

Cost-Optimal Symbolic Abduction for Improved Security

Stefan Edelkamp and Thomas Wagner
TZI
Universität Bremen

Peter Kissmann
Computer Science Department
Technische Universität Dortmund

Abstract

Abduction is a backward inference process for generating and narrowing hypotheses. It has been applied for plan recognition and diagnosis and is crucial for detecting attack plans security applications. Cost-optimal abduction drives the selection of hypotheses towards the ones with minimal costs. Abduction is inherently complex. In this paper, we describe solutions for overcoming the computational burden by exploiting a symbolic representation of state sets that has successfully been exploited in AI planning systems. Given a model specified in PDDL-like syntax, we infer a discrete encoding of the domain and study symbolic algorithms to compute cost-optimal hypotheses and according explanations.

Introduction

The ability to recognize plans is a fundamental tasks for a wide range of applications ranging from coordinated actions in dynamic multi-agent systems to infer attack plans in network security applications.

The problem has been addressed with different techniques. Probabilistic approaches like dynamic Bayesian networks (Albrecht *et al.* 1997) have been applied in multi-user scenarios, while relational Markov models (RMM) (Anderson & Weld 2002) have been applied for the recognition of user actions in adaptive web interfaces. Although these approaches have been successful even in the face of noisy observations, their use is limited to specific areas of application. While Bayesian models support the representation of complex structural dependencies they lack the ability to represent relation properties in terms of a RMM and vice versa. Otherwise, more expressive probabilistic representations naturally result in a significant decrease in efficiency. In contrast, classification-based approaches (Intille & Bobick 1999) allow more expressive/flexible representations, but are limited to domains with precisely given sets of possible plans/intentions like, e.g., in American football.

One expressive and flexible approach that has been applied to plan recognition is abduction. According to Peirce (1955) abduction is the process of finding the cause for a set of assumptions and a theory provided. Its philosophical roots go back to Aristotle, while for AI (Pople 1973)

abduction is commonly viewed as a form of reasoning, allowing one to find explanations¹ for certain symptoms (Ng & Mooney 1990).

Our core interest is the abductive inference of intended plans. Different to the plan synthesis problem, in such plan recognition problems (Kautz, Pelavin, & Tenenbergs 1991), the recognizer is given a fragmented description of the problem and expected to refine it. We concentrate on plan hypotheses generation wrt. a fixed domain theory, a set of observations and a set of assumptions. To deal with the inherent complexity, we apply symbolic abduction, where *symbolic* refers to the use of efficient data structures for representing and operating on Boolean functions.

Our research is motivated by a daunting application domain, where we consider inferences on top of an intrusion detection system for an improved level of security. Abduction can be used to reason about the attackers' plans subject to the incidents established. Short attack plans correspond to more plausible ones, and uncertainty centers around the attacker's initial states.

The paper is structured as follows. Firstly, we reflect previous work on symbolic AI such as validating knowledge databases, model-based diagnosis, and BDD-based planning. The core of the exposition is devoted to modeling and designing algorithms for the abductive inference in planning problems. We address the issues of computing all valid hypotheses, the uni- and bi-directional inferences of uniform-cost abductions, a setting that is then extended to cover cost-based abductive inferences. User supervision to manually drive the selection of plan hypotheses is discussed next. As the set of abductive inference problems usually addressed in literature is small, in the experiments we address modified planning benchmarks.

Symbolic Inference in AI

Binary decision diagrams (BDDs) together with efficient operations on them have been proposed by Bryant (1985). Minato, Ishiura, & Yajima (1990) have illustrated how to store several BDDs in a joint structure. One of the most

¹In literature as well as in this text, the terms *explanation* and *hypothesis* are often used interchangeably. However, we prefer the *explanation* to refer to the plan generated, and the *hypothesis* to refer to the possible extensions for the assumptions made.

widely used BDD libraries (CUDD) is maintained by Fabio Somenzi. Many aspects to the theory of decision diagrams have been given by Wegener (2000). BDDs are less compact than other structures like d-DNNFs (Darwiche 1999), but provide a unique representation of Boolean functions.

Validating Knowledge Bases

Especially when applied to business settings, checking for anomalies in a given knowledge-base becomes a very important task. The efficiency of labeling clearly depends on the compactness of the generated labels. As these may require exponential size, with the exponent being in the depth of the rule sets, more efficient representations like BDDs are needed (Torasso & Torta 2003; Mues & Vanthienen 2004). Such symbolic approaches encode the system's input in binary form and traverse the rule base, thereby constructing the BDDs instead of labels that describe the in- and output dependencies of the system, checking BDDs against each other and reporting any observed anomaly. Alternative compilations of knowledge bases are possible (Darwiche 1999).

Action Planning

Symbolic planning has been pioneered by Cimatti *et al.* (1997). It extends to adversarial planning (Cimatti, Roveri, & Traverso 1998), partial observable planning (Bertoli *et al.* 2001), and conformant planning (Bertoli, Cimatti, & Roveri 2001). Symbolic heuristic search invented by Edelkamp & Reffel (1998) has been applied to domain-independent planning (Edelkamp & Helmert 2001). Jensen, Bryant, & Veloso (2002) have introduced branching partitions, while Jensen *et al.* (2006) have proposed memory-limited symbolic branch-and-bound search. Symbolic heuristic search planning with penalties and rewards formulated in logic using DNNFs has been considered by Bonet & Geffner (2008).

