VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kansliets noteringar Kod	Dnr
2014-45145-115393-21	2014-4086

2014
Project Research Grant

Area of science
Natural and Engineering Sciences
Announced grants
Research grants NT April 9, 2014
Total amount for which applied (kSEK)

2015	2016	2017	2018	2019
1680	1715	1752	1789	1827

APPLICANT

Name(Last name, First name)
Bäckström, Christer

Date of birth
621230-1912

Gender
Male

Email address
christer.backstrom@liu.se

Academic title
Associate professor

Position
Universitetslektor

Phone
013-28 24 92

Doctoral degree awarded (yyyy-mm-dd)
1992-09-18

WORKING ADDRESS

University/corresponding, Department, Section/Unit, Address, etc.
Linköpings universitet
Institutionen för Datavetenskap, (IDA)
SaS

58183 Linköping, Sweden

ADMINISTRATING ORGANISATION

Administrating Organisation
Linköpings universitet

DESCRIPTIVE DATA

Project title, Swedish (max 200 char)
Ett mångdimensionellt angreppssätt för komplexitetsanalys inom planering

Project title, English (max 200 char)
A Multi-dimensional Approach to Complexity Analysis in Planning

Abstract (max 1500 char)
Planning is to find a sequence of actions achieving a specified goal. Finite domain planning is PSPACE-complete, ie. it is considerably harder than the NP-complete problems, cf. SAT and TSP. Investigating the complexity of restricted subclasses of planning and understanding what factors contribute to the hardness of the problem is very important, indirectly also for other PSPACE-complete problems. Previous complexity studies of planning used standard complexity theory, giving a very coarse and not very informative picture. We will use a large number of different methods for complexity analysis, combining the results in order to get a much more fine-grained and informative view. We will both use established methods, cf approximation theory and parameterised complexity, and more specialized novel methods, cf padding with limited non-determinism and studying compact plan representations. As a spin-off effect, these novel methods will be useful also for analysing other PSPACE-complete problems. Achieving such refined results is important for the theroretical foundations of planning to understand why and when planning is difficult. The research is also very important for practical planning, since planning is starting to become commercially important for applications in computer security and system verfication. These are large-scale applications that are impossible to cope with efficiently without a thourough theoretical understanding of the planning problem.

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2014-45145-115393-21

Name of Applicant
Bäckström, Christer

Date of birth
621230-1912

Abstract language
English

Keywords
Automated Planning, Computational Complexity ,

Review panel
NT-2

Project also includes other research area

Classification codes (SCB) in order of priority
10201

Aspects

Continuation grant
Application concerns: New grant
Registration Number:

Application is also submitted to

similar to: **identical to:**

ANIMAL STUDIES

Animal studies
No animal experiments

OTHER CO-WORKER

Name(Last name, First name) **University/corresponding, Department, Section/Unit, Addressetc.**
,

Date of birth **Gender**

Academic title **Doctoral degree awarded (yyyy-mm-dd)**

Name(Last name, First name) **University/corresponding, Department, Section/Unit, Addressetc.**
,

Date of birth **Gender**

Academic title **Doctoral degree awarded (yyyy-mm-dd)**

Name(Last name, First name) **University/corresponding, Department, Section/Unit, Addressetc.**
,

Date of birth **Gender**

Academic title **Doctoral degree awarded (yyyy-mm-dd)**

Name(Last name, First name) **University/corresponding, Department, Section/Unit, Addressetc.**
,

Date of birth **Gender**

Academic title **Doctoral degree awarded (yyyy-mm-dd)**

Kod
2014-45145-115393-21

Name of Applicant
Bäckström, Christer

Date of birth
621230-1912

ENCLOSED APPENDICES

A, B, C, N, S

APPLIED FUNDING: THIS APPLICATION

Funding period (planned start and end date)

2015-01-01 -- 2019-12-31

Staff/ salaries (kSEK)

Main applicant	**% of full time in the project**	**2015**	**2016**	**2017**	**2018**	**2019**
Christer Bäckström	80	586	601	616	631	647
Other staff						
New PhD student	80	391	400	410	421	431
	Total, salaries (kSEK):	977	1001	1026	1052	1078

Other project related costs (kSEK)	**2015**	**2016**	**2017**	**2018**	**2019**
Indirect costs	496	507	519	530	542
Occupancy costs	106	106	106	106	106
Computer support	26	26	26	26	26
Travel costs	75	75	75	75	75
Total, other costs (kSEK):	703	714	726	737	749

Total amount for which applied (kSEK)

2015	2016	2017	2018	2019
1680	1715	1752	1789	1827

ALL FUNDING

Other VR-projects (granted and applied) by the applicant and co-workers, if applic. (kSEK)

Funds received by the applicant from other funding sources, incl ALF-grant (kSEK)

POPULAR SCIENCE DESCRIPTION

Popularscience heading and description (max 4500 char)

Ett mångdimensionellt angreppssätt för komplexitetsanalys inom planering

Planering handlar om att välja en sekvens av handlingar, en plan, att utföra för att uppnå ett givet mål. Automatisk planering har studerats i över 40 år, och i takt med att man stött på praktiska problem med effektivitet så har den teoretiska förståelsen för planering, i synnerhet beräkningskomplexitet, blivit allt viktigare. Applikationer för planering var förr typiskt inom robotik och liknande områden där man försökte efterlikna människans förmåga att planera. Idag har planering börjat mogna som teknik och användas för nya och helt andra applikationer. Speciellt så börjar planering bli en kommersiellt viktig teknik inom områden som webbsystem, datorsäkerhet och systemverfiering. Dessa problem är ofta svåra i första hand för att de är mycket storskaliga och inte längre överblickbara för en människa. Sådana tillämpningar gör det ännu viktigare med en ökad förståelse för när planering är lätt respektive svårt och varför det är så.

