A Generic Approach to Planning in the Presence of Incomplete Information: Theory and Implementation (Extended Abstract)[*]

Son Thanh To
Knexus Research Corporation
National Harbor, MD 20745
son.to@knexusresearch.com

Tran Cao Son and **Enrico Pontelli**
Department of Computer Science
New Mexico State University
{tson | epontell}@cs.nmsu.edu

Abstract

This paper proposes a generic approach to planning in the presence of incomplete information. The approach builds on an abstract notion of a belief state representation, along with an associated set of basic operations. These operations facilitate the development of a sound and complete transition function, for reasoning about effects of actions in the presence of incomplete information, and a set of abstract algorithms for planning. The paper demonstrates how the abstract definitions and algorithms can be instantiated in three concrete representations: minimal-DNF, minimal-CNF, and prime implicates, resulting in three highly competitive conformant planners: DNF, CNF, and PIP. The paper includes an experimental evaluation of the planners DNF, CNF, and PIP and proposes a new set of conformant planning benchmarks that are challenging for state-of-the-art conformant planners.

Planning in the presence of incomplete information [Goldman and Boddy 1996; Smith and Weld, 1998] is the problem of generating a plan that can achieve a given goal regardless of what the actual truth value of unknown information about the initial world is. This paper focuses on *conformant planning*—a special class of planning problems in the presence of incomplete information—that deals with incomplete information about the initial state and with deterministic and non-sensing actions.

One of the challenges in conformant planning is the problem of reasoning about action effects in the presence of incomplete information. Specifically, consider an action a with a set of conditional effects of the form $(a : \psi \rightarrow \eta)$, where ψ and η are a set of literals. Each effect $(a : \psi \rightarrow \eta)$ denotes that the set of literals η will be true in the resulting state after the execution of a in the current state if the *e-condition* ψ is true in the current state. Due to incomplete information, the current state and, hence, the truth value of ψ in it can be unknown. As a consequence, the fact that η must be true in the successor state can be unknown, making the computation of the successor state particularly challenging. This agrees with the results in [Baral *et al.*, 2000; Haslum and Jonsson, 1999], that conformant planning for domains with conditional effects is at a higher complexity level than that without conditional effects. Since its introduction, conformant planning has attracted the attention of several researchers—leading to the development of several state-of-the-art conformant planners. It is important to observe that most efficient conformant planners are *best-first search* and *progression-based* planners, whose development starts with the selection of a representation language for incomplete (partially known) states and the definition of a transition (progression) function that, given a state S and an action, computes the next state S' of the world, where S and S' are generally incomplete and encoded in the representation.

To deal with incomplete information about the world, the notion of a *belief state*—defined as the set of possible states—has been introduced and is widely used in planning in presence of incomplete information [Bonet and Geffner, 2000; Smith and Weld, 1998]. An advantage of this notion lies in its simplicity in representing and reasoning about the effects of actions in presence of incomplete information. Indeed, any formalism for *Reasoning about Action and Change (RAC)*, that assumes completeness of the state, can be straightforwardly generalized to deal with incomplete states, by dealing with each individual possible state in the belief state separately, and by collecting the set of resulting states as the successor belief state. More precisely, let s be a state and a be an action executable in s (s satisfies the precondition of a). The *effect* of executing a in s is the set of literals that become true in the successor state, defined as

$$e(a, s) = \{\ell \mid \exists (a : \psi \rightarrow \eta).\, \psi \subseteq s \wedge \ell \in \eta\} \qquad (1)$$

The transition function $\Phi(a, s)$ that returns the result of

[*] This paper is based on the journal article [To, Son, and Pontelli, 2015].

the execution of a in s is defined by

$$\Phi(a, s) = s \setminus \overline{e(a, s)} \cup e(a, s) \qquad (2)$$

where $\overline{e(a, s)}$ denotes the set of negations of literals in $e(a, s)$. Then the function $\widehat{\Phi}(a, S) = \{\Phi(a, s) \mid s \in S\}$ characterizes the transition between belief states and can be used for reasoning with incomplete information.

