Current Research in RKLLAB (Linköping) using the Features & Fluents Framework

Editor: Erik Sandewall
Email: erisa@ida.liu.se

Abstract

This memo contains an overview of current and recent research in our laboratory related to the "Features and fluents" approach. The current specific topics in this research are:

1. Semantics-based assessment of nonmonotonic entailment methods for actions and change.
2. Implementation techniques for nonmonotonic entailment methods.

IDA RKLLAB Memo 1995
IDA-RKL-95-TR02
ISSN-0281-4250

Department of Computer
and Information Science,
Linköping University,
S-581 83 Linköping, Sweden

Current Research in RKLLAB (Linköping) using the Features & Fluents Framework

Editor: Erik Sandewall
Email: erisa@ida.liu.se

March 13, 1995

1 Topic of Research in our Laboratory: Reasoning about Actions and Change

RKLLAB (The Laboratory for the Representation of Knowledge in Logic) is one of about fifteen research groups in the Department of Computer and Information Science at Linköping University in Sweden. The main research topic of RKLLAB is *computational methods for reasoning about actions and change*. Its purpose is to make possible the following types of automated reasoning about the behavior of a system over time:

- *Prediction* of the future behavior of a given object system (for example, a mobile robot or an industrial plant) on the discrete level of description;
- *Planning* of a sequence of actions which, with sufficient certainty, will take the object system to a given goal state;
- *Chronicle supervision*, where an incoming flow of basic sensory data are compared with an 'agenda' of expected changes (possibly the agenda is a plan that is being executed), so that unexpected events or other deviations from the agenda can be recognized;
- *Plan execution*, involving supervision of the executing plan and *plan revision* when problems arise;
- *Diagnosis*, involving an analysis of what events (which are not directly observable) must have occurred in a given object system, when certain observations of the current and recent states of the system are given.

Typically, the object system about whose behavior one is reasoning is a physical one. Possibly, but not necessarily, it occurs in an industrial environment. The case where the object system is another piece of software is not formally excluded, but it is not the main consideration.

The reasoning tasks of prediction and planning are the ones that we have concretely addressed so far. The other three represent additional, potential applications for the same theory and methodology.

This problem area is presently a topic of active international research, spanning over several disciplines. We approach it from a background of computer science and artificial intelligence, but in active cooperation with other disciplines and, in particular, with the control theory group in our university (headed by Professor Lennart Ljung).

Current challenges in this area of research include *modelling* (how is an aspect of the physical world best described on the level of discrete change?), *logics* (which systems of formal logic are capable of obtaining exactly those conclusions that are warranted in the system being described?), *algorithms* for planning and prediction, and *implementation techniques* including the choice of software architectures, inference 'engines' based, for example, on logic programming or on tableau methods, and compilation and other preprocessing techniques.

We originally entered this research topic because of a number of industrial applications, namely in industrial robotics, autonomous vehicles, and co-driver systems. The prototypical application for us is an autonomous intelligent agent which needs to continuously predict the immediate future and to choose its actions in order to avoid problems and to achieve goals. However, the same technology is used elsewhere for problems of human-machine interaction, common-sense reasoning, natural language understanding, etc.

In our research we address the following more specific research topics:

1. Semantics-based assessment of nonmonotonic entailment methods for actions and change.
2. Implementation techniques for nonmonotonic entailment methods.
3. Algorithmic methods for planning and temporal reasoning.
4. Hybrid modelling of physical systems.
5. Design methodology for intelligent autonomous robotic systems.
6. 'Fuzzy'-related design principles for intelligent autonomous robotic systems.
7. Several industrial application projects.

The first two of these topics are addressed using the *Features and Fluents framework* (FFF), which has been presented in a recent monograph [10], as well as in a number of earlier articles. The present memo describes them only. For information about other activities in RKLLAB or in the department as a whole, see the last section of this memo.

2 Semantics-based assessments for nonmonotonic entailment methods for actions and change

The group for semantical aspects of reasoning about actions and change has the following members at present: professor Erik Sandewall (group leader), professor Witold Łukaszewicz (Warsaw, visiting periodically, shared with

the group of Dr. Patrick Doherty), Ph.D. Pavlos Peppas (Sydney, visiting researcher for parts of 1994–95), graduate students Thomas Drakengren and Choong-Ho Yi, and masters students Silvia Coradeschi and Sergio Brandano (Erasmus students).

