

Imperfect-Information Games and Generalized Planning

Giuseppe De Giacomo

SAPIENZA Università di Roma
Rome, Italy

degiamoco@dis.uniroma1.it

Aniello Murano, Sasha Rubin, Antonio Di Stasio

Università degli Studi di Napoli “Federico II”
Naples, Italy

first.last@unina.it

Abstract

We study a generalized form of planning under partial observability, in which we have multiple, possibly infinitely many, planning domains with the same actions and observations, and goals expressed over observations, which are possibly temporally extended. By building on work on two-player (non-probabilistic) games with imperfect information in the Formal Methods literature, we devise a general technique, generalizing the belief-state construction, to remove partial observability. This reduces the planning problem to a game of perfect information with a tight correspondence between plans and strategies. Then we instantiate the technique and solve some generalized-planning problems.

1 Introduction

Automated planning is a fundamental problem in Artificial Intelligence. Given a deterministic dynamic system with a single known initial state and a goal condition, automated planning consists of finding a sequences of actions (the plan) to be performed by agents in order to achieve the goal [M. Ghallab and Traverso, 2008]. The application of this notion in real-dynamic worlds is limited, in many situations, by three facts: i) the number of objects is neither small nor predetermined, ii) the agent is limited by its observations, iii) the agent wants to realize a goal that extends over time. For example, a preprogrammed driverless car cannot know in advance the number of obstacles it will enter in a road, or the positions of the other cars not in its view, though it wants to realize, among other goals, that every time it sees an obstacle it avoids it. This has inspired research in recent years on *generalized forms of planning* including conditional planning in partially observable domains [Levesque, 1996; Rintanen, 2004], planning with incomplete information for temporally extended goals [De Giacomo and Vardi, 1999; Bertoli and Pistore, 2004] and generalized planning for multiple domains or infinite domains [Levesque, 2005; Srivastava *et al.*, 2008; Bonet *et al.*, 2009; Hu and Levesque, 2010; Hu and De Giacomo, 2011; Srivastava *et al.*, 2011; Felli *et al.*, 2012; Srivastava *et al.*, 2015].

We use the following running example, taken from [Hu and Levesque, 2010], to illustrate a generalized form of planning:

Example 1 (Tree Chopping). The goal is to chop down a tree, and store the axe. The number of chops needed to fell the tree is unknown, but a look-action checks whether the tree is up or down. Intuitively, a plan solving this problem alternates looking and chopping until the tree is seen to be down, and then stores the axe. This scenario can be formalized as a partially-observable planning problem on a single infinite domain (Example 2, Figure 1), or on the disjoint union of infinitely many finite domains (Section 3).

The standard approach to solve planning under partial observability for *finite* domains is to reduce them to planning under complete observability. This is done by using the *belief-state construction* that removes partial observability and passes to the *belief-space* [Goldman and Boddy, 1996; Bertoli *et al.*, 2006]. The motivating problem of this work is to solve planning problems on infinite domains, and thus we are naturally lead to the problem of removing partial-observability from infinite domains.

In this paper we adopt the Formal Methods point of view and consider generalized planning as a game \mathbf{G} of imperfect information, i.e., where the player under control has partial observability. The game \mathbf{G} may be infinite, i.e., have infinitely many states.

Our technical contribution (Theorem 4.5) is a general sound and complete mathematical technique for removing imperfect information from a possibly infinite game \mathbf{G} to get a game \mathbf{G}^β , possibly infinite, of perfect information. Our method builds on the classic belief-state construction [Reif, 1984; Goldman and Boddy, 1996; Raskin *et al.*, 2007], also adopted in POMDPs [Kaelbling *et al.*, 1998; LaValle, 2006].¹ The classic belief-state construction fails for certain infinite games, see Example 3. We introduce a new component to the classic belief-state construction that isolates only those plays in the belief-space that correspond to plays in \mathbf{G} . This new component is necessary and sufficient to solve the general case and capture all infinite games \mathbf{G} .

We apply our technique to the decision problem that asks, given a game of imperfect information, if the player under control has a winning strategy (this corresponds to deciding if there is a plan for a given planning instance). We remark that we consider strategies and plans that may de-

¹However, our work considers nondeterminism rather than probability, and qualitative objectives rather than quantitative objectives.

pend on the history of the observations, not just the last observation. Besides showing how to solve the running Tree Chopping example, we report two cases. The first case is planning under partial observability for temporally extended goals expressed in LTL in finite domains (or a finite set of infinite domains sharing the same observations). This case generalizes well-known results in the AI literature [De Giacomo and Vardi, 1999; Rintanen, 2004; Bertoli *et al.*, 2006; Hu and De Giacomo, 2011; Felli *et al.*, 2012]. The second case involves infinite domains. Note that because game solving is undecidable for computable infinite games (simply code the configuration space of a Turing Machine), solving games with infinite domains requires further computability assumptions. We focus on games generated by pushdown automata; these are infinite games that recently attracted the interest of the AI community [Murano and Perelli, 2015; Chen *et al.*, 2016]. In particular these games have been solved assuming perfect information. By applying our technique, we extend their results to deal with imperfect information under the assumption that the stack remains observable (it is known that making the stack unobservable leads to undecidability [Azhar *et al.*, 2001]).

2 Generalized-Planning Games

In this section we define generalized-planning (GP) games, known as games of imperfect information in the Formal Methods literature [Raskin *et al.*, 2007], that capture many generalized forms of planning.

Informally, two players (agent and environment) play on a transition-system. Play proceeds in rounds. In each round, from the current state s of the transition-system, the agent observes $obs(s)$ (some information about the current state), and picks an action a from the set of actions Ac , and then the environment picks an element of $tr(s, a)$ (tr is the transition function of the transition-system) to become the new current state. Note that the players are asymmetric, i.e., the agent picks actions and the environment resolves non-determinism.

