Logic for Knowledge Representation, Learning, and Inference

Luciano Serafini

serafini@fbk.eu

Version August 23, 2023

Contents

CHAPTER 1

Herbrand Theorem and Skolemization

To check satisfiability of a first order sentence ϕ on the signature Σ we have to produce an Σ-structure $\mathcal{I}$ that satisfies ϕ, i,e $\mathcal{I} \models \phi$. The naive procedure used for propositional logic, in which we check for all possible interpretations, is not working for FOL, since there are infinite many interpretations. Indeed we are free to choose the interpretation domain $\Delta^{\mathcal{I}}$ with any possibly infinite set, and therefore we have infinite possibilities to interpret the symbols of Σ.

The question, is whether there is a sistematic method to generate the Σ-strucutre for ϕ such that if ϕ is satisfiable, sooner or later we will encounter an interpretation that satisfies it.

The Herbrand's Theorem, called so after Jacques Herbrand (1908-1931), allows to reduces the problem of checking the satisfiability of a first-orde formula to the check of satisfiability of a set of propositional formulas. In this chapter we gradually introduce the theorem.

To keep the treatment simle in this chapter we consider only the case of First order language without equality.

1. Herbrand interpretation

Herbrand proposes the main idea to interpret terms in themselves. Notice that the definiton of Σ-structure $(\Delta^{\mathcal{I}}, \mathcal{I})$ $\Delta^{\mathcal{I}}$ can be any non empty set. Herbrand poposed to consider $\Delta^{\mathcal{I}}$ as the set of all ground terms that can be built from the signature Σ. Since $\Delta^{\mathcal{I}}$ must contain at least one elment, Herbrand required that Σ contains at least one constant symbol.

Definition 1.1 (Herbrand Universe). *The* Herbrand's universe *of a signature Σ that contains at least one constant symbol, is the set, denoted by $\Delta^{\mathcal{H}}$ of ground terms of Σ.*

Example 1.1. *If Σ contains two constants a and b and no function symbol then, the Herbrand's Universe of Σ is $\{a, b\}$ since a and b are the only ground terms that one can build in Σ. If, instead Σ contains a binary function symbo f then the set of ground terms, and therefore the Herbrand's Univers of Σ contains* *an infinite set of terms. i.e.,*

a	b		
$f(a,a)$	$f(a,b)$	$f(b,a)$	$f(b,b)$
$f(a,f(a,a))$	$f(a,f(a,b))$	$f(a,f(b,a))$	$f(a,f(b,b))$
$f(b,f(a,a))$	$f(b,f(a,b))$	$f(b,f(b,a))$	$f(b,f(b,b))$
$f(f(a,a),a)$	$f(f(a,b),a)$	$f(f(b,a),a)$	$f(f(b,b),a)$
$f(f(a,a),b)$	$f(f(a,b),b)$	$f(f(b,a),b)$	$f(f(b,b),b)$
$f(f(a,a),f(a,a))$	$f(f(a,a),f(a,b))$	$f(f(a,a),f(b,a))$	$f(f(a,a),f(b,b))$
$\dots$			

One can easily se that with one constant and a function symbol the Herbrand's Universe is infinite. Instead if there is no function symbols then the Herbrand universe has the same size of the number of constants in Σ.

An alternative way to define the Herbrand's Univers for Σ is by induction i.e., the herbrand universe $\Delta^{\mathcal{H}}_{\Sigma}$ for Σ is the smalles set that satisfies the following conditions

(1) Every constant of Σ belongs to $\Delta^{\mathcal{H}}_{\Sigma}$
(2) if $t_1, \dots, t_n \in \Delta^{\mathcal{H}}_{\Sigma}$ and f is an n-ary function symbol of Σ, then $f(t_1, \dots, t_n) \in \Delta^{\mathcal{H}}_{\Sigma}$.

Once we have defined the set $\Delta^{\mathcal{H}}$ to fully define an interpretation, we have to specify the interpretation function for the elements of Σ. The obvious way is to define the interpretation of constants and function symbols so that every terms is interpreted in itself, and every predicate with arity equal to n as a set of n-tuples of terms. i.e., in a subset of $\Delta^{\mathcal{H}}_{\Sigma}$.

