Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling

Fault Tolerant Planning: Complexity and Compilation

Carmel Domshlak
Technion
Haifa, Israel
dcarmel@ie.technion.ac.il

Abstract

In the context of modeling and reasoning about agent actions, contingent and classical planning can often be respectively seen as adopting "extreme pessimism" and "extreme optimism" about the action outcomes. For many everyday scenarios of human reasoning (and thus for many types of autonomous systems), both these approaches are just too extreme. Following Jensen, Veloso, and Bryant (2004), we examine a planning model that interpolates between classical and contingent planning via tolerance to arbitrary κ faults occurring during plan execution. We show that an important fragment of this *fault tolerant planning (FT-planning)* exhibits both an appealing solution structure, as well as appealing worst-case time-complexity properties. We also show that such FT-planning tasks can be efficiently compiled into classical planning as long as the number of possible faults per operator is bounded by a constant, and we show that this compilation can be attractive in practice.

Introduction

To date, contingent and classical planning appear to be the two major approaches to non-probabilistic planning under full observability. In contingent planning, at least some aspects of system dynamics are modeled by operators with non-deterministic effects, and a plan should guarantee reaching a goal state under any realization of the actions it prescribed. In classical planning, the operators are all set to be deterministic, modeling only the singular intended effects of each action. While contingent plans provide much stronger guarantees on reaching the goal with respect to the true physics of the modeled system, they are also much harder to generate (both worst-case and empirically), and quite often they may simply not exist.

In the physical world, no actions are really guaranteed to succeed. However, non-determinism in real-world domains is often caused by infrequent errors that make otherwise deterministic operators fail. Hence, many unsolvable contingent planning tasks become solvable if we *assume* that *no more than some κ exceptional/faulty action effects will occur along the purported plan to the goal*. In the past, this observation brought numerous researchers to consider explicit representation and reasoning about faults of agents' actions (Georgeff and Lansky 1986; Williams et al. 2003; Giunchiglia, Spalazzi, and Traverso 1994). In particular, Jensen, Veloso, and Bryant (2004) suggested a model of *fault tolerant planning (FT-planning)*, and developed first algorithms for generating plans that are robust for a single fault occurring during plan execution. This model is of our focus here.

Departing from contingent planning and generalizing the FT-planning model of Jensen, Veloso, and Bryant (2004), we show that, while FT-planning remains in general as computationally hard as contingent planning, one of its practicably most valuable fragments, namely the one considered by Jensen, Veloso, and Bryant (2004), (1) is in PSPACE, (2) falls into NP when restricted to plans with only polynomial-length executions, and (3) is guaranteed to admit stationary solutions for solvable problems, solutions that sometimes induce (possibly cyclic) strong contingent plans. Furthermore, we show that these FT-planning tasks can be *efficiently* compiled into equivalent classical planning tasks in a way that is *sound*, *complete*, and practicable.

Our results join a growing body of work on planning under uncertainty and/or partial observability via compilation to classical planning (Palacios and Geffner 2009; Albore, Palacios, and Geffner 2009; Bonet and Geffner 2011; Brafman and Shani 2012a; Taig and Brafman 2013). At a high level, FT-planning is an instance of "assumption-based planning," and the latter term has already been used for a broad range of ideas and techniques (Albore and Bertoli 2004; 2006; Albore and Geffner 2009; Bonet and Geffner 2011; Göbelbecker, Gretton, and Dearden 2011; Davis-Mendelow, Baier, and McIlraith 2012). Closest in spirit to our work here–though in two different ways–are probably the works of Albore and Bertoli (2006) and Davis-Mendelow, Baier, and McIlraith (2012). Albore and Bertoli suggested an interesting planning approach in which assumptions about operator effects are provided a priori as a linear temporal logic formula, and the planner takes these assumptions as axioms. In the worst-case, however, this approach remains as hard as contingent planning. Davis-Mendelow et al. exploit assumption-based assertions about the initial state to suggest a middle-ground between classical planning and conformant, or zero observability, planning. The latter, however, is very different from contingent planning, both conceptually and complexity-wise (Bonet 2010).

Planning Formalisms and Solution Concepts

Non-deterministic planning tasks with full observability correspond to succinctly represented, goal-oriented non-deterministic Markov decision processes (Puterman 1994). Several languages for succinctly representing such tasks are in use (Hoffmann and Brafman 2005; Bonet 2010; Davis-Mendelow, Baier, and McIlraith 2012). To simplify presentation, here we adopt a minimalistic extension of STRIPS (Fikes and Nilsson 1971) to non-deterministic operator effects.

A **planning task** is given by a quadruple $\Pi = \langle P, O, s_0, G \rangle$. P is a set of n propositions, with world states S being represented by complete valuations of P, and usually discussed as sets of propositions that hold true in them. $s_0 \in S$ is an initial state[1], and G is a subset of P: a state s is a goal state iff $G \subseteq s$, and the set of all goal states is denoted by S_G. O is a set of operators $o = \langle \mathsf{pre}(o), \mathsf{eff}(o) \rangle$ where the *precondition* pre is a subset of propositions P, and $\mathsf{eff} = \{e_1, \ldots, e_{n_o}\}$ is a set of *possible* effects of o. Each possible effect $e \in \mathsf{eff}$ is given by a pair $\langle \mathsf{add}(e), \mathsf{del}(e) \rangle$ of subsets of P, corresponding to its add and delete lists, respectively. An operator o is applicable in state s iff $\mathsf{pre}(o) \subseteq s$, and the set of all such operators is denoted by $O(s)$. If $o \in O(s)$ is applied in s, it changes the world to *one of* the states $Res[s; o] = \bigcup_{e \in \mathsf{eff}(o)} \{Res[s; e]\}$, where $Res[s; e] = (s \setminus \mathsf{del}(e)) \cup \mathsf{add}(e)$ is the state resulting from the effect e occurring in s. A (contingent) **plan** for a task Π is an action strategy that guarantees reaching a goal state $s \in S_G$ from s_0, under any realization of the operators applied along the way; the process of search for contingent plans is called **contingent planning**.

While nondeterministic operators must be dealt with in one form or another in many planning applications, two problems particular to contingent planning must be taken into account. First, deciding whether a contingent plan exists is EXPTIME-complete (Rintanen 2004).[2] Second, many tasks admit no contingent plans, and this is true even for simple tasks that humans feel comfortable dealing with (Cimatti et al. 2003; Pistore and Vardi 2007). In that respect, a pragmatic alternative to contingent planning is **classical planning**, operators are deterministic. By adopting classical planning as an abstraction of contingent planning, we *assume* that we know precisely what will happen when an operator o is applied in state s. This assumption is then "encoded" at the level of individual operators by what is called determinization, reducing the set of possible effects of each operator to exactly one effect.

While being much more restricted, classical planning resolves to a large extent the two aforementioned shortcomings of contingent planning. First, restricting each state/action pair to a sole possible successor often renders unsolvable problems solvable. Second, classical planning is in PSPACE (Bylander 1994), and more importantly, it is in NP if restricted to polynomial-length plans. Last but not least, classical plans are structurally simple, constituting linear sequences of operators. Together with NP-membership, this structural simplicity allows for exploiting various OR-graph search techniques for developing empirically efficient solvers for classical planning (Hoffmann and Nebel 2001; Helmert 2006; Rintanen, Heljanko, and Niemelä 2006; Kissmann and Edelkamp 2012). As a result, combining classical planning with online re-planning in unexpected situations is a popular and effective approach to closed-loop control of autonomous systems (Yoon et al. 2008; Talamadupula et al. 2010; Domshlak et al. 2011; Bonet and Geffner 2011; Brafman and Shani 2012b).

[1] We assume here that there is no uncertainty about the initial state, and later discuss the impact of this assumption.

[2] EXPTIME-completeness still holds even for testing the existence of plans that reach the goal with probability exceeding p for probabilistic problems with full observability (Littman 1997).