Notation. Write X^ω for the set of infinite sequences whose elements are from the set X , write X^* for the finite sequences, and X^+ for the finite non-empty sequences. If π is a finite sequence then $Last(\pi)$ denotes its last element. The positive integers are denoted \mathbb{N} , and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.

Linear-temporal logic. We define LTL over a finite set of letters Σ .² The *formulas of LTL (over Σ)* are generated by the following grammar: $\varphi ::= x \mid \varphi \wedge \varphi \mid \neg\varphi \mid X\varphi \mid \varphi \cup \varphi$ where $x \in \Sigma$. We introduce the usual abbreviations for, e.g., \vee, F . Formulas of LTL (over Σ) are interpreted over infinite words $\alpha \in \Sigma^\omega$. Define the satisfaction relation \models as follows: i) $(\alpha, n) \models x$ iff $\alpha_n = x$; ii) $(\alpha, n) \models \varphi_1 \wedge \varphi_2$ iff $(\alpha, n) \models \varphi_i$ for $i = 1, 2$; iii) $(\alpha, n) \models \neg\varphi$ iff it is not the case that $(\alpha, n) \models \varphi$; iv) $(\alpha, n) \models X\varphi$ iff $(\alpha, n+1) \models \varphi$; v) $(\alpha, n) \models \varphi_1 \cup \varphi_2$ iff there exists $i \geq n$ such that $(\alpha, i) \models \varphi_2$ and for all $j \in [n, i)$, $(\alpha, j) \models \varphi_1$. Write $\alpha \models \varphi$ if $(\alpha, 0) \models \varphi$ and say that α *satisfies* the LTL formula φ .

²This is without loss of generality, since if LTL were defined over a set of atomic propositions AP we let $\Sigma = 2^{AP}$ and replace atoms $p \in AP$ by $\bigvee_{p \in x} x$ to get equivalent LTL formulas over Σ .

Arenas. An *arena of imperfect information*, or simply an *arena*, is a tuple $\mathbf{A} = (S, I, Ac, tr, Obs, obs)$, where S is a (possibly infinite) set of *states*, $I \subseteq S$ is the set of *initial states*, Ac is a finite set of *actions*, and $tr : S \times Ac \rightarrow 2^S \setminus \{\emptyset\}$ is the *transition function*, Obs is a (possibly infinite) set of *observations*, and $obs : S \rightarrow Obs$, the *observation function*, maps each state to an observation. We extend tr to sets of states: for $\emptyset \neq Q \subseteq S$, let $tr(Q, a)$ denote the set $\bigcup_{q \in Q} tr(q, a)$.

Sets of the form $obs^{-1}(x)$ for $x \in Obs$ are called *observation sets*. The set of all observation sets is denoted $ObsSet$. Non-empty subsets of observation sets are called *belief-states*. Informally, a belief-state is a subset of the states of the game that the play could be in after a given finite sequence of observations and actions.

Finite and finitely-branching. An arena is *finite* if S is finite, and *infinite* otherwise. An arena is *finitely-branching* if i) I is finite, and ii) for every s, a the cardinality of $tr(s, a)$ is finite. Clearly, being finite implies being finitely-branching.

Strategies. A *play* in \mathbf{A} is an infinite sequence $\pi = s_0 a_0 s_1 a_1 s_2 a_2 \dots$ such that $s_0 \in I$ and for all $i \in \mathbb{N}_0$, $s_{i+1} \in tr(s_i, a_i)$. A *history* $h = s_0 a_0 \dots s_{n-1} a_{n-1} s_n$ is a finite prefix of a play ending in a state. The set of plays is denoted $Ply(\mathbf{A})$, and the set of histories is denoted $Hist(\mathbf{A})$ (we drop \mathbf{A} when it is clear from the context). For a history or play $\pi = s_0 a_0 s_1 a_1 \dots$ write $obs(\pi)$ for the sequence $obs(s_0) a_0 obs(s_1) a_1 \dots$. A *strategy* (for the agent) is a function $\sigma : Hist(\mathbf{A}) \rightarrow Ac$. A strategy is *observational* if $obs(h) = obs(h')$ implies $\sigma(h) = \sigma(h')$. In Section 3 we will briefly mention an alternative (but essentially equivalent) definition of observational strategy, i.e., as a function $Obs^+ \rightarrow Ac$. We do not define strategies for the environment. A play $\pi = s_0 a_0 s_1 a_1 \dots$ is *consistent* with a strategy σ if for all $i \in \mathbb{N}$ we have that if $h \in Hist(\mathbf{A})$ is a prefix of π , say $h = s_0 a_0 \dots s_{n-1} a_{n-1} s_n$, then $\sigma(h) = a_{n+1}$.

GP Games. A *generalized-planning (GP) game*, is a tuple $\mathbf{G} = \langle \mathbf{A}, W \rangle$ where the *winning objective* $W \subseteq Obs^\omega$ is a set of infinite sequences of observation sets. A *GP game with restriction* is a tuple $\mathbf{G} = \langle \mathbf{A}, W, F \rangle$ where, in addition, $F \subseteq S^\omega$ is the *restriction*. Note that unlike the winning objective, the restriction need not be closed under observations. A GP game is *finite* (resp. *finitely branching*) if the arena \mathbf{A} is finite (resp. finitely branching).

Winning. A strategy σ is *winning* in $\mathbf{G} = \langle \mathbf{A}, W \rangle$ if for every play $\pi \in Ply(\mathbf{A})$ consistent with σ , we have that $obs(\pi) \in W$. Similarly, a strategy is winning in $\mathbf{G} = \langle \mathbf{A}, W, F \rangle$ if for every play $\pi \in Ply(\mathbf{A})$ consistent with σ , if $\pi \in F$ then $obs(\pi) \in W$. Note that a strategy is winning in $\langle \mathbf{A}, W, Ply(\mathbf{A}) \rangle$ if and only if it is winning in $\langle \mathbf{A}, W \rangle$.

