Minimal Models and Expressiveness Hierarchy*

Francicleber M. Ferreira[1], **Ana Teresa Martins**[1]

[1]Departamento de Computação – Universidade Federal do Ceará
Caixa Postal 12.166 – 60.455-760 – Fortaleza – CE – Brazil

{fran,ana}@lia.ufc.br

Abstract. *The minimality concept is widely used in Computer Science to define objects. We find this concept in the definition of inductive sets, in denotational semantics for recursive programs, in Logic Programming, in Artificial Intelligence, etc. We analysed the minimality concept from a logical standpoint. We investigated logics whose semantics are based on minimal models, and its applications to Computer Science. Among our main results, we demonstrated theorems about definability and the expressive power of Least Fixed-Point Logic (LFP), Lifschitz's Nested Abnormality Theories (NATs), McCarthy's Circumscription and the MIN extension of van Benthem's MIN(FO) Logic.*

1. Introduction

Several important notions in Computer Sciences are defined through minimal objects. For instance, we see the use of the minimality concept: in the precise definition of the meaning of recursivity through the Denotational Semantics of Programming Languages [Scott 1982, Schmidt 1986], in the minimal Herbrand model semantics for Horn clauses in Logic Programming [Lloyd 1987], in the minimal models semantics used by McCarthy's Circumscription for the representation of and the reasoning over incomplete knowledge [McCarthy 1986, Lifschitz 1994] in Artificial Intelligence, etc. The concept of minimal object (or element) arises in the Theory of Order. Given a pre-order $\leq$ (a transitive and reflexive binary relation) over a set A, an element $a \in A$ is called *minimal* if there is no $b \in A$ such that $b \leq a$ and $a \not\leq b$. Hence, when we talk about minimal objects, we have in mind an underlying partial order relation.

In the study of Programming Languages, a function or procedure which makes a call to itself, that is, which occurs in its definition body, is called a recursive function. Many times, a recursive program enters a sequence of recursive calls which does not reach an end (in other words, which does not reach the basis of the recursion). If one looks at the recursive definition of the program, one can see that, in the cases that a recursive program gets into a endless sequence of recursive calls, there are many possible interpretations for the function computed by the recursive program [Schmidt 1986]. One can ask: what interpretation is correct? Both the meaning given by the semantic of the language and the operational behavior of the program's computation must coincide. In [Scott 1982], Scott gives an answer to this question. In Scott's Domain Theory, a system of recursive equations gives rise to a partially ordered structure and an operator, over that structure, whose fixed-points are solutions to the system of recursive equations. Moreover, it can be proved that such operator have a least (a particular case of minimal) fixed-point. That least

*This Master's Thesis is available for download at:
http://www.lia.ufc.br/%7efran/TeseFerreira.pdf.

fixed-point is the interpretation given to recursive programs by the denotational semantics of programming languages.

In Logic Programming, programs are written as sets of statements which can be thought of as Horn clauses [Lloyd 1987], a fragment of first-order logic. Again, it may be the case that many possible interpretations to this set of Horn clauses are possible, in other words, this set of Horn clauses may have many different non-isomorphic models. However, sets of Horn clauses have the least Herbrand model property, that is, a set of Horn clauses has only one least Herbrand model, up to isomorphism. This least Herbrand model gives the interpretation to the logic program.

In Artificial Intelligence, we often have to deal with the situation where the knowledge about the problem to be solved is not completely available. And we may find ourselves in a position where the available information does not suffice to decide which action has to be taken. In this case, we may have to suppose certain hypotheses and further discover that such hypotheses are not the case. To treat such sort of problems, many non-monotonic logical formalisms were proposed in the 80's. One of them is McCarthy's Circumscription [McCarthy 1986]. According to Circumscription, the objects and the properties they have in a problem description are supposed to be as few as possible. That is, in Circumscription, the intended interpretations of the problem description are those in which the set of involved objects and the extension of the predicates are the smallest possible ones, in other words, the minimal models of the description.

The few examples above rapidly illustrate the importance of minimal objects in Computer Science. More specifically, all the examples above are similar in the sense that the desired minimal object is a minimal interpretation of a description written in certain formal language, that is, the minimal models of a theory in some formal language.