Definition 1.2. *An herbrand interpretation of a signature Σ is composed by the pair $(\Delta^{\mathcal{H}}_{\Sigma}, \mathcal{H})$, where*

(1) $\Delta^{\mathcal{H}}_{\Sigma}$ is the Herbrand's universe of Σ;
(2) $\mathcal{H}(c) = c$ for every constant symbol $c \in \Sigma$;
(3) $\mathcal{H}(f) : t_1, \dots, t_n \mapsto f(t_1, \dots, t_n)$ is the function that maps an n-tuple of terms of $\Delta^{\mathcal{H}}_{\Sigma}$ in a term of Δ^{H}_{Σ}, for every n-ary function symbol f;
(4) $\mathcal{H}(P) \subseteq (\Delta^{H}_{\Sigma})^n$ is a set of n-tuples of terms in $\Delta^{\mathcal{H}}_{\Sigma}$, for evert n-ary predicate symbol $P \in \Sigma$.

A simpler way to see an Herbrand interpretation is by seeing it as a mapping from ground atomic formulas to $\{0, 1\}$.

$$\mathcal{H} : \text{GroundAtoms}(\Sigma) \to \{0, 1\} \tag{1}$$

This definition is very close to the definition of propositional interpretation, where GroundAtoms(Σ) is the set of proposiional variabels. The set GroundAtoms(Σ) is called the *Herbrand's base for* Σ.

Example 1.2. *The following is an example of an Herbrand Interpretation that satisfies the following set of formulas:*

$$\Gamma = \left\{ \begin{array}{c} \neg \mathit{friend}(x,x) \\ \mathit{friend}(x,y) \to \mathit{friend}(x,y) \\ \mathit{friend}(x,y) \to \mathit{knows}(x, \mathit{mother}(y)) \\ \mathit{friend}(\mathit{Mary}, \mathit{John}) \end{array} \right\}$$

$$\Delta_\Gamma^{\mathcal{H}} = \left\{ \begin{array}{l} \mathsf{Mary}, \mathsf{John}, \\ \mathsf{mother}(\mathsf{Mary}), \mathsf{mother}(\mathsf{John}), \\ \mathsf{mother}(\mathsf{mother}(\mathsf{Mary})), \mathsf{mother}(\mathsf{mother}(\mathsf{John})) \\ \mathsf{mother}(\dots \mathsf{mother}(\mathsf{Mary}) \dots), \ \mathsf{mother}(\dots \mathsf{mother}(\mathsf{John}) \dots), \dots \end{array} \right\}$$

$$\mathcal{H} = \left\{ \begin{array}{l} \mathsf{friend}(\mathsf{John}, \mathsf{Mary}), \ \mathsf{friend}(\mathsf{Mary}, \mathsf{John}), \\ \mathsf{knows}(\mathsf{John}, \mathsf{mother}(\mathsf{Mary})), \\ \mathsf{knows}(\mathsf{Mary}, \mathsf{mother}(\mathsf{John})), \\ \mathsf{knows}(\mathsf{mother}(\mathsf{Mary}), \mathsf{mother}(\mathsf{John})) \end{array} \right\}$$

2. Satisfiability in Herbrand's Interpretation

Satisfiability in Herbrand interpretations is defined as the problem of checking if a formula ϕ is satisfiable by an Herbrand's Interpretation on the signature of ϕ. One of the main version of the Herbrand's theorem states that satisfiability in general, can be reduced to satisfiability by an Herbrand's interpretation

PROPOSITION 1.1. *If $\mathcal{H}$ is an Herbrand interpretation then for every ground term t $\mathcal{H}(t) = t$.*

PROOF. By induction on the complexity of t. If t is the constant c then $\mathcal{H}(c) = c$ by definition. If $t = f(t_1, \dots, t_n)$ then

$$\begin{aligned} \mathcal{H}(f(t_1, \dots, t_n)) &= \mathcal{H}(f)(\mathcal{H}(t_1), \dots, \mathcal{H}(t_n)) \\ &= \mathcal{H}(f)(t_1, \dots, t_n) && \text{By induction hypothesis} \\ &= f(t_1, \dots, t_n) && \text{By definition } \mathcal{H}(f) \end{aligned}$$