Fault Tolerant (Contingent) Planning

Given the relative pros and cons of contingent and classical planning, the first question one might ask is: If these are the two extremes, how can we interpolate between them in a simple and useful manner? This question brought us to consider *fault tolerant planning*: planning under the assumption that no more than some κ unintended effects of the operators will occur along the purported plan to the goal, but at the same time, under a requirement for the plans to be provably robust for up to κ such operator faults during plan execution. Fault tolerant planning was originally introduced by Jensen, Veloso, and Bryant (2004) in order to bring some key information from probabilistic uncertainty models to qualitative non-deterministic planning. The basic idea is to associate the contingent planning task at hand with an explicit distinction between the primary and exceptional effects of its operators. The model we adopt for that purpose is simply a function $\mathcal{F}$ that maps each possible effect of each operator to the "number of exceptions," or unintended artifacts, associated with this effect. The operator effects e for which $\mathcal{F}(e) = 0$ correspond to the primary effects of the respective operators in s.

Definition 1 *Let* $\Pi = \langle P, O, s_0, G \rangle$ *be a contingent planning task. An* **exception model** *for* Π *is a function* $\mathcal{F} : \bigcup_{o \in O} \mathsf{eff}(o) \rightarrow \mathbb{N}$*, computable in time polynomial in* $||\Pi||$*. If, for each operator* $o \in O$, $|\{e \mid e \in \mathsf{eff}(o), \mathcal{F}(e) = 0\}| \leq \alpha$*, then* $\mathcal{F}$ *is called* α**-primary***. Likewise, if, for each operator* $o \in O$, $|\{e \mid e \in \mathsf{eff}(o), \mathcal{F}(e) = 0\}| > 0$*, then* $\mathcal{F}$ *is called* **normative**.

In simple terms, an exception model is α-primary if *at most* α effects of each operator are considered to be its primary effects, and it is normative if each operator is associated with *at least* one primary effect. In these terms, the work of Jensen, Veloso, and Bryant (2004) has been devoted to fault tolerant planning under 1-primary normative exception models, which seem to cover well operator non-determinism that stems from physical complications of executing agent actions in the real world.[3] Associating non-deterministic operators with exception models allows for a

[3] If, however, some operators model knowledge acquisition, i.e., sensing, then (part of the) operator non-determinism will be due to primary operator effects, and thus planning with α-primary models for $\alpha > 1$ is not of theoretical interest only.

simple relaxation of contingent planning to planning under fault tolerance requirements as above. Let Π be a contingent planning task, $\mathcal{F}$ be an exception model for Π, and π be an action policy for Π. Overloading the notation, for an execution $\rho = (s_0, e_0, \ldots, s_i, e_i, \ldots)$ of π, we define $\mathcal{F}(\rho) \triangleq \sum_{i=0}^{\infty} \mathcal{F}(e_i)$.

– An execution ρ of π is called **κ-admissible** if $\mathcal{F}(\rho) \leq \kappa$.

– Action policy π is a **κ-plan** for Π if each of its κ-admissible executions is finite and reaches the goal.

In what follows, we refer to triplets $\langle \Pi, \mathcal{F}, \kappa \rangle$ as above as **fault tolerant (FT) planning tasks**, and solutions for such tasks are precisely κ-plans for Π under $\mathcal{F}$.

In general, κ-plan π can be either a stationary (possibly partial) policy $\pi : S \rightarrow O$, or a non-stationary policy $\pi : S \times \mathbb{N} \rightarrow O$ that depends on the current state and the number of "failures so far." As noted by Jensen et al. (2004), the latter can be captured as a stationary policy for a certain contingent planning task $\Pi^{(\mathcal{F},\kappa)}$ that we refer to as **$(\mathcal{F}, \kappa)$-reformulation** of Π: Given a FT-planning task $\langle \Pi = \langle P, O, s_0, G \rangle, \mathcal{F}, \kappa \rangle$, $\Pi^{(\mathcal{F},\kappa)}$ is a contingent planning task over states $S^{(\mathcal{F},\kappa)} = S \times \{0, \ldots, \kappa\}$, operators O, initial state $s_0^{(\mathcal{F},\kappa)} = (s_0, 0)$, and goal states $S_G^{(\mathcal{F},\kappa)} = \bigcup_{i=0}^{\kappa} \{(s, i) \mid s \in S_G\}$. For each $s \in S$, $o \in O(s)$, and $0 \leq i \leq \kappa$, o is applicable in (s, i), and if applied, it changes the world to one of the states

$$Res[(s,i);o] = \bigcup_{\substack{e \in \mathsf{eff}(o), \\ \kappa - (i + \mathcal{F}(e)) \geq 0}} \{Res[(s,i);e]\}, \tag{1}$$

where $Res[(s,i);e] = (Res[s;e], i + \mathcal{F}(e))$.

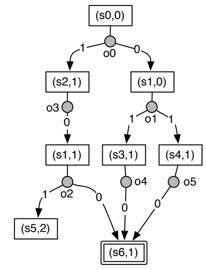
It is not hard to verify that there is a bijective correspondence between plans for $\Pi^{(\mathcal{F},\kappa)}$ and *non-stationary* κ-plans for $\langle \Pi, \mathcal{F}, \kappa \rangle$, but the relation to stationary κ-plans for $\langle \Pi, \mathcal{F}, \kappa \rangle$ is less immediate. Theorem 1 clarifies this matter. Let π be a (not necessarily a plan) policy for $\Pi^{(\mathcal{F},\kappa)}$. The *execution tree* $T_\pi(s, i)$ is the tree of possible executions of π starting at (s, i), with nodes corresponding to states of $\Pi^{(\mathcal{F},\kappa)}$, edges corresponding to operator effects, and $S_\pi^{(\mathcal{F},\kappa)} \subseteq S^{(\mathcal{F},\kappa)}$ denoting the set of *internal* nodes of T_π. (Henceforth, $T_\pi(s_0, 0)$ is referred to for short as T_π.)

Theorem 1 *Let $\langle \Pi, \mathcal{F}, \kappa \rangle$ be a solvable FT-planning task. If $\mathcal{F}$ is normative, then there exists a stationary κ-plan π for $\langle \Pi, \mathcal{F}, \kappa \rangle$. In contrast, there exist solvable FT-planning tasks (with nonnormative exception models) for which there are no stationary κ-plans.*

The proof of the second sub-claim is by example: Let Π be a contingent planning task over states $S = \{s_0, \ldots, s_6\}$, operators $O = \{o_0, \ldots, o_5\}$ as in Figure 1a, initial state s_0, and $S_G = \{s_6\}$. Assume an exception model $\mathcal{F}$ for Π as in the last column of Figure 1a. Figure 1b depicts the only contingent plan for the reformulation $\Pi^{(\mathcal{F},1)}$, and the respective 1-plan for $\langle \Pi, \mathcal{F}, 1 \rangle$ is not stationary: different actions, o_1 and o_2, are taken at state s_1 with 0 and 1 "exceptions so far," respectively, and such a history-dependent choice of operator at s_1 is unavoidable.

op	pre	eff	$\mathcal{F}$
o_0	s_0	$e_1 = \langle s_1, s_0 \rangle$	$\mathcal{F}(e_1) = 0$
		$e_2 = \langle s_2, s_0 \rangle$	$\mathcal{F}(e_2) = 1$
o_1	s_1	$e_1 = \langle s_3, s_1 \rangle$	$\mathcal{F}(e_1) = 1$
		$e_2 = \langle s_4, s_1 \rangle$	$\mathcal{F}(e_2) = 1$
o_2	s_1	$e_1 = \langle s_6, s_1 \rangle$	$\mathcal{F}(e_1) = 0$
		$e_2 = \langle s_5, s_1 \rangle$	$\mathcal{F}(e_2) = 1$
o_3	s_2	$e_1 = \langle s_1, s_2 \rangle$	$\mathcal{F}(e_1) = 0$
o_4	s_3	$e_1 = \langle s_6, s_3 \rangle$	$\mathcal{F}(e_1) = 0$
o_5	s_4	$e_1 = \langle s_6, s_4 \rangle$	$\mathcal{F}(e_1) = 0$

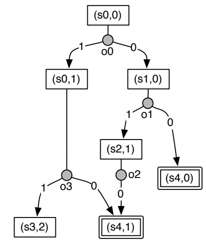
(a)



(b)

op	pre	eff	$\mathcal{F}$
o_0	s_0	$e_1 = \langle s_1, s_0 \rangle$	$\mathcal{F}(e_1) = 0$
		$e_2 = \langle \emptyset, \emptyset \rangle$	$\mathcal{F}(e_2) = 1$
o_1	s_1	$e_1 = \langle s_4, s_1 \rangle$	$\mathcal{F}(e_1) = 0$
		$e_2 = \langle s_2, s_1 \rangle$	$\mathcal{F}(e_2) = 1$
o_2	s_2	$e_1 = \langle s_4, s_2 \rangle$	$\mathcal{F}(e_1) = 0$
o_3	s_0	$e_1 = \langle s_4, s_0 \rangle$	$\mathcal{F}(e_1) = 0$
		$e_2 = \langle s_3, s_0 \rangle$	$\mathcal{F}(e_2) = 1$

(c)



(d)

Figure 1: Illustrations for the examples around Theorem 1.

