Current Research in RKLLAB, Linköping University's Laboratory for the Representation of Knowledge in Logic

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

Abstract

This memo presents current research in our laboratory. We address the following 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.

IDA RKLLAB Memo 1995
IDA-RKL-95-TR01
ISSN-0281-4250

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

Current Research in RKLLAB, Linköping University's Laboratory for the Representation of Knowledge in Logic

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

March 11, 1995

1 Topic of Research: Reasoning about Actions and Change

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.

Internally, we view these topics as closely related, and there is a high degree of interaction between our researchers in these topics. From an external point of view, however, these topics have their own communities, literatures, and research styles. For this reason, and also because each of our researchers is primarily affiliated with one of the topics, we choose to present them in separate sections of this memo. The first three topics will be presented in more detail than the latter ones. A final section discusses the interactions and interrelationships between the topics.

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

The group for 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* [10]. 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 [6], Lifschitz [4], Kartha and Lifschitz [1], and Lin and Reiter [5].

- 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 [3].

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 [7, 9, 10], and in a monograph that has appeared very recently [8].

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 [11]. 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 [2], 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 Lukaszewicz and Andrzej 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" [37]. 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 [24], 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 [12]. 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 [23] or Winslett [40].

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: [22],[25], [27], [29], [34]. Additionally, the technique of "explanation closure" has very strong connections with syntactic-based approaches ([36], [38]). 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 [17].

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 Lukaszewicz [20], 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 [18]. By restricting syntax for action types in PMON, it can even be shown that certain scenarios can be compiled into logic programs [17].

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. [19](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 [27],[28], [29], [26], [21], and [39]. 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 ([38], [30]).
- 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 [13],[14] 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 [32], we have applied Dijkstra's methodology to formalize reasoning about deterministic actions. In [33] 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 Algorithmic methods for planning and temporal reasoning

The group for complexity of planning presently consists of asst.prof. Christer Bäckström and graduate student, M.Sc. Peter Jonsson. The group cooperates with Dr. Inger Klein in the Divison of Automatic Control in our university. In 1994 we have had a three-month visit by Dr. Jana Köhler, DFKI, Saarbrücken, Germany.

4.1 Topic of research and state of the art

4.1.1 Action planning

The *action planning problem*[1] can be briefly described as follows: Given an initial (current) state, a goal (desired) state, and a collection of operators, the planning problem is to find a sequence of operators that, when applied in the initial state, changes the state of the world to the goal state. Although the research field of action planning has reached the age of some 25 years, many unsolved problems remain and the attempts at a theoretical understanding of the field has evolved only during the past five to ten years.

One of the most important problems faced by the planning research community is how to plan efficiently. Planning is known to be very hard in general; even for very simple formalisms, deciding whether a solution exists for a given problem instance is undecidable in the first-order case [71, 77] and PSPACE-complete in the propositional case [66, 69].

The most commonly studied type of planner is the so-called *precondition-achievement (PA) planner* which works by repeatedly selecting some goal or subgoal (an action precondition) which is not satisfied and try to make this true by adding constraints or new actions to the plan. Many choices are involved in each step, so planning is done by search. In order to reduce the search effort, one must choose an appropriate search strategy together with appropriate, typically domain-dependent, heuristics in order to reduce the search effort.

Some form of hierarchical abstraction is often used to make planning more efficient. State abstraction [95, 87] first solves an abstract version of the problem instance and refines this solution at less and less abstract levels by inserting more detailed subplans. Operator sequences can be abstracted as macro operators [88], *ie.* treating a whole sequence as a single operator, which are used as ordinary operators and expanded in the final solution. *Hierarchical Task Network (HTN) planning* [98, 100, 73] recursively expands goal tasks into subplans—which contain new goal tasks and operators—according to predefined expansion schemas. All these approaches can make planning very efficient under ideal circumstances, but there is no guarantee in the general case. Typically, we have to rely on handcrafted, domain-dependent knowledge for these approaches to work well.

[1] We use action planning to denote the traditional AI planning field within the nowadays large and fuzzy area of AI planning.

The relationship between algorithmic and logic-based planning will be discussed in Section 4, below.

4.1.2 Restricted problems

Although the planning problem is proven to be very hard even for the standard formalisms, one must keep in mind that these complexity figures apply to the hardest problems that can be encoded in each formalism. They say nothing about the inherent complexity of application problems. It is often claimed that even the simple, standard planning formalisms are too restricted to be useful in practice. However, planning is undecidable even in the TWEAK planning formalism [71], for instance, which, hence, can encode more than we could ever hope to compute. Of course, most problems would have to be encoded in a very contrived and unnatural way in order to fit into the TWEAK formalism. Yet, the problem with the TWEAK formalism is not lack of expressive power—rather a lack of naturalness or easiness to express real problems. This means that although we could wish for a number of more advanced constructs in the standard planning formalisms, we could, at the same time, most likely limit the use of many other constructs in order to reduce the complexity—still being able to model most application problems.