The downside of the direct use of belief states lies in its size, which is exponential in the number of unknown fluents. This presents two challenges for the planners employing this representation. First, it can quickly increase the memory usage for maintaining the set of generated belief states to avoid repeated search. This can often lead to the 'Out of memory' situation before the planner finds a solution. Second, it directly influences the time complexity in computing the successor belief states. To address this problem, different approaches have been proposed in state-of-the-art planners, resulting in significant improvement in their performance, both planning time and scalability on the problem size.

In the planner KACMBP [Cimatti, Roveri, and Bertoli, 2004], belief states are represented as OBDDs [Bryant, 1992] with the use of model checking techniques for expanding the search space. The OBDD formula was also used in the planner POND [Bryce *et al.*, 2006] to represent literals and actions in the planning graph for computing heuristics. This BDD-based approach does not require any extra BDD manipulation operations and it can be applied in both directions: progression and regression. However, the transition function defined for OBDDs may require a huge memory consumption in the manipulation of the OBDDs. This is a potential reason why planners employing this formulation, like POND or KACMBP, do not scale well as shown in the experimental evaluation [To, Son, and Pontelli, 2015].

Hoffmann and Brafman (2006) proposed implicit encoding of belief states with the sequence of actions that leads to the belief state from the initial state. To determine whether an unknown proposition holds in a belief state, the resulting conformant planner CFF has to reason about a CNF formula that captures the semantic of the initial belief state and the entire action sequence from it to the current belief state. This representation requires very little memory. However, it incurs excessive repeated computations and checking whether two belief states under this representation are equivalent is challenging and expensive. Furthermore, checking whether a proposition holds after the execution of even one single action in the presence of incomplete information is co-NP complete. We believe those are the main reasons why CFF has difficulties in finding a solution for even small instances of harder problems, where there are unknown propositions in the e-conditions [Palacios and Geffner, 2009; Tran *et al.*, 2009; To, Pontelli, and Son, 2009].

The approach employed in the planners `T0` [Palacios and Geffner, 2007; Palacios and Geffner, 2009] and `T1` [Albore, Ramirez, and Geffner, 2011] compiles incomplete information away by translating a conformant problem into a classical problem and then using the efficient classical planner FF [Hoffmann and Nebel, 2001] to solve the resulting problem. This approach demonstrates great performance improvements, e.g., `T0` and `T1` can solve many hard problems of large size. However, the complete translation is exponential in the *conformant width* of the problem, related to the number of relevant unknown literals, and the number of literals in the resulting problem can be exponential in that of the original problem. To reduce the complexity of the translation, `T0` sacrifices the completeness of the translation in certain problems. This explains why `T0` is unable to return a solution in several problems, that have solutions, or the translation fails for large instances of several domains, as shown in our experiments.

GC[LAMA] [Nguyen *et al.*, 2012] is another conformant planner that uses a classical planner. This approach is based on the observation that every solution for a conformant problem is also a solution for the classical problem obtained by replacing the initial belief state with a state in it. A tentative plan α for such a subproblem then can be corrected, by inserting an action sequence to produce precondition for the action the plan α violates at that place. If α cannot be fixed to become a plan for the conformant problem, GC[LAMA] continues with another plan of the same or different subproblem. GC[LAMA] is efficient on problems where the action effects are monotonic, i.e., useful actions create useful literals without destroying other useful literals, and the belief state is not large. Otherwise, it is not as efficient, as reported in the experimental evaluation.

Instead of using a complete transition function in the search, the planner CpA [Son *et al.*, 2005] *approximates* a belief state with the intersection of the states in it [Son and Baral, 2001]. The approximation is polynomial, yet it is incomplete and so are planners employing it. Later, a complete condition for the approximation was identified and corresponding techniques were developed in CpA to make the planner complete [Son and Tu, 2006; Tu *et al.*, 2011]. The computation of successor belief states in this approach is very simple, i.e., it can be performed in the same manner as that for belief states. However, these techniques require the system to deal with *sets* of approximated states, which—in the worst case—are the same as the actual belief states being represented. Hence, the approximated formula explodes in many problems, as observed in the experiments, preventing the planner to effectively scale. The authors of CpA [Tran *et al.*, 2009] developed preprocessing techniques which help reduce the size of the initial disjunctive formula in certain classes of problems, enabling the planner to perform very well in several benchmarks and to win the conformant planning category in the IPC-08.