Solving a GP game. A central decision problem is the following, called *solving a GP game*: given a (finite representation of a) GP game of imperfect information \mathbf{G} , decide if the agent has a winning observational-strategy.

GP games of perfect information. An arena/GP game has *perfect information* if $Obs = S$ and $obs(s) = s$ for all s . We thus suppress mentioning Obs and obs completely, e.g., we write $\mathbf{A} = (S, I, Ac, tr)$ and $W, F \subseteq S^\omega$. Note that in a GP game of perfect information every strategy is observational.

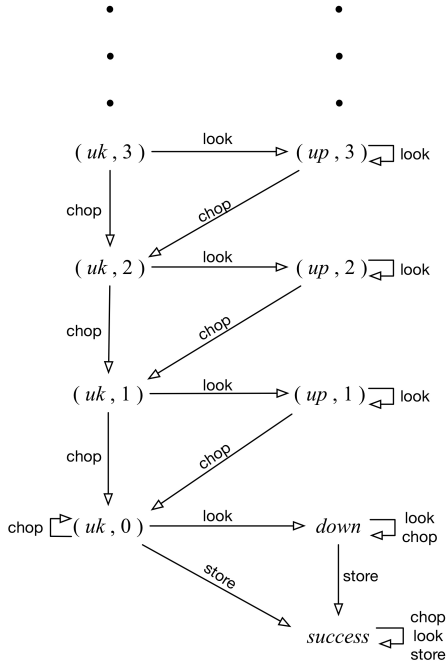


Figure 1: Part of the arena \mathbf{A}_{chop} (missing edges go to the *failure* state). The numbers correspond to the number of chops required to fell the tree. The arena is not finitely-branching since it has infinitely many initial states $\{uk\} \times \mathbb{N}_0$.

Example 2 (continued). We formalize the tree-chopping planning problem. Define the GP game $\mathbf{G}_{\text{chop}} = \langle \mathbf{A}_{\text{chop}}, W \rangle$ where $\mathbf{A}_{\text{chop}} = \langle S, I, Ac, tr, Obs, obs \rangle$, and:

- $S = \{down, success, failure\} \cup (\{uk\} \times \mathbb{N}_0) \cup (\{up\} \times \mathbb{N})$,
- $Ac = \{\text{chop}, \text{look}, \text{store}\}$, $I = \{uk\} \times \mathbb{N}$,
- tr is illustrated in Figure 1,
- $Obs = \{\text{DN}, \checkmark, \times, \text{UK}, \text{UP}\}$,
- obs maps $down \mapsto \text{DN}$, $(up, i) \mapsto \text{UP}$ for $i \in \mathbb{N}$, $(uk, i) \mapsto \text{UK}$ for $i \in \mathbb{N}_0$, $failure \mapsto \times$, and $success \mapsto \checkmark$, and
- the objective W is defined as $\alpha \in W$ iff $\alpha \models F \checkmark$.

The mentioned plan is formalized as the observational-strategy σ_{chop} that maps any history ending in (uk, i) to look (for $i \in \mathbb{N}_0$), (up, i) to chop (for $i \in \mathbb{N}$), $down$ to store , and all others arbitrarily (say to store).

Note: σ_{chop} is a winning strategy, i.e., no matter which initial state the environment chooses, the strategy ensures that the play (it is unique because the rest of the GP game is deterministic) reaches the state *success* having observation \checkmark .

3 Generalized-Planning Games and Generalized Forms of Planning

In this section we establish that generalized-planning (GP) games can model many different types of planning from the

AI literature, including a variety of generalized forms of planning:

1. planning on finite transition-systems, deterministic actions, actions with conditional effects, partially observable states, incomplete information on the initial state, and temporally extended goals [De Giacomo and Vardi, 1999];
2. planning under partial observability with finitely many state variables, nondeterministic actions, reachability goals, and partial observability [Rintanen, 2004];
3. planning on finite transition systems, nondeterministic actions, looking for strong plans (i.e., adversarial non-determinism) [Bertoli *et al.*, 2006];
4. generalized planning, consisting of multiple (possibly infinitely many) related finite planning instances [Hu and Levesque, 2010; Hu and De Giacomo, 2011].

We discuss the latter in detail. Following [Hu and De Giacomo, 2011], a *generalized-planning problem* \mathfrak{P} is defined as a sequence of related classical planning problems. In our terminology, fix finite sets Ac, Obs and let \mathfrak{P} be a countable sequence $\mathbf{G}_1, \mathbf{G}_2, \dots$ where each \mathbf{G}_n is a finite GP game of the form $\langle S_n, \{t_n\}, Ac, tr_n, Obs, obs_n, W_n \rangle$. In [Hu and De Giacomo, 2011], a plan is an observational-strategy $p : Obs^+ \rightarrow Ac$, and a solution is a single plan that solves all of the GP games in the sequence. Now, we view \mathfrak{P} as a single infinite GP game as follows. Let $\mathbf{G}_{\mathfrak{P}}$ denote the *disjoint* union of the GP games in \mathfrak{P} . Formally, $\mathbf{G}_{\mathfrak{P}} = \langle S, I, Ac, tr, Obs, obs, W \rangle$ where

- $S = \{(s, n) : s \in S_n, n \in \mathbb{N}\}$,
- $I = \{(t_n, n) : n \in \mathbb{N}\}$,
- $tr((s, n), a) = \{(t, n) : t \in tr_n(s, a)\}$,
- $obs(s, n) = obs_n(s)$,
- $W = \cup_n W_n$.

Then: there is a correspondence between solutions for \mathfrak{P} and winning observational-strategies in $\mathbf{G}_{\mathfrak{P}}$.