As an example, take the ubiquitous *blocks-world* problem which in its simple, standard form can be encoded in the propositional version of the STRIPS formalism [78]. We know that for this formalism, it is PSPACE-complete to decide whether a plan exists or not [66]. However, for the blocks-world problem *per se* a plan can be found in low-order polynomial time [51, 52, 80] while the decision problem for bounded plans is NP-complete, *ie.* still better than PSPACE-complete. Furthermore, the modification where several blocks may have identical labels cannot be naturally modelled in propositional STRIPS, yet this problem is also NP-complete [72]. This shows that propositional STRIPS is more expressive than necessary for this problem, yet it lacks the constructs to find a straight-forward, natural modelling of the problem.

The conclusion of this seems to be that there are likely to be many planning applications that do not need the full expressive power of even the standard formalisms (although they may benefit from adding constructs to these formalisms). Similar observations have motivated research into identifying restrictions that make hard problems tractable in other areas than planning, *eg.*, *concept languages* [65, 75] and *abduction* [70]. Tractable planning problems have been identified, especially by ourselves (see Section 4.2.1) but also by others [66, 67, 76].

4.1.3 Applications in Automatic Control

Although of general, theoretical nature, our research has to a large extent been motivated by the application area of *sequential control* within the subarea *discrete event dynamic systems* of *automatic control* where many problems can be viewed as planning problems. Researchers in this area see the lack of mathematically well-founded methods with provable properties as one of the major problems with AI planning [92]. Furthermore, a recent study [47, 46] of how computer software is used in large-scale industrial control applications has shown that the industry demands better mathematical tools for modelling dynamical systems of combinatorial nature, since heuristics and expert systems are not considered safe. We have found this a strong reason for insisting in doing research into provably correct planners and complexity analyses of restricted planning problems.

One of our target test applications is the LEGO Car Factory [97], which is a miniature, automated assembly line for LEGO cars, used for undergraduate laboratory assignments in sequential control at the department of EE at Linköping University. This application is interesting because it contains many of the problems and properties found in real industrial processes, while being small enough to be used as a test application and to analyse theoretically.

4.2 Research results

Our main line of research during the past five years has been to study the computational complexity of various restricted planning problems, especially identifying tractable ones, to some extent motivated by applications in automatic control. This research has many connections into other areas, which has forced us, at times, to also carry out research into related areas within planning and temporal reasoning.

4.2.1 Planning Complexity

The anticipated need for provably correct and efficient planners have motivated us to follow the restricted-problems line of research, trying to identify tractable planning problems.

In 1990 we presented the, to our knowledge, first ever tractable planning problem, the SAS-PUBS problem [59, 61]. In contrast to other, more recent publications into this topic [66, 76] we did not use the formalism *per se* as a guide to which restrictions to use. Instead we identified inherent restrictions in a toy problem in the sequential control domain, the tunnel problem [51]. These restrictions, denoted P, U, B and S, are essentially syntactic restrictions on the sets of operators and state variables. Some of these are local restrictions on operators, similar to the ones studied by Bylander [66] and Erol *et.al.* [76]. The others are global restrictions on the whole set of operators, similar to the ones used by Korf [88]. However, while it is PSPACE-complete to test whether a certain problem satisfies Korf's restrictions [68], all our restrictions can be tested in polynomial time [51].

We continued pursuing a bottom-up approach; starting with the SAS-PUBS problem, we repeatedly removed restrictions such that the resulting problem is still tractable. This led to proving tractability for the SAS-PUS [60], SAS$^+$-PUS [51, 52] and SAS$^+$-US [63, 64] problems and we developed sound and complete, polynomial-time planning algorithms for each of these problems. In one of these steps we removed restrictions on the basic formalism (from SAS to SAS$^+$) rather than removing any of the P, U, B and S restrictions. This first phase of our research was concluded by an analysis of the complexity for all combinations of the SAS and SAS$^+$ formalisms and of the P, U, B and S restrictions [63, 64]. In this analysis we studied both the existence problem (finding out whether a solution exists or not) and the generation problem (generating a solution or answer that none exists) and we considered both bounded and unbounded solutions. The result was that under the restrictions considered, SAS$^+$-PUS is the maximal tractable problem for bounded planning and SAS$^+$-US is the maximal tractable problem for unbounded planning.