In this paper, we propose a generic approach to plan-

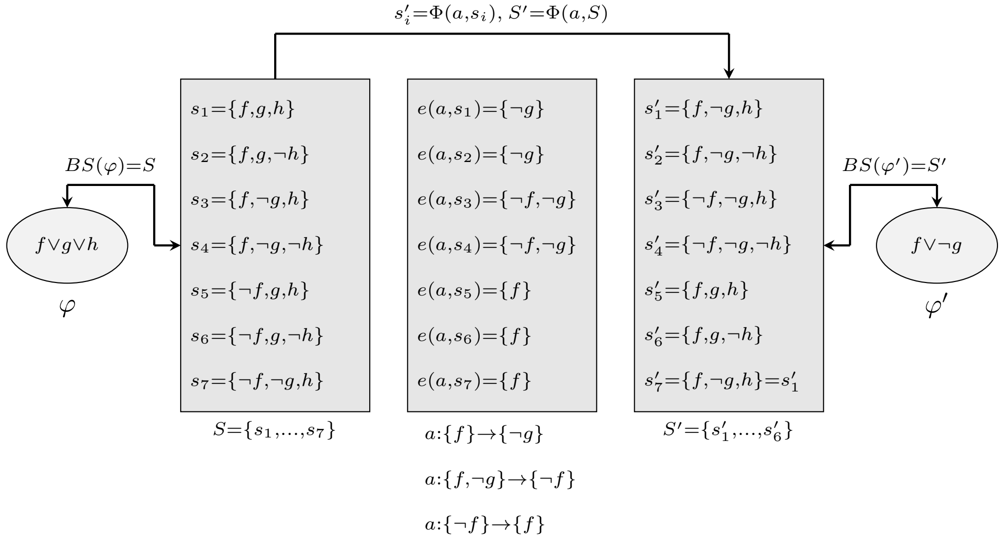
$s'_i{=}\Phi(a,s_i),\ S'{=}\Phi(a,S)$
$BS(\varphi){=}S$
$f{\vee}g{\vee}h$
φ
$s_1{=}\{f,g,h\}$
$s_2{=}\{f,g,\neg h\}$
$s_3{=}\{f,\neg g,h\}$
$s_4{=}\{f,\neg g,\neg h\}$
$s_5{=}\{\neg f,g,h\}$
$s_6{=}\{\neg f,g,\neg h\}$
$s_7{=}\{\neg f,\neg g,h\}$
$S{=}\{s_1,\ldots,s_7\}$
$e(a,s_1){=}\{\neg g\}$
$e(a,s_2){=}\{\neg g\}$
$e(a,s_3){=}\{\neg f,\neg g\}$
$e(a,s_4){=}\{\neg f,\neg g\}$
$e(a,s_5){=}\{f\}$
$e(a,s_6){=}\{f\}$
$e(a,s_7){=}\{f\}$
$a{:}\{f\}{\rightarrow}\{\neg g\}$
$a{:}\{f,\neg g\}{\rightarrow}\{\neg f\}$
$a{:}\{\neg f\}{\rightarrow}\{f\}$
$s'_1{=}\{f,\neg g,h\}$
$s'_2{=}\{f,\neg g,\neg h\}$
$s'_3{=}\{\neg f,\neg g,h\}$
$s'_4{=}\{\neg f,\neg g,\neg h\}$
$s'_5{=}\{f,g,h\}$
$s'_6{=}\{f,g,\neg h\}$
$s'_7{=}\{f,\neg g,h\}{=}s'_1$
$S'{=}\{s'_1,\ldots,s'_6\}$
$BS(\varphi'){=}S'$
$f{\vee}\neg g$
φ'

Figure 1: An example of computing successor belief states by using the function Φ

ning with incomplete information. The key idea underlying our approach is the observation that belief states are *propositional formulae*. Thus, reasoning about effects of actions on belief states can be expressed using suitable operations on formulae, which is usually (possibly exponentially) more compact than the belief states they represent.

As an example, let us consider a domain $F = \{f, g, h\}$, an action a with three conditional effects $C_a = \{f \rightarrow \neg g, f \wedge \neg g \rightarrow \neg f, \neg f \rightarrow f\}, pre(a) = \{\mathit{True}\}$, and a belief state S that contains all possible states of the world (i.e., all possible interpretations of propositions in F) except for $\{\neg f, \neg g, \neg h\}$. Figure 1 shows the belief state $S = \{s_1, \ldots, s_7\}$ and the computation of the successor belief state S' after executing a in S using the function Φ (Equation 2).

