[VG88] A. Van Gelder. Negation as failure using tight derivations for general logic programs. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 149–176. Morgan Kaufmann, 1988.

[VG89] A. Van Gelder The alternating fixpoint of logic programs with negation. *Proceedings of the Eighth ACM Symposium on Principles of Database Systems*, 1-10, 1989.

[VRKSS90] J. Vaghani, K. Ramamohananrao, D. Kemp, Z. Somogyi, and P. Stuckey. Design overview of the Aditi deductive database system. Tech. Report, Dept. of Computer Science, Univ. of Melbourne, July 1990.

[VRS88] A. Van Gelder, K. Ross, and J.S. Schlipf. Unfounded sets and well-founded semantics for general logic programs. In *Proc.* 7^{th} *Symposium on Principles of Database Systems*, pages 221–230, 1988.

[Wey80] R.W. Weyrauch. Prologomena to a theory of mechanized formal reasoning. *Artificial Intelligence*, 13(1 and 2):133–170, April 1980.

[Win80] T. Winograd. Extended inference modes in reasoning by computer systems. *Artificial Intelligence*, 13(1 and 2):5–26, April 1980.

[Sch87] J.S. Schlipf. Decidability and definability with circumscription. *Intl. Annals of Pure and Applied Logic*, 35(2):173–191, 1987.

[Sch88] J.S. Schlipf. When is closed world reasoning tractable? In *Proc. of the 3rd International Symposium on Methodologies for Intelligent Systems(ISMIS-88)*, 485-494, 1988.

[Sch90] J.S. Schlipf. The expressive powers of the logic programming semantics. In *Proc. of the Ninth Conference on Principles of Database Systems (PODS-90)*, 196-204, 1990.

[Sel90] B. Selman. *Tractable Defeasible Reasoning.* PhD thesis, Computer Science Department, University of Toronto, 1990.

[She88] J.C. Shepherdson. Negation in logic programming. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 19–88. Morgan Kaufman Pub., 1988.

[Shv90] G. Shvarts Autoepistemic modal logics *Proceedings TARK-1990*, Ed. R. Parikh, pp. 97-109, Morgan Kaufmann, publishers, 1990

[Smy78] M. Smyth Power domains. *Journal of Computer and System Sciences*, 16(1), pp. 23-36, 1978.

[Sta68] R. Stalnaker. A theory of conditionals. *Studies in Logical Theory, American Philosophical Quarterly Monograph Series, No. 2*, 1968.

[TG91] S. Tsur and N. Garrison. LDL User's Guide, Edition 1 MCC Technical Report Number: STP-LD-295-91, Aug. 91.

[TS86] H. Tamaki and T. Sato OLD resolution with tabulation. In: *Proceedings of the Third International Conference on Logic Programming*, 84-98, 1986.

[Tou84] D.S. Touretzky. *The Mathematics of Inheritance Systems.* Morgan Kaufmann, Los Altos, CA. 1986.

[Tru91] M. Truszczyński. Embedding default logic into modal non-monotonic logics. In A. Nerode, W. Marek, and V.S. Subrahmanian, editors, *Logic Programming and Non-monotonic Reasoning: Proceedings of the First International Workshop*, pages 151–165. MIT Press, 1991.

[Rei87] R. Reiter. Nonmonotonic reasoning. *Annual Review of Computer Science*, 2:147–186, 1987.

[Rei90] R. Reiter. On asking what a database knows. In J. Lloyd, editor, *Computational Logic: Symposium Proceedings*. Springer-Verlag, 1990.

[Rob65] J.A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1), 23-41, Jan. 1965.

[Ros89] K. Ross. Well-founded semantics for disjunctive logic programs. In *Proc. of the first International Conference on Deductive and Object Oriented Databases*, 352–369, Kyoto, Japan, Dec. 4-6,1989.

[SI92] K. Satoh and N. Iwayama. A query evaluation method for abductive logic programming. In *Proceedings of the Joint International Conference and Symposium on Logic Programming*, pages 671–685, 1992.

[SK91] B. Selman and H. Kautz. Knowledge compilation using Horn approximations. In *Proceedings of AAAI-91*, 904–909, AT&T Bell Laboratories, Murray Hill, NJ 07974, 1991.

[SL88] V . S . Subrahmanian and J. Lu. Protected completions of first order general logic programs. *Journal of Automated Reasoning*, 6(2), 147–172, June 1990.

[SL89] B. Selman and H.J. Levesque. The tractability of path-based inheritance. In *Proceedings IJCAI-89*, pages 1140–1145, Detroit, MI, 1989.

[SL90] B. Selman and H.J. Levesque. Abductive and default reasoning: A computational core. In *Proc. of the Eighth National Conference on Artificial Intelligence*, 343-348, 1990.

[SNV92] V.S. Subrahmanian, D. Nau, and C. Vago. Wfs + branch and bound = stable models. Technical report, Dept. of Computer Science, University of Maryland at College Park, 1992.

[San85] E. Sandewall. An approach to the frame problem and its implementation. In *Machine Intelligence 7*, pages 195–204. Edinburgh University Press, 1985.

[Prz91b] T. Przymusinski. Stable semantics for disjunctive programs. *New Generation Computing*, 9:401–42, 1991.

[RC81] R. Reiter and G. Criscuolo. On interacting defaults. In *Proceedings of IJCAI-81*, pages 270–276, 1981.

[RD87] R. Reiter and J. de Kleer. Formal foundations for assumption-based truth maintenance systems: preliminary report. In Proceedings of AAAI 87, 183–188, Seattle, Washington, July, 1987.

[RLM89] A. Rajasekar, J. Lobo, and J. Minker. Weak generalized closed world assumption. *Journal of Automated Reasoning*, 5:293–307, 1989.

[RSS92] R. Ramakrishnan, D. Srivastava and S. Sudarshan CORAL—Control, Relations and Logic *Proceedings of the 18th International Conference on Very Large Databases*, L.-Y. Yuan, editor, 238-250, Vancouver, Canada, 1992.

[RT88] K.A. Ross and R.W. Topor. Inferring negative information from disjunctive databases. *Journal of Automated Reasoning*, 4(2):397–424, December 1988.

[Rap71] B. Raphael. The frame problem in problem solving systems. In N. Findler and B. Meltzer, editors, *Artificial Intelligence and Heuristic Programming*, pages 159–169, New York, 1971. American Elsevier.

[Rei78a] R. Reiter. On Closed World Data Bases. In *Logic and Data Bases*, eds. H. Gallaire and J. Minker, 55–76, Plenum, New York, 1978.

[Rei78b] R. Reiter. On reasoning by default. In *Proceedings of TINLAP-2, Theoretical Issues in Natural Language Processing-2*, pages 210–218, University of Illinois at Urbana-Champaign, 1978.

[Rei80a] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1 and 2):81–132, April 1980.

[Rei82] R. Reiter. Circumscription implies predicate completion (sometimes). *Proc. Amer. Assoc. for Art. Intell. National Conference*, pages 418–420, 1982.

[PR90] Y. Peng and J. Reggia *Abductive inference methods for diagnostic problem solving.* Springer-Verlag, 1990.

[PS92] C.H. Papadimitriou and M. Sideri. On finding extensions of default theories. In *International Conference on Database Theory: Lecture Notes in Computer Science 646, Springer Verlag*, pages 276–281, 1992.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, Los Altos, California, 1988.

[Per86] D. Perlis. On the consistency of commonsense reasoning. *Computational Intelligence*, 2:180–190, 1986.

[Per87] D. Perlis. Circumscribing with sets. *Artificial Intelligence*, 31:201–211, 1987.

[Per88] D. Perlis. Autocircumscription. *Artificial Intelligence*, 36(2):223–236, 1988.

[Phi91] G. Phipps. Glue Manual, Version 1.0. Technical Report, Dept. of Computer Science, Stanford University, Feb. 1991.

[Prz88] T.C. Przymusinski. On the declarative semantics of deductive databases and logic programming. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 193–216. Morgan Kaufmann Pub., 1988.

[Prz89] T. C. Przymusinski. An algorithm to compute circumscription. *Artificial Intelligence*, 38:49–73, 1989.

[Prz89a] T. C. Przymusinski. The well-founded semantics coincides with the three-valued stable semantics. *Fundamenta Informaticae*, 1989.

[Prz90] T. Przymusinski. Stationary semantics for disjunctive logic programs and deductive databases. In S. Debray and M. Hermenegildo, editors, *Proc. of the North American Conference on Logic Programming*, 40-59, Austin, Texas, Oct. 1990.

[Prz91a] T. C. Przymusinski *Autoepistemic logic of closed beliefs and logic programming.* In: *Proceedings of the First International Workshop on Logic Programming and nonmonotonic Reasoning.* A. Nerode, W. Marek, and V.S. Subrahmanian, editors. MIT Press, 1991 3-20.

[Min88] J. Minker. Perspectives in deductive databases. *Journal of Logic Programming*, 5:33–60, 1988.

[Mins74] M. Minsky. A framework for representing knowledge. Technical Report 306, MIT Artificial Intelligence Laboratory, 1974. Reprinted without appendix in The Psychology of Computer vision, P. Winston (ed.), 177–211, McGraw-Hill (1975).

[Mins91] M. Minsky. Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI Magazine*, 12(2):34–51, 1991.

[Min91] J. Minker An overview of nonmonotonic reasoning and logic programming. University of Maryland Technical Report, UMIACS TR-91-112, CS-TR-2736, 31pp., August 1991.

[Moo85a] R. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25(1):75–94, 1985.

[NNS92] A. Nerode, R. Ng, and V.S. Subrahmanian. Computing circumscriptive databases, part i: Theory and algorithms. *Information and Computation*. to appear.

[NS91] R. Ng and V.S. Subrahmanian. Relating dempster-shafer to stable semantics. In *International Symposium on Logic Programming*, pages 551–566, 1991.

[NS92] R. Ng and V.S. Subrahmanian. Dempster-shafer logic programs and stable semantics. 1992. Submitted.

[Nie92] I.N.F. Niemela. On the decidability and complexity of autoepistemic reasoning. *Fundamenta Informaticae*, 17:117–155, 1992.

[PAA91] L.M. Pereira, J.N. Aparico, and J.J. Alferes. Counterfactual reasoning based on revising assumptions. In *International Symposium on Logic Programming*, pages 566–577, 1991.

[PGA] D. Poole, R. Goebel, and R. Aleliunas. Theorist: A logical system for defaults and diagnosis. In N. Cercone and G. McCalla, editors, *The Knowledge Frontier*, pages 331–352. Springer, 1987.

[PM86] D. Perlis and J. Minker. Completeness results for circumscription. *Artificial Intelligence*, 28(1):29–42, 1986.

[MT89] W. Marek and M. Truszczyński. Relating autoepistemic and default logic. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning (KR-89)*, pages 276–288, Toronto, Ontario, 1989.

[MT91] V. Marek and M. Truszczynski. Autoepistemic logic. *Journal of ACM*, 38(3), 588-619, July 1991.

[MT92] V. Marek and M. Truszczynski. More on modal aspects of default logic. *Fundamenta Informaticae*, 17:99–116, 1992.

[MT93] V. Marek and M. Truszczynski. *Nonmonotonic Logic – Context-Dependent Reasonings* Springer Verlag, 1993

[MZ82] J. Minker and G. Zanon. An extension to linear resolution with selection function. *Information Processing Letters*, 14(3):191–194, June 1982.

[McC58] J. McCarthy. Programs with common sense. *Mechanization of Thought Processes, Proceedings of the Symposium of the National Physics Laboratory*, vol. 1, 77-84. London. Reprinted in *Semantic Information Processing*, pages 403–418. The MIT Press, Cambridge, MA, 1968, 1958.

[McC77] J. McCarthy. Epistemological problems of artificial intelligence. In *Pro. Int. Joint. Conf. on Art. Intell.*, pages 223–227, Cambridge, MA, 1977.

[McC80] J. McCarthy. Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[McC91] J. McCarthy. *Artificial Intelligence and Mathematical Theory of Computation, Papers in Honor of John McCarthy* ed. V. Lifschitz, Academic Press, 1991.

[Min82] J. Minker. On indefinite databases and the closed world assumption. In *Proceedings of 6th Conference on Automated Deduction*, pages 292–308, New York, 1982.

[Min88] J. Minker. (editor) Foundations of Deductive Databases and Logic Programming. Morgan Kaufmann, 1988 .

[MH69] J. McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[MLR91] J. Minker, J. Lobo, and A. Rajasekar. Circumscription and Disjunctive Logic Programming. In *Artificial Intelligence and Mathematical Theory of Computation, Papers in Honor of John McCarthy*, 281–305, Academic Press, 1991.