The next phase, thus, necessitated stepping outside the previous restrictions. It seemed obvious that simple syntactic restrictions like the previous ones would not take us much further. Ideally, we should explore structure of the state transition graph to detect properties that make planning tractable. However, this is not practical since this graph is typically exponentially larger than the problem instance itself, making testing of arbitrary structural properties very costly. Hence, we took an intermediate approach: we started investigating structural properties of the state transition graphs for each state variable in isolation, taking also some inter-variable structure into account. This led us to identify three new, structural restrictions, I, A and O, which are tractable to test and which make planning tractable. We have devised a polynomial-time algorithm which is sound for SAS$^+$ under the I and A restrictions and further complete if also O holds [84]. The SAS$^+$-IAO problem properly includes the SAS$^+$-PUS problem, thus generalizing the previous results. Similar to the previous complexity analysis we analysed all combinations of the old syntactic and the new structural restrictions for (un)bounded plan existence/generation [83].

In this research, we have used a non-standard formalism, SAS$^+$, which is a slightly restricted version of the action structures formalism [96] and can be viewed as a variation on propositional STRIPS. We have proven [51, 52, 56] the SAS$^+$ formalism to be expressively equivalent, under polynomial reduction, to two variants of the propositional STRIPS language and to the ground TWEAK language. This result shows that neither negative pre-conditions, negative goals, partial initial states, nor multi-valued state variables add to the expressiveness of propositional STRIPS. Despite this equivalence, it was crucial that we used the SAS$^+$ formalism, since some of the restrictions we have studied are not very useful for binary state variables. Recent results [57] also show that despite the equivalence between the formalisms, the formalisms are not equivalent with respect to actual planning algorithms—it may be exponentially more costly to reencode a SAS$^+$ problem in STRIPS and then solve it using a standard partial-order planner, SNLP [89], while there is no efficiency loss in the other direction.

4.2.2 Other related research

Dean and Boddy [74] analyzed the problem of *temporal projection*, *i.e.* deciding the effects of a set of partially ordered events, for a certain variant of propositional STRIPS. They found the problem to be NP-hard, even under severe restrictions, and conjectured these complexity results to carry over to the problems of planning, *plan validation* and *story understanding*. We showed [62, 90, 91] that for all three of these problems, there are restrictions making them tractable, while the corresponding temporal projection problems are still NP-hard under the same restrictions, thus disproving Dean and Boddy's conjecture. We further showed that plan validation is, in fact, tractable for the important class of all unconditional plans [90, 91].

Our research into planning complexity has also forced us to reconsider some results from the literature on converting total-order plans into partial-order plans, subject to certain constraints. We found [54] that two algorithms in the literature for converting total-order plans into partial-order plans that are *least constrained* [99] or have a *shortest parallel execution* [93] respectively, seem not to satisfy these optimality claims. In the absence of definitions from the authors of the algorithms, we suggest some reasonable definitions for least-constrainment and optimal parallel executions. Most of these criteria are proven NP-hard and even impossible to approximate within a constant. Further, one least-constrainment criterion is tractable to achieve, but the algorithm in the literature fails to satisfy even this kind of (local) optimality. Continuing this line of research we have further found [55] that while a sequential plan will necessarily execute slower if actions are added to it, a parallel plan can in some cases be made to execute faster by adding actions to it.

4.2.3 Ongoing research

The SAS$^+$-IAO problem allows only certain, limited types of cycles in the state transition graph, which is why it is not quite sufficient for modelling our main target application—the LEGO car factory. However, by attempting such a modelling and using this as a feedback for the theory, we have developed a provably correct, polynomial-time planning method based on the SAS$^+$-IAO algorithm which handles the LEGO car factory [85]. The method builds on automatically partitioning the problem instance into two parts. The first of these is solved using the SAS$^+$-IAO algorithm, resulting in a plan skeleton which can be filled in by applying the SAS$^+$-IAO planner to the second part.

We have noted [61, 64, 83] that, even for many of the restricted problems, one source of intractability is that the minimal solutions are exponentially sized. Such problems are inherently intractable since it would take exponential time even to output the solution. However, problems of this type are usually regarded as unrealistic [79, 51], since there seems to be little practical interest in exponentially sized solutions. This has led us to pay attention not only to the computational complexity figures, but also to the length of minimal solutions, since we believe this to be an important factor. We are working on these issues and have a preliminary result [82], the class 3S, which contains problem instances such that plan existence is tractable, but plan generation is inherently intractable, due to exponentially sized minima solutions. This result puts into question whether it is relevant to study the complexity of plan existence only, as has been the case in many other reported research in the literature. Furthermore, this also allows us to devise a planner which is prefix-generating, that is, it can output a prefix of the final solution, before it has finished generating the whole solution, and it may also recursively output prefixes of prefixes *etc.*