Let φ be a formula and S be a belief state. We say that φ *represents* S or S is the *belief state of* φ, denoted by $BS(\varphi)$, if S is the set of states that satisfy φ. In the above example, we observe that S and S' can be represented by the more compact CNF-formulae $\varphi = f \vee g \vee h$ and $\varphi' = f \vee \neg g$, respectively. Since each formula encodes a unique belief state, we want to use their compact encoding φ and φ' instead of the belief states S and S' of exponential size. To this end, we need to construct a transition function (e.g., $\Phi_{\mu CNF}$) such that (1) it returns a CNF-formula representing the successor belief state given a CNF-formula and an action (e.g., $\Phi_{\mu CNF}(a, \varphi) = \varphi'$) and (2) its computation involves operations over a small set of formulae instead of enumerating each individual in belief states.

To compute the result s' of execution of an action a in a state s, we need to compute the effect $e(a, s)$ of executing a in s, the set of literals that must be true in the successor state s'. Observe from Figure 1 that, the effects of executing a in s_1, s_3, and s_5 are all different. However, $e(a, s_1) = e(a, s_2) = \{\neg g\}$, $e(a, s_3) = e(a, s_4) = \{\neg f, \neg g\}$, and $e(a, s_5) = e(a, s_6) = e(a, s_7) = \{f\}$. Hence, if we divide $BS(\varphi)$ into three sets of states $S_1 = \{s_1, s_2\}$, $S_2 = \{s_3, s_4\}$, and $S_3 = \{s_5, s_6, s_7\}$ (Figure 2) then for each $i \in \{1, 2, 3\}$ and $u, v \in S_i$, we have that $e(a, u) = e(a, v)$. This is because the truth value of each e-condition of a in all the states in each subset S_i is the same and it can be different in states from different subsets (e.g., the e-condition f in $(a : \{f\} \rightarrow \{\neg g\})$ is true in all states s_1 and s_2 in S_1 and it is false in all states s_5, s_6, and s_7 in S_3.) Observe also that S_1, S_2, and S_3 can be represented by the formulae $\varphi_1 = f \wedge g$, $\varphi_2 = f \wedge \neg g$, and $\varphi_3 = \neg f \wedge (g \vee h)$ respectively. Since the effects of a in all states in $BS(\varphi_i)$, for each φ_i, are the same, we define $e(a, \varphi_i) = e(a, s)$ for $s \in BS(\varphi_i)$ and $i \in \{1, 2, 3\}$. Let $s'_j = \Phi(a, s_j)$ (for $j = 1, \ldots, 7$) and $S'_i = \Phi(a, S_i)$. Let φ'_i be the formula obtained by updating φ_i with the minimum change such that the literals in $e(a, \varphi_i)$ become true in the new formula φ'_i. Intuitively, $BS(\varphi'_i) = S'_i$ and, hence, $BS(\varphi'_1) \cup BS(\varphi'_2) \cup BS(\varphi'_3) = BS(\varphi')\ (= S')$. Indeed, it is easy to see that $\varphi'_1 = f \wedge \neg g$, $\varphi'_2 = \neg f \wedge \neg g$, $\varphi'_3 = f \wedge (g \vee h)$ and the disjunction $\varphi'_1 \vee \varphi'_2 \vee \varphi'_3$ can be simplified to the formula φ'. On the other hand, $S = S_1 \cup S_2 \cup S_3$ and $\varphi \equiv \varphi_1 \vee \varphi_2 \vee \varphi_3$. In summary, the computation of φ' includes three major steps as described in Figure 2.

