Logic for Computer Science. Knowledge Representation and Reasoning.

Lecture Notes

for

Computer Science Students
Faculty EAIiIB-IEiT AGH

Antoni Ligęza

Other support material:

http://home.agh.edu.pl/~ligeza

https://ai.ia.agh.edu.pl/pl:dydaktyka:logic:start#logic_for_computer_science2020

Limitations of Propositional Calculus: Expressive Power

Consider the following examples in Natural Language (NL):

Adam is Bob's brother. Bob is Adam's brother.

If Adam is Bob's brother then Bob is Adam's brother.

If X is Y's brother then Y is X's brother.

If X is father of Y, and Y is father of Z, then X is grandfather of Z.

If block A is on block B, and block B is on block C, then A is above C.

Everything has it's price.

There is no free lunch. ("No free lunch" (NFL) theorem...)

Everybody loves someone. Everybody is loved by someone.

If everybody loves someone then anyone is loved by somebody.

If X is connected to Y, and Y is connected to Z, then X is connected to Z.

Every student of AGH is smart. Jan is a student of AGH. Jan is smart.

There exists a set of all sets

The barber shaves anyone that does not shave himself.

Predicate Logic – New Opening: High Expressive Power. Objects, Variables, Relations, Constructions and Operations

New components:

- Constants — representation of individual (atomic) objects,
- Variables — symbols of unknown/universal objects
- Predicate symbols — names of relations among objects,
- Quantifiers — there exists/existential quantifier; for all/universal quantifier,
- Terms — objects of complex structure; connected atomic objects.

Logical connectives:

- negation,
- conjunction, disjunction,
- implication, equivalence.

Operations:

- Abstraction — from individual objects properties to universal properties,
- Specification — from universal properties to specific ones,
- Properties of Relations — e.g. symmetry, transitivity,
- Specification of Constraints — relations + variables + quantification
- Complex Logical formulas — with use of logical connectives,
- Complex Logical Inference — general rules, universal laws.

Basic limitation of Propositional Logic: no Universal Rules

Consider the following classical example:

Socrates is a man.

Every man is mortal.

Socrates is mortal.

```
man(plato).
man(socrates).
mortal(X):- man(X).
```

```
mother(eva,nadjed).
father(john,tom).
father(john,ted).
father(john,eva).
father(ted,jimmy).
man(tom).
man(ted).
woman(eva).

parent(X,Y):- father(X,Y).
parent(X,Y):- mother(X,Y).

brother(B,X):-
   parent(P,B),
   parent(P,X),
   man(B),
   B \= X.
```

```
uncle(U,X):-
   parent(P,X),
   brother(U,P).
```

Alphabet and Notation

Definition 1 *A relation R is any subset of Cartesian Product of some given sets:*

$$R \subseteq X_1 \times X_2 \times \ldots \times X_n$$

Relation is a set. Elements of any relations are tuples of the form $(x_1, x_2, \ldots, x_n)$.

Notation: $R(x_1, x_2, \ldots, x_n)$ is read: R holds for arguments $x_1, x_2, \ldots, x_n$.

Let there be given the following, pairwise disjoint four sets of symbols:

- C — a set of constant symbols (or constants, for short),
- V — a set of variable symbols (or variables, for short),
- F — a set of function (term) symbols,
- P — a set of relation (predicate) symbols.

Definition 2 **Terms:**

- **if** c **is a constant,** $c \in C$**, then** $c \in TER$**;**
- **if** X **is a variable,** $X \in V$**, then** $X \in TER$**;**
- **if** f **is an** n**-ary function symbol,** $f \in F$**, and** $t_1, t_2, \ldots, t_n$ **are terms, then** $f(t_1, t_2, \ldots, t_n) \in TER$**;**
- **all the elements of** TER **are generated only by applying the above rules.**

The number n is referred to as the arity of f. Notation:

$$f/n$$

Examples of Terms

Assume that $a, b, c \in C$, $X, Y, Z \in V$, $f, g \in F$, and arity of f and g is 1 and 2, respectively.

Then, all the following expressions are examples of terms:

- a, b, c;
- X, Y, Z;
- $f(a), f(b), f(c), f(X), f(Y), f(Z)$;
  $g(a, b), g(a, X), g(X, a), g(X, Y)$;
  $f(g(a, b)), g(X, f(X)), g(f(a), g(X, f(Z)))$.

The set of terms (even for one constant and functional symbol) is:

- infinite,
- countable.

Each term can be represented as a tree.

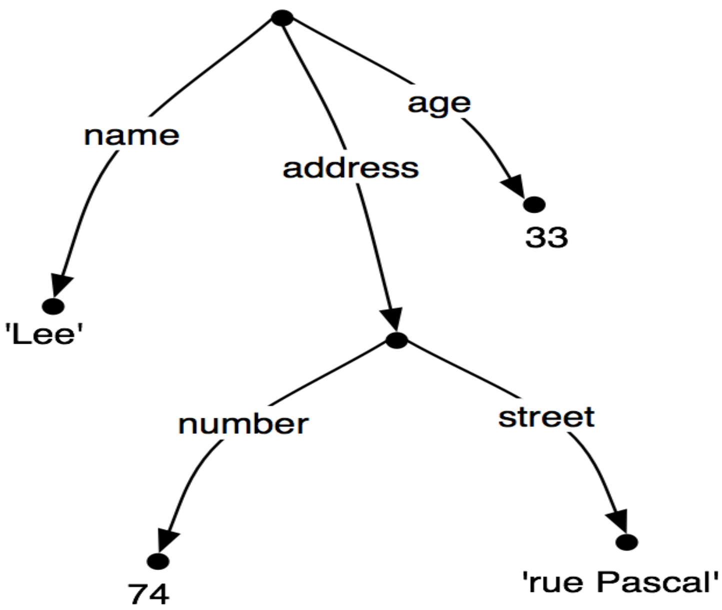

Figure 1: Visualization of the tree-like structure of a term

Example applications of Terms

Prolog:

```
book (book_title,
       author(first_name,last_name),
       publisher_name,
       year_of_publication
      )
```

