Predicate logic

Reading:
- Chapter 1 [Rosen]

Additional reading:
- Chapters 8, 9 [Russell + Norvig]
- Chapters 15,16 [Nilsson]

Bibliography:

- A. Deaño “Introducción a la Lógica Formal”,
- E. Paniagua Arís, J. L. Sánchez González, F. Martín Rubio, “Lógica computacional”, Thomson
- Melvin Fitting “First-order logic and automated theorem proving”, Springer-Verlag (New-York 1990)

Higher-order logics

- Propositional logic is not sufficiently powerful to express certain types of knowledge.
- We need a language to
  - » Give a concise description of domains with many objects.
  - » Formulate general statements about all the objects in a given domain, over the relationships among them, over the existence of objects that have certain properties or are related in some form, etc.
- Types of logic
  - » **Propositional logic** or **first order logic**

    Constant objects, logical connectives
  - » **Predicate logic** or **first order logic (FOL)**

    ++ functions, predicates, quantifiers over objects

    e.g. $(\forall x)\ Human(x) \Rightarrow Mortal(x)$
  - » **Second order logic**:

    ++ functions, predicates, quantifiers over objects and predicates

    e.g. $(\exists P)\ [P(A) \wedge P(B)]$

    $(\forall x)(\forall y)\ [(x=y) \Leftrightarrow (\forall P)(P(x) \Leftrightarrow P(y))]$

    [***Leibniz' principle***, 1666]
  - » **Higher order logics …**
  - » **Theory of types.**

First order logic

Example: Family + friends

- **Ontology**
  - » **Logical connectives, variables, quantifiers**
  - » **Constants**

    Objects constants: Peter, Pablo, Mary

    Functions: motherOf^1, fatherOf^1, bestFriendOf^1

    Predicates: Married^1, Happy^1, Mother^2, Father^2, Aunt^2, Uncle^2, Daughter^2, Son^2, Sister^2, Brother^2,…
- **Definition**: "Someone's uncle is the brother of the father or the mother of the person in question"

  $\forall x\ \forall y\ [Uncle(x,y) \Leftrightarrow \exists z\ (Brother(x,z) \wedge (Father(z,y) \vee Mother(z,y)))]$
- **General assertions**:

  "All of Mary's sons are married but are unhappy"

  $\forall x\ [Son(x,Mary) \Rightarrow (Married(x) \wedge \neg Happy(x))]$
- **Existential assertions**: "Some of the daughters of Peter's best friend are mothers"

$\exists x \exists y [Daughter(x,bestFriendOf(Peter)) \wedge Mother(x,y)]$

Language, I

- **Constants**:

  » **Symbolic Objects** (in general, capitalized)

  `e.g. P,Q, John, DeathValley (symbolic)`

  » **Functions** (in general, lower case)

  Input arguments: list of terms between parentheses
  Evaluates to: a term

  E.g. $\texttt{fatherOf}^1$, $\texttt{distanceBetween}^2$

  » **Relations or predicates** (in gral. capitalized)

  Input arguments : list of terms between parentheses.
  Evaluates to: a truth value ("True" or "False").

  E.g. $\texttt{Father}^2$, $\texttt{White}^1$, $\texttt{Triunvirate}^3$

  Unary relations are also known as properties

  Note:

  » The superscript indicates the arity (the number of arguments) of the function or predicate.

  » Symbolic objects can be considered as functions of arity 0.

Language, II

- **Punctuation signs**
  - » **Comma: ,**
  - » **Parentheses:** ( ) [ ] { }

- **Logical connectives**:
  ¬, ∧, ∨, ⇒, ⇔ (in order of precedence)

- **Variables** (in general, lower case)

  The set of values a variable can take is its domain
  e.g. x (reals), n (integers), p (people), etc.

  E.g. x, y, p, q,…, p1, p2,…

- **Quantifiers**
  - » **Universal quantifier (∀)**
  - » **Existential quantifier (∃)**

  Used to quantify variables.

$$(\forall x)\ \underbrace{\{P(x) \Rightarrow (\exists y)\underbrace{[R(x,y) \Rightarrow S(f(x))]}_{\text{Scope of } y}\}}_{\text{Scope of } x}$$

Term, atom, literal, clause, CNF

- **A term can be**
  - » An **object (constant)**

    `E.g.   Peter, P, P1,…`
  - » A **variable**

    `E.g.   x,y`
  - » A **function** of arity n, followed by **n terms** separated by commas and between parentheses.

