)* is an expression $p(t_1, \ldots, t_n)$, where $p \in \mathcal{P}$ is an n-ary predicate symbol and $t_1, \ldots, t_n$ are *terms*.

Formulas are recursively defined :

— An *atomic formula* is a formula.
— *true*, *false* are formulas.
— If A is a formula, then $\neg A$ is a formula.
— If A_1 and A_2 are formulas, then $(A_1 \vee A_2)$, $(A_1 \wedge A_2)$, $(A_1 \Rightarrow A_2)$, $(A_1 \equiv A_2)$ . . . are formulas.
— If A is a formula and x is a variable, then $\forall x A$ and $\exists x A$ are formulas.

Parentheses, precedence

Some precedence rules are used :

Precedence for connectives :

Quantifications and negation are more binding than binary connectives.

Example :

$$\Big(\forall x\Big((\neg(\exists y\, p(x,y))) \vee (\neg(\exists y\, p(y,x)))\Big)\Big)$$

can be written

$$\forall x\Big(\neg\exists\, yp(x,y) \vee \neg\exists\, yp(y,x)\Big)$$

Quantification, scope, bound variable, free variable

— In $\forall x A$ and $\exists x A$,
the scope of x is A.

Scope in logic is similar to scope in programming languages.

— The occurrence of variable x in quantification $\forall x$ or $\exists x$ is *quantified*.

— Any occurrence of x in the scope of a quantification is *bound*.

— A variable occurrence is *free* when it is neither quantified nor bound.

— The scopes of two distinct variables x and y are disjoint or one is included in the other.

Scope in programming languages

```
program Principal;
var x : integer;

procedure p;
var x : integer;
begin
    x := 1;
    writeln(x + x)
end;

procedure q;
var y : integer;
begin
    y := 1;
    writeln(x + y)
end;

begin
    x := 5 ;
    p;
    q
end.
```

The scopes of two distinct variables are disjoint or one is included in the other.

Free and bound variables, examples

1. $\varphi_1 =_{def} \forall x (p(x,a) \Rightarrow \exists x\, q(x))$

   Two imbricated quantifications on x.

   Best avoid this and rewrite
   into $\forall x(p(x,a) \Rightarrow \exists y\, q(y))$ or into $\forall y(p(y,a) \Rightarrow \exists x\, q(x))$.

2. $\varphi_2 =_{def} \exists x \forall x A$

   Two imbricated quantifications on x.

   Best avoid this and rewrite into $\forall x A$.

3. $\varphi_3 =_{def} \forall x\, p(x,a) \Rightarrow \exists x\, q(x)$

   Two distinct variables share the same name.

   As the scopes are disjoint, there is no problem..

4. $\varphi_4 =_{def} \forall x\, p(x,a) \Rightarrow q(x)$

   A free variable and a bound variable share the name x.

   It is best to rewrite this into $\forall y\, p(y,a) \Rightarrow q(x)$.

   Renaming is allowed for bound variables only !

Universal and existential closure

A formula is *closed* if all variable occurrences are bound.

When a formula A contains free occurrences of $x_1, x_2, \ldots, x_n$, this can be emphasised by using the notation $A(x_1, x_2, \ldots, x_n)$.

If $x_1, x_2, \ldots, x_n$ (and only them) have free occurrences in formula A,

— $\forall x_1 \forall x_2 \cdots \forall x_n A$
  is the *universal closure* of A (sometimes denoted $\forall * A$).

— $\exists x_1 \exists x_2 \cdots \exists x_n A$
  is the *existential closure* de A (sometimes denoted $\exists * A$).

The universal closure of $p(x) \Rightarrow q(x)$
is $\forall x\,(p(x) \Rightarrow q(x))$ and not $\forall x\, p(x) \Rightarrow q(x)$.

Examples of formulas

1. $\forall x \forall y \big(p(x,y) \Rightarrow p(y,x)\big)$
   This formula is true for all interpretations associating with p a symmetric relation.

2. $\forall x \exists y\, p(x,y)$

3. $\exists y \forall x\, p(x,y)$

4. $\forall x\, p(a,x)$

5. $\exists x \exists y \big(p(x) \wedge \neg p(y)\big)$
   This formula cannot be true if the interpretation relies on a one-element domain.

6. $\forall x \big(p(x) \wedge q(x)\big) \equiv \big(\forall x\, p(x) \wedge \forall x\, q(x)\big)$
   This formula is valid.

7. $\exists x \big(p(x) \vee q(x)\big) \equiv \big(\exists x\, p(x) \vee \exists x\, q(x)\big)$
   This formula is valid.

8. $\forall x \big(p(x) \Rightarrow q(x)\big) \Rightarrow \big(\forall x\, p(x) \Rightarrow \forall x\, q(x)\big)$
   This formula is valid.

9. $\big(\forall x\, p(x) \Rightarrow \forall x\, q(x)\big) \Rightarrow \forall x \big(p(x) \Rightarrow q(x)\big)$
   This formula is not valid.

PREDICATE CALCULUS (without functions)
THE SEMANTICS

In predicate logic, valuations assign objects to terms and truthvalues to formulas.

A *valuation* or *interpretation* $\mathcal{I}$ is a triple (D, I_c, I_v) such that :

— D is a non-empty set, the *domain* ;
— I_c is a function that maps
  — an objet $I_c[a] \in D$ to each constant a,
  — an n-ary relation on D to each n-ary predicate symbol p
    (an n-ary relation on D is also a function $D^n \mapsto \{T, F\}$) ;
— I_v is a function that maps an objet $I_v[x] \in D$ to each variable x.

Interpretations : examples

Even a short formula like $\forall x\, p(a,x)$ can be interpreted in infinitely many ways ; some of them are :

— $\mathcal{I}_1 = (\mathbf{N}, \quad I_{1c}[p] = \leq, \quad I_{1c}[a] = 0)$
— $\mathcal{I}_2 = (\mathbf{N}, \quad I_{2c}[p] = \leq, \quad I_{2c}[a] = 1)$
— $\mathcal{I}_3 = (\mathbf{Z}, \quad I_{3c}[p] = \leq, \quad I_{3c}[a] = 0)$
— $\mathcal{I}_4 = (\mathcal{S}, \quad I_{4c}[p] = \sqsubseteq, \quad I_{4c}[a] = \varepsilon)$

The first three interpretations are about numbers ; the fourth one is about strings.

Interpretation rules I

An interpretation $\mathcal{I} = (D, I_c, I_v)$ assigns an element of D to every term and a truthvalue to every formula.

Interpretation : terms :

— If x is a free variable, $\mathcal{I}[x] = I_v[x]$.

— If a is a constant, $\mathcal{I}[a] = I_c[a]$.

Interpretation : formulas :

— If p is an n-ary predicate symbol and if $t_1, \ldots, t_n$ are terms, then $\mathcal{I}[p(t_1, \ldots, t_n)] = (I_c[p])(\mathcal{I}[t_1], \ldots, \mathcal{I}[t_n])$.

— $\mathcal{I}[\mathit{true}] = T$ and $\mathcal{I}[\mathit{false}] = F$.

— If A is a formula, then

$\mathcal{I}[\neg A] = T$ if $\mathcal{I}[A] = F$,

$\mathcal{I}[\neg A] = F$ if $\mathcal{I}[A] = T$.

Interpretation rules II

— If A_1 and A_2 are formulas, then $(A_1 \vee A_2)$, $(A_1 \wedge A_2)$, $(A_1 \Rightarrow A_2)$, $(A_1 \equiv A_2)$ are interpreted as in propositional logic :

$\mathcal{I}[(A_1 \wedge A_2)]$
  $= T$ if $\mathcal{I}[A_1] = T$ and $\mathcal{I}[A_2] = T$,
  $= F$ else.

$\mathcal{I}[(A_1 \vee A_2)]$
  $= T$ if $\mathcal{I}[A_1] = T$ or $\mathcal{I}[A_2] = T$,
  $= F$ else.

$\mathcal{I}[(A_1 \Rightarrow A_2)]$
  $= T$ if $\mathcal{I}[A_1] = F$ or $\mathcal{I}[A_2] = T$,
  $= F$ else.

$\mathcal{I}[(A_1 \equiv A_2)]$
  $= T$ if $\mathcal{I}[A_1] = \mathcal{I}[A_2]$,
  $= F$ if $\mathcal{I}[A_1] \neq \mathcal{I}[A_2]$.

Interpretation rules III

Notation : If $\mathcal{I} = (D_{\mathcal{I}}, I_c, I_v)$ is an interpretation, if x is a variable and d is an element of $D_{\mathcal{I}}$, then $\mathcal{I}_{x/d}$ is the interpretation $\mathcal{J} = (D_{\mathcal{J}}, J_c, J_v)$ such that

- $D_{\mathcal{J}} = D_{\mathcal{I}}$,
- $J_c = I_c$,
- $J_v[x] = d$ et $J_v[y] = I_v[y]$ for each variable y other than x.

— If A is a formula and x is a variable,
  $\mathcal{I}[\forall x A]$
  $= T$ if $\mathcal{I}_{x/d}[A] = T$ for each element $d \in D$,
  $= F$ else.

— If A is a formula and x a variable,
  $\mathcal{I}[\exists x A]$
  $= T$ if $\mathcal{I}_{x/d}[A] = T$ for at least one element $d \in D$,
  $= F$ else.

Avoid capturing variables. . .

Formal rules only make the intuition more precise, but . . .

Observe that $\mathcal{I}[\forall x\, A(x)]$ does not depend on $\mathcal{I}[x]$.

Observe that if $\mathcal{I}[\forall x\, A(x)] = T$ then $\mathcal{I}[A(t)] = T$ for each term t.