XML:

```
<book>
      <book_title> Learning XML </book_title>
      <author>
            <first_name> Erik </first_name>
            <last_name>  Ray  </last_name>
      </author>
      <publisher_name> O'Reilly & Associates, Inc. </publisher
      <year_of_publication> 2003 </year_of_publication>
</book>
```

YAML:

```
book:
   title:      book_title
   author:     author_name
   publisher:  publisher_name
   year:       year_of_publication
```

Applications of Terms

LaTeX

$$\frac{\frac{x}{y}}{\sqrt{1+\frac{x}{y}}},$$

```
\frac{
      \frac{x}{y}
     }
     {
      \sqrt{1+\frac{x}{y}}
     }
```

Lists:

```
[red, green, blue, yellow]
[red|green, blue, yellow]
list(red,list(green,list(blue,list(yellow,nil))))
```

Trees:

```
tree (
      tree (left_left,  left_right),
      tree (right_left, right_right)
     )
```

Other: records, complex structures, natural numbers (Peano arithmetics),...

Formulas

Definition 3 *The set of Atomic Formulas* $ATOM$ *is defined as one satisfying the following conditions:*

- *if* p *is an* n*-ary predicate symbol,* $p \in P$*, and* $t_1, t_2, \ldots, t_n$ *are terms, then* $p(t_1, t_2, \ldots, t_n) \in ATOM$*;*
- *all the elements of* $ATOM$ *are generated by applying the above rule.*

The elements of $ATOM$ *are called* atomic formulae *or* atoms*, for short.*

Examples of atomic formulas:

- $p(a), p(b), q(a, a), q(a, c)$;
- $p(X), p(Y), q(X, X), q(X, Z)$;
- $p(f(a)), p(f(X)), q(f(g(a, b)), g(X, f(X))), q(g(f(a), g(X, f(Z))), a)$.

Terms vs. Atomic Formulas — what is the difference?

Definition 4 *Formulas:* FOR

- $ATOM \subseteq \mathrm{FOR}$*;*
- *if* Φ *is a formula,* $\Phi \in \mathrm{FOR}$*, then* $\neg(\Phi) \in \mathrm{FOR}$*;*
- *if* Φ *and* Ψ *are formulae,* $\Phi, \Psi \in \mathrm{FOR}$*, then* $(\Phi \wedge \Psi)$*,* $(\Phi \vee \Psi)$*,* $(\Phi \Rightarrow \Psi)$*,* $(\Phi \Leftrightarrow \Psi) \in \mathrm{FOR}$*;*
- *if* $\Phi \in \mathrm{FOR}$*,* X *denotes a variable, then* $\forall X(\Phi) \in \mathrm{FOR}$ *and* $\exists X(\Phi) \in \mathrm{FOR}$*;*
- *all the elements of* FOR *must be generated by applying the above rules.*

General note:

Although we can define formulas such as $\forall X : p$ or $\exists X : q$ it seems no to be rational; it is reasonable to qunatify over variables occurring in a formula, e.g. $\forall X \exists Y : p(X, Y)$

Notation:

- restricted general quantifier: $\forall X \in D_X$, $\forall_{X \in D_X}$,
- there exists exactly one element: $\exists ! X$,
- $\forall$ — general qunatifier (generalized conjunction); also: $\bigwedge$,
- $\exists$ — existential quantifier (generalized disjunction); also: $\bigvee$,

Generalization of conjunction:

$$\forall X : p(X) \quad \stackrel{?}{\equiv} \quad p(a) \wedge p(b) \wedge p(c) \wedge \ldots$$

Generalization of disjunction:

$$\exists X : p(X) \quad \stackrel{?}{\equiv} \quad p(a) \vee p(b) \vee p(c) \vee \ldots$$

Formulas (terms) with no variables: ground formulas/ground instances.

Free variable: X is a free variable in $p(X)$.

Bound variable: X is a bound variable in $\forall X : p(X)$.

We can construct formulas with free and bound variables...

The Roles of Variables. Free and Bound Variables

The role of variables is three-fold:

- names/references to unknown objects — they define boundings to quantifiers; free variables are not in use,
- placeholders — they keep place for (unknown) objects; arity!
- coreference constraints — they define coreferrence constraints (binding of occurrences; data carriers).

Occurrence of a variable in a formula can be:

- bound — within the scope of a quantifier,
- free — out of the scope of any quantifier,

A variable is bound in a formula iff all its occurrences are bound.

Definition 5 *Free variables in a formula:* $FV()$

- *if* $t \in V$ *then* $FV(t) = \{t\}$;
- *if* $t \in C$ *then* $FV(t) = \emptyset$;
- *if* $t = f(t_1, t_2, \ldots, t_n) \in \mathrm{TER}$ *then* $FV(t) = FV(t_1) \cup FV(t_2) \cup \ldots \cup FV(t_n)$;
- *if* $q = p(t_1, t_2, \ldots, t_n) \in ATOM$ *then* $FV(q) = FV(t_1) \cup FV(t_2) \cup \ldots \cup FV(t_n)$;
- $FV(\neg\Phi) = FV(\Phi)$;
- $FV(\Phi \diamond \Psi) = FV(\Phi) \cup FV(\Psi)$ *for any* $\diamond \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$;
- $FV(\nabla X(\Phi)) = FV(\Phi) \setminus \{X\}$ *for* $\nabla \in \{\forall, \exists\}$.

The Universe, Interpretation and Variable Assignment

In order to define the sematics we need:

- D — an non-empty set, the Universe,
- I — an Interpretation — a mapping of constants, function symbols, and predicate symbols into elements of D, functions over D and relations over D,
- v — Variable Assignment — assignment of elements of D (ground terms) to free variables.