For example, consider the tree-chopping problem as formalized in [Hu and Levesque, 2010; Hu and De Giacomo, 2011]: there are infinitely many planning instances which are identical except for an integer parameter denoting the number of chops required to fell the tree. The objective for all instances is to fell the tree. Using the translation above we get a GP game with an infinite arena which resembles (and, in fact, can be transformed to) the GP game in Example 2.

4 Generalized Belief-State Construction

In this section we show how to remove imperfect information from generalized-planning (GP) games \mathbf{G} . That is, we give a transformation of GP games of imperfect information \mathbf{G} to GP games of perfect information \mathbf{G}^β such that the agent has a winning observational-strategy in \mathbf{G} if and only if the agent has a winning strategy in \mathbf{G}^β . The translation is based on the classic belief-state construction [Reif, 1984; Raskin *et al.*, 2007]. Thus, we begin with a recap of that construction.

Belief-state Arena.³ Let $\mathbf{A} = (S, I, Ac, tr, Obs, obs)$ be an arena (not necessarily finite). Recall from Section 2 that *observation sets* are of the form $obs^{-1}(x)$ for $x \in Obs$, and are collectively denoted $ObsSet$. Define the arena of perfect information $\mathbf{A}^\beta = (S^\beta, I^\beta, Ac, tr^\beta)$ where,

- S^β is the set of belief-states, i.e., the non-empty subsets of the observation-sets,
- I^β consists of all belief-states of the form $I \cap X$ for $X \in ObsSet$,
- $tr^\beta(Q, a)$ consists of all belief-states of the form $tr(Q, a) \cap X$ for $X \in ObsSet$.

The idea is that $Q \in S^\beta$ represents a refinement of the observation set: the agent, knowing the structure of \mathbf{G} ahead of time, and the sequence of observations so far in the game, may deduce that it is in fact in a state from Q which may be a strict subset of its corresponding observation set X .⁴

NB. Since \mathbf{A}^β is an arena, we can talk of its histories and plays. Although we defined S^β to be the set of all belief-states, only those belief-states that are reachable from I^β are relevant. Thus, overload notation and write S^β for the set of reachable belief-states, and \mathbf{A}^β for the corresponding arena. This notation has practical relevance since if \mathbf{A} is countable there are uncountably many belief-states; but in many cases only countably many (or, as in the running example, finitely many) reachable belief-states.

The intuition for the rest of this section is illustrated in the next example.

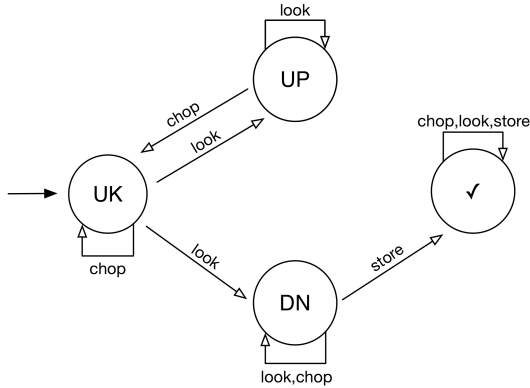


Figure 2: Part of the arena \mathbf{A}_{chop}^β (missing edges go the *failure* state). Each circle is a belief-state. The winning objective is $F\checkmark$, and the restriction is $\neg GF[UK \wedge XX \text{ look} \wedge XX \text{ UP}]$.

Example 3 (continued). Figure 2 shows the arena \mathbf{A}_{chop}^β corresponding to the arena from tree-chopping game \mathbf{G}_{chop} , i.e.,

³In the AI literature, this is sometimes called the *belief-space*.

⁴To illustrate simply, suppose there is a unique initial state s , and that it is observationally equivalent to other states. At the beginning of the game the agent can deduce that it must be in s . Thus, its initial belief-state is $\{s\}$ and not its observation-set $obs^{-1}(obs(s))$. This belief can (and, in general, must) be exploited if the agent is to win.

- S^β are the following belief-states: $\{(uk, n) \mid n \in \mathbb{N}_0\}$, denoted UK; $\{(up, n) \mid n \in \mathbb{N}\}$, denoted UP; $\{down\}$, denoted DN; $\{success\}$, denoted \checkmark ; and $\{failure\}$.
- I^β is the belief-state UK,
- and tr^β is shown in the figure.

Note that the agent does not have a winning strategy in the GP game with arena \mathbf{A}_{chop}^β and winning condition $F\checkmark$. The informal reason is that the strategy σ_{chop} (which codifies “alternately look and chop until the tree is sensed to be down, and then store the axe”), which wins in \mathbf{G} , does not work. The reason is that after every look the opponent can transition to UP (and never DN), resulting in the play $\rho = (UK \text{ look } UP \text{ chop})^\omega$, i.e., the repetition of (UK look UP chop) forever. Such a play of \mathbf{A}_{chop}^β does not correspond to any play in \mathbf{A}_{chop} . This is a well known phenomena of the standard belief-set construction [Sardiña *et al.*, 2006], which our construction overcomes by adding a restriction that removes from consideration plays such as ρ (as discussed in Example 5).

The following definition is central. It maps a history $h \in Hist(\mathbf{A})$ to the corresponding history $h^\beta \in Hist(\mathbf{A}^\beta)$ of belief-states.

Definition 4.1. For $h \in Hist(\mathbf{A})$ define $h^\beta \in Hist(\mathbf{A}^\beta)$ inductively as follows.

- For $s \in I$, define $s^\beta \in I^\beta$ to be $I \cap obs^{-1}(obs(s))$. In words, s^β is the set of initial states the GP game could be in given the observation $obs(s)$.
- If $h \in Hist(\mathbf{A})$, $a \in Ac$, $s \in S$, then $(has)^\beta := h^\beta a B$ where $B := tr(Last(h^\beta), a) \cap obs^{-1}(obs(s))$. In words, B is the set of possible states the GP game could be in given the observation sequence $obs(has)$.