Example. If $\forall x\, \exists y\, p(x, y)$ is true, then so are its *instances*, like $\exists y\, p(a, y)$, $\exists y\, p(x, y)$ and $\exists y\, p(z, y)$

However, the "pseudo-instance" $\exists y\, p(y, y)$ might be false !

If we want to instantiate x into y, we must first rewrite $\forall x\, \exists y\, p(x, y)$ into, say, $\forall x\, \exists z\, p(x, z)$; instantiation into $\exists z\, p(y, z)$ is now possible.

Conclusion. Formula $\exists y\, p(y, y)$ is not really an instance of $\forall x\, \exists y\, p(x, y)$; y cannot be substituded to x in $\exists y\, p(x, y)$, since the free x would be replaced by a bound y. Binding without warrant is forbidden !

Satisfaction, model

A formula A *is true for an interpretation* $\mathcal{I}$ or A *is satisfied by an interpretation* $\mathcal{I}$ or $\mathcal{I}$ *is a model of* A if $\mathcal{I}[A] = T$. This can be written as $\models_{\mathcal{I}} A$.

NB : This is sometimes written $\mathcal{I} \models A$, which can be misleading.

Examples : Formula A : $\forall x\, p(a, x)$

— $D_{\mathcal{I}_1} = \mathbf{N}$, $I_{1c}[p] = \leq$, $I_{1c}[a] = 0$: $\models_{\mathcal{I}_1} A$
— $D_{\mathcal{I}_2} = \mathbf{N}$, $I_{2c}[p] = \leq$, $I_{2c}[a] = 1$: $\not\models_{\mathcal{I}_2} A$
— $D_{\mathcal{I}_3} = \mathbf{Z}$, $I_{3c}[p] = \leq$, $I_{3c}[a] = 0$: $\not\models_{\mathcal{I}_3} A$
— $D_{\mathcal{I}_4} = \mathcal{S}$, $I_{4c}[p] = \sqsubseteq$, $I_{4c}[a] = \varepsilon$: $\models_{\mathcal{I}_4} A$

Satisfiability, validity

Just like in propositional logic !

Definitions.

If A is a formula of predicate logic,

— A is *satisfiable* or *consistent*
 if A has at least one model.

— A is *valid* (this is noted $\models A$),
 if $\mathcal{I}[A] = T$ for each interpretation $\mathcal{I}$.
 (The word *tautology* is more often used in propositional logic.)

— A is *unsatisfiable* or *inconsistent*
 if A is not satisfiable, that is,
 if $\mathcal{I}[A] = F$ for each interpretation $\mathcal{I}$.

Theorem (duality between validity and consistency).

A is valid iff $\neg A$ is inconsistent.

Examples

— $\forall x\, p(a,x)$ is consistent but not valid.
$D_{\mathcal{I}_1} = \mathbf{N}$, $I_{1c}[p] = \leq$, $I_{1c}[a] = 0$: $\models_{\mathcal{I}_1} A$
$D_{\mathcal{I}_3} = \mathbf{Z}$, $I_{3c}[p] = \leq$, $I_{3c}[a] = 0$: $\not\models_{\mathcal{I}_3} A$
— $\forall x\, p(x) \Rightarrow p(a)$ is valid.
— $\exists x\, p(x) \Rightarrow p(a)$ is consistent but not valid.

Simply consistent or *contingent* means consistent but not valid.

Comment. Propositional valid schemes are also predicate valid schemes ; for instance, from

$$\neg\neg A \equiv A,$$

$\neg\neg(p \wedge q) \equiv (p \wedge q)$ can be deduced, but also $\neg\neg\forall x\, p(x) \equiv \forall x\, p(x)$.

Examples, valid formulas

- $(\forall x A \land \forall x B) \equiv \forall x(A \land B)$
- $(\forall x A \lor \forall x B) \Rightarrow \forall x(A \lor B)$
- $\forall x(A \Rightarrow B) \Rightarrow (\forall x A \Rightarrow \forall x B)$
- $\forall x(A \equiv B) \Rightarrow (\forall x A \equiv \forall x B)$

- $\exists x(A \lor B) \equiv (\exists x A \lor \exists x B)$
- $\exists x(A \land B) \Rightarrow (\exists x A \land \exists x B)$
- $\exists x(A \Rightarrow B) \equiv (\forall x A \Rightarrow \exists x B)$

- $\forall x A \equiv \neg(\exists x \neg A)$

- $\forall x \forall y A \equiv \forall y \forall x A$
- $\exists x \exists y A \equiv \exists y \exists x A$
- $\exists x \forall y A \Rightarrow \forall y \exists x A$

Logical consequence, logical equivalence I

Just like in propositional logic !

Definitions. If U is a formula set and if A and B are formulas,

— A is a *logical consequence* of U (noted $U \models A$) if A is true in all models of U.

  Comment. Usually, U contains only closed formulas ; in this case, $U \models A$ iff $U \models \forall * A$.

— A et B are *logically equivalent* (noted $A \leftrightarrow B$) if $\mathcal{I}[A] = \mathcal{I}[B]$ for each interpretation $\mathcal{I}$.

As in propositional logic,

$$\models A \quad \text{ssi} \quad \emptyset \models A\,.$$

Logical consequence, logical equivalence II

Theorem. A formula is valid (resp. consistent) if and only if its universal (resp. existential) closure is valid (resp. consistent).

Theorem. Two formulas A and B are logically equivalent if and only if formula $A \equiv B$ is valid.

As a result, validity and consistency proving methods can be restricted to closed formulas.

Substitution theorem and replacement theorem can be adapted to predicate logic.

SEMANTIC TABLEAUX

Principle : systematic search for models.

Quantifications are *instantiated.*

Example 1.

$$\neg\Big(\forall x(p(x) \Rightarrow q(x)) \Rightarrow (\forall x\, p(x) \Rightarrow \forall x\, q(x))\Big)$$
$$\downarrow$$
$$\forall x(p(x) \Rightarrow q(x)), \neg(\forall x\, p(x) \Rightarrow \forall x\, q(x))$$
$$\downarrow$$
$$\forall x(p(x) \Rightarrow q(x)), \forall x\, p(x), \neg\forall x\, q(x)$$
$$\downarrow$$
$$\forall x(p(x) \Rightarrow q(x)), \forall x\, p(x), \neg q(a)$$
$$\downarrow$$
$$\forall x(p(x) \Rightarrow q(x)), p(a), \neg q(a)$$
$$\downarrow$$
$$p(a) \Rightarrow q(a), p(a), \neg q(a)$$
$$\swarrow \qquad\qquad \searrow$$
$$\neg p(a), p(a), \neg q(a) \qquad\qquad q(a), p(a), \neg q(a)$$
$$\times \qquad\qquad\qquad\qquad \times$$

Example 2. *This is not correct!*

$$\neg\Big(\forall x(p(x) \vee q(x)) \Rightarrow (\forall x\, p(x) \vee \forall x\, q(x))\Big)$$
$$\downarrow$$
$$\forall x(p(x) \vee q(x)), \neg(\forall x\, p(x) \vee \forall x\, q(x))$$
$$\downarrow$$
$$\forall x(p(x) \vee q(x)), \neg\forall x\, p(x), \neg\forall x\, q(x)$$
$$\downarrow$$
$$\forall x(p(x) \vee q(x)), \neg\forall x\, p(x), \neg q(a)$$
$$\downarrow$$
$$\forall x(p(x) \vee q(x)), \neg p(a), \neg q(a)$$
$$\downarrow$$
$$p(a) \vee q(a), \neg p(a), \neg q(a)$$
$$\swarrow \qquad\qquad \searrow$$
$$p(a), \neg p(a), \neg q(a) \qquad\qquad q(a), \neg p(a), \neg q(a)$$
$$\times \qquad\qquad\qquad\qquad \times$$

$\forall x(p(x) \vee q(x)) \Rightarrow (\forall x\, p(x) \vee \forall x\, q(x))$ is not valid !

Where is the mistake?

Example 2 : the correct version.

$$\neg\Big(\forall x\big(p(x) \lor q(x)\big) \Rightarrow \big(\forall x\, p(x) \lor \forall x\, q(x)\big)\Big)$$

$$\downarrow$$

$$\forall x\big(p(x) \lor q(x)\big), \neg\big(\forall x\, p(x) \lor \forall x\, q(x)\big)$$

$$\downarrow$$

$$\forall x\big(p(x) \lor q(x)\big), \neg\forall x\, p(x), \neg\forall x\, q(x)$$

$$\downarrow$$

$$\forall x\big(p(x) \lor q(x)\big), \neg\forall x\, p(x), \neg q(a)$$

$$\downarrow$$

$$\forall x\big(p(x) \lor q(x)\big), \neg p(b), \neg q(a)$$

$$\downarrow$$

$$\forall x\big(p(x) \lor q(x)\big), p(a) \lor q(a), \neg p(b), \neg q(a)$$

$$\downarrow$$

$$\forall x\big(p(x) \lor q(x)\big), p(b) \lor q(b), p(a) \lor q(a), \neg p(b), \neg q(a)$$

$$\swarrow \qquad\qquad \searrow$$

$$\begin{array}{ll} \forall x\big(p(x) \lor q(x)\big), p(b), & \forall x\big(p(x) \lor q(x)\big), q(b) \\ p(a) \lor q(a), \neg p(b), & p(a) \lor q(a), \neg p(b), \neg q(a) \\ \neg q(a) & \\ \times & \end{array}$$

$$\swarrow \qquad\qquad \searrow$$

$$\begin{array}{cc} \forall x\big(p(x) \lor q(x)\big), & \forall x\big(p(x) \lor q(x)\big), \\ q(b), p(a), & q(b), q(a), \\ \neg p(b), \neg q(a) & \neg p(b), \neg q(a) \\ \bigcirc & \times \end{array}$$

Formula $\neg\Big(\forall x\big(p(x) \lor q(x)\big) \Rightarrow \big(\forall x\, p(x) \lor \forall x\, q(x)\big)\Big)$ has a model $\mathcal{I}$ such that $\mathcal{I}[p(a)] = \mathcal{I}[q(b)] = T$ and $\mathcal{I}[p(b)] = \mathcal{I}[q(a)] = F$.