2.1 Topic of research and state of the art

Our research topic is to investigate nonmonotonic logics for actions and change with a systematic methodology. Earlier work in this field was usually done by *ad hoc* methods: new logics were proposed and defended by their intuitive plausibility and by a small number of examples where they (were claimed to) give the intended results. Instead, we address the question "for which class of reasoning tasks does a given logic give exactly the intended set of conclusions?", based on a precise definition of intended conclusions. A result of this kind is an *assessment*. A special case of assessment is *validation*, where the definition of intended conclusions is given and one asks whether a given logic always gives the intended set of conclusions (only restricted by the limitations on the definition of intended conclusions).

Several activities alternate in this work: (1) finding an appropriate definition of "intended conclusions" for some choice of characteristics (persistence, qualification, ramification, causation, surprises, etc); (2) finding assessments of existing nonmonotonic logics or entailment methods; (3) revising those logics and entailment methods in order to extend the assessments. (Implementation issues will be considered in the next section).

A fair account of the state of the art in this area is found in the handbook article by Sandewall and Shoham in the *Handbook of Artificial Intelligence and Logic Programming* [13]. Here we just add a few brief observations.

A number of results on assessments or other validations of logics for actions and change have appeared in recent years. Several approaches are being used for defining the set of intended conclusions:

- Definition by another, monotonic axiomatization with a larger set of axioms. This method has been used by Lin and Shoham [22], Lifschitz [20], Kartha and Lifschitz [17], and Lin and Reiter [21].
- Definition by an underlying semantics, which is formally specified and which identifies the set of intended models for a given scenario description. This is the approach that we have used.
- Definition by logical postulates, which specify necessary requirements on the relationship between the set of premises and the intended set of conclusions. This approach was first applied to actions and change (or the "update problem") by Katsuno and Mendelson [19].

Besides the present trend towards assessment and validation studies, there is also a continuing flow of proposals for new nonmonotonic logics without assessments, in particular those which provide new solutions to the classical problems of ramification, qualification, surprises, concurrency, and cause-effect chains. Please refer to the Handbook article for more detail.

2.2 Research results

Our work so far has addressed the relatively limited case of scenario descriptions with strict inertia (no qualification, ramification, concurrency, or surprises), but allowing nondeterministic actions and actions with extended duration in time. They have been reported in conference, journal, and handbook articles [11, 12, 13], and in a monograph that has appeared very recently [10].

The development of the coherent framework for the assessment-oriented approach has been a major part of that work. It included:

- The definition of the underlying semantics. Our choice of underlying semantics is based on the use of an *ego-world game*, that is, a formalized version of the interaction between the intelligent system and the world in which it operates. In the case of strict inertia, the "ego" has a very simplistic definition, and the "world" is defined in terms of *trajectories*. We define a trajectory as a finite sequence of partial states of the world. From a semantic point of view, each action denotes a set of trajectories, representing the various ways that the action can be executed.

- The definition of an ontological (and epistemological) *taxonomy* of scenario descriptions. This taxonomy is based on the underlying semantics, and it is used as the coordinate system for expressing the range of applicability of the various entailment methods.

- The definition of a coherent perspective on the various temporal reasoning operations that may be needed in applications: prediction, explanation, various varieties of planning and replanning, etc.

With this framework in place, we have assessed the range of applicability of about one dozen nonmonotonic entailment methods (that is, variants of the logic which only differ with respect to the definition and the use of the preference relation on models). The following difficulties are allowed for in the results:

- Forward-branching as well as linear, integer time: definition and assessment of entailment methods for both alternatives.

- Actions with extended duration in time.

- Composite actions (composed using sequencing, conditionals, and loops): definition and and assessment of entailment methods which apply for composite actions.

- Both lower bounds and upper bounds on the range of applicability have been identified, and are sometimes equal (so that the range of applicability has been precisely determined).

- Range of soundness as well as range of completeness has been identified.

A number of more general results can be obtained already within the existing framework, or with relatively minor extensions of it. Thus, *Choong-ho Yi* is just completing a licentiate thesis where the range of applicability is analyzed for prototypical chronological minimization, PCM (with and without filtering), for a restricted case of concurrent actions, namely concurrent actions without interaction [15]. It seems a priori likely that PCM will apply to some cases of concurrent actions with interaction, and that the modified method also using occlusion will cover a significantly broader set of cases. We propose to continue his research in these directions.