□

PROPOSITION 1.2. *$\mathcal{H} \models \phi(x)[a_{x \leftarrow t]}]$ iff $\mathcal{H} \models \phi(t)$*

PROOF. By induction on the complexity of ϕ (exercize) □

PROPOSITION 1.3. *$\mathcal{H} \models \forall x\, \phi(x)$ if and only if $\mathcal{H} \models \phi(t)$ for all ground term t.*

PROOF.

$$\begin{aligned} \mathcal{H} \models \forall x\, \phi(x) \text{ fff } & \mathcal{H} \models \phi(x)[a_{x \leftarrow t}] \text{ for all ground terms } t \\ \text{fff } & \mathcal{H} \models \phi(t) \text{ for all ground terms } t \end{aligned}$$

□

DEFINITION 1.3 (quantifier-free formula). *A formula ϕ is quantifier-free if ϕ has no occurrence of either of the quantifiers $\forall$ or $\exists$.*

Notice that a quantifier-free formula is the combination of a set of atoms using the propositional connectives. Notice that all the individual variables that occours in a quantifier-free formula are free. Furthermore if a uantified free formula do not contains individual variables, then it is just a propositional formula.

EXAMPLE 1.3. *The following are examples of quantified free formulas.*

$$P(a) \vee Q(b, x) \rightarrow R(x, y, z)$$
$$R(a, b, f(c)) \vee R(b, a, g(a, b))$$

the second one does not contains individual variables, hence it is a propositional formula.

If we quantify universally the free variables of a quantified free formula we obtain a *universal sentence*.

DEFINITION 1.4 (Universal sentence). *A* universal sentence *is a sentence (closed formula of the form*

$$\forall x_1 \forall x_2 \dots \forall x_n \, \phi(x_1, \dots, x_n)$$

where $\phi(x_1, \dots, x_n)$ is a quantifier-free formula.

In other words universal sentences admit only universal quantifiers at the beginnin of the formula. We will see later that every formula can be transformed in an equi-satisfiable universal sentence. If we instantiate every variable of a universal sentence we obtain a propositional formula, that is called a ground ifnstance of the universal sentence.

DEFINITION 1.5 (Ground instance). *A* ground instance *of an universal sentence $\forall x_1 \dots \forall x_n . \phi(x_1, \dots, x_n)$ is a sentence $\phi(t_1, \dots, t_n)$ obtained by replacing each occurrence of x_i with a term t_i that does not contain variables.*

THEOREM 1.1 (Herbrand's Theorem). *A universal formula $\Phi = \forall x_1, \dots, \forall x_n \phi(x_1, \dots, x_n)$ is satisfiable if and only if it is true in an Herbrand interpretation in the signature of ϕ (if ϕ does not contain any constant we extend the signature with a constant a)*

PROOF (SKETCH). If Φ is satisfiable by an Herbrand interpretation then it is satisfiable. Let us proove the contrary. Suppose that Φ is satisfied by the interpretation $\mathcal{I}$. Starting from $\mathcal{I}$ we can build the following herbrand interpretation $\mathcal{H}$, on the domain of ground terms $\Delta_\Sigma^\mathcal{H}$ where Σ is the signature of Φ possibly extended with a constant a if Φ does not contain constant symbols. For every n-ary predicate p we define $\mathcal{H}(p)$

$$\mathcal{H}(p) = \{(t_1, \dots, t_n) \in \Delta_\Sigma^\mathcal{H} \mid \mathcal{I} \models p(t_1, \dots, t_n)\}$$

For every formula $\phi(x_1, \dots, x_n)$ we can prove by induction that the formula $\forall x_1, \dots, x_n \phi(x_1, \dots, x_n) \rightarrow \phi(t_1, \dots, t_n)$ is valid for every n-tuple of ground terms. This implies that

$$\mathcal{I} \models \phi(t_1, \dots, t_n) \tag{2}$$

and therefore that $\mathcal{H} \models \phi(t_1, \dots, t_n)$. This implies that

$$\mathcal{H} \models \phi(x_1, \dots, x_n)[a_{x_1 \leftarrow t_1, \dots, x_n \leftarrow t_n}]$$

and therefore that $\mathcal{H} \models \Phi$. □

The immediate consequence of the Herbrand's theorem is that, to check if $\Phi = \forall x_1, \dots, x_n \, \phi(x_1, \dots, x_n)$ is satisfiable we can check if it is satisfiable only in the herbrand interpretations. If there is no herbrand interpretations that satisfies Φ then the formula is surely unsatisfiable.