Example 3.

$$\forall x \exists y\, p(x,y) \wedge \forall x \neg p(x,x) \wedge \forall x \forall y \forall z \big(p(x,y) \wedge p(y,z) \Rightarrow p(x,z)\big)$$

$$\downarrow$$

$$\forall x \exists y\, p(x,y), \forall x \neg p(x,x) \wedge \forall x \forall y \forall z \big(p(x,y) \wedge p(y,z) \Rightarrow p(x,z)\big)$$

$$\downarrow$$

$$\forall x \exists y\, p(x,y), \forall x \neg p(x,x), \forall x \forall y \forall z \big(p(x,y) \wedge p(y,z) \Rightarrow p(x,z)\big)$$

$$\downarrow$$

$$\begin{gathered}\forall x \exists y\, p(x,y), \exists y\, p(a_1,y),\\ \forall x \neg p(x,x), \forall x \forall y \forall z \big(p(x,y) \wedge p(y,z) \Rightarrow p(x,z)\big)\end{gathered}$$

$$\downarrow$$

$$\begin{gathered}\forall x \exists y\, p(x,y), p(a_1,a_2),\\ \forall x \neg p(x,x), \forall x \forall y \forall z \big(p(x,y) \wedge p(y,z) \Rightarrow p(x,z)\big)\end{gathered}$$

$$\downarrow$$

$$\begin{gathered}\forall x \exists y\, p(x,y), p(a_1,a_2), \exists y\, p(a_2,y),\\ \forall x \neg p(x,x), \forall x \forall y \forall z \big(p(x,y) \wedge p(y,z) \Rightarrow p(x,z)\big)\end{gathered}$$

$$\downarrow$$

$$\begin{gathered}\forall x \exists y\, p(x,y), p(a_1,a_2), p(a_2,a_3),\\ \forall x \neg p(x,x), \forall x \forall y \forall z \big(p(x,y) \wedge p(y,z) \Rightarrow p(x,z)\big)\end{gathered}$$

$$\downarrow$$

$$\begin{gathered}\forall x \exists y\, p(x,y), p(a_1,a_2), p(a_2,a_3), \exists y\, p(a_3,y),\\ \forall x \neg p(x,x), \forall x \forall y \forall z \big(p(x,y) \wedge p(y,z) \Rightarrow p(x,z)\big)\end{gathered}$$

$$\downarrow$$

$$\vdots$$

Infinite semantic tableau !
The formula has only infinite models.

Example 4.

$$\forall x \exists y\, p(x,y) \land \forall x \neg p(x,x) \land \forall x \forall y \forall z \big(p(x,y) \land p(y,z) \Rightarrow p(x,z)\big) \land \forall x \big(q(x) \land \neg q(x)\big)$$
$$\downarrow$$
$$\forall x \exists y\, p(x,y), \forall x \neg p(x,x) \land \forall x \forall y \forall z \big(p(x,y) \land p(y,z) \Rightarrow p(x,z)\big) \land \forall x \big(q(x) \land \neg q(x)\big)$$
$$\downarrow$$
$$\begin{gathered}\forall x \exists y\, p(x,y),\\ \forall x \neg p(x,x), \forall x \forall y \forall z \big(p(x,y) \land p(y,z) \Rightarrow p(x,z)\big) \land \forall x \big(q(x) \land \neg q(x)\big)\end{gathered}$$
$$\downarrow$$
$$\begin{gathered}\forall x \exists y\, p(x,y),\\ \forall x \neg p(x,x), \forall x \forall y \forall z \big(p(x,y) \land p(y,z) \Rightarrow p(x,z)\big),\\ \forall x \big(q(x) \land \neg q(x)\big)\end{gathered}$$
$$\downarrow$$
$$\begin{gathered}\forall x \exists y\, p(x,y),\\ \forall x \neg p(x,x), \forall x \forall y \forall z \big(p(x,y) \land p(y,z) \Rightarrow p(x,z)\big),\\ \forall x \big(q(x) \land \neg q(x)\big), q(a) \land \neg q(a)\end{gathered}$$
$$\downarrow$$
$$\begin{gathered}\forall x \exists y\, p(x,y),\\ \forall x \neg p(x,x) \land \forall x \forall y \forall z \big(p(x,y) \land p(y,z) \Rightarrow p(x,z)\big),\\ \forall x \big(q(x) \land \neg q(x)\big), q(a), \neg q(a)\\ \times\end{gathered}$$

Should we endlessly instantiate $\forall x \exists y\, p(x,y)$,
the branch would not close !

Example 5.

$$\neg\big(\forall x \exists y\,(p(x) \Rightarrow q(y)) \Rightarrow \exists y \forall x\,(p(x) \Rightarrow q(y))\big)$$

$$\downarrow$$

$$\forall x \exists y\,(p(x) \Rightarrow q(y))\,,\ \neg\exists y \forall x\,(p(x) \Rightarrow q(y))$$

$$\downarrow$$

$$\forall x \exists y\,(p(x) \Rightarrow q(y))\,,\ \neg\forall x\,(p(x) \Rightarrow q(c))\,,\ \neg\exists y \forall x\,(p(x) \Rightarrow q(y))$$

$$\downarrow$$

$$\forall x \exists y\,(p(x) \Rightarrow q(y))\,,\ \neg(p(a) \Rightarrow q(c))\,,\ \neg\exists y \forall x\,(p(x) \Rightarrow q(y))$$

$$\downarrow$$

$$\forall x \exists y\,(p(x) \Rightarrow q(y))\,,\ p(a)\,,\ \neg q(c)\,,\ \neg\exists y \forall x\,(p(x) \Rightarrow q(y))$$

$$\downarrow$$

$$\forall x \exists y\,(p(x) \Rightarrow q(y))\,,\ \neg\exists y \forall x\,(p(x) \Rightarrow q(y))\,,\ \exists y\,(p(a) \Rightarrow q(y))\,,\ p(a)\,,\ \neg q(c)$$

$$\downarrow$$

$$\forall x \exists y\,(p(x) \Rightarrow q(y))\,,\ \neg\exists y \forall x\,(p(x) \Rightarrow q(y))\,,\ p(a) \Rightarrow q(b)\,,\ p(a)\,,\ \neg q(c)$$

$$\downarrow$$

$$\forall x \exists y\,(p(x) \Rightarrow q(y))\,,\ \neg\forall x\,(p(x) \Rightarrow q(b))\,,\ \neg\exists y \forall x\,(p(x) \Rightarrow q(y))\,,\ p(a) \Rightarrow q(b)\,,\ p(a)\,,\ \neg q(c)$$

$$\downarrow$$

$$\forall x \exists y\,(p(x) \Rightarrow q(y))\,,\ \neg\exists y \forall x\,(p(x) \Rightarrow q(y))\,,\ \neg(p(d) \Rightarrow q(b))\,,\ p(a) \Rightarrow q(b)\,,\ p(a)\,,\ \neg q(c)$$

$$\downarrow$$

$$\begin{array}{c} \forall x \exists y\,(p(x) \Rightarrow q(y))\,,\ \neg\exists y \forall x\,(p(x) \Rightarrow q(y)) \\ p(d)\,,\ \neg q(b)\,,\ p(a) \Rightarrow q(b)\,,\ p(a)\,,\ \neg q(c) \end{array}$$

$$\swarrow \qquad\qquad \searrow$$

$$\begin{array}{cc} \forall x \exists y\,(\ldots)\,,\ \neg\exists y \forall x\,(\ldots) & \forall x \exists y\,(\ldots)\,,\ \neg\exists y \forall x\,(\ldots) \\ p(d), \neg q(b), \neg p(a), p(a), \neg q(c) & p(d), \neg q(b), q(b), p(a), \neg q(c) \\ \downarrow & \downarrow \\ \times & \times \end{array}$$

Example 6.