Det föreslagna projektet syftar till en sådan ökad teoretisk förståelse av planeringsproblemet och olika begränsningar av det. Ur en teoretisk synvinkel är planering ett kombinatoriskt problem som i sin generella form är PSPACE-fullständigt, dvs. det är betydligt svårare än de flesta andra vanliga kombinatoriska problem, som typiskt är NP-fullständiga. De kartläggningar som hitills har gjorts av komplexiteten för olika begränsade klasser av planeringsproblem har bara använt traditionell komplexitetsteori, som är ett tämligen grovt verktyg. Att använda en enda analysmetod kan dessutom ses som en endimensionell analys. Vi kommer därför istället att använda en mångdimensionell analys, dvs. vi tänker använda ett stort antal olika metoder för komplexitetsanalys och kombinera resultaten för att på så vis uppnå en mycket mera detaljerad och informativ bild av när planering är lätt respektive svårt. Denna metodik bygger på att två problem som har samma komplexitet enligt en analysmetod ofta har olika komplexitet för någon annan analysmetod. Genom att använda ett stort antal olika analysmetoder får vi därför stora möjligheter att göra en väldigt finmaskig kartläggning av olika begränsade planeringsklasser. Vi avser att dessutom dela in projektet i två spår, beroende på vilka metoder som används. Det första spåret kommer att använda metoder som är väletablerade och ofta använda för andra problem än planering, t.ex. parametriserad komplexitetsanalys och approximationsteori, och som därför är någorlunda rättframma att tillämpa även på planering. Det andra spåret kommer att använda metoder som är mer eller mindre nya och ofta specialanpassade för planering, t.ex. att kombinera paddning med begränsad icke-determinism och att analysera komplexitet utifrån vilka komprimeringsegenskaper planerna har. Några av dessa metoder har vi själva utvecklat specifikt för att komma runt olika problem med komplexitetsanalys för planering, och det kommer i olika omfattning även att krävas vidareutveckling av själva metoderna. Motiveringen för dessa metoder är att planering har en sådan karaktär att det ofta är svårt att göra meningsfulla komplexitetsanalyser med traditionella metoder, och att det därför behövs nya och mera specialanpassade metoder. Som en viktig bieffekt kommer dessa nya metoder även att vara intressanta för att analysera andra liknande problem, i första hand andra PSPACE-fullständiga problem.

Den praktiska nyttan av teoretisk forskning av detta slag är att den ökade förståelsen för svårigheterna i planering bidrar till att förbättra existerande metoder för planering och till att utveckla nya och bättre sådana. T.ex. så har vi under det senaste deceniet sett en trend mot att använda planeringsalgoritmer som utnyttjar applikationsoberoende heuristiker. Sådana heuristiker baseras allt oftare på att lösa enkla delproblem och sedan kombinera dessa resultat till en heuristik för hela problemet. Det kan också vara så att vissa problem är "nästan enkla" och att svårigheten är helt avhängig en specifik parameter som vi kan ha under kontroll.

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2014-45145-115393-21
Name of applicant
Bäckström, Christer
Date of birth
621230-1912

Title of research programme
A Multi-dimensional Approach to Complexity Analysis in Planning

Appendix A

Research programme

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

A Multi-dimensional Approach to Complexity Analysis in Planning

Christer Bäckström*

This application is for a new project to be carried out by Dr. Christer Bäckström and one new graduate student. The work will be performed at the Laboratory for Theoretical Computer Science (TCSLAB), Dept. of Computer and Information Science (IDA), Linköping University.

1 Purpose and Aims

The goal of this project is to attain a better and more refined understanding of when and why finite-domain planning is hard and easy, respectively. Previous studies of the computational complexity of planning has almost exclusively used traditional complexity analysis, which is a very coarse tool. Using only one method can be viewed as a one-dimensional analysis. The proposed methodology of this project is to use a multitude of complementary methods for complexity analysis and combine the results, ie. a multi-dimensional analysis. Two problems having the same complexity when analysed with one method, ie. along one dimension, often have different complexity when anlysed with another method, ie. along another dimension. By using a large number of quite different methods we will get a very fine-grained picture of when there are fundamental differences in hardness between problems and what the reason for this is. Furthermore, we will use a combination of well established methods and more novel methods, many of which are still under development. The established methods we plan to use include approximation theory and parameterised complexity theory, while the novel methods include combining padding with limited non-determinism, indirectly studying complexity through compact representations and using abstraction. As an important spin-off result, these novel methods will also be useful for analysing many other problems that share some characteristics with planning. Planning is furthermore a very suitable test bench for developing new methods for analysing **PSPACE**-complete problems since it is a very flexible modelling language and it is often easy to model other problems as planning problems.

The practical relevance of complexity analysis for planning is steadily growing, in particular since planning is nowadays mature enough to be used commercially in large-scale applications. Theoretical results of the kind we settle for, and the deeper understanding that follows, are very important for improving existing planning algorithms as well as for developing new and better ones. For example, during the past decade we have seen a trend towards basing domain-independent heuristics on solving tractable fragments of an instance and then combine these results to form a heuristic value. There are also applications that are 'almost tractable' and where the hardness depends on the value of some specific factor. While theory has often lagged behing practical research in planning, this is likely to change in the future since intuition and ad hoc methods will not suffice for the large-scale demanding applications that are are emerging as planning is getting increasingly mature. The 'Google-size' systems today could hardly be imagined ten years ago, and efficiency is of paramount importance for such systems.

2 Survey of the Field

Planning is the problem of finding a plan of actions to achieve a specified goal, and has been an established research area since the early 1970's. The various efficiency problems encountered in practice have made the theoretical understanding of planning, and complexity issues in particular, an increasingly important topic in the literature. Planning is also closely related to many other areas, eg. sequential control and model checking. This section is divided into two parts, the first one briefly describes planning as a theoretical problem and the second part discusses some of the results from the literature that are most relevant for this project.

*Department of Computer Science, Linköping University, SE-581 83 Linköping, Sweden. Email: christer.backstrom@liu.se

Formal Model of Planning: The most common model is the following. There is a set V of variables with some finite domain D. This implicitly defines a state space S, containing one state for each combination of values for the variables. In addition there is a set A of actions, each defined by a precondition *pre* and a postcondition *post* specifying some subset of the variables and a value for each. For instance, an action a with precondition *pre*$(a) = \{x = 0, y = 2\}$ and postcondition *post*$(a) = \{x = 1, z = 0\}$ can be executed in any state where x has value 0 and y has value 2. The result is that x will change to value 1 and z will get value 0, whatever its previous value was, and all other variables remain unchanged. A specific planning instance also specifies an initial state in S and a goal, which is on the same form as an action precondition. A sequence $a_1, \ldots, a_\ell$ of actions is a plan for this instance if there is a sequence $s_0, \ldots, s_\ell$ in S such that s_0 is the initial state, the goal is satisfied in s_ℓ and for each i, where $1 \leq i \leq \ell$, action a_i can be executed in s_{i-1} and the result is s_i.

An alternative view is that a planning instance defines a directed graph G where the states are the vertices and there is an edge from a state s to a state t if there is some action that can be executed in s with t as result. We may thus define planning as the problem of finding a path in this graph, but contrary to the traditional path-finding problem, the graph G is not explicit but implicitly specified by the planning instance. If we have n variables with domain size d, then there are d^n states, ie. the state space is exponentially larger than its specification. Similarly, an action with k variables in its precondition can be executed in d^{n-k} different states, so each action specifies an exponential number of edges in the graph. Although standard algorithms, eg. Dijkstra's algorithm, can find a path in a graph in polynomial time, this would mean polynomial time in the size of a graph that is exponentially larger than the instance, which is clearly not very efficient. However, since the graph is implicitly specified it will not be an arbitrary graph but have some implicit structure that is possible to exploit for efficiency. Planning is about studying this problem and finding efficient algorithm for it.