In the propositional STRIPS formalism (Fikes & Nilsson 1971) for describing planning domains the initial state is total, while the goal state is partial. For abduction, however, both states are partial. The partially given initial state denotes the assumptions, one possible completion is a hypothesis. The partially given goal state denotes the observations; a completion is a prediction.

Symbolic Diagnosis

In diagnosis, we are not only concerned with detecting errors, but additionally with explaining them. This is done by propagating the error in the model and probing on more and more specific issues. Since a diagnosis task is a search in a space of different hypotheses on the values of variables, it deals with uncertainties in background knowledge.

By explicitly modeling possible system flaws, the single-fault diagnosis problem can be transformed into an inference problem, for which the uncertainty is contained in the limited set of assumptions. For multiple faults, assumption-based truth maintenance systems (ATMSs) have been suggested (Forbus & de Kleer 1993). Their model is an undirected network with the edges labeled with discrete variables, whose values are of a certain range. Devices in the network to be diagnosed manipulate and propagate the information found at incident edges. Assignments to edges

represent the knowledge about and the influence the variables have on each other.

BDDs have been used for covering the amount of uncertainty through compactly representing all possible worlds (Bertoli, Bozzano, & Cimatti 2006). Assignments to variables are often restricted to small ranges. Probing an edge is an assignment to a variable.

Abduction

Abduction has taken on fundamental importance in AI (Morgan 1971) including planning (Bäckström & Nebel 1988), database updates (Kakas & Mancarella 1990), text understanding (Norvig 1987), and others. In abduction, for a logical theory T and some manifestation M of a set of individual hypotheses, we are interested in Δ such that $T \cup \Delta \models M$.

For logic-based abduction, Eiter & Gottlob (1995) classified many problems to be located on the second or third level of the polynomial hierarchy, while Bylander *et al.* (1991) proved the NP-hardness for pure propositional problems to explain a set of data best. There is a balance between the flexibility in the modeling language and the inferences that can be drawn. The problem of generating abductive explanations is usually divided into two subproblems that can be addressed separately:

1. generating the set of all possible explanations, and
2. selecting the most appropriate hypothesis among the set of possible explanations

Efficient approaches to abduction are limited. A tractable solution to the generation problem has been proposed by Eiter & Makino (1992) that is limited to Horn theories and positive observation literals.

The second problem has been addressed in very different ways. The most widely used selection criteria is Occam's razor (Thorburn 1918). It states that for two explanations, the simpler one is preferable. A different domain-independent criterion has been applied by Ng & Mooney (1990) in the domain of text understanding, *the coherence metric*. In addition different *domain-specific* solutions have been proposed (Appelt & Pollack 1992; Hobbs *et al.* 1990).

An inherent weakness of the logic-based approach to abduction² is the very specific interpretation of the logical implications. Abductive inference assumes that logical implication encodes causal knowledge (relations). Although this property may hold in some scenarios³ it is clearly not valid in general. As a consequence, abductive inference often leads to non-causal explanations. We claim to overcome this problem. Instead of Horn or first-order logic programs as the basis for backward inference, we chose PDDL as the modeling language, which is automatically converted to a plan model with discrete state variables. Thereby, we obtain access to a wealth of planning benchmarks to be exploited for abductive reasoning. In contrast to many other approaches to logical abduction we do not address the problem of hypothesis generation and hypothesis selection inde-

²Abductive inference is not necessarily limited to logical representations. For an overview see (Paul 1993).

³E.g., in diagnostic domains.

pendently. Instead of calculating the set of all possible explanations we directly calculate the *best* hypothesis. In this paper we apply Occam's razor in form of cost-optimal explanation as the fundamental selection criterion. Nevertheless, we claim that our approach extends to domain specific criteria in terms of *weighted* abduction (Appelt & Pollack 1992; Hobbs *et al.* 1990).

Symbolic Abduction

Driven by the success of recent BDD-based planning systems on international planning competitions the large BDD compression ratios for many planning benchmarks obtained by Ball & Holte (2008), we aim at solving the abduction problem with BDDs by embedding it into the planning domain definition language PDDL.

PDDL problems can be grounded by instantiating predicates, actions and fluents with all possible instantiations of domain objects, yielding a (usually fully instantiated) initial state $I \subseteq AP$, a set of operators $O = (P, A, D)$ with $P, A, D \subseteq AP$, and a goal description $G \subseteq AO$. Despite its binary representation for symbolic search, it is best to consider all states and operators as sets. For example, set intersection matches Boolean conjunction, set complementation matches Boolean negation, and set unification matches Boolean disjunction.

For the sake of simplicity, we assume a propositional domain theory T over a set of atomic proposition AP to be encoded in STRIPS planning operators O with $o = (P, A, D)$ and $P, A, D \subseteq AP$. In BDD terminology, we construct a transition relation T_o , encoding all (predecessor, successor) state pairs valid under operator $o \in O$. This yields the domain theory $T = \bigvee_{o \in O} T_o$. Logical subsumption $\phi \models_T \psi$ inherits the semantics that there is a sequence of operators applied to $\phi \subseteq AP$, which entails $\psi \subseteq AP$.