A second, and related consequence, is that if Φ is unsatisfiable, then also the set

$$\text{Ground}(\Phi) = \{\phi(t_1, \dots, t_n) \mid t_i \in \Delta_\Sigma^\mathcal{H}\}$$

is not sartistiable. But one can notice that Ground(Φ) is a set of propositional formula, and therefore we can apply the main results of satisfiability in proposiitional formula. In particulare, we use the compactenss theorem (Theorem **??**) that states that an infinite set of proposiitonal formula Γ is not satisfiable if and only if there is a finite subset Γ_0 of Γ that is not satisfiable. We can therefore conclude that Φ is unsatisfiable there is a finite set $G \subset \text{Ground}(\Phi)$ of groundings of Φ that is unsatisfiable.

If we find a way to enumerate $G_0, G_1, G_2, \ldots$ (i.e., generate an infinite sequence) of all the finite subsets, of $\text{Ground}(\Phi)$ such that for every finite subset $G \subset \text{Ground}(\Phi)$ there is an i such that $C_i = C$ we could check at every iteration if G_i is satisfiable, and if $\text{Ground}(\Phi)$ is not satisfiable we enventuall find an i such that G_i is not satisfiable. This very naive idea is implemented in Algorithm 1. If

Algorithm 1 First Order Satisfiability,

Require: A universal formula $\Phi = \forall x_1, \ldots, \forall x_n \, \phi(x_1, \ldots, x_n)$
$\Sigma \leftarrow$ the signature of Φ
if $\text{Constants}(\Sigma) = \emptyset$ **then**
  $\Sigma \leftarrow \Sigma \cup \{a\}$
end if
$\Delta \leftarrow \text{Constants}(\Sigma)$
while True **do**
  $G \leftarrow \textsc{Ground}(\Phi, \Delta)$
  if PropositionalSat(G)=Unnsat **then**
    return Unsat
  end if
  $\Delta \leftarrow \Delta \cup \{f(t_1, \ldots, t_n) \mid f \in n\text{-ary-Funct}(\Sigma),\ t_i \in \Delta\}$
end while

Φ is unsat, then by the Herbrand theorem we have that there is a finite subset of $\text{Grounding}(\Phi)$ that is unsat let k be the masimum dephth of the terms that appear in G, then at the k-th iteration the set G will be a subset of $\textsc{Grounding}(\Phi, \Delta)$ which sill be inconsistent, and therefore the algorithm terms returning Unsat

3. Prenex normal form

In the previous section we only consider universally quantified formulas. In this section we show how to extend this result to the entire set of first order formulas, that inculdes also existential quantified formulas.

Definition 1.6. *A formula is in* prenex normal form *if it is in the form of*

$$Q_1 x_1 \ldots, Q_n x_n \, \phi(x_1, \ldots, x_n) \tag{3}$$

where each Q_i is either $\exists$ or $\forall$ and $\phi(x_1, \ldots, x_n)$ is a quantified free first order formula.

Every formula can be reduced in prenex normal form by using the following rewriting rules:

- rewrite the $\rightarrow$ and $\leftrightarrow$ in terms of $\neg$ and $\vee$ and $\wedge$;
- switch the $\neg$ and the quantifiers with the rule:

$$\neg \forall x \phi \Longrightarrow \exists x \neg \phi$$
$$\neg \exists x \phi \Longrightarrow \forall x \neg \phi$$

- switch the binary connectives $\wedge$ and $\vee$ and the quantifier with the rules under the hypothesis that x does not

$$\begin{aligned}\forall x\, \phi(x) \wedge \psi &\Longrightarrow \forall x(\phi(x) \wedge \psi)\\ \forall x\, \phi(x) \vee \psi &\Longrightarrow \forall x(\phi(x) \vee \psi)\\ \exists x\, \phi(x) \wedge \psi &\Longrightarrow \exists x(\phi(x) \wedge \psi)\\ \exists x\, \phi(x) \vee \psi &\Longrightarrow \exists x(\phi(x) \vee \psi)\end{aligned}$$