Note that $\mathcal{F}$ in the above example is not normative because neither of the effects of operator o_1 is primary. With normative exception models, the situation is indeed different. Let $\langle \Pi, \mathcal{F}, \kappa \rangle$ be a solvable FT-planning task with a normative model $\mathcal{F}$, and π be a contingent plan for $\Pi^{(\mathcal{F},\kappa)}$. If $\pi(s, i) = \pi(s, j)$ for all pairs of reformulation states $(s, i), (s, j) \in S_\pi^{(\mathcal{F},\kappa)}$, then we are done because the κ-plan for $\langle \Pi, \mathcal{F}, \kappa \rangle$ corresponding to π is stationary. Otherwise, let $(s, i), (s, j) \in S_\pi^{(\mathcal{F},\kappa)}$, $i < j$ be a pair of reformulation states for which $\pi(s, i) \neq \pi(s, j)$. The proof is accomplished by showing that π', obtained from π by replacing $\pi(s, j)$ with $\pi(s, i)$, is also a plan for $\Pi^{(\mathcal{F},\kappa)}$, and thus we can always iteratively reduce a non-stationary κ to a stationary one.

Note that, by the construction used in the proof of Theorem 1, if π is a non-stationary κ-plan for a FT-planning task $\langle \Pi, \mathcal{F}, \kappa \rangle$ with a normative model $\mathcal{F}$, then π can be efficiently translated into a stationary κ-plan π' for $\langle \Pi, \mathcal{F}, \kappa \rangle$. Likewise, for some of such pairs π and π', π' may turn out to be a strong cyclic contingent plan for Π. For instance, let Π be a contingent planning task over states $S = \{s_0, \ldots, s_4\}$, operators $O = \{o_0, \ldots, o_3\}$, initial state s_0, and $G = s_4$. The operators are defined as in the table in Figure 1c, and the exception model $\mathcal{F}$ associated with Π is given in the last column of that table. Figure 1d depicts a contingent plan π for the reformulation $\Pi^{(\mathcal{F},\kappa)}$, and the respective 1-plan for $\langle \Pi, \mathcal{F}, 1 \rangle$ is not stationary: different actions, o_0 and o_3, are taken at state s_0 with 0 and 1 "exceptions so far," respectively. However, if we modify π as in the proof of Theorem 1, the resulting plan π' for $\Pi^{(\mathcal{F},\kappa)}$ will induce a strong cyclic contingent plan $\pi^*(s_i) = o_i$ for Π.

Complexity and Compilation

Two decision problems are of interest in the context of FT-planning: Let Π be a contingent planning task, and $\mathcal{F}$ be an α-primary model for Π.

FT-PLAN-α-κ**:** Does Π have a κ-plan?

POLY-FT-PLAN-α-κ: Does Π have a κ-plan such that all its κ-admissible executions reach the goal after a polynomial number of steps?

At first view, the effective difference between FT-PLAN-α-κ and contingent planning is not clear. In general, for sufficiently large values of α (e.g., $\alpha = |S|$), contingent planning can trivially be reduced to FT-PLAN-α-κ for *any* κ, and hence the latter decision problem is EXPTIME-hard. In fact, FT-PLAN-α-κ can be polynomially reduced to FT-PLAN-2-κ by simulating each operator with α primary effects by a "ladder" of $\log \alpha$ operators, each with at most two primary effects. Hence, even FT-PLAN-2-κ is EXPTIME-hard. However, while the definition of exception models is rather general, the specific settings that brought us to this investigation correspond to the normative 1-primary exception models considered by Jensen, Veloso, and Bryant (2004). In what comes next, we focus on that fragment of FT-planning.[4]

FT-planning with 1-primary models

Unlike FT-PLAN-2-κ, FT-PLAN-1-κ nicely generalizes classical planning, which simply corresponds to first associating the contingent planning task with a normative 1-primary model, and then adopting the extreme optimism that no failures will occur along the purported plan to the goal. In other words, *the decision version of classical planning is precisely the FT-planning class* FT-PLAN-1-0. At the same time, FT-PLAN-1-1 already goes way beyond classical planning. While plans for FT-PLAN-1-1 are restricted to at most one operator failure per possible plan execution, these failures are bounded neither to specific operators nor to specific stages of the purported plan. Hence, while plans for FT-PLAN-1-0 are linear sequences of actions, plans for FT-PLAN-1-κ with $\kappa > 0$ are tree-structured, and may actually exhibit substantial branching: unlike in (EXPSPACE-complete) planning under the k-branching assumption (Bonet 2010), plans for FT-PLAN-1-κ may have to always interleave between acting and branching, even for $\kappa = 1$.

For example, suppose that a robot should move from x_1 to x_5 on the map depicted in Figure 2(a). Movements on the segments (x_1, x_2) and (x_1, x_3) are considered safe and thus are modeled by deterministic operators. Movements $move(x_i, x_j)$ on the other three segments are modeled by non-deterministic operators with three possible effects: $move(x_i, x_j)$ typically brings the robot to x_j, with no side effects, but it may also bring the robot to x_j with a flat tire, or keep it at x_i for the same reason. Initially the robot has no flats, but also no spare tires. A single spare tire can be picked up at each of the two intermediate locations x_2 and x_3. Figures 2(b-d) depict stationary 0-plan, 1-plan, and 2-plan for the respective FT-planning tasks, under

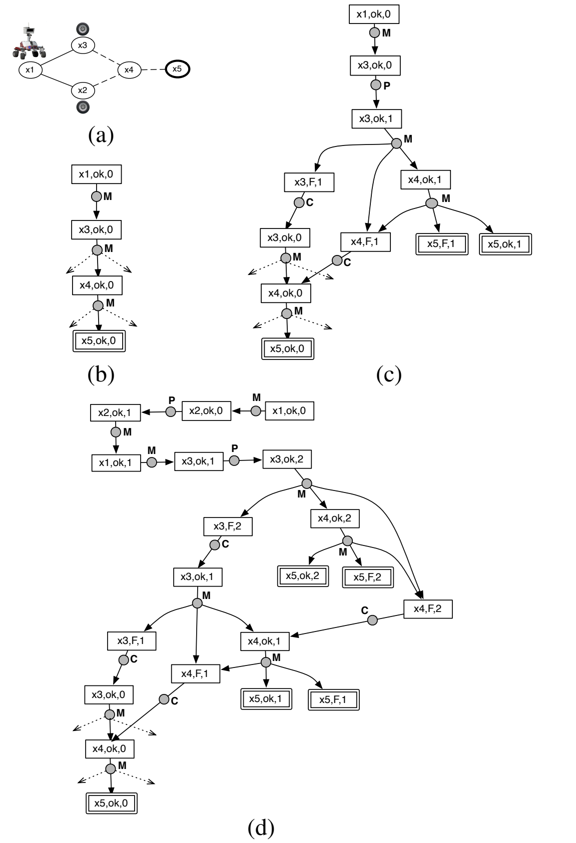


Figure 2: κ-plans for an inline example

a normative 1-primary exception model that maps the primary effects of all actions to 0, and the exceptional effects of the non-deterministic move actions to 1. In the triplet denotation $[x, y, z]$ of the states, x is the robot's location, $y \in \{\text{ok}, \text{F}\}$ is the status of the tire, and z is the number of spare tires in the robot's possession. State representation in this problem also addresses the availability of the spare tires at x_2 and x_3, but we omit this information in the figure for brevity. The dashed arrows depict possible effects of the actions that the agent ignores at planning. The 0-plan is a classical plan, and it is as simple as the example itself. The 1-plan is already more involved: to guarantee reaching x_5 under a possibility of a single fault, the robot picks up a spare tire at x_3. In turn, the 2-plan in Figure 2d prescribes that the robot first collect both spare tires at x_2 and x_3, and only then start moving towards x_5, replacing flat tires on the way, if needed.