For abductive inferences we may assume an SAS⁺ encoding of a propositional planning problem (Bäckström & Nebel 1995) that is induced by static analyzers. The SAS⁺ formalism uses multi-valued state variables instead of propositional atoms. An SAS⁺ structure $M = (V, S, O)$ is over a set of state variables $V = (v_1, \dots, v_m)$ defines a space $S = S_1 \times \dots \times S_m$ of all possible states, where S_j is the domain $Domain(v_j)$ of mutually exclusive values for the j th variable, $j = 1, \dots, m$. Operators change assignments to states according to their pre- and postconditions. Preconditions are Boolean formulas over variable assignments and postconditions are updates of variables to new values. Partial states are states with some variables being undefined.

Starting from PDDL, for the process of partition propositions into mutually exclusive fact groups the initial state does not have to be total (Helmert 2008). The information on mutual exclusion that is encoded in the finite domain variable description belongs to the set of consistency conditions provided to the abductive inference module. In propositional terms, we assume the manifestation to be separated in the set of observations $G \subseteq AP$ in form of a partial description of the goal state, and the set of assumptions $A \subseteq AP$ in form of a partial description of the initial state. We are interested in some hypothesis $\Delta \subseteq AP$, such that $A \cup \Delta \models_T G$.

Any aductive inference process partitions in two stages: (1) generating all, a subset of them, or only one hypothesis, and (2) selecting the hypothesis that is best, which can either be automated wrt. some optimality criterion, or interactive by modifying the assumptions or the observations, in which case the abductive inference iterates.

Computing all Valid Hypotheses

The ultimate goal is to compute all valid hypotheses. Eiter & Makino (1992) give an algorithm that generates all non-trivial explanations of a Horn CNF wrt. some positive letter. For each generated hypothesis the algorithm is polynomial, but the number of hypotheses can be exponential.

To compute all possible hypotheses $\Delta \subseteq AP$, we first define the preimage of a state set *States* (on variable set x') as

$$PreImg(States) = \bigvee_{o \in O} (\exists x'. T_o(x, x') \wedge States(x')) [x \leftrightarrow x'].$$

The suffix $[x \leftrightarrow x']$ denotes that the state variable set is swapped after the operation. If $States_i$ denotes the representation of the set of states in some backward breadth-first search (BFS) level i (minimal goal distance i), then $PreImg(States_i)$ denotes the set of states in backward BFS level $i + 1$. Moreover, the set of all states that are reachable via pre-image is defined as

$$BackReach(States) = \mu X. PreImg(X) \vee States(x'),$$

where μ denotes the fixpoint operator induced by repeated pre-image application. When initializing *States* with goal condition G , with the above equation we compute all possible states that reach G , i.e., *BackReach* is partitioned in BFS levels $BackReach_i$, with $BackReach_0 = G$ and $BackReach_{i+1}$ being computed from $BackReach_i$. To guarantee termination of the exploration, it is recommended to subtract $BackReach_j$ from $BackReach_{i+1}$ for $0 \leq j \leq i$.

For a set of assumptions encoded as a formula $A(x)$ the set of all valid hypotheses is now computed as

$$BackReach(G(x)) \wedge A(x),$$

i.e., the set of all possible states that can reach the observations and that satisfy the assumptions.

Uniform-Cost Abductive Inference

In the case of uniform-cost abductive inference, we follow the principle of Occam's razor to compute the step-minimal explanation. As an example, we take the case of John being depressed under the condition that his girl-friend Mary has had a heart attack. The step-minimal explanation for him being depressed is that John is a pessimistic person, but this is commonly interpreted as the unlikely explanation, given the evidence of Mary's illness.

An algorithm for backward uniform-cost abduction is shown in Figure 1. It repeatedly applies preimages until the assumptions A are hit, inducing a plan to be generated⁴.

⁴For the sake of simplicity, the test of termination for inconsistent assumptions is not shown. It requires a closed list and a full backward exploration of the state space.

Procedure Backward BDD-Abduction

Input: Uniform cost planning problem with theory
 $T = \bigvee_{o \in O} T_o$, set of assumptions $A \subseteq AP$,
and set of observations G
Output: Step-optimal explanation Δ that is
consistent with $A \wedge \Delta \models_T G$

```

BackReach0(x) ← G(x)
for each i = 0, 1, ...
  Min(x) ← BackReachi(x)
  if (Min(x) ∧ A(x) ≠ false)
    return ConstructExplanation(Min(x) ∧ A(x))
  Predi(x') ←  $\bigvee_{o \in O} (\exists x' (Min(x') \wedge T_o(x, x')))[x \leftrightarrow x']$ 
  BackReachi+1(x) ← BackReachi+1(x) ∨ Predi(x)

```

Figure 1: Backward Abduction on Uniform Cost Problems.