    `E.g.   fatherOf(x), distanceBetween(A,B)`
- **An atomic formula (atom) is** a **relation** of arity n, followed by **n terms** separated by commas and between parentheses.

  `E.g.   Father(x,Peter)`
- **A literal** can be:
  - » A **positive literal**: An atomic formula

    `E.g.   Brother(x,Peter)`
  - » A **negative literal**: The negation of an atomic formula

    `E.g.   ¬ Father(John,fatherOf(Peter))`
- **A clause is a disjunction of literals**
- **An WFF in Conjunctive Normal Form (CNF) is a conjunction of clauses.**

Well-formed formulas

- A **WFF** is built using atoms and connectives in the same way as in propositional logic.
- If `w` is a WFF and `x` is a variable, the following expressions are also WFF's
  - » $(\forall x)\ w$
  - » $(\exists x)\ w$

  In general (but not always), variable `x` appears in `w`.

  To indicate that `x` appears in `w`, one can write `w(x)`
- **Closed WFF**: A WFF all of whose variables are quantified.
  - » $(\forall x)\ [P(x) \Rightarrow R(x)]$
  - » $(\exists x)\ [P(x) \Rightarrow (\exists y)(R(x,y) \Rightarrow S(f(x)))]$

Note: The **order** in which $\forall, \exists$ appear in the WFF is **important.**

$x, y \in \{people\}$

$Father(x,y)$ "x is the father of y" (binary relation)

$(\forall x)(\forall y)Father(x,y)$ "Everyone is everyone else's father"

$(\exists x)(\exists y)Father(x,y)$ "There is someone who has a father"

$(\forall x)(\exists y)Father(x,y)$ "Everyone is the father of someone"

$(\exists y)(\forall x)Father(x,y)$ "There is someone who has everyone as a father"

$(\exists x)(\forall y)Father(x,y)$ "There is someone who is the father of everyone"

$(\forall y)(\exists x)Father(x,y)$ "Everyone has a father"

Interpretation and semantics

- **Constants**
  - » **The constants that are symbolic objects** correspond to entities in **the real world**.
  - » **Predicates of arity 1** correspond to properties of these **objects**.
  - » **Predicates of arity n > 1** express **relationships among objects**.

- **Variables**
  - » The **domain** of a variable is the range of values among the symbolic objects that the variable can take.
  - » **Assignment**: Operation as the result of which a variable in a WFF is replace by a particular value in its domain.
- **Quantifiers**
  - » **Universal quantifier**: $(\forall x)\,w(x)$ has the truth value "`True`" iff $w(x)$ has the truth value "`True`" for all possible assignments of the variable x.
  - » **Existential quantifier**: $(\exists x)\,w(x)$ has the truth value "`True`" iff $w(x)$ has the truth value "`True`" for at least one of the possible assignments of x.

Equivalence rules and inference rules

- **Equivalence rules**
  - » **Equivalence rules of propositional logic**.
  - » **Renaming variables**. The new symbol has to be different from the symbols previously used for the other variables in the WFF.
    - $(\forall x)\ w(x) \equiv (\forall y)\ w(y)$
    - $(\exists x)\ w(x) \equiv (\exists y)\ w(y)$
  - » $\neg(\forall x)\ w(x) \equiv (\exists x)\ \neg w(x)$
  - » $\neg(\exists x)\ w(x) \equiv (\forall x)\ \neg w(x)$

- **Inference rules**
  - » **Inference rules from propositional logic**.
  - » **Instantiation of the universal** (IU) [Correct]

    $(\forall x)\ w(x) \vdash_{IU} w(A)$, where A is a value in the domain of x.
  - » **Generalization of the existential** (GE) [Correct]

    $w(A) \vdash_{GE} (\exists x)\ w(x)$, where x is the symbol of a variable whose domain includes A.

Skolemization

An **existential quantifier** can be eliminated from a WFF by **replacing in** each of the places in which the existentially quantified variable appears for

» A **Skolem object (constant),** if there are no universally quantified variables whose scope includes the scope of the variable to eliminate

E.g. $(\exists x)\ w(x)$

Skolem form: $w(SK)$

SK is an object whose identity we ignore, of which we know that it exists.