Our work explores the consequences to Computer Science of two important properties of logics from the standpoint of Model Theory, namely, expressive power and definability. Expressive power is the capability of a logic to axiomatize or distinguish between classes of models, and definability is the power of a logic to define, in the sense of Beth [Beth 1953] and Padoa [Padoa 1900], the symbols of the underlying formal language. The main logics which we investigated were the Least Fixed-Point Logic (LFP) [Dawar and Gurevich 2002] (Section 2 below), McCarthy's Circumscription and Lifschitz's Nested Abnormality Theories (Section 3 below) and our extension of van Benthem MIN(FO) logic: the MIN logic (Section 4 below). These logics are characterised by using minimal models in their semantics. In Section 5, we will make some conclusions about our results presented in the previous sections. The theorems and lemmas explicitly stated below belong to the main contributions of our work in [Ferreira 2007].

Throughout this paper, we will use the commonly used logic notation, see [Ebbinghaus et al. 1994] for the notational details and definitions. The minimal models we will talk about are defined as follows. A symbol set S is a set of relation, function and constant symbols. Models are mathematical structures on some symbol set, that is, an S-structure $\mathfrak{A} = \langle A, \sigma \rangle$ is a pair where A is the domain and σ is a function which maps each n-ary relation (function) symbol in S to n-ary relation (function) on A, and each constant symbol in S to an element of A. We define the relation $\leq^{P;\overline{Z}}$ on $S \cup \{P\}$-structures on the same domain as:

$$\mathfrak{A} \leq^{P;\overline{Z}} \mathfrak{B} \quad \text{iff} \quad \sigma^{\mathfrak{A}}(s) = \sigma^{\mathfrak{B}}(s), \text{ for } s \in S \cup \{P\} - \{\overline{Z}\}, \text{ and } \sigma^{\mathfrak{A}}(P) \subseteq \sigma^{\mathfrak{B}}(P). \quad (1)$$

Given a class $\mathbb{C}$ of $S \cup \{P\}$-structures, a structure $\mathfrak{A} \in \mathbb{C}$ is a $P;\overline{Z}$-minimal element of $\mathbb{C}$ iff there is no $\mathfrak{B}$ such that $\mathfrak{B} \leq^{P;\overline{Z}} \mathfrak{A}$ but not $\mathfrak{A} \leq^{P;\overline{Z}} \mathfrak{B}$. Given a sentence ϕ of, for example, first-order logic, let $Mod(\phi)$ be the class of models of ϕ. A $P, \overline{Z}$-minimal model of ϕ is a $P, \overline{Z}$-minimal element of $Mod(\phi)$ with respect to $\leq^{P;\overline{Z}}$.

2. Least Fixed-Point Logic

The Least Fixed-Point Logic (see [Dawar and Gurevich 2002] for a survey on fixed-point logics) extends classical First-Order (FO) logic by adding to the language a syntactic construction that allows one to write predicate-like expressions interpreted as specific inductive sets. These inductive sets are the least fixed-points of the monotone operators defined by positive formulas on the domain of a given structure. The existence of such least fixed-point is assured by the Knaster-Tarski theorem.

The possibility of defining such inductive sets gives to LFP greater expressive power than FO. This can be seen, for example, from the fact that LFP can axiomatize the standard model of Peano Arithmetics (see, for instance, [Ferreira 2007, Exemplo 2.3]), which cannot be done in FO [Ebbinghaus et al. 1994].

A general question regarding the expressive power of a logic is whether it can express cardinality constraints, or, in other words, whether it can axiomatize classes of structures with a given cardinality. An important theorem regarding cardinality issues is the Downward Löwenheim-Skolem Theorem (DLST), which states that any satisfiable sentence of FO has a countable model [Ebbinghaus et al. 1994]. This theorem is true for LFP, although proof of it is hard to find in literature. Two sketches of the proof can be found in papers of Grädel and Flum (see [Ferreira 2007] for references).

The importance of DLST for LFP can be better understood under the light of Lindström's Second Theorem [Ebbinghaus et al. 1994]. Lindström's Second Theorem states that, under certain conditions, a logic at least as expressive as FO for which DLST holds, and whose set of valid sentences is recursively enumerable, has exactly the same expressive power of FO (that is, it is not more expressive than FO). Applying Lindström's Second Theorem to LFP together with the fact that LFP is strictly more expressive than FO, DLST for LFP implies that the set of valid formulas of LFP is not even semi-decidable.

In [Ferreira 2007, Ferreira and Martins 2007], we have stated and proved a generalized version of DLST for LFP:

Theorem 1 (Generalized DLST for LFP) *Any satisfiable theory Γ of LFP, with cardinality $|\Gamma|$ has a model with cardinality less than or equal to $|\Gamma|$.*

A generalized version of DLST also holds for FO, as it is well known (see [Ebbinghaus et al. 1994]).