[MNR90] W. Marek, A. Nerode, and J. Remmel. A theory of non-monotonic rule systems, part i (preliminary version). In *Proc. 1990 Intl. Symp. on Logic in Computer Science, IEEE*, 1990. Annals of Math. and AI. (to appear).

[MNR92] V. Marek, A. Nerode, and J.B. Remmel. The stable models of a predicate logic program. In *Proceedings of the Joint International Conference and Symposium on Logic Programming*, pages 446–460, 1992.

[MP85] J. Minker and D. Perlis. Computing protected circumscription. *Journal of Logic Programming*, 2(4):235–249, December 1985.

[MR90] J. Minker and A. Rajasekar. A fixpoint semantics for disjunctive logic programs. *Journal of Logic Programming*, 9(1):45–74, July 1990.

[MR93] J. Minker and C. Ruiz. On extended disjunctive logic programs. In *Proceedings of the Seventh International Symposium on Methodologies for Intelligent Systems*, Norway, June 1993.

[MS91] W. Marek and V.S. Subrahmanian. The relationship between stable, supported, default and auto-epistemic semantics for general logic programs. *Theoretical Computer Science*, v.103(2), 365–386, Sept. 1992.

[MS92] W. Marek and V.S. Subrahmanian. The relationship between stable, supported, default and autoepistemic semantics for general logic programs. *Theoretical Computer Science*, 103:365–386, 1992.

[Lev86] H.J. Levesque. Knowledge representation and reasoning. *Annual Review of Computer Science*, 1:255–287, 1986.

[Lev90] H.J. Levesque. All I know: A study in Autoepistemic Logic. *Artificial Intelligence* 42,3/4, (1990) pp. 263-309.

[Lew73] D. Lewis. *Counterfactuals.* Oxford University Press, 1973.

[Lif85] V. Lifschitz. Computing circumscription. In *Proceedings of IJCAI-85*, pages 121–127, 1985.

[Lif86] V. Lifschitz. Pointwise circumscription: a preliminary report. In *Proc. Amer. Assoc. for Art. Intell. National Conf.*, pages 406–410, Philadelphia, 1986.

[Lif89a] V. Lifschitz. Benchmark problems for formal nonmonotonic reasoning. In *Nonmonotonic Reasoning: 2nd Intl. Workshop*, pages 202–217, 1989. (Lecture Notes in Artificial Intelligence 346).

[Lif89b] V. Lifschitz. Between circumscription and autoepistemic logic. Research Report, Stanford University, Stanford, CA (1989).

[Lif91] V. Lifschitz. Nonmonotonic databases and epistemic queries: Preliminary report. In *Proceedings of 12th International Joint Conference on Artificial Intelligence*, 381–386, Sydney, Australia, 1991.

[Llo87] J.W. Lloyd. *Foundations of Logic Programming*, Springer-Verlag, Second Edition, 1987.

[Lob90] J. Lobo. On constructive negation for disjunctive logic programs. In *Proc. North American Conference on Logic Programming*, 704–718, Austin, Texas, 1990.

[Luk90] W. Lukaszewicz *Non-monotonic reasoning* Ellis-Horwood, Chichester, West Sussex, England, 1990.

[MD80] D. McDermott and J. Doyle. Non-monotonic logic I. *Artificial Intelligence 25*, pages 41–72, 1980.

[MD82] D. McDermott and J. Doyle. Non-monotonic logic II. *Journal of the ACM*, 29, pages 33–57, 1982.

[KS90] R. Kowalski and F. Sadri. Logic programs with exceptions. In *Proc. 7th International Conference on Logic Programming, Jerusalem, Israel*, 598–613, 1990.

[Kon88] K. Konolige. On the relation between default and autoepistemic logic. *Artificial Intelligence*, 35(3):343-382, July 1988.

[Kon90] K. Konolige Errata for: On the relation between default and autoepistemic logic. *Artificial Intelligence*, 41:115, 1990.

[Kon91] K. Konolige. Quantification in autoepistemic logic. *Fundamenta Informaticae*, 15:275–300, 1991.

[Kri71] S. Kripke. Semantical considerations on modal logic. In L. Lindsay, editor, *Reference and Modality*, pages 63–72, London, UK, 1971. Oxford University Press.

[Kun89] K. Kunen. Signed dependencies in logic programs. *Journal of Logic Programming*, 7, 231-245, 1989.

[LM92] D. Lehmann and M. Magidor. What does a conditional knowledge base entail? *Artificial Intelligence*, 55(1), 1-60, May 1992.

[LMR92] J. Lobo, J. Minker, and A. Rajasekar. *Foundations of Disjunctive Logic Programming.* MIT Press, 1992.

[LS89] F. Lin and Y. Shoham. Argument systems: A uniform basis for nonmonotonic reasoning. In *Proceedings of 1st Intl. Conference on Principles of Knowledge Representation and Reasoning*, pages 245–255, Stanford University, Stanford, CA, 1989.

[LS90] F. Lin and Y. Shoham. Epistemic semantics for fixed-points nonmonotonic logics. In R. Parikh, editor, *Theoretical Aspects of Reasoning and Knowledge: Proc. of the Third Conf.*, pages 111–120, Stanford University, Stanford, CA, 1990.

[LS92] J. Lobo and V.S. Subrahmanian. Relating minimal models and pre-requisite-free normal defaults. *Information Processing Letters*, 44:129–133, 1992.

[Lev84] H.J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23(2):155–212, 1984.

[Ino90] K. Inoue. An abductive procedure for the CMS/ATMS. In *Proc. European Conference on Artificial Intelligence*, ECAI 90 International Workshop on Truth Maintenance, Stockholm, Springer Verlag Lecture Notes in Computer Science, 1990.

[Ino91a] K. Inoue. Hypothetical reasoning in logic programs. Technical Report ICOT 607, Tokyo, 1991.

[Ino91b] K. Inoue. Extended logic programs with default assumptions. In *Proc. 8th International Conference on Logic Programming*, 490-504, Paris, 1991.

[Jer88] R.E. Jeroslow. Computation-Oriented Reductions of Predicate to Propositional Logic. Decision Support Systems, 4, 183-187, 1988.

[KK89] M. Kifer and T. Krishnaprasad. An evidence based framework for a theory of inheritance. In *Proceedings of 11th Intl. Joint Conf. on Artificial Intelligence*, pages 1093–1098, Morgan-Kaufmann, 1989.

[KKT92] A.C. Kakas, R.A. Kowalski, and F. Toni. Abductive Logic Programming. Technical Report, Dept. of Computing, Imperial College, London, Oct. 1992.

[KLM90] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 14(1):167–207, 1990.

[KM90] A.C. Kakas and P. Mancarella. Database updates through abduction. In *Proc. 16th International Conference on Very Large Databases*, VLDB 90, Brisbane, Australia, 1990.

[KM91] A.C. Kakas and P. Mancarella. Stable theories for logic programs. In *International Symposium on Logic Programming*, pages 85–100 1991.

[KPH91] S. Kraus, D. Perlis, and J. Horty. Reasoning about ignorance: A note on the bush-gorbachev problem. *Fundamenta Informaticae*, 15:325–332, 1991.

[KS89] H.A. Kautz and B. Selman. Hard problems for simple default logics. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning (KR-89)*, pages 189–197, Toronto, Ontario, 1989.

[Gro91] B. Grosof. Generalizing prioritization. In J. Allen, R. Fikes, and E. Sandewall, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pages 289–300, San Mateo, California, April 22-25 1991. Morgan Kaufman.

[HP87] L. Henschen and H. Park. Compiling the GCWA in Indefinite Deductive Databases. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 395-439. Morgan Kaufmann Pub., Washington, D.C., 1988.

[HT92] J. Horty and R. Thomason. Conditionals and artificial intelligence. *Fundamenta Informatica (forthcoming)*, 1993.

[HTT90] J. Horty, R. Thomason, and D. Touretzky. A skeptical theory of inheritance in nonmonotonic semantic networks. *Artificial Intelligence*, 42:311–348, 1990. abbreviated version appears in Proceedings of AAAI-87, Morgan Kaufmann, 358-363.

[Hay73] P.J. Hayes. Computation and deduction. In *Proceedings of the Second Symposium on Mathematical Foundations of Computer Science*, pages 105–118. Czechoslovakia: Czechoslovakian Academy of Sciences, 1973.

[Hew69] C.E. Hewitt. PLANNER: A language for proving theorems in robots. *First International Joint Conference on Artificial Intelligence*, pages 295–301, 1969.

[IKH92] K. Inoue, M. Koshimura, and R. Hasegawa. Embedding negation as failure into a model generation theorem prover. In *International Conference on Automated Deduction (CADE-11)*, pages 400–415, Saratoga Springs, NY, 1992.

[IOHN92] K. Inoue, Y. Ohta, R. Hasegawa and M. Nakashima. Bottom-Up abduction by model generation. Technical Report TR-816, ICOT, Tokyo, October 1992.

[IOHN92] K. Inoue, Y. Ohta, R. Hasegawa, and M. Nakashima. Hypothetical reasoning systems on the MGTP. Technical Report, ICOT, Tokyo (in Japanese), 1992.

[Imi87] T. Imielinski. Results on translating defaults to circumscription. *Artificial Intelligence*, 32:131–146, 1987.

[GM78] H. Gallaire and J. Minker, editors. *Logic and Databases.* Plenum Press, New York, April 1978.

[GPP86] M. Gelfond, H. Przymusinska, and T.C. Przymusinski. The extended closed world assumption and its relation to parallel circumscription. *Proc. Fifth ACM SIGACT-SIGMOD Symposium on Principle of Database Systems*, pages 133–139, 1986.

[GPP89] M. Gelfond, H. Przymusinska, and T.C. Przymusinski. On the relationship between circumscription and negation as failure. *Artificial Intelligence.*, 38(1), 75-94, 1989.

[GS90] J. Grant and V.S. Subrahmanian. Reasoning in inconsistent knowledge bases. to appear in IEEE Trans. Knowledge and Data Engineering.

[Gef89] H. Geffner. Default Reasoning: Causal and Conditional Theories. Ph.D thesis, Computer Science Department, UCLA, 1989.

[Gel87] M. Gelfond. On stratified autoepistemic theories. In *Proc. AAAI-87*, pages 207–211, 1987.

[Gel92] M. Gelfond. Logic programming and reasoning with incomplete information. Technical report, Dept of Computer Science, University of Texas at El Paso, 1992.

[Gin86] M. Ginsberg. Counterfactuals. *Artificial Intelligence*, 30:35–79, 1986.

[Gin87A] M. Ginsberg. *Readings in Nonmonotonic Reasoning.* Morgan Kaufmann, Los Altos, California, 1987.

[Gin89] M. Ginsberg. A circumscriptive theorem prover. *Artificial Intelligence*, 39:265–316, 1989.

[Gin91] M. Ginsberg. The MVL theorem proving system. *SIGART Bulletin: Special Issue on implemented knowledge representation and reasoning systems*, 2(3):57–60, 1991.

[Gir92] F. Gire. Well founded semantics and stable semantics of semi-strict programs. In *International Conference on Database Theory: Lecture Notes in Computer Science 646, Springer Verlag*, pages 261–275, 1992.

[Got91] G. Gottlob. Complexity results for nonmonotonic logics. Technical report, Institut fur Informationssysteme, Technishe Universitat, Wien, 1991.

[FM92b] J.A. Fernandez and J. Minker. Disjunctive deductive databases. In *International Conference on Logic Programming and Automated Reasoning: Lecture Notes in AI 624, Springer Verlag*, pages 332–356, 1992.

[FM93] J.A. Alberto Fernández and J. Minker. Theory and algorithms for disjunctive deductive databases. *Programmirovanie Journal*, 1993. Invited Paper by the Academy of Sciences of Russia (To appear).

[Fah79] S.E. Fahlman. *Netl: A System for Representing and Using Real-World Knowledge.* M.I.T. Press, 1979.

[Fit85] M. Fitting. A Kripke-Kleene semantics for logic programs. *Journal of Logic Programming*, 2(4):295–312, 1985.

[Fit91] M.C. Fitting. Well-founded semantics, generalized. In *International Symposium on Logic Programming*, pages 71–84, 1991.

[GL88] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proceedings of the 5th Logic Programming Symposium*, pages 1070–1080, Association for Logic Programming, MIT Press, Cambridge, Mass, 1988.

