Game-Theoretic Reasoning about Actions in Nonmonotonic Causal Theories

Alberto Finzi[1,2] and Thomas Lukasiewicz[1,2]

[1] Institut für Informationssysteme, Technische Universität Wien
Favoritenstraße 9-11, A-1040 Vienna, Austria
{finzi,lukasiewicz}@kr.tuwien.ac.at

[2] Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza"
Via Salaria 113, I-00198 Rome, Italy
{finzi,lukasiewicz}@dis.uniroma1.it

Abstract. We present the action language $GC+$ for reasoning about actions in multi-agent systems under probabilistic uncertainty and partial observability, which is an extension of the action language $\mathcal{C}+$ that is inspired by partially observable stochastic games (POSGs). We provide a finite-horizon value iteration for this framework and show that it characterizes finite-horizon Nash equilibria. We also describe how the framework can be implemented on top of nonmonotonic causal theories. We then present acyclic action descriptions in $GC+$ as a special case where transitions are computable in polynomial time. We also give an example that shows the usefulness of our approach in practice.

1 Introduction

There are several important problems that we have to face in reasoning about actions for mobile agents in real-world environments. First and foremost, we have to deal with uncertainty, both about the initial situation of the agent's world and about the results of the actions taken by the agent (due to noisy effectors and/or sensors). Second, a closely related problem is that the properties of real-world environments are in general not fully observable (due to noisy and inaccurate sensors, or because some relevant parts of the environment simply cannot be sensed), and thus we also have to deal with partial observability. One way of adding uncertainty and partial observability to reasoning about actions is based on qualitative models in which all possible alternatives are equally taken into consideration. Another way is based on quantitative models where we have a probability distribution on the set of possible alternatives, and thus can numerically distinguish between the possible alternatives.

Well-known first-order formalisms for reasoning about actions such as the situation calculus [24] easily allow for expressing qualitative uncertainty about the effects of actions and the initial situation of the world through disjunctive knowledge. Furthermore, there are generalizations of the action language $\mathcal{A}$ [12] that allow for qualitative uncertainty in the form of nondeterministic actions. An important recent formalism in this family is the action language $\mathcal{C}+$ [13], which is based on the theory of nonmonotonic causal reasoning presented in [18], and has evolved from the action language $\mathcal{C}$. In addition to allowing for conditional and nondeterministic effects of actions, $\mathcal{C}+$ also supports concurrent actions as well as indirect effects and preconditions of actions through static causal laws. Closely related to it is the recent planning language $\mathcal{K}$ [7].

There are a number of formalisms for probabilistic reasoning about actions. In particular, Bacchus et al. [1] propose a probabilistic generalization of the situation calculus, which is based on first-order logics of probability, and which allows to reason about an agent's probabilistic degrees of belief and how these beliefs change when actions are executed. Poole's independent choice logic [22] is based on acyclic logic programs under different "choices". Each choice along with the acyclic logic program produces a first-order model. By placing a probability distribution over the different choices, we then obtain a distribution over the set of first-order models. Boutilier et al. [5] introduce and explore an approach to first-order (fully observable) Markov decision processes (MDPs) [23] that are formulated in a probabilistic generalization of the situation calculus. A companion paper [6] presents a generalization of Golog, called DTGolog, that combines agent programming in Golog with decision-theoretic planning in MDPs. Probabilistic extensions of the action language $\mathcal{A}$ and its most recent variant $\mathcal{C}+$ have especially been proposed by Baral et al. [2] and Eiter and Lukasiewicz [8].

Many of the above logical formalisms for reasoning about actions under probabilistic uncertainty take inspiration from decision-theoretic planning in fully observable Markov decision processes (MDPs) [23] and the more general partially observable Markov decision processes (POMDPs) [16]. Such logical formalisms for reasoning about actions that are inspired by decision-theoretic planning are also appealing from the perspective of decision-theoretic planning, since they allow for [11,14] (i) compactly representing MDPs and POMDPs without explicitly referring to atomic states and state transitions, (ii) exploiting such compact representations for efficiently solving large-scale problems, and (iii) nice properties such as *modularity* (parts of the specification can be easily added, removed, or modified) and *elaboration tolerance* (solutions can be easily reused for similar problems with few or no additional effort).

The above generalizations of $\mathcal{A}$ and $\mathcal{C}+$ in [2,8] assume that the model of the world consists of a single agent that we want to control and the environment summarized in "nature". In realistic applications, however, we often encounter multiple agents, which may compete or cooperate with each other. Here, the optimal actions of one agent generally depend on the actions of all the other agents. In particular, there is a bidirectional dependence between the actions of two agents, which generally makes it inappropriate to model enemies and friends of the controlled agent simply as a part of "nature".

There are generalizations of MDPs and POMDPs to multi-agent systems with cooperative agents, called multi-agent MDPs [4] and decentralized POMDPs [3,20], respectively. Similarly, there are also generalizations of MDPs and POMDPs to multi-agent systems with competing (that is, not necessarily cooperative) agents, called stochastic games [21] (or Markov games [25,17]) and partially observable stochastic games (POSGs) [15,9], respectively. Multi-agent MDPs (resp., decentralized POMDPs) and stochastic games (resp., POSGs) are similar to MDPs (resp., POMDPs), except that actions (and decisions) are distributed among multiple agents, where the optimal actions of each agent may depend on the actions of all the other agents. Stochastic games (resp., POSGs) generalize both normal form games [26] and MDPs (resp., POMDPs).