Definition 6 *The Variable Assignment v:*

$$v\colon V \to D$$

may be defined over the elements of the universe (for simplicity)

Definition 7 *An interpretation I:*

- *for any constant $c \in C$, $I(c) \in D$;*
- *for any free occurrence of variable $X \in V$, $I(X) = v(X)$, where $v(X) \in D$;*
- *for any function symbol $f \in F$ of arity n, $I(f)$ is a function of the type*

$$I(f)\colon D^n \to D;$$

- *for any predicate symbol $p \in P$ of arity n, $I(p)$ is a relation such that*

$$I(p) \subseteq D^n;$$

- *for any term $t \in \mathrm{TER}$, such that $t = f(t_1, t_2, \ldots, t_n)$,*

$$I(t) = I(f)(I(t_1), I(t_2), \ldots, I(t_n)).$$

Semantics of the Predicate Calculus

Semantics = assignment of the meaning in the considered World; let:

- D — be the Universe, ,
- I — be the Interpretation, and
- v — be the Variable Assignment (or, for simplicity, we consider only closed formulas).

Definition 8 **Formulas satisfaction**

1. $\models_{I,v} p(t_1, t_2, \ldots, t_n)$ **iff (if and only if)** $(I(t_1), I(t_2), \ldots, I(t_n)) \in I(p)$ **(recall that** $I(X) = v(X)$ **for any free variable** $X \in VAR$**;**
2. $\models_{I,v} \neg\Phi$ **iff** $\not\models_{I,v} \Phi$**;**
3. $\models_{I,v} \Phi \wedge \Psi$ **iff both** $\models_{I,v} \Phi$ **and** $\models_{I,v} \Psi$**;**
4. $\models_{I,v} \Phi \vee \Psi$ **iff** $\models_{I,v} \Phi$ **or** $\models_{I,v} \Psi$**;**
5. $\models_{I,v} \Phi \Rightarrow \Psi$ **iff** $\not\models_{I,v} \Phi$ **or** $\models_{I,v} \Psi$**;**
6. $\models_{I,v} \Phi \Leftrightarrow \Psi$ **iff** $\models_{I,v} \Phi$ **and** $\models_{I,v} \Psi$**, or,** $\not\models_{I,v} \Phi$ **and** $\not\models_{I,v} \Psi$**;**
7. $\models_{I,v} \forall X \Phi$ **iff for any** $d \in D$ **and any variable assignment** u **such that** $u(X) = d$ **and** $u(Y) = v(Y)$ **for any** $Y \neq X$**, there is** $\models_{I,u} \Phi$**;**
8. $\models_{I,v} \exists X \Phi$ **iff there exists** $d \in D$ **such that for variable assignment** u **defined as** $u(X) = d$ **and** $u(Y) = v(Y)$ **for any** $Y \neq X$**, there is** $\models_{I,u} \Phi$**.**

Important Comments

For simplicity we consider closed formulas — no free variables are allowed.

If in a formula there are free variables, should they be considered universally quantified or existentially quantified? Or a combination of that? Example: $p(X) \vee \neg p(Y)$ is a tautology or not?

For convenience and for clarity, all the occurrences of variables are renamed in a consequent way so that no conflicts of variable names exist.

Definition 9 **Logical Consequence** *(case of closed formulas):*

A formula H is a logical consequence of set of formulas $\Delta_1, \Delta_2, \ldots, \Delta_n$ if and only if for any interpretation I (and universe D) satisfying $\Delta_1 \wedge \Delta_2 \wedge \ldots \wedge \Delta_n$, H is also satisfied under interpretation I (and universe D).

Example: How many possible interpretations can be defined for the following formulas:

- p,
- $p \wedge q$, $p \vee q$, $p \Rightarrow q$,
- $p(a)$,
- $p(f(a)))$,
- $\forall X: p(X)$,
- $\exists X: p(X)$.

The Herbrand Universe

Definition 10 *Herbranda Universe:*

Let $H_0 = C(\Delta)$*, i.e.* H_0 *contains all the constants occurring in some set of formulas* Δ *(if* $C(\Delta) = \emptyset$ *then one defines* H_0 *in such a way that it contains a single arbitrary symbol, say* $H_0 = \{c\}$*).*

Now, for $i = 0, 1, 2, \ldots,$ *let* $H_{i+1} = H_i \cup \{f(t_1, t_2, \ldots, t_n) \colon f \in F(\Delta)\ and\ t_1, t_2, \ldots, t_n \in H_i\}$ *(where the arity of* f *is* n*). Then* H_∞ *is called the* Herbrand Universe *of* Δ*.*

Definition 11 *The Herbrand Base:*

Let Δ *be a set of formulas and let* ***H*** *be the Herbrand Universe of* Δ*. A set* $B_H = \{p(h_1, h_2, \ldots, h_n) \colon h_1, h_2, \ldots, h_n \in$ ***H***$, p \in P(\Delta)\}$ *(where the arity of* p *is* n*) is called the* Herbrand base *or the* atom set *of* Δ*.*

Definition 12 *The Herbrand Interpretation:*

Let Δ *be a set of formulas and let* ***H*** *be the Herbrand Universe of* Δ*. Any interpretation* I_H *is called a* Herbrand interpretation *(**H**-interpretation) if the following conditions are satisfied:*

- *for any constant* $c \in$ ***H****,* $I_H(c) = c$*;*
- *for any* n*-ary functional symbol* $f \in F(\Delta)$*, and any* $h_1, h_2, \ldots, h_n \in$ ***H****,*

$$I_H(f) \colon (h_1, h_2, \ldots, h_n) \to f(h_1, h_2, \ldots, h_n).$$

The Herbrand Theorems

Theorem 1 *Herbrand Theorem I: A set of clauses S (a formula in CNF with all variables universally quantified) is unsatisfiable iff it is unsatisfiable under any Herbrand Interpretation.*

Theorem 2 *Herbrand Theorem II: A set of clauses S (a formula in CNF with all variables universally quantified) is unsatisfiable iff there exists finite and unsatisfiable set S' of ground instances of clauses of S.*

Formulas Transformation Rules FOPC

Notation: $\Phi[X]$ — explicit occurrence of variable X in formula Φ.