This model of planning is often referred to as the SAS$^+$ formalism, while the special case with binary variables is referred to as (propositional) Strips. Historically, Strips came first and we [17] introduced SAS$^+$ as a generalisation of Strips. Although these formalisms are equivalent in the general case [2], this equivalence often breaks down for restricted cases and it has been repeatedly observed that non-binary variables are important since they enforce more structure to an instance [18, 35, 45, 32]. Although some variants of Strips and its derivatives, cf. PDDL, use logic predicates over some domain instead of variables, this is nowadays almost always restricted to finite domains, which corresponds to the propositional case by grounding. Strips planning with predicates over infinite domains is undecidable [22]. Planning is also often extended by more complex actions, but this does typically not make planning any harder as long as the actions are deterministc; it is **PSPACE**-complete already for Strips and stays **PSPACE**-complete also for most such extensions. Furthermore, even such a simple formalism as Strips is still very poorly understood from a theoretical perspective.

In practice, planning is often solved by general search-based algorithms, usually with further amendments, like heuristics or abstraction, to help improving performance in practice. Our interest is not in such general algorithms but in studying various restricted subclasses of planning and analyse whether efficient algorithms can exist or not for these classes. A related but different problem is to instead study the complexity of actual application or benchmark problems for planning [29, 31], rather than particular restrictions on the planning formalisms.

Complexity Results and Related Research: Although it was known that Strips planning is **PSPACE**-complete, complexity was not generally considered until we identified a tractable subclass of planning [17]. We did so by imposing four restrictions, P, U, B and S, on the SAS$^+$ formalism, based on an example in sequential control that was considered difficult. Later, we classified all combinations of these four restriction with respect to complexity [18], considering both optimal and non-optimal planning, and also extended this study to a number of restrictions of more structural character [35]. We further demonstrated how to use these results in practice by solving tractable parts of an instance separately and then combine the solutions [40].

Independently of our work, Bylander [20] provided complexity results for various subcases of Strips planning, restricting the number and polarity of the pre- and postconditions.

State abstraction is a common technique for first focusing on the essential part of an instance, by ignoring variables that are considered easier to plan for. Knoblock [41] developed an algorithm for automatically generating abstraction hierchies that can speed up planning exponentially in the best case. We proved that it may also slow down planning exponentially in the worst case [10], which resulted in a shift of focus from refinement planning (first finding an abstract plan and then filling in the details) to using abstraction for computing heuristics.

Knoblock's algorithm was based on exploiting the causal graph, an explicit model of the variable dependencies that are implicit in the actions. Causal graphs were independently exploited also by others. Williams and Nayak [45] derived a planning instance and its causal graph from a causal model of a spacecraft system). They enforced acyclic graphs and used only actions that change exactly one variable, our U restriction, claiming that this is often sufficient in many real applications. The graph was then used to automatically choose actions and an execution order on them. We defined the 3S class [36] by imposing restrictions on the variables and the causal graph. 3S instances may have shortest plans of exponential length, which necessarily takes exponential time to generate, yet it is possible to decide in polynomial time if there is a plan or not. This result demonstrates that one must be very careful to distinguish between the decision problem, finding *if there is* a plan, and the generation problem, *finding* a plan, in the context of planning. Giménez and Jonsson [26] further showed that it is always possible to generate a polynomial-size macro representation of a plan for a 3S instance in polynomial time, which makes the issue even more intricate. These different ways of exploiting causal graphs convereged in the work by Brafman and Domshlak [19], who combined our idea of defining tractable subclasses by restricting the structure of the causal graph with the observations of Williams and Nayak. This has continued to be an active research area with many further results both for non-optimal planning [23, 27, 28] and optimal planning [37].

Heuristics for planning algorithms were hand-crafted and domain-specific until it was demonstrated that also domain-independent heuristics can be efficient [30]. A large number of different domain-independent heuristics and planners that exploit them have appeared over the past decade, to a large extent driven by the International Planning Competititions (IPC). Computational complexity is important also for heuristics, since they must be efficient to compute. One of the most succesful heuristics is h^+ [34]. This is **NP**-hard to compute so it must be further approximated in practice. Helmert [32] combined domain-independent heuristics with causal graphs, defining a heuristic computed recursively from solving tractable subgraphs. An important aspect of Helmert's approach was to use non-binary variables in order to get a more informative graph structure. This approach was used in his very succesful Fast Downward planner [33] and it has been extended to using many different types of tractable subgraph structures to compute a heuristic and coupling this to pattern database heuristics [38].

3 Project Description

The primary purpose of this project is to provide a much more fine-grained and more informative complexity classification of planning subclasses than in previous research. From a theoretical perspective, this will lead to a deeper understanding of planning, and why and when it is difficult, while from a practical perspective it will contribute to making complexity analysis more useful and relevant for improving existing planning methods and inventing new ones. Previous studies of this type in planning has almost only employed standard complexity analysis, while we will use a multitude of different methods in order to achieve finer separations than otherwise possible. Some of these methods are well known from other areas than planning, while others are of more novel character, often specialized for planning and similar problems. An important spin-off effect of this will be that the methodology and the novel methods will be useful also for analysing other **PSPACE**-complete problems, since planning has favourable properties that makes it a good representative for the whole class of problems in **PSPACE**; for every complexity class C in **PSPACE** that is closed under polynomial reductions, there is a corresponding C-complete class of planning instances, i.e. planning can capture any reasonable complexity class in **PSPACE** [1].

We propose an approach of two parallel tracks, one for the more established methods and one for the more novel methods. This has the advantage that the first track is straightforward enough to guarantee a large number of results and to be suitable for the student. The second track is more opportunistic, and we expect it to contribute novel results and analysis techniques that will be useful in a much broader context than just planning. Our preliminary research has already contributed such new results to the theory of parameterised complexity.

We propose a project time of five years, for two reasons: It increases the chances that the second track will not only contribute new results but also new methods of general interest, and it allows the new student to finish a PhD within the time frame of the project. The following subsections describe the methodology and the methods we intend to use.

Methodology: Our proposed methodology is to combine many different methods for complexity analysis in order to get a more refined and informative picture of the computational complexity of different subclasses of planning than otherwise possible. As a non-planning example, consider Minimum Vertex Cover and Minimum Dominating Set, two closely related graph problems which are both **NP**-complete to solve optimally. The first problem can be approximated within a factor 2 in polynomial time, while the second one cannot even be approximated within a factor $c \log n$ for some c, where n is the number of vertices. Approximation theory can thus separate two otherwise indistinguishable problems. Parameterised complexity can be similarly used. Minimum Vertex Cover is in **FPT**, i.e. tractable in the parameterised sense, while Minimum Dominating Set is **W**[2]-complete, which is much harder.