Another important property of a logic, narrowly related to its expressive power, is its capability of defining the symbols of the underlying language. Given a logic, say $\mathcal{L}$, and a set Δ of $\mathcal{L}$-sentences written from a symbol set $S \cup \{P\}$, we say that Δ implicitly defines the symbol P iff any two models of Δ with the same domain and which gives exactly the same interpretation to the symbols in S also gives the same interpretation to the symbol P. A question that immediately arises is whether the symbol P can be explicitly defined from the symbols in S or not. By an explicit definition we mean an expression that syntactically defines P from the symbols of S only. For instance, in FO, an explicit definition for P has the form $\forall\overline{x}(P(\overline{x}) \leftrightarrow \psi(\overline{x}))$, where P does not occur in $\psi(\overline{x})$.

The so called Padoa's Method [Padoa 1900] consists of showing that in FO an explicit definition implicitly defines the explicitly defined symbol. Beth showed that the converse also holds for FO: whenever a set of FO-sentences Δ implicitly defines a predicate symbol P, there is an explicit definition $\delta = \forall\overline{x}(P(\overline{x}) \leftrightarrow \psi(\overline{x}))$ such that $\Delta \models \delta$ [Beth 1953]. In [Ferreira 2007], we gave a proof that Beth's Definability Theorem (BDT) does not holds for LFP using a simple cardinality argument.

Theorem 2 (Failure of BDT for LFP) *Beth's Definability Theorem does not hold for LFP.*

Our proof shows that there is an infinite set Δ which implicitly defines P for which there is not explicit definition. One may ask whether BDT holds for LFP with the restriction to finite implicit definitions. However, BDT fails for LFP even in this restricted form. This second result follows from the already known failure of BDT for LFP with finite models semantics (see [Ferreira 2007, Seção 2.3.1]). Ian Hodkinson showed how the failure of BDT for LFP with finite models semantics can be inferred from results of Gurevich and Shelah. We then showed that Hodkinson's counter-example for BDT can be extended for LFP without the constraint of considering finite models only.

Lemma 1 (Strong Failure of BDT for LFP) *There is a single formula of LFP which implicitly defines a symbol P for which there is no explicit definition in LFP.*

Besides the obvious use of minimal objects in LFP, namely, the use of a least fixed-point, in the Section 4, another relation, due to van Benthem, between LFP and the minimal models we have defined in the previous section will be presented. In the following section, we concentrate on the model theory of McCarthy's Circumscription and Lifschitz's NATs.

3. Circumscription and NATs

McCarthy's Circumscription arises in the 80's as one of the logical approaches for Artificial Intelligence problems, mainly for problems involving reasoning over incomplete information. Circumscription allows default reasoning assuming that the objects which have a determined property are only those necessary to satisfy the problem description. Further information that more objects also have that property forces the review of the assumptions introduced by the circumscription of the theory describing the underlying problem. This behavior gives a non-monotonic character to Circumscription.

The circumscription of a predicate symbol P in an FO sentence $A(P,\overline{Z})$ with the predicates $\overline{Z} = Z_1,\ldots,Z_n$ varying corresponds to the Second-Order (SO) sentence

$$Circ[A(P,\overline{Z});P;\overline{Z}] := A(P,\overline{Z}) \land \forall P'\forall\overline{Z}(A(P',\overline{Z'}) \to \neg(P' \subsetneq P)). \quad (2)$$

$P' \subsetneq P$ means $\forall\overline{x}(P'(\overline{x}) \to P(\overline{x}))$. The models of $Circ[A(P,\overline{Z});P;\overline{Z}]$ are exactly the $P;\overline{Z}$-minimal models of $A(P,\overline{Z})$ (see [Lifschitz 1994]).

In 1987, Schlipf studied model-theoretical questions regarding Circumscription, such as how difficult it is to decide whether a circumscribed formula (or finite theory) has a countable model or not (see [Ferreira 2007, Seção 3.4]). Among the results of Schlipf, is that it was showed that some circumscribed finite theories do not have a countable model. This shows that an analog to DLST does not hold for Circumscription. In [Ferreira 2007, Seção 3.4], we also gave an alternative proof of the failure of DLST.