$$\neg\big(\forall x\exists y\,(p(x)\wedge q(y))\Rightarrow\exists y\forall x\,(p(x)\wedge q(y))\big)$$
$$\downarrow$$
$$\forall x\exists y\,(p(x)\wedge q(y))\,,\ \neg\exists y\forall x\,(p(x)\wedge q(y))$$
$$\downarrow$$
$$\forall x\exists y\,(p(x)\wedge q(y))\,,\ \exists y\,(p(c)\wedge q(y))\,,\ \neg\exists y\forall x\,(p(x)\wedge q(y))$$
$$\downarrow$$
$$\forall x\exists y\,(p(x)\wedge q(y))\,,\ (p(c)\wedge q(a))\,,\ \neg\exists y\forall x\,(p(x)\wedge q(y))$$
$$\downarrow$$
$$\forall x\exists y\,(p(x)\wedge q(y))\,,\ p(c)\,,\ q(a)\,,\ \neg\exists y\forall x\,(p(x)\wedge q(y))$$
$$\downarrow$$
$$\forall x\exists y\,(p(x)\wedge q(y))\,,\ \neg\exists y\forall x\,(p(x)\wedge q(y))\,,\ p(c)\,,\ q(a)\,,\ \neg\forall x\,(p(x)\wedge q(a))$$
$$\downarrow$$
$$\forall x\exists y\,(p(x)\wedge q(y))\,,\ \neg\exists y\forall x\,(p(x)\wedge q(y))\,,\ p(c)\,,\ q(a)\,,\ \neg(p(b)\wedge q(a))$$
$$\downarrow$$
$$\forall x\exists y\,(p(x)\wedge q(y))\,,\ \neg\exists y\forall x\,(p(x)\wedge q(y))\,,\ \exists y\,(p(b)\wedge q(y))\,,\ p(c)\,,\ q(a)\,,\ \neg(p(b)\wedge q(a))$$
$$\downarrow$$
$$\forall x\exists y\,(p(x)\wedge q(y))\,,\ \neg\exists y\forall x\,(p(x)\wedge q(y))\,,\ (p(b)\wedge q(d))\,,\ p(c)\,,\ q(a)\,,\ \neg(p(b)\wedge q(a))$$
$$\downarrow$$
$$\forall x\exists y\,(p(x)\wedge q(y))\,,\ \neg\exists y\forall x\,(p(x)\wedge q(y))\,,\ p(b)\,,\ q(d)\,,\ p(c)\,,\ q(a)\,,\ \neg(p(b)\wedge q(a))$$

$$\begin{array}{cc}
\swarrow & \searrow \\
\forall x\exists y\,(\ldots)\,,\ \neg\exists y\forall x\,(\ldots) & \forall x\exists y\,(\ldots)\,,\ \neg\exists y\forall x\,(\ldots) \\
p(b)\,,\ q(d)\,,\ p(c)\,,\ q(a)\,,\ \neg p(b) & p(b)\,,\ q(d)\,,\ p(c)\,,\ q(a)\,,\ \neg q(a) \\
\downarrow & \downarrow \\
\times & \times
\end{array}$$

Examples : the interpretation

Examples 1 and 2 suggest that *examplification*, i.e. existential instantiations, should use *fresh* constants. As constants are used as model elements, is there a risk to miss small models ? No : our basic predicate logic is missing identity ; nothing prevents us assigning the same semantic object to distinct syntactic constants. Any model in our logic can be extended into a larger model, simply by adding "clones" to existing elements. Otherwise stated, there is no quantification like $\exists! x\, P(x)$, which would be logically equivalent to $\exists x\, [P(x) \wedge \forall y(P(y) => x = y)]$.

Example 3 shows that semantic tableaux can be infinite, when the corresponding formula is consistent but has only infinite models. Hopefully, this will not preclude completeness, which is concerned with inconsistent formulas.

Example 4 shows however that unfair selections in the construction process might prevent inconsistency detection. The culprit is universal formula instantiation, which can be applied endlessly, and must be applied every time a new constant appears on an open branch.

Decomposition rules

— *Prolongation rules (α-rules) and ramification rules (β-rules)*

α	α_1	α_2
$A_1 \wedge A_2$	A_1	A_2
$\neg(A_1 \vee A_2)$	$\neg A_1$	$\neg A_2$
$\neg(A_1 \Rightarrow A_2)$	A_1	$\neg A_2$
$\neg(A_1 \Leftarrow A_2)$	$\neg A_1$	A_2

β	β_1	β_2
$B_1 \vee B_2$	B_1	B_2
$\neg(B_1 \wedge B_2)$	$\neg B_1$	$\neg B_2$
$B_1 \Rightarrow B_2$	$\neg B_1$	B_2
$B_1 \Leftarrow B_2$	B_1	$\neg B_2$

— *Generative rule (γ-rule) and examplification rule (δ-rule)*

γ	$\gamma(a)$
$\forall x\, A(x)$	$\forall x\, A(x)\,,\; A(c)$
$\neg\exists x\, A(x)$	$\neg\exists x\, A(x)\,,\; \neg A(c)$

δ	$\delta(a)$
$\exists x\, A(x)$	$A(a)$
$\neg\forall x\, A(x)$	$\neg A(a)$

The choice of constant c is free but a must be a fresh constant.

Construction of a semantic tableau

Init : a root labelled $\{A\}$.

Induction step : select an unmarked leaf ℓ ; let $U(\ell)$ the labelling formula set..

- If $U(\ell)$ contains a complementary pair, mark ℓ as *closed* '$\times$' ;
- If $U(\ell)$ is not a literal set, select a non-literal formula in $U(\ell)$:
  - if it is an α-formula A, create a single child ℓ' and label it with
    $U(\ell') = (U(\ell) - \{A\}) \cup \{\alpha_1, \alpha_2\}$;
  - if it is a β-formula B, create two children ℓ' and ℓ'' and label them with
    $U(\ell') = (U(\ell) - \{B\}) \cup \{\beta_1\}$ and $U(\ell'') = (U(\ell) - \{B\}) \cup \{\beta_2\}$;
  - if it is a γ-formula C, create a single child ℓ' and label it with
    $U(\ell') = U(\ell) \cup \{\gamma(c)\}$, where c is a constant occurring in $U(\ell)$ (if any).
  - if it is a δ-formula D, create a single child ℓ' and label it with
    $U(\ell') = (U(\ell) - \{D\}) \cup \{\delta(a)\}$ where a is a fresh constant, not occurring in $U(\ell)$.

Termination : occurs when each leaf is either closed, or contains only literals and fully instantiated γ-formulas ; such a leaf can be marked open.
The γ-rule may prevent termination.

Semantic tableaux, soundness

Theorem. If $T(A)$ is closed then A is inconsistent.

Proof. We show that all labelling sets in a closed tableau are inconsistent, by induction on the height of the nodes.

— $h = 0$: n is a closed leaf, so $U(n)$ is inconsistent.

— $h > 0$: a rule, α, β, γ or δ, has been used to create n's child(ren).

α-**rule or** β-**rule** : like in propositional logic.

γ-**rule** :

$$\begin{array}{rcc} n & : & \{\forall x\, A(x)\} \cup U_0 \\ & & \downarrow \\ n' & : & \{\forall x\, A(x), A(a)\} \cup U_0 \end{array}$$

$U(n')$ is inconsistent (induction hypothesis),
so $U(n)$ is inconsistent (why ?).

Regle δ :

$$\begin{array}{rcc} n & : & \{\exists x\, A(x)\} \cup U_0 \\ & & \downarrow \\ n' & : & \{A(a)\} \cup U_0 \end{array}$$

where a does not occur in any formula of $U(n)$.
Should $U(n)$ be consistent, a valuation $\mathcal{I} = (D, I_c, I_v)$: $\mathcal{I}[\exists x\, A(x)] = T$, would exist, and a $d \in D$ such that $\mathcal{I}_{x/d}[A(x)] = T$.
Define $\mathcal{J} = (D, J_c, I_v)$ with J_c like I_c but with $J_c[a] = d$.
Then, $\mathcal{J}[A(a)] = T$ and $\mathcal{J}[U_0] = \mathcal{I}[U_0] = T$,
so $\mathcal{J}$ would be a model of $U(n')$, which is impossible.

Construction strategy

Two conditions must be fulfilled.

— Every non literal formula on an open branch is decomposed on this branch.

— For each γ-formula A and each constant a occurring on an open branch, an a-instantiation of A occurs on the branch.

Hintikka set, definition

Definition. Let U a formula set and C_U the set of constants occurring in U.
U is a *Hintikka set* if five conditions are fulfilled :

1. No complementary pair is included in U.
2. If $\alpha \in U$ is an α-formula, then $\alpha_1 \in U$ and $\alpha_2 \in U$.
3. If $\beta \in U$ is a β-formula, then $\beta_1 \in U$ or $\beta_2 \in U$.
4. If γ is a γ-formula, then for each $a \in C_U$, $\gamma(a) \in U$.
5. If δ is a δ-formula, then for some $a \in C_U$, $\delta(a) \in U$.

Open branch lemma

Lemma. The union of the sets labelling an open branch is a Hintikka set.

Proof. The construction algorithm ensures that each of the five conditions is fulfilled . . . provided the construction strategy is implemented.

Comment. Infinite branches are open branches ; the lemma also applies to them.

Model construction for Hintikka sets

Hintikka's theorem. All Hintikka sets are consistent.

Proof. Let U be a Hintikka set. Its canonical model $\mathcal{I}_U = (D, I_c, I_v)$ is easily defined :

1. $D = \{a, b, \ldots,\}$ is the set of constants occurring in U ;
2. - For each $d \in D$:$I_c[d] = d$.
   - For each predicate symbol p (arity m) occurring in U :
   $I_c[p](I_c[a_1], \ldots, I_c[a_m]))] = T$ if $p(a_1, \ldots, a_m) \in U$
   $I_c[p](I_c[a_1], \ldots, I_c[a_m]))] = F$ if $p(a_1, \ldots, a_m) \notin U$
3. I_v is arbitrary (no free variable).

It is easy to check by structural induction that this interpretation is indeed a model.

Corollary. If $T(A)$ is open, it has an open branch ; the union of the labels of this branch is a Hintikka set containing A so A is consistent. As a result, the semantic tableaux method is complete.

Conclusion

A is inconsistent if and only if $T(A)$ is closed.

A is valid if and only if $T(\neg A)$ is closed.

A is contingent if and only if both $T(A)$ and $T(\neg A)$ are open.

Non-termination occurs only in the open case (as long as the construction strategy is implemented) ; that is the reason why soundness and completeness are preserved.

HILBERT SYSTEM

Formal system $\mathcal{H}$ contains five axioms and two inference rules.