Approximation analysis and parameterised analysis are standard tools today for studying, primarily **NP**-complete, problems. Planning is **PSPACE**-complete, though, so we must consider a much larger number of restricted subproblems. Such classifications of subclasses of planning have been done, but they have almost invariably used only standard complexity analysis, classifying the problems as tractable, **NP**-complete or **PSPACE**-complete. We will use a large number of methods to maximise the number of cases where problems can be separated by at least one of the methods, to allow for a much more fine-grained separation between subclasses than possible with any method alone. In addition, depending on which method(s) can be used to separate two problems we will often also be able to say something about why one of the problems is harder.

The obvious subclasses of planning to start with are the ones that have been studied before in the literature, with standard complexity. Which ones to continue with is best left as an open question, to be answered incrementally during the project, depending on the results we get. The separations we prove and fail to prove is the best guide guide for selecting subclasses incrementally. A failure to separate two subclasses indicates that the restrictions used are irrelevant or inadequate, and analysing the reasons for the failure will often help to identify new and better restrictions.

Methods of First Track: *Parameterised complexity* is an alternative to standard complexity analysis. Standard problem complexity measures time (or space) only as a function of the instance size, and a problem is considered tractable if it can be solved in $O(n^c)$ time for some constant c, where n is the size of the instance. If a problem has a numeric parameter k, then k is treated as a part of the instance so only its representation size matters, not its value. Parameterised complexity instead treats the instance size and the parameter separately, as two dimensions. A problem is *fixed parameter tractable* if it can be solved in $O(f(k)n^c)$ time for some computable function f and some constant c, ie. the complexity is separated into a hard part, $f(k)$, and an easy part, n^c, which do not depend on each other. While the total expression is typically not polynomial in k, a problem may still be tractable in practice if k is small compared to n and f grows moderately fast.

Parameterised hardness is based on fixed-parameter tractable reductions instead of polynomial reductions This results in a hierarchy of complexity classes **FPT** $\subseteq$ **W**$[1]$ $\subseteq$ **W**$[2], \ldots,$ where **FPT** is the class of all fixed-parameter tractable problems and **W**[1] is considered as a parameterised analogue of **NP**. However, no relationships between these classes and the standard ones is known, except that **P** $\subseteq$ **FPT**. In our case, this is largely an asset, since we get two orthogonal dimensions of classes which increases the possibilities to prove separation in at least one of them.

For standard complexity analysis, we only need to decide which problems to analyse, while for parameterised analysis, we must also decide which parameter(s) to consider. This opens up for more interesting and more fine-grained analyses, since two different subclasses of planning may both be fixed-parameter tractable, but for different parameters, ie. the tractability is not inherent in the problem itself, but in the combination of problem and parameter. So far, only obvious parameters like plan length, domain size, maximum number of pre- and post-conditions of actions and similar have been considered in the literature [4, 42]. It is more interesting, though, to look at parameters that are not immediately visible, but that must be tested/computed. Treewidth of the causal graph has been suggested [24] and we have recently combined this with properties of the domain-transition graphs for the variables to identify new classes in **FPT** [3]. This approach will be pursued within our ongoing cooperation with Prof. Stefan Szeider (Technical U., Vienna) and Dr. Sebastian Ordyniak (Masaryk U., Brno).

Approximation theory studies how hard or easy it is to approximate hard optimisation problems. A problem is approximable within a factor α if it is possible to find a solution at most α times larger than the optimum in polynomial time, and it is not approximable within α if no such algorithm can exist unless **P** $=$ **NP**. Ideally we want to show that a problem is approximable within some factor α but not within some other factor $\alpha' < \alpha$, such that the gap between α and α' is very small. The approximation factor is often a function of some parameter of the instance; for a graph, this may be the number of vertices or the number of edges. Identifying parameters interesting and relevant parameters is thus something that this approach has in common with parameterised analysis, so although the two techniques are quite different, they are obvious to use in parallel, trying the same parameters for both. Planning poses new theoretical challenges since the literature on approximation focuses heavily on **NP**-complete problems, not paying much attention to harder problems in **PSPACE**.

Non-uniform complexity is based on *advice taking Turing machines* instead of standard ones. This technique can be used to prove hardness results where no ordinary polynomial reduction is possible and also to prove stronger hardness results in some cases. The non-uniform complexity classes have well known relationships to the usual complexity classes.

The *exponential time hypothesis (ETH)* is increasingly used in complexity analyses. Somewhat simplified, the ETH postulates that there is some constant c such that the SAT problem cannot be solved in $O(2^{cn})$ time, where n is the number of variables, ie. that SAT cannot be solved in subexponential time. Using the ETH can allow for proving intractability results that are otherwise difficult or impossible to prove and it can also allow for more precise results than just a classification into the usual complexity classes.

Methods of Second Track: *Padding* refers to representing one or more numbers in unary notation, ie. using the same number of bits as the value of the number. A problem with a numeric parameter k is **NP**-complete in the *strong* sense if it remains **NP**-complete when k is represented in unary. This can be used to emphasize or deemphasize various inner structural parts of an instance. We introduced the technique of padding planning instances with the plan length [11], which supresses the influence of the plan length on the complexity of planning and makes the contribution of other factors more visible. This technique is particularly useful to separate **PSPACE**-complete problems from each other. For instance, Strips planning is **PSPACE**-complete, because optimal plans may be of exponential length, but it remains **PSPACE**-complete even when allowing actions with conditional effects and quantified formulae in the preconditions. Our padding technique can distinguish such cases and demonstrate a spectrum of problems of varying hardness that are otherwise all just **PSPACE**-complete.

Limited non-determinism [39] treats non-determinism as a resource separate from time and space, allowing for more fine-grained complexity classes than the usual ones. A problem is in **NP** if there is some polynomial p such that every instance of size n can be solved in $p(n)$ time on a non-deterministic Turing machine M, i.e. M can guess at most $p(n)$ bits and then use at most $p(n)$ time to verify this guess. The guessing corresponds to a deterministic search space of size $2^{p(n)}$ at most. If the problem is instead in the limited non-deterministic class $f(n)$-**P**, for some specific function f, then there is some polynomial p such that every instance of size n can be solved in $p(n)$ time on some non-deterministic Turing machine M that guesses at most $f(n)$ bits. This is a sharper classification since it corresponds to a search space of size $2^{f(n)}$ at most and $f(n)$ may be a small polynomial or even a sublinear function, eg. log n-**P** $=$ **P**. Limited non-determinism can be used directly, but it seems most interesting to use in combination with our padding technique [11].