[GL90] M. Gelfond and V. Lifschitz Logic programs with classical negation. *Logic Programming: Proceedings of the Seventh International Conference*, D.H.D. Warren and P. Szeredi, eds., 579-597, 1990.

[GL91] M. Gelfond and V. Lifschitz. Classical negation in logic programming and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[GL92] M. Gelfond and V. Lifschitz. Representing actions in extended logic programming. In *Proceedings of the Joint International Conference and Symposium on Logic Programming*, pages 559–573, 1992.

[GLPT91] M. Gelfond, V. Lifschitz, H. Przymusińska, and M. Truszczyński. Disjunctive defaults. In J. Allen, R. Fikes, and E. Sandewall, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pages 230–237, San Mateo, California, April 22-25 1991. Morgan Kaufman.

[EKP91] D. Etherington, S. Kraus, and D. Perlis. Nonmonotonicity and the scope of reasoning. *Artificial Intelligence*, 52:221–261, 1991.

[EMR85] D. Etherington, R. Mercer, and R. Reiter. On the adequacy of predicate circumscription for closed world reasoning. *Computational Intelligence 1*, pages 11–15, 1985.

[ER83] D.W. Etherington and R. Reiter. On inheritance hierarchies with exceptions. In *Proceedings of American Association for Artificial Intelligence at National Conference*, pages 104–108, 1983.

[Esh88] K. Eshghi. Abductive planning with event calculus. In *Proc. 5th International Conference and Symposium on Logic Programming*, 562-579, Seattle, Washington, 1988.

[Esh90] K. Eshghi. Diagnoses as stable models. In *Proc. 1st International Workshop on Principles of Diagnoses*, Menlo Park, CA, 1990.

[Eva89] C. Evans Negation-as-failure as an approach to the Hanks and McDermott problem. *Proceedings of the Second International Conference on Artificial Intelligence*, 1969.

[FLM91] M. Freund, D. Lehmann and P. Morris. Rationality, transitivity and contraposition. *Artificial Intelligence*, 52, 191-203, 1991.

[FLMS93] J.A. Fernandez, J. Lobo, J. Minker, and V.S. Subrahmanian. Disjunctive LP + integrity constraints = stable model semantics. In *Annals of Mathematics and AI* 8, No. 3-4 (1993)

[FM91a] J.A. Fernandez and J. Minker. Bottom-up evaluation of hierarchical disjunctive deductive databases. In *Proc. International Conference on Logic Programming*, 660–675, Paris, 1991. MIT Press.

[FM91b] J.A. Fernandez and J. Minker. Computing perfect models of disjunctive stratified databases. In *Proc. ILPS '91 Workshop on Disjunctive Logic Programs*, 110-117, eds. D. Loveland, J. Lobo, and A. Rajasekar, San Diego, CA, Oct. 1991.

[FM92a] J.A. Fernandez and J. Minker. Semantics of disjunctive deductive databases. In *International Conference on Database Theory: Lecture Notes in Computer Science 646, Springer Verlag*, pages 21–50, 1992.

[DeK86] J. de Kleer. An assumption-based TMS. *Artificial Intelligence*, 32, 1986.

[Del88] J. Delgrande. An approach to default reasoning based on a first-order conditional logic: revised report. *Artificial Intelligence*, 36:63–90, 1988.

[Dem68] A.P. Dempster. A Generalization of Bayesian Inference. J. of the Royal Statistical Society, Series B, 30, 205-247, 1968.

[Dix91] J. Dix. Classifying semantics of logic programs. In A. Nerode, W. Marek, and V.S. Subrahmanian, editors, *Logic Programming and Non-monotonic Reasoning: Proceedings of the First International Workshop*, pages 166–180. MIT Press, 1991.

[Dix92] J. Dix. Classifying semantics of disjunctive logic program. In *Proceedings of the Joint International Conference and Symposium on Logic Programming*, pages 798–812, 1992.

[Doy79] J. Doyle. A truth maintenance system. *Artificial Intelligence*, 12:231–272, l979.

[Doy82] J. Doyle. Some theories of reasoned assumptions: An essay in rational psychology. Technical Report, Department of Computer Science, Carnegie-Mellon University, 1982.

[Doy83] J. Doyle. The ins and outs of reason maintenance. In Proceedings of Eighth IJCAI, pp. 349-351, 1983.

[Dung92] P.M. Dung. On the relations between stable and well-founded semantics of logic programs. *Theoretical Computer Science: Logic, semantics and theory of programming*, 105:7–26, 1992.

[EDP90] J. Elgot-Drapkin and D. Perlis. Reasoning situated in time i: basic concepts. *J. of Experimental and Theoretical Artificial Intelligence*, 2:75–98, 1990.

[EK76] M.H. van Emden and R.A. Kowalski The semantics of predicate logic as a programming language *Journal of the ACM*, 23(4), pp. 733-742, 1976.

[EK88] K. Eshghi and R.A. Kowalski. Abduction through deduction. Technical Report, Dept. of Computing, Imperial College, London, 1988.

[EK89] K. Eshghi and R.A. Kowalski. Abduction compared with negation by failure. In *Proc. 6th International Conference on Logic Programming*, 234-255, Lisbon, 1989.

[CH82] A. Chandra and D. Harel. Structure and complexity of relational queries. *Journal of Computer System Sciences*, 25:99–128, 1982.

[CKPR73] A. Colmerauer, H. Kanoui, R. Pasero, and P. Roussel. Un systeme de communication homme-machine en francais. Technical report, Groupe de Intelligence Artificielle Universitae de Aix-Marseille II, Marseille, 1973.

[CL90] M. Cadoli and M. Lenzerini. The complexity of closed world reasoning and circumscription. *Proceedings of the AAAI*, pages 550–555, 1990 Also Technical report, Universita di Roma "La Sapienza", Dipartimento di Informatica e Sistemistica, Roma, Italy, 1990.

[CS89] J. Chomicki and V.S. Subrahmanian. Generalized closed world assumption is π_2^0 complete. Technical Report TR-89-036, Computer Science Department, University of North Carolina, 1989. *Information Processing Letters.*, 34:289–291, 1990.

[CS92] M. Cadoli and M. Schaerf. A survey on complexity results for non-monotonic logics. Technical report, Universita di Roma "La Sapienza", Dipartiment o di Informatica e Sistemistica, Roma, Italy, 1992.

[CW92] W. Chen and D.S. Warren. A goal oriented approach to computing well founded semantics. In *Proceedings of the Joint International Conference and Symposium on Logic Programming*, pages 589–606, 1992.

[Cha88] D. Chan. Constructive negation based on the completed databases. In R.A. Kowalski and K.A. Bowen, editors, *Proc. 5^{th} International Conference and Symposium on Logic Programming*, pages 111–125, Seattle, Washington, August 15-19, 1988.

[Cla78] K.L. Clark. Negation as failure. In *Logic and Databases*, eds. H. Gallaire and J. Minker, 293-322, Plenum, New York, 1978.

[DD92] M. Denecker and D. De Schreye. Sldnfa: an abductive procedure for normal abductive programs. In *Proceedings of the Joint International Conference and Symposium on Logic Programming*, pages 686–702, 1992.

[DLNPS90] F.M. Donini, M. Lenzerini, D. Nardi, F. Pirri, M. Schaerf Nonmonotonic reasoning *Artificial Intelligence Review*, vol. 4, 163-210, 1990.

[BS89] H. Blair and V.S. Subrahmanian. Paraconsistent Logic Programming. *Theoretical Computer Science*, 68:135–154, 1989.

[BS91] C. Baral and V. S. Subrahmanian. Dualities between alternative semantics of logic programs and nonmonotonic formalisms. In A. Nerode, W. Marek, and V.S. Subrahmanian, editors, *Logic Programming and Non-monotonic Reasoning: Proceedings of the First International Workshop*, pages 69–86. MIT Press, 1991.

[BS92] C. Baral and V.S. Subrahmanian. Stable and extension class theory for logic programs and default logics. *Journal of Automated Reasoning*, 8:345–366, 1992.

[Bar91] C. Baral. *Issues in Knowledge Representation: Semantics and Knowledge Combination*. PhD thesis, University of Maryland, College Park, Maryland 20742, 1991.

[Bar92] C. Baral. Generalized negation as failure and semantics of normal disjunctive logic programs. In *International Conference on Logic Programming and Automated Reasoning: Lecture Notes in AI 624, Springer Verlag*, pages 309–319, 1992.

[Bel77a] N. Belnap. How a computer should think. In G. Ryle, editor, *Contemporary Aspects of Philosophy*, pages 30–56. Oriel Press, 1977.

[Bel77b] N. Belnap. A useful four-valued logic. In J.M. Dunn and G. Epstein, editors, *Modern Uses of Multiple-valued Logic*, pages 8–37. D. Reidel, 1977.

[Bid91] N. Bidoit. Negation in rule-based database languages: a survey. *Theoretical Computer Science*, 78, 3-83, 1991.

[Bon92] P.A. Bonatti. Autoepistemic logics as a unifying framework for the semantics of logic programs. In *Proceedings of the Joint International Conference and Symposium on Logic Programming*, pages 417–430, 1992.

[Bre89] Ph. Bresnard. *An Introduction to Default Logic*. Springer-Verlag, 1989.

[Bre91] G. Brewka. *Nonmonotonic Reasoning: Logical Foundations of Common Sense*. Cambridge University Press, Cambridge, 1991.

[BF88] N. Bidoit and C. Froidevaux. General logical databases and programs: Default logic semantics and stratification. *Journal of Information and Computation*, v.91, no.1, 15–54, March, 1991.

[BH86] N. Bidoit and R. Hull. Positivism vs. minimalism in deductive databases. *Proc. ACM SIGACT-SIGMOD Symposium on the Principles of Database Systems*, pages 123–132, 1986.

[BKMS91] C. Baral, S. Kraus, J. Minker, and V.S. Subrahmanian. Combining knowledge bases consisting of first order theories. *Computational Intelligence* , 8(1), 45–71, March 1992.

[BLM90a] C. Baral, J. Lobo, and J. Minker. Generalized disjunctive well-founded semantics for logic programs: Declarative semantics. In *Proc. of Fifth International Symposium on Methodologies for Intelligent Systems*, 465–473, Knoxville, TN, 1990.

[BLM90b] C. Baral, J. Lobo, and J. Minker. Generalized disjunctive well-founded semantics for logic programs: Procedural semantics. In *Proc. of Fifth International Symposium on Methodologies for Intelligent Systems*, 456–464, Knoxville, TN, 1990.

[BLM90c] C. Baral, J. Lobo, and J. Minker. Generalized well-founded semantics for logic programs. In M. E. Stickel, editor, *Proc. of Tenth International Conference on Automated Deduction*, pages 102–116, Kaiserslautern, FRG, July, 1990. Springer-Verlag.

[BLM91] C. Baral, J. Lobo, and J. Minker. WF^3: A semantics for negation in normal disjunctive logic programs. In *Proc. of sixth International Symposium on Methodologies for Intelligent Systems*, 459–468, Charlotte, NC, 1991.

[BNNS91] C. Bell, A. Nerode, R. Ng, and V.S. Subrahmanian. Computation and implementation of non-monotonic deductive databases. Technical report, Dept. of Computer Science, University of Maryland at College Park, 1991.

[BR92] L. Bertossi and R. Reiter. Circumscription and Generic Mathematical Objects. Technical Report, University of Toronto, 1992.

[BS85] G. Bossu and P. Siegel. Saturation, nonmonotonic reasoning and the closed-world assumption. *Artificial Intelligence*, 25(1):13–63, 1985 January 1985.

the '90s. I am sure that we will have expanded our knowledge of nonmonotonic reasoning and logic programming and will have answers to some, if not all of the questions I have posed.

6 Acknowledgements

I would like to express my appreciation to Chitta Baral, José Alberto Fernández, Parke Godfrey, Jeff Horty, Yuan Liu, Sarit Kraus, Jorge Lobo, Don Perlis, Shekhar Pradhan, Arcot Rajasekar, Carolina Ruiz and V.S. Subrahmanian for their many suggestions concerning the paper. Errors of commission or omission are due to the author.

Support for work on the paper was provided by the Air Force Office of Scientific Research under grant number 91-0350, the Army Research Office under grant number DAAL03-88-K-0087 and the National Science Foundation under grant number IRI-8916059

References

[AB88] K.R. Apt and H.A. Blair. Arithmetic classification of perfect models of stratified programs. *R.A. Conference and Symposium on Logic Programming*, pages 766–779, 1988.

[AB90] K.R. Apt and M. Bezem Acyclic programs. *Logic Programming: Proceedings of the Seventh International Conference*, D.H.D. Warren an P. Szeredi, eds. MIT Press, 617–633, 1990.