1. $\vdash A \Rightarrow (B \Rightarrow A)$
2. $\vdash (A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$
3. $\vdash (\neg B \Rightarrow \neg A) \Rightarrow (A \Rightarrow B)$
4. $\vdash \forall x\, A(x) \Rightarrow A(t)$ (capture is not allowed)
5. $\vdash \forall x(A \Rightarrow B(x)) \Rightarrow (A \Rightarrow \forall x\, B(x))$ where x is not free in A

MP $\dfrac{\vdash A \qquad \vdash A \Rightarrow B}{\vdash B}$ Modus Ponens

Gen $\dfrac{\vdash A(x)}{\vdash \forall x\, A(x)}$ Generalisation rule

Deduction rule

Proof, derivation : just as in the propositional case.

Deduction rule.

$$\frac{U, A \vdash B}{U \vdash A \Rightarrow B}$$

Restriction : free variables in A cannot be generalised.

If $C(x)$ is a theorem, then $\forall x\, C(x)$ is a theorem,
but $C(x) \Rightarrow \forall x\, C(x)$ is (in general) not a theorem.

Comment. There is no problem with closed hypotheses.

Soundness of the deduction rule

How to convert a derivation for $U, A \vdash B$ into a derivation for $U \vdash (A \Rightarrow B)$? As in the propositional case, except for

$$
\begin{aligned}
&U, A \vdash C(x)\,, \\
&U, A \vdash \forall x\, C(x)\,.
\end{aligned}
$$

The conversion is

$$
\begin{aligned}
&U \vdash A \Rightarrow C(x)\,, \\
&U \vdash \forall x\,(A \Rightarrow C(x))\,, \\
&U \vdash \forall x\,(A \Rightarrow C(x)) \Rightarrow (A \Rightarrow \forall x\, C(x))\,, \\
&U \vdash A \Rightarrow \forall x\, C(x)\,.
\end{aligned}
$$

The third line is correct only if x does not occur in A, hence the aforementioned restriction.

Uniform substitution, replacement

Thee *uniform substitution principle* still applies in predicate logic. For instance,

$$\neg p \Rightarrow (p \Rightarrow q)$$

is a theorem, therefore

$$\neg A \Rightarrow (A \Rightarrow B)$$

is a theorem scheme (a tautology scheme), and

$$\neg \forall x\, P(x) \Rightarrow (\forall x\, P(x) \Rightarrow \forall y\, (R(y) \Rightarrow Q(z)))$$

is a theorem.

The *replacement theorem* also applies.
If $A \equiv B$ is a theorem and if C is a theorem, then any formula obtained by replacing some occurrences of A by B in C is also a theorem.

Exercise. Provide justification and determine whether the capture phenomenon can be troublesome here.

PC-rule

$A =_{def}\ [p \Rightarrow \neg q] \equiv [q \Rightarrow \neg p]\,,$
$B =_{def}\ [P(x,y) \Rightarrow \neg\forall z Q(z,a)] \equiv [\forall z Q(z,a) \Rightarrow \neg P(x,y)]\,.$
Since A is a theorem, B is also a theorem.

PC-rule.
If B is a (predicate) instance of a (propositional) tautology, then $\vdash B$.

Justification.
It is easy to convert a proof of A into a proof of B.
Comments. When $\vee$, $\wedge$ and $\equiv$ are allowed, they stand for mere abbreviations. For instance, $A \wedge B$ stands for $\neg(A \Rightarrow \neg B)$.

Comment.
"$\exists$" is also introduced as a mere abbreviation ; $\exists x \phi$ stands for $\neg\forall x \neg\phi$.

Simulation of tableaux rules within Hilbert system

Every use of α, β, γ, δ-rules can be simulated within Hilbert system.

$$\gamma : \quad \frac{\vdash V \wedge \forall x\, A(x)}{\vdash V \wedge \forall x\, A(x) \wedge A(c)} \qquad \frac{\vdash V \vee \exists x\, A(x) \vee A(c)}{\vdash V \vee \exists x\, A(x)}$$

1.	$\vdash \forall x \neg A(x) \Rightarrow \neg A(c)$	(Axiom 4)
2.	$\vdash \neg\forall x \neg A(x) \vee \neg A(c)$	(PC 1)
3.	$\vdash V \vee \neg\forall x \neg A(x) \vee \neg A(c)$	(PC 2)
4.	$\vdash V \vee \exists x\, A(x) \vee \neg A(c)$	($\exists$)
5.	$\vdash V \vee \exists x\, A(x) \vee A(c)$	(hypothesis)
6.	$\vdash V \vee \exists x\, A(x)$	(PC 4, 5)

$$\delta : \quad \frac{\vdash V \wedge \exists x A(x)}{\vdash V \wedge A(x)} \qquad \frac{\vdash V \vee A(x)}{\vdash V \vee \forall x A(x)}$$

1.	$\vdash V \vee A(x)$	(hypothesis)
2.	$\vdash \neg V \Rightarrow A(x)$	(PC 1)
3.	$\vdash \forall x(\neg V \Rightarrow A(x))$	(Generalisation 2)
4.	$\vdash \neg V \Rightarrow \forall x\, A(x)$	(Axiom 5, PC 4)
5.	$\vdash V \vee \forall x\, A(x)$	(PC 4)

Some examples of (proved) theorems

Theorem. $\vdash p(a) \Rightarrow \exists x\, p(x)$

1.	$\vdash \forall x \neg p(x) \Rightarrow \neg p(a)$	(Axiom 4)
2.	$\vdash p(a) \Rightarrow \neg\forall x \neg p(x)$	(PC 1)
3.	$\vdash p(a) \Rightarrow \exists x p(x)$	(Def. $\exists$)

Theorem. $\vdash (A \Rightarrow \forall x\, C(x)) \Rightarrow \forall x\, (A \Rightarrow \forall x\, C(x))$ if no free x in A.

1.	$A,\, A \Rightarrow \forall x\, C(x) \vdash \forall x\, C(x)$	(Hypoth., MP)
2.	$A,\, A \Rightarrow \forall x\, C(x) \vdash C(x)$	(Ax. 4, 1)
3.	$A \Rightarrow \forall x\, C(x) \vdash (A \Rightarrow C(x))$	(Deduction, 2)
4.	$A \Rightarrow \forall x\, C(x) \vdash \forall x\, (A \Rightarrow C(x))$	(Gener., 3)
5.	$\vdash (A \Rightarrow \forall x\, C(x)) \Rightarrow \forall x\, (A \Rightarrow C(x))$	(Deduction, 4)

Theorem. $\vdash \forall x\, (p(x) \Rightarrow q) \equiv \exists x\, p(x) \Rightarrow q$ if no free x in q.

1.	$\forall x\, (p(x) \Rightarrow q) \vdash \forall x\, (p(x) \Rightarrow q)$	(Hypothesis)
2.	$\forall x\, (p(x) \Rightarrow q) \vdash \forall x\, (\neg q \Rightarrow \neg p(x))$	(PC, repl., 1)
3.	$\forall x\, (p(x) \Rightarrow q) \vdash \neg q \Rightarrow \forall x\, \neg p(x))$	(Ax. 5, 2)
4.	$\forall x\, (p(x) \Rightarrow q) \vdash \exists x\, p(x) \Rightarrow q$	(PC, $\exists$, 3)
5.	$\exists x\, p(x) \Rightarrow q \vdash \exists x\, p(x) \Rightarrow q$	(Hypothesis)
6.	$\exists x\, p(x) \Rightarrow q \vdash \neg q \Rightarrow \forall x\, \neg p(x)$	(PC, $\exists$, 5)
7.	$\exists x\, p(x) \Rightarrow q \vdash \forall x\, (\neg q \Rightarrow \neg p(x))$	(Theorem, 6)
8.	$\exists x\, p(x) \Rightarrow q \vdash \forall x\, (p(x) \Rightarrow q)$	(PC, 7)
9.	$\vdash \forall x\, (p(x) \Rightarrow q) \equiv \exists x\, p(x) \Rightarrow q$	(Deduction, 4, 8)

Constant rule I

In order to use an existential hypothesis $\exists x\, p(x)$, it is customary to say "assume a such that $p(a)$" (where a is a fresh constant). The C-rule formalizes this.

Theorem. (C-Rule). If $U \vdash \exists x\, p(x)$, if there is no free x in U nor in A and if $U, p(x) \vdash A$ can be derived without generalisation on x, then $U \vdash A$.

Comment. Forbidding generalisation on x is essential, otherwise $\exists x\, p(x) \vdash \forall x\, p(x)$ could be derived :

1.	$\exists x\, p(x) \vdash \exists x\, p(x)$	(hypothesis)
2.	$\exists x\, p(x)\,, p(x) \vdash p(x)$	(hypothesis)
3.	$\exists x\, p(x)\,, p(x) \vdash \forall x\, p(x)$	(Generalisation)
4.	$\exists x\, p(x) \vdash \forall x\, p(x)$	(incorrect use of C-rule)

Proof.

1.	$U, p(x) \vdash A$	(hypothesis)
2.	$U \vdash p(x) \Rightarrow A$	(Deduction 1)
3.	$U \vdash \forall x(p(x) \Rightarrow A)$	(Generalisation 2)
4.	$U \vdash \exists x\, p(x) \Rightarrow A$	(Theorem)
5.	$U \vdash \exists x\, p(x)$	(hypothesis)
6.	$U \vdash A$	(PC 4,5)

Comment. Obviously the conversion of a derivation of $U \vdash \exists x\, p(x)$ into a derivation of $U \vdash p(x)$ or of $U \vdash p(a)$ (a fresh constant) should and does remain impossible ; in particular, $\exists x\, p(x) \vdash p(a)$ is clearly incorrect.

Constant rule II

A direct derivation of

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

might be tricky, but C-rule makes things easier by reduction to

$$\forall y\, p(a,y) \Rightarrow \forall y \exists x\, p(x,y)$$

or to

$$p(a,y) \Rightarrow \exists x\, p(x,y)$$

which is obvious.