In this paper, we present the language $G\mathcal{C}+$ for reasoning about actions in multi-agent systems under probabilistic uncertainty and partial observability, which is an extension of the language $\mathcal{C}+$ that takes inspirations from partially observable stochastic games (POSGs) [15]. The main contributions of this paper are as follows:

- We present the action language $G\mathcal{C}+$ for reasoning about actions in multi-agent systems under probabilistic uncertainty and partial observability, which is an extension of both the action language $\mathcal{C}+$ and POSGs. We consider the very general case in which the agents may have different rewards, and thus may be competitive. Here, we assume that planning and control are centralized as follows. All agents transmit their local belief states and/or observations to a central agent, which then computes and returns the optimal local action for each agent.
- Under the above assumption, the high worst-case complexity of POSGs (NEXP-completeness for the special case of decentralized POMDPs [3]) is avoided, since the POSG semantics of $G\mathcal{C}+$ can be translated into a belief state stochastic game semantics. We use the latter to define a finite-horizon value iteration for $G\mathcal{C}+$, and show that it characterizes finite-horizon Nash equilibria.
- We show that the $G\mathcal{C}+$ framework can be implemented on top of reasoning in nonmonotonic causal theories. We present acyclic action descriptions in $G\mathcal{C}+$ as a special case where transitions are computable in polynomial time. We also provide an example that shows the usefulness of our approach in practice.

Note that further technical details are given in the extended paper [10].

2 Preliminaries

In this section, we recall the basic concepts of the action language $\mathcal{C}+$, normal form games, and partially observable stochastic games.

2.1 The Action Language $\mathcal{C}+$

We first recall the main concepts of the action language $\mathcal{C}+$; see especially [13] for further details, motivation, and background.

Syntax. Properties of the world are represented by rigid variables, simple fluents, and statically determined fluents, while actions are expressed by action variables. The values of rigid variables do not change when actions are performed, while the ones of simple (resp., statically determined) fluents may directly (resp., indirectly) change through actions. The knowledge about the latter is encoded through dynamic (resp., static) causal laws over formulas, which are Boolean combinations of atomic assignments.

Formally, we thus assume a finite set V of *variables*, which are divided into *rigid variables*, *simple fluents*, *statically determined fluents*, and *action variables*. Every variable $X \in V$ may take on *values* from a nonempty finite *domain* $D(X)$, where every action variable has the Boolean domain $\{\bot, \top\}$. We define *formulas* inductively as follows. *False* and *true*, denoted $\bot$ and $\top$, respectively, are formulas. If $X \in V$ and $x \in D(X)$, then $X = x$ is a formula (called *atom*). If ϕ and ψ are formulas, then also $\neg\phi$ and $(\phi \wedge \psi)$. A *literal* is an atom $X = x$ or a negated atom $\neg X = x$ (abbreviated as $X \neq x$). We often abbreviate $X = \top$ (resp., $X = \bot$) as X (resp., $\neg X$).

Static causal laws express static knowledge about fluents and rigid variables. They are expressions of the form

$$\textbf{caused}\ \psi\ \textbf{if}\ \phi\,, \tag{1}$$

where ψ and ϕ are formulas such that either (a) every variable in ψ is a fluent, and no variable in ϕ is an action variable, or (b) every variable in ψ and ϕ is rigid. Informally, (1) encodes that every state of the world that satisfies ϕ should also satisfy ψ. If $\phi = \top$, then (1) is abbreviated by **caused** ψ. *Dynamic causal laws* express how simple fluents change when actions are performed. They have the form

$$\textbf{caused}\ \psi\ \textbf{if}\ \phi\ \textbf{after}\ \theta\,, \tag{2}$$

where ψ, ϕ, and θ are formulas such that every variable in ψ is a simple fluent, and no variable in ϕ is an action variable. Informally, (2) encodes that every next state of the world satisfying ϕ should also satisfy ψ, if the current state and the executed action satisfy θ. If $\phi = \top$, then (2) is abbreviated by **caused** ϕ **after** θ. If also $\theta = a_1 \wedge \cdots \wedge a_k \wedge \delta$, where every a_i is an assignment of $\top$ to an action variable, then (2) is abbreviated by $a_1, \ldots, a_k$ **causes** ψ **if** δ. Informally, if the current state of the world satisfies δ, then the next state after concurrently executing $a_1, \ldots, a_k$ satisfies ψ. If $\psi = \bot$ and $\phi = \top$, then (2) is an *execution denial* and abbreviated by

$$\textbf{nonexecutable}\ \theta\,. \tag{3}$$

Informally, if a state s and an action α satisfy θ, then α is not executable in s. If $\theta = a_1 \wedge \cdots \wedge a_k \wedge \delta$, then (3) is abbreviated by **nonexecutable** $a_1, \ldots, a_k$ **if** δ. Informally, $a_1, \ldots, a_k$ cannot be concurrently executed in a state satisfying δ. The expression **inertial** X, where $X \in V$, abbreviates the set of all laws (2) such that $\phi = \psi = \theta = X{=}x$ and $x \in D(X)$. Informally, the value of X remains unchanged when actions are executed, as long as this does not produce any inconsistencies.