» A **Skolem function**, whose arguments are the universally quantified variables whose scope includes the scope of the variable to eliminate

Example:

"Everyone has a height"

$(\forall p)[(\exists h)\ Height(p,h)]$

domain of p: People.

domain of h: Positive reals.

Skolem form: $(\forall p)\ Height(p,h(p))$

$h(p)$ is a Skolem function (unknown, but we know that it exists): It takes as its argument a reference to a person and returns his or her height.

Examples of Skolemization

$x, y \in \{people\}$

$\text{Father}(x,y)$ "x is the father of y" (binary relation)

$(\forall x)(\forall y)\text{Father}(x,y)$ "Everyone is everyone else's father"

$(\exists x)(\exists y)\text{Father}(x,y) \vdash_{SK} \text{Father}(SK_1,SK_2)$
"There is someone who has a father"

$(\forall x)(\exists y)\text{Father}(x,y) \vdash_{SK} (\forall x)\ \text{Father}(x,h(x))$
"Everyone is the father of someone"

$(\exists y)(\forall x)\ \text{Father}(x,y) \vdash_{SK} (\forall x)\ \text{Father}(x,SK_3)$
"There is someone who has everyone as a father"

$(\exists x)(\forall y)\text{Father}(x,y) \vdash_{SK} (\forall y)\ \text{Father}(SK_4,y)$
"There is someone who is the father of everyone"

$(\forall y)(\exists x)\text{Father}(x,y) \vdash_{SK} (\forall y)\ \text{Father}(p(y),y)$
"Everyone has a father"

Metattheorems for Skolem forms

- Metatheorem SK1: The Skolem form of a WFF is NOT equivalent to the original WFF

  $w(SK) \models (\exists x)\ w(x)$ BUT $(\exists x)\ w(x) \not\models w(SK)$

	w(A)	**w(B)**		**(∃x) w(x)** **x∈{A,B}**		**w(SK)**	
						Si SK fuera A	Si SK fuera B
I_2	V	V		V		V	V
I_2	V	F		V		V	F
I_3	F	V		V		F	V
I_4	F	F		F		F	F

E.g. $P(A) \vee P(B) \models (\exists x)\ P(x)$

BUT $P(A) \vee P(B) \not\models P(SK)$

- Metatheorem SK2 (Loveland, 1978):

  The Skolem form of a set of WFF's is satisfiable exactly in those cases in which the original set of WFF's is satisfiable.

  » A set of WFF's is satisfiable if its Skolem form is satisfiable.

  » A set of WFF's is unsatisfiable if its Skolem form is insatisfiable.

CNF in first order logic

1. **Eliminate the implications and double implications** $\Leftrightarrow$, $\Rightarrow$
2. **Reduce the scope of negation** $\neg$
   - De Morgan's laws

     $\neg(w1 \lor w2) \equiv \neg w1 \land \neg w2$

     $\neg(w1 \land w2) \equiv \neg w1 \lor \neg w2$
   - Elimination of double negation $(\neg\neg w \equiv w)$
   - Combination of $\neg$ with the quantifiers

     $\neg(\forall x)\ w(x) \equiv (\exists x)\ \neg w(x)$

     $\neg(\exists x)\ w(x) \equiv (\forall x)\ \neg w(x)$
3. **Standardize the variables**: Rename variables, so that different variables are denoted by different symbols

   $[(\forall x)\ [P(x) \Rightarrow R(x)]] \lor [(\exists x)\ P(x)]$

   $\equiv [(\forall x)\ [P(x) \Rightarrow R(x)]] \lor [(\exists y)\ P(y)]$
4. **Skolemization**: Eliminate existential quantifiers by replacing Skolem objects or Skolem functions for the corresponding variables.
5. **Transform into prenex form** moving to the beginning of the WFF all universal quantifiers.

   WFF in **prenex form** = **Prefix** (list of quantifiers)

   \+ **Matrix** (formula without quantifiers)
6. **Eliminate the quantifiers at the beginning of the WFF**.
7. **Use distributive laws and other equivalence laws** from propositional logic to transform the matrix to **CNF**.