In the same way, for $\pi \in Ply(\mathbf{A})$ define $\pi^\beta \in Ply(\mathbf{A}^\beta)$. Extend the map pointwise to sets of plays $P \subseteq Ply(\mathbf{A})$, i.e., define $P^\beta := \{\pi^\beta \in Ply(\mathbf{A}^\beta) \mid \pi \in P\}$. Finally, we give notation to the special case that $P = Ply(\mathbf{A})$: write $lm(\mathbf{A})$ for the set $\{\pi^\beta \mid \pi \in Ply(\mathbf{A})\}$, called the *image of A*.

By definition, $lm(\mathbf{A}) \subseteq Ply(\mathbf{A}^\beta)$. However, the converse is not always true.

Example 4 (continued). There is a play of \mathbf{A}_{chop}^β that is not in $lm(\mathbf{A}_{chop})$, e.g., $\rho = (uk \text{ look } up \text{ chop})^\omega$. Indeed, suppose $\pi^\beta = \rho$ and consider the sequence of counter values of π . Every look action establishes that the current counter value in π is positive (this is the meaning of the tree being *up*), but every chop action reduces the current counter value by one. This contradicts that counter values are always non-negative.

Remark 4.2. If \mathbf{A} is finitely-branching then $lm(\mathbf{A}) = Ply(\mathbf{A}^\beta)$. To see this, let ρ be a play in \mathbf{A}^β , and consider the forest whose nodes are the histories h of \mathbf{A} such that h^β is a prefix of ρ . Each tree in the forest is finitely branching (because \mathbf{A} is), and at least one tree in this forest is infinite. Thus, by König’s lemma, the tree has an infinite path π . But π is a play in \mathbf{A} and $\pi^\beta = \rho$.

Definition 4.3. For $\rho \in Ply(\mathbf{A}^\beta)$, say $\rho = B_0 a_0 B_1 a_1 \dots$, define $obs(\rho)$ to be the sequence $obs(q_0) a_0 obs(q_1) a_1 \dots$

where $q_i \in B_i$ for $i \in \mathbb{N}_0$ (this is well defined since, by definition of the state set S^β , each B_i is a subset of a unique observation-set).

The classic belief-state construction transforms $\langle \mathbf{A}, W \rangle$ into $\langle \mathbf{A}^\beta, W \rangle$. Example 3 shows that this transformation may not preserve the agent having a winning strategy if \mathbf{A} is infinite. We now define the generalized belief-state construction and the main technical theorem of this work.

Definition 4.4. Let $\mathbf{G} = \langle \mathbf{A}, W \rangle$ be a GP game. Define $\mathbf{G}^\beta = \langle \mathbf{A}^\beta, W, \text{Im}(\mathbf{A}) \rangle$, a GP game of perfect information with restriction. The restriction $\text{Im}(\mathbf{A}) \subseteq \text{Ply}(\mathbf{A}^\beta)$ is the image of $\text{Ply}(\mathbf{A})$ under the map $\pi \mapsto \pi^\beta$.

Theorem 4.5. Let \mathbf{A} be a (possibly infinite) arena of imperfect information, \mathbf{A}^β the corresponding belief-state arena of perfect information, and $\text{Im}(\mathbf{A}) \subseteq \text{Ply}(\mathbf{A}^\beta)$ the image of \mathbf{A} . Then, for every winning objective W , the agent has a winning observational-strategy in the GP game $\mathbf{G} = \langle \mathbf{A}, W \rangle$ if and only if the agent has a winning strategy in the GP game $\mathbf{G}^\beta = \langle \mathbf{A}^\beta, W, \text{Im}(\mathbf{A}) \rangle$. Moreover, if \mathbf{A} is finitely-branching then $\mathbf{G}^\beta = \langle \mathbf{A}^\beta, W \rangle$.⁵

Proof. The second statement follows from the first statement and Remark 4.2. For the first statement, we first need some facts that immediately follow from Definition 4.1.

1. $(h_1)^\beta = (h_2)^\beta$ if and only if $\text{obs}(h_1) = \text{obs}(h_2)$.
2. For every $h \in \text{Hist}(\mathbf{A}^\beta)$ that is also a prefix of π^β there is a history $h' \in \text{Hist}(\mathbf{A})$ that is also a prefix of π such that $(h')^\beta = h$. Also, for every $h' \in \text{Hist}(\mathbf{A})$ that is also a prefix of π there is a history $h \in \text{Hist}(\mathbf{A}^\beta)$ that is also a prefix of π^β such that $(h')^\beta = h$.

Second, there is a natural correspondence between observational strategies of \mathbf{A} and strategies of \mathbf{A}^β .

- If σ is a strategy in \mathbf{A}^β then define the strategy $\omega(\sigma)$ of \mathbf{A} as mapping $h \in \text{Hist}(\mathbf{A})$ to $\sigma(h^\beta)$. Now, $\omega(\sigma)$ is observational by Fact 1. Also, if π is consistent with $\omega(\sigma)$ then π^β is consistent with σ . Indeed, let h be a history that is also a prefix of π^β . We need to show that $h\sigma(h)$ is a prefix of π^β . Suppose that $\sigma(h) = a$. Take prefix h' of π such that $(h')^\beta = h$ (Fact 2). Then $\omega(\sigma)(h') = \sigma((h')^\beta) = \sigma(h) = a$. Since π is assumed consistent with $\omega(\sigma)$, conclude that $h'a$ is a prefix of π . Thus ha is a prefix of π^β .
- If σ is an observational strategy in \mathbf{A} then define the strategy $\kappa(\sigma)$ of \mathbf{A}^β as mapping $h \in \text{Hist}(\mathbf{A}^\beta)$ to $\sigma(h')$ where h' is any history such that $(h')^\beta = h$. This is well-defined by (†) and the fact that σ is observational. Also, if ρ is consistent with $\kappa(\sigma)$, then every π with $\pi^\beta = \rho$ (if there are any) is consistent with σ . Indeed, let h' be a history of π and take a prefix h of π^β such that $(h')^\beta = h$ (Fact 2). Then $\kappa(\sigma)(h) = \sigma(h')$, call this action a . But π^β is assumed consistent with $\kappa(\sigma)$, and thus ha is a prefix of π^β . Thus $h'a$ is a prefix of π .