Basic rules for quantifiers (X does not occur in Ψ):

- $\forall X \Phi[X] \wedge \Psi \equiv \forall X (\Phi[X] \wedge \Psi)$,
- $\forall X \Phi[X] \vee \Psi \equiv \forall X (\Phi[X] \vee \Psi)$,
- $\exists X \Phi[X] \wedge \Psi \equiv \exists X (\Phi[X] \wedge \Psi)$,
- $\exists X \Phi[X] \vee \Psi \equiv \exists X (\Phi[X] \vee \Psi)$.

Generalized De Morgan Rules:

- $\neg(\forall X \Phi[X]) \equiv \exists X (\neg \Phi[X])$,
- $\neg(\exists X \Phi[X]) \equiv \forall X (\neg \Phi[X])$.

Distribution Rules for Quantifiers:

- $\forall X \Phi[X] \wedge \forall X \Psi[X] \equiv \forall X (\Phi[X] \wedge \Psi[X])$,
- $\exists X \Phi[X] \vee \exists X \Psi[X] \equiv \exists X (\Phi[X] \vee \Psi[X])$.

Auxiliary Rules with Renaming of Variables:

- $\forall X \Phi[X] \vee \forall X \Psi[X] \equiv \forall X \Phi[X] \vee \forall Y \Psi[Y] \equiv \forall X \forall Y (\Phi[X] \vee \Psi[Y])$,
- $\exists X \Phi[X] \wedge \exists X \Psi[X] \equiv \exists X \Phi[X] \wedge \exists Y \Psi[Y] \equiv \exists X \exists Y (\Phi[X] \wedge \Psi[Y])$.

Most important equivalent transformations — Analogs to Propositional Calculus

- $\neg\neg\phi \equiv \phi$ — double negation elimination,
- $\phi \wedge \psi \equiv \psi \wedge \phi$ — conjunction alternation,
- $\phi \vee \psi \equiv \psi \vee \phi$ — disjunction alternation,
- $(\phi \wedge \varphi) \wedge \psi \equiv \phi \wedge (\varphi \wedge \psi)$ — commutativity,
- $(\phi \vee \varphi) \vee \psi \equiv \phi \vee (\varphi \vee \psi)$ — commutativity,
- $(\phi \vee \varphi) \wedge \psi \equiv (\phi \wedge \psi) \vee (\varphi \wedge \psi)$ — distributive law,
- $(\phi \wedge \varphi) \vee \psi \equiv (\phi \vee \psi) \wedge (\varphi \vee \psi)$ — distributive law,
- $\phi \wedge \phi \equiv \phi$ — idempotency,
- $\phi \vee \phi \equiv \phi$ — idempotency,
- $\phi \wedge \bot \equiv \bot$, $\phi \wedge \top \equiv \phi$ — identity,
- $\phi \vee \bot \equiv \phi$, $\phi \vee \top \equiv \top$ — identity,
- $\phi \vee \neg\phi \equiv \top$ — *tertium non datur*; excluded middle,
- $\phi \wedge \neg\phi \equiv \bot$ — falsification,
- $\neg(\phi \wedge \psi) \equiv \neg(\phi) \vee \neg(\psi)$ — De Morgan rule,
- $\neg(\phi \vee \psi) \equiv \neg(\phi) \wedge \neg(\psi)$ — De Morgan rule,
- $\phi \Rightarrow \psi \equiv \neg\psi \Rightarrow \neg\phi$ — contraposition,
- $\phi \Rightarrow \psi \equiv \neg\phi \vee \psi$ — implication elimination.

Normal Forms

Definition 13 *Formula* Φ *is in* Prenex Normal Form *iff it is written as*

$$(Q_1X_1)\dots(Q_nX_n)(M),$$

where (Q_kX_k) *(for* $1 \leq k \leq n$*) is* $(\forall X_k)$ *or* $(\exists X_k)$*, and* M *— the so called Matrix — is a quantifier-free formula.*

Transforming to Prenex Normal Form (PNF):

1. Elimination of $\Leftrightarrow$ i $\Rightarrow$ by equivalent transformations,
2. Move all negation signs directly before predicate symbols (De Morgan Rules – also for quantifiers; double negation elimination),
3. Move all quantifiers into the prefix (prenex) using the distributivity rules.

Examples:

Example: transform: $\exists Z \forall X((r(Z) \wedge p(X)) \Rightarrow \exists Y q(X,Y))$ to PNF.

$\exists Z \forall X((r(Z) \wedge p(X)) \Rightarrow \exists Y q(X,Y)) \equiv \exists Z \forall X(\neg(r(Z) \wedge p(X)) \vee \exists Y q(X,Y)) \equiv$
$\exists Z \forall X((\neg r(Z) \vee \neg p(X)) \vee \exists Y q(X,Y)) \equiv \exists Z \forall X \exists Y(\neg r(Z) \vee \neg p(X) \vee q(X,Y))$

Skolem Normal Form and Skolemization

Definition 14 *Skolem Normal Form:*

Formula Φ *is in* Skolem Normal Form *if it is in Prenex Normal Form and:*

- *There are no existential quantifiers in the prenex (prefix),*
- *the matrix* M *in in CNF.*

Transforming to Skolem Normal Form:

1. Transform the formula to PNF,
2. Transform the matrix M to CNF,
3. Sequentially transform the prenex $(Q_1X_1)\ldots(Q_nX_n)$, untill all existential quantifiers are eliminated:
   - if there is in the prenex $Q_rX_r = \exists X_r$ and there are no preceding universal quantifiers, then in the matrix M we replace X_R with an arbitrary new constant c, which does not appear in$M$ and we delete $\exists X_r$ from the prenex,
   - if before $Q_rX_r = \exists X_r$ there occur s universal quantifiers $(\forall X_{j_1})\ldots(\forall X_{j_s})$, where $(1 \leq j_1 \leq j_s \leq r)$, then in the matrix M we replace X_R with a new term of arity s, i.e. $f(X_{j_1},\ldots,X_{j_s})$ (f does not appear in M and we delete $\exists X_r$ from the prenex.

Example:

$\exists Z \forall X \exists Y(\neg r(Z) \vee \neg p(X) \vee q(X,Y)) \equiv \forall X(\neg r(c) \vee \neg p(X) \vee q(X, f(X)))$,
where c is a new constant and f is a functional symbol.