As shown in our analysis and experiments [To, Son, and Pontelli, 2010a; To, Son, and Pontelli, 2010b; To, Son, and Pontelli, 2011c; To, Son, and Pontelli, 2015], the size of formulae representing belief states and, hence, the performance of a corresponding planner depend sub-

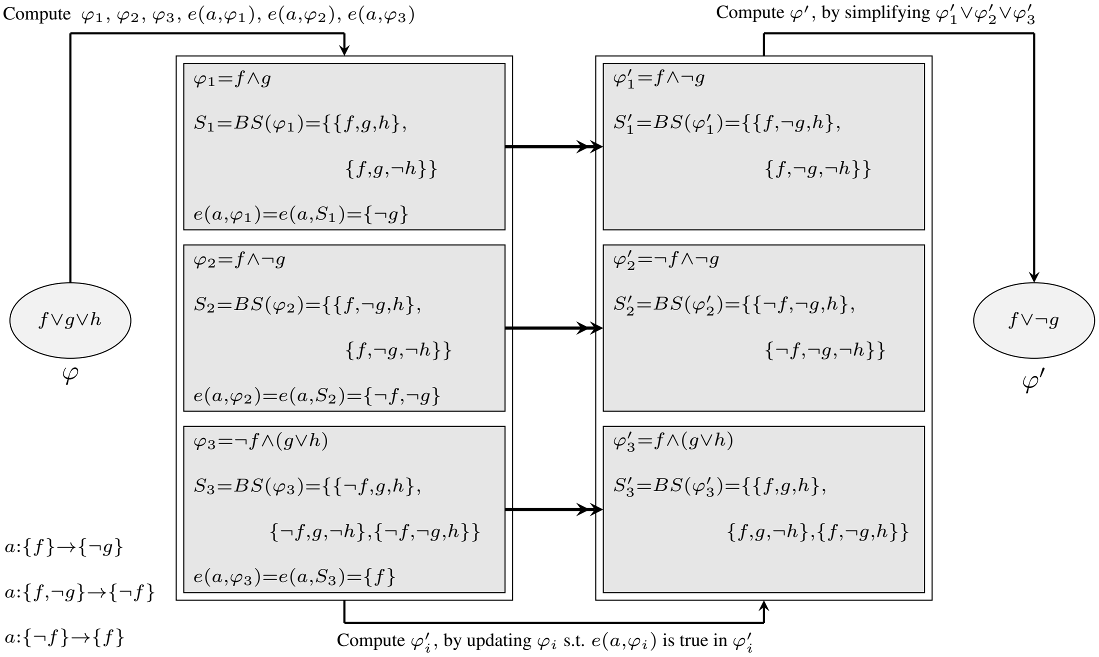
Compute φ_1, φ_2, φ_3, $e(a,\varphi_1)$, $e(a,\varphi_2)$, $e(a,\varphi_3)$
Compute φ', by simplifying $\varphi'_1 \vee \varphi'_2 \vee \varphi'_3$
$\varphi_1 = f \wedge g$
$S_1 = BS(\varphi_1) = \{\{f,g,h\},$
$\{f,g,\neg h\}\}$
$e(a,\varphi_1) = e(a,S_1) = \{\neg g\}$
$\varphi'_1 = f \wedge \neg g$
$S'_1 = BS(\varphi'_1) = \{\{f,\neg g,h\},$
$\{f,\neg g,\neg h\}\}$
$\varphi_2 = f \wedge \neg g$
$S_2 = BS(\varphi_2) = \{\{f,\neg g,h\},$
$\{f,\neg g,\neg h\}\}$
$e(a,\varphi_2) = e(a,S_2) = \{\neg f,\neg g\}$
$\varphi'_2 = \neg f \wedge \neg g$
$S'_2 = BS(\varphi'_2) = \{\{\neg f,\neg g,h\},$
$\{\neg f,\neg g,\neg h\}\}$
$f \vee g \vee h$
φ
$f \vee \neg g$
φ'
$\varphi_3 = \neg f \wedge (g \vee h)$
$S_3 = BS(\varphi_3) = \{\{\neg f,g,h\},$
$\{\neg f,g,\neg h\},\{\neg f,\neg g,h\}\}$
$e(a,\varphi_3) = e(a,S_3) = \{f\}$
$\varphi'_3 = f \wedge (g \vee h)$
$S'_3 = BS(\varphi'_3) = \{\{f,g,h\},$
$\{f,g,\neg h\},\{f,\neg g,h\}\}$
$a{:}\{f\} \rightarrow \{\neg g\}$
$a{:}\{f,\neg g\} \rightarrow \{\neg f\}$
$a{:}\{\neg f\} \rightarrow \{f\}$
Compute φ'_i, by updating φ_i s.t. $e(a,\varphi_i)$ is true in φ'_i

Figure 2: Computing successor formula φ' given action a and formula φ (Figure 1) in three steps

stantially on both the type of the formulae being used and the specific problem. For this reason, we propose a *generic* approach based on the abstract notion *representation*, instead of limiting to some specific types of formulae, aimed at the use of any type of propositional formulae to represent belief states.