[ABW88] K.R. Apt, H.A. Blair, and A. Walker. Towards a theory of declarative knowledge. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 89–148. Morgan Kaufmann Pub., Washington, D.C., 1988.

[AP92] J.J. Alferes and L.M. Pereira. On logic program semantics with two kinds of negation. In *Proceedings of the Joint International Conference and Symposium on Logic Programming*, pages 574–588, 1992.

[BE89] A. Borgida and D.W. Etherington. Hierarchical knowledge bases and efficient disjunction. In *Proceedings of the First International Conference on Principle of Knowledge Representation and Reasoning (KR-89)*, pages 33–43, Toronto, Ontario, Canada, 1989.

problems with the alternative theories. In this connection, Lifschitz has specified a list of benchmark problems whose solution in nonmonotonic reasoning would be very productive [Lif89a]. However, work on large applications will lead to insight into scientific issues that must be addressed, tell us how practical our theories are, and potentially be of benefit to the military and to industry. The foundations of nonmonotonic reasoning have been sufficiently developed to warrant implementations.

Unless we do some, if not all, of the above, the complaints that Minsky and others have - that the work is nice from a logician's view, but has little relevance to artificial intelligence - will remain. As evidenced by the work cited in this paper, we know a great deal about nonmonotonic reasoning, tractability, and logic programming, but we have not yet assimilated what to do with this knowledge.

The theories that we develop are important in that they provide a semantics for nonmonotonic reasoning. A common mistaken critique of the use of logic or nonmonotonic reasoning is that one cannot use heuristics or deal with context. This critique is not relevant as it has been shown that whether or not we are dealing with a form of logic, heuristics can be used (e.g. which rule to select, or which proof path should be explored first). It also has nothing to do with whether or not we can use bottom-up, top-down, or inside-out approaches to search. All of these are compatible with each approach to nonmonotonic reasoning. The introduction of context to an application is also compatible with logic and nonmonotonic reasoning. How we implement the logic theory is another issue.

In an invited paper that I gave at the **Principles of Database Systems** [Min88], I stated,

> It seems to me that the fields of databases, logic programming, deductive databases, artificial intelligence, and expert systems will move towards one another. Formalisms and techniques developed in each of these areas will assist the others. Science builds upon theories. Theories developed for deductive databases and logic programming will, therefore, be built upon to further developments in the above subjects.

The comments concerning artificial intelligence refer, of course, to nonmonotonic reasoning. If artificial intelligence is to be a mature science, formal theories are essential. Tractability results are also important, but it would seem that average case analyses might be even more important. It is clear from the results sketched in this paper that the fields of logic programming and nonmonotonic reasoning are on solid scientific ground and that significant results have been developed in both areas and in their relationships. I look forward to looking back at the field at the end of the decade of

3. Given a particular theory, what are tractable classes of problems for which it is effective?

4. Given large problems that may be either tractable, or intractable, what are effective techniques that may be used to implement the theories?

5. Are there approximations to the semantics of nonmonotonic reasoning that are sound and may be more tractable than the full theory?

6. How does one select from alterative semantics for logic programs that contain different rules of default and classical negation?

7. What additional characterizations of nonmonotonic reasoning can be developed and how do they help in understanding or in developing new semantics [Dix91, Dix92, KLM90, LM92, FLM91]?

Although partial answers to some of these questions exist, more work is needed. I believe that the above issues should be given more attention than that of developing additional theories of nonmonotonic reasoning or theories of negation in logic programming. I must admit that I am one of the perpetrators of the problem. Nevertheless, we cannot continue to develop new theories with new procedures without addressing the issues I have listed above. As noted by Marek and Subrahmanian [MS92], a thorough study of the interrelationships between differing formalisms for treating negation in logic programming is necessary as there are now too many formalisms. A continuation of the development of the logical characterization of general properties may assist in this process. It may provide tools to evaluate existing nonmonotonic reasoning formalisms and may lead to the development of new theories that satisfy the properties.

Perhaps it will be possible to develop an automatic decision-maker to determine which nonmonotonic formalism should be used in a given context; however, I have my doubts as to the feasibility of such a system. We must also show practical problems where the theories may be applied. At the present time we have only addressed "toy" problems, with the notable exception of the work by Kowalski [KS90] on legal reasoning in the British naturalization laws and the VLSI design and other work on legal systems developed as part of Japan's Fifth Generation Project [IOHN92]. Although no papers have been published on the subject, I understand that the Federal Bureau of Investigation is implementing a very large *Prolog* program that uses the nonmonotonic features of the language. I have no objection to toy problems. Indeed, we need more toy problems as they elucidate some of the logic to make it nonmonotonic. Minsky's argument that logic as traditionally understood is monotonic is correct, but the theories described in this paper use classical logic as a basis and extend it to be nonmonotonic. That they can become inconsistent can be handled as described in the following references. Belnap [Bel77a, Bel77b] has done fundamental work in dealing with inconsistent systems of logic. Some work has also been done by Baral, Kraus, Minker and Subrahmanian [BKMS91], and Grant and Subrahmanian [GS90] to handle inconsistent nonmonotonic theories. Step-logic, developed by Elgot-Drapkin and Perlis [EDP90], is another approach to dealing with nonmonotonic inconsistent theories. Blair and Subrahmanian [BS89] show how to reason formally in systems containing inconsistencies. Kifer and Krishnaprasad [KK89] develop a foundation for inheritance hierarchies based on these logics of inconsistency. However, more work is needed in dealing with inconsistent nonmonotonic theories.

As stated above, the field of nonmonotonic reasoning has made impressive steps in developing alternative theories, in showing some of the relationships among them, and in finding computational bounds on the various methods. Unfortunately, most of the bounds show that nonmonotonic reasoning is not tractable. As noted by Levesque [Lev86], "...it remains to be seen how the computational promise of nonmonotonic reasoning can be correctly realized." That there are alternative theories is not, I believe, an impediment, since some of the theories solve problems that other theories cannot solve. There does not appear to be one "all embracing theory" that will handle everything, although some consider that the ultimate aim is to develop such a theory. See Lin and Shoham [LS89], and Marek, Nerode and Remmel [MNR90] for attempts to develop a unifying framework. It is not clear that an all embracing theory can be developed that will capture all of nonmonotonic reasoning. In attempting to develop such a theory, we may, however, learn a great deal.

In a certain sense we have still not solved the problem that Minsky posed. We have developed a large number of theories, but there has been little work done that provides guidelines as to which of these theories should be used and under what circumstances. It seems to me that if nonmonotonic reasoning is to become a reality, we must be able to provide answers to the following questions:

1. What are the precise relationships among the various theories of nonmonotonic reasoning?

2. What are the conditions under which a particular nonmonotonic theory should be used?

5 Summary

Having reviewed the literature in logic programming and nonmonotonic reasoning, I have been impressed by the depth and maturity of the work. It is also impressive that most of the research, following the seminal 1980 **Artificial Intelligence Journal**, has taken place since 1986, which is the year of a workshop on deductive databases and logic programming, that I organized [Min88]. I would like to believe that the workshop and the book that presented the important papers from the workshop, has had some influence on the direction of the research. The research indicates several mature theories including circumscription, default reasoning, autoepistemic logic, and negation in logic programming and disjunctive logic programming. Not only have these theories been explicated formally, but there exist several results with respect to their tractability. Furthermore, the relationships between the theories have started to develop.

A fundamental theme that I detect is that implementation of the alternative nonmonotonic theories have been achieved mostly through logic programs or disjunctive logic programs with some exceptions, for example, the implementation of Ginsberg's theorem prover, MVL [Gin89, Gin91]. For other work on implementation of nonmonotonic theories see the the description of the *Theorist* system of Poole, Goebel and Aleliunas [PGA]. The theories that have been developed also provide mechanisms to handle the default rules that are implied in Minsky's frame paper [Mins74]. If one looks at the problems that have been proposed for solution that are nonmonotonic, there are many that are function-free. If this is the case, then one can use techniques from deductive databases to solve these problems. Work in deductive databases has matured to the point where we may expect such systems to become widely available. There are currently several systems that have been developed in laboratories and universities [VRKSS90, RSS92, Phi91, TG91]. The Bull Corporation in France is developing a deductive database system for the market place that is expected to be announced in December 1993. Other developments related to implementation of deductive or disjunctive deductive databases have been cited earlier in this paper (see [BNNS91, FM91a].

In the introduction I presented Minsky's arguments against using logic for commonsense reasoning that appeared in the appendix to his 1974 paper. I then presented the various approaches to nonmonotonic reasoning. Have these theories overcome Minsky's objections? In a certain sense they have. They formalize commonsense reasoning to a certain extent. They have modified first-order been taken by Dix [Dix91] to characterize the semantics of a program according to their properties. He suggests that the selection of a method be based upon general properties of nonmonotonic inference operations in the spirit of Kraus, Lehmann and Megidor [KLM90]. He suggests that the decision as to the semantics to be used be based on general principles, two of which he calls *cumulativity* and *rationality.* He has compared different semantics according to these properties and others for normal logic programs.

Lehmann and his colleagues have extended the work of [KLM90]. In [FLM91] they compare the rule of *rational monotonicity* of [KLM90] and different rules expressing some weak forms of *transitivity* and *contraposition.* They present four weak forms of *transitivity* that, in preferential logic, are equivalent to *contraposition.* They are able to clarify some points about circumscription and show, for example, that the relations defined by circumscription do not satisfy, in general, *contraposition.* In [LM92] they study, in inference systems, the following question: given a knowledge base **KB** what are the assertions that are entailed by **KB**? They argue that any reasonable nonmonotonic inference procedure should produce a set of consequences that satisfy properties like *cautious monotonicity* and *rationality* (among others) and they denote by *rational* any extension of **KB** that satisfies them. The desired set of consequences of **KB** is the minimal rational extension of **KB** (with respect to a particular preferential order). Of interest is that in the propositional case, this particular rational extension is computationally tractable, that is, it is decidable in polynomial time.

As a continuation of his earlier work, Dix [Dix92] deals with disjunctive theories. He introduces additional interesting principles to compare semantics: *modularity, relevance* and the *principle of partial evaluation.* The satisfaction of these properties is related to the complexity of the semantics. He shows that these properties are not all satisfied by some existing semantics (e.g. the stationary semantics [Prz90]). He defines additional semantics for normal disjunctive theories that have many if not all of these so-called desired properties.

It is not clear, however, why the criteria suggested by Dix or by Lehmann and his colleagues are ones that should be used in selecting a semantics for normal logic or normal disjunctive logic programs. However, the fact that satisfying some of these properties leads to tractable theories is of considerable interest. More work is required here. Although we have come a long way, there is still a long way to go.

$GCWA$ in disjunctive theories is highly complex, it is Π_2^0-complete. Cadoli and Lenzerini [CL90] provide complexity results of closed world reasoning and circumscription for the propositional case. They provide complexity results for the CWA, $GCWA$, and $EGCWA$ for the following theories: Horn (at most one positive literal per clause); dual-Horn (at most one negative literal per clause); Krom (at most two literals per clause-either positive or negative); Definite (exactly one positive literal per clause); Horn-Krom (Horn and Krom); dual-Horn-Krom (dual-Horn and Krom); HornKrom$^-$ (HornKrom with no negative clauses having 2 literals); 2-positive-Krom (exactly two positive literals, and no negative literal, per clause).

Selman and Kautz [SK91] propose to approximate disjunctive theories by bounding the set of models from below and from above by Horn theories. If Σ is the set of clauses in the theory, and $\mathcal{M}(\Sigma)$ is the set of minimal models of Σ, the lower and upper bounds yield the approximations:

$$\mathcal{M}(\Sigma_{lb}) \subseteq \mathcal{M}(\Sigma) \subseteq \mathcal{M}(\Sigma_{ub})$$

The use of approximate answers is an interesting approach, as it can lead to fast answers, and the user can decide whether or not it is desirable to obtain the complete answer. Selman and Kautz note that the notion of Horn upper-bound can be viewed as an abstraction of the original theory, because the upper-bound is a more general, weaker theory. They show that the notion of abstraction as introduced for disjunctive theories by Borgida and Etherington [BE89] corresponds to the least Horn upper-bound of the theories they consider. Borgida and Etherington also discuss some disjunctive theories in which one can compute reasonably.

For a comprehensive survey and extensive bibliography on complexity results for nonmonotonic logics, see Cadoli and Schaerf [CS92].