Recent research on the 3S problem turned out the be closely related to the work on state abstraction in planning, leading to some new insight into this area. State abstraction is known to speed up planning exponentially under ideal circumstances [87], but these circumstances are probably rare in practice. It is also known that state abstraction can make planning slower in other cases, but it seems to be assumed to be relatively harmless slowdowns. We have shown [58] that not only can planning be exponentially slower with state abstraction, we may also end up having to generate a solution which is exponentially longer than the minimal one! We also analysed two algorithms for generating abstraction hierarchies, ALPINE [87, 86] and HIGHPOINT [48]. It turns out that both algorithms may generate such bad abstraction hierarchies as described above. Furthermore, it is even impossible to generate abstraction hierarchies which allows generating minimal solutions.

Some researchers have suggested to us that it is better to look for good heuristics for standard general-purpose planners than to identify restricted tractable planning problems—*ie.* they advocate a top-down approach rather than a bottom-up approach. We believe that both these approaches are good and that they complement each other. Furthermore, we have recently shown [unpublished result] that standard planners equipped with heuristics are unlikely to achieve what our planning algorithms do. We have identified instances of our tractable problem classes where SNLP is forced to take exponential time, while our algorithms run in polynomial time. For each such instance, we have identified a heuristic that helps SNLP to run in polynomial time, but these are different heuristics for different instances. There seems to be little hope of finding heuristics that help SNLP for a whole tractable problems class, *eg.* SAS^+-PUS, since SNLP simply cannot make use of all structure in the instances.

5 Modelling and analysis of hybrid systems

The group for modelling and analysis of hybrid systems consists of Dr. Simin Nadjm-Tehrani, graduate student Ulf Söderman (finishing his Ph.D. in May, 1995), in cooperation with Dr. Jan-Erik Strömberg in the Division of Automatic Control in our university.

By a hybrid model we mean a description of a system which includes both continuous and discrete elements. Earlier work in our group has used hybrid models in two different ways:

(1) as a means for describing the interacting behaviour of an embedded real-time computer system and its physical environment;

(2) as a means of modelling the behaviour of physical systems which exhibit discrete structural changes, typically due to existence of a switching device such as a gear-box or a relay.

The first line of research is intended to lead to formal analysis methods arriving at a proof of expected or desired properties of the embedded system. Interdependence between discrete and continuous system descriptions in terms of state transition systems have been studied [101]. This work has among others considered the case when one is satisfied with a description of physical quantities in terms of distinct ranges, such as high/ medium/ low or increasing/ constant/ decreasing. The question addressed was the loss of information when the continuous description is replaced by a discrete one [102]. An alternative to the systematic derivation of discrete models is the use of *Hybrid Transition Systems* for representation of the embedded system. This formalism incorporates differential and algebraic equations for description of continuous change in each (discrete) mode of the physical environment. Modular developmemt of models, and parallel composition is supported by the formalism.

In the second line of research we have developed an extension of the bond graph language by the introduction of a `switch` concept [103]. Bond graphs are a classical method in control engineering for describing the abstract behavior of a system, but traditional bond graphs can only describe continuous change. Our work has resulted in a methodology for the systematic modelling of physical systems which allows for discontinuous approximations of switching devices. In the final stage of this methodology a mathematical representation of a physical system is obtained in a variant of hybrid transition systems.

Hybrid transition systems have been applied for modelling several standard as well as some non-trivial examples. The current work in the group includes application of the modelling techniques in a realistic setting (see the description under the COHSY project). We are now in a position to study possible approaches to the problem of analysis. Automatic verification of hybrid systems is undecidable even for very restricted classes of systems. The aim of our work is to analyze a hybrid model using a combination of logical reasoning and traditional analysis techniques for continuous systems. Identifying interesting properties used as a requirements specification for hybrid systems is also an area for research.

6 Design methodology for intelligent autonomous systems

The group for design methodology for intelligent autonomous systems consists of Dr. Jacek Malec and graduate student Magnus Morin.

Intelligent autonomous systems, realized for example as robotic vehicles, provide an avant-garde application domain for the techniques that were described in previous sections. The synthesis of such systems involve a number of design issues, in particular:

- The choice of a hardware/software architecture which can serve as a backbone for the various contributing modules. In this context the use of layered architectures is often advocated.
- The choice of strategy guiding system implementation. The traditional model-based top-down approach, often adopted in AI and control engineering, is confronted with the bottom-up, behaviour-based strategy

proposed by Brooks. Recently, many compound (i.e. mixing elements of both approaches) policies have been proposed.

- The choice of language for specifying the expected reactive (stimulus-response) behaviour of the designed system. Assuming that a discrete description is sufficient, there exist a number of well-known languages to choose from: automata, statecharts, Petri nets, etc.