A *causal law* (or *axiom*) is a static or dynamic causal law. An *action description* D is a finite set of causal laws. An *initial database* ϕ is a formula without action variables.

Semantics. An action description D represents a system of transitions from states to sets of possible successor states, while an initial database ϕ encodes a set of possible initial states. We now define states and actions, the executability of actions in states, and the above transitions through actions.

An *interpretation* I of a set of variables $V' \subseteq V$ assigns to every $X \in V'$ an element of $D(X)$. We say I *satisfies* an atom $Y = y$, where $Y \in V'$, denoted $I \models Y = y$, iff $I(Y) = y$. Satisfaction is extended to all formulas over V' as usual.

Let s be an interpretation of all rigid variables and fluents in V. Let D^s be the set of all ψ such that either (a) $s \models \phi$ for some **caused** ψ **if** ϕ in D, or (b) $s \models \psi$ and $\psi = X{=}x$ for some simple fluent $X \in \mathcal{X}$ and $x \in D(X)$. A *state* s of D is an interpretation s as above that is a unique model of D^s. An *action* α is an interpretation of all action variables in V. The action α is *executable* in a state s, denoted $Poss(\alpha, s)$, iff $s \cup \alpha$ satisfies $\neg\theta$ for every **nonexecutable** θ in D.

An *action transition* is a triple (s, α, s'), where s and s' are states of D such that $s(X) = s'(X)$ for every rigid variable $X \in V$, and α is an action that is executable in s. A formula ψ is *caused* in (s, α, s') iff either (a) $s' \models \phi$ for some **caused** ψ **if** ϕ in D, or (b) $s \cup \alpha \models \theta$ and $s' \models \phi$ for some **caused** ψ **if** ϕ **after** θ in D. The triple (s, α, s') is *causally explained* iff s' is the only interpretation that satisfies all formulas caused in (s, α, s'). For every state s and action α, define $\Phi(s, \alpha)$ as the set of all states s' such that (s, α, s') is *causally explained*. Note that $\Phi(s, \alpha) = \emptyset$ if no such (s, α, s') exists, in particular, if α is not executable in s. We say that D is *consistent* iff $\Phi(s, \alpha) \neq \emptyset$ for all actions α and states s such that α is executable in s. Informally, $\Phi(s, \alpha)$ is the set of all possible successor states after executing α in s.

2.2 Normal Form Games

Normal form games from classical game theory [26] describe the possible actions of $n \geqslant 2$ agents and the rewards (or utilities) that the agents receive when they simultaneously execute one action each. For example, in *two-finger Morra*, two players E and O simultaneously show one or two fingers. Let f be the total numbers of fingers shown. If f is odd, then O gets f dollars from E, and if f is even, then E gets f dollars from O. Formally, a *normal form game* $G = (I, (A_i)_{i \in I}, (R_i)_{i \in I})$ consists of a set of *agents* $I = \{1, \ldots, n\}$, $n \geqslant 2$, a nonempty finite set of *actions* A_i for each agent $i \in I$, and a *reward* (or *utility*) *function* $R_i \colon A \to \mathbf{R}$ for each agent $i \in I$, which associates with every *joint action* $a \in A = \times_{i \in I} A_i$ a *reward* (or *utility*) $R_i(a)$ to agent i.

A pure (resp., mixed) strategy specifies which action an agent should execute (resp., which actions an agent should execute with which probability). Formally, a *pure strategy* for agent $i \in I$ is any action $a_i \in A_i$. A *pure strategy profile* is any joint action $a \in A$. If the agents play a, then the *reward* to agent $i \in I$ is $R_i(a)$. A *mixed strategy* for agent $i \in I$ is any probability distribution π_i over A_i. A *mixed strategy profile* $\pi = (\pi_i)_{i \in I}$ consists of a mixed strategy π_i for each agent $i \in I$. If the agents play π, then the *expected reward* to agent $i \in I$, denoted $\mathbf{E}[R_i(a) \mid \pi]$ (or $R_i(\pi)$), is defined as

$$\textstyle\sum_{a=(a_j)_{j \in I} \in A} R_i(a) \cdot \mathit{\Pi}_{j \in I} \pi_j(a_j) \,.$$

We are especially interested in mixed strategy profiles π, called Nash equilibria, where no agent has the incentive to deviate from its part, once the other agents play their parts. A mixed strategy profile $\pi = (\pi_i)_{i \in I}$ is a *Nash equilibrium* for G iff for every agent $i \in I$, it holds that $R_i(\pi'_i \circ \pi_{-i}) \leqslant R_i(\pi)$ for every mixed strategy π'_i, where π_{-i} (resp., $\pi'_i \circ \pi_{-i}$) is obtained from π by removing π_i (resp., replacing π_i by π'_i). Every normal form game G has at least one Nash equilibrium among its mixed (but not necessarily pure) strategy profiles, and many have multiple Nash equilibria. A *Nash selection function* f associates with every normal form game G a unique Nash equilibrium $f(G)$. The expected reward to agent $i \in I$ under $f(G)$ is denoted by $v^i_f(G)$.