  If x appears free in ψ we can rewrite $Qx\, \phi(x)$ into the equivalent formula $Qy\, \phi(y)$ for some new variable y before applying the rules.
- the following rules can also be used but not strictly necessary

$$\begin{aligned}\exists x\, \phi(x) \vee \exists x \psi(x) &\Longrightarrow \exists x(\phi(x) \vee \psi(x))\\ \forall x\, \phi(x) \wedge \forall x \psi(x) &\Longrightarrow \forall x(\phi(x) \wedge \psi(x))\end{aligned}$$

- finally it is possible to switch the existential and universal quantifier with the following rule

$$\forall x \exists y(\phi(x) \circ \psi(y)) \Longrightarrow \exists y \forall x(\phi(x) \circ \psi(y))$$

  if x is not free in $\psi(y)$ and y is not free in $\phi(x)$. As it will be clearer later, moving the existential quantifier out of the scope of an universal quantifier can be convenient.

Let us see an example about how to rewrite a formula in prenex normal form

Example 1.4. *Consider the formula*

$$(\forall x \exists y P(x, y) \rightarrow \exists x Q(x)) \vee \forall x Q(x)$$

We first rewrite the $\rightarrow$

$$(\neg \forall x \exists y P(x, y) \vee \exists x Q(x)) \vee \forall x Q(x)$$

Then we push the $\neg$ *in front of atoms*

$$(\exists x \forall y \neg P(x, y) \vee \exists x Q(x)) \vee \forall x Q(x)$$

Then we can apply the rule that commutes $\exists x$ *and* $\vee$ *on the first disjunct*

$$\exists x(\forall y \neg P(x, y) \vee Q(x)) \vee \forall x Q(x)$$

and push out the $\forall x$ *quantifier*

$$\exists x \forall y(\neg P(x, y) \vee Q(x)) \vee \forall x Q(x)$$

We can also push out the first existential quantifier since x *is not free in* $\forall x Q(x)$

$$\exists x \forall y(\neg P(x, y) \vee Q(x) \vee \forall x Q(x))$$

Now if we want to push out the quantifier $|forallx$ *since* x *is free in* $\neg P(x, y) \vee Q(x)$ *we have to rename the variablel obtaining*

$$\exists x \forall y(\neg P(x, y) \vee Q(x) \vee \forall z Q(z))$$

now we can apply the rule to obtain

$$\exists x \forall y \forall z(\neg P(x, y) \vee Q(x) \vee Q(z))$$

which is in prenex normal form.

4. Skolemization

Skolem normal form is named after the late Norwegian mathematician Thoralf Skolem.(1887–1963). Skolemization is the operator of replacing existential quantifiers either with constants (0-ary functions) or with functions, obtaining an equisatisfiable formula.

Before providing the general definition let us consider the following siple example of proposition in FOL. Consider the proposition "Every preogrammer has written at least one computer program", In FOL this can be formalized as

$$\forall x(\mathsf{Programmer}(x) \rightarrow \exists y(\mathsf{Program}(y) \wedge \mathsf{Author}(x,y))$$

If it is the case that for every programmer we can find a program written by him/her, there exists a function from programmers to programs that selects one program for every programmer, and such that the author of the program selected by this function for some programmer x is x him/herself. Notice that there might be more than one program written by the same programmer, however f will pik one of them. To formalize this line of reasoning, we can extend the signature with a new symbol f that intuitively represent the funciton that select one program for every programmer, and we use f in place of the existential quantifier, by rewriting the original formula in

$$\forall x(\mathsf{Programmer}(x) \rightarrow \mathsf{Program}(\mathsf{f}(x)) \wedge \mathsf{Author}(x, \mathsf{f}(x)))$$

The previous example can be generalized by rewriting any formula of tghe form $\forall x \exists y \phi(x, y)$ in $\forall x \phi(x, f(x))$ for some new function symbol f. This is also possible when there is no universal quantifier in front of $\exists$. I.e., The formula $\exists x \phi(x)$ can be rewritten in $\phi(a)$ for some new constant a. Generalizing even more the formula $\forall x \forall y \exists z \phi(x, y, z)$ can be rewritten in $\forall x \forall y \phi(x, y, f(x, y))$ for some new binary function symbol f. Let us make this process fully general.