Finally, the so called S-Form (Clausal Form) is: $\{\neg r(c) \vee \neg p(X) \vee q(X, f(X))\}$
Note that all the quantifiers can be omitted, since now all the variables are universally quantified.

- Theorem Proving — Verification of Logical Consequence:

$$\Delta \models H;$$

- Method of Theorem Proving: Automated Inference —- Derivation:

$$\Delta \vdash H;$$

- SAT (checking for models) — satisfiability:

$$\models_I H \quad \text{(if such I exists)};$$

- un-SAT verification — unsatisfiability:

$$\not\models_I H \quad \text{(for any I)};$$

- Tautology verification (completeness):

$$\models H$$

- Unsatisfiability verification

$$\not\models H$$

Two principal issues:

- valid inference rules — checking:

$$(\Delta \vdash H) \quad \longrightarrow \quad (\Delta \models H)$$

- complete inference rules — checking:

$$(\Delta \models H) \quad \longrightarrow \quad (\Delta \vdash H)$$

The Deduction Theorems

Theorem 3 *Let $\Delta_1, \Delta_2, \ldots \Delta_n$ and Ω are logical formulas. Ω is their logical consequence iff $\Delta_1 \wedge \Delta_2 \wedge \ldots \Delta_n \Rightarrow \Omega$ is a tautology.*

Theorem 4 *Let $\Delta_1, \Delta_2, \ldots \Delta_n$ and Ω are logical formulas. Ω is their logical consequence iff $\Delta_1 \wedge \Delta_2 \wedge \ldots \Delta_n \wedge \neg\Omega$ is invalid (false under any interpretation).*

Theorem proving: having $\Delta_1, \Delta_2, \ldots \Delta_n$ assumed to be true show that so is Ω. Hence:

$$\Delta_1 \wedge \Delta_2 \wedge \ldots \Delta_n \models \Omega$$

Basic methods for theorem proving:

- evaluation of all possible interpretations (the 0-1 method),
- direct proof (forward chaining) – derivation of Ω from initial axioms; KRR: Rule-Based Systems, Expert Systems, Inference Graphs,...
- search for proof (backward chaining) – search for derivation of Ω from initial axioms; KRR: Backtracking Search, Abductive Reasoning, Diagnostic Systems, Question-Answering Systems, Prolog,...
- proving tautology – from the Deduction Theorem 1 we prove that $\Delta_1 \wedge \Delta_2 \wedge \ldots \Delta_n \Rightarrow \Omega$ is a tautology,
- indirect proof – through constraposition: $\neg\Omega \Rightarrow \neg(\Delta_1 \wedge \Delta_2 \wedge \ldots \Delta_n)$.
- Reductio ad Absurdum; basing on Deduction Theorem 2 we show that $\Delta_1 \wedge \Delta_2 \wedge \ldots \Delta_n \wedge \neg\Omega$.
  is unsatisfiable

Some most important Inference Rules (Fitch)

- AND Introduction (AI):

$$\frac{\phi_1, \ldots, \phi_n}{\phi_1 \wedge \ldots \wedge \phi_n}$$

- AND Elimination (AE):

$$\frac{\phi_1 \wedge \ldots \wedge \phi_n}{\phi_i}$$

- OR Introduction (OI):

$$\frac{\phi_i}{\phi_1 \wedge \ldots \wedge \phi_n}$$

- OR Elimination (OE):

$$\frac{\phi_1 \vee \ldots \vee \phi_n, \phi_1 \Rightarrow \psi, \ldots \phi_n \Rightarrow \psi}{\psi}$$

- Negation Introduction (NI):

$$\frac{\phi \Rightarrow \psi, \phi \Rightarrow \neg\psi}{\neg\phi}$$

- Negation Elimination (NE):

$$\frac{\neg\neg\phi}{\phi}$$

- Implication Introduction (II):

$$\frac{\phi \vdash \psi}{\phi \Rightarrow \psi}$$

- Implication Elimination (IE):

$$\frac{\phi, \phi \Rightarrow \psi}{\psi}$$

- Equivalence Introduction (EI),
- Equivalence Elimination (EE)

Extra Rules for Quantifiers (Fitch)

Universal Introduction (UI)

$$\frac{\Phi}{\forall X : \Phi}$$

Universal Elimination (UE)

$$\frac{\forall X : \Phi[X]}{\Phi[t]}$$

gdzie $t \in \mathrm{TER}$.

Existential Introduction (EI)

$$\frac{\Phi[t]}{\exists X : \Phi[X]}$$

Existential Elimination (EE)

$$\frac{\exists X : \Phi[X], \quad \forall Y : (\Phi[Y] \Rightarrow \Psi)}{\Psi}$$

where the variable Y does not occur in formula Ψ.

Dowodzenie Metodą Rezolucji

1. Zamiast dowodzić, że:

   $$\{\Delta_1, \Delta_2, \ldots, \Delta_n\} \models H$$

   dowodzimy niespełnialności

   $$\{\Delta_1, \Delta_2, \ldots, \Delta_n\} \cup \{\neg H\}$$

2. Wszystkie kwantyfikatory przesuwamy przed formułę.
3. Formułę bez kwanytyfikatorów przekształcamy do postaci CNF.
4. Eliminujemy kwantyfikatory egzystencjalne, zastępując zmienne termami; pozostają tylko kwantyfikatory ogólne (można je pominąć, gdyż wszystkie pozostałe zmienne są kwantyfikowane ogólnie).
5. W efekcie dostajemy zbiór zdań — jest to tzw. S-postać.
6. Stosując metodę rezolucji wyprowadzamy zdanie puste.

Reguła rezolucji dla zdań $C_1 = \phi \vee q_1$ oraz $C_2 = \varphi \vee \neg q_2$; σ jest podstawieniem unifikującym (mgu):

$$\frac{\phi \vee q_1, \varphi \vee \neg q_2}{\phi\sigma \vee \varphi\sigma}$$

Reguła faktoryzacji:

$$\frac{C}{C\theta}$$

Reguła faktoryzacji jest niezbędna dla przypadków typu:

$$\{p(X) \vee p(Y), \neg p(U) \vee \neg p(V)\}$$

Paradoks Fryzjera

There is a barber who was ordered to shave anyone who does not shave himself. Should he shave himself or not?