Still, despite the structural complexity of κ-plans, the EXPTIME-hardness proof for FT-PLAN-2-κ does not carry over to FT-PLAN-1-κ, and for a good reason: Theorem 2 below shows that FT-PLAN-1-κ is in PSPACE, that is, worst-case not harder than classical planning. Moreover, Theorem 3 then shows even closer resemblance between these two formalisms, namely that POLY-FT-PLAN-1-κ is in NP.

Theorem 2 FT-PLAN-1-κ *is in* PSPACE.

A non-deterministic algorithm (that can also be compiled to a Turing machine) for deciding whether there is a κ-plan

[4]While the motivation for the FT-PLAN-1-κ fragment comes from its excellent applicability in practice, it is obviously not the only fragment of FT planning to be so motivated. For instance, if some operators model knowledge acquisition, i.e., sensing, then (part of the) operator non-determinism will be due to primary operator effects, and thus planning with α-primary models for $\alpha > 1$ is not of a theoretical interest only.

Algorithm BO-PLAN-1-$\kappa(\Pi, \mathcal{F})$

main
PLAN-FT(s_0, κ)
accept

procedure *Plan-FT*(s, k)
$steps \leftarrow 0$
while $steps < 2^n$
do
if $s \models G$ **then return**
choose operator o s.t. $s \models \mathsf{pre}(o)$
for $e \in \mathsf{eff}(o)$
do
if $\mathcal{F}(e) > k$
then continue *// under assumed κ, e cannot happen here*
else if $\mathcal{F}(e) > 0$
then *Plan-FT*$(Res[s; e], k - \mathcal{F}(e))$
else
// proceed with the primal effect of $\langle \mathsf{pre}, \mathsf{eff} \rangle$ at s
$steps \leftarrow steps + 1$
$s \leftarrow Res[s; e]$
reject

Figure 3: PSPACE algorithm for deciding FT-PLAN-1-κ.

for an FT-planning task $\langle \Pi, \mathcal{F}, \kappa \rangle$ with 1-primary $\mathcal{F}$ and $|P| = n$ is depicted in Figure 3. The respective Turing machine is in PSPACE because, at any point, there are at most κ open calls to the *Plan-FT* procedure, each storing a single state in n bits and a single counter $steps$ in n bits. Finally, since it is in PSPACE, FT-PLAN-1-κ is also PSPACE-complete by the PSPACE-hardness of classical planning under the description language we use, and equivalence of the latter to FT-PLAN-1-0.

Theorem 3 POLY-FT-PLAN-1-κ *is in* NP.

The proof is by showing that, if $\langle \Pi, \mathcal{F}, \kappa \rangle$ has a κ-plan π' such that all its κ-admissible executions reach the goal after $O(n^c)$ steps, then there is a κ-plan π with the same property such that $||\pi|| = O(b^{(\kappa+1)} n^{c(\kappa+2)})$, where $b = \max_{o \in O} |\mathsf{eff}(o)|$. Since $b = O(||\Pi||)$ and both c and κ are $O(1)$, the rest stems from the standard guess-and-verify argument of NP-membership.

First, since κ-plans guarantee goal reachability only along executions with κ or fewer exceptions, let π follow π' on states reachable by the κ-admissible executions of π', and make a random operator choice everywhere else. Thus, only the κ-admissible executions of π should be represented. Second, as we upper-bound the description size of π, for simplicity we assume (i) extensive, tree-structured representation of π, and (b) that the range of $\mathcal{F}$ is $\{0, 1\}$. For $0 \leq i \leq \kappa$, let $f(\pi, i)$ be the number of i-admissible executions of π that are not $(i-1)$-admissible. Clearly, $\sum_{i=0}^{\kappa} f(\pi, i)$ is the overall number of κ-admissible executions, and thus $||\pi|| = O\left(n^c \sum_{i=0}^{\kappa} f(\pi, i)\right)$. Since $\mathcal{F}$ is 1-primary, there is at most one 0-admissible execution of π per possible initial state, and thus $f(\pi, 0) = 1$. Recursively, due to the same argument of $\mathcal{F}$ being 1-primary, each of the $O(n^c)$ operator instances along that single 0-admissible execution may branch into b 1-admissible executions of π. However, these are the only possible sources of 1-admissible executions. Thus, $f(\pi, 1) = O(f(\pi, 0) \cdot bn^c) = O(bn^c)$, and in general, $f(\pi, i) = O((bn^c)^i)$. Hence, $||\pi|| = O\left(n^c \cdot (bn^c)^{\kappa+1}\right) = O(b^{(\kappa+1)} n^{c(\kappa+2)})$.

Note that, while $\kappa = O(1)$ should suffice for most interests in practice, the PSPACE-membership result of Theorem 2 holds for $\kappa = O(\mathrm{poly}(||\Pi||))$. This is not so, however, with the NP-membership result of Theorem 3, which relies upon $\kappa = O(1)$.

Compilation to Classical Planning

Theorems 2 and 3 put FT-planning under 1-primary exception models rather close to classical planning. On the one hand, that suggests that classical planning machinery can possibly be adapted to solve such FT-planning tasks. On the other hand, the non-linearity of κ-plans under 1-primary models seems to complicate applying classical planning algorithms to FT-planning.

We now show that FT-planning under 1-primary models can be efficiently compiled into classical planning, at least as long as the number of non-deterministic effects per operator is bounded by a constant. The compilation is to STRIPS with negative preconditions and conditional effects. In this formalism, a planning task is given by a quadruple $\Pi = \langle P, O, s_0, G \rangle$, with P, s_0, and G being as in our formalism for contingent planning. Operators $o \in O$ are pairs $\langle \mathsf{pre}(o), \mathsf{eff}(o) \rangle$ where the precondition $\mathsf{pre}(o)$ is a subset of literals over P, and $\mathsf{eff}(o)$ is a set of conditional effects. A conditional effect e is a triplet $\langle \mathsf{con}(e), \mathsf{add}(e), \mathsf{del}(e) \rangle$ of condition, add, and delete lists, respectively, where $\mathsf{con}(e)$ is a subset of literals over P, while $\mathsf{add}(e)$ and $\mathsf{del}(e)$ are subsets of propositions. Operator o is applicable in state s iff $s \models \mathsf{pre}(o)$. If $o \in O(s)$ is applied in s, it deterministically changes the world to state $res[s; o] = \bigcup_{e \in \mathsf{eff}(o), s \models \mathsf{con}(e)} (s \setminus \mathsf{del}(e)) \cup \mathsf{add}(e)$.

We start with a compilation of a simple fragment of FT-PLAN-1-κ, corresponding to FT-planning tasks $\langle \Pi = \langle P, O, s_0, G \rangle, \mathcal{F}, \kappa \rangle$ such that (i) each operator of Π has at most two effects, and (ii) $\mathcal{F}$ is a normative 1-primary exception model for Π such that, for each $o \in O(s)$, if $\mathsf{eff}(o) = \{e_0\}$, then $\mathcal{F}(e_0) = 0$, and if $\mathsf{eff}(o) = \{e_0, e_1\}$, then $\mathcal{F}(e_0) = 0$ and $\mathcal{F}(e_1) = 1$. We begin with an example that illustrates the basic idea behind this compilation. Let $\kappa = 2$, and let Figure 4a depict an irreducible contingent plan π for $\Pi^{(\mathcal{F}, \kappa)}$: the arcs correspond to the operator effects and are labeled with the respective values of $\mathcal{F}$, and double-frame states are the goal states.