Definition 1.7 (Skolemization). *Let Φ be a formula in prenex normal form that start with m universal quantifiers followed by an existential quantifier. I.e., Φ is in the form:*

$$\forall x_1 \forall x_2 \ldots \forall x_m \exists x_{m+1} Q_{m+1} x_{m+2} \ldots Q_n x_n \phi(x_1, \ldots, x_n)$$

a formula in prenex normal form the Skolemization if the operation of introducing a new n-ary function symbol f and replace x_{m+1} with $f(x_1, \ldots, x_m)$, and remove the existential quantifier. I.e., transforming the formula in

$$\forall x_1 \forall x_2 \ldots \forall x_m Q_{m+2} x_{m+2} \ldots Q_n x_n \phi(x_1, \ldots, x_m, f(x_1, \ldots, x_m), x_{m+2}, \ldots, x_n)$$

Proposition 1.4. *Let Ψ be the Skolemization of a forula Φ Every model that satisfies Φ can be extended to an interpretation $\mathcal{I}$ by providing the interpretation of the skolem function f that satisfies Ψ.*

Proof. We prove the property for the special case where Φ is $\forall x \exists y R(x, y)$, The general proof looks the same. In this special case Ψ, the skolemization of Φ is $\forall x R(x, f(x))$. Let us show that Φ is satisfiable iff Ψ is satisfiable

($\Longrightarrow$) If $\forall x \exists y R(x, y)$ is satisfiable, then there is an interpretation $\mathcal{I}$, such that $\mathcal{I} \models \forall x \exists y\, R(x, y)$. This implies that, for every element $d \in \Delta^{\mathcal{I}}$, there is an elmenet $d' \in \Delta^{\mathcal{I}}$ such that $(d, d') \in \mathcal{I}(R)$. Let $\mathcal{I}'$ be the interpretation on the same domain of $\mathcal{I}$ with $\mathcal{I}(R) = \mathcal{I}'(R)$ and $\mathcal{I}'(f)$ is a function that maps d into a d' such that

$(d, d') \in \mathcal{I}(R)$. This implies that for every $d \in \Delta^{\mathcal{I}'}$, $\mathcal{I}' \models R(x, f(x))[x \leftarrow d]$; and therefore that $\mathcal{I}' \models \forall x R(x, f(x))$.

($\Longleftarrow$) If $\forall x R(x, f(x))$ is satisfiable, there is an interpretation $\mathcal{I}$ of R and f such that for every $d \in \Delta^{\mathcal{I}}$, $(d, \mathcal{I}(f)(d)) \in \mathcal{I}(R)$ and therefore for every $d \in \Delta^{\mathcal{I}}$ there is a d' (which is $\mathcal{I}(f)(d)$) such that $(d, d') \in \mathcal{I}$. This implies that $\mathcal{I} \models \forall x \exists y R(x, y)$. If we consider $\mathcal{I}'$ the restriction of $\mathcal{I}$ to the signature that contains only R, we have that $\mathcal{I}' \models \forall x \exists y R(x, y))$ and therefore that $\forall x \exists y\, R(x, y)$ is satisfiable. $\square$

Bibliography

Arp, Robert, Barry Smith, and Andrew D Spear (2015). *Building ontologies with basic formal ontology*. Mit Press.

Badreddine, Samy et al. (2022). “Logic tensor networks”. In: *Artificial Intelligence* 303, p. 103649.

Baumgartner, Peter et al. (2009). “Computing finite models by reduction to function-free clause logic”. In: *Journal of Applied Logic* 7.1, pp. 58–74.

Birnbaum, Elazar and Eliezer L Lozinskii (1999). “The good old Davis-Putnam procedure helps counting models”. In: *Journal of Artificial Intelligence Research* 10, pp. 457–477.