Formalizacja problemu w FOPC:

A. $\forall X \neg \text{shaves}(X, X) \Rightarrow \text{shaves}(\text{barber}, X)$ — anyone who does not shave himself is shaved by the barber.

B. $\forall Y \text{shaves}(\text{barber}, Y) \Rightarrow \neg \text{shaves}(Y, Y)$ — anyone who is not shaved by the barber shaves himself.

Transformacja do S-postaci:

- $C_1 = \text{shaves}(X, X) \vee \text{shaves}(\text{barber}, X)$,
- $C_2 = \neg \text{shaves}(\text{barber}, Y) \vee \neg \text{shaves}(Y, Y)$.

Niech $\theta = \{X/\text{barber}, Y/\text{barber}\}$.

$$C_1\theta = \text{shaves}(\text{barber}, \text{barber})$$

$$C_2\theta = \neg \text{shaves}(\text{barber}, \text{barber})$$

W wyniku rezolucji mamy:

$$\frac{\text{shaves}(\text{barber}, \text{barber}), \neg \text{shaves}(\text{barber}, \text{barber})}{\bot}.$$

Logiki Deskrypcyjne

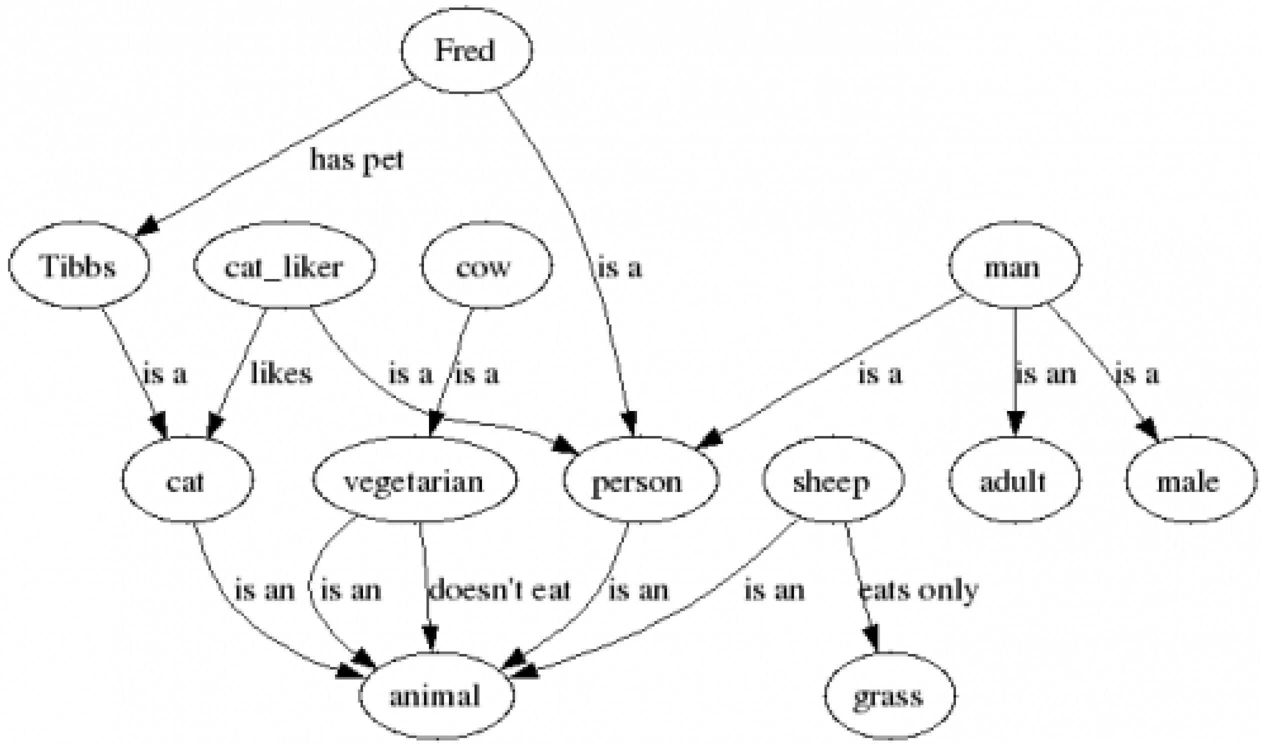

Figure 2: Example semantic net

Logiki Deskrypcyjne

Logiki deskrypcyjne (opisowe) — formalny sposób zapisu wiedzy (taksonomie, relacje, ograniczona kwantyfikacja: ontologie; umożliwiają ograniczone wnioskowanie.

- Logiki deskrypcyjne (opisowe) (ang. Description Logics, DL) są rodziną formalizmów reprezentacji wiedzy.
- Elementami reprezentacji są pojęcia (klasy) i instancje oraz role (relacje) (obiekty ((Nazwy w nawiasach używane są zwykle w ontologiach zapisanych w języku OWL, opartym na formalizmie DL))).
- Logiki opisowe są koncepcyjnie powiązane z sieciami semantycznymi (ang. semantic networks) i ramami (ang. frames), jednak w przeciwieństwie do nich, przez swoje powiązanie z logiką pierwszego rzędu, posiadaję formalnie zdefiniowaną semantykę i zapewniają możliwość automatycznego wnioskowania.
- Intuicyjnie można powiedzieć, że logiki opisowe łączą paradygmat obiektowy (ramy, sieci semantyczne) z logiką (rachunek predykatów, logika 1. rzędu).