Note that, in Figure 4a, the states and operator instances in π are numbered consistently with a DFS traversal of the execution tree T_π. Therefore, the operator sequence $o_1, \ldots, o_7$ induces a sequence of policies $\pi_0, \ldots, \pi_7$ for $\Pi^{(\mathcal{F}, \kappa)}$ such that π_0 is an empty policy, $\pi_7 = \pi$, and each π_i extends π_{i-1} with mapping a single leaf of $T_{\pi_{i-1}}$ to operator o_i. An important property of this sequence of policies $\pi_0, \ldots, \pi_7$ is that, for $0 \leq j \leq 2$, each π_i induces *at most one* execution ρ with $\mathcal{F}(\rho) = j$ that does not achieve the goal within T_{π_i}. The latter is emphasized by the tabular representation of this sequence of policies in Figure 4b. The columns in the table capture certain *subsets* $\sigma_0, \ldots, \sigma_7$ of leaves of $T_{\pi_0}, \ldots, T_{\pi_7}$, that is, of the end-states of $\pi_0, \ldots, \pi_7$, respectively. For

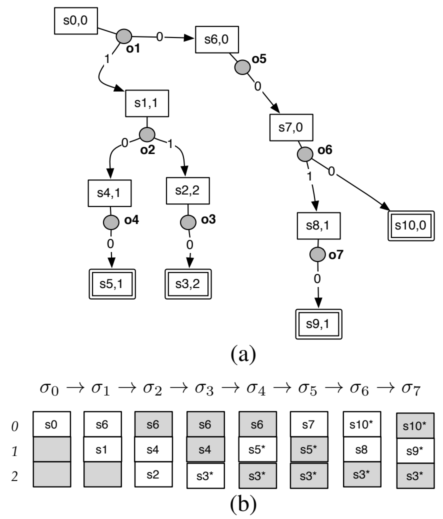


Figure 4: A plan π for $\Pi^{(\mathcal{F},\kappa)}$, and the end-state set representation of the induced sequence of π's sub-policies.

$0 \leq j \leq 2$, each σ_i contains at most one state with "j failures so far," denoted $\sigma_i(j)$, with $\sigma_i(j) = \perp$ denoting that σ_i does not contain a state for j. For $i = 0$, $\sigma_0(0) = (s_0, 0)$, and $\sigma_0(1) = \sigma_0(2) = \perp$. For $i > 0$, if T_{π_i} has a (unique) leaf (s, j) such that $s \notin S_G$, then $\sigma_i(j) = (s, j)$, and otherwise, $\sigma_i(j) = \sigma_{i-1}(j)$. Asterisks in the table are by the goal states, and note that the last set σ_7 is the first in the sequence to contain only goal states.

It turns out that any irreducible κ-plan for any FT-planning task from the fragment in question induces such a DFS-ordered sequence of sub-policies with "at most one non-goal leaf with j failures so far." It is precisely this property that provides a basis for our compilation of $\langle \Pi = \langle P, O, s_0, G \rangle, \mathcal{F}, \kappa \rangle$ to an equivalent classical planning task $\Pi' = \langle P', O', \sigma'_0, G' \rangle$. For now, we postpone the formal statement and proof of this property, and formulate the actual compilation. After that, in Lemma 5, we formally state the task properties underlying the compilation, and use this lemma to specify compilation for a wider class of FT-planning tasks.

Propositions. For clarity, we use letters s and σ to denote states of Π and Π', respectively. The subsets $\sigma_0, \ldots, \sigma_7$ of $S^{(\mathcal{F},\kappa)}$ in Figure 4b hint at the state space structure of Π': Each reachable state σ of Π' represents a partial policy π_σ for $\Pi^{(\mathcal{F},\kappa)}$ that corresponds to a concrete stage of a certain DFS traversal of the purported tree-structured plan for $\Pi^{(\mathcal{F},\kappa)}$. To support that, P' contains $\kappa + 1$ replicas $P_0, \ldots, P_\kappa$ of P, as well as a set of auxiliary propositions $\{\mathit{open}_i\}_{i=0}^{\kappa}$. The interpretation of $\mathit{open}_i \in \sigma$ is that "the policy for $\Pi^{(\mathcal{F},\kappa)}$ represented by σ induces an i-admissible execution that does not achieve the goal, and the end-state of that execution is captured by the values of P_i in σ." In what follows, by σ/P_i we denote the valuation provided by σ to propositions P_i. Likewise, if ϕ is a set of literals either over P or over one of the proposition sets $P_0, \ldots, P_\kappa$ of Π', then, for $0 \leq i \leq \kappa$, $\lfloor \phi \rfloor_i$ is a set of literals over P_i, obtained from ϕ by replacing all the propositional symbols with their counterparts in P_i. For example, $\lfloor \{p, \neg q\} \rfloor_i = \{p_i, \neg q_i\}$, and $\lfloor \{p_i, \neg q_i\} \rfloor_j = \{p_j, \neg q_j\}$.

Operators. O' contains $\kappa + 1$ sets of operators $O_0, \ldots, O_\kappa$, with $O_i = \{o_i \mid o \in O\}$. For each $o \in O$, the precondition of the operator o_i is

$$\mathsf{pre}(o_i) = \lfloor \mathsf{pre}(o) \rfloor_i \cup \{\mathit{open}_i\} \cup \bigcup_{j=i+1}^{\kappa} \{\neg \mathit{open}_j\}.$$

In other words, if state σ of Π' represents a policy for $\Pi^{(\mathcal{F},\kappa)}$ such that some admissible executions of that policy do not reach the goal, then the planner is forced to extend the i-admissible such execution with the *highest* index i. If either o is deterministic or $i = \kappa$, then $\mathsf{eff}(o_i) = Res[\sigma/P_i; \lfloor e_0 \rfloor_i]$. Otherwise, if $\mathsf{eff}(o) = \{e_0, e_1\}$ and $i < \kappa$, then

$$\mathsf{eff}(o_i) = Res[\sigma/P_i; \lfloor e_0 \rfloor_i] \wedge \lfloor Res[\sigma/P_i; \lfloor e_1 \rfloor_i] \rfloor_{i+1} \wedge \mathit{open}_{i+1}.$$

In the formalism of our choice, such operator effects are captured as follows. If $\mathsf{eff}(o) = \{e_0\}$ or $i = \kappa$, then $\mathsf{eff}(o_i)$ contains a single unconditional effect:

$$\mathsf{eff}(o_i) = \{\ \langle \emptyset, \lfloor \mathsf{add}(e_0) \rfloor_i, \lfloor \mathsf{del}(e_0) \rfloor_i \rangle\ \}.$$

Otherwise, if $\mathsf{eff}(o) = \{e_0, e_1\}$ and $i < \kappa$, then

$$\mathsf{eff}(o_i) = \Phi_i \cup \left\{ \begin{array}{llll} \langle \emptyset, & \lfloor \mathsf{add}(e_0) \rfloor_i, & \lfloor \mathsf{del}(e_0) \rfloor_i & \rangle, \\ \langle \emptyset, & \{\mathit{open}_{i+1}\}, & \emptyset & \rangle, \\ \langle \emptyset, & \lfloor \mathsf{add}(e_1) \rfloor_{i+1}, & \lfloor \mathsf{del}(e_1) \rfloor_{i+1} & \rangle \end{array} \right\},$$

where

$$\Phi_i = \bigcup_{p \notin \mathsf{add}(e_1) \cup \mathsf{del}(e_1)} \left\{ \begin{array}{llll} \langle \{p_i\}, & \{p_{i+1}\}, & \emptyset & \rangle, \\ \langle \{\neg p_i\}, & \emptyset, & \{p_{i+1}\} & \rangle \end{array} \right\}$$

compactly encodes the situation calculus frame axioms between the "current situation with i failures so far" and the "next situation with $i + 1$ failures so far." In addition, O' contains a set of auxiliary "goal-achieving" operators $\{o_i^*\}_{i=0}^{\kappa}$ with

$$\mathsf{pre}(o_i^*) = \lfloor G \rfloor_i \cup \{\mathit{open}_i\} \cup \bigcup_{j=i+1}^{\kappa} \{\neg \mathit{open}_j\},$$

$$\mathsf{eff}(o_i^*) = \{\ \langle \emptyset, \emptyset, \{\mathit{open}_i\} \rangle\ \}.$$

Initial state and goal. The P_0-part of the initial state σ'_0 captures the sole initial state of Π, and for $i > 0$, the P_i-parts of the initial state are actually not important, and can be set arbitrarily. The auxiliary variable open_0 is initially set to true, the rest of the variables open_i are initially set to false, and the goal of Π' is to negate open_0. That is, $\sigma'_0 = \lfloor s_0 \rfloor_0 \cup \{\mathit{open}_0\}$ and $G' = \{\neg \mathit{open}_0\}$.