We have described a large number of different theories of negation for normal logic programs and normal disjunctive logic programs. What should one make of them? What should be done when a system is needed that will handle normal or normal disjunctive logic programs? Which theory should be used? No insights have been developed except, perhaps, that each developer of a theory probably favors his own. Since my students and I have developed several theories of negation, we favor each one on different days. There also are no guidelines for the difficulty in computing answers to problems, and in some cases there are no procedures to compute answers to the theory. One promising approach has

For a summary of complexity results on abductive reasoning see the excellent survey by Cadoli and Schaerf [CS92].

A formal definition of *abductive logic programming* is given by Kakas and Mancarella [KM90] to be a triple $\{P, \Gamma, \mathcal{I}\}$, where P is a normal logic program, Γ is a set of *abducible predicates*, and I is a set of *integrity constraints*. A *generalized stable model* of $\{P, \Gamma, \mathcal{I}\}$ is defined as the stable model of $P \cup E$ which satisfies $\mathcal{I}$, where E is any set of ground atoms with predicates from Γ (E plays the role of Δ in the previous definition of abductive reasoning). In abductive logic programming the presence of integrity constraints restricts, even further, the possibilities for the ground atoms in E. This is a stronger condition than simple consistency.

Satoh and Iwayama [SI92] propose a procedure for abduction that is correct for any abductive framework. If the procedure succeeds there is a set of hypotheses which satisfies a query and if it finitely fails there is no such set. They guarantee correctness as they adopt a forward evaluation of rules and check consistency of "implicit deletion." Because of the forward evaluation of rules they can handle arbitrary integrity constraints. Denecker and De Schreye [DD92] develop a family of extensions of $SLDNF$ resolution for normal abductive programs. They can handle non-ground abductive goals. A completion semantics is given and the soundness and completeness of the procedures has been proven. A framework is provided of abductive procedures, in which a number of parameters can be set, to fit the abductive procedure to the application considered. In a personal conversation with Inoue, he states that he and his colleagues are relating abductive logic programming with disjunctive logic programming. He transforms an abductive logic program to a disjunctive logic program whose stable models correspond to the generalized stable models of the abductive logic program.

4.4 Complexity and Properties of Logic Programs

Apt and Blair [AB88] show that for all n, there exists a logic program P such that the standard model of P is Σ_n^0-complete. The same result applies to default and autoepistemic logics, as shown by Apt and Blair, and also Marek and Truszczyński [MT89].

The close relationship between circumscription and logic programming is important, as we know how to compute in the case of logic programs. However, the computational complexity is high if the circumscriptive theory is disjunctive. Chomicki and Subrahmanian [CS89] show that computing the shows that a 3-valued autoepistemic logic provides a unifying framework for most of the major semantics of normal logic programs. The framework extends to disjunctive logic programs and induces the natural counterparts of the well-founded and the stable model semantics. The resulting semantics are different from the stationary semantics [Prz90], the generalized disjunctive well-founded semantics [BLM90a], and its extension, WF^3 [BLM91].

4.3 Abductive Logic Programming

Abductive reasoning is an important part of commonsense reasoning. Abduction is the process of finding explanations for observations in a given theory. There have been two major approaches to abduction: one based on logic as defined by Selman and Levesque [SL90] and described in Kakas, Kowalski and Toni [KKT92], and the other based on set-covering methods as given in the book by Peng and Reggia [PR90]. We discuss only the logic-based approach.

The formal definition of *abductive reasoning* is that given a set of sentences T (a theory), and a sentence G (an observation), the abductive task is to find a set of sentences Δ (abductive explanation for G) such that

(1) $T \cup \Delta \models G$,

(2) $T \cup \Delta$ is consistent.

(3) Δ is minimal with respect to set inclusion [SL90, KKT92, CS92].

A comprehensive survey and critical overview of the extension of logic programming to perform abductive reasoning (referred to as *abductive logic programming*) is given in [KKT92] together with an extensive bibliography on abductive reasoning. They outline the framework of abduction and its applications to default reasoning; and introduce an argumentation theoretic approach to the use of abduction as an interpretation of negation-as-failure. They show that abduction has strong links to extended disjunctive logic programming. Abduction is shown to generalize negation-as-failure to include not only negative but also positive hypotheses, and to include general integrity constraints. They show that abductive logic programming is related to the justification-based truth maintenance system of Doyle [Doy79] and the assumption-based truth maintenance system of de Kleer [DeK86].

stratified and stable theories. They use model trees, as described originally in [FM91a].

Przymusinski [Prz91a] introduces the stable model semantics for disjunctive logic programs and disjunctive deductive databases. This generalizes the stable model semantics for normal logic programs. Depending upon whether only total (2-valued) or all partial (3-valued) models are used, one obtains the *disjunctive stable semantics* or the *partial disjunctive stable semantics*, respectively. He shows the following results: for normal programs, disjunctive (respectively partial disjunctive) stable semantics coincides with stable (respectively, partial stable) semantics; for normal programs, partial disjunctive stable models coincide with well-founded semantics; for locally stratified disjunctive programs, both (total and partial) disjunctive semantics coincide with the *perfect model semantics*; the work extends to disjunctive programs with classical negation; after translation of a program P into a suitable autoepistemic theory $\overline{P}$ the disjunctive (respectively, partial disjunctive) stable semantics of P coincides with the autoepistemic (respectively, 3-valued autoepistemic) semantics of $\overline{P}$. Alferes and Pereira [AP92] define a parameterizable schema to encompass and characterize a diversity of proposed semantics for extended logic programs. By adjusting their parameters they can specify several semantics using two kinds of negation: stationary [Prz90], extended stable semantics [Prz91b] and well-founded semantics [VRS88]. Minker and Ruiz [MR93] develop general techniques for dealing with extended disjunctive logic programs and extend the model, fixpoint and proof theories of an arbitrary semantics of normal disjunctive logic programs to cover the case of extended programs. Illustrations of these techniques are given for stable models, disjunctive well-founded and stationary semantics. They summarize results on the declarative complexity of extended logic programs and the algorithmic complexity of the proof procedures analyzed.

Inoue, Koshimura and Hasegawa [IKH92] develop a system that computes answers to function-free extended logic programs and extended disjunctive deductive databases. They use bottom-up incremental, backtrack-free computation of the minimal models of positive disjunctive programs, together with integrity constraints over beliefs and disbeliefs.

Gelfond and Lifschitz [GL92] extend the work of Eshghi and Kowalski [EK88], Evans [Eva89] and Apt and Bezem [AB90] on representing actions in logic programming languages with negation-as-failure. Extended logic programs are used for this purpose. The method is applicable to temporal projection problems with incomplete information, as well as reasoning about the past. Bonatti [Bon92] minimal models. Minker and Rajasekar [MR90] describe how one can compute in this theory. The $GCWA$ is needed for disjunctive theories since Reiter [Rei78a] has shown that the CWA is inconsistent with respect to disjunctive theories. A weaker form of the $GCWA$, the *weak generalized closed world assumption (WGCWA)* was developed by Rajasekar, Lobo and Minker [RLM89]. The complexity of the $WGCWA$ is the same as that of the CWA. The $WGCWA$ is equivalent to the *disjunctive database rule (DDR)*, developed by Ross and Topor [RT88]. Lobo [Lob90] has extended constructive negation to apply to normal disjunctive logic programs.

There have been a number of theories for normal disjunctive logic programs. Each of these theories is different and one does not imply the other. These are: *generalized disjunctive well founded semantics (GDWFS)* developed by Baral, Lobo and Minker [BLM90a, BLM90b]; *WF^3 semantics* by Baral, Lobo and Minker [BLM91]; *WFS for disjunctive theories* by Ross [Ros89]; *DWFS, disjunctive well-founded semantics* by Baral [Bar92]; and *stationary semantics* by Przymusinski [Prz90]. A procedural semantics exists for the $GDWFS$ and WF^3. The procedures are based on *linear resolution with selection function for indefinite clauses (SLI).* This inference system was developed by Minker and Zanon [MZ82], and renamed by Minker and Rajasekar [MR90]. It requires the use of ancestry resolution and factoring, and hence is more complex than SLD resolution which is used for Horn theories. Lobo, Minker and Rajasekar [LMR92] have written a research monograph that provides the basic foundation for disjunctive logic programming. They provide a formal treatment of the three semantics for disjunctive logic programs: fixpoint, model theoretic and proof theory. They describe two forms of default negation, the *generalized closed world assumption (GCWA)* [Min82] and variations on that assumption, and the *weak generalized closed world assumption (WGCWA)* [RLM89, RT88].

Baral [Bar91] gives a classification of various iterative fixpoint semantics of normal and disjunctive logic programs. He develops a unifying framework that captures the generalized well-founded semantics, well-founded semantics, generalized disjunctive well-founded semantics and various other fixpoint semantics of disjunctive logic programs. It is not clear, however, which of the these theories should be used, and under what circumstances. For additional work in this area, see Bidoit and Froidevaux [BF88], Bidoit and Hull [BH86], and Bossu and Siegel [BS85].

Fernández and Minker [FM92a, FM92b, FM93] develop methods to deal with disjunctive deductive databases. They present the theory and algorithms to obtain answers to queries for disjunctive, Gelfond [Gel92] expands the syntax and semantics of logic programs and deductive databases to allow for the correct representation of incomplete information in the presence of multiple extensions. He expands the language of logic programs with classical negation, epistemic disjunction, and negation-as-failure by a new modal operator **K** (where for the set of rules T and formulas F, **K**F denotes "F is known to be *true* by a reasoner with a set of premises T"). Sets of rules in the extended language are called *epistemic specifications*. Epistemic theories differ from autoepistemic theories in their use of the different connectives.

Bell, Nerode, Ng and Subrahmanian [BNNS91] focus their attention on the computation and implementation of nonmonotonic deductive databases. Their approach differs from the conventional approaches. They base their method on using linear programming-based automated inference as introduced by Jeroslow [Jer88] instead of using the Robinson resolution method [Rob65]. They concentrate their development on compilers which move as much of the deduction to compile-time as possible. They compare, implement, and experiment with linear constraints corresponding to several uses of "classical" negation in logic programs.

For a detailed description of negation in logic programs, current up to 1988, see Shepherdson [She88]. For a thorough survey of negation in rule-base database languages, see Bidoit [Bid91]. The Bidoit survey focuses on the problems of defining the declarative semantics of logic programs with negation. Negation based on fixpoint techniques, three-valued logic and nonmonotonic logics are presented for positive logic programs, (locally) stratified logic programs and unstratifiable logic programs.

4.2 Disjunctive Logic Programming

In the area of disjunctive logic programming, there are also several approaches to negation: *the generalized closed world assumption (GCWA)* developed by Minker [Min82] and a derivative of this theory, the *extended closed world assumption* [GPP86] and [HP87] that apply to disjunctive logic programs. The model theoretic definition of the $GCWA$ states that one can conclude the negation of a ground atom if it is false in all minimal Herbrand models. The proof theoretic definition states that one can conclude **not** a if for any disjunction of atoms, B, if $a \vee B$ is provable from the program, then B is provable. The proof theoretic and model theoretic definitions are equivalent [Min82]. The concept of minimal models is closely related to McCarthy's circumscription, which also deals with classes that were minimal with respect to Smyth's power domain ordering [Smy78] were selected. The well-founded semantics then corresponds to a class that is minimal with respect to Hoare's power domain ordering: the dual of Smyth's ordering. The result also suggests how to define a well-founded semantics for default logic in such a way that the dualities that hold for logic programs continue to hold for default theories. They further show how the same techniques may be applied to "strong" autoepistemic logic: the logic of strong expansions as proposed by Marek and Truszczyński [MT89]. Przymusinski [Prz89a] shows that the well-founded semantics coincides with the three-valued stable model semantics. Dung [Dung92] shows that stable semantics can be defined in the same way as well-founded semantics based on the basic notion of unfounded sets and thus stable semantics can be considered to be a "two-valued well-founded semantics." He further provides an axiomatic characterization of stable and well-founded semantics of logic programs, called *strong completion*. Like Clark's [Cla78] completion, *strong completion* can be interpreted in either two-valued or three-value logics. He shows that two-valued *strong completion* specifies the stable model semantics while three-valued *strong completion* specifies the well-founded semantics. The call-consistency condition, as developed by Kunen [Kun89], is sufficient for the existence of at least one stable model. Gire [Gir92] exhibits a condition on the syntax of logic programs with negation, the *semi-strictness property* (referred to as the call-consistency property by Kunen [Kun89]), which assures the equivalence of the well-founded semantics and the stable semantics in the sense that the well-founded model of an effectively semi-strict program P is total if-and-only-if P has a unique stable model. This class of programs strictly contains the effectively stratifiable programs. Subrahmanian, Nau and Vago [SNV92] develop techniques to compute the well-founded model of a logic program. Experiments with their implementation compare favorably with the standard alternating fixpoint computation [BS91, VG89]. They compute the stable models of a deductive database by first computing the well-founded semantics and then use a branch-and-bound strategy to compute the stable models.