We now put everything together and show that the agent has a winning observational-strategy in \mathbf{G} iff the agent has a winning strategy in \mathbf{G}^β .

⁵The case that \mathbf{A} is finite appears in [Raskin *et al.*, 2007].

Suppose σ is a winning strategy in \mathbf{G}^β . Let $\pi \in \text{Ply}(\mathbf{A})$ be consistent with the observational strategy $\omega(\sigma)$ of \mathbf{G} . Then $\pi^\beta \in \text{Im}(\mathbf{A})$ is consistent with σ . But σ is assumed to be winning, thus $\text{obs}(\pi^\beta) \in W$. But $\text{obs}(\pi) = \text{obs}(\pi^\beta)$. Conclude that $\omega(\sigma)$ is a winning strategy in \mathbf{G} .

Conversely, suppose σ is a winning strategy in \mathbf{G} . Let $\rho \in \text{Im}(\mathbf{A})$ be consistent with the strategy $\kappa(\sigma)$ of \mathbf{G}^β , and take $\pi \in \text{Ply}(\mathbf{A})$ be such that $\pi^\beta = \rho$ (such a π exists since we assumed $\rho \in \text{Im}(\mathbf{A})$). Then π is consistent with σ . But σ is assumed to be winning, thus $\text{obs}(\pi) \in W$. But $\text{obs}(\rho) = \text{obs}(\pi^\beta) = \text{obs}(\pi)$. Conclude that $\kappa(\sigma)$ is a winning strategy in \mathbf{G}^β . \square

Remark 4.6. The proof of Theorem 4.5 actually shows how to transform strategies between the GP games, i.e., $\sigma \mapsto \omega(\sigma)$ and $\sigma \mapsto \kappa(\sigma)$, and moreover, these transformations are inverses of each other.

We end with our running example:

Example 5 (Continued). Solving \mathbf{G}_{chop} (Figure 1) is equivalent to solving the finite GP game $\mathbf{G}_{\text{chop}}^\beta$ of perfect information, i.e., $\langle \mathbf{A}_{\text{chop}}^\beta, W, \text{Im}(\mathbf{A}_{\text{chop}}) \rangle$, where the arena \mathbf{A} is shown in Figure 2. To solve this we should understand the structure of $\text{Im}(\mathbf{A}_{\text{chop}})$. It is not hard to see that a play $\rho \in \text{Ply}(\mathbf{A}_{\text{chop}}^\beta)$ is in $\text{Im}(\mathbf{A}_{\text{chop}})$ if and only if it contains only finitely many infixes of the form “UK look UP”. This property is expressible in LTL by the formula $\neg \text{GF}[\text{UK} \wedge \text{X look} \wedge \text{XX UP}]$. Thus we can apply the algorithm for solving finite games of perfect information with LTL objectives (see, e.g., [Pnueli and Rosner, 1989; de Alfaro *et al.*, 2001]) to solve $\mathbf{G}_{\text{chop}}^\beta$, and thus the original GP game \mathbf{G}_{chop} .

5 Application of the Construction

We now show how to use generalized-planning (GP) games and our generalized belief-state construction to obtain effective planning procedures for sophisticated problems. For the rest of this section we assume Obs is finite (\mathbf{A} may be infinite) so that we can consider LTL temporally extended goals over the alphabet Obs . For instance, LTL formulas specify persistent surveillance missions such as “get items from region A, drop items at region B, infinitely often, and always avoid region C”.

Definition 5.1. Let φ be an LTL formula over $\text{Obs} \times \text{Ac}$. For an arena \mathbf{A} , define $[[\varphi]] = \{\pi \in \text{Ply}(\mathbf{A}) \mid \text{obs}(\pi) \models \varphi\}$.

The following is immediate from Theorem 4.5 and the fact that solving finite LTL games of perfect information is decidable [Pnueli and Rosner, 1989; de Alfaro *et al.*, 2001]:

Theorem 5.2. Let $\mathbf{G} = \langle \mathbf{A}, [[\varphi]] \rangle$ be a GP game with a finite arena (possibly obtained as the disjoint union of several arenas sharing the same observations), and φ be an LTL winning objective. Then solving \mathbf{G} can be reduced to solving the finite GP game $\mathbf{G}^\beta = \langle \mathbf{A}^\beta, [[\varphi]] \rangle$ of perfect information, which is decidable.

Although we defined winning objectives to be observable, one may prefer general winning conditions, i.e., $W \subseteq S^\omega$. In this case, for finite arenas there is a translation from parity-objectives to observable parity-objectives [Chatterjee

and Doyen, 2010]; moreover, for reachability objectives, a plan reaches a goal $T \subseteq S$ iff it reaches a belief-state $B \subseteq T$ [Bertoli *et al.*, 2006].