Procedure Forward BDD-Abduction

Input: Uniform cost planning problem with theory
 $T = \bigvee_{o \in O} T_o$, set of assumptions $A \subseteq AP$,
and set of observations G
Output: Step-optimal explanation Δ that is
consistent with $A \wedge \Delta \models_T G$

```

ForwReach0(x) ← A(x)
for each i = 0, 1, ...
  Min(x) ← ForwReachi(x)
  if (Min(x) ∧ G(x) ≠ false)
    return ConstructExplanation(Min(x) ∧ G(x))
  Succi(x) ←  $\bigvee_{o \in O} (\exists x (Min(x) \wedge T_o(x, x')))[x \leftrightarrow x']$ 
  ForwReachi+1(x) ← ForwReachi+1(x) ∨ Succi(x)

```

Figure 2: Forward Abduction on Uniform Cost Problems.

In $Min \wedge A$ there might be several valid minimum-step hypotheses. If A' is one, then Δ is the completion to the set of assumptions already made.

As a feature of the algorithm, a prediction Γ to the set of observations can also be returned, by completing the partial goal G to the complete one G' as found through the construction of the explanation.

The advantage of BDD inference is that it is easy to invert the direction of chaining and compute the image of some state set $States$ as follows

$$Image(States) = \bigvee_{o \in O} (\exists x. T_o(x, x') \wedge States(x))[x' \leftrightarrow x].$$

For abductive inference the same (or an equivalent) minimal-cost plan can be obtained by chaining forward from the set of assumptions to the observations. An according implementation is shown in Figure 2.

The problem here is that the hypothesis Δ is not computed directly from the last set of states that has been reached, as it was inferred in backward search. However, after having extracted the step-minimal explanation, besides the set of

```

(:action r1 :parameters (?X ?Y - person)
:precondition (and (like ?X ?Y) (ill ?Y)
(irreplacable ?Y))
:effect (and (depressed ?X)
(increase(total-cost) 3)))
(:action r2 :parameters (?X - person)
:precondition (and (pessimist ?X))
:effect (and (depressed ?X)
(increase (total-cost) 10)))
(:action r3 :parameters (?X - person ?O - organ)
:precondition (and (has ?X ?O) (illness ?O))
:effect (and (ill ?X)
(increase (total-cost) 1)))
(:action r4 :parameters (?O - organ)
:precondition (and (heart-attack ?O))
:effect (and (illness ?O)
(increase (total-cost) 3)))

```

Figure 3: PDDL Actions for Example Domain.

actions, the set of states in the plan and especially the completion to A is also computed.

Having forward and backward inferences, it is also possible to operate bi-directional, reducing the complexity of finding the smallest explanation drastically. A coarse argumentation is that bi-directional breadth-first search with a goal distance d in a graph with uniform branching factor b (and no duplicate elimination) looks at $2b^{d/2}$ states, a number being exponentially smaller than b^d , the efforts for uni-directional breadth-first search.

Cost-Optimal Abductive Inference

In many cases, the principle of Occam's razor to compute the step-minimal explanation in abductive reasoning is insufficient. In other words, the relevance of operators for the inference process is not uniform. We assign costs $cost(o)$, $o \in O$, denoting how important individual actions are (the higher the cost, the less important the operator).

As an example we once again take the case of John being depressed. Fragments of the PDDL model are shown in Figure 3. If we assign costs to inference operators such as $cost(r_1) = 3$, $cost(r_2) = 10$, $cost(r_3) = 1$, and $cost(r_4) = 3$ we get the cost-minimal explanation that John is depressed because of Mary's heart attack.

A pseudo-code implementation of the algorithm is shown in Figure 4. The core difference to uniform cost abduction is that the preimages of the transition relation are computed for each cost value l in $1, \dots, C$. Zero-cost actions can be included by computing a transitive closure wrt. all such actions before expanding a bucket.

Similar to the uniform case, forward and backward induction work similar by simulating Dijkstra's algorithm (1959). Bidirectional search now faces the problem that the first intersection of the search frontiers does not necessarily yield the minimum-cost solution. For such case, symbolic perimeter search is applied as follows. In a first phase we construct a perimeter database, storing the backward layers up to some depth. This database then serves as a heuristic for guiding the search in forward direction.

Procedure Backward Cost-based BDD-Abduction

Input: Cost-based planning problem with theory
 $T = \bigvee_{o \in O} T_o$, set of assumptions $A \subseteq AP$,
and set of observations G
Output: Cost-optimal explanation Δ that is
consistent with $A \wedge \Delta \models_T G$

```

BackReach0(x) ← G(x)
for each i = 0, 1, ...
  Min(x) ← BackReachi(x)
  if (Min(x) ∧ A(x) ≠ false)
    return ConstructExplanation(Min(x) ∧ A(x))
  for all l = 1, ..., C
    Predl(x') ←
       $\bigvee_{o \in O, cost(o)=l} (\exists x' (Min(x') \wedge T_o(x, x')) [x \leftrightarrow x'])$ 
    BackReachi+l(x) ← BackReachi+l(x) ∨ Predl(x)

```

Figure 4: Algorithm for Cost-based Backward Abduction.

Manual Selection Strategies

Having computed at least one valid hypothesis, it is conceptually easy to generate the corresponding plan. The simplest solution is to chain the sequence $BackReach_i$ down to $BackReach_0$ backwards starting with a state in the intersection and computing forward images of one selected state in $BackReach_j$ that is intersected with the next state set $BackReach_{j+1}$.