Comment. Some authors allow

$$\frac{U \vdash \exists x\, p(x)}{U \vdash p(a)}$$

where a is a fresh constant ; should this be accepted, other rules should be weakened in order to block

$$\frac{U \vdash p(a)}{U \vdash \forall x\, p(x)}$$

Comment. A fresh constant has no specific meaning. For instance, in group theory, the neuter constant e has a specific meaning, so the "derivation"

1.	$U \vdash \forall x\, [x * i(x) = e]$	(hypothese)
2.	$U \vdash \forall y\, \forall x\, [x * i(x) = y]$	(Generalisation 1)

is clearly incorrect.

Soundness and completeness

Soundness of predicate Hilbert system is easily proved, just as in the propositional case.

Completeness also can be proved in a similar way, but since truthtables do not extend to predicate logic, semantic tableaux are used instead.

Since tableaux rules can be simulated within Hilbert system, we need only to chain applications of the simulating derived rules into a full proof.

If some predicate formula A is valid, there is a r-rooted closed tableau for $\neg A$. For each node n labelled with S_n, the associated formula H_n is the disjunction of the negations of S_n-elements ; H_n is always a valid formula. If node n has a single child n', then $H_{n'} \vdash H_n$ holds ; if node n has two children n' and n'', then $H_{n'}, H_{n''} \vdash H_n$ holds. Besides, for each leaf n, $\vdash H_n$ holds. It is now easy to chain all these derivations into a proof of H_r, which is simply A.

FUNCTION SYMBOLS

Mathematical formulas like

$$x > y \Rightarrow (x+1) > (y+1)$$

or, in prefix notation,

$$> (x, y) \Rightarrow\ > (+(x, 1), +(y, 1))\ ,$$

are instances of logical formulas like

$$p(x, y) \Rightarrow p(f(x, a), f(y, a))\ ,$$

provided function symbols are allowed.

We therefore introduce :

— $\mathcal{F} = \{f, g, h, \ldots\}$: a set of *function symbols* (each with its arity).

This, with predicate symbols, constants (often seen as 0-ary functions) and variables, makes our lexicon.

SYNTAX OF PREDICATE LOGIC

The syntax of *terms* is generalized ; the syntax for formulas does not change.

The concept of *term* is recursively defined :

— A *variable* is a term.
— A *constant* is a term.
— If f is an m-ary *function symbol* and if $t_1, t_2, \ldots, t_m$ are *terms*, then $f(t_1, \ldots, t_m)$ is a term.

Comments. Constants are 0-ary function symbols. A term is *closed* if no variable occurs in it.

Examples (terms) :

$$a \qquad x \qquad f(a,x) \qquad g(f(a)) \qquad f\big(g\big(x,h(y)\big)\big)$$

Examples (atoms) :

$$p(a,b) \qquad p\big(x,f(a,x)\big) \qquad p\Big(f(a,b),f\big(g(x),g(x)\big)\Big)$$

SEMANTICS OF PREDICATE LOGIC

An *interpretation* $\mathcal{I}$ is a triple (D, I_c, I_v) such that :

- D is a non empty set, the *domain* ;
- I_c is a function which maps
  - to each *constant* a, some object $I_c[a]$, an element of D,
  - to each m-ary *function symbol* f, a function $I_c[f]$ of type $D^m \mapsto D$;
  - to each n-art predicate symbol p, a n-ary relation on D, i.e. a function $I_c[p]$ of type $D^n \mapsto \{T, F\}$;
- I_v is a function that associates with each variable x an element $I_v[x]$ of D.

Interpretation rules

Let $\mathcal{I} = (D, I_c, I_v)$ be an interpretation.

— If x is a (free) variable, then $\mathcal{I}[x] = I_v[x]$.

— If a is a constant, then $\mathcal{I}[a] = I_c[a]$.

— If f is an m-ary function symbol and if $t_1, t_2, \ldots, t_m$ are terms, then $\mathcal{I}[f(t_1, t_2, \ldots, t_m)] = I_c[f](\mathcal{I}[t_1], \mathcal{I}[t_2], \ldots, \mathcal{I}[t_m])$.

Interpretation rules for formulas do not change.

Example : Formula

$$\forall x \forall y \Big(p(x, y) \Rightarrow p\big(f(x, a), f(y, a)\big) \Big)$$

is satisfied by interpretation

$$\mathcal{I}_1 = (\mathbf{Z}, I_c, I_v) \; : \; I_c[a] = 1, \; I_c[f] = +, \; I_c[p] = \leq,$$

but not by interpretation

$$\mathcal{I}_2 = (\mathbf{Z}, I_c, I_v) \; : \; I_c[a] = -1, \; I_c[f] = *, \; I_c[p] = > .$$

Prenex form

A formula is in *prenex form* if it has the form

$$\underbrace{Q_1 x_1 \cdots Q_n x_n}_{\textit{prefix}} \; \underbrace{\mathsf{M}}_{\textit{matrix}}$$

where each Q_i is either $\forall$ or $\exists$, for $i = 1, \ldots, n$ and where the *matrix* M is a quantification-free formula.

Comments. We usually assume that all quantified variables have some (free) occurrence in the matrix. The scope of the prefix must be the whole matrix.

Theorem. For every predicate formula, some logically equivalent prenex form always exists.

Reduction to prenex form

Example : $\forall x\Big(p(x) \wedge \neg\exists y\forall x\neg\big(\neg q(x,y) \Rightarrow \forall z\, r(a,x,y)\big)\Big)$.

1. Eliminate all Boolean connectives except $\neg$, $\vee$, $\wedge$.
   Ex. : $\forall x\Big(p(x) \wedge \neg\exists y\forall x\neg\big(\neg\neg q(x,y) \vee \forall z\, r(a,x,y)\big)\Big)$.
2. Rename bound variables (if necessary) in such a way no variable has both free and bound occurrence in any subformula.
   Ex. : $\forall x\Big(p(x) \wedge \neg\exists y\forall u\neg\big(\neg\neg q(u,y) \vee \forall z\, r(a,u,y)\big)\Big)$.
3. Eliminate spurious quantifications.
   Ex. : $\forall x\Big(p(x) \wedge \neg\exists y\forall u\neg\big(\neg\neg q(u,y) \vee r(a,u,y)\big)\Big)$.
4. Propagate $\neg$ occurences downwards (the syntactic tree) and eliminate double negations (so only atomic subformulas can be negated).

   $$\neg\forall x A \to \exists x\neg A\,, \quad \neg\exists x A \to \forall x\neg A\,, \quad \neg\neg C \to C$$

   Ex. : $\forall x\Big(p(x) \wedge \forall y\exists u\big(q(u,y) \vee r(a,u,y)\big)\Big)$.
5. Propagate quantifications upwards.

   $\forall x A \wedge \forall x B \to \forall x(A \wedge B) \qquad \exists x A \vee \exists x B \to \exists x(A \vee B)$
   if x does not occur in B :
   $\forall x\, A \wedge B \ \to\ \forall x(A \wedge B) \qquad \exists x\, A \vee B \ \to\ \exists x(A \vee B)$
   $\forall x\, A \vee B \ \to\ \forall x(A \vee B) \qquad \exists x\, A \wedge B \ \to\ \exists x(A \wedge B)$
   Rename if necessary : $\exists x\, p(x) \wedge \forall x\, q(x) \ \to\ \exists x\, p(x) \wedge \forall y\, q(y) \ \to$ $\exists x\,\forall y\,(p(x) \wedge q(y))$.
   Ex. : $\forall x\forall y\exists u\Big(p(x) \wedge \big(q(u,y) \vee r(a,u,y)\big)\Big)$.

Clauses, cubes, normal forms

- A *literal* is an atom or the negation of an atom.
- A *clause* (a *cube*) is a disjunction (a conjunction) of literals.
- A *conjunctive (disjunctive) normal form* is a conjunction (disjunction) of clauses (of cubes).
- A prenex form is *conjunctive* (*disjunctive*) if its matrix is in conjunctive (disjunctive) normal form.

Skolem form

A *Skolem form* is a prenex form with only universal quantifications. A Skolem form can be *associated with* every prenex form, according to the following algorithm.

For each existential quantification $\exists x$ in the scope of $k \geq 0$ universal quantifications $(\forall x_1 \cdots \forall x_k)$,

1. replace each occurrence of x in the matrix by $f(x_1, \ldots, x_k)$ where f is a *fresh* k-ary function symbol ($k = 0$: x replaced by a fresh constant),
2. suppress the quantification $\exists x$.

Examples :

— $\forall x \forall y \exists u \left(q(u, y) \Rightarrow r(a, u, y, z)\right)$
is associated with
$\forall x \forall y \left(q(f(x, y), y) \Rightarrow r(a, f(x, y), y, z)\right)$.

— $\forall x \exists u \forall v \exists w \forall x \forall y \exists z \, M(u, v, w, x, y, z)$
is simplified into
$\exists u \forall v \exists w \forall x \forall y \exists z \, M(u, v, w, x, y, z)$
and then associated with
$\forall v \forall x \forall y \, M(a, v, f(v), x, y, g(v, x, y))$.

Why using Skolem forms ?

Formula $A =_{def} \forall x \forall y \exists u \, [q(u,y) \Rightarrow r(a,u,y,x)]$
asserts the existence of some u, depending on x and y,
such that $q(u,y) \Rightarrow r(a,u,y,x)$ is true.
Associating S_A with A amounts to *name* this u ; the name $f(x,y)$
emphasises dependency. Symbol f denotes a *choice function*.

Formulas A and S_A are usually not logically equivalent
although the former is a logical consequence of the latter.

But A and S_A are either both consistent, or both inconsistent.