Our visiting researcher *Pavlos Peppas* addresses the problem of ramification. He has his background in a group that has traditionally used the postulate-based approach, and contributes those impulses to our group. The problem of ramification is also addressed in some present work by *Erik Sandewall*, where I define intended models for the ramification case using the variant of the approach of Kartha and Lifschitz [18], and investigate range of applicability for other current methods on that basis.

Silvia Coradeschi investigates he problem of scenarios with unreliable observations. We view this as a first attempt at a difficult problem which is of great practical importance.

Thomas Drakengren has extended and corrected earlier results on the use of composite actions.

The main effect of these and other similar projects is to obtain the full benefit from the existing framework, in particular the trajectory semantics that has been used in our work so far.

3 Implementation techniques for nonmonotonic entailment methods

The nonmonotonic reasoning group currently has the following members: assistant professor Patrick Doherty (group leader), professors Witold Łukaszewicz and Andzrej Szałas (Warsaw, visit 1-3 months per year), and graduate student Lars Karlsson (completing his licentiate degree in February, 1995). In addition we have cooperation with professor John-Jules Meyer (Utrecht) and his group.

3.1 Topic of research and state of the art

The research problem for our group is to find efficient implementations for (nonmonotonic) logics for actions and change. We are particularly interested in those logics using explicit, metric time, since this appears to be essential for robotics-type applications where time-stamped sensory data are involved. We also focus on those logics for which there exists an assessment of their range of applicability, since it is only in this way that one can rely on the logic from the point of view of the application. In particular, we have investigated many of the different nonmonotonic temporal logics that are defined and assessed in Sandewall's book "Features and Fluents" [10]. As assessments are obtained for logics for broader ontological classes (compare section 1), we will address their implementation problems as well.

For the implementation problem, the following is typically given:

- A non-monotonic logic for actions and change, which is characterized model-theoretically in terms of a preference relation on models and various specificities concerning how that preference relation is put to use.
- A particular reasoning operation which one wants implemented, such as verification of a proposed consequence, deduction to find a member of a given class, or abduction as an implicit deduction problem.

The specific goal is then to identify and implement an efficient method for this reasoning operation in the given logic, and to investigate the logical and computational properties of that method. The "method" may involve uniform theorem provers, specialized algorithms, software architectures, pre-transformations on the formulae, and so on.

Much of the current research in this field has its roots in methods for general theorem-proving (where we include the work on nonmonotonic logic programming). Such methods may be specialized to the particular conditions of reasoning about time, actions, and change. One can distinguish the following basic approaches to implementation:

- Semantic-based approach
- Syntactic-based approach
- Operational-based approach

The following discussion of these approaches will be restricted to logics with explicit time, omitting work based on modal-temporal logics for reasons of space.

The semantics-based approach is related to model-checking [29], but modified towards dealing with minimal models. The following reasoning tasks are considered: 1) model finding - Find a minimal model for Γ, 2) model checking - Check whether a given interpretation is a minimal model for Γ, 3) minimal entailment - Is α true in all minimal models of Γ?, 4) minimal membership - Is α true in at least one minimal model of Γ. Although complexity results regarding reasoning with minimal models are very discouraging, certain properties on formulas such as *head-cycle freeness*, provide for more efficient algorithms for the above reasoning tasks [23]. Although, not as easily classified under the model checking approach, we would also include in this class, research with specialized algorithms for computing preferred entailments such as those of Ginsberg [28] or Winslett [41].

The syntactic-based approach appeals to more conventional theorem-proving techniques in the following manner. The non-standard temporal logic is translated into a conventional n-sorted first-order language. Circumscription axioms are associated with each of the definitions of preferential entailment. By taking advantage of the syntactic and semantic constraints associated with each ontological class, it is often the case that the 2nd-order circumscription axioms can be reduced to equivalent 1st-order formulas. The net result is monotonic 1st-order theories as the target for implementation.

Consequently, one can appeal to classical theorem-provers or logic programming as a vehicle for implementation. There are a number of related results in this area which could prove useful. Some representative references are: [27],[30], [32], [34], [37]. Additionally, the technique of "explanation closure" has very strong connections with syntactic-based approaches ([39], [40]). In fact, it is becoming increasingly more popular and can be explained in terms of circumscriptive reductions which provide a means for automatically generating closure axioms [3].