Rather than proposing and implementing additional approaches, we have focused on assessing properties of the existing ones. This research is using a generic three-layered software architecture as a frame of reference. We have participated in the systematic comparison of various architectures and modelling languages in the context of specific classes of applications. For this purpose we have also investigated computational models and real-time properties of the software tools associated with the layered architecture. The results we have obtained so far justify our choice and prove its usefulness in several application domains.

7 'Fuzzy'-related design principles for continuous and discrete control

The research is done by Dr. Dimiter Driankov in a close cooperation with Siemens AG, Corporate Research and Development, Munich, Germany and ABB Industrial Systems, Dept. LA, Västerås, Sweden.

Fuzzy control of continuous nonlinear processes has recently attracted widespread attention both within the process control industry and the conventional control community. Our group has been involved in research in the area of fuzzy control and fuzzy logic since 1988 which is manifested in a recent book on Fuzzy Control, as well as in a long list of research publications. Recent work has been concerned with design techniques for fuzzy control based on deep models of the process under control and utilizing the power of existing modern nonlinear and linear conventional methods, eg., fuzzy sliding mode control. This is in contrast with current work on fuzzy control which tends to use shallow models reflecting heuristic knowledge about the process behaviour and/or the related human operator control actions.

The present and future research focus is on the utilization of Lyapunov linearization for the systematic design of a particular class of fuzzy control rules (Takagi-Sugeno fuzzy controllers) that approximate in a smooth piecewise manner a nonlinear process and are capable of stabilizing a nonlinear process around an arbitrary set point. This approach makes possible the use of conventional linear control techniques for the purpose of stability analysis, optimal control design, and robustness analysis. We are also concerned with utilizing the principles of sliding mode fuzzy control for the simplification of the stability analysis of Takagi-Sugeno fuzzy controllers as well as the use of this type of fuzzy controllers within a gain scheduling framework. The research results will be published in 1996 by Springer Verlag in a book on model-based fuzzy control.

The second research topic of interest concerns general purpose PLC's (Programmable Logic Controllers) and the work is done in cooperation with ABB Industrial Systems, Dept. LA. A PLC can be abstracted as a discrete control algorithm executing different sequences of $\langle$condition, control action$\rangle$ pairs, where the 'control action' is executed whenever a predefined 'condition' holds. It is also expected that the prescribed effect of the control action (the 'action'-part of a consecutive $\langle$condition, control action$\rangle$ pair) will take place if and only if the 'modelling assumptions' used in the construction of the $\langle$condition, control action$\rangle$ pairs are valid and not violated during the execution of the control algorithm. Of particular interest are those modelling assumptions which are not represented in the control algorithm, the so called 'ontological assumptions'. Since the controller itself is "unaware" of the ontological assumptions, possible violations of these cannot be detected by it. A control paradigm which can be used to provide a proof that the ontological assumptions are wrong or violated and thus can decide whether these assumptions underlying a control algorithm are adequate for achieving prespecified control goals is called *ontological control*. The major goal of our work is to develop the formal means and implementation techniques for the design of an *ontological controller*. Such a controller is an autonomous device intended to supervise a conventional PLC in order to:

- detect dynamically when the PLC acts in a situation where the ontological assumptions underlying the control algorithm are wrong or violated, and

- when possible, to move the control system into a state from which it can regain its control over the environment and still achieve a prespecified goal despite previous violations of ontological assumptions.

8 Areas of applied research in our group

The following are brief descriptions of those applied research projects where we are presently engaged.

8.1 Driver Assistance and Local Traffic Management (DALTM) (Malec)

We are participating in a project within the Driver Assistance and Local Traffic Management area of the Swedish RTI programme, with Volvo Cars, SAAB Automobile, the Swedish National Road Administration (Vägverket) and NUTEK as the industrial partners. The project is aimed at designing and implementing a *driver information unit* intelligently managing the information and warnings passed to the driver by several active modules mounted in a test car (autonomous intelligent cruise control, friction estimation, communication with roadsid, etc.). To this end we use the software architecture and design methodology developed in our group (and mentioned earlier in section 6). The prototype system has been successfully tested in the actual motorway traffic in late 1994.

8.2 Hybrid Modelling and Verification of COmplex Heterogenous SYstems (COHSY) (Nadjm-Tehrani)

The work in this project aims to integrate modelling, synthesis and analysis techniques rooted in different disciplines; i.e. various engineering domains (hydraulic, mechanical, electrical, automatic control), and computer science (software development methods, formal techniques, database technology). The industrial participants are SAAB military aircraft, Volvo Aerospace and VOAC Hydraulics.