Specifically, a collection $\mathcal{R}$ of formulae is a *representation of belief states* (or *representation*, for short) if for every belief state S there exists a formula $\varphi \in \mathcal{R}$ that represents S ($BS(\varphi) = S$). A formula φ in $\mathcal{R}$ is called an *$\mathcal{R}$-state*. A function $\Phi_{\mathcal{R}}$ that maps pairs composed of an action a and $\mathcal{R}$-state φ into $\mathcal{R}$-states is a *transition function for* $\mathcal{R}$ if $\Phi_{\mathcal{R}}(a, \varphi)$ represents the belief state $\Phi(a, BS(\varphi))$, i.e., $BS(\Phi_{\mathcal{R}}(a, \varphi)) = \Phi(a, BS(\varphi))$.

As shown in the above example (Figure 2), a transition function $\Phi_{\mathcal{R}}$ for $\mathcal{R}$ will be developed based on three computation steps as follows:

- **Step 1:** Compute the set of $\mathcal{R}$-states $\{\varphi_i \mid i = 1, \ldots, n\}$ such that $BS(\varphi_1) \cup \ldots \cup BS(\varphi_n) = BS(\varphi)$ and $\forall i. \forall s, s' \in BS(\varphi_i).\ e(a, s) = e(a, s')$. Compute $e(a, \varphi_i)$ for $i = 1, \ldots, n$;
- **Step 2:** For each $\mathcal{R}$-state φ_i, compute φ'_i by updating φ_i such that $e(a, \varphi_i)$ is true in φ'_i and nothing else changes;
- **Step 3:** Convert the disjunction $\varphi'_1 \vee \ldots \vee \varphi'_n$ into an equivalent $\mathcal{R}$-state φ', i.e., $BS(\varphi') = BS(\varphi'_1 \vee \ldots \vee \varphi'_n)$.

To realize the above computation steps, we then specify a set of abstract operations on a representation $\mathcal{R}$ and use these operations in defining the abstract transition function $\Phi_{\mathcal{R}}$. We devise a set of abstract algorithms for computing the transition function that can be instantiated in each concrete representation and used in the implementation of conformant planners.

We illustrate the generic framework by instantiating it in three concrete representations of belief states:

- *minimal-DNF*, a compact form of disjunctive normal form formulae,
- *minimal-CNF*, a compact form of conjunctive normal form formulae, and
- *prime implicates*, a well-known, special form of minimal-CNF.

For each representation, we define its set of operations, provide a concrete implementation of these operations, and instantiate the abstract algorithms to create a conformant planner. This results in three conformant planning systems, called DNF, CNF, and PIP, that employ the minimal-DNF, minimal-CNF, and prime implicates as the belief state representation, respectively.

We evaluate these systems using benchmarks available in the literature as well as new benchmarks, that appear to be particularly challenging for other planners. The experimental evaluation shows that the three systems are very competitive with other planners. In addition, the two systems using conjunctive representations (CNF and PIP) are particularly effective on the new problems.

Acknowledgements

Thanks to Knexus Research Corporation for sponsoring this article.

References

[Albore, Ramirez, and Geffner, 2011] Alexandre Albore, Miquel Ramirez, and Hector Geffner. Effective Heuristics and Belief Tracking for Planning with Incomplete Information. In Proceedings of *ICAPS*.

[Baral *et al.*, 2000] Chitta Baral, Vladik Kreinovich, and Raul Trejo. Computational complexity of planning and approximate planning in the presence of incompleteness. *AIJ*, 122:241–267.

[Bonet and Geffner, 2000] Blai Bonet and Hector Geffner. Planning with Incomplete Information as Heuristic Search in Belief Space. *International Conference on AI Planning Systems (AIPS).*

[Bryant, 1992] Randal E. Bryant. Symbolic boolean manipulation with ordered binary decision diagrams. *ACM Computing Surveys*, 24(3):293–318.