The operational-based approach appeals to the strong formal relation between the underlying semantics and the models associated with the temporal non-monotonic logics. The underlying semantics can be viewed as a generalization of the state transitions systems used as operational models for programming languages, reactive systems, and as a basis for the formal semantics of dynamic and tense logics. The idea here is to take advantage of this formal relation and view action scenarios as programs in a programming language and the underlying semantics as an operational semantics for the action scenarios. One can then use conventional verification techniques as a tool for reasoning about action and change. Again, we are able to do this by capitalizing on the various syntactic and semantic constraints associated with each of the ontological classes.

There are also other implementation methods which are more remote from the logical basis, such as truth maintenance systems and planning systems. The principled development of practical software systems for planning is a very active research area. It is possible to view a planning system abstractly as a very specialized system for temporal reasoning, but practical planning systems are usually described, instead, in more pragmatic and algorithmic terms.

3.2 Research results

At this stage in our research, it is not yet clear which of these approaches toward implementation of existing and proposed logics for actions and change is the most feasible, in general or in the Features and Fluents framework. Perhaps all of them are feasible, but for different classes of scenarios, or perhaps they need to be combined. So far we have addressed the following approaches:

- Syntactic-based general-purpose approach
- Operational-based general-purpose approach
- Implemented planning systems

One of the advantages of the Features and Fluents methodology, as described in the previous section, is that each of the logics is customized towards the ontological characteristics which are defined in the underlying semantics, and which determine the classes in the ontological taxonomy. Rather than searching for implementations for one general nonmonotonic temporal logic, we can take advantage of the particular characteristics of each class. These constraints, both syntactic and semantic, provide an opportunity for more efficient implementations, at least for some of these classes.

3.2.1 The syntactic-based approach

We have placed most of our resources on the syntactic-based approach due to the somewhat surprising initial success with these techniques. In Doherty and Łukaszewicz [6], we provided syntactic characterizations of each of the logics of action and change in terms of classical FOL and circumscription axioms. The intent of the exercise was to take advantage of a number of reduction results which exist in the literature on circumscription. Although a long shot, it turned out that one of the logics in Features and Fluents, PMON, had a circumscriptive definition that was found to be reducible to the first-order case. Not only does PMON cover the broadest class of action scenarios considered so far in the Features and fluents framework, but it subsumes the other logics described in the book. This work is reported in [4]. By restricting syntax for action types in PMON, it can even be shown that certain scenarios can be compiled into logic programs [3].

We are currently investigating the generic case of reducing 2nd-order circumscription axioms to equivalent 1st-order formulas. More about this in subsection 2.3.1.

We are currently investigating the generic case of reducing 2nd-order circumscription axioms to equivalent 1st-order formulas. Initial results in this direction are promising. In Doherty et. al. [5](Annex 9), we present an implementable algorithm which takes a 2nd-order formula as input and either returns an equivalent 1st-order formulas as output or terminates. Of course, termination does not mean that there doesn't exist an equivalent 1st-order formula, just that the algorithm can not generate one. More surprisingly, this algorithm provably subsumes existing reduction results in the literature, namely those of [32],[33], [34], [31], [26], and [14]. We propose to continue work in this direction in the following manner:

- Integrate special purpose temporal constraint algorithms with the circumscription reduction algorithm.
- Specialize the circumscription reduction algorithm to take advantage of the various syntactic and semantic constraints associated with the various ontological classes.
- Generalize the algorithm to deal with as yet non-reducible classes of 2nd-order circumscription axioms. In particular we expect a reduction to fixpoint logics for certain types of temporal nonmonotonic entailment. Results in this direction will also be useful for a number of situation calculus based approaches ([40], [35]).
- Study how closure axioms in the context of the explanation closure technique can be automatically generated for theories via the reduction of circumscription axioms.

3.2.2 The operational-based approach

An obvious similarity between actions and commands of programming languages, and between action scenarios and programs, suggests that it may

be fruitful to formalize reasoning about action and change using methods that have been introduced to reason about programs. Dijkstra's approach [24],[25] provides one such possibility.

The basic idea underlying Dijkstra's methodology is to specify effects of actions in terms of *formula transformers*, i.e. functions from formulae into formulae. More specifically, with each action A, there are associated two formula transformers, called the *strongest postcondition for* A and the *weakest liberal precondition for* A. The former, when applied to a formula α, returns a formula representing the set of all states that can be achieved by starting execution of A in some state satisfying α. The latter, when applied to a formula α, returns a formula providing a description of all states such that whenever execution of A starts in any one of them and terminates, the output state satisfies α.