Gelfond and Lifschitz [GL90] expand the class of logic programs to include *logical negation* ($\neg$), in addition to negation-as-failure. They base their semantics on stable models. They argue that some facts of commonsense reasoning can be represented more easily when classical negation is available. Computationally, classical negation can be eliminated from extended programs by a simple transformation that changes a classically negated atom $\neg p$ to a new atom p'.

semantics. Their work provides a means to generalize autoepistemic reasoning and applies to work in abductive reasoning. Ng and Subrahmanian [NS91] investigate Dempster-Shafer [Dem68] probabilistic logic programs and develop a declarative semantics for this class. They transform each such program into a new program whose clauses may contain nonmonotonic negations in their bodies and develop a stable semantics for the new program. Hence, the meaning of a Dempster-Shafer theory is identical to that of the transformed program as defined by the stable semantics. Fernández, Lobo, Minker and Subrahmanian [FLMS93] have shown how to compute the stable model semantics by transforming the logic program into a disjunctive logic program that contains integrity constraints. They show that the stable models of the original program are the minimal models of the transformed disjunctive logic program. In the case of deductive databases, they develop algorithms to operate over the *model tree* of the disjunctive theory so as to answer queries in the stable model semantics. The model tree is a shared structure that contains the minimal models of the disjunctive deductive database. Marek, Nerode and Remmel [MNR92] classify the Turing complexity of stable models of finite and recursive predicate logic programs. They show that up to a recursive 1-1 encoding, the sets of all stable models of a finite predicate logic program and the Π^0_1 classes (the sets of all finite branches of a recursive tree) coincide.

In their work on *stable classes*, Baral and Subrahmanian [BS92] introduce the idea of *extension classes* for default logics and of stable classes for logic programs which generalize the extension and stable model semantics, respectively. They are able to reason about inconsistent default theories, and about logic programs with inconsistent completions. They extend the results of Marek and Truszczyński [MT89] relating to logic programs and default logics. Marek and Subrahmanian [MS92] develop connections between stable and supported models. They show relationships between supported models and default logic. They develop conditions which guarantee that for every normal logic program P there exists a default theory (D, W) that relates to the supported models of P. They show further connections between supported models of a program an expansions of the program's autoepistemic translation.

The relationships that exist between stable model and well-founded semantics have been investigated by a large number of individuals. Baral and Subrahmanian [BS91] show the *duality* between stable class theory and the well-founded semantics for logic programs. In the stable class semantics, *constructive negation*. Whereas Clark, using *SLDNF* can find answers to negated ground atoms, it does not work for non-ground atoms. Chan's constructive negation solves this problem. In the case of Horn theories there is a single model, given by the unique minimal model of the program [EK76]. Theories then exist for stratified normal programs and normal programs in general. In the area of stratified programs, there is a single meaning that can be ascribed to a program, as developed by Apt, Blair and Walker [ABW88], and referred to as the *perfect model* by Przymusinski [Prz88]. In the area of normal programs that are not stratified, there are several competing theories: the theory of *stable models* developed by Gelfond and Lifschitz [GL88]; the theory of *well-founded semantics* developed by Van Gelder, Ross and Schlipf [VRS88], and elaborated upon by Przymusinski [Prz89a]; the theory of *generalized well-founded semantics* by Baral, Lobo and Minker [BLM90c] and the theory of *stable classes* developed by Baral and Subrahmanian [BS92]. As shown by Baral and Subrahmanian [BS91], the concept of stable classes is able to capture both the notion of stable models and the well-founded semantics.

There have been a large number of papers that relate to *well-founded semantics, stable semantics*, the relationships between them and the three methods of nonmonotonic reasoning described in the previous sections, and to computing in these semantics. We discuss some of these results below.

In the area of *well-founded semantics*, Fitting [Fit91] has shown how well-founded semantics extends naturally to a family of lattice-based logic programming languages. The generalization simplifies the proofs of the basic results in well-founded semantics. Chen and Warren [CW92] developed a goal-oriented method, *XOLDTNF resolution*, of computing the well-founded semantics. The inference system extends *OLDT resolution* developed by Tamaki and Sato [TS86] in that the truth value of an answer to a query may be *undefined*, due to the three-valued nature of the well-founded semantics. They avoid negative loops in the computation by negative contexts. The inference system provides a smooth interface with *Prolog*.

In the area of *stable model semantics*, Kakas and Mancarella [KM91] define a class of stable theories by treating negation in any logic program as a form of hypothesis. The definition is given in terms of a *stability property* on negative hypotheses that corresponds to negation-as-failure literals which attempt to formalize the usual understanding of a default rule of negation. The stable theories generalize stable models. They also identify a "minimal" semantics for any logic program analogous to the well-founded default and *logical negation.* There has been a large amount of work in these areas. The work relates to *normal logic programs*, *normal disjunctive logic programs*, *extended logic programs*, and *extended disjunctive logic programs.* In a normal disjunctive program there may be a positive disjunction in the head of a clause, and a conjunction of atoms and atoms preceded by a default rule of negation written as **not** in the body of a clause. In an extended disjunctive logic program there may be a disjunction of literals in the head of a clause (where a literal is an atom or the logical negation of an atom (written $\neg$), and the body may contain a conjunction of literals and literals preceded by a default rule of negation. An extended clause is written as:

$$l_1 \vee ... \vee l_k \leftarrow l_{k+1}, ..., l_m, not\ l_{m+1}, ..., not\ l_n$$

where $l_1, ..., l_n$ are literals and $k \leq m \leq n$. The explicit use of classical negation suggests the introduction of a new truth value, namely, *logical falsehood* (in contrast to *falsehood by failure*) in the semantics.

When $k = 1$ and literals in the head and body of the clause are atoms, we refer to the clause as a *normal clause.* When $k = 1$ and literals in the head and body need not be atoms, we refer to the clause as an *extended clause.* When $k > 1$ and literals in the head and body are restricted to atoms, then we refer to the clause as a *normal disjunctive clause.* When $k > 1$ and literals in the head and body are not restricted to atoms, we refer to the clause as an *extended disjunctive clause.*

The major issue with respect to nonmonotonicity is the way in which the default rule of negation is used to develop the semantics of the set of clauses that constitute the logic program. There have been a large number of different theories. This section is subdivided into several subsections. These deal with: normal and extended logic programs; disjunctive logic programs; abductive logic programs; and complexity and properties of logic programs. In each subsection the relationships between alternative logic programs and their relationships to the three major classes of nonmonotonic reasoning are described.

4.1 Normal and Extended Logic Programs

In the case of normal logic programs, theories of negation for Horn theories have been developed: the *closed world assumption (CWA)* [Rei78a], *Clark's completion theory* [Cla78], and Chan's [Cha88] using a certain type of name for the individual.

Marek and Truszczyński [MT91] systematically study properties of AEL. A major issue in AEL is to define semantics for programs with negation. Gelfond [Gel87] observed that the logic program $p \leftarrow q, \textbf{not}\ r$ can be expressed in autoepistemic logic as $q \wedge \neg \mathbf{K} r \Rightarrow p$, where **K** is the modal operator interpreted to be "is known". He showed that the theory obtained by translating the clauses of a stratified logic program has a unique expansion. [MT91] extends the work of Gelfond and shows that there is a wide class of theories (stratified theories) that possess a unique expansion. Their results imply algorithms to determine whether a theory has a unique expansion. They develop connections between autoepistemic logic and stable model semantics for logic programs and prove that the problem of existence of stable models is NP-hard.

Marek and Truszczyński [MT92] also introduce the concept of *weak extensions* and study its properties. The notion of weak extensions permits a precise description of the relationship between default and autoepistemic logics. They show that default logic with weak extensions is essentially equivalent to autoepistemic logic. It is nonmonotonic logic **KD 45** (which corresponds to Moore's autoepistemic logic). They study the notion of a set of formulas closed under a default theory and show that they correspond to stable theories and to modal logic $\mathbf{S_5}$. They further show that skeptical reasoning with sets closed under default theories is closely related with provability in $\mathbf{S_5}$. They provide complexity results.

See Gottlob [Got91] and Niemelä [Nie92] for complexity results involving expansions of default theories and autoepistemic expansions. In view of the relationships between the alternative theories, the tractability results of Kautz and Selman [KS89] and Selman [Sel90] may apply to autoepistemic theories.

4 Logic Programming

In the previous sections we have noted where the alternative nonmonotonic theories relate to logic programming. In this section we describe how logic programming incorporates the ability to handle data in a nonmonotonic fashion. Work in logic programming and nonmonotonic reasoning deals with the handling of negation. We consider two types of negation in logic programs: negation by rules of

Marek and Subrahmanian [MS91] show the relationship between supported models of normal programs and expansions of autoepistemic theories. Gelfond [Gel87] shows that general logic programs can be translated into autoepistemic theories. Gelfond and Lifschitz [GL88] show that stable model semantics is also equivalent to the translation of logic programs into autoepistemic theories as proposed by Gelfond. Lifschitz [Lif89a] shows that autoepistemic logic, stable models and introspective circumscription provide three equivalent descriptions of the meaning of propositional logic programs. Lifschitz also notes that default logic and autoepistemic logic provide more expressive possibilities that apparently have no counterpart in circumscription, except in the versions of Perlis and Lifschitz.

Przymusinski [Prz91a] notes several drawbacks of autoepistemic logic, notably some "reasonable" theories are often inconsistent in AEL; even for consistent theories AEL does not always lead to expected semantics; it insists upon completely deciding all of our beliefs; and it does not offer flexibility in terms of selecting application-dependent formalisms on which to base our beliefs. He shows that autoepistemic logics of closed beliefs of Moore coincides with autoepistemic logic of closed beliefs in which the negative introspection operator used is Reiter's CWA. He then extends autoepistemic logic to *generalized autoepistemic logic (GAEL)*, which uses Minker's generalized closed world assumption [Min82] as the basis for the negative introspection operator and demonstrates how other forms of AEL may be achieved.

Lifschitz [Lif91] brings together work by Reiter [Rei90], and Levesque [Lev84] who discuss query evaluation in databases that are treated as first-order formulas that also contain an epistemic modal operator, and work in epistemic formulas used in knowledge representation for expressing defaults. He describes a new version of the logic of grounded logic, proposed by Lin and Shoham [LS90], which is similar to the Levesque/Reiter theory of epistemic queries. Using this formalism he gives meaning to epistemic queries in the context of logic programming, and can ask, 'What does a program know?". Because Lifschitz's version of the logic of grounded knowledge contains some forms of default logic, he is able to give meaning to epistemic queries in the context of a default theory or a circumscriptive theory. Lifschitz's logic as well as the logic of Lin and Shoham is bimodal. Lifschitz's logic contains the full propositional default logic by Reiter [Rei80a].

Konolige [Kon91] discusses quantification in autoepistemic logic and proposes several different semantics, all based on the idea that having beliefs about an individual amounts to having a belief

[Kri71]. As observed by Shvarts [Shv90], AEL is one of the nonmonotonic logics that can be obtained from the approach of McDermott and Doyle [MD82].

Levesque [Lev90] generalizes Moore's notion of a stable expansion [Moo85a] to the full first-order case. He provides a semantic account of stable expansions in terms of a second modal operator $\mathcal{O}$, where $\mathcal{O}(w)$ is read as "w is all that is believed." He characterizes stable expansions as: $\mathcal{O}(w)$ is *true* exactly when all formulas that are believed form a stable expansion of w. Perlis [Per88] and Lifschitz [Lif89b] have developed variants of circumscription analogous to autoepistemic logic.

Although default logic and autoepistemic logic are seemingly different, and motivated by slightly different concerns, Konolige [Kon88, Kon90] shows that autoepistemic logic can be strengthened so that there is an equivalence between the propositional form of both logics. He gives an effective translation of default logic into autoepistemic logic and shows there is a reverse translation; every set of sentences in autoepistemic logic can be effectively rewritten as a default theory.