2.3 Partially Observable Stochastic Games

We will use POSGs [15] to define the semantics of the action language $G\mathcal{C}+$, where we assume that planning and control are centralized as follows. There exists a central agent, which (i) knows the local belief state of every other agent, (ii) computes and sends them their optimal local actions, and (iii) thereafter receives their local observations. Hence, we assume a transmission of local belief states and local observations to a central agent from all other agents, and of the optimal local actions in the reverse direction. Using this assumption, we can translate POSGs into belief state stochastic games, and then perform a finite-horizon value iteration.

Roughly, a POSG consists of a nonempty finite set of states S, a normal form game for each state $s \in S$, a set of joint observations of the agents O, and a transition function that associates with every state $s \in S$ and joint action of the agents $a \in A$ a probability distribution on all combinations of next states $s' \in S$ and joint observations $o \in O$. Formally, a *partially observable stochastic game (POSG)* $G = (I, S, (A_i)_{i \in I}, (O_i)_{i \in I}, P, (R_i)_{i \in I})$ consists of a set of *agents* $I = \{1, \ldots, n\}$, $n \geqslant 2$, a nonempty finite set of *states* S, two nonempty finite sets of *actions* A_i and *observations* O_i for each agent $i \in I$, a transition function $P\colon S \times A \to PD(S \times O)$, which associates with every state $s \in S$ and joint action $a \in A = \times_{i \in I} A_i$ a probability distribution over $S \times O$, where $O = \times_{i \in I} O_i$, and a *reward function* $R_i\colon S \times A \to \mathbf{R}$ for each agent $i \in I$, which associates with every state $s \in S$ and joint action $a \in A$ a *reward* $R_i(s, a)$ to agent i.

Since the actual state $s \in S$ of the POSG G is not fully observable, every agent $i \in I$ has a belief state b_i that associates with every state $s \in S$ the belief of agent i about s being the actual state. Formally, a *belief state* $b = (b_i)_{i \in I}$ of G consists of a probability function b_i over S for each agent $i \in I$. The POSG G then defines probabilistic transitions between belief states as follows. The new belief state $b^{a,o} = (b_i^{a,o})_{i \in I}$ after executing the joint action $a \in A$ in $b = (b_i)_{i \in I}$ and jointly observing $o \in O$ is given by:

$$b_i^{a,o}(s') = \textstyle\sum_{s \in S} P(s', o \mid s, a) \cdot b_i(s) \;/\; P_b(b_i^{a,o} \mid b_i, a), \text{ where}$$
$$P_b(b_i^{a,o} \mid b_i, a) = \textstyle\sum_{s' \in S} \sum_{s \in S} P(s', o \mid s, a) \cdot b_i(s)$$

is the probability of observing o after executing a in b_i. These probabilistic transitions define the fully observable stochastic game over belief states $G' = (I, B, (A_i)_{i \in I}, P_b, (R_i)_{i \in I})$, where B is the set of all belief states of G.

We next define finite-horizon pure and mixed policies and their rewards and expected rewards, respectively, using the above fully observable stochastic game over belief states. Assuming a finite horizon $H \geqslant 0$, a pure (resp., mixed) time-dependent policy associates with every belief state b of G and number of steps to go $h \in \{0, \ldots, H\}$ a pure (resp., mixed) normal form game strategy. Formally, a *pure policy* α assigns to each belief state b and number of steps to go $h \in \{0, \ldots, H\}$ a joint action from A. A *mixed policy* is of the form $\pi = (\pi_i)_{i \in I}$, where every π_i assigns to each belief state b and number of steps to go $h \in \{0, \ldots, H\}$ a probability function $\pi_i[b, h]$ over A_i. The H-step reward (resp., expected H-step reward) for pure (resp., mixed) policies can now be defined as usual. In particular, the *expected H-step reward* to agent $i \in I$ under a start belief state $b = (b_i)_{i \in I}$ and the mixed policy π, denoted $G_i(H, b, \pi)$, is defined as

$$\begin{cases} \sum_{a \in A} (\mathit{\Pi}_{j \in I} \pi_j[b, 0](a_j)) \cdot \sum_{s \in S} b_i(s) R_i(s, a) & \text{if } H = 0; \\ \sum_{a \in A} (\mathit{\Pi}_{j \in I} \pi_j[b, H](a_j)) \cdot (\sum_{s \in S} b_i(s) R_i(s, a) + & \\ \qquad \sum_{o \in O} P(b_i^{a,o} | b_i, a) \cdot G_i(H{-}1, b^{a,o}, \pi)) & \text{otherwise.} \end{cases}$$

The notion of a finite-horizon Nash equilibrium for a POSG G is then defined as follows. A policy π is a *Nash equilibrium* of G under a belief state b iff for every agent $i \in I$, it holds that $G_i(H, b, \pi'_i \circ \pi_{-i}) \leqslant G_i(H, b, \pi_i \circ \pi_{-i})$ for all policies π'_i. A policy π is a *Nash equilibrium* of G iff it is a Nash equilibrium of G under every belief state b.