As we have seen, the circumscription of an FO formula $A(P, \overline{Z})$ is an SO formula $Circ[A(P, \overline{Z}); P; \overline{Z}]$. Once SO does not have a complete finitary deductive system, which means that the set of SO theorems is not even semi-decidable, an interesting fragment of Circumscription for Computer Science is composed by those $Circ[A(P, \overline{Z}); P; \overline{Z}]$ which have an FO equivalent. This is the same as studying the classes of minimal models of FO formulas which can be axiomatized by FO theories. We call a class of models Δ-*elementary* if it can be axiomatized by a (possibly infinite) set of FO sentences. A Δ-elementary class of models is *elementary* if it can be axiomatized by a single FO sentence. We have stated and proved the following very interesting theorem.

Theorem 3 (Δ-Elementary Classes of Minimal Models Are Elementary) *If a class of $P; \overline{Z}$-minimal models of an FO sentence is Δ-elementary, then it is elementary.*

The importance of Theorem 3 is that, whenever a circumscribed theory can be represented in FO, for which we have a complete deductive system, such circumscribed theory can be represented by a single FO sentence. It follows that the set of logical consequences for the circumscribed theory can be recursively enumerable.

Another important consequence of Theorem 3 arises in Definability Theory. When the circumscription of a finite theory has an FO equivalent and implicitly defines the circumscribed predicate, it follows from BDT that there is an explicit definition for that circumscribed predicate. We then proved that, in such cases, there exists a special explicit definition for P that, together with the initial theory $A(P, \overline{Z})$, axiomatizes the class of $P; \overline{Z}$-minimal models of $A(P, \overline{Z})$.

Theorem 4 (Explicit Definition for Minimal Predicates) *If $Circ[A(P, \overline{Z}); P; \overline{Z}]$ is logically equivalent to an FO theory (that is, axiomatizes a Δ-elementary class of minimal models) and implicitly defines the circumscribed predicate P, then there exists an explicit definition $\forall \overline{x}(P(\overline{x}) \leftrightarrow \psi)$ for P, such that $Circ[A(P, \overline{Z}); P; \overline{Z}]$ is logically equivalent to $A(P, \overline{Z}) \wedge \forall \overline{x}(P(\overline{x}) \leftrightarrow \psi)$.*

As we saw, a circumscribed theory is an SO sentence like $Circ[A(P, \overline{Z}); P; \overline{Z}]$ in which $A(P, \overline{Z})$ is a first-order sentence. In [Lifschitz 1995], Lifschitz introduce the Nested Abnormality Theories (NATs), which allows the nesting of the $Circ$ operator. In NATs, the formula $A(P, \overline{Z})$ may contain occurrences of the $Circ$ operator inside. Lifschitz argues that some problems can be better described using NATs [Lifschitz 1995]. We investigated the expressive power of NATs and demonstrated that NATs' expressive power is similar to Second-Order Logic one in the following sense:

Theorem 5 (NATs' Expressive Power) *NATs has the same expressive power as Second-Order Logic,* modulo *additional symbols. That is, for each SO sentence ϕ in a symbol set S' there is a logically equivalent NAT T in a extended symbol set $S' \cup S$ whith the same $S' \cup S$-models. In particular, has the same logical consequences. Moreover, the size of the NAT T and the time needed to construct it are linear in the size of ϕ.*

In the next section, we present the MIN(FO) Logic [van Benthem 2005] of van Benthem, which was proved equivalent in expressive power to LFP [van Benthem 2005], and our extension of MIN(FO): the MIN Logic.

4. The MIN Logic

In [van Benthem 2005], van Benthem introduces a class of syntactically defined FO formulas called PIA-Conditions. A PIA-condition is a formula with the form $\forall \overline{x}(\phi(P,\overline{x}) \rightarrow P(\overline{x}))$, where P is a predicate symbol and $\phi(P,\overline{x})$ is an FO formula where all occurrences of P are positive (that is, occur under the scope of an even number of negation symbols) and $\overline{x}$ is a tuple o FO variables whose length is equal to the arity of P. Beside this, van Benthem defines a semantic property of some formulas, in particular PIA-conditions: the Intersection Property (IP). A formula $\phi(P)$ has the IP for some predicate symbol P if the class $\mathbb{C}$ of its models is such that, for any set $\mathbb{I}$ of structures in $\mathbb{C}$ on the same domain A and which give the same interpretations to the symbols of the language, except for P, the structure $\mathfrak{I}$ with domain A obtained interpreting $P^{\mathfrak{I}}$ as the intersection of all $P^{\mathfrak{A}}$, $\mathfrak{A} \in \mathbb{I}$, and interpreting the remaining symbols as the other structures in $\mathbb{I}$ do, is also a model of $\phi(P)$ ($\mathfrak{I} \in \mathbb{C}$). We say that a set of sentences has the IP if the class of its models has the property explained above. van Benthem showed in [van Benthem 2005] that any set of sentences which has the IP is logically equivalent to a set of PIA-conditions.