For an illustration, consider a small and simplified variant of our running example in which there are only two locations, x and $not(x)$, the robot and a single spare tire are initially at x, and the goal is for the robot to be at $not(x)$. Movement of the robot either succeeds (which is the primary effect of that operator), or fails, with the robot staying at the original location with a flat tire. This planning task Π is encoded using propositions $P = \{x, \mathit{noflat}, \mathit{spare}\}$ by the operators as in the Table 1a, initial state $s_0 = \{x, \mathit{spare}, \mathit{noflat}\}$, and goal $G =$

	o	pre	eff	$\mathcal{F}$
(a)	move	$\{x, \mathit{noflat}\}$	$e_0 = \langle \emptyset, \{x\} \rangle$	$\mathcal{F}(e_0) = 0$
			$e_1 = \langle \emptyset, \{\mathit{noflat}\} \rangle$	$\mathcal{F}(e_1) = 1$
	fix	$\{x, \mathit{spare}\}$	$e_0 = \langle \{\mathit{noflat}\}, \{\mathit{spare}\} \rangle$	$\mathcal{F}(e_0) = 0$

	o	pre	eff
(b)	move_0	$\left\{ \begin{array}{c} x_0, \\ \mathit{noflat}_0, \\ \mathit{open}_0, \\ \neg \mathit{open}_1 \end{array} \right\}$	$\left\{ \begin{array}{lll} \langle \emptyset, & \emptyset, & \{x_0\} \rangle \\ \langle \emptyset, & \{\mathit{open}_1\}, & \emptyset \rangle \\ \langle \emptyset, & \emptyset, & \{\mathit{noflat}_1\} \rangle \\ \langle \{x_0\}, & \{x_1\}, & \emptyset \rangle \\ \langle \{\neg x_0\}, & \emptyset, & \{x_1\} \rangle \\ \langle \{\mathit{spare}_0\}, & \{\mathit{spare}_1\}, & \emptyset \rangle \\ \langle \{\neg \mathit{spare}_0\}, & \emptyset, & \{\mathit{spare}_1\} \rangle \end{array} \right\}$
	move_1	$\{x_1, \mathit{noflat}_1, \mathit{open}_1\}$	$\{\langle \emptyset, \emptyset, \{x_1\} \rangle\}$
	fix_0	$\{x_0, \mathit{spare}_0, \mathit{open}_0, \neg \mathit{open}_1\}$	$\{\langle \emptyset, \{\mathit{noflat}_0\}, \{\mathit{spare}_0\} \rangle\}$
	fix_1	$\{x_1, \mathit{spare}_1, \mathit{open}_1\}$	$\{\langle \emptyset, \{\mathit{noflat}_1\}, \{\mathit{spare}_1\} \rangle\}$
	o^*_0	$\{\neg x_0, \mathit{open}_0, \neg \mathit{open}_1\}$	$\{\langle \emptyset, \emptyset, \{\mathit{open}_0\} \rangle\}$
	o^*_1	$\{\neg x_1, \mathit{open}_1\}$	$\{\langle \emptyset, \emptyset, \{\mathit{open}_1\} \rangle\}$

Table 1: Operators from the compilation example.

$\{\neg x\}$. The compilation $\Pi' = \langle P', O', \sigma'_0, G' \rangle$ of the FT-planning task $\langle \Pi, \mathcal{F}, 1 \rangle$ is defined over propositions $P' = \bigcup_{i \in \{0,1\}} \{x_i, \mathit{noflat}_i, \mathit{spare}_i, \mathit{open}_i\}$, and operators as in Table 1b. The initial state is $\sigma'_0 = \{x_0, \mathit{spare}_0, \mathit{noflat}_0, \mathit{open}_0\}$, and the goal is $G' = \{\neg \mathit{open}_0\}$. It is not hard to verify that $\pi = \langle \mathsf{move}_0, \mathsf{fix}_1, \mathsf{move}_1, o^*_1, o^*_0 \rangle$ is (the only) plan for the classical planning task Π', and that the respective contingent plan for $\Pi^{(\mathcal{F},1)}$ can be decoded from π in linear time.

In the spirit of this example, we have generated a set of tasks in which a robot should move from the bottom-left to the top-right corner of a 4-connected grid, in which some of the edges are "safe," with moves along them being deterministic, while other edges are "unsafe," with moves on them either succeeding (which is the primary, expected effect of that operator) or resulting in the robot getting a flat and staying where it was. A limited number of spare tires are available on some nodes of the grid. The six sets of five tasks each correspond to 5×5 and 7×7 grids, with each edge of the grid being independently marked as safe with probability $p \in \{0.1, 0.2, 0.5\}$, and 10 spare tires, independently positioned on the grid nodes at random.

The runtimes of different approaches on these tasks are depicted in Table 2. The three approaches we examined were (col. 2) contingent planning with Contingent-FF (Hoffmann and Brafman 2005); (col. 3-6) FT-planning with Contingent-FF over $(\mathcal{F}, \kappa)$-reformulations, $\kappa \in \{0, 1, 2, 4\}$, and (col. 7-10) FT-planning with Fast Downward's *GBFS* with FF heuristic over the classical planning reductions of the $(\mathcal{F}, \kappa)$-reformulations. Each task/planner was given a 10 minute time limit; cases in which the planner neither solved the task nor proved it unsolvable within the time bound are marked with '-'. If a planner solved a task within the time bound, then the respective entry in the table is shaded.

As Table 2 shows, all but one of these tasks were proven by Contingent-FF to have no strong contingent plans (and cyclic contingent plans are also of no help in this domain), while all (effectively classical) FT-planning tasks with $\kappa = 0$ were easily solved by both Contingent-FF and Fast Downward. For us, of course, the interesting part was in between these two extremes, and both Contingent-FF and Fast Downward found non-trivial κ-plans for numerous tasks here. In terms of performance, compiling the contingent $(\mathcal{F}, \kappa)$-reformulations to classical planning strictly dominated solving the former directly, in terms of the coverage of both solvable and unsolvable FT-planning tasks. In sum, the direction of compiling FT-planning tasks to classical planning appears promising, and clearly deserves further investigation.