Wybrane fragmenty wiedzy zapisane w logice opisowej:

- $Fred : person$, $Tibbs : cat$, $(Tred, Tibbs) : has\_pet$
- $man \equiv person \sqcap adult \sqcap male$, $cat\_liker \equiv person \sqcap \exists likes.cat$
- $(cat\_liker, cat) : likes$, $(sheep, grass) : eats\_only$
- $cat \sqsubseteq animal$, $sheep \sqsubseteq animal \sqcap \forall eats.grass$

Logiki Deskrypcyjne: Taksonomie i Relacje

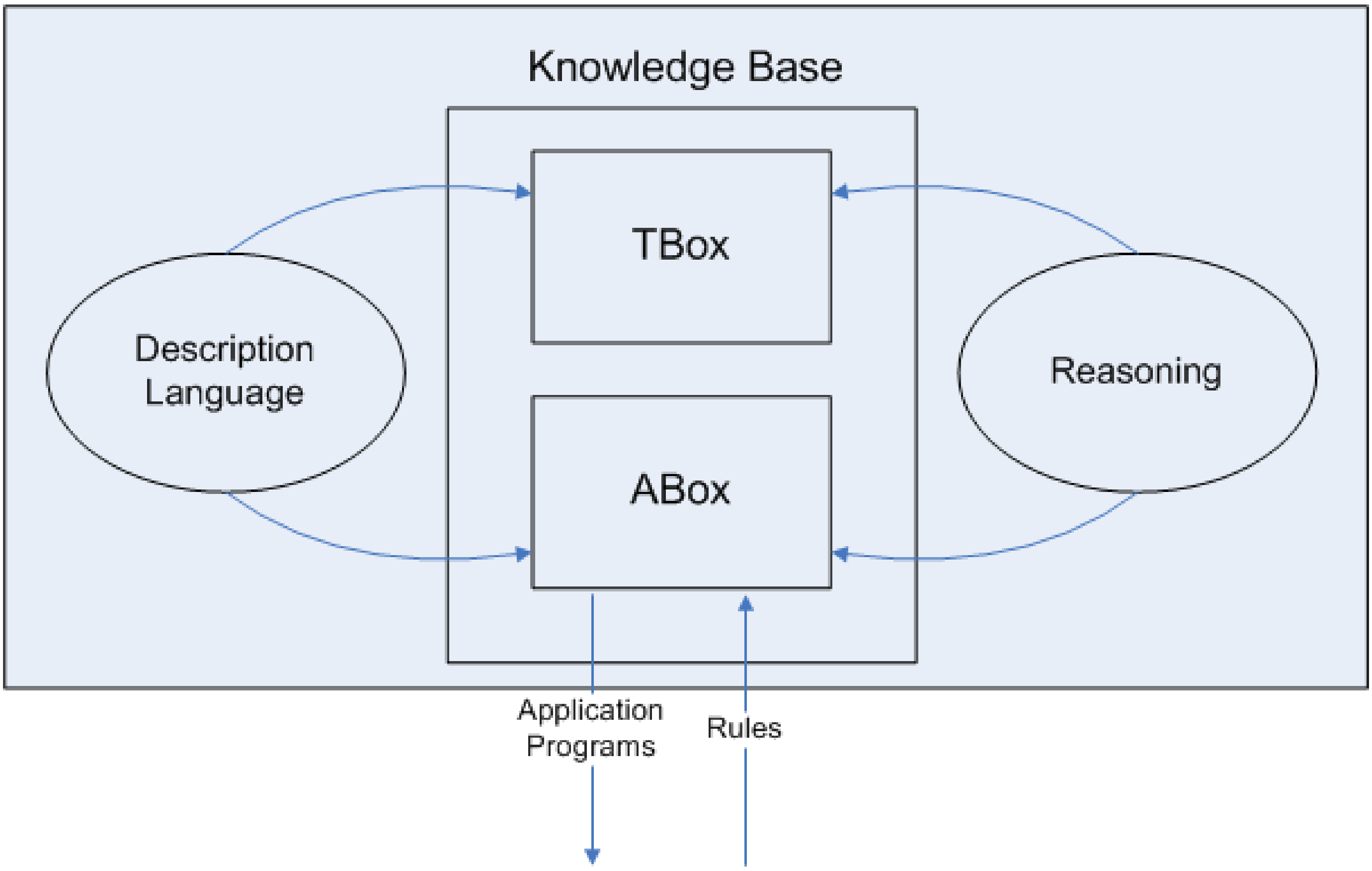

Figure 3: Komponenty bazy DL

Zastosowanie logik opisowych

Reprezentacja własności instancji (obiektu):

- przynależność obiektu do klasy (ang. concept assertions), np:
  - $Fred : person$ - Fred jest osobą
  - $Tibbs : cat$ - Tibbs jest kotem
- relacja między dwoma obiektami:
  - $(Fred, Tibbs) : has\_pet$ - Fred ma zwierzę, którym jest Tibbs

Definicje i własności pojęć:

- definicje pojęć (warunki konieczne i wystarczające), np.
  - $man \equiv person \sqcap adult \sqcap male$ - Mężczyzna to dorosła osoba rodzaju męskiego
  - $cat\_liker \equiv person \sqcap \exists likes.cat$ - Miłośnik kotów to osoba, która lubi (jakiegoś) kota
- relacje między pojęciami (klasami)
  - $(cat\_liker, cat) : likes$ - (każdy) miłośnik kotów lubi (jakiegoś) kota
  - $(sheep, grass) : eats\_only$ - (każda) owca je tylko trawę
- aksjomaty
  - $cat \sqsubseteq animal$ (każdy) kot jest zwierzęciem (hierarchia pojęć)
  - $sheep \sqsubseteq animal \sqcap \forall eats.grass$ owce to zwierzęta, które jedzą tylko trawę (warunek konieczny, ale nie wystarczający)

Bazowy język DL

Podstawowy język DL

- W języku logiki opisowej tworzymy opisy — formalizujemy ontologie.
- Podstawowe elementy języka to: atomicze pojęcia i atomiczne role.
- Złożone opisy tworzy się indukcyjnie za pomocą konstruktorów.
- Poszczególne języki DL różnią się między sobą zbiorem dopuszczalnych konstruktorów
- Najprostszy język to AL (ang. Attributive Language)