The formula transformers approach to reasoning about action and change is limited in expressibility and no claim is made that it will provide a basis for implementation for all classes in the Features and Fluents framework. However, the technique can be applied to a broad class of actions types, including non-deterministic actions. It can also be extended to model certain limited types of ramification. In addition, both temporal prediction and postdiction reasoning tasks can be specified without restricting initial or final states to be completely specified. The major advantage of this approach is that it provides a basis for a straightforward and efficient implementation of a number of interesting ontological classes.

In [8], we have applied Dijkstra's methodology to formalize reasoning about deterministic actions. In [9] we have generalized the approach to actions where non-deterministic and indirect effects are allowed.

3.2.3 Reconstruction of known systems

A number of practical software designs have been developed, in a more or less *ad hoc* manner, for action planning and to some extent for other reasoning operations. In principle, these systems can be seen as realizations of reasoning operations, but in practice they have not been conceived and built that way. However, they do embody a considerable amount of practical know-how, which one would like to extract and transfer to the context of systematic logic-based methods.

Our graduate student, Lars Karlsson, has just completed a licentiate thesis that takes a step in that direction. Starting from the general "Features and fluents" framework, he has reconstructed a number of classical planners in a systematic way: STRIPS, TWEAK, SNLP, and the sequential restriction of NONLIN. This has given an insight into how one can go from the logical basis to a practically working system. As far as we know, his work is unique in its kind.

4 Other research in RKLLAB

Under the term "reasoning about actions and change", we include several kinds of reasoning operations, with *planning* and *prediction* as two specific

instances. The traditional approach, in particular for planning, has been systems-oriented and emphasized the principled design of computer programs for this purpose. Besides the work based on the FFF approach, our laboratory also contains the group of Christer Bäckström and his affiliates, who address *algorithmic methods for planning and temporal reasoning*. In that approach, the planning system is viewed in abstract form as an algorithm, and the computational complexity properties of that algorithm is investigated, in order to find algorithms that are tractable (or otherwise have manageable properties) in well defined domains.

In the FFF approach, on the other hand, the planning system is viewed instead as a specialized implementation of a logical reasoning operation – typically an abduction operation in a particular nonmonotonic logic. The goal of the research is then to identify the underlying nonmonotonic logic and its range of applicability (work described in section 2), and to show how a systematic implementation can be built, in a well-founded way, on that logic. One step on that way has been the licentiate work of Lars Karlsson, who showed how the core algorithms of several well-known planning systems (STRIPS, TWEAK, SNLP, and the sequential restriction of NONLIN) can be reconstructed from the logics that were assessed in the FFF approach. At the same time, we also start from the general perspective of modern techniques for nonmonotonic theorem-proving, and study how they can be applied in an efficient fashion to various logics whose range of applicability is well understood.

What has now been said about planning applies equally for other temporal reasoning operations, such as prediction and explanation. There are well-known traditional arguments in favor of the algorithmic or the logic-based approach. Direct implementations of the logic-based approach using conventional theorem-provers obtain systems with unacceptable computational properties, and it does not seem likely that heuristic guidance can help this situation in any fundamental way. This speaks in favor of the algorithmic approach. On the other hand, tractable algorithms are only obtained for limited subclasses of planning scenarios with strict inertia, and they would seem to be out of reach for those cases that involve ramification, qualification, surprises, cause-effect chains, and so on. Such a richer range of problems would therefore seem to require the logic-based approach.

To a certain extent it is appropriate to consider these approaches as alternative tracks, each with its pros and cons, and to pursue them independently of each other. That is also how they have been presented in the previous sections of this memo. However, we also see several kinds of cross-fertilization between these approaches. Sometimes they can use similar underlying techniques, or borrow techniques from each other. The "action structures" view of actions underlies both the FFF approach and the planning complexity work in our laboratory. Also, the technique of dividing the set of state variables into partitions has been found useful in the logic-based FFF approach (cf. Sandewall's occlusion operator, Lifschitz's frame concept, the released concept of Kartha and Lifschitz, the persists concept of del Val and Shoham, etc.), but Bäckström et al. equally use a partitioning of the state variables in their most recent algorithmic method. The list could be

continued.

In a longer perspective, one would also be interested in hybrid implementations, where algorithmic methods are included as subsystems of logic-based software systems, or where algorithmic systems escape to logic-based ones in particular situations. For example, in logics with explicit time (that is, first-order formalizations) it would be reasonable to separate the management of the time domain as a separate sub-system of the reasoning system, and to implement it using known techniques for time-map management.