	CFF	CFF$\left(\Pi^{(\mathcal{F},\kappa)}\right)$				FD$\left(\Pi'\right)$			
task		0	1	2	4	0	1	2	4
5×5 (0.1)	0.08	0.00	–	–	–	0.00	0.10	0.02	0.04
	0.08	0.00	–	–	–	0.00	0.31	–	0.13
	0.08	0.00	–	–	–	0.00	0.10	–	0.13
	0.08	0.00	–	–	–	0.00	0.00	0.01	0.03
	0.08	0.00	4.99	–	–	0.00	0.01	0.01	0.03
5×5 (0.2)	0.08	0.00	0.13	–	–	0.00	0.01	0.03	0.06
	0.08	0.00	–	–	–	0.00	0.19	0.02	0.04
	0.08	0.00	–	–	–	0.00	0.01	0.01	0.03
	0.08	0.00	–	–	–	0.00	0.00	0.01	0.03
	0.08	0.00	1.28	2.59	–	0.00	0.70	–	0.21
5×5 (0.5)	0.08	0.00	–	–	–	0.00	7.50	0.02	0.04
	0.08	0.00	0.55	0.59	0.79	0.00	0.01	0.18	106.53
	0.08	0.00	0.12	5.04	5.43	0.00	0.01	0.02	0.04
	0.08	0.00	–	–	–	0.00	0.01	310.59	–
	0.07	0.00	–	–	–	0.00	0.01	0.02	0.04
7×7 (0.1)	0.12	0.00	–	–	–	0.00	0.02	0.03	0.06
	0.13	0.00	2.10	–	–	0.00	1.67	0.04	0.07
	0.13	0.00	–	–	–	0.00	0.21	0.03	0.07
	0.13	0.00	–	–	–	0.00	0.02	0.03	0.06
	0.13	0.00	–	–	–	0.00	0.09	0.04	0.07
7×7 (0.2)	0.13	0.00	–	–	–	0.00	27.32	0.04	0.08
	0.13	0.00	–	–	–	0.00	0.01	0.03	0.06
	0.13	0.00	–	–	–	0.00	0.02	0.03	0.06
	0.13	0.00	–	–	–	0.00	0.01	0.03	0.06
	0.13	0.00	–	–	–	0.00	5.96	0.04	0.07
7×7 (0.5)	0.13	0.00	–	–	–	0.00	0.38	0.05	0.09
	0.13	0.00	3.32	4.13	–	0.00	0.04	0.63	11.56
	0.13	0.00	–	–	–	0.00	0.31	38.86	–
	0.13	0.00	0.14	0.15	0.15	0.00	0.01	0.03	0.06
	0.13	0.00	–	–	–	0.00	0.89	17.37	1.25

Table 2: Planner runtimes on different formulations of FT-planning tasks in the spirit of our example.

In Lemma 5 below, we now formalize the properties of FT-PLAN-1-κ that are exploited by the compilation of its fragment above. In particular, this lemma allows for extending this compilation scheme to arbitrary *fixed* bounds on the number of non-deterministic effects per operator, as well as to arbitrary normative 1-primary exception models.

Lemma 4 *Let $\langle \Pi = \langle P, O, s_0, G \rangle, \mathcal{F}, \kappa \rangle$ be an FT-planning task with a 1-primary model $\mathcal{F}$, and $\max_{o \in O} \mathsf{eff}(o) = b$. If π is an irreducible contingent plan for $\Pi^{(\mathcal{F},\kappa)}$, then there exists a set of policies $\pi_0, \ldots, \pi_m$ over $S^{(\mathcal{F},\kappa)}$ such that*

(1) π_0 is an empty policy, $\pi_m = \pi$, and each π_i extends π_{i-1} by prescribing an action for a single additional state of $\Pi^{(\mathcal{F},\kappa)}$ such that the execution tree $T_{\pi_{i-1}}$ is a proper sub-tree of T_{π_i}, and

(2) for $0 \le i \le m$ and $0 \le j \le \kappa$, π_i induces at most b executions ρ that do not achieve the goal within T_{π_i} and have $\mathcal{F}(\rho) = j$.

The proof is as follows. Let π be an irreducible contingent plan for the $(\mathcal{F}, \kappa)$-reformulation of $\langle \Pi, \mathcal{F}, \kappa \rangle$ as in the claim, and let $\{(s_1, k_1), \ldots, (s_m, k_m)\}$ be a relabeling of the nodes $S^{(\mathcal{F},\kappa)}_\pi$ consistently with the order in which they are expanded by a depth-first traversal of T_π, with the "depth" of a node (s, k) being given by k. Given that, let a sequence of policies $M = \pi_0, \ldots, \pi_m$ be defined as

$$\pi_i(s_j, k_j) = \begin{cases} \pi_(s_j, k_j), & j \leq i \\ \text{undefined}, & j > i \end{cases}.$$

It is immediate that M satisfies condition (1) of the lemma, and so what remains to be shown is condition (2). The proof is by induction on i. For $i = 0$, the condition is trivially satisfied since π_0 is empty. Assuming that the condition is satisfied for $i - 1 \geq 0$, the proof for i is as follows. By the DFS construction of M, we have $S^{(\mathcal{F},\kappa)}_{\pi_i} = S^{(\mathcal{F},\kappa)}_{\pi_{i-1}} \cup \{(s_i, k_i)\}$, where (s_i, k_i) is a non-goal leaf node in $T_{\pi_{i-1}}$. Furthermore, for all other non-goal leaf nodes (s_j, k_j) of $T_{\pi_{i-1}}$, it holds that $k_j \leq k_i$, or otherwise DFS would expand (s_j, k_j) prior to (s_i, k_i). Given that, consider the extension of $T_{\pi_{i-1}}$ to T_{π_i} by $\pi(s_i, k_i)$.

By the definition of exception models, for each $k < k_i$, the number of executions ρ of π_i that do not achieve the goal within T_{π_i} and have $\mathcal{F}(\rho) = k$ is the same as for π_{i-1}, and this because the number of "exceptions so far" cannot decrease with the progress of the execution. For $k = k_i$, since $\mathcal{F}$ is 1-primary, π_i replaces a single execution ρ of π_{i-1} that does not achieve the goal within $T_{\pi_{i-1}}$ and has $\mathcal{F}(\rho) = k_i$, with at most one such execution, namely the one that extends ρ with the sole primary effect of $\pi(s_i, k_i)$. Finally, for all $k > k_i$, there are no executions of π_{i-1} that do not achieve the goal within $T_{\pi_{i-1}}$ and have $\mathcal{F}(\rho) = k$, and thus there are at most b such executions of π_i.

Given an FT-planning task $\langle \Pi = \langle P, O, s_0, G\rangle, \mathcal{F}, \kappa\rangle$ with a normative 1-primary model $\mathcal{F}$ and $\max_{o \in O} \mathsf{eff}(o) = O(1)$, a polynomial-time, sound, and complete compilation of $\langle \Pi, \mathcal{F}, \kappa\rangle$ to a classical planning task $\Pi' = \langle P', O', \sigma'_0, G'\rangle$ is specified below. For ease of presentation, for each operator $o \in O(s)$, if $\mathsf{eff}(o) = \{e_0, e_1, \ldots e_{b'}\}$, then $\mathcal{F}(e_0) = 0$. The set of propositions P' contains $\kappa(b-1)+1$ replicas of propositions P, as well as $\kappa(b-1)+1$ auxiliary propositions *open*, denoted as

$$P' = P_{0,0} \cup \{\mathit{open}_{0,0}\} \cup \bigcup_{\substack{1 \leq i \leq \kappa,\\ 1 \leq j \leq b-1}} P_{i,j} \cup \{\mathit{open}_{i,j}\}.$$

The set of operators O' contains $\kappa(b - 1) + 1$ sets of operators $O_{0,0}, O_{1,1}, \ldots, O_{1,b-1}, \ldots, O_{\kappa,1}, \ldots, O_{\kappa,b-1}$, with $O_{i,j} = \{o_{i,j} \mid o \in O\}$. For each $o \in O$, the precondition of the operator $o_{i,j}$ is

$$\mathsf{pre}(o_{i,j}) = \lfloor \mathsf{pre}(o) \rfloor_{i,j} \cup \{\mathit{open}_{i,j}\} \cup \bigcup_{x=j+1}^{b-1} \{\neg \mathit{open}_{i,x}\}$$
$$\cup \bigcup_{y=i+1}^{\kappa} \bigcup_{x=1}^{b-1} \{\neg \mathit{open}_{y,x}\}.$$