The IP for a predicate symbol P implies that, once the interpretation of the other symbols of the language is fixed, there is a least interpretation for P in each structure, namely, the intersection of all interpretations for P. These are the interpretations for P given by the $P;\emptyset$-minimal models (we write $\emptyset$ for the empty sequence) of the PIA-condition. van Benthem suggests creating a syntactic construct allowing the definition of a predicate interpreted as the least possible extension for the predicate symbol P in a PIA-condition. van Benthem introduces the MIN(FO) logic which extends FO with formulas of the form $[MIN P \bullet \psi(P)](\overline{t})$, where $\psi(P)$ is a PIA-condition, $\overline{t}$ is a tuple of terms whose length is equal to the arity of P. This formula is satisfied by a model $\mathfrak{A}$ in the cases that $\overline{t}^{\mathfrak{A}}$ belongs to the minimal predicate which satisfies the PIA-condition $\psi(P)$. van Benthem showed that MIN(FO) is equivalent in expressive power to LFP.

However, there may be the case that a formula does not have a least predicate. In face of this fact, in [Ferreira and Martins 2006], we suggested two extensions to MIN(FO): the MIN^{i} and MIN^{u} Logics. The MIN^{i} (respectively MIN^{u}) is obtained by allowing formulas like $[MIN^{i} P \bullet \phi(P)](\overline{t})$ (resp. $[MIN^{u} P \bullet \phi(P)](\overline{t})$) where $\phi(P)$ is any formula. A model $\mathfrak{A}$ satisfies $[MIN^{i} P \bullet \phi(P)](\overline{t})$ (resp. $[MIN^{u} P \bullet \phi(P)](\overline{t})$) iff $\overline{t}^{\mathfrak{A}}$ belongs to the intersection (resp. the union) of the predicates $\mathbf{P}$, where $(\mathfrak{A}, \mathbf{P})$ is a $P;\emptyset$-minimal model of $\phi(P)$. We have demonstrated that MIN^{i} and MIN^{u} are equivalent in expressive power (see [Ferreira and Martins 2006, Ferreira 2007]).

Theorem 6 (Expressiveness Equivalence Between MIN^{i} and MIN^{u}) *The* MIN^{i} *logic is equivalent to* MIN^{u} *logic in expressive power, that is, each sentence in* MIN^{i} *has an equivalent in* MIN^{u}*, and vice-versa.*

We continued in the study of the expressiveness of MIN^{i} and MIN^{u} logics and proved that MIN^{u} (and also MIN^{i}) has the same expressive power of Second-Order Logic (see [Ferreira and Martins 2006, Ferreira 2007]).

Theorem 7 (Expressiveness Equivalence of MIN^u, MIN^i and SO) *Any Second-Order sentence has equivalent in* MIN^u *and* MIN^i*, and vice-versa.*

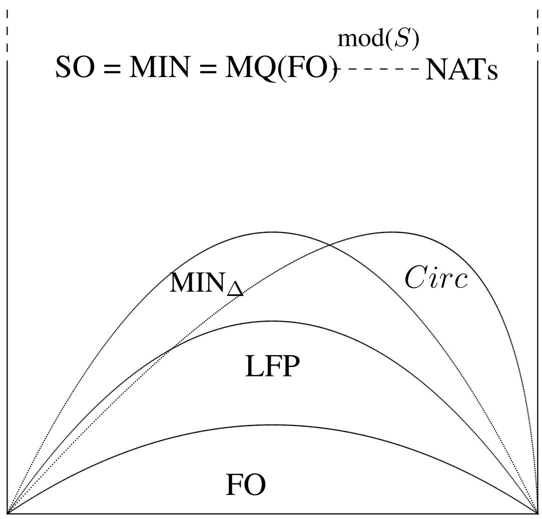
5. Conclusions

We presented the main consequences of our research on the minimality concept, particularly on minimal models, to Computer Science. We focused on two important aspects of model theory: expressiveness and definability. Our results divide into three main topics: Least Fixed-Point Logic, Circumscription and the MIN Logics. We stated and proved several theorems for these logics as presented in the previous sections. We showed a general version of Löwenheim-Skolem Theorem for the Least Fixed-Point Logic, and gave an alternative proof for the failure of a version of this theorem to Circumscription. The expressiveness equivalence of MIN and SO also implies the failure of DLST for MIN.