Compact representations of plans should be considerably smaller than corresponding explict representations. Although this is closely related to string compression, plans can be assumed to have a certain structure that can be exploited, while strings are assumed to contain arbitrary data. This topic is little studied, except for macro plans that have a long history in planning for various purposes. It is known that some classes of planning instances that have exponential-length optimal plans always admit polynomial-size macro-plans [26]. We have introduced the technique of determining the complexity of planning indirectly from the properties of plan representations [6]. Let ρ be some type of compact plan representation, let P be some planning class where each solvable instance of size n has a plan with a ρ-representation of polynomial size in n. Suppose that plan validation for ρ-representations for P is in some complexity class C. Then planning for P is in **NP**^{C} (guess a polynomial-size ρ-representation, then use an oracle for C to verify it). While other approaches to studying restricted planning classes tend to classify them only into the most common complexity classes like **P**, **NP** and **PSPACE**, this technique allows for an interesting way to identify planning classes at different levels of the polynomial hierarhcy, between **NP** and **PSPACE**-complete. It is common to encode planning instances as SAT instances and use a SAT solver for planning. This requires encoding a 'skeleton' for the plan in the instance, making the plan size influence the instance size. However, SAT solvers are usually more efficient the more compactly the instance is encoded, so it has been suggested to encode planning instances as QSAT instances to allow for even more compact encodings [21]. Compact plan representations are obviously very important for such planning methods. This approach will be purspued within our ongoing cooperation with Dr. Anders Jonsson (Universitat Pompeu Fabra, Barcelona).

Compilation of planning instances was introduced by Nebel [43], to study when different planning formalisms are equally expressive or not, by observing how much the size of an instance blows up when compiling it from one planning formalism into another. Nebel distinguished between linear, polynomial and super-polynomial blow-up. We had earlier proved that the SAS$^+$ formalism is equivalent to three common variants of Strips in the sense that there are polynomial reductions between them that preserve the size of solutions exactly [2]. A compilation that combines these two methods would take both instance blow-up and plan blow-up into account, which would have interesting connections with the complexity of planning.

Abstraction views a problem instance as consisting of a hard part and an easy part, where we first abstract away the easy part and focus on solving the hard part, and then 'fill in' by solving also the easy part. This has obvious similarities with the concept of kernels in parameterised complexity. A kernel is a mapping of an instance to a new instance of the same (or another) problem, In particular, it is a polynomial kernel if every instance is mapped to an instance of size polynomial in the parameter. An abstraction can, thus, be viewed as a kernel, which immediately leads to the question of how abstractions and kernels relate to each other. Further investigations into this relationship is likely to lead to interesting results that can benefit both the area of abstraction and parameterised complexity. This research will be pursued in cooperation with Prof. Robert Holte (U. of Alberta) and Dr. Sandra Zilles (U. of Regina).

4 Significance

This project is very significant both for the theory of planning and for practical applications in the following ways:

Contrary to most previous complexity analyses in planning we will step outside standard complexity theory and use a number of alternative methods and combine these, which will enable us to advance the research in this area considerably. Complexity analysis in planning has often been accused of being irrelevant and uninteresting because 'everything interesting is **PSPACE**-complete anyway'. We already show in our preliminary studies that we can get around this in many different ways, separating different **PSPACE**-complete problems from each other. Furthermore, many established techniques like parameterised complexity and approximation theory have mostly been succesful on **NP**-complete problems so we will have to find new ways of applying these techniques to maximise their usefulness also for planning. We will also use and develop novel methods specifically directed towards the difficulties previously encountered in analysing the complexity of planning, thus making this approach highly original. As a spin-off effect we expect these new methods to also be useful in a much broader context than planning, in particular for other **PSPACE**-complete problems like QSAT and various game and Petri net problems. Many combinatorial problems have been analysed by several methods before, but the common methodology has been to focus on one single method at a time and analyse many different problems with it, thus not systematically combining methods. We will instead focus on one single problem, planning, and use many different methods to systematically analyse it and combine the results. Our methodology thus defines an entirely new research area which will provide new synergetic effects between different methods and areas of complexity analysis.

Planning has traditionally assumed applications of the type humans are good at, with robotics as an archetypical example. The world and the actions may be complex, but the instances and the plans are typically small. However, a rapidly growing application area is planning within digital systems. For instance, an intelligent web interface may compose a plan for using and combining different web services to perform a task. Planning is also used in computer security for penetration testing of computer systems. Based on the knowledge of the hardware and the software used, a planner tries to create a plan that can intrude into the system by exploiting the known security issues. The computer-security company Core Technologies considers planning as a very important technique of high commericial value [44]. There are also close ties between planning and system verification [25], also an area of growing commercial importance. Typical for these applications is that they often deal with very large systems, in terms of the number of variables required to model them. This is a type of applications where humans are no longer very good at planning, since the complexity stems mainly from the large size of the instance, rather than from complex actions and similar features. This also means that computational complexity becomes an increasingly important tool both for understanding the problems and how to solve them efficiently.

5 Preliminary Results

We have provided a complete map of the parameterised complexity for all combinations of the PUBS restrictions for SAS$^+$ and for all combinations of pre- and postconditions for Strips [4, 15, 14]. These results combined with the previously known results gives a more detailed classification than either method alone, and neither method specializes the other. For instance, SAS$^+$-U is **W**[1]-complete and SAS$^+$-S is **W**[2]-complete, but they are both **PSPACE**-complete, while SAS$^+$-US is **NP**-complete and SAS$^+$-UB is **PSPACE**-complete, but they are both **W**[1]-complete. For those subproblems that are in **FPT** (ie. fixed-parameter tractable), we have further proved that none of these admits a polynomial kernel. The theorems in the literature that are commonly used for such proofs were not applicable since they assume problems in **NP**. Hence, we had to first derive new and more general versions of these 'kernel theorems' [15, 14], thus contributing to the foundations of parameterised complexity theory. In recent work, we also identify some new classes in **FPT** based on the parameters domain size, treewidth of the causal graph and the number of paths in the domain transition graphs [3].

We have analysed the approximability and non-approximability properties of Strips planning with positive postconditions under all combinations of the number of pre- and postconditions [8]. This is immediately relevant for approximating the popular and very succesful h^+ heuristic, which is based on ignoring negative post-conditions.

We have used non-uniform complexity to provide strong hardness results for a particular type of compact representation and for reformulation of planning [12]. We have further used non-uniform complexity to prove a strong hardness result for families of instances with certain 'polynomial sparsity' properties of the causal graph [13]. This 'sparsity' avoids a problem that makes many similar results difficult to apply to practical classes of application instances.