If $\mathsf{eff}(o) = \{e_0, e_1, \ldots e_{b'}\}$, then

$$\mathsf{eff}_{i,j} = \{\langle \emptyset, \lfloor \mathsf{add}(e_0) \rfloor_{i,j}, \lfloor \mathsf{del}(e_0) \rfloor_{i,j} \rangle\} \;\cup$$
$$\bigcup_{\substack{1 \leq x \leq b',\\ i+\mathcal{F}(e_x) \leq \kappa}} \left\{ \begin{array}{ll} \langle \emptyset, \{\mathit{open}_{i+\mathcal{F}(e_x),x}\}, & \emptyset \rangle, \\ \langle \emptyset, \lfloor \mathsf{add}(e_1) \rfloor_{i+\mathcal{F}(e_x),x}, & \lfloor \mathsf{del}(e_1) \rfloor_{i+\mathcal{F}(e_x),x} \rangle \end{array} \right\}$$
$$\cup\; \Phi_{i,j,x}$$

where

$$\Phi_{i,j,x} = \bigcup_{p \notin \mathsf{add}(e_x) \cup \mathsf{del}(e_x)} \left\{ \begin{array}{ll} \langle \{p_{i,j}\}, \{p_{i+\mathcal{F}(e_x),x}\}, \emptyset & \rangle, \\ \langle \{\neg p_{i,j}\}, \emptyset, & \{p_{i+\mathcal{F}(e_x),x}\} \rangle \end{array} \right\}.$$

Likewise, O' contains a set of auxiliary operators $\{o^*_{0,0}, o^*_{1,1}, \ldots, o^*_{1,b-1}, \ldots, o^*_{\kappa,1}, \ldots, o^*_{\kappa,b-1}\}$, with

$$\mathsf{pre}(o^*_{i,j}) = \lfloor G \rfloor_{i,j} \cup \{\mathit{open}_{i,j}\} \cup \bigcup_{x=j+1}^{b-1} \{\neg \mathit{open}_{i,x}\}$$
$$\cup \bigcup_{y=i+1}^{\kappa} \bigcup_{x=1}^{b-1} \{\neg \mathit{open}_{y,x}\},$$
$$\mathsf{eff}(o^*_i) = \{\langle \emptyset, \emptyset, \{\mathit{open}_{i,j}\} \rangle\}.$$

Finally, as in the basic case, the initial state and goal are specified as $\sigma'_0 = \lfloor s_0 \rfloor_{0,0} \cup \{\mathit{open}_{0,0}\}$ and $G' = \{\neg \mathit{open}_{0,0}\}$.

Theorem 5 *Let $\langle \Pi =, \mathcal{F}, \kappa\rangle$ be an FT-planning task with a 1-primary model $\mathcal{F}$, and Π' be the compilation of Π. There is a bijective, efficiently computable mapping between the irreducible plans for $\Pi^{(\mathcal{F},\kappa)}$ and those for Π'.*

Summary

We studied computational properties of fault tolerant planning, a simple and natural planning formalism that interpolates between contingent and classical planning. We showed that an important spot along this interpolation exhibits attractive worst-case time complexity, and for most, can be efficiently compiled to classical planning in a sound, complete, and practicable manner.

The palette of possible (and impossible) extensions to FT-planning that call for investigation is wide. For instance, it is possibly more natural in some contexts to assume bounds on the number of failures per operator, and not on the number of failures overall, as we do here. It is easy to show that this type of assumption-based planning is also in PSPACE, but our more practicable results here (NP-membership for problems with polynomially-long executions and the specific compilation scheme to classical planning) do not apply there. Also, for simplicity, throughout the paper we assumed a single (aka fully known a priori) initial state. It is not hard to verify, however, that our PSPACE-membership, NP-membership, and compilation results can be straightforwardly extended to arbitrary, polynomial, and fixed numbers of possible initial states, respectively, as long as listing these possible initial states does not introduce further complexity. Finally, we believe that FT-planning to classical planning compilation can be substantially stratified by exploiting the structure of the FT-planning tasks, similarly to the way the structure of the tasks is exploited in recent compilations from conformant to classical planning (Palacios and Geffner 2009).

Acknowledgments. This work was partially supported by the EOARD grant FA8655-12-1-2096, and the ISF grant 1045/12.

References

Albore, A., and Bertoli, P. 2004. Generating safe assumption-based plans for partially observable, nondeterministic domains. In *AAAI*, 495–500.

Albore, A., and Bertoli, P. 2006. Safe LTL assumption-based planning. In *ICAPS*, 193–202.

Albore, A., and Geffner, H. 2009. Acting in partially observable environments when achievement of the goal cannot be guaranteed. In *ICAPS Workshop on Planning and Plan Execution for Real-World Systems*.

Albore, A.; Palacios, H.; and Geffner, H. 2009. A translation-based approach to contingent planning. In *IJCAI*, 1623–1628.

Bonet, B., and Geffner, H. 2011. Planning under partial observability by classical replanning: Theory and experiments. In *IJCAI*, 1936–1941.

Bonet, B. 2010. Conformant plans and beyond: Principles and complexity. *AIJ* 174(3-4):245–269.

Brafman, R. I., and Shani, G. 2012a. A multi-path compilation approach to contingent planning. In *AAAI*.

Brafman, R. I., and Shani, G. 2012b. Replanning in domains with partial information and sensing actions. *JAIR* 45:565–600.

Bylander, T. 1994. The computational complexity of propositional STRIPS planning. *AIJ* 69(1-2):165–204.

Cimatti, A.; Pistore, M.; Roveri, M.; and Traverso, P. 2003. Weak, strong, and strong cyclic planning via symbolic model checking. *AIJ* 147(1-2):35–84.

Davis-Mendelow, S.; Baier, J. A.; and McIlraith, S. A. 2012. Making reasonable assumptions to plan with incomplete information. In *ICAPS Workshop on Planning and Plan Execution for Real-World Systems*.

Domshlak, C.; Even-Zur, Z.; Golany, Y.; Karpas, E.; and Nus, Y. 2011. Command and control training centers: Computer generated forces meet classical planning. In *ICAPS Workshop on Scheduling and Planning Applications (SPARK)*.

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

Georgeff, M., and Lansky, A. L. 1986. Procedural knowledge. *Proceedings of IEEE* 74(10):1383–1398.

Giunchiglia, F.; Spalazzi, L.; and Traverso, P. 1994. Planning with failure. In *AIPS*, 74–79.

Göbelbecker, M.; Gretton, C.; and Dearden, R. 2011. A switching planner for combined task and observation planning. In *AAAI*.

Helmert, M. 2006. The Fast Downward planning system. *JAIR* 26:191–246.

Hoffmann, J., and Brafman, R. I. 2005. Contingent planning via heuristic forward search with implicit belief states. In *ICAPS*, 71–80.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *JAIR* 14:253–302.

Jensen, R. M.; Veloso, M. M.; and Bryant, R. E. 2004. Fault tolerant planning: Toward probabilistic uncertainty models in symbolic non-deterministic planning. In *ICAPS*, 335–344.

Kissmann, P., and Edelkamp, S. 2012. Improving cost-optimal domain-independent symbolic planning. In *AAAI*.

Littman, M. 1997. Probabilistic propositional planning: representations and complexity. In *AAAI*, 748754.

Palacios, H., and Geffner, H. 2009. Compiling uncertainty away in conformant planning problems with bounded width. *JAIR* 35:623–675.

Pistore, M., and Vardi, M. Y. 2007. The planning spectrum - One, two, three, infinity. *JAIR* 30:101–132.

Puterman, M. 1994. *Markov Decision Processes*. Wiley.

Rintanen, J.; Heljanko, K.; and Niemelä, I. 2006. Planning as satisfiability: parallel plans and algorithms for plan search. *AIJ* 170(12-13):1031–1080.

Rintanen, J. 2004. Complexity of planning with partial observability. In *ICAPS*, 345–354.

Taig, R., and Brafman, R. I. 2013. Compiling conformant probabilitic planning into classical planning. In *ICAPS*.

Talamadupula, K.; Benton, J.; Schermerhorn, P. W.; Kambhampati, S.; and Scheutz, M. 2010. Integrating a closed world planner with an open world robot: A case study. In *AAAI*.

Williams, B. C.; Ingham, M.; Chung, S. H.; and Elliott, P. H. 2003. Procedural knowledge. *Proceedings of IEEE* 9:212237.

Yoon, S. W.; Fern, A.; Givan, R.; and Kambhampati, S. 2008. Probabilistic planning via determinization in hindsight. In *AAAI*, 1010–1016.