The common aspect of the different applications is the complexity arising due to the interaction of embedded computer systems with the physical environment and the human operator. The distributed software modules perform tasks of control, chronicle supervision and diagnosis.

In our part in the project we have so far studied decompositions of the complex system in accordance with the software architecture referred to in section 6, and application of formal modelling and verification techniques (including those described in section 5) in order to prove safety and timeliness properties of some reactive control modules. The novelty of the approach has been to incorporate the model of the physical environment as derived by bond graph techniques, for the purpose of analysing the properties of the embedded software.

8.3 Marine Diesel Engines (Driankov)

Mr. George Fodor from ABB Industrial Systems AB, Dept. LA is completing a PhD thesis under the supervision of Dr. Driankov. The thesis defines the basic principles, techniques, and implementation issues underlying the design of a general purpose 'ontological controller'. The general purpose ontological controller will be applied to the new DEGO III control unit for marine diesel engines.

9 Relationships between the research topics

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. The work described in sections 3 and 4 represent two different directions of development from the classical perspective. In section 4, 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 work described in section 3, 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 1), 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 "Features and Fluents" 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 "Features and fluents" approach and the planning complexity work in our group. Also, the technique of dividing the set of state variables into partitions has been found useful in the logic-based 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 and Jonsson 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.

10 Additional information

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

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

and by remote ftp on

`ftp.ida.liu.se`

from the RKLLAB ftp directory,

`/pub/labs/rkllab`

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

`/pub/labs/rkllab/general`

for general material about the laboratory. The present overview is updated from time to time; the present version can be accessed as

`/pub/labs/rkllab/general/annrep95.ps.gz`

or from the WWW home page.

References

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

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

[3] 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.

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

[5] 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.

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

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

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

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

[10] 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.

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

References for section 3

[12] 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.

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

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

[15] 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.

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

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

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

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

[29] 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.

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

[31] 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.

[32] W. Łukaszewicz and E. Madalińska-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.

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

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

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

[36] 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.

[37] E. Sandewall. *Features and Fluents: A Systematic Approach to the Representation of Knowledge about Dynamical Systems.* Oxford University Press, 1994.

[38] 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.

[39] 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.

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

References for section 4

[41] James Allen, James Hendler, and Austin Tate, editors. *Readings in Planning.* Morgan Kaufmann, San Mateo, CA, 1990.

[42] Roland Almgren. *Topological Modelling of Assembly Systems for General Applicability of Automatic Assembly Planning.* PhD thesis, Linköping University, Linköping, Sweden, 1994.

[43] American Association for Artificial Intelligence. *Proceedings of the 9th (US) National Conference on Artificial Intelligence (AAAI-91)*, Anaheim, CA, USA, July 1991. AAAI Press/MIT Press.

[44] American Association for Artificial Intelligence. *Proceedings of the 10th (US) National Conference on Artificial Intelligence (AAAI-92)*, San José, CA, USA, July 1992.

[45] American Association for Artificial Intelligence. *Proceedings of the 12th (US) National Conference on Artificial Intelligence (AAAI-94)*, Seattle, WA, USA, July–August 1994.

[46] Karl-Johan Åström and Albert Benveniste. Meeting the challenge of computer science in the industrial applications of control: An introductury discussion to the special issue. *IEEE Transactions on Automatic Control*, 38(7):1004–1010, July 1993.

[47] Karl-Johan Åström, Albert Benveniste, et al. Facing the challenge of computer science in the industrial applications of control: a joint IEEE CSS–IFAC project. Progress Report, March 1991.

[48] Fahiem Bacchus and Qiang Yang. Downward refinement and the efficiency of hierarchical problem solving. *Artificial Intelligence*, 71, 1994.

[49] Christer Bäckström. Action structures with implicit coordination. In Tim O'Shea and Vasil Sgurev, editors, *Proceedings of the 3rd International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA-88)*, pages 103–110, Varna, Bulgaria, September 1988. North-Holland.

[50] Christer Bäckström. A representation of coordinated actions characterized by interval valued conditions. In Zbigniew W Ras and Lorenza Saitta, editors, *Proceedings of the 3rd International Symposium on Methodologies for Intelligent systems (ISMIS-88)*, pages 220–229, Torino, Italy, October 1988. North-Holland.

[51] Christer Bäckström. *Computational Complexity of Reasoning about Plans.* Doctoral dissertation, Linköping University, Linköping, Sweden, June 1992.

[52] Christer Bäckström. Equivalence and tractability results for SAS$^+$ planning. In Bill Swartout and Bernhard Nebel, editors, *Proceedings of the*

3rd International Conference on Principles on Knowledge Representation and Reasoning (KR-92), pages 126–137, Cambridge, MA, USA, October 1992. Morgan Kaufmann.