This explanation can be returned to the user who refines the result by either strengthening or weakening the assumption A . Alternatively, he can simply reject the plan, giving rise to a Taboo list D that is subtracted from the goal, i.e. setting G to $G \setminus D$. Last but not least, we may allow to eliminate the impact of certain operators from the plan by rescaling their influence.

Finding Critical Query Variables

For an interactive fault diagnosis with a small number of queries, it is important to reduce the uncertainty in the domain of the variables. One promising aspect is to query the variable that reduces the set of possible worlds by the largest margin, by means that the set of satisfying assignments to the variables is minimal.

For the problem of finding the critical variables in the valid hypothesis we exploit the fact that model counting (the process of determining the number of satisfying assignments to a Boolean formula) in a BDD is efficient. Hence for each SAS⁺ variable v and each possible assignment i in the domain $Domain(v)$ of v we determine

$$\sum_{i \in Domain(v)} ModelCount(ValidHypothesis \wedge (v = i))$$

and take the variable for which this quantity is the smallest.

Experiments

For executing abductive reasoning, we adapted the planner GAMER that won the sequential optimal and optimal

Procedure Bidirectional Cost-based BDD-Abduction

Input: Cost-based planning problem with theory
 $T = \bigvee_{o \in O} T_o$, set of assumptions $A \subseteq AP$,
and set of observations G , backward layers $BackReach_i$
Output: Cost-optimal explanation Δ that is
consistent with $A \wedge \Delta \models_T G$

```

for all h ∈ {0, ..., maxh}
  Open[0, h](x) ← A(x) ∧ BackReachh(x)
for all f ∈ {0, 1, 2, ..., f}, g ∈ {0, ..., f}
  Min(x) ← Open[g, f - g](x)
  if (Min(x) ∧ G(x) ≠ false)
    return ConstructExplanation(Min(x) ∧ G(x))
  for all i ∈ {1, ..., C}
    Succi(x') ←
       $\bigvee_{o \in O, cost(o)=i} \exists x. Min(x) \wedge Trans_o(x, x') [x \leftrightarrow x']$ 
  for all h ∈ {0, ..., maxh}
    Open[g + i, h](x) ← Open[g + i, h](x) ∨
      Succi(x) ∧ BackReachh(x)

```

Figure 5: Bidirectional Abduction with Perimeter Database.

net-benefit track in the international planning competition IPC-2008. We ran the experiments on a machine with two Opteron 250 processors with 2.4 GHz (only one is used for computation) and limited the memory to 2 GB RAM.

There is initial work by Boddy *et al.* (2005) and Bhattacharya & Ghosh (2008) in modeling security problems as planning problems, we do not have sufficiently many security benchmarks to evaluate our approach. Hence, we adopted planning competition benchmarks for abductive reasoning tasks.

Grounding the planning domains yields an SAS⁺ encoding of the problems. The original results of fully specified benchmark problems under closed world assumption (cwa), meaning that the parts not mentioned in the initial state are false, are compared to an open world (no cwa), meaning that the parts not mentioned in the initial state are unknown. Then we omit every second fact in the initial state (half init), i.e., we eliminate parts of the initial state (in an open world) in order to reconstruct it.

The abduction algorithm we chose was bidirectional cost-based BDD abstraction (Figure 5). The total time bound was set to 5 minutes, from which we took at most 150s for backward perimeter construction and the remaining time for forward search. In some domains (*ParcPrinter*, *PegSolitaire*, and *Sokoban*) dropping the closed world assumption leads to empty plans (the intersection of the initial state with the goal states is not empty), such that we dropped these examples from the presentation.

Table 1 shows the results in the *Elevator* domain. Here, we see that dropping the closed world assumption has no effect. This is an immediate consequence of the fact that the initial state is fully specified with positive literals. For a half-way specified initial state we see some advances. In some cases new problems could be solved, in others the abductive inference is harder (Problems 24 and 25) than the original

Table 1: Results in Elevator.

	cwa					no cwa					half init				
	cost	steps	back.	forw.	total	cost	steps	backw.	forw.	total	cost	steps	backw.	forw.	total
1	42	14	2:31	0:06	2:38	42	14	2:30	0:08	2:40	15	6	2:29	0:07	2:38
2	26	9	2:30	0:05	2:37	26	9	2:29	0:05	2:37	19	6	2:29	0:05	2:37
3	55	18	2:30	0:06	2:38	55	18	2:30	0:06	2:37	27	10	2:29	0:05	2:37
4	40	18	2:29	0:09	2:40	40	18	2:29	0:09	2:41	20	9	2:29	0:08	2:40
5	55	22	2:30	0:09	2:40	55	22	2:29	0:09	2:40	43	16	2:29	0:13	2:45
6	53	26	2:29	0:57	3:28	53	26	2:29	0:57	3:29	25	14	2:29	0:07	2:38
7	62	27	2:29	0:42	3:14	62	27	2:29	0:37	3:08	27	12	2:30	0:04	2:36
8	53	25	2:28	2:14	4:46	53	25	2:29	2:16	4:48	30	12	2:29	0:33	3:05
11	56	17	2:29	0:05	2:36	56	17	2:30	0:05	2:36	34	10	2:30	0:05	2:36
12	54	16	2:29	0:06	2:38	54	16	2:30	0:06	2:38	37	11	2:29	0:09	2:41
13	59	17	2:30	0:07	2:38	59	17	2:30	0:07	2:38	40	10	2:30	0:06	2:37
14	63	22	2:29	0:21	2:53	63	22	2:30	0:20	2:52	39	11	2:29	0:12	2:44
15	66	24	2:30	0:12	2:44	66	24	2:30	0:13	2:44	40	14	2:30	0:08	2:40
16	—	—	—	—	—	—	—	—	—	—	43	15	2:28	2:16	4:48
17	—	—	—	—	—	—	—	—	—	—	42	13	2:28	0:14	2:47
21	48	17	2:30	0:07	2:39	48	17	2:30	0:07	2:39	31	10	2:30	0:06	2:38
22	54	19	2:29	0:37	3:09	54	19	2:29	0:38	3:10	36	12	2:30	0:46	3:17
23	—	—	—	—	—	—	—	—	—	—	39	15	2:29	0:18	2:49
24	56	24	2:29	2:10	4:42	56	24	2:29	2:12	4:44	—	—	—	—	—
25	63	27	2:29	0:57	3:29	63	27	2:29	1:00	3:32	43	20	2:30	2:28	5:00
26	—	—	—	—	—	—	—	—	—	—	29	12	2:29	1:33	4:06