Marek and Truszczyński [MT89] extend the work by Konolige. They take a syntactic approach to investigating the relationships between autoepistemic and default logics. They state:

> In each logic we find three classes of objects - minimal sets closed under defaults, weak extensions, extensions for default logic, and minimal stable theories, expansions and robust expansions for autoepistemic logic - so that for a default theory (D,W), E is a minimal set closed under defaults (respectively weak extension, extension) if and only if E is the objective part of a minimal stable theory (respectively expansion, robust expansion) for the autoepistemic interpretation of (D,W). Similar results for the converse direction hold only in the case of minimal stable sets and minimal sets closed under defaults, and expansions and weak expansions. A weaker result holds for robust expansions and extensions.

Truszczyński [Tru91] develops a natural modal interpretation of defaults. He shows that under this interpretation there are whole families of modal nonmonotonic logics that accurately represent default reasoning. He applies the method to logic programs and obtains results that relate stable models to several classes of $S - expansions$. His results show that there is no single modal logic for describing default reasoning and that there exist a whole range of modal logics that can be used in the embedding as a "host" logic.

for the remaining theories. They also show that the complexity of determining if a given literal p appears in any extension of a Horn default theory or a normal unary theory is $\mathcal{O}(n)$, where n is the number of occurrences of literals in the theory, but otherwise is NP-hard. In skeptical theories where one wants to determine if a literal is in all extensions, they show that for a normal unary theory the time-complexity is $\mathcal{O}(n^2)$, where n is the number of occurrences of literals in the theory. For other theories it is co-NP.

Path-based inheritance reasoning as defined by Touretzky is NP-hard, even when restricted to acyclic unambiguous networks. Horty, Thomason and Touretzky [HTT90] identify tractable forms of defeasible reasoning. Selman and Levesque [SL89, Sel90] show that the tractability of an inheritance theory depends upon the kinds of chaining involved in the path construction. Selman [Sel90] further shows that the standard upward chaining notion of path construction leads to tractable algorithms; those definitions based on a double chaining notion lead to NP-complete algorithms.

Papadimitriou and Sideri [PS92] show that all default theories that have no cycles (in some precise sense) have an extension, which can be found efficiently. They further prove that it is NP-complete to find extensions even for default theories with no prerequisites and at most two literals per default. They also characterize precisely the complexity of finding extensions in general default theories. The problem is $\Sigma_2^p - complete$.

3.4 Modal Theories

McDermott and Doyle [MD80] introduced a modal operator M into first-order logic. If p is a sentence in first-order logic, then Mp denotes the sentence in modal logic whose intended meaning is "p is consistent with what is known," or "maybe p."

Moore [Moo85a] developed *autoepistemic logic (AEL)*, which is an improvement of the work by McDermott and Doyle. Instead of using a "possibly" modal operator, he uses a "necessarily" modal operator L. Intuitively, Lp is to be read as "I know p." Moore reconstructs nonmonotonic logic as a model of an ideally rational agent's reasoning about its own beliefs. He defines a semantics for which he shows that autoepistemic logic is sound and complete. There is a strong relationship between the two modalities, as p is possible if-and-only-if $\neg p$ is not necessary. Hence, Mp is equivalent to $\neg L\ \neg p$. Ginsberg shows the relationship between autoepistemic logic and Kripke's approach to modal logic *default theory (ddt)* is a set of disjunctive defaults. Gelfond et al. [GLPT91] show that one cannot simulate a *ddt* with a standard default theory. They note that *ddt* is a generalization of the semantics for disjunctive databases proposed by [GL91].

Baral and Subrahmanian [BS92] show that even though all default theories do not necessarily have extensions, Reiter's operator always has fixed-points over the power lattice. Such fixed-points indicate that when extensions do not exist, the original fixed-point operator may form a loop around possible extensions (sets of formulas). They refer to the class of such sets of formulas as *strict extension classes*, and define the *extension class semantics*. They also define well-founded semantics of default theories [BS91] as a particular extension class.

Etherington and Reiter [ER83] use default logic to formalize NETL-like inheritance hierarchies [Fah79]. They provide the first attempt at a semantics for such hierarchies; a provably correct inference algorithm for acyclic networks; a guarantee that the acyclic network has extensions; and a provably correct quasi-parallel inference algorithm for such networks.

Lobo and Subrahmanian [LS92] show that given any disjunctive logic program P, the minimal Herbrand models of P are in a precise 1-1 correspondence with the default logic theory Δ_P obtained by adding to P the default logic schema $\frac{:\neg A}{\neg A}$. See Imielinski [Imi87] for connections between default logic and circumscription.

Selman [Sel90] in an excellent thesis, explores three default reasoning formalisms. He obtains the first characterization of tractable forms of default reasoning. He also gives a high-level characterization of the main factors contributing to the intractability of the general reasoning. He considers the following formalisms: *model-preference defaults*, *default reasoning* ([Rei80a]), and *path-based defeasible inheritance* ([Tou84, HTT90]). In model-preference defaults, the preference ordering on models is defined by statements of the form "a model where α holds is preferable to a model where β holds." He proves that only systems with quite limited expressible power lead to tractable reasoning, e.g., $\mathcal{DH}$ and $\mathcal{DH}_a^+$, containing respectively Horn defaults and acyclic specificity-ordered Horn defaults.

Kautz and Selman [KS89] and Selman [Sel90] consider the complexity of various forms of propositional default reasoning. They consider *unary*, *disjunction-free ordered*, *ordered unary*, *disjunction-free normal*, *Horn*, and *normal unary* theories. These theories form a hierarchy and Kautz and Selman show that to find an extension in disjunction-free and unary theories is NP-hard, whereas it is $\mathcal{O}(n^3)$

state: ‘if α is *true*, and it is consistent to assume that β is *true*, then conclude that γ is *true*.

Default rules act as mappings from some incomplete theory to a more complete *extension* of the theory. An extension is a maximal set of conclusions that can be drawn from the default theory. Reiter defines a *default theory* to be a pair (D, W), where D is a set of closed default rules and W is a set of first-order sentences. Extensions are defined by a fixed point construction. For any set of first-order sentences S, define $\Gamma(S)$ to be the smallest set satisfying the following three properties:

1. $W \subset \Gamma(S)$.
2. $\Gamma(S)$ is closed under first-order logical consequence.
3. If $\frac{\alpha:\beta}{\gamma}$ is a default rule of D and $\alpha \epsilon \Gamma(S)$ and $\neg\beta \not\epsilon S$, then $\gamma \epsilon \Gamma(S)$.

Then E is defined to be an *extension* of the default theory (D, W) iff $\Gamma(E) = E$, that is, E is a fixed point of the operator Γ.

A theory consisting of general default rules does not always have an extension. A subclass consisting of default rules called *normal defaults*, and of the form, $\frac{\alpha:\beta}{\beta}$ always has an extension. Reiter develops a complete proof theory for normal defaults and shows how it interfaces with a top-down resolution theorem prover.

Reiter and Criscuolo [RC81] show that default rules may be normal when viewed in isolation, however, they can interact in ways that lead to derivations of anomalous default assumptions. Non-normal default rules are required to deal with default interactions. Handling non-normal default rules is computationally more complex than dealing with normal default rules.

Gelfond, Lifschitz, Przymusinska and Truszczyński [GLPT91] generalize Reiter's default logic to handle disjunctive information. The generalization arises because of difficulties with disjunctive information where there may be multiple extensions - one containing a sentence α and another a sentence β - and the theory with a simple extension $\alpha \vee \beta$.

A *disjunctive default* is an expression of the form

$$\frac{\alpha:\beta_1,\ldots,\beta_m}{\gamma_1|\ldots|\gamma_n}$$

where $\alpha_1, \beta_1, \ldots, \beta_m, \gamma_1, \ldots, \gamma_n (m, n \geq 0)$ are quantifier free formulas. Formula α is the prerequisite of the default, $\beta_1, \ldots, \beta_m$ are justifications, and $\gamma_1, \ldots, \gamma_n$, disjoined, are its consequents. A *disjunctive* other's knowing a given proposition P. They solve the Bush-Gorbachev problem, defined by McCarthy, to illustrate their approach.

It is important to be able to compute in circumscriptive theories. The use of logic programming is a natural computation vehicle for a large class of circumscription problems as they both deal with the concept of minimal models. In general, it will be difficult to compute in circumscriptive theories, except for those that are equivalent to *normal Horn theories* which allow a single atom in the head of a clause and literals in the body of a clause and also in the case of stratified disjunctive databases. Fernández and Minker [FM91b] develop a fixpoint operator for stratified disjunctive deductive databases that captures the *perfect models* of Przymusinski [Prz88], and is able to compute prioritized circumscription. Przymusinski [Prz89] has developed an algorithm to compute circumscription in a wide class of circumscriptive theories. Ginsburg has implemented a circumscriptive theorem prover [Gin89]. Nerode, Ng and Subrahmanian [NNS92] have developed and implemented algorithms to compute the preferred models of circumscriptive databases at compile-time using mixed integer linear programming techniques. Their method is bottom-up and permits re-use of previous computations. The method accommodates database updates.

Schlipf [Sch87] has shown that circumscription is Π^1_2-complete. In disjunctive theories, the computational complexity increases so that unless the theory is *near-Horn*, in which there are $log(n)$ non-Horn formulas, where n is the number of formulas in the theory, it will be computationally complex.

Various nonmonotonic formalisms based on *conditional entailment* have been developed by Geffner [Gef89], Pearl [Pea88], Delgrande [Del88] and by Ginsberg [Gin86]. Conditional entailment bears some similarity to circumscription in that they both induce preferences between models. The original ideas on conditional entailment were set forth by Stalnaker [Sta68] and Lewis [Lew73]. For a detailed description of the role of conditionals in artificial intelligence, see Horty and Thomason [HT92].

3.3 Default Reasoning

Default reasoning, developed by Reiter [Rei80a], is an important approach to nonmonotonic reasoning and is one of the more extensively studied formalisms of nonmonotonic reasoning. See the book by Bresnard [Bre89] on default logic. A default is a rule of the form $\frac{\alpha:\beta}{\gamma}$, whose meaning is intended to from some counterintuitive limitations concerning expectations on "counterexamples" to defaults. These limitations are not confined to circumscription, but are endemic to all nonmonotonic reasoning formalisms. Etherington, Kraus and Perlis [EKP91] develop a general approach to solve the problem that involves restricting the *scope* of nonmonotonic reasoning and show that it remedies these problems in a variety of formalisms. They refer to their approach as *scoped circumscription*. Their solution requires no modification of the underlying formalisms, and the result is semantically compatible with existing approaches. The idea of scoping is to limit the applicability of the use of defaults to restricted situations rather than to broader classes to which it may not apply. Bertossi and Reiter [BR92] show that although circumscription does not give the expected results for characterizing the concept of a generic object in the context of a formalized mathematical theory, scoped circumscription provides the right mechanism.

There have been other results about circumscription. Perlis and Minker [PM86] develop completeness results for circumscription. McCarthy's original paper [McC80] discussed only the soundness of circumscription. Reiter [Rei82] was the first to relate circumscription to logic programming. Minker and Perlis [MP85] introduced *protected circumscription* and demonstrated how one can compute in this theory with logic programs. Lu and Subrahmanian [SL88] have extended the concept of protected circumscription. Gelfond, Przymusinska and Przymusinski [GPP89] relate a propositional form of circumscription to stratified theories. (See Chandra and Harel [CH82], Apt, Blair and Walker [ABW88], and Van Gelder [VG88] for work on stratified theories.) Minker, Lobo and Rajasekar [MLR91] complement the work by Gelfond, Przymusinska and Przymusinski [GPP89] by developing a procedure to compute circumscription in disjunctive logic programs that are stratified. See, Lifschitz [Lif89a] for additional work in this area.

Perlis [Per87] argues that sets play an important role in circumscription's ability to deal in a general way with certain aspects of commonsense reasoning. He notes that sentences that intuitively one would want circumscription to prove are nonetheless not so provable without using sets. He shows that when sets are introduced, first-order circumscription handles these cases easily, obviating the need for second-order circumscription.

Kraus, Perlis and Horty [KPH91] show that one can assess another's ignorance by default using what they call *autocircumscription*, to assess our own ignorance of anything that might suggest the negation. The collected works of John McCarthy [McC91] should be read as it is fundamental to an understanding of thoughts that led to the development and foundation of nonmonotonic reasoning.