[53] Christer Bäckström. Uncertainty and parallelism in action structures. In Gerd Große, editor, *ECAI-92 Workshop: Beyond Sequential Planning*, Vienna, Austria, Aug 1992.

[54] Christer Bäckström. Finding least constrained plans and optimal parallel executions is harder than we thought. In Christer Bäckström and Erik Sandewall, editors, *Current Trends in AI Planning: EWSP'93—2nd European Workshop on Planning*, pages 46–59, Vadstena, Sweden, December 1993. IOS Press.

[55] Christer Bäckström. Executing parallel plans faster by adding actions. In Anthony G. Cohn, editor, *Proceedings of the 11th European Conference on Artificial Intelligence (ECAI-94)*, pages 615–619, Amsterdam, Netherlands, August 1994. Wiley.

[56] Christer Bäckström. Expressive equivalence of planning formalisms. *Artificial Intelligence, Special Issue on Planning and Scheduling*, 1994. To appear.

[57] Christer Bäckström. Planning using transformation between equivalent formalisms: A case study of efficiency. In David Wilkins, editor, *Comparative Analysis of AI Planning Systems*, 1994. Held in conjunction with AAAI-94 [45].

[58] Christer Bäckström and Peter Jonsson. Planning with abstraction hierarchies can be exponentially less efficient. Submitted to IJCAI'95.

[59] Christer Bäckström and Inger Klein. Planning in polynomial time. In G Gottlob and W Nejdl, editors, *Expert Systems in Engineering: Principles and Applications. International Workshop*, volume 462 of *Lecture Notes in Artificial Intelligence*, pages 103–118, Vienna, Austria, September 1990. Springer.

[60] Christer Bäckström and Inger Klein. Parallel non-binary planning in polynomial time. In Reiter and Mylopoulos [94], pages 268–273.

[61] Christer Bäckström and Inger Klein. Planning in polynomial time: The SAS-PUBS class. *Computational Intelligence*, 7(3):181–197, August 1991.

[62] Christer Bäckström and Bernhard Nebel. On the computational complexity of planning and story understanding. In Bernd Neumann, editor, *Proceedings of the 10th European Conference on Artificial Intelligence (ECAI-92)*, pages 349–353, Vienna, Austria, August 1992. Wiley.

[63] Christer Bäckström and Bernhard Nebel. Complexity results for SAS^+ planning. In Ruzena Bajcsy, editor, *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, Chambéry, France, August–September 1993. Morgan Kaufmann.

[64] Christer Bäckström and Bernhard Nebel. Complexity results for SAS$^+$ planning. *Computational Intelligence*, 1993.

[65] Ronald J Brachman and Hector J Levesque. The tractability of subsumption in frame-based description languages. In *Proceedings of the 4th (US) National Conference on Artificial Intelligence (AAAI-84)*, pages 34–37, Austin, TX, USA, 1984. American Association for Artificial Intelligence.

[66] Tom Bylander. Complexity results for planning. In Reiter and Mylopoulos [94], pages 274–279.

[67] Tom Bylander. Complexity results for extended planning. In Hendler [81], pages 20–27.

[68] Tom Bylander. Complexity results for serial decomposability. In AAAI-92 [44], pages 729–734.

[69] Tom Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69:165–204, 1994.

[70] Tom Bylander, Dean Allemang, Michael C Tanner, and John R Josephson. Some results concerning the computational complexity of abduction. In Ronald J Brachman, Hector J Levesque, and Raymond Reiter, editors, *Proceedings of the 1st International Conference on Principles on Knowledge Representation and Reasoning (KR-89)*, pages 44–54, Toronto, ON, Canada, May 1989. Morgan Kaufmann.

[71] David Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32:333–377, 1987.

[72] Stephen V. Chenoweth. On the NP-hardness of blocks world. In AAAI-91 [43], pages 623–628.

[73] Ken Currie and Austin Tate. O-Plan: The open planning architecture. *Artificial Intelligence*, 52:49–86, 1991.

[74] Thomas Dean and Mark Boddy. Reasoning about partially ordered events. *Artificial Intelligence*, 36:375–399, 1988.

[75] Francesco M Donini, Maurizio Lenzerini, Daniele Nardi, Bernhard Hollunder, Werner Nutt, and Alberto M Spaccamela. The complexity of existential quantification in concept languages. *Artificial Intelligence*, 53:309–327, 1992.

[76] Kutluhan Erol, Dana S Nau, and V S Subrahmanian. On the complexity of domain-independent planning. In AAAI-92 [44], pages 381–386.