Składnia DL/AL

- atomiczne pojęcia $(A, B, ...)$
- atomiczne role $(R, S, ...)$
- opisy $(C, D, ...)$; mogą nimi być:
  - A - pojęcie atomiczne
  - $\top$ - //top concept//, pojęcie uniwersalne oznaczające 'wszystko'
  - $\bot$ - //bottom concept//, pojęcie puste, oznaczające 'nic'
  - $\neg A$ - negacja
  - $C \sqcap D$ - koniunkcja
  - $\forall R.C$ - kwantyfikator uniwersalny: "dla każdego"
  - $\exists R.\top$ - kwantyfikator egzystencjalny/szczegółowy: "istnieje"

Semantyka DL/AL

Semantyka DL/AL — Semantyka zdefiniowana jest poprzez //interpretację// składającą się z:

- dziedziny intepretacji: $\Delta^{\mathcal{I}}$ - niepustego zbioru, na który mapowane są symbole i relacje
- funkcji interpretacji, która przypisuje:
  - każdemu atomicznemu pojęciu zbiór: $A \rightarrow A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
  - każdej atomicznej roli relację binarną: $R^{\mathcal{I}} \rightarrow \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

Konstruktor	Składnia	Semantyka
pojęcie atomiczne (atomic concept)	A	$A^{\mathcal{I}} = A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
atomiczna rola (atomic role)	R	$R^{\mathcal{I}} = R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
pojęcie uniwersalne (universal concept)	$\top$	$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
pojęcie puste (bottom concept)	$\bot$	$\bot^{\mathcal{I}} = \emptyset$
(**atomic** negation)	$\neg A$	$(\neg A)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$
koniunkcja/przecięcie (intersection)	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
ograniczenie wartości (value restriction)	$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \vert \forall b, (a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$
ograniczony kwantyfikator egzystencjalny (limited existential quantification)	$\exists R.\top$	$(\exists R.\top)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \vert \exists b, (a,b) \in R^{\mathcal{I}}\}$

Przykłady formalizacji

Przykład:

- pojęcia atomiczne: //Person//, //Female//, //Elephant// ++|uwaga: dopóki nie zapisze się tego explicite, pomiędzy pojęciami nie występują żadne relacje. Są to po prostu oznaczenia jakichś zbiorów++
  - osoba rodzaju żeńskiego: $Person \sqcap Female$
  - słonica: $Elephant \sqcap Female$
  - osoba, które nie jest rodzaju żeńskiego: $Person \sqcap \neg Female$
- atomiczna rola: //hasChild//
  - osoba, która ma (jakieś) dziecko/dzieci: $Person \sqcap \exists hasChild.\top$
  - osoba, której wszystkie dzieci są rodzaju zeńskiego $Person \sqcap \forall hasChild.Female$
  - osoba bezdzietna $Person \sqcap \forall hasChild.\bot$

Rodzina języków DL

Poszczególne języki DL rozróżniamy poprzez konstruktory, które dopuszczają.

Przykładowe konkstruktory:

- $\mathcal{U}$ - suma : $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
- $\mathcal{E}$ - pełny kwantyfikator egzystencjalny : $(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} | \exists b, (a, b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$
- $\mathcal{N}$ - ograniczenia liczbowe:
- $(\geq nR)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} || \{b \mid (a, b) \in R^{\mathcal{I}}\} |\geq n\}$
- $(\leq nR)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} || \{b \mid (a, b) \in R^{\mathcal{I}}\} |\leq n\}$
- $\mathcal{C}$ - negacja : $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$

Używające powyższych konstruktorów języki nazywają się odpowiednio:

- $\mathcal{ALU}$
- $\mathcal{ALE}$
- $\mathcal{ALN}$
- $\mathcal{ALC}$ -> odpowiada podzbiorowi logiki pierwszego rzędu ograniczonemu do formuł z dwoma zmiennymi

Powiązanie z innymi rachunkami (logicznymi)

Większość logik opisowych jest podzbiorem logiki pierwszego rzędu:

- nazwy pojęć — predykaty unarne
- relacje (atomiczne) — predykaty binarne
- pojęcia — formuły z jedną wolną zmienną

Formuły logiki opisowej można intuicyjnie interpretować poprzez analogię do algebry zbiorów.

Przykład użycia	Składnia DL	Składnia FOL	Algebra zbiorów
Mężczyzna: **dorosły i osoba i rodzaju męskiego**	$C_1 \sqcap \ldots \sqcap C_n$	$C_1(x) \wedge \ldots \wedge C_n(x)$	$C_1 \cap \ldots \cap C_n$
Gazeta to **dziennik lub czasopismo**	$C_1 \sqcup \ldots \sqcup C_n$	$C_1(x) \vee \ldots \vee C_n(x)$	$C_1 \cup \ldots \cup C_n$
Wszystko, co jedzą wegetarianie to **nie mięso**	$\neg C$	$\neg C(x)$	C^c (dopełnienie zbioru)
Kraje UE to: **Niemcy, Francje, ..., Polska**	$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	$x = x_1 \vee \ldots \vee x = x_n$	$\{x_1\} \cup \ldots \cup \{x_n\}$
Każde zwierzę, które ma starsza pani **to kot**	$\forall P.C$	$\forall y.P(x,y) \rightarrow C(y)$	$\pi_Y(P) \subseteq C$
Właściciel psa **ma jakiegoś psa**	$\exists P.C$	$\exists y.P(x,y) \wedge C(y)$	$\pi_Y(P) \cap C \neq \emptyset$ ($sqcap$ - projekcja)
Rozsądny mężczyzna spotyka się z **maksymalnie 1 kobietą równocześnie** ;-)	$\leq nP$	$\exists^{\leq n} y.P(x,y)$	$card(P) \leq n$ (card - liczność zbioru)
Miłośnik zwierząt **ma minimum 3 zwierzaki**	$\geq nP$	$\exists^{\geq n} y.P(x,y)$	$card(P) \geq n$
Dziecko **to to samo co** młoda osoba	$C \equiv D$	$\forall x.C(x) \leftrightarrow D(x)$	$C \equiv D$
Każda foka jest zwierzęciem	$C \sqsubseteq D$	$\forall x.C(x) \rightarrow D(x)$	$C \subseteq D$