3.2 Circumscription

Circumscription has generated a great deal of interest in the nonmonotonic reasoning community. McCarthy's 1980 paper on circumscription [McC80] is a formalization of the work he described first in [McC77]. Circumscription deals with the minimization of predicates subject to restrictions expressed by predicate formulas. If A is a sentence of a first-order language containing a predicate symbol $P(x_1, \ldots, x_n)$, written $P(\overline{x})$, then the result of replacing all occurrences of P in A by the predicate expression Φ is written as $A(\Phi)$. The *circumscription* of P in $A(P)$ is the sentence schema: for all Φ

$$[A(\Phi) \wedge \forall \overline{x}(\Phi(\overline{x}) \supset P(\overline{x}))] \supset \forall \overline{x}(P(\overline{x}) \supset \Phi(\overline{x}))$$

which states that P is minimal among the predicates which make $A(\Phi)$ *true*, in the sense that the only tuples $\overline{x}$ that satisfy P are those that have to - assuming the sentence $A(P)$. McCarthy shows how a slight generalization allows circumscribing several predicates jointly.

Lifschitz [Lif86] modifies circumscription so that instead of being a single minimality condition, it becomes an "infinite conjunction" of "local" minimality conditions; each condition expresses the impossibility of changing the value of a predicate from *true* to *false* at one point. This is referred to as *pointwise circumscription*. Lifschitz then defines *prioritized circumscription* which provides for priorities between predicates. Lifschitz [Lif85] describes the concept of *parallel circumscription* and also treats prioritized circumscription. Grosof [Gro91] has generalized prioritized circumscription to a partial order of priorities.

Lifschitz [Lif85] addresses the problem of computing circumscription. He notes that circumscription is difficult to implement because its definition involves a second-order quantifier. He introduces metamathematical results that allow, in some cases, circumscription to be replaced by an equivalent first-order formula. Etherington, Mercer and Reiter [EMR85] establish results about the consistency of circumscription, showing that predicate circumscription cannot account for some kinds of default reasoning, and also provides no information about equality predicates. Perlis [Per86] shows the inadequacies of circumscription to deal with counterexamples. He has shown that circumscription suffers introduced earlier [McC77]; by Reiter [Rei80a] who introduced his theory of *default reasoning*; and by McDermott and Doyle [MD80] who used *modal logic* to handle nonmonotonicity. The second approach views logic as an object, and extends the reasoning system with meta-devices. This is explored by Weyrauch [Wey80] and Winograd [Win80].

I will not describe all of these theories as there is insufficient space to do so in this paper. I will focus on the first approach. The approach to nonmonotonic reasoning evidenced by circumscription, default reasoning, modal theories, and logic programming have led to a large literature since 1980. Building upon this work, numerous results have been obtained. I briefly review some of this work. A full treatment of all of the work is not possible here. Those of you who have a "pet" theory that is omitted, please forgive me. I must say that in preparing this paper I have been overwhelmed by the breadth and depth of the research that has been achieved since 1980. It is not possible to do justice to all of the work in this paper. I have also decided not to include a discussion of truth maintenance systems. For approaches to the semantics of these systems see [RD87, Doy79, Doy82, Doy83].

Ginsberg [Gin87A] has captured the significant developments in nonmonotonic logic up to approximately 1987 in his book **Readings in Nonmonotonic Reasoning**. This is a major source document for work up to that date. It consists of the original articles on the subject of nonmonotonic reasoning. Ginsberg ties the work together nicely with comments that are interspersed at the beginning of the book and throughout the various sections. There are several major sections: background and historical papers; formalizations of nonmonotonic reasoning; truth maintenance; applications of nonmonotonic reasoning; and an appendix by Perlis that provides an extensive bibliography on nonmonotonic reasoning. Other extensive surveys of nonmonotonic reasoning can be found in [Bre91, DLNPS90, Luk90]. As will be seen in the following sections, nonmonotonic logic has a rigorous mathematical basis. Although grounded in classical logic, it is a new discipline that extends classical logic and, in a short period of time - since 1980 has become a mature part of logic. A major text book on the subject has been written by Marek and Truszczyński [MT93]. The book emphasizes work in default logic and autoepistemic reasoning. I will concentrate on the formalizations of nonmonotonic reasoning and their relationships to logic programming. See [Rei87] for an earlier survey of the field. Books on the foundation of logic programming [Llo87] and on the foundations of disjunctive logic programming [LMR92] provide theoretical background on the semantics of logic programming including rules of

3 Nonmonotonic Reasoning

3.1 Beginning Research

The start of the field of nonmonotonic reasoning is an outgrowth of McCarthy's 1958 paper on commonsense reasoning [McC58]. The paper by Hayes [Hay73] is another important early development. The *Prolog* programming language developed by Colmerauer and his students [CKPR73] and the PLANNER language developed by Hewitt [Hew69] were the first languages to have a nonmonotonic component. The **not** operator in *Prolog*, and the **THNOT** capability in PLANNER provided default rules for answering questions about data where the facts did not appear explicitly in the program.

The formalization of the field of nonmonotonic reasoning as we know it today started approximately in 1975/1976, with papers published in the 1977-1979 time period. Two important papers, one by Reiter [Rei78a], and the other by Clark [Cla78] appeared in the book "Logic and Data Bases," edited by Gallaire and Minker [GM78]. Reiter set forth the rule of negation called the *closed world assumption, (CWA).* The *CWA* states that in Horn logic theories if we cannot prove an atom p, then we can assume **not** p. Clark related negation to the **only if** counter-part of **if** statements in a logic program. The **if-and-only-if** (**iff**) statements form a theory in which negated atoms can be proven using a full theorem prover. The importance of Clark's observation is that he showed that for ground atoms, an inference system called *SLDNF* resolution, operating on the **if** statements of logic programs was sufficient to find the ground negated atoms in the **iff** theory that can be assumed *true.* These two *rules of negation* are, I believe, the first formalizations of nonmonotonic reasoning. McCarthy first introduced his theory of *circumscription* in 1977 [McC77], and Doyle developed his *truth maintenance system* in 1979 [Doy79]. Reiter gave preliminary material on default reasoning in 1978 [Rei78b].

Nonmonotonic reasoning obtained its impetus in 1980 with the publication of an issue of the **Artificial Intelligence Journal**, devoted exclusively to nonmonotonic reasoning. In that seminal issue the initial theories of nonmonotonic logic were presented. As noted by Bobrow in his "Editor's Preface" to the journal, the approaches to nonmonotonic reasoning can be characterized broadly as falling in two different classes. The first approach extends the logic system in different ways. This is characterized in papers by McCarthy [McC80] who formalized his theory of *circumscription*,

1. "Logical" reasoning is not flexible enough to serve as a basis for thinking; I prefer to think of it as a collection of heuristic methods, effective only when applied to starkly simplified plans. The Consistency that Logic absolutely demands is not otherwise usually available – and probably not even desirable! – because consistent systems are likely to be too "weak".
2. I doubt the feasibility of representing ordinary knowledge effectively in the form of many small, independently "true" propositions.
3. The strategy of complete separation of specific knowledge from general rules of inference is much too radical. We need more direct ways for linking fragments of knowledge to advice about how they are to be used.
4. It was long believed that it was crucial to make all knowledge accessible to deduction in the form of declarative statements; but this seems less urgent as we learn ways to manipulate structural and procedural descriptions.

Minsky wrote these statements in 1974. Indeed, in 1991 [Mins91] he holds the same view and states

> This focus on well-defined problems produced many successful applications, no matter that the underlying systems were too inflexible to function well outside the domains for which they were designed. (It seems to me that this occurred because of the researcher's excessive concern with consistency and provability. Ultimately this concern would be a proper one but not in the subject's current state of immaturity.)

At the time Minsky wrote his paper [Mins74], he was commenting upon logic and noting that it was monotonic. Minsky's views are, I believe, also shared by Herbert Simon. I believe that Minsky and Simon are more interested in cognitive modeling where logic has not been shown to be useful. What are we to make of these statements made by a distinguished colleague who is among the small number of founders of the field of artificial intelligence, and supported by another distinguished founder of artificial intelligence? Have we answered their objections after these 15 to 20 years of work in nonmonotonic reasoning? Before I comment upon this, I would like to review what has been accomplished since 1974 in nonmonotonic reasoning.

place, the action has no effect upon many of the other objects in the world.

Hayes [Hay73] was perhaps the first to recognize the need for a nonmonotonic logic when he noted that rules of default fail to satisfy what he referred to as the *extension property*, which he stated all "respectable" logics should satisfy. A logical theory has the extension property iff whenever a formula is provable from a theory P, it is provable from any set P' such that $P \subseteq P'$. The term "nonmonotonic reasoning" is probably attributable to Minsky's "Frame Paper" [Mins74]. Minsky informally addresses the notion of a frame (which does not relate to the frame problem) and states,

> A frame is a data-structure for representing a stereotype situation, like being in a certain kind of living-room, or going to a child's birthday party. Attached to each frame are several kinds of information. Some of this is about how to use the frame. Some is about what one can expect to happen next. Some is about what to do if these expectations are not confirmed.

The statement, 'about what to do if these expectations are not confirmed,' is a default rule of some sort. This influential paper, in its original, and widely disseminated form, had an appendix entitled, "Criticism of the Logistic Approach." In the appendix Minsky states,

> MONOTONICITY: Even if we formulate relevancy restrictions, logistic systems have a problem in using them. In any logistic system, all the axioms are necessarily "permissive" – they all help to permit new inferences to be drawn. Each added axiom means more theorems, none can disappear. There simply is no direct way to add information to tell such (sic) the about kinds of conclusions that should not be drawn! To put it simply: If we adopt enough axioms to deduce what we need, we deduce far too many other things. But if we try to change this by adding axioms about relevancy, we still produce all the unwanted statements about their irrelevancy.
>
> Because Logicians are not concerned with systems that will later be enlarged, they can design axioms that permit only the conclusions they want. In the development of intelligence the situation is different. One has to learn which features of situations are important, and which kinds of deductions are not to be regarded seriously.

Minsky summarizes his critique of logic as follows.

intelligence have shown that classical logic is not sufficiently robust to adequately reason as humans do. Humans do not always reason as would a classical reasoning system. They leap to conclusions based on *commonsense reasoning*. By *commonsense reasoning* humans generally refer to such statements as, "... it is my experience that 'this' must be the case.", or "... there is no good reason not to believe 'this'."

The subject matter of *nonmonotonic reasoning* is that of developing reasoning systems that model the way in which commonsense is used by humans. Nonmonotonic reasoning must, therefore, be able to leap to conclusions and be sufficiently robust so that when a conclusion reached by nonmonotonic reasoning is shown to be wrong, it may be revised. Nonmonotonic reasoning is based on classical logic, but it is a new logic developed exclusively by workers in artificial intelligence. It is a significant departure from the views of logicians and philosophers concerning humans and reasoning. Two other subjects related to classical logic are also discussed in this paper: logic programming and computational complexity. It is of interest to note that all three topics: nonmonotonic logic, logic programming and computational complexity are unique contributions made to logic by computer scientists.

In this paper I address my remarks primarily to nonmonotonic reasoning as it relates to logic programming. I discuss the pre-history of nonmonotonic reasoning; what has been accomplished in the areas of circumscription, default theory, modal theories and logic programming; and then provide a summary which includes an assessment of the field and what must be done to further nonmonotonic research.

2 Pre-History of Nonmonotonic Reasoning

Nonmonotonic reasoning is connected intimately with the desire to perform *commonsense* reasoning in artificial intelligence (AI). McCarthy [McC58] was perhaps the first individual to discuss the need for the automation of commonsense reasoning, before any theory existed on the subject. Initial formalizations were propounded by McCarthy and Hayes [MH69], who discussed philosophical problems from the standpoint of AI and introduced the frame problem, and by Sandewall [San85] who attempted to find a solution to the frame problem, discussed in terms of robotics by Raphael [Rap71]. The frame problem deals with how one specifies that when an action that is restricted to a set of objects takes

An Overview of Nonmonotonic Reasoning and Logic Programming[1]

Jack Minker

Department of Computer Science and
Institute for Advanced Computer Studies
University of Maryland
College Park, Maryland 20742

Abstract

The focus of this paper is nonmonotonic reasoning as it relates to logic programming. I discuss the pre-history of nonmonotonic reasoning starting from approximately 1958. I then review the research that has been accomplished in the areas of circumscription, default theory, modal theories and logic programming. The overview includes the major results developed including complexity results that are known about the various theories. I then provide a summary which includes an assessment of the field and what must be done to further research in nonmonotonic reasoning and logic programming.

1 Introduction

Classical logic has played a major role in computer science. It has been an important tool both for the development of architecture and of software. Logicians have contended that reasoning, as performed by humans, is also amenable to analysis using classical logic. However, workers in the field of artificial

[1] This paper is an updated version of an invited Banquet Address, First International Workshop on Logic Programming and Non-monotonic Reasoning, Washington, D.C., July 23, 1991 [Min91]