Example 1: Transformation to CNF

$[(\forall x)\ Q(x)] \Rightarrow$
$(\forall x)(\forall y)\ [(\exists z)\ [P(x,y,z) \Rightarrow (\forall u)\ R(x,y,u,z)]$

1. Eliminate the implications $\Leftrightarrow, \Rightarrow$

$\neg[(\forall x)\ Q(x)] \vee$
$(\forall x)(\forall y)\ [(\exists z)\ [\neg P(x,y,z) \vee (\forall u)\ R(x,y,u,z)]$

2. Reduce the scope of negation

$[(\exists x)\ \neg Q(x)] \vee$
$(\forall x)(\forall y)\ [(\exists z)\ [\neg P(x,y,z) \vee (\forall u)\ R(x,y,u,z)]$

3. Standardize the variables

$[(\exists w)\ \neg Q(w)] \vee$
$(\forall x)(\forall y)\ [(\exists z)\ [\neg P(x,y,z) \vee (\forall u)\ R(x,y,u,z)]$

4. Skolemization :

$\neg Q(A) \vee (\forall x,y)[\neg P(x,y,f(x,y)) \vee (\forall u) R(x,y,u,f(x,y))]$

5. Transform into prenex form:

$(\forall x,y,u)[\neg Q(A) \vee \neg P(x,y,f(x,y)) \vee R(x,y,u,f(x,y))]$

6. Eliminate the universal quantifiers:

$\neg Q(A) \vee \neg P(x,y,f(x,y)) \vee R(x,y,u,f(x,y))$

7. Transform to CNF: The WFF is already in CNF.

Example 2: Transformation to CNF

"Everybody who loves all animals is loved by someone"

$$(\forall x)[(\forall y)\{Animal(y) \Rightarrow Loves(x,y)\} \Rightarrow (\exists y)\ Loves(y,x)]$$

1. Eliminate the implications $\Leftrightarrow$, $\Rightarrow$

$$(\forall x)[\neg(\forall y)\{\neg Animal(y) \vee Loves(x,y)\} \vee (\exists y)\ Loves(y,x)]$$

2. Reduce the scope of negation

$$(\forall x)[(\exists y)\{Animal(y) \wedge \neg Loves(x,y)\} \vee (\exists y)\ Loves(y,x)]$$

3. Standardize the variables

$$(\forall x)[(\exists y)\{Animal(y) \wedge \neg Loves(x,y)\} \vee (\exists z) Loves(z,x)]$$

4. Skolemization :

$$(\forall x)[\{Animal(f(x)) \wedge \neg Loves(x,f(x))\} \vee Loves(g(x),x)]$$

5. Transform into prenex form: WFF already in prenex form

6. Eliminate the universal quantifiers

$$\{Animal(f(x)) \wedge \neg Loves(x,f(x))\} \vee Loves(g(x),x)$$

7. Transform to CNF using distributive laws and other equivalence rules:

$$\{Animal(f(x)) \vee Loves(g(x),x)\} \wedge \{\neg Loves(x,f(x))\} \vee Loves(g(x),x)\}$$

Resolution + refutation in FOL

Consider the set of WFF's Δ and the WFF w

Is w logical consequence of Δ ?

$\Delta \models w$?

1. Include the negation of the goal: $\alpha = \{\Delta \wedge \neg w\}$
2. Transform to CNF
3. Apply resolution (possibly using instantiation of variables)
   (i) If it is possible to derive the empty clause using resolution (α is UNSAT) then $\Delta \models w$
   (ii) If it is possible to derive the empty clause using resolution (α is SAT), then $\Delta \not\models w$

» If $\Delta \models w$, the algorithm halts and output “YES”.
» If $\Delta \not\models w$, the algorithm could
   – halt and output “NO” (correct)
   – not halt.

Resolution is **correct**, **complete in proof by refutation,** but **semidecidible**.

Tricks

When we translate a sentence from a natural language to WFF's in predicate logic, one normally encounters only the following combinations

$$(\forall x)\ [w1(x) \Rightarrow w2(x)]$$

$$(\exists x)\ [w1(x) \wedge w2(x)]$$

E.g. "Everyone at UAM is intelligent"

$$(\forall x)\ [At(x,UAM) \Rightarrow Intelligent(x)]$$

"There are people at UAM who are intelligent"

$$(\exists x)\ [At(x,UAM) \wedge Intelligent(x)]$$

COMMON ERRORS

$$(\exists x)\ [At(x,UAM) \Rightarrow Intelligent(x)]$$

[TOO GENERAL]

"There is someone who is either intelligent or is not at UAM"

$$(\forall x)\ [At(x,UAM) \wedge Intelligent(x)]$$

[TOO SPECIFIC]

"Everyone is at UAM and is intelligent"

It would be better to restrict the domain of x to intelligent people who are at UAM.