plan finding process.

Table 2 depicts the results in the *Openstacks* domain. Here we see that abduction gets harder. This is due to the fact that cost-based backward induction to construct the perimeter database does not provide any information, as cost zero results in no information. The forward abduction phase thus degrades to uniform cost search.

Table 3 shows the results in the *Transport* domain. Here we have considerably larger costs than in the previous ones. Table 4 gives the results in the *Woodworking* domain, showing a similar trend as the results in the *Transport* domain.

Conclusion and Discussion

Abductive reasoning selects hypotheses that explain the evidences best. It is of high relevance for AI, but – due to its large complexity demands – received limit attention in the last decade. As computational power on modern CPUs and planner technology have improved substantially, we showed a promising technique to compute valid hypotheses time- and space-efficiently. Our main focus is the hypothesis generation problem in the context of SAS⁺ planning. Moreover, we discussed different options for interaction to overcome the limitations of Occam’s razor, including interaction based modified operator costs, altering the set of assumptions, maintaining Taboo lists for goals and applying model-counting. They all help to reduce the amount of uncertainty.

One question that may arise is how much symbolic abductive planning differs from ordinary symbolic planning as, besides the initial state being partial, there is not much change in the exploration algorithms. This is true, but it also highlights the advantage symbolic search approaches have with respect to explicit-state ones, which would have to enumerate all possible completions either for the initial state in

order to perform the inference process. Moreover, most existing explicit-state planning systems are not cost-optimal.

One difference between abduction and ATMS inference is that the latter logs justifications to assignments to allow multiple fault analyses. From a logical perspective, abduction chains backwards in time, from the set of observation towards the set of assumptions, while the inference in ATMS is multi-directional, depending on the update to the set of assignments to an incident variable of a device, while propagating the effects through the network.

Note that there is precursing work on cost-based abduction using binary decision diagrams (Kato *et al.* 1999). In the logical context of propositional Horn clauses, the authors compile a BDD for the theory, such that each BDD variable corresponds to a different hypothesis. BDD edges are associated with costs and additional consistency criteria lead to pruning of edges in the BDD. Satisfying paths in the BDDs correspond to the set of possible hypotheses (0-Edges are neglected). As the BDD for the entire theory may be too large, alternative data structures are proposed that handle inference rules lazily. (Cost-annotated) Horn inferences are much more restricted than (cost-annotated) SAS⁺ planning inferences that are considered here. While the former can be solved by variable substitution in the BDD, the latter requires exploration with BDDs.

References

- Albrecht, D. W.; Zukerman, I.; Nicholson, A. E.; and Bud, A. 1997. Towards a Bayesian model for keyhole plan recognition in large domains. In *International Conference on User Modelling*. 365–376.
- Anderson, C., and Weld, D. 2002. Relational markov models and their application to adaptive web navigation. In *In-*

Table 2: Results in Openstacks (ADL).