Skolem theorem

Theorem. Skolem form S_A is consistent if and only if prenex form A is consistent.

Proof. Every model of S_A is also a model of A, and every model of A can be extended into a model of S_A (details are left to the reader).

Categorical syllogism theory

Predicate logic is undecidable, but some fragments are decidable.

The simplest one of these interesting fragments is the *categorical syllogism theory*, which is based on a single formula :

$$\forall x \, (P(x) \Rightarrow Q(x))\,.$$

This theory was introduced by Aristotle (384 BC – 322 BC) and perfected by his successors ; it was the central part of logic through the Middle Ages and even until George Boole and Gottlob Frege (19th century).

The basic formula and its variants I

$\boxed{\forall x\,(P(x) \Rightarrow Q(x))}$	**A**	$\neg\exists x\,(P(x) \wedge \neg Q(x))$
$\boxed{\forall x\,(P(x) \Rightarrow \neg Q(x))}$	**E**	$\neg\exists x\,(P(x) \wedge Q(x))$
$\neg\forall x\,(P(x) \Rightarrow \neg Q(x))$	**I**	$\boxed{\exists x\,(P(x) \wedge Q(x))}$
$\neg\forall x\,(P(x) \Rightarrow Q(x))$	**O**	$\boxed{\exists x\,(P(x) \wedge \neg Q(x))}$

"**A**ff**I**rmo" "n**E**g**O**"

The basic formula and its variants II

If roles of P and Q can be switched, eight "$\{P, Q\}$-formulas" are obtained.

A	universal affirmative
E	universal negative
I	particular affirmative
O	particular negative

"**A**ff**I**rmo"
"n**E**g**O**"

The syllogistic game

Pick a $\{P, Q\}$-formula, a $\{Q, R\}$-formula and a $\{P, R\}$-formula and determine whether the third one (the *conclusion*) is a logical consequence of the first ones (the *premises*).

There are $8^3 = 512$ possibilities.

As far as P and R have similar roles,
we restrict to the case where the conclusion is PR (and not RP) ;
256 possibilities are left.

Minor, Major, Midterm, Syllogism

The Minor is $P(x)$ (and also the premise containing it).
The Major is $R(x)$ (and also the premise containing it).
The Midterm is $Q(x)$.

(The word "term" is traditional here, even if a predicate is not a term.)

The (categorical) syllogism is the inference rule

$$\frac{\textit{Major} \qquad\qquad \textit{Minor}}{\textit{Conclusion}}$$

It can be valid, correct, sound (if the conclusion is a logical consequence of the premises) or not.

Examples (French)

Tout sot est ennuyeux
Or certains bavards ne sont pas ennuyeux
Donc certains bavards ne sont pas sots

Les puissants ne sont pas miséricordieux
Or les enfants ne sont pas puissants
Donc les enfants ne sont pas miséricordieux

Certains poètes sont agréables
Or tous les poètes sont des génies
Donc certains génies sont agréables

Tout ce qui est vénéneux est nuisible
Or certains champignons sont nuisibles
Donc certains champignons sont vénéneux

Examples (bis)

Tout sot est ennuyeux
Or certains bavards ne sont pas ennuyeux
Donc certains bavards ne sont pas sots

baroco

Les puissants ne sont pas miséricordieux
Or les enfants ne sont pas puissants
Donc les enfants ne sont pas miséricordieux

invalid

Certains poètes sont agréables
Or tous les poètes sont des génies
Donc certains génies sont agréables

disamis

Tout ce qui est vénéneux est nuisible
Or certains champignons sont nuisibles
Donc certains champignons sont vénéneux

invalid

Figures and modes

Figure :	First	Second	Third	Fourth
Major	QR	RQ	QR	RQ
Minor	PQ	PQ	QP	QP
Conclusion	PR	PR	PR	PR

The *mode* records the nature of the premises and the conclusion. Mode AEI means an A-major, an E-minor and a I-conclusion. There are $4^3 = 64$ possible modes, each of them present in the four figures.

Only 12 modes give rise to at least one valid syllogism ; they are

AAA AAI AEE AEO AII AOO EAE EAO EIO IAI IEO OAO

Mnemonic rules for potentially valid modes

1. If both premises are negative, the syllogism is not valid.
2. If both premises are particular, the syllogism is not valid.
3. If one premise is negative, the conclusion must be negative.
4. If one premise is particular, the conclusion must be particular.
5. If both premises are affirmative, the conclusion must be affirmative.

Examples. First statement rules out modes like EEE and OEO ; last statement rules out modes like AIO and IIE.

Between them, potentially valid modes give rise to no more than 15 valid syllogisms, although 9 more are said "pseudo-valid" or "quasi-valid" .

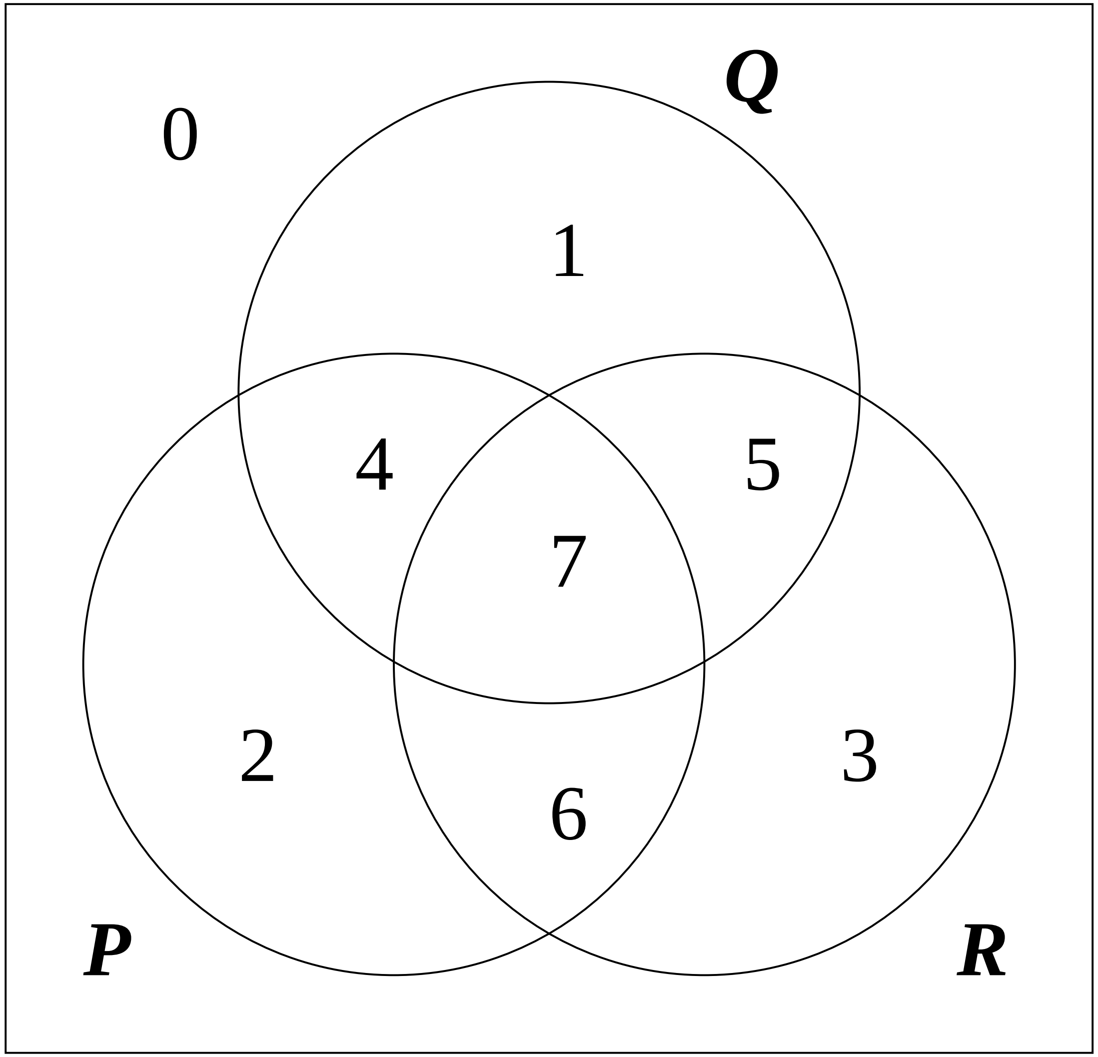
Systematic naming technique

The order of the premises does not matter but it is convenient to list Major first.

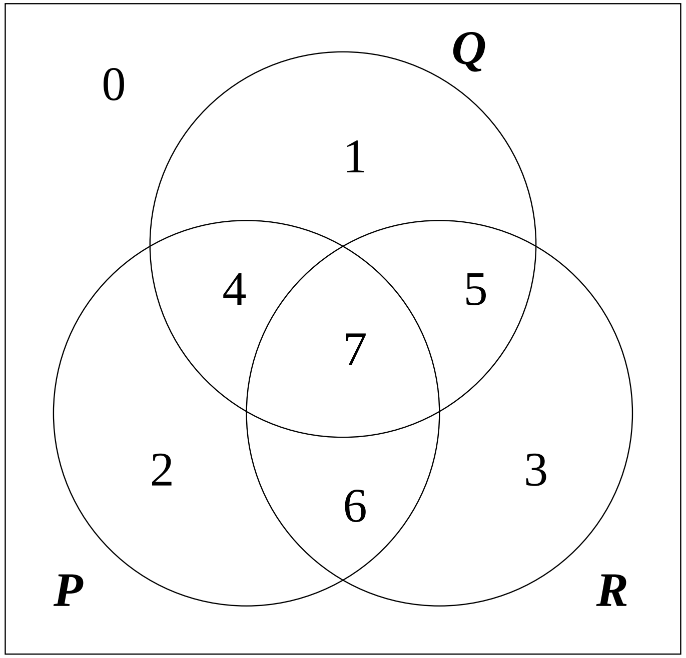
> `(M)` *All humans are mortal.*
> `(m)` *All Greeks are humans.*
> `(C)` *All Greeks are mortal.*

First figure, mode AAA ; noted AAA-1 or **barbara**.