[Bryce *et al.*, 2006] Daniel Bryce, Subbarao Kambhampati, and David E. Smith. Planning Graph Heuristics for Belief Space Search. *Journal of Artificial Intelligence Research*, 26:35–99.

[Cimatti, Roveri, and Bertoli, 2004] Alessandro Cimatti, Marco Roveri, and P. Bertoli. Conformant Planning via Symbolic Model Checking and Heuristic Search. *Artificial Intelligence Journal*, 159:127–206.

[Goldman and Boddy 1996] Robert P. Goldman and Mark S. Boddy. 1996. Expressive Planning and Explicit Knowledge. In *Proc. of 3rd Int. Conf. AI Planning Systems*, 110–117.

[Haslum and Jonsson, 1999] Patrik Haslum and Peter Jonsson. Some results on the complexity of planning with incomplete information. In *Proc. of ECP-99, Lect. Notes in AI Vol 1809.* Springer.

[Hoffmann and Brafman, 2006] Jorg Hoffmann and Ronen I. Brafman. Conformant planning via heuristic forward search: A new approach. *Artificial Intelligence*, 170(6-7), 507-541.

[Hoffmann and Nebel, 2001] Jorg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14, 253-302.

[Nguyen *et al.*, 2012] Khoi H. Nguyen, Vien D. Tran, Tran C. Son, and Enrico Pontelli, 2012. On Computing Conformant Plans Using Classical Planners: A Generate-And-Complete Approach. In Proceedings of *ICAPS.*

[Palacios and Geffner, 2007] Hector Palacios and Hector Geffner. From Conformant into Classical Planning: Efficient Translations that may be Complete Too. In Proceedings of *ICAPS.*

[Palacios and Geffner, 2009] Hector Palacios and Hector Geffner. Compiling Uncertainty Away in Conformant Planning Problems with Bounded Width. In *JAIR* 35:623-675.

[Smith and Weld, 1998] David .E. Smith and Daniel S. Weld. Conformant graphplan. In *AAAI*, pages 889–896.

[Son and Baral, 2001] Tran C. Son and Chitta Baral. Formalizing sensing actions - a transition function based approach. *Artificial Intelligence*, 125(1-2):19–91, January 2001.

[Son and Tu, 2006] Tran C. Son and Phan H. Tu. On the Completeness of Approximation Based Reasoning and Planning in Action Theories with Incomplete Information. In *KRR*, 481–491.

[Son *et al.*, 2005] Tran C. Son, Phan H. Tu, Michael Gelfond, and A. Ricardo Morales Conformant Planning for Domains with Constraints — A New Approach. In *AAAI*, pages 1211–1216.

[Tran *et al.*, 2009] Vien D. Tran, Khoi H. Nguyen, Enrico Pontelli, and Tran C. Son. Improving performance of conformant planners: Static analysis of declarative planning domain specifications. In *PADL*, pages 239–253. Springer.

[To, Son, and Pontelli, 2015] Son T. To, Tran C. Son, and Enrico Pontelli. A generic approach to planning in the presence of incomplete information: Theory and implementation. In *Artificial Intelligence* 227, 1-51.

[To, Pontelli, and Son, 2009] Son T. To, Enrico Pontelli, and Tran C. Son. A Conformant Planner with Explicit Disjunctive Representation of Belief States. In Proceedings of *ICAPS.*

[To, Son, and Pontelli, 2010a] Son T. To, Tran C. Son, and Enrico Pontelli. A New Approach to Conformant Planning using CNF. In Proceedings of *ICAPS*

[To, Son, and Pontelli, 2010b] Son T. To, Tran C. Son, and Enrico Pontelli. On the Use of Prime Implicates in Conformant Planning. In Proceedings of *AAAI*

[To, Son, and Pontelli, 2011c] Son T. To, Tran C. Son, and Enrico Pontelli. Conjunctive Representations in Contingent Planning: Prime Implicates versus Minimal-CNF Formula. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI).*

[Tu *et al.*, 2011] Phan H. Tu, Tran C. Son, Michael Gelfond, and A. Ricardo Morales. Approximation of action theories and its application to conformant planning. In *Artificial Intelligence*, 175 79–119.