	cwa					no cwa					half init				
	cost	steps	back.	forw.	total	cost	steps	backw.	forw.	total	cost	steps	backw.	forw.	total
1	2	17	0:01	0:01	0:02	1	11	0:01	0:01	0:02	0	6	0:00	0:00	0:02
2	2	20	0:01	0:01	0:02	1	13	0:01	0:01	0:02	0	7	0:00	0:00	0:02
3	2	23	0:01	0:01	0:03	1	15	0:01	0:01	0:02	0	8	0:00	0:00	0:02
4	3	27	0:01	0:01	0:04	1	17	0:01	0:01	0:04	0	9	0:01	0:00	0:03
5	4	31	0:02	0:01	0:04	1	19	0:02	0:01	0:04	0	10	0:01	0:01	0:04
6	2	32	0:04	0:01	0:08	1	21	0:05	0:01	0:07	0	11	0:04	0:01	0:07
7	5	38	0:13	0:02	0:16	1	23	0:13	0:01	0:16	0	12	0:12	0:02	0:15
8	5	41	0:33	0:02	0:37	1	25	0:35	0:02	0:38	0	13	0:34	0:02	0:38
9	3	42	0:47	0:02	0:51	1	27	0:51	0:02	0:54	0	14	0:51	0:04	0:56
10	3	45	2:30	0:03	2:34	1	29	2:30	0:03	2:34	0	15	2:29	0:19	2:51
11	4	49	2:30	0:05	2:36	1	31	2:30	0:03	2:35	0	16	2:29	0:48	3:20
12	3	51	2:30	0:08	2:40	1	33	2:30	0:04	2:36	0	17	2:29	0:54	3:26
13	4	55	2:29	0:26	2:59	1	35	2:29	0:07	2:40	—	—	—	—	—
14	4	58	2:29	0:11	2:44	1	37	2:29	0:12	2:44	—	—	—	—	—
15	—	—	—	—	—	1	39	2:29	0:13	2:45	—	—	—	—	—
16	4	64	2:29	1:20	3:52	1	41	2:29	1:12	3:44	—	—	—	—	—
17	—	—	—	—	—	1	43	2:29	2:00	4:33	—	—	—	—	—
18	3	69	2:29	0:43	3:16	—	—	—	—	—	—	—	—	—	—
19	4	73	2:29	2:21	4:54	—	—	—	—	—	—	—	—	—	—
22	4	82	2:28	0:29	3:02	—	—	—	—	—	—	—	—	—	—

Table 3: Results in Transport.

	cwa					no cwa					half init				
	cost	steps	back.	forw.	total	cost	steps	backw.	forw.	total	cost	steps	backw.	forw.	total
1	54	5	0:01	0:01	0:04	54	5	0:01	0:01	0:02	52	3	0:01	0:01	0:04
2	131	12	0:10	0:02	0:12	131	12	0:10	0:02	0:12	98	8	0:11	0:02	0:13
3	250	17	2:30	0:08	2:40	250	17	2:30	0:08	2:40	88	7	2:30	0:07	2:40
4	318	22	2:29	1:08	3:42	318	22	2:29	1:12	3:45	170	13	2:29	0:47	3:21
11	456	9	0:01	0:01	0:03	456	9	0:01	0:01	0:03	243	5	0:01	0:01	0:03
12	594	16	0:54	0:05	0:57	594	16	0:54	0:05	0:57	264	8	0:59	0:05	1:02
13	550	21	2:29	0:25	2:58	550	21	2:29	0:25	2:58	294	11	2:30	0:38	3:11
21	478	7	0:00	0:00	0:02	478	7	0:00	0:00	0:02	160	3	0:00	0:00	0:02
22	632	12	0:13	0:03	0:16	632	12	0:15	0:03	0:17	431	8	0:15	0:03	0:17
23	630	17	2:30	0:09	2:42	630	17	2:30	0:10	2:42	265	7	2:30	0:09	2:41
24	614	19	2:28	0:53	3:26	614	19	2:28	0:57	3:30	231	9	2:29	0:13	2:47

ternational Conference on Knowledge Discovery and Data Mining, 143–152.

Appelt, D. E., and Pollack, M. E. 1992. Weighted abduction for plan ascription. In *User Modeling and User-Adapted Interaction*, 2(1-2), 1–25.

Bäckström, C., and Nebel, B. 1988. Abductive planning with event calculus. In *International Conference and Symposium on Logic Programming*, 562–579.

Bäckström, C., and Nebel, B. 1995. Complexity results for SAS⁺ planning. *Computational Intelligence* 11(4):625–655.

Ball, M., and Holte, R. C. 2008. The compression power of symbolic pattern databases. In *ICAPS*, 2–11.

Bertoli, P.; Cimatti, A.; Roveri, M.; and Traverso, P. 2001. Planning in nondeterministic domains under partial observability via symbolic model checking. In *IJCAI*, 473–478.

Bertoli, P.; Bozzano, M.; and Cimatti, A. 2006. A symbolic

model checking framework for safety analysis, diagnosis, and synthesis. In *MoChArt*, 1–18.

Bertoli, P.; Cimatti, A.; and Roveri, M. 2001. Heuristic search symbolic model checking = efficient conformant planning. In *IJCAI*, 467–472.

Bhattacharya, S., and Ghosh, S. K. 2008. Attack trees: Modeling security threats. *Journal of Information Assurance and Security* 3(2):119–127.

Boddy, M. S.; Gohde, J.; Haigh, T.; and Harp, S. A. 2005. Course of action generation for cyber security using classical planning. In *ICAPS*, 12–21.

Bonet, B., and Geffner, H. 2008. Heuristics for planning with penalties and rewards formulated in logic and computed through circuits. *Artificial Intelligence* 172:1579–1604.

Bryant, R. E. 1985. Symbolic manipulation of boolean functions using a graphical representation. In *DAC*, 688–694.