Nash equilibria of G can be characterized by finite-horizon value iteration from local Nash equilibria of normal form games as follows. Let f be an arbitrary Nash selection function for normal form games with the action sets $(A_i)_{i\in I}$. For every belief state $b{=}(b_i)_{i\in I}$ and number of steps to go $h \in \{0,\ldots,H\}$, let $G[b,h] = (I,(A_i)_{i\in I}, (Q_i[b,h])_{i\in I})$, where $Q_i[b,h](a)$ is defined as follows (for all $a\in A$ and $i\in I$):

$$\begin{cases} \sum_{s\in S} b_i(s)R_i(s,a) & \text{if } h=0; \\ \sum_{s\in S} b_i(s)R_i(s,a) + \sum_{o\in O} P(b_i^{a,o}\,|b_i,a)\cdot v_f^i(G[b^{a,o},h{-}1]) & \text{otherwise.} \end{cases}$$

Let the mixed policy $\pi=(\pi_i)_{i\in I}$ for the POSG G be defined by $\pi_i(b,h)=f_i(G[b,h])$ for all $i\in I$, belief states b, and number of steps to go $h\in\{0,\ldots,H\}$. Then, π is a Nash equilibrium of G, and $G_i(H,b,\pi)=v_f^i(G[b,H])$ for every $i\in I$ and belief state b.

3 The Action Language G$\mathcal{C}+$

In this section, we define the action language G$\mathcal{C}+$, which generalizes both the action language $\mathcal{C}+$ and POSGs.

Syntax. We extend $\mathcal{C}+$ by formulas that express probabilistic transitions and agent rewards as in POSGs as well as formulas that encode the initial belief state of the agents.

We assume a set of $n\geqslant 2$ agents $I=\{1,\ldots,n\}$. Each agent $i\in I$ has (i) a nonempty set of action variables AV_i, where $AV_1,\ldots,AV_n$ partitions the set of all action variables $AV\subseteq V$, and (ii) a nonempty set of possible observations O_i. Every $o\in O = \times_{i\in I}O_i$ is a *joint observation.* A *probabilistic dynamic causal law* is of the form

$$\textbf{caused}\ [(\psi_1 \textbf{ if } \phi_1; o_1)\colon p_1,\ldots,(\psi_k \textbf{ if } \phi_k; o_k)\colon p_k] \textbf{ after } \delta\,, \tag{4}$$

where every **caused** ψ_j **if** ϕ_j **after** δ with $j\in\{1,\ldots,k\}$ is a dynamic causal law, every o_j is a joint observation, $p_1,\ldots,\, p_k>0$, $p_1{+}\cdots{+}p_k=1$, and $k\geqslant 1$. Informally, if an action α is executed in a state s, where $s\cup\alpha\models\delta$, then with the probability p_j the successor states satisfy **caused** ψ_j **if** ϕ_j and the agents observe o_j. We omit "**if** ϕ_j" in (4), when $\phi_j=\top$. A *reward law* for agent $i\in I$ is of the form

$$\textbf{reward } i\colon r \textbf{ after } \delta\,, \tag{5}$$

where r is a real. Informally, if an action α is executed in a state s, where $s\cup\alpha\models\delta$, then agent i receives the reward r. A *probabilistic initial database law* for $i\in I$ is of form

$$i\colon [\psi_1\colon p_1,\ldots,\psi_k\colon p_k]\,, \tag{6}$$

where each ψ_j with $j\in\{1,\ldots,k\}$ is a formula without action variables, $p_1,\ldots,p_k>0$, $p_1{+}\cdots{+}p_k=1$, and $k\geqslant 1$. Informally, the initial belief of agent i is that the set of states satisfying ψ_j holds with the probability p_j.

A *probabilistic action description* P is a finite set of causal, probabilistic dynamic causal, and reward laws. A *probabilistic initial database* $\Psi=(\Psi_i)_{i\in I}$ consists of a probabilistic initial database law Ψ_i for every agent $i\in I$.

Semantics. A probabilistic action description P represents a transition system, where every state s and action α executable in s is associated with a reward to every agent and a probability distribution over possible successor states. A probabilistic initial database $\Psi=(\Psi_i)_{i\in I}$ encodes each agent's probabilistic belief about the possible initial states.

The set of all states and actions of P and the executability of an action in a state are defined as in Section 2.1. An *action for agent* $i \in I$ is any interpretation over AV_i. The set of all actions for agent i is denoted by A_i. We next define the probabilistic transitions and the rewards encoded in P.

Let s be a state, and let α be an action executable in s. Suppose that P contains exactly one law F of the form (4) such that $s \cup \alpha \models \delta$. For every $j \in \{1, \ldots, k\}$, let P_j be obtained from P by replacing F by **caused** ψ_j **if** ϕ_j **after** δ. Let $\Phi_j(s, \alpha)$ be the set of all states s' such that (s, α, s') is causally explained relative to P_j. For each state s' and $o \in O$, let $P_j(s', o|s, \alpha) = p_j \,/\, |\Phi_j(s, \alpha)|$, if $s' \in \Phi_j(s, \alpha)$ and $o = o_j$, and $P_j(s', o|s, \alpha) = 0$, otherwise. Informally, p_j is uniformly distributed among all $s' \in \Phi_j(s, \alpha)$. For each state s' and $o \in O$, the probability of moving to the successor state s' along with jointly observing o, when executing α in s, denoted $P(s', o|s, \alpha)$, is defined as $\sum_{j=1}^{k} P_j(s', o|s, \alpha)$.