Did curiosity kill the cat?

- Everyone who loves all animals is loved by someone.
- Anyone who kills an animal is loved by no one.
- Jack loves all animals.
- Jack or Curiosity killed the cat, who is named Tuna.
- Did Curiosity kill the cat?

Did curiosity kill the cat?

- Everyone who loves all animals is loved by someone.

  $(\forall x)[(\forall y)[Animal(y) \Rightarrow Loves(x,y)] \Rightarrow (\exists z)\ Loves(z,x)]$

- Anyone who kills an animal is loved by no one.

  $(\forall x)[(\exists y)(Animal(y) \wedge Kills(x,y)) \Rightarrow (\forall z)\ \neg Loves(z,x)]$

- Jack loves all animals.

  $(\forall x)\ [Animal(x) \Rightarrow Loves(Jack,x)]$

- Jack or Curiosity killed the cat, who is named Tuna.

  $Kills(Jack,Tuna) \vee Kills(Curiosity,Tuna)$

  $Cat(Tuna)$

- A cat is an animal

  $(\forall x)\ Cat(x) \Rightarrow Animal(x)$

- Did Curiosity kill the cat?

  Kills(Curiosity,Tuna) ???

Did curiosity kill the cat?

A. Animal(F(x)) ∨ Loves(G(x),x)

B. ¬Loves(u,F(u)) ∨ Loves(G(u),u)

C. ¬Animal(y) ∨ ¬Kills(x,y) ∨ ¬Loves(z,x)

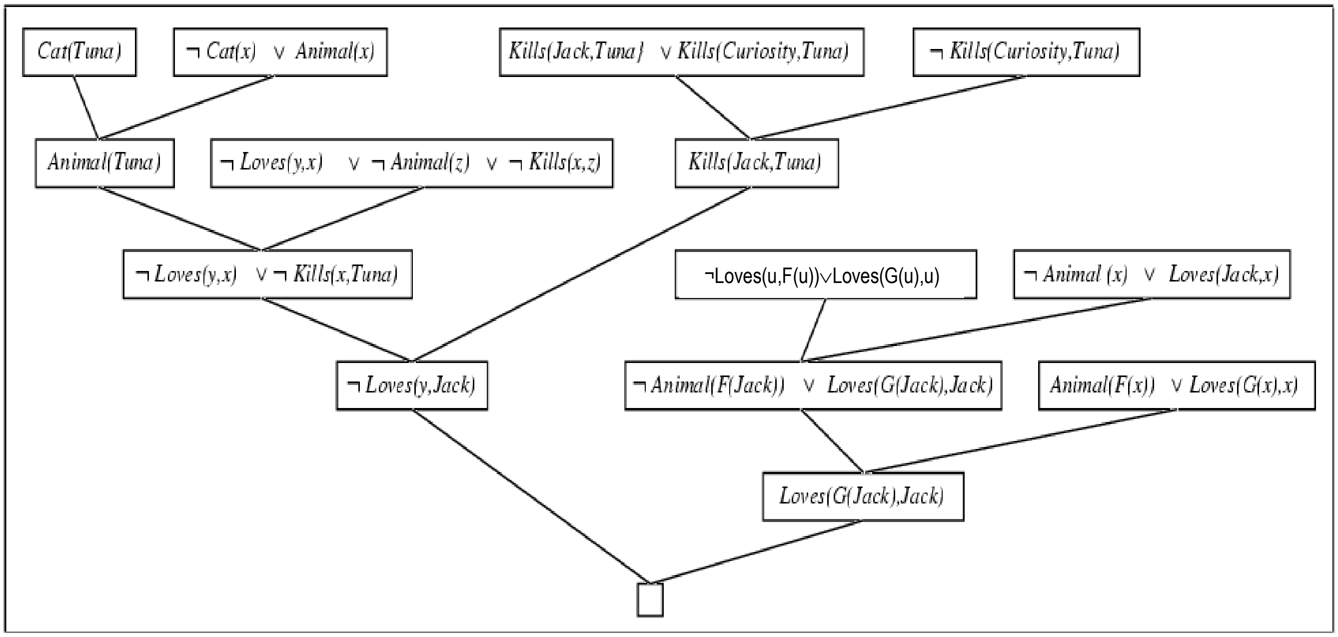
D. ¬Animal(x) ∨ Loves(Jack,x)

E. Kills(Jack,Tuna) ∨ Kills(Curiosity,Tuna)

F. Cat(Tuna)

G. ¬Cat(x) ∨ Animal(x)

H. ¬Kills(Curiosity,Tuna)



Note: In this example variables in different clauses are different, even if we use the same name for them.