Table 4: Results in Woodworking.

	cwa					no cwa					half init				
	cost	steps	back.	forw.	total	cost	steps	backw.	forw.	total	cost	steps	backw.	forw.	total
1	170	9	0:01	0:01	0:04	170	9	0:01	0:01	0:04	75	5	0:02	0:01	0:04
2	185	9	0:04	0:02	0:07	185	9	0:04	0:02	0:08	135	8	0:05	0:02	0:08
3	275	18	1:07	0:07	1:17	275	18	1:02	0:08	1:12	80	8	1:11	0:04	1:17
4	—	—	—	—	—	—	—	—	—	—	100	8	2:29	0:06	2:38
11	130	7	0:01	0:01	0:04	130	7	0:01	0:01	0:04	45	4	0:01	0:01	0:04
12	225	12	0:13	0:04	0:19	225	12	0:11	0:04	0:18	130	7	0:14	0:03	0:19
13	215	13	2:05	0:06	2:14	215	13	1:57	0:06	2:07	110	7	2:05	0:05	2:13
14	225	16	2:30	1:24	3:56	—	—	—	—	—	145	8	2:30	0:05	2:38
16	—	—	—	—	—	—	—	—	—	—	120	8	2:29	0:14	2:47
21	95	6	0:01	0:01	0:04	95	6	0:01	0:01	0:04	35	2	0:01	0:01	0:03
22	185	9	0:04	0:02	0:07	185	9	0:04	0:02	0:07	105	5	0:04	0:01	0:07
23	195	13	0:23	0:03	0:28	195	13	0:23	0:03	0:28	95	7	0:26	0:02	0:30
24	245	15	2:29	0:14	2:47	245	15	2:29	0:19	2:51	145	7	2:29	0:05	2:38

Bylander, T.; Allemang, D.; Tanner, M. C.; and Josephson, J. R. 1991. The computational complexity of abduction. *Artif. Intell.* 49(1-3):25–60.

Cimatti, A.; Giunchiglia, E.; Giunchiglia, F.; and Traverso, P. 1997. Planning via model checking: A decision procedure for AR. In *ECP*, 130–142.

Cimatti, A.; Roveri, M.; and Traverso, P. 1998. Automatic OBDD-based generation of universal plans in non-deterministic domains. In *AAAI*, 875–881.

Darwiche, A. 1999. Compiling knowledge into decomposable negation normal form. In *IJCAI*, 284–289.

Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* 1:269–271.

Edelkamp, S., and Helmert, M. 2001. MIPS: the model-checking integrated planning system. *AI Magazine* 22(3):67–72.

Edelkamp, S., and Reffel, F. 1998. OBDDs in heuristic search. In *KI*, 81–92.

Eiter, T., and Gottlob, G. 1995. The complexity of logic-based abduction. *Journal of the ACM* 42:3–42.

Eiter, T., and Makino, K. 1992. On computing all abductive explanations. In *AAAI*, 62–67.

Fikes, R. E., and Nilsson, N. J. 1971. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208.

Forbus, K. D., and de Kleer, J. 1993. *Building Problem Solvers*. MIT Press.

Helmert, M. 2008. *Understanding Planning Tasks: Domain Complexity and Heuristic Decomposition*, volume 4929 of *LNCS*. Springer.

Hobbs, J. R.; Stickel, M.; Appelt, D.; and Martin, P. 1990. Interpretation as abduction. *Artificial Intelligence* 63:69–142.

Intille, S. S., and Bobick, A. F. 1999. A framework for recognizing multi-agent action from visual evidence. In *AAAI*, 518–525.

Jensen, R. M.; Hansen, E. A.; Richards, S.; and Zhou,

R. 2006. Memory-efficient symbolic heuristic search. In *ICAPS*, 304–313.

Jensen, R. M.; Bryant, R. E.; and Veloso, M. M. 2002. SetA*: An efficient BDD-based heuristic search algorithm. In *AAAI*, 668–673.

Kakas, A., and Mancarella, P. 1990. Database updates through abduction. In *Very Large Data Bases*, 650–661.

Kato, S.; Oono, S.; Seki, H.; and Itoh, H. 1999. Cost-based abduction using binary decision diagrams. In *IAE/AIE*, 215–225.

Kautz, H. A.; Pelavin, R. N.; and Tenenber, J. D. 1991. A formal theory of plan recognition and its implementation. In *Reasoning about Plans*, 69–125. Morgan Kaufmann.

Minato, S.; Ishiura, N.; and Yajima, S. 1990. Shared binary decision diagram with attributed edges for efficient boolean function manipulation. In *DAC*, 52–57.

Morgan, C. G. 1971. Hypothesis generation by machine. *Artif. Intell.* 2:179–187.

Mues, C., and Vanthienen, J. 2004. Improving the scalability of rule base verification. In *KI*, 381–395.

Ng, H. T., and Mooney, R. J. 1990. On the role of coherence in abductive explanation. In *AAAI*, 337–342.

Norvig, P. 1987. *A Unified Theory of Inference for Text Understanding*. Ph.D. Dissertation, EECS Department, University of California, Berkeley.

Paul, G. 1993. Approaches to abductive reasoning - an overview. *Artificial Intelligence Review* 7:109–152.

Peirce, C. S. 1955. Abduction and induction. In Buchler, J., ed., *Philosophical Writings of Peirce*. Dover Books. 150–156.

Pople, H. E. 1973. On the mechanism of abductive logic. In *IJCAI*, 147–152.

Thorburn, W. M. 1918. The myth of Occam’s razor. *Mind* 27(107):345–353.

Torasso, P., and Torta, G. 2003. Computing minimal-cardinality diagnoses using bdds. In *KI*, 224–238.

Wegener, I. 2000. *Branching programs and binary decision diagrams - theory and applications*. SIAM.