Let s be a state, and let α be an action executable in s. Suppose for every agent $i \in I$, exactly one law **reward** $i\colon r$ **after** δ with $s \cup \alpha \models \delta$ belongs to P. Then, the *reward* to i when executing α in s, denoted $R_i(s, \alpha)$, is defined as r.

We next define the initial probabilistic belief of every agent $i \in I$, which is encoded in the law Ψ_i of the form (6). For each $j \in \{1, \ldots, k\}$, let Φ_j be the set of all states satisfying ψ_j. For each state s, let $P_j(s) = p_j \,/\, |\Phi_j|$, if $s \in \Phi_j$, and $P_j(s) = 0$, otherwise. Agent i's belief about s being the initial state, denoted $b_i^0(s)$, is defined as $\sum_{j=1}^{k} P_j(s)$.

In the sequel, we implicitly assume that all P and Ψ are consistent: We say that P is *consistent* iff for each state s and action α executable in s, (i) there is exactly one law (4) in P with $s \cup \alpha \models \delta$, (ii) each $\Phi_j(s, \alpha)$ as above is nonempty, and (iii) for every agent $i \in I$, there is exactly one law **reward** $i\colon r$ **after** δ in P with $s \cup \alpha \models \delta$. We say that Ψ is *consistent* iff, for every $i \in I$, each Φ_j as above is nonempty.

Example 3.1 (Two Robots). We consider the scenario shown in Fig. 1: There are two robots a_1 and a_2 in a room looking for an object o_1, and trying to bring it out through the only door d_1. Both robots can pick up the object, and also pass it to another robot. A pass attempt is only possible if the two robots are facing in adjacent positions. If the receiving robot is not expecting the object, then it falls down. If the two robots are in the same location, then they both cannot perform any pick up and door crossing action. We assume that the reward for the robot bringing out the object is a bit higher. Hence, there is an additional individual payoff for the robot able to accomplish the goal.

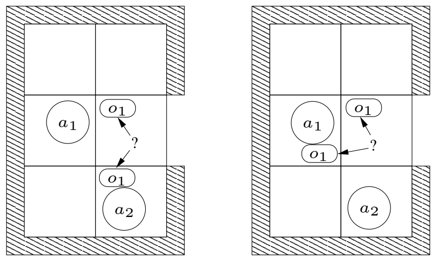

Fig. 1. Initial belief states of a_2 and a_1, respectively.

Let $\mathcal{L} = \{l_{1,1}, \ldots, l_{2,3}, d_1, \mathit{nil}\}$ be the set of possible locations of the robots and the object, where $l_{i,j}$ encodes the field (i, j), and d_1 represents the door. For locations L and L', let $\mathit{close}(L, L')$ be true iff L and L' are adjacent. We assume the simple fluents $\mathit{at}(X)$, where $X \in \{a_1, a_2, o_1\}$, with the domain $\mathcal{L}$, as well as $\mathit{holds}(R)$, where $R \in \{a_1, a_2\}$, with the domain $\{o_1, \mathit{nil}\}$. Let the action variables be given by $\mathit{goTo}(R, L)$, $\mathit{pickUp}(R)$, $\mathit{passTo}(R, R')$, $\mathit{receive}(R)$, where $R, R' \in \{a_1, a_2\}$, $R \neq R'$, and $L \in \mathcal{L}$. Each robot's set of observations is $\{\mathit{obs}(\mathit{holds}),\ \mathit{obs}(\mathit{notHolds})\}$. Informally, each robot can only check if it is carrying something or not after a pick up. We assume the following static causal law:

$$\textbf{caused } \mathit{at}(O) = \mathit{nil} \textbf{ if } \mathit{holds}(R) = O\,.$$

We introduce the following dynamic causal laws for the action variables $\mathit{passTo}(R,R')$, $\mathit{receive}(R)$, and $\mathit{goTo}(R, L)$ (they abbreviate probabilistic causal laws (4) with $k = 1$):

$$\begin{aligned}
&\textbf{caused } \mathit{holds}(R) = \mathit{nil} \textbf{ after } \mathit{passTo}(R, R') \text{ with } R \neq R'\,,\\
&\textbf{caused } \mathit{holds}(R) = O \textbf{ after } \mathit{holds}(R') = O \wedge \mathit{passTo}(R', R) \wedge\\
&\qquad \mathit{receive}(R) \text{ with } R \neq R'\,,\\
&\textbf{caused } \mathit{at}(O) = L \textbf{ after } \mathit{holds}(R) = O \wedge \mathit{passTo}(R, R') \wedge\\
&\qquad \neg\mathit{receive}(R') \wedge \mathit{at}(R, L) \text{ with } R \neq R'\,,\\
&\textbf{caused } \mathit{at}(R) = L \textbf{ after } \mathit{goTo}(R, L)\,.
\end{aligned}$$