Ontology for sets

- **Ontology** for **set theory**

  Vocabulary of objects, predicates and functions necessary to characterize sets

  » Object constants:
    - The empty set: `{}`
    - Elements of a set: `A, B, C, ...`

  » Predicates: `Set`^{1}, `BelongsTo`^{2} ($\in$), `Subset`^{2} ($\subset$), `Equal`^{2} ($=$)

  » Functions: `insert`^{2} ($\{\,|\,\}$), `union`^{2} ($\cup$), `intersection`^{2} ($\cap$)

- **Axioms** for set theory

1. $\forall s [Set(s) \Leftrightarrow$

   $(s=\{\}) \vee (\exists x, s')[Set(s') \wedge (s=\{x|s'\})]]$

   [From this point on, the domain of variables starting with s is the set of sets]

2. $\neg(\exists x, s)[\{x|s\}=\{\}]$

3 $(\forall x, s)[x \in s \Leftrightarrow s=\{x|s\}]$

4. $(\forall x, s)[x \in s \Leftrightarrow (\exists y, s')[s=\{y|s'\} \wedge (x=y \vee x \in s')]]$

5. $(\forall s_1, s_2)[s_1 \subset s_2 \Leftrightarrow (\forall x)[x \in s_1 \Rightarrow x \in s_2]]$

6. $(\forall s_1, s_2)[s_1 = s_2 \Leftrightarrow (s_1 \subset s_2 \wedge s_2 \subset s_1)]$

7. $(\forall x, s_1, s_2)[x \in (s_1 \cap s_2) \Leftrightarrow (x \in s_1 \wedge x \in s_2)]$

8. $(\forall x, s_1, s_2)[x \in (s_1 \cup s_2) \Leftrightarrow (x \in s_1 \vee x \in s_2)]$

Natural numbers

[http://mathworld.wolfram.com/NaturalNumber.html]

The term "natural number" refers either to a member of the set of positive integers 1, 2, 3, ... (Sloane's A000027) or to the set of nonnegative integers 0, 1, 2, 3, ... (Sloane's A001477; e.g., Bourbaki 1968, Halmos 1974).

Regrettably, there seems to be no general agreement about whether to include 0 in the set of natural numbers.

In fact, Ribenboim (1996) states "Let P be a set of natural numbers; whenever convenient, it may be assumed that $0 \in P$."

Note:

In logic, set theory and computer science the most common convention (because it is the one that is more convenient for the developments that are made) is that natural numbers include the element zero.

Mathematicians that work in number theory prefer to exclude zero from the natural numbers.

Natural numbers

- Ontology for natural numbers
  - » A constant object: `0`
  - » Predicates
    - – Predicate to determine whether an object is a natural number: $\mathtt{NatNum}^1$
    - – Equality: $\mathtt{Equal}^2$ (=)
  - » Functions
    - – Successor: $\mathtt{suc}^1$
    - – Sum: $\mathtt{sum}^2$ (+)
- Peano's axioms

$\mathtt{NatNum(0)}$

$(\forall n)[\mathtt{NatNum}(n) \Rightarrow \mathtt{NatNum}(\mathtt{suc}(n))]$

[From now on, the domain of all variables is the set of natural numbers, including zero]

$(\forall n)[0 \neq \mathtt{suc}(n)]$

$(\forall m,n)[m \neq n \Rightarrow \mathtt{suc}(m) \neq \mathtt{suc}(n)]$

$(\forall n)\ [\mathtt{sum}(n,0)=n]$

$(\forall m,n)[\mathtt{sum}(\mathtt{suc}(m),n)=\mathtt{suc}(\mathtt{sum}(m,n))]$

Note: The **induction principle** can be formulated only in **second order logic**, in which first order relations and functions can be used as arguments of statements.

(e.g. One can write a second-order predicate that specifies that a first order relation is transitive)