Venn diagrams I



Venn diagrams II



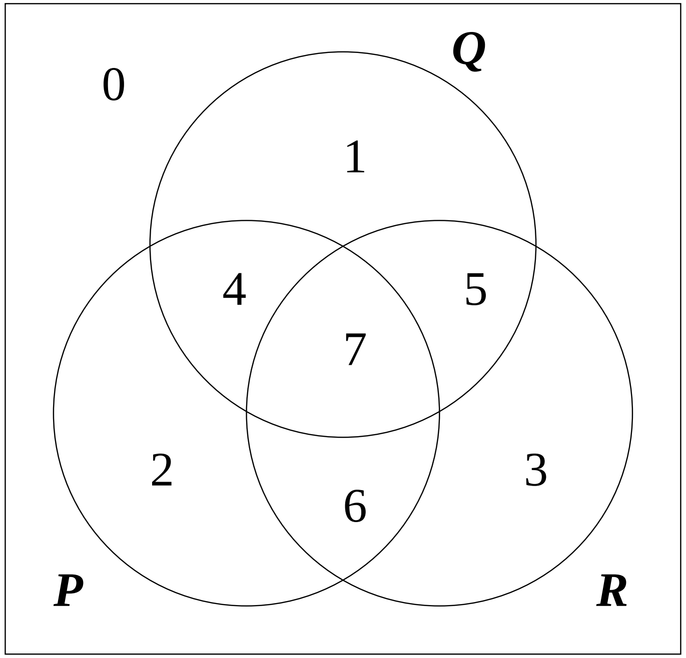
Syllogism AOO-2 baroco.

Major $\forall x\,(R(x) \Rightarrow Q(x))$ 3 and 6 are empty.

Minor $\exists x\,(P(x) \land \neg Q(x))$ 2 or 6 is non-empty.

Conclusion $\exists x\,(P(x) \land \neg R(x))$ 2 or 4 is non-empty.

Venn diagrams III



Syllogism AIO-4	*invalid*.
Major $\forall x\,(R(x) \Rightarrow Q(x))$	$3 \cup 6 = \emptyset$.
Minor $\exists x\,(Q(x) \wedge P(x))$	$4 \cup 7 \neq \emptyset$.
Conclusion $\exists x\,(P(x) \wedge \neg R(x))$	$2 \cup 4 \neq \emptyset$? ? ?

First figure

barbara	celarent
$\dfrac{\forall x\,[B(x) \Rightarrow C(x)] \quad \forall x\,[A(x) \Rightarrow B(x)]}{\forall x\,[A(x) \Rightarrow C(x)]}$	$\dfrac{\forall x\,[B(x) \Rightarrow \neg C(x)] \quad \forall x\,[A(x) \Rightarrow B(x)]}{\forall x\,[A(x) \Rightarrow \neg C(x)]}$
darii	ferio
$\dfrac{\forall x\,[B(x) \Rightarrow C(x)] \quad \exists x\,[A(x) \wedge B(x)]}{\exists x\,[A(x) \wedge C(x)]}$	$\dfrac{\forall x\,[B(x) \Rightarrow \neg C(x)] \quad \exists x\,[A(x) \wedge B(x)]}{\exists x\,[A(x) \wedge \neg C(x)]}$
barbari	celaro
$\dfrac{\exists x\, A(x) \quad \forall x\,[B(x) \Rightarrow C(x)] \quad \forall x\,[A(x) \Rightarrow B(x)]}{\exists x\,[A(x) \wedge C(x)]}$	$\dfrac{\exists x\, A(x) \quad \forall x\,[B(x) \Rightarrow \neg C(x)] \quad \forall x\,[A(x) \Rightarrow B(x)]}{\exists x\,[A(x) \wedge \neg C(x)]}$

Second figure

camestres $\forall x\,[C(x) \Rightarrow B(x)]$ $\forall x\,[A(x) \Rightarrow \neg B(x)]$ ——— $\forall x\,[A(x) \Rightarrow \neg C(x)]$	cesare $\forall x\,[C(x) \Rightarrow \neg B(x)]$ $\forall x\,[A(x) \Rightarrow B(x)]$ ——— $\forall x\,[A(x) \Rightarrow \neg C(x)]$
baroco $\forall x\,[C(x) \Rightarrow B(x)]$ $\exists x\,[A(x) \wedge \neg B(x)]$ ——— $\exists x\,[A(x) \wedge \neg C(x)]$	festino $\forall x\,[C(x) \Rightarrow \neg B(x)]$ $\exists x\,[A(x) \wedge B(x)]$ ——— $\exists x\,[A(x) \wedge \neg C(x)]$
camestro $\exists x\, A(x)$ $\forall x\,[C(x) \Rightarrow B(x)]$ $\forall x\,[A(x) \Rightarrow \neg B(x)]$ ——— $\exists x\,[A(x) \wedge \neg C(x)]$	cesaro $\exists x\, A(x)$ $\forall x\,[C(x) \Rightarrow \neg B(x)]$ $\forall x\,[A(x) \Rightarrow B(x)]$ ——— $\exists x\,[A(x) \wedge \neg C(x)]$

Third figure

datisi $\forall x\,[B(x) \Rightarrow C(x)]$ $\exists x\,[B(x) \wedge A(x)]$ ——— $\exists x\,[A(x) \wedge C(x)]$	disamis $\exists x\,[B(x) \wedge C(x)]$ $\forall x\,[B(x) \Rightarrow A(x)]$ ——— $\exists x\,[A(x) \wedge C(x)]$
bocardo $\exists x\,[B(x) \wedge \neg C(x)]$ $\forall x\,[B(x) \Rightarrow A(x)]$ ——— $\exists x\,[A(x) \wedge \neg C(x)]$	ferison $\forall x\,[B(x) \Rightarrow \neg C(x)]$ $\exists x\,[B(x) \wedge A(x)]$ ——— $\exists x\,[A(x) \wedge \neg C(x)]$
darapti $\exists x\, B(x)$ $\forall x\,[B(x) \Rightarrow C(x)]$ $\forall x\,[B(x) \Rightarrow A(x)]$ ——— $\exists x\,[A(x) \wedge C(x)]$	felapton $\exists x\, B(x)$ $\forall x\,[B(x) \Rightarrow \neg C(x)]$ $\forall x\,[B(x) \Rightarrow A(x)]$ ——— $\exists x\,[A(x) \wedge \neg C(x)]$

Fourth figure

<u>camenes</u> $\forall x\,[C(x) \Rightarrow B(x)]$ $\forall x\,[B(x) \Rightarrow \neg A(x)]$ ――― $\forall x\,[A(x) \Rightarrow \neg C(x)]$	<u>dimaris</u> $\exists x\,[C(x) \wedge B(x)]$ $\forall x\,[B(x) \Rightarrow A(x)]$ ――― $\exists x\,[A(x) \wedge C(x)]$
cameno $\exists x\, A(x)$ $\forall x\,[C(x) \Rightarrow B(x)]$ $\forall x\,[B(x) \Rightarrow \neg A(x)]$ ――― $\exists x\,[A(x) \wedge \neg C(x)]$	<u>fresison</u> $\forall x\,[C(x) \Rightarrow \neg B(x)]$ $\exists x\,[B(x) \wedge A(x)]$ ――― $\exists x\,[A(x) \wedge \neg C(x)]$
bramantip $\exists x\, C(x)$ $\forall x\,[C(x) \Rightarrow B(x)]$ $\forall x\,[B(x) \Rightarrow A(x)]$ ――― $\exists x\,[A(x) \wedge C(x)]$	fesapo $\exists x\, B(x)$ $\forall x\,[C(x) \Rightarrow \neg B(x)]$ $\forall x\,[B(x) \Rightarrow A(x)]$ ――― $\exists x\,[A(x) \wedge \neg C(x)]$

Pseudo-valid syllogisms

Pseudo-valid syllogisms are invalid syllogisms that become valid if some existance assumption is done and used as an implicit third premise. Aristotle and some of his successors accepted "existential import" and that $\exists x\, P(x)$ was a logical consequence of $\forall x\,(P(x) \Rightarrow Q(x))$. Modern formal logic denies this but, in natural language, existential import often seems reasonable and *subalternation*, i.e. the deduction of $\exists x\,(P(x) \wedge Q(x))$ from $\forall x\,(P(x) \Rightarrow Q(x))$, might appear as a valid inference rule.

Some medieval authors accepted only some of the pseudo-valid syllogisms, deemed "useful". In Figures 1, 2 and 3, there are four valid syllogisms and two pseudo-valid ones ; in Figure 4, there are three valid syllogisms and three pseudo-valid ones.

Modern view of syllogism theory

Syllogism theory is only a tiny part of predicate logic but arguably the most useful one in practice. The taxonomy of syllogisms is one of the first attempt at systematic classification in science and, existential import aside, it remains acceptable from the mathematical point of view.

Research in formal logic has tended to sieve, among the many restrictions of the theory, those that were essential to retain decidability.

For instance, there is no need to limit the number of predicate and the number of premises ; furthermore, a more liberal use of quantification is possible. In fact, the only unmoveable restriction appears to be that only monadic predicates are allowed. Indeed, monadic predicate logic remains decidable, although the decision procedure in the general case is more complicated than the Venn diagram method.