Here, if R fails to pass the object O, the latter remains in the location of R. For $\mathit{pickUp}(R)$, we introduce the following probabilistic causal law, assuming $\mathit{pickUp}(R)$ can fail, and $\mathit{obs}(\mathit{notHolds})$ can give incorrect positive results:

$$\begin{aligned}
\textbf{caused } [&(\mathit{holds}(R) = O\,;\, \mathit{obs}(\mathit{holds}))\colon 0.7,\\
&(\mathit{holds}(R) = O\,;\, \mathit{obs}(\mathit{notHolds}))\colon 0.1,\\
&(\mathit{holds}(R) = \mathit{nil}\,;\, \mathit{obs}(\mathit{notHolds}))\colon 0.2]\\
&\qquad\qquad \textbf{after } \mathit{pickUp}(R) \wedge \mathit{at}(R) = L \wedge \mathit{at}(O) = L\,,
\end{aligned}$$

We assume the following execution denials:

$$\begin{aligned}
&\textbf{nonexecutable } \mathit{pickUp}(R) \wedge \mathit{holds}(R) \neq \mathit{nil}\,,\\
&\textbf{nonexecutable } \mathit{pickUp}(R) \wedge \mathit{at}(R) = L \wedge \mathit{at}(o_1) \neq L\,,\\
&\textbf{nonexecutable } \mathit{pickUp}(R) \wedge \mathit{at}(R') = L \wedge \mathit{at}(R) = L\,,\\
&\textbf{nonexecutable } \mathit{goTo}(R, L) \wedge \mathit{at}(R) = L' \wedge \neg\mathit{close}(L, L')\,,\\
&\textbf{nonexecutable } \mathit{goTo}(R, d_1) \wedge \mathit{at}(R) = L \wedge \mathit{at}(R') = L\,,\\
&\textbf{nonexecutable } \mathit{passTo}(R, R') \wedge \mathit{at}(R){=}L \wedge \mathit{at}(R'){=}L' \wedge \neg\mathit{close}(L, L')\,.
\end{aligned}$$

where $R' \neq R$ and $L' \neq L$. Furthermore, every robot can execute only one action at a time, that is, for any two distinct actions α and α' of either robot a_1 or a_2:

$$\textbf{nonexecutable } \alpha \wedge \alpha'\,.$$

For every simple fluent X, we assume the inertial law **inertial** X. Finally, the reward function is defined as follows:

$$\begin{aligned}
&\textbf{reward } a_i : 100 \textbf{ after } \alpha_i \wedge \mathit{holds}(a_i, O)\,,\\
&\textbf{reward } a_i : 90 \textbf{ after } \alpha_i \wedge \mathit{holds}(a_j, O) \text{ with } i \neq j\,,\\
&\textbf{reward } a_i : 10 \textbf{ after } \alpha \wedge \mathit{holds}(a_i, O) \text{ with } \alpha \neq \alpha_i\,,\\
&\textbf{reward } a_i : 0 \textbf{ after } \alpha \wedge \textstyle\bigwedge_{i=1,2} \neg\mathit{holds}(a_i, O) \text{ with } \alpha \neq \alpha_i.
\end{aligned}$$

where $\alpha_i = \mathit{goTo}(a_i, d_1)$. The robot achieving the goal receives a high reward, the other one a bit less. If a robot moves carrying something, it also receives a small payoff.

4 Finite-Horizon Value Iteration

In this section, we define finite-horizon Nash equilibria for probabilistic action descriptions P in $\mathcal{GC}+$ and provide a finite-horizon value iteration for computing them.

Nash Equilibria. We first define belief states and probabilistic transitions between them. A *belief state* of P is of the form $b = (b_i)_{i\in I}$, where every b_i is a probability function over the set of states of P. An action α is *executable* in $b = (b_i)_{i\in I}$ iff for every $i \in I$ the action α is executable in some state s with $b_i(s) > 0$. Then, the new belief state $b^{\alpha,o} = (b_i^{\alpha,o})_{i\in I}$ after executing α in b and observing $o \in O$ is given by:

$$b_i^{\alpha,o}(s') = \textstyle\sum_{s\in S, Poss(\alpha,s)} P(s', o|s,\alpha)\cdot b_i(s)/P(b_i^{\alpha,o}|b_i,\alpha), \text{ where}$$
$$P(b_i^{\alpha,o}\,|\,b_i,\alpha) = \textstyle\sum_{s'\in S}\sum_{s\in S,\, Poss(\alpha,s)} P(s', o\,|\,s,\alpha)\cdot b_i(s)$$

is the probability of observing o after executing α in b_i.