Next we look a case where the arena is actually infinite. Recently, the AI community has considered games generated by pushdown-automata [Murano and Perelli, 2015; Chen *et al.*, 2016]. However, the games considered are of perfect information and cannot express generalized-planning problems or planning under partial observability. In contrast, our techniques can solve these planning problems on pushdown domains assuming that the stack is not hidden (we remark that if the stack is hidden, then game-solving becomes undecidable [Azhar *et al.*, 2001]):

Theorem 5.3. *Let $\mathbf{G} = \langle \mathbf{A}, [[\varphi]] \rangle$ be a GP game with a pushdown-arena with observable stack, and φ is an LTL formula. Then solving \mathbf{G} can be reduced to solving $\mathbf{G}^\beta = \langle \mathbf{A}^\beta, [[\varphi]] \rangle$, a GP game with pushdown-arena with perfect information, which is decidable.*

Proof. Let P be a pushdown-automaton with states Q , initial state q_0 , finite input alphabet Ac , and finite stack alphabet Γ . We call elements of Γ^* stacks, and denote the empty stack by ϵ . Also, fix an observation function on the states, i.e., $f : Q \rightarrow \text{Obs}$ for some set Obs (we do not introduce notation for the transition function of P). A pushdown-arena $\mathbf{A}_P = (S, I, \text{Ac}, \text{tr}, \text{Obs}, \text{obs})$ is generated by P as follows: the set of states S is the set of configurations of P , i.e., pairs (q, γ) where $q \in Q$ and $\gamma \in \Gamma^*$ is a stack-content of P ; the initial state of \mathbf{A} is the initial configuration, i.e., $I = \{(q_0, \epsilon)\}$; the transition function of \mathbf{A} is defined as $\text{tr}((q, \gamma), a) = (q', \gamma')$ if P can move in one step from state q and stack content γ to state q' and stack content γ' by consuming the input letter a ; the observation function obs maps a configuration (q, γ) to $f(q)$ (i.e., this formalizes the statement that the stack is observable). Observe now that: (1) the GP game \mathbf{A} is finitely-branching; (2) the GP game \mathbf{A}^β is generated by a pushdown automaton (its states are subsets of Q). Thus we can apply Theorem 4.5 to reduce solving \mathbf{A} , a GP-game with imperfect information and pushdown arena, to \mathbf{A}^β , a GP-game with perfect information and pushdown arena. The latter is decidable [Walukiewicz, 2001]. \square

6 Related work in Formal Methods

Games of imperfect information on *finite* arenas have been studied extensively. Reachability winning-objectives were studied in [Reif, 1984] from a complexity point of view: certain games were shown to be universal in the sense that they are the hardest games of imperfect information, and optimal decision procedures were given. More generally, ω -regular winning-objectives were studied in [Raskin *et al.*, 2007], and symbolic algorithms were given (also for the case of randomized strategies).

To solve (imperfect-information) games on infinite arenas one needs a finite-representation of the infinite arena. One canonical way to generate infinite arenas is by parametric means. In this line, [Jacobs and Bloem, 2014] study the synthesis problem for distributed architectures with a parametric number of finite-state components. They leverage re-

sults from the Formal Methods literature that say that for certain types of token-passing systems there is a cutoff [Emerson and Namjoshi, 1995], i.e., an upper bound on the number of components one needs to consider in order to synthesize a protocol for any number of components. Another way to generate infinite arenas is as configuration spaces of pushdown automata. These are important in analysis of software because they capture the flow of procedure calls and returns in reactive programs. Module-checking pushdown systems of imperfect information [Bozzelli *et al.*, 2010; Aminof *et al.*, 2013] can be thought of as games in which the environment plays non-deterministic strategies. Although undecidable, by not hiding the stack (cf. Theorem 5.3) decidability of module-checking is regained.

Finally, we note that synthesis of distributed systems has been studied in the Formal Methods literature using the techniques of games, starting with [Pnueli and Rosner, 1990]. Such problems can be cast as multi-player games of imperfect information, and logics such as ATL with knowledge can be used to reason about strategies in these games. However, even for three players, finite arenas, and reachability goals, the synthesis problem (and the corresponding model checking problem for ATLK) is undecidable [Dima and Tiplea, 2011].

7 Critical Evaluation and Conclusions

Although our technique for removing partial observability is sound and complete, it is, necessarily, not algorithmic: indeed, no algorithm can always remove partial observability from computable infinite domains and result in a solvable planning problem (e.g. one with a finite domain).⁶

The main avenue for future technical work is to establish natural classes of generalized-planning problems that can be solved algorithmically. We believe the methodology of this paper will be central to this endeavor. Indeed, as we showed in Section 5, we can identify $\text{Im}(\mathbf{A})$ in a number of cases. We conjecture that one can do the same for all of the one-dimensional planning problems of [Hu and Levesque, 2010; Hu and De Giacomo, 2011].

The framework presented in this paper is non-probabilistic, but extending it with probabilities and utilities associated to agent choices [Kaelbling *et al.*, 1998; LaValle, 2006; Bonet and Geffner, 2009; Geffner and Bonet, 2013] is of great interest. In particular, POMDPs with temporally-extended winning objectives (e.g., LTL, Büchi, parity) have been studied for finite domains [Chatterjee *et al.*, 2010]. We leave for future work the problem of dealing with such POMDPs over infinite domains.

Acknowledgments

We thank the reviewers for their constructive comments. This research was partially supported by the Sapienza project “Immersive Cognitive Environments”. Aniello Murano was supported in part by GNCS 2016 project: Logica, Automi e Giochi per Sistemi Auto-adattivi. Sasha Rubin is a Marie Curie fellow of the Istituto Nazionale di Alta Matematica.

⁶In fact, there is no algorithm solving games of perfect observation on computable domains with reachability objectives.