We introduced the technique of padding planning instances with the plan length [11]. This gives a new type of complexity measure where the basic case, Strips and SAS$^+$, is **NP**-complete instead of **PSPACE**-complete, but where the complexity increases with the addition of various language features, such as axioms, quantifiers in preconditions and conditional actions. This allows to distinguish an infinite spectrum of increasing complexity within the class of planning problems that are otherwise all **PSPACE**-complete, and thus indistinguishable. There are obvious connections between this technique and Nebel's compilations [43], although we only have partial results on this so far. We further combined padding with limited non-determinism and the ETH to make even finer distinctions possible. This also allowed for a well-founded conjecture that it is a simpler problem to generate a plan with an optimal parallel schedule directly than to first generate an optimally flexible plan and then find such a schedule.

We have defined and investigated two compact representations for plans with different access properties [12], and extended this to formally considering also macro plans and reactive plans as compact representation and compared all these concepts [6]. We have also presented the novel concept of *automata plans* [5], a plan representation based on finite automata which is strictly more expressive than macro representations, allowing for plans that are both more compact and more flexible, yet restrictive enough to have well defined theoretical properties. For instance, plan validation is **NP**-complete for automata plans so planning is $\mathbf{NP}^{\mathbf{NP}}$-complete. This is one of the first examples in the literature of a natural planning class higher up in the polynomial hierarchy than **NP**.

In order to find tighter couplings between abstraction and complexity for planning, we first need a better theoretical understanding of abstraction, in particular how the uses of abstraction in refinement and heuristics relate to each other. We have started to investigate this in a more formal manner than done before, by defining a general framework for modelling and analysing abstraction in search and planning. Using this framework, we have modelled and compared various abstraction methods in planning on a formal abstract level [7] and also analysed the connections and differences between refinement and heuristics in a formal way [9].

The planning community is currently very focused on problems which always have a solution, but many real applications are not like that. For instance, in penetration testing, the goal is to prove that there is no plan for intrusion into the system. We have recently brought attention to this neglect for unsolvable instances and demonstrated that a very simple algorithm can often be efficient where standard planners fail due to their heavy focus on finding a solution [16]. From a complexity point of view, this is a strange assymetry since there seems to be no fundamental difference in complexity between the two problems [1].

6 International and National Collaboration

We have active international cooperation with: Prof. Stefan Szeider, Prof. Reinhard Pichler, Andreas Pfandler and Mattias Kronegger, Inst. of Information Systems, Vienna University of Technology, Austria; Dr. Sebastian Ordyniak, Faculty of Informatics, Masaryk University, Brno, Czech Republic; Dr. Anders Jonsson, Dept. of Technology, Universitat Pompeu Fabra, Barcelona, Spain; Prof. Robert C. Holte, Faculty of Science, University of Alberta, Edmonton, Canada and Dr. Sandra Zilles, Dept. of Computer Science, University of Regina, Canada.

References

[1] Aghighi, M., Bäckström, C., Jonsson, P., and Ståhlberg, S. "ETH and planning" (working title). *Journal article in preparation*.

[2] Bäckström, C. Expressive equivalence of planning formalisms. *Artif. Intell. 76*, 1-2 (1995), 17--34.

[3] Bäckström, C. Parameterising the complexity of planning by the number of paths in the domain-transition graphs. Conference paper under review.

[4] Bäckström, C., Chen, Y., Jonsson, P., Ordyniak, S., and Szeider, S. The complexity of planning revisited - a parameterized analysis. In *Proc. 26th AAAI Conf. Artif. Intell. (AAAI 2012), Toronto, ON, Canada* (2012).

[5] Bäckström, C., Jonsson, A., and Jonsson, P. From macro plans to automata plans. In *Proc. 20th European Conf. Artif. Intell. (ECAI 2012)* (2012), pp. 91--96. (Extended journal version under review.).

[6] Bäckström, C., Jonsson, A., and Jonsson, P. Macros, reactive plans and compact representations. In *Proc. 20th European Conf. Artif. Intell. (ECAI 2012)* (2012), pp. 85--90.

[7] Bäckström, C., and Jonsson, P. Abstracting abstraction in search with applications to planning. In *Proc. 13th Int'l Conf. Principles Knowledge Repr. and Reasoning (KR 2012)*.

[8] Bäckström, C., and Jonsson, P. Approximation of monotone STRIPS planning. *Journal article in preparation*.

[9] Bäckström, C., and Jonsson, P. Bridging the gap between refinement and heuristics in abstraction. In *Proc. 23rd Int'l Joint Conf. Artif. Intell. (IJCAI 2013)*, pp. 2261--2267.

[10] Bäckström, C., and Jonsson, P. Planning with abstraction hierarchies can be exponentially less efficient. In *Proc. the 14th Int'l Joint Conf. on Artif. Intell. (IJCAI 1995)*, pp. 1599--1605.

[11] Bäckström, C., and Jonsson, P. All PSPACE-complete planning problems are equal but some are more equal than others. In *Proc. 4th Ann. Symp. Combinatorial Search, (SOCS 2011), Castell de Cardona, Barcelona, Spain* (2011). Best paper award.

[12] Bäckström, C., and Jonsson, P. Algorithms and limits for compact plan representations. *J. Artif. Intell. Res. (JAIR) 44* (2012), 141--177.

[13] Bäckström, C., and Jonsson, P. A refined view of causal graphs and component sizes: SP-closed graph classes and beyond. *J. Artif. Intell. Res. 47* (2013), 575--611.

[14] Bäckström, C., Jonsson, P., Ordyniak, S., and Szeider, S. A complete parameterized complexity analysis of bounded planning. *Journal article under review. (arXiv:1310.7828)*.

[15] Bäckström, C., Jonsson, P., Ordyniak, S., and Szeider, S. Parameterized complexity and kernel bounds for hard planning problems. In *Proc. 8th Int'l Conf. Algorithms and Complexity, (CIAC 2013), Barcelona, Spain, May* (2013), pp. pp. 13--24.

[16] Bäckström, C., Jonsson, P., and Ståhlberg, S. Fast detection of unsolvable planning instances using local consistency. In *Proc. 6th Int'l Symp. Combinatorial Search, (SoCS 2013), Leavenworth, WA, USA, July* (2013), pp. 29--37.

[17] Bäckström, C., and Klein, I. Planning in polynomial time: the SAS-PUBS class. *Comput. Intell. 7* (1991), 181--197.

[18] Bäckström, C., and Nebel, B. Complexity results for SAS^{+} planning. *Comput. Intell. 11* (1995), 625--656.

[19] Brafman, R. I., and Domshlak, C. Structure and complexity in planning with unary operators. *J. Artif. Intell. Res. (JAIR) 18* (2003), 315--349.