[77] Kutluhan Erol, Dana S Nau, and V S Subrahmanian. When is planning decidable? In Hendler [81], pages 222–227.

[78] Richard E Fikes and Nils J Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[79] Michael Garey and David Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

[80] Naresh Gupta and Dana S Nau. On the complexity of blocks-world planning. *Artificial Intelligence*, 56:223–254, 1992.

[81] James Hendler, editor. *Artificial Intelligence Planning Systems: Proceedings of the 1st International Conference*, College Park, MD, USA, June 1992. Morgan Kaufmann.

[82] Peter Jonsson and Christer Bäckström. Tractable plan existence does not imply tractable plan generation. Submitted to IJCAI'95.

[83] Peter Jonsson and Christer Bäckström. Complexity results for state-variable planning under mixed syntactical and structural restrictions. In Philippe Jorrand, editor, *Proceedings of the 6th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA-94)*, Sofia, Bulgaria, September 1994. World Scientific Publishing.

[84] Peter Jonsson and Christer Bäckström. Tractable planning with state variables by exploiting structural restrictions. In AAAI-94 [45].

[85] Inger Klein, Peter Jonsson, and Christer Bäckström. Tractable planning for an assembly line. Submitted to IJCAI'95.

[86] Craig A. Knoblock. *Automatically Generating Abstraction Hierarchies for Problem Solving*. PhD thesis, Carnegie-Mellon University, Pittsburgh, PA, USA, May 1991. Available as Technical Report CMU-CS-91-120.

[87] Craig A Knoblock. Search reduction in hierarchical problem solving. In AAAI-91 [43], pages 686–691.

[88] Richard E Korf. Planning as search: A quantitative approach. *Artificial Intelligence*, 33:65–88, 1987.

[89] David McAllester and David Rosenblitt. Systematic nonlinear planning. In AAAI-91 [43], pages 634–639.

[90] Bernhard Nebel and Christer Bäckström. On the computational complexity of temporal projection and plan validation. In AAAI-92 [44], pages 748–753.

[91] Bernhard Nebel and Christer Bäckström. On the computational complexity of temporal projection, planning and plan validation. *Artificial Intelligence*, 66(1):125–160, 1994. ARTINT 1063.

[92] K M Passino and P J Antsaklis. A system and control theoretic perspective on artificial intelligence planning systems. *Applied Artificial Intelligence*, (3):1–32, 1989.

[93] Pierre Regnier and Bernard Fade. Complete determination of parallel actions and temporal optimization in linear plans of action. In Joachim Hertzberg, editor, *European Workshop on Planning*, volume 522 of *Lecture Notes in Artificial Intelligence*, pages 100–111, Sankt Augustin, Germany, March 1991. Springer.

[94] Ray Reiter and John Mylopoulos, editors. *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, Sydney, Australia, August 1991. Morgan Kaufmann.

[95] Earl D Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5(2):115–135, 1974.

[96] Erik Sandewall and Ralph Rönnquist. A representation of action structures. In *Proceedings of the 5th (US) National Conference on Artificial Intelligence (AAAI-86)*, pages 89–97, Philadelphia, PA, USA, August 1986. American Association for Artificial Intelligence, Morgan Kaufmann.

[97] Jan-Erik Strömberg. Styrning av LEGO-bilfabrik. andra omarbetade upplagan. Department of Electrical Engineering, Linköping University, 1991.

[98] Austin Tate. Generating project networks. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence (IJCAI-77)*, pages 888–893, Cambridge, MA, USA, August 1977. Reprinted in Allen *et al* [41], pages 291–296.

[99] Manuela M Veloso, M Alicia Pérez, and Jaime G Carbonell. Nonlinear planning with parallel resource allocation. In Katia P Sycara, editor, *Workshop on Innovative Approaches to Planning, Scheduling and Control*, pages 207–212, San Diego, CA, USA, November 1990. Morgan Kaufmann.

[100] David E Wilkins. *Practical Planning*. Morgan Kaufmann, San Mateo, CA, 1988.

References for section 5

[101] S. Nadjm-Tehrani. Reactive Systems in Physical Environments: Compositional Modelling and Framework for Verification. Dissertation No. 338, Dept. of Computer and Information Science, Linköping University, 1994. ISBN 91-7871-237-8.

[102] S. Nadjm-Tehrani and J-E. Strömberg. From Physical models to Compositional modelling of Hybrid Systems. Proc. of the 3rd. International Conference on Formal Techniques in Real-time and Fault-tolerant Systems. Lecture Notes in Computer Science, No 863, pp. 583–604. Springer Verlag, 1994.

[103] Ulf Söderman. On Conceptual Modelling of Mode Switching Systems. Licentiate thesis, Linköping University, 1995.