References

- [Aminof *et al.*, 2013] B. Aminof, A. Legay, A. Murano, O. Serre, and M. Y. Vardi. Pushdown module checking with imperfect information. *Inf. Comput.*, 223, 2013.
- [Azhar *et al.*, 2001] S. Azhar, G. Peterson, and J. Reif. Lower bounds for multiplayer non-cooperative games of incomplete information. *J. Comp. Math. Appl.*, 41, 2001.
- [Bertoli and Pistore, 2004] P. Bertoli and M. Pistore. Planning with extended goals and partial observability. In *Proc. of ICAPS 2004*, 2004.
- [Bertoli *et al.*, 2006] P. Bertoli, A. Cimatti, M. Roveri, and P. Traverso. Strong planning under partial observability. *Artif. Intell.*, 170(4-5), 2006.
- [Bonet and Geffner, 2009] B. Bonet and H. Geffner. Solving POMDPs: Rtdp-bel vs. point-based algorithms. In *Proc. of IJCAI 2009*, 2009.
- [Bonet *et al.*, 2009] B. Bonet, H. Palacios, and H. Geffner. Automatic derivation of memoryless policies and finite-state controllers using classical planners. In *Proc. of ICAPS 2009*, 2009.
- [Bozzelli *et al.*, 2010] L. Bozzelli, A. Murano, and A. Peron. Pushdown module checking. *FMSD*, 36(1), 2010.
- [Chatterjee and Doyen, 2010] K. Chatterjee and L. Doyen. The complexity of partial-observation parity games. In *Proc. of LPAR 2010*, 2010.
- [Chatterjee *et al.*, 2010] K. Chatterjee, L. Doyen, and T.A. Henzinger. Qualitative analysis of partially-observable markov decision processes. In *Proc. of MFCS*, 2010.
- [Chen *et al.*, 2016] T. Chen, F. Song, and Z. Wu. Global model checking on pushdown multi-agent systems. In *Proc. of AAI 2016*, 2016.
- [de Alfaro *et al.*, 2001] L. de Alfaro, T. A. Henzinger, and R. Majumdar. From verification to control: Dynamic programs for omega-regular objectives. In *Proc. of LICS 2001*, 2001.
- [De Giacomo and Vardi, 1999] G. De Giacomo and M. Y. Vardi. Automata-theoretic approach to planning for temporally extended goals. In *Proc. of ECP 1999*, 1999.
- [Dima and Tiplea, 2011] C. Dima and F. L. Tiplea. Model-checking ATL under imperfect information and perfect recall semantics is undecidable. *CoR*, abs/1102.4225, 2011.
- [Emerson and Namjoshi, 1995] E. A. Emerson and K. S. Namjoshi. Reasoning about rings. In *Proc. of POPL 1995*, 1995.
- [Felli *et al.*, 2012] P. Felli, G. De Giacomo, and A. Lomuscio. Synthesizing agent protocols from LTL specifications against multiple partially-observable environments. In *Proc. of KR 2012*, 2012.
- [Geffner and Bonet, 2013] H. Geffner and B. Bonet. *A Concise Introduction to Models and Methods for Automated Planning*. Morgan & Claypool, 2013.
- [Goldman and Boddy, 1996] R. P. Goldman and M. S. Boddy. Expressive planning and explicit knowledge. In *Proc. of AIPS 1996*, 1996.
- [Hu and De Giacomo, 2011] Y. Hu and G. De Giacomo. Generalized planning: Synthesizing plans that work for multiple environments. In *Proc. of IJCAI 2011*, 2011.
- [Hu and Levesque, 2010] Y. Hu and H. J. Levesque. A correctness result for reasoning about one-dimensional planning problems. In *Proc. of KR 2010*, 2010.
- [Jacobs and Bloem, 2014] S. Jacobs and R. Bloem. Parameterized synthesis. *LMCS*, 10(1), 2014.
- [Kaelbling *et al.*, 1998] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1-2), 1998.
- [LaValle, 2006] S. M. LaValle. *Planning algorithms*. Cambridge, 2006.
- [Levesque, 1996] H. J. Levesque. What is planning in the presence of sensing. In *Proc. of AAI 1996*, 1996.
- [Levesque, 2005] H. Levesque. Planning with loops. In *Proc. of IJCAI 2005*, 2005.
- [M. Ghallab and Traverso, 2008] D. Nau M. Ghallab and P. Traverso. *Automated Planning: Theory & Practice*. Elsevier, 2008.
- [Murano and Perelli, 2015] A. Murano and G. Perelli. Pushdown multi-agent system verification. In *Proc. of IJCAI 2015*, 2015.
- [Pnueli and Rosner, 1989] A. Pnueli and R. Rosner. On the synthesis of an asynchronous reactive module. In *Proc. of ICALP 1989*, 1989.
- [Pnueli and Rosner, 1990] A. Pnueli and R. Rosner. Distributed reactive systems are hard to synthesize. In *Proc. of STOC 1990*, 1990.
- [Raskin *et al.*, 2007] J. Raskin, K. Chatterjee, L. Doyen, and T. A. Henzinger. Algorithms for omega-regular games with imperfect information. *LMCS*, 3(3), 2007.
- [Reif, 1984] J. H. Reif. The complexity of two-player games of incomplete information. *JCSS*, 29(2), 1984.
- [Rintanen, 2004] J. Rintanen. Complexity of planning with partial observability. In *Proc. of ICAPS 2004*, 2004.
- [Sardiña *et al.*, 2006] S. Sardiña, G. De Giacomo, Y. Lespérance, and H. J. Levesque. On the limits of planning over belief states under strict uncertainty. In *Proc. of KR 2016*, 2006.
- [Srivastava *et al.*, 2008] S. Srivastava, N. Immerman, and S. Zilberstein. Learning generalized plans using abstract counting. In *Proc. of AAI 2008*, 2008.
- [Srivastava *et al.*, 2011] S. Srivastava, N. Immerman, and S. Zilberstein. A new representation and associated algorithms for generalized planning. *Artif. Intell.*, 175(2), 2011.
- [Srivastava *et al.*, 2015] S. Srivastava, S. Zilberstein, A. Gupta, P. Abbeel, and S. J. Russell. Tractability of planning with loops. In *Proc. of AAI 2015*, 2015.
- [Walukiewicz, 2001] I. Walukiewicz. Pushdown processes: Games and model-checking. *Inf. Comput.*, 164(2), 2001.