[20] Bylander, T. The computational complexity of propositional Strips planning. *Artif. Intell. 69*, 1-2 (1994), 165--204.
[21] Cashmore, M., Fox, M., and Giunchiglia, E. Partially grounded planning as quantified boolean formula. In *Proc. 23rd Int'l Conf. Automated Planning and Scheduling (ICAPS 2013)* (2013).
[22] Chapman, D. Planning for conjunctive goals. *Artif. Intell. 32*, 3 (1987), 333--377.
[23] Chen, H., and Giménez, O. Causal graphs and structurally restricted planning. *J. Comput. Syst. Sci. (JCSS) 76*, 7 (2010), 579--592.
[24] Downey, R., Fellows, M., and Stege, U. *Parameterized Complexity: A Framework for Systematically Confronting Computational Intractability*, vol. 49 of *DIMACS Series in Disc. Math. Theor. Comput. Sci.* 1999, pp. 49--99.
[25] Edelkamp, S., Leue, S., and Visser, W. Summary of Dagstuhl seminar 06172 on directed model checking. In *Directed Model Checking* (2007), Dagstuhl Seminar Proceedings.
[26] Giménez, O., and Jonsson, A. The complexity of planning problems with simple causal graphs. *J. Artif. Intell. Res. (JAIR) 31* (2008), 319--351.
[27] Giménez, O., and Jonsson, A. Planning over chain causal graphs for variables with domains of size 5 is NP-hard. *J. Artif. Intell. Res. (JAIR) 34* (2009), 675--706.
[28] Giménez, O., and Jonsson, A. The influence of k-dependence on the complexity of planning. *Artif. Intell. 177-179* (2012), 25--45.
[29] Gupta, N., and Nau, D. S. On the complexity of blocks-world planning. *Artif. Intell. 56*, 2-3 (1992), 223--254.
[30] Haslum, P., and Geffner, H. Admissible heuristics for optimal planning. In *Proc. 5th Int'l Conf. Artif. Intell. Planning Syst. (AIPS'00)* (2000), pp. 140--149.
[31] Helmert, M. Complexity results for standard benchmark domains in planning. *Artif. Intell. 143*, 2 (2003), 219--262.
[32] Helmert, M. A planning heuristic based on causal graph analysis. In *Proc. 14th Int'l Conf. Automated Planning and Scheduling (ICAPS 2004)* (2004), pp. 161--170.
[33] Helmert, M. The fast downward planning system. *JAIR 26* (2006), 191--246.
[34] Hoffmann, J. Where 'ignoring delete lists' works: Local search topology in planning benchmarks. *J. Artif. Intell. Res. (JAIR) 24* (2005), 685--758.
[35] Jonsson, P., and Bäckström, C. State-variable planning under structural restrictions: Algorithms and complexity. *Artif. Intell. 100*, 1-2 (1998), 125--176.
[36] Jonsson, P., and Bäckström, C. Tractable plan existence does not imply tractable plan generation. *Annals Math. Artif. Intell. 22*, 3-4 (1998), 281--296.
[37] Katz, M., and Domshlak, C. New islands of tractability of cost-optimal planning. *J. Artif. Intell. Res. 32* (2008), 203--288.
[38] Katz, M., and Domshlak, C. Implicit abstraction heuristics. *J. Artif. Intell. Res. 39* (2010), 51--126.
[39] Kintala, C., and Fischer, P. Computations with a restricted number of nondeterministic steps. In *9th ACM Symp. Theory Comput. (STOC'77)* (1977), pp. 178--185.
[40] Klein, I., Jonsson, P., and Bäckström, C. Efficient planning for a miniature assembly line. *AI in Engineering 13*, 1 (1999), 69--81.
[41] Knoblock, C. A. Automatically generating abstractions for planning. *Artif. Intell. 68*, 2 (1994), 243--302.
[42] Kronegger, M., Pfandler, A., and Pichler, R. Parameterized complexity of optimal planning: A detailed map. In *Proc. 23rd Int'l Joint Conf. Artif. Intell. (IJCAI 2013)*, pp. 954--961.
[43] Nebel, B. On the compilability and expressive power of propositional planning formalisms. *J. Artif. Intell. Res. (JAIR) 12* (2000), 271--315.
[44] Sarraute, C., and Pickering, K. Encounters of the third kind between pentesting and automated planning. In *Invited talk at AAAI 2012 workshop on Problem Solving Using Classical Planners, , Toronto, ON, Canada* (2012).
[45] Williams, B., and Nayak, P. P. A reactive planner for a model-based executive. In *Proc. 15th Int'l Joint Conf. Artif. Intell. (IJCAI'97), Nagoya, Japan* (1997), pp. 1178--1185.

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2014-45145-115393-21

Name of applicant
Bäckström, Christer

Date of birth
621230-1912

Title of research programme
A Multi-dimensional Approach to Complexity Analysis in Planning

Appendix B

Curriculum vitae

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

Curriculum Vitae

Christer Bäckström

1. Higher Education Qualification(s)

MSc in Computer Science and Technology (civ. ing. datateknik), Linköping University, 1986

2. Degree of Doctor

PhD in Computer Science (tekn. dr. i datalogi), Linköping University, 1992
Dissertation Title: Computational Complexity of Reasoning about Plans
Supervisor: Prof. Erik Sandewall

3. Postdoctoral Positions

4. Qualification Required for Appointment as Docent

Docent, 1996

5. Present Position

Associate Professor, Linköping University, since 1996 (permanent position)
Time for research: approx. 80 %

6 Previous Positions and Periods of Appointment

Assistant Professor, Linköping University, 1992--1995.

7. Interruption in Research

I was away from work from 1999-01-01 to 2010-08-31 due to serious illness. I am now back on full time and active as a researcher, as my recent publications demonstrate.

8. Supervision

I have been the supervisor for the following PhD students:

- Peter Jonsson, PhD 1996
- Thomas Drakengren, PhD 1997

9. Additional Information

Previously member of editorial board, Journal of Artificial Intelligence Research.

Recent journal reviewing for Journal of Artificial Intelligence Research and Fundamenta Informaticae.