Borgo, Stefano and Claudio Masolo (2009). “Foundational choices in DOLCE”. In: *Handbook on ontologies*. Springer, pp. 361–381.

Boros, Endre and Peter L Hammer (2002). “Pseudo-boolean optimization”. In: *Discrete applied mathematics* 123.1-3, pp. 155–225.

Brewka, Gerhard (1989). “Nonmonotonic Logics–A Brief Overview”. In: *AI Communications* 2.2, pp. 88–97.

Chakraborty, Supratik et al. (2015). “From weighted to unweighted model counting”. In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Chavira, Mark and Adnan Darwiche (2008a). “On probabilistic inference by weighted model counting”. In: *Artificial Intelligence* 172.6-7, pp. 772–799.

— (2008b). “On probabilistic inference by weighted model counting”. In: *Artificial Intelligence* 172.6, pp. 772–799. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2007.11.002. URL: https://www.sciencedirect.com/science/article/pii/S0004370207001889.

Daniele, Alessandro and Luciano Serafini (2019). “Knowledge enhanced neural networks”. In: *PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26–30, 2019, Proceedings, Part I 16*. Springer, pp. 542–554.

Darwiche, Adnan (2020). “Three modern roles for logic in AI”. In: *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pp. 229–243.

Davis, Ernest (2017). “Logical formalizations of commonsense reasoning: a survey”. In: *Journal of Artificial Intelligence Research* 59, pp. 651–723.

Davis, Martin, George Logemann, and Donald Loveland (1962). “A machine program for theorem proving”. In: *Communications of the ACM* 5.7, pp. 394–397.

Davis, Martin and Hillary Putnam (1960). “A computing procedure for quantification theory”. In: *Journal of ACM* 7, pp. 201–215.

De Raedt, Luc et al. (2020). “From statistical relational to neuro-symbolic artificial intelligence”. In: *arXiv preprint arXiv:2003.08316*.

Franco, John and Marvin Paull (1983). “Probabilistic analysis of the Davis Putnam procedure for solving the satisfiability problem”. In: *Discrete Applied Mathematics* 5.1, pp. 77–87.

Fu, Zhaohui and Sharad Malik (2006). “On solving the partial MAX-SAT problem”. In: *International Conference on Theory and Applications of Satisfiability Testing*. Springer, pp. 252–265.

Gomes, Carla P., Ashish Sabharwal, and Bart Selman (2009). “Model Counting”. In: *Handbook of Satisfiability*, pp. 633–654.

Gruber, Thomas R (1993). “A translation approach to portable ontology specifications”. In: *Knowledge acquisition* 5.2, pp. 199–220.

Guizzardi, RS (2015). “Towards ontological foundations for conceptual modeling: the unified foundational ontology (UFO) story Appl”. In: *Ontol* 10, pp. 3–4.

Gurevich, Yuri (1985). “Chapter XIII: Monadic second-order theories”. In: *Model-theoretic logics* 8, pp. 479–506.

Gutmann, Bernd, Ingo Thon, and Luc De Raedt (2011). “Learning the parameters of probabilistic logic programs from interpretations”. In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I 11*. Springer, pp. 581–596.

Holtzen, Steven, Guy Van den Broeck, and Todd Millstein (2020). “Scaling exact inference for discrete probabilistic programs”. In: *Proceedings of the ACM on Programming Languages* 4.OOPSLA, pp. 1–31.

Jaeger, Manfred and Guy Van den Broeck (2012). “Liftability of probabilistic inference: Upper and lower bounds”. In: *Proceedings of the 2nd international workshop on statistical relational AI*.

Kimmig, Angelika et al. (2011). “On the implementation of the probabilistic logic programming language ProbLog”. In: *Theory and Practice of Logic Programming* 11.2-3, pp. 235–262.

Kuusisto, Antti and Carsten Lutz (2018). “Weighted model counting beyond two-variable logic”. In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*. Ed. by Anuj Dawar and Erich Grädel. ACM, pp. 619–628. DOI: 10.1145/3209108.3209168. URL: https://doi.org/10.1145/3209108.3209168.

Lenzerini, Maurizio (2002). “Data integration: A theoretical perspective”. In: *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 233–246.