We demonstrated that, if Circumscription has an equivalent First-Order theory, then it is equivalent to a single First-Order sentence. We also showed that when the circumscription of a theory Γ has a First-Order equivalent and implicitly defines the circumscribed predicate, Circumscription can be replaced with Γ plus a special explicit definition for P. We gave a proof of the failure of Beth's Definability Theorem for the Least Fixed-Point Logic and used a result of Hodkinson to give a stronger version of this theorem.

We introduced the MIN^i and MIN^u extensions to van Benthem's MIN(FO) logic and proved the expressiveness equivalence between MIN^i and MIN^u. We also demonstrated the expressiveness equivalence of these logics and Second-Order Logic.

The theorems and lemmas explicitly stated above belong to the main contributions of our work in [Ferreira 2007]. Besides all these results, and also based on them, we constructed an expressiveness map of all the logics treated in [Ferreira 2007]:



In the picture above, $\text{MIN}_\triangle$ is a fragment of MIN which we proposed and investigated in [Ferreira 2007] (see [Ferreira and Martins 2007] also). We demonstrated that $\text{MIN}_\triangle$ is strictly more expressive than LFP and less expressive than Second-Order Logic. It has nice properties, like Löwenheim-Skolem Theorem. MQ(FO) (see the picture above) is another possibility which we proposed (see [Ferreira 2007, Ferreira and Martins 2007]) to deal with minimal predicates using quantifiers instead of predicate-like expressions, as the MIN logics do. The regions in the picture (even the narrow region of LFP which does not belong to $Circ$) represent distinct expressiveness levels.

The theorems in the previous sections represent the main contributions of [Ferreira 2007]. "Minimalidade e Hierarquia de Expressividade" [Ferreira 2007] can be found at `http://www.lia.ufc.br/%7efran/TeseFerreira.pdf`.

During his Master's Course, the first author was advised by Prof[a]. Ana Teresa Martins (the second author), and his research activities were partially supported by CNPq and FUNCAP. The research activities of the second author are partially supported by CNPq(PQ) and CAPES(PROCAD).

References

Beth, E. W. (1953). On Padoa's method in the theory of definitions. *Indag. Math.*, 15.

Dawar, A. and Gurevich, Y. (2002). Fixed-point logics. *Bull. Symb. Logic*, 8(1):65–88.

Ebbinghaus, H.-D., Flum, J., and Thomas, W. (1994). *Mathematical Logic*. Springer-Verlag, New York, NY.

Ferreira, F. M. (2007). Modelos Minimais e Hierarquia de Expressividade. Master's thesis, Universidade Federal do Ceará. URL: *www.lia.ufc.br/∼fran/TeseFerreira.pdf*.

Ferreira, F. M. and Martins, A. T. (2006). The Predicate-Minimizing Logic MIN. In Sichman, J. S., editor, *Lecture Notes in Artificial Intelligence: Proceedings of IB-ERAMIA/SBIA 2006*, volume 4140, pages 581–591. Springer-Verlag, Berlin.

Ferreira, F. M. and Martins, A. T. (2007). On Minimal Models. *Logic Journal of the IGPL*, 15(5-6):503–526.

Lifschitz, V. (1994). Circumscription. In Gabbay, D. M., Hogger, C. J., and Robinson, J. A., editors, *Handbook of Logic in Artificial Intelligence and Logic Programming-Nonmonotonic Reasoning and Uncertain Reasoning(Volume 3)*, pages 297–352. Clarendon Press, Oxford.

Lifschitz, V. (1995). Nested abnormality theories. *Artif. Intell.*, 74(2):351–365.

Lloyd, J. W. (1987). *Foundations of Logic Programming, 2nd Edition*. Springer.

McCarthy, J. (1986). Applications of circumscription to formalizing common-sense knowledge. *Artif. Intell.*, 28(1).

Padoa, A. (1900). Essai d'une théorie algébrique des nombres entiers, precedé d'une introduction logique a une théorie deductive qelconque. In *Bibliothèque Du Congrès Int. de Philos.*, volume 3, pages 118–123.

Schmidt, D. A. (1986). *Denotational Semantics*. Allyn and Bacon, Massachusetts.

Scott, D. S. (1982). Domains for denotational semantics. In *Proc. 9th Colloquium on Automata, Languages and Programming*, pages 577–613, London, UK. Springer-Verlag.

van Benthem, J. (2005). Minimal predicates, fixed-points, and definability. *J. Symbolic Logic*, 70(3):696–712.