Recent conference assignments:

- Senior programme committee:
  - 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013).
- Programme committee:
  - 28th AAAI Conference on Artificial Intelligence (AAAI 2014),
  - 24th International Conference on Automated Planning and Scheduling (ICAPS 2014),
  - 7th Annual Symposium on Combinatorial Search (SoCS 2014),
  - 23rd International Conference on Automated Planning and Scheduling (ICAPS 2013),
  - 6th Annual Symposium on Combinatorial Search (SoCS 2013),

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2014-45145-115393-21
Name of applicant
Bäckström, Christer
Date of birth
621230-1912

Title of research programme
A Multi-dimensional Approach to Complexity Analysis in Planning

Appendix C

List of publications

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

Publication List

Christer Bäckström

Citation database: Google Scholar (date: 2014-04-03)

Five most Cited Publications

Christer Bäckström and Bernhard Nebel **Complexity results for SAS$^+$ planning** *Computational Intelligence, 11(4):625--655, 1995.* Number of citations: 275

Christer Bäckström **Computational aspects of reordering plans** *Journal of Artificial Intelligence Research, 9:99--137, 1998.* Number of citations: 103

Peter Jonsson and Christer Bäckström **A unifying approach to temporal constraint reasoning** *Artificial Intelligence, 102(1):143--155, 1998.* Number of citations: 67

Peter Jonsson and Christer Bäckström **State-variable planning under structural restrictions: Algorithms and complexity** *Artificial Intelligence, 100(1--2):125--176, 1998.* Number of citations: 67

Christer Bäckström and Inger Klein. **Planning in polynomial time: The SAS-PUBS class** *Computational Intelligence, 7(3):181--197, 1991.* Number of citations: 49

All these articles are still relevant and have been cited during the past year. The first one has had 45 additional citations in the past year.

Publications During Past 8 Years

NB: I have been on leave from research, so during the past 8-year period I have only been active in research again since 2010.

1. Peer-reviewed Original Articles

Christer Bäckström and Peter Jonsson **A Refined View of Causal Graphs and Component Sizes: SP-closed Graph Classes and Beyond** *Journal of Artificial Intelligence Research, vol. 47, 2013, pp. 575-611.* Number of citations: 1 (*)

Christer Bäckström and Peter Jonsson **Algorithms and Limits for Compact Plan Representations** *Journal of Artificial Intelligence Research, vol. 44, 2012, pp. 141-177.* Number of citations: 3 (*)

2. Peer-reviewed Conference Contributions

Christer Bäckström and Peter Jonsson **Bridging the Gap Between Refinement and Heuristics in Abstraction** In *proc. 23rd International Joint Conference on Artificial Intelligence (IJCAI-2013), Beijing, China, Aug. 2013, pp. 2261-2267.* Number of citations: 0 (*)

Christer Bäckström, Peter Jonsson and Simon Ståhlberg **Fast Detection of Unsolvable Planning Instances Using Local Consistency** In *proc. 6th International Symposium on Combinatorial Search (SoCS-13), Leavenworth, WA, USA, July 2013, pp. 29-37.* Number of citations: 2

Christer Bäckström, Peter Jonsson, Sebastian Ordyniak and Stefan Szeider. **Parameterized Complexity and Kernel Bounds for Hard Planning Problems** In *proc. 8th International Conference on Algorithms and Complexity (CIAC 2013), Barcelona, Spain, May, 2013, pp. 13-24.* Number of citations: 1 (*)

Christer Bäckström, Anders Jonsson and Peter Jonsson **Macros, Reactive Plans and Compact Representations** In *proc. 20th European Conference on Artificial Intelligence (ECAI-12), Montpellier, France, Aug. 2012, pp. 85-90.* Number of citations: 0 (*)

Christer Bäckström, Anders Jonsson and Peter Jonsson **From Macro Plans to Automata Plans** In *proc. 20th European Conference on Artificial Intelligence (ECAI-12), Montpellier, France, Aug. 2012, pp. 91-96.* Number of citations: 1

Christer Bäckström and Peter Jonsson **Abstracting Abstraction in Search II: Complexity Analysis** In *proc. 5th International Symposium on Combinatorial Search (SoCS-12) Niagara Falls, ON, Canada, July 2012, pp. 10-17.* Number of citations: 1

Christer Bäckström, Yue Chen, Peter Jonsson, Sebastian Ordyniak and Stefan Szeider
The Complexity of Planning Revisited--A Parameterized Analysis
In *proc. 26th AAAI Conference on Artificial Intelligence (AAAI-12) Toronto, ON, Canada, July 2012, 1735-1741.* Number of citations: 9 (*)

Christer Bäckström and Peter Jonsson **Abstracting Abstraction in Search with Applications to Planning** In *proc. 13th International Conference on Principles of Knowledge Representation and Reasoning (KR-12) Rome, Italy, June 2012, 446-456* Number of citations: 3

Christer Bäckström and Peter Jonsson **All PSPACE-complete Planning Problems are Equal but some are more Equal than Others** In *proc. 4th International Symposium on Combinatorial Search (SoCS-11) Castell de Cardona, Barcelona, Spain, July 2011, pp. 10-17. (Best paper award.)* Number of citations: 4 (*)

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2014-45145-115393-21

Name of applicant
Bäckström, Christer

Date of birth
621230-1912

Title of research programme

Appendix N

Budget and research resources

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

Notes on the Budget

The applicant: 586 kSEK/year (80 %) for active research, supervising the student and leading the project.
New PhD student: 391 kSEK/year (80 %) for research; remaining 20 % will be funded by teaching.
Salaries are adjusted for an annual increase of +2.5 %.

Indirect costs: 46 % overhead.
Computer support: Covers the basic costs for computer support.
Occupancy cost: Covers the cost for office space.
Travel expenses: 75 kSEK/year. International collaborations and conference publications require quite a lot of travelling and we also consider such travelling escpecially important for PhD students.

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2014-45145-115393-21
Name of applicant
Bäckström, Christer
Date of birth
621230-1912

Dnr
2014 -
Reg date
2014-04-08 15:24:31

Project title
A Multi-dimensional Approach to Complexity Analysis in Planning

NT

NT-2

Research grants NT April 9, 2014
Project Research Grant
*Övrigt generellt
Linköpings universitet

Appendix S

Signatures

2015	2016	2017	2018	2019
1680	1715	1752	1789	1827

2015-01-01 -- 2019-12-31

Applicant

Date

Head of department at host University

Clarifi cation of signature

Telephone

A signature on the application is required not only from the applicant but also from the authorised representative of the administrating organisation/institution or equivalent (normally the head of the department or establishment where the research is to be conducted). The signature confirms that the department can accommodate the proposed research, position or equipment; that the costing in the application is approved for the department's part, that any proposed experimentation on human or animal subjects has been reported, and that the applicant has reported any secondary occupations and commercial ties (s)he may have, and nothing inconsistent with good research practice has thereby emerged. The applicant must have discussed these conditions with the representative of the administrating organisation/institution or equivalent before the latter approves and signs the application.

NOTE the exceptions!
Please note that a different procedure applies for the following types of grants:

- Infrastructure- Operation Grants
- Infrastructure- Grants for Expensive Equipment
- Infrastructure- Grants for Large Databases
- Framework Grant SIMSAM
- International Recruitment

For these types of grants the vice-chancellor, instead of the head of department, must sign Appendix S. Thereby (s)he confirms the commitments of the administrative organization. The details are specified in the Call for Proposals, published at www.vr.se.

Vetenskapsrådets noteringar
Kod

2014-45145-115393-21

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se