Lifschitz, Vladimir, Bruce Porter, and Frank Van Harmelen (2008). *Handbook of Knowledge Representation*. Elsevier.

Lowd, Daniel and Pedro Domingos (2007). “Efficient weight learning for Markov logic networks”. In: *Knowledge Discovery in Databases: PKDD 2007: 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007. Proceedings 11*. Springer, pp. 200–211.

Lukasiewicz, Thomas (1998). “Probabilistic Logic Programming.” In: *ECAI*, pp. 388–392.

Maathuis, Marloes et al. (2018). *Handbook of graphical models*. CRC Press.

Mala, Firdous Ahmad (2022). “On the number of transitive relations on a set”. In: *Indian Journal of Pure and Applied Mathematics* 53.1, pp. 228–232.

Manhaeve, Robin et al. (2018). "Deepproblog: Neural probabilistic logic programming". In: *advances in neural information processing systems* 31.

Manquinho, Vasco, Joao Marques-Silva, and Jordi Planes (2009). "Algorithms for weighted boolean optimization". In: *International conference on theory and applications of satisfiability testing*. Springer, pp. 495–508.

McCarthy, John (1959). "Programs with Common Sense". In: pp. 77–84.

McCune, William (2003). "Mace4 reference manual and guide". In: *arXiv preprint cs/0310055*.

Minker, Jack (2012). *Logic-based artificial intelligence*. Vol. 597. Springer Science & Business Media.

Mohamedou, Nouredine Ould and Jordi Planes (2009). "Solver MaxSatz in Max-SAT Evaluation 2009". In: *SAT 2009 competitive events booklet: preliminary version*, p. 155.

OEIS Foundation Inc. (n.d.). *The On-Line Encyclopedia of Integer Sequences*. Published electronically at `http://oeis.org`.

Poole, David (2003). "First-order probabilistic inference". In: *IJCAI*. Vol. 3, pp. 985–991.

Reger, Giles, Martin Suda, and Andrei Voronkov (2016). "Finding finite models in multi-sorted first-order logic". In: *International Conference on Theory and Applications of Satisfiability Testing*. Springer, pp. 323–341.

Richardson, Matthew and Pedro Domingos (2006). "Markov logic networks". In: *Machine learning* 62, pp. 107–136.

Russell, Stuart and Peter Norvig (2010). *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall.

Sang, Tian, Paul Beame, and Henry Kautz (2005). "Solving Bayesian networks by weighted model counting". In: *Proc.AAAI-05*. Vol. 1, pp. 475–482.

Saveri, Gaia and Luca Bortolussi (2022). "Graph Neural Networks for Propositional Model Counting". In: *arXiv preprint arXiv:2205.04423*.

Scott, D. (1962). "A decision method for validity of sentences in two variables". In: *Journal of Symbolic Logic* 27, p. 377.

Selman, Bart, Henry A Kautz, Bram Cohen, et al. (1993). "Local search strategies for satisfiability testing." In: *Cliques, coloring, and satisfiability* 26, pp. 521–532.

Song, Shaoxu et al. (2014). "Repairing vertex labels under neighborhood constraints". In: *Proceedings of the VLDB Endowment* 7.11, pp. 987–998.

Torlak, Emina and Daniel Jackson (2007). "Kodkod: A relational model finder". In: *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, pp. 632–647.

Tseytin, Grigori (1966). *On the complexity of derivation in propositional calculus*. Presented at the Leningrad Seminar on Mathematical Logic. URL: `http://www.decision-procedures.org/handouts/Tseitin70.pdf`.

Wei, Wei and Bart Selman (2005). "A new approach to model counting". In: *International Conference on Theory and Applications of Satisfiability Testing*. Springer, pp. 324–339.

Wilf, Herbert S (2005). *Generatingfunctionology*. CRC press.

Xiao, Guohui et al. (2018). "Ontology-based data access: A survey". In: International Joint Conferences on Artificial Intelligence.

Zhang, Jian and Hantao Zhang (1996). "System description generating models by SEM". In: *Automated Deduction—Cade-13: 13th International Conference on Automated Deduction New Brunswick, NJ, USA, July 30–August 3, 1996 Proceedings 13*. Springer, pp. 308–312.