5 Additional information

Research reports from RKLLAB can be accessed by WWW from the departmental home page:

`http://www.ida.liu.se/`

In the present layout of the home page, please find the item "publications" and click the phrase "technical reports". This gives access to a search facility and to complete lists of titles and abstracts for browsing, with direct links for those reports that are electronically accessible. Alternatively, the RKLLAB home page:

`http://www.ida.liu.se/labs/rkllab/`

refers to one structure for each laboratory member, which contains (or at least, is intended to contain) references to his or her publications. – Reports may also be accessed directly from the FTP address,

`ftp.ida.liu.se`

under the RKLLAB ftp directory,

`/pub/labs/rkllab`

This directory has one subdirectory for each laboratory member, containing his or her publications, for example

`/pub/labs/rkllab/people/erisa`

for erik sandewall. Information about our department as a whole and about its other laboratories can be accessed from the RKLLAB home page.

References

[1] P. Doherty. A constraint-based approach to proof procedures for multi-valued logics. In *1st World Conference on the Fundamentals of Artificial Intelligence, Paris, 1-5 July, Proceedings*, 1991. WOCFAI-91.

[2] P. Doherty. *NML3 - A Non-Monotonic Formalism with Explicit Defaults*. PhD thesis, University of Linköping, Sweden, 1991.

[3] P. Doherty. Notes on PMON circumscription. Technical Report LITH-IDA-94-43, Department of Computer and Information Science, Linköping University, Linköping, Sweden, December 1994.

[4] P. Doherty. Reasoning about action and change using occlusion. In *Proceedings of the 11th European Conference on Artificial Intelligence, Aug. 8-12, Amsterdam*, pages 401–405, 1994.

[5] P. Doherty, W Łukaszewicz, and A. Szałas. Computing circumscription revisited: A reduction algorithm. Technical Report LITH-IDA-94-42, Department of Computer and Information Science, Linköping University, Linköping, Sweden, December 1994.

[6] P. Doherty and W. Łukaszewicz. Circumscribing features and fluents. In D. Gabbay and H. J. Ohlbach, editors, *Proceedings of the 1st International Conference on Temporal Logic*, volume 827 of *Lecture Notes in Artificial Intelligence*, pages 82–100. Springer, 1994.

[7] Lars Karlsson. Specification and Synthesis of Plans Using the Features and Fluents Framework. Licentiate thesis. Linköping Studies in Science and Technology: Thesis No 469. 1995.

[8] W. Łukaszewicz and E. Madaliǹska-Bugaj. Program verification techniques as a tool for reasoning about actiona and change. In *Proceedings of the 18th German Conference on Artificial Intelligence*, volume 861 of *Lecture Notes in Artificial Intelligence*, pages 226–236. Springer-Verlag, 1994.

[9] W. Łukaszewicz and E. Madaliǹska-Bugaj. Reasoning about actiona and change using dijkstra's semantics for programmming languages. Submitted to IJCAI-95, 1995.

[10] Erik Sandewall. *Features and Fluents. The Representation of Knowledge about Dynamical Systems. Volume I.* Oxford University Press, 1994.

[11] Erik Sandewall. The range of applicability of nonmonotonic logics for the inertia problem. In *International Joint Conference on Artificial Intelligence*, 1993.

[12] Erik Sandewall. The range of applicability of some nonmonotonic logics for strict inertia. *Journal of Logic and Computation*, 4(5), October 1994.

[13] Erik Sandewall and Yoav Shoham. Nonmonotonic temporal reasoning. In Dov Gabbay, editor, *Handbook of Artificial Intelligence and Logic Programming (to appear)*. Oxford University Press, 1994.

[14] A. Szałas. On an automated translation of modal proof rules into formulas of the classical logic. *Journal of Applied Non-classical Logics*, 4::119–128, 1994.

[15] Choong-Ho Yi. Towards the assessment of logics for concurrent actions. In *AAAI 1995 Spring Symposium*, 1995.

[16] Choong-Ho Yi. Reasoning about Concurrent Actions in the Trajectory Semantics. Licentiate thesis. Linköping Studies in Science and Technology: Thesis No 475. 1995.

External references for section 2

[17] G. Neelakantan Kartha. Soundness and completeness theorems for three formalizations of action. In *International Joint Conference on Artificial Intelligence*, pages 724–729, 1993.

[18] G. Neelakantan Kartha and Vladimir Lifschitz. Actions with indirect effects (preliminary report). In *International Conference on Knowledge Representation and Reasoning*, pages 341–350, 1994.

[19] Hirofumi Katsuno and Alberto O. Mendelzon. A unified view of propositional knowledge base update. In *International Joint Conference on Artificial Intelligence*, pages 1413–1419, 1989.

[20] Vladimir Lifschitz. Toward a metatheory of action. In *International Conference on Knowledge Representation and Reasoning*, pages 376–386, 1991.

[21] Fangzhen Lin and Ray Reiter. How to progress a database (and why) I. Logical foundations. In *International Conference on Knowledge Representation and Reasoning*, pages 425–436, 1994.

[22] Fangzhen Lin and Yoav Shoham. Provably correct theories of action (preliminary report). In *National Conference on Artificial Intelligence*, pages 349–354, 1991.

External references for section 3

[23] R. Ben-Eliyahu and L. Palopoli. Reasoning with minimal models: Efficient algorithms and applications. In *KR-91: Proceedings of the Int'l conference on principles of knowledge representation and reasoning*, pages 39–50, 1994.

[24] E. W. Dijkstra. *A Discipline of Programming.* Prentice Hall, 1976.

[25] E. W. Dijkstra and C. S. Scholten. *Predicate Calculus and Program Semanitcs.* Springer-Verlag, 1990.

[26] D. Gabbay and H. J. Ohlbach. Quantifier elimination in second-order predicate logic. Technical Report MPI-I-92-231, Max-Planck Institut für Informatik, Saarbrücken, Germany, 1992.

[27] M. Gelfond and V. Lifschitz. Representing actions in extended logic programming. In K. Apt, editor, *Proc. Joint Int'l Conf. and Symp. on Logic Programming*, pages 559–573, 1992.

[28] M. L. Ginsberg. A circumscriptive theorem prover. *Artificial Intelligence*, 39::209–230, 1989.

[29] J. Halpern and M. Vardi. Model checking vs. theorem proving: a manifesto. In *KR-91: Proceedings of the Int'l conference on principles of knowledge representation and reasoning*, pages 325–334, 1991.

[30] G. N. Kartha and V. Lifschitz. Actions with indirect effects (preliminary report). In *Proc. of the 4th Int'l Conf. on Principles of Knowledge Representation and Reasoning, (KR-94)*, pages 341–350, 1994.

[31] P. Kolaitis and C. Papadimitriou. Some computational aspects of circumscription. In *AAAI-88: Proceedings of the 7th National Conference on Artificial Intelligence*, pages 465–469, 1988.

[32] V. Lifschitz. Computing circumscription. In *Proc. IJCAI-85*, volume 1, pages 121–127, 1985.

[33] V. Lifschitz. Pointwise circumscription. In M. Ginsberg, editor, *Readings in Nonmonotonic Reasoning*, pages 179–193. Morgan Kaufmann, 1988.

[34] V. Lifschitz. Circumscription. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Nonmonotonic Reasoning and Uncertain Reasoning*, volume 3 of *Handbook of Artificial Intelligence and Logic Programming.* Oxford University Press, 1994.

[35] F. Lin and R. Reiter. State constraints revisited. *Journal of Logic and Computation*, 1994. (To Appear).

[36] J. J. Lu, L. J. Henschen, V. S. Subrahmanian, and N. C. A. da Costa. Reasoning in paraconsistent logics. In R. Boyer, editor, *Automated Reasoning: Essays in Honor of Woody Bledsoe*, pages 181–210. Kluwer Academic, 1991.

[37] T. Przymusinski. An algorithm to compute circumscription. *Artificial Intelligence*, 38::49–73, 1991.

[38] Raymond Reiter and Johan de Kleer. Foundations of assumption-based truth maintenance systems. In *National Conference on Artificial Intelligence*, 1987.

[39] R. Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation*, pages 359–380. Academic Press, 1991.

[40] L. Schubert. Monotonic solution of the frame problem in situation calculus. In H. E. Kyburg, R. P. Loui, and G. N. Carlson, editors, *Knowledge Representation and Defeasible Reasoning*, pages 23–67. Kluwer, 1990.

[41] M. Winslett. *Updating Logical Databases.* Number 9 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1990.