A *mixed policy* is of the form $\pi = (\pi_i)_{i\in I}$, where each π_i assigns to every belief state b and number of steps to go $h \in \{0,\dots,H\}$ a probability function over A_i. The *expected H-step reward* to $i \in I$ under an initial belief state $b{=}(b_i)_{i\in I}$ and the mixed policy π, denoted $G_i(H,b,\pi)$, is defined as

$$\begin{cases} \sum_\alpha (\Pi_{j\in I}\pi_j[b,0](\alpha_j))\cdot \sum_{s\in S, Poss(\alpha,s)} b_i(s)R_i(s,\alpha) & \text{if } H=0;\\ \sum_\alpha (\Pi_{j\in I}\pi_j[b,H](\alpha_j))\cdot (\sum_{s\in S, Poss(\alpha,s)} b_i(s)R_i(s,\alpha) + \\ \qquad \sum_{o\in O} P(b_i^{\alpha,o}\,|b_i,\alpha)\cdot G_i(H{-}1, b^{\alpha,o},\pi)) & \text{otherwise.}\end{cases}$$

A policy π is a *Nash equilibrium* of G iff for each agent $i \in I$ and each belief state b, it holds that $G_i(H,b,\pi'_i\circ\pi_{-i}) \leqslant G_i(H,b,\pi_i\circ\pi_{-i})$ for all π'_i. We are especially interested in *partial Nash equilibria*, which are only defined for an initial belief state and all future belief states within a fixed horizon.

Algorithm. We characterize Nash equilibria of P by finite-horizon value iteration from local Nash equilibria of normal form games. We assume an arbitrary Nash selection function f for normal form games with action set $(A_i)_{i\in I}$. For every belief state $b{=}(b_i)_{i\in I}$ and number of steps to go $h \in \{0,\dots,H\}$, we consider the normal form game $G[b,h] = (I, (A_i)_{i\in I}, (Q_i[b,h])_{i\in I})$, where $Q_i[b,h](\alpha)$ is defined as follows (for all actions α and agents $i \in I$):

$$\begin{cases} \sum_{s\in S,\, Poss(\alpha,s)} b_i(s)R_i(s,\alpha) & \text{if } h=0;\\ \sum_{s\in S,\, Poss(\alpha,s)} b_i(s)R_i(s,\alpha) + \sum_{o\in O} P(b_i^{\alpha,o}\,|b_i,\alpha)\cdot v_f^i(G[b^{\alpha,o},h{-}1]) & \text{otherwise.}\end{cases}$$

The next result shows that the above finite-horizon value iteration computes a Nash equilibrium for consistent probabilistic action descriptions P in $\mathcal{GC}+$.

Theorem 4.1. *Let P be a consistent probabilistic action description in $\mathcal{GC}+$, and $\pi = (\pi_i)_{i\in I}$ be defined by $\pi_i(b,h) = f_i(G[b,h])$ for all agents $i \in I$, belief states b, and number of steps to go $h \in \{0,\dots,H\}$. Then, π is a Nash equilibrium of G, and $G_i(H,b,\pi) = v_f^i(G[b,H])$ for all $i \in I$ and b.*

The following theorem shows that every POSG can be encoded as a consistent probabilistic action description in $\mathcal{GC}+$.

Theorem 4.2. *Let $G = (I, S, (A_i)_{i\in I}, (O_i)_{i\in I}, P, (R_i)_{i\in I})$ be a POSG. Then, there exists a consistent probabilistic action description D in $\mathcal{GC}+$ that encodes G.*

Example 4.1 (Two Robots cont'd). Suppose the initial belief of robot a_1 (resp., a_2) is as in Fig. 1, right (resp., left) side. In particular, a_1 initially believes that o_1 is at $l_{1,2}$ or $l_{2,2}$, while a_2 initially believes that o_1 is at $l_{2,3}$ or $l_{2,2}$. Given the three possible states $s_{1,2}$, $s_{2,2}$, and $s_{2,3}$ such that $s_{i,j} \models at(o_1){=}l_{i,j}$, let the probabilities be given by $b_1(s_{1,2}) = 0.2$ and $b_1(s_{2,2}) = 0.8$ for a_1, and by $b_2(s_{2,3}) = 0.4$ and $b_2(s_{2,2}) = 0.6$ for a_2.

How should the two robots act in such an initial situation? We now apply our finite-horizon value iteration algorithm to compute a partial Nash equilibrium. Notice that $pickUp(a_1, l_{2,3})$ and $pickUp(a_2, l_{1,2})$ are not executable in the initial belief states of a_2 and a_1, respectively. Hence, $pickUp$ can only be executed in $l_{2,2}$. In this case, each agent wants to get to $l_{1,2}$ first, execute $pickUp$, and cross the door. Assuming a 3-step horizon, we obtain two pure partial policies α_i, one for each agent a_i: (1) at 3 steps to go, α_i assigns the action $a = goTo(a_i, l_{2,2})$ to b_i, while any executable action b_j except for $goTo(a_j, l_{2,2})$ is assigned to b_j; (2) at 2 steps to go, a_i executes $pickUp_i(a_i, l_{2,2})$ from b_i^a, while a_j avoids $goTo(a_j, l_{2,2})$ from b_j^b; (3) at 1 step to go, a_i performs $goTo(a_i, d_1)$ in any reached belief state (both after $obs(holds)$ and $obs(notHolds)$), while a_j can execute any action. Both α_1 and α_2 represent a pure partial Nash equilibrium, where the expected 3-step reward of α_1 and α_2 for the robot pair (a_1, a_2) is $(70.4, 43.2)$ and $(52.8, 57.6)$, respectively. Another Nash equilibrium can be obtained form the previous policies by randomizing the first action selection with $\pi_1(b_1, a) = 0.55$ for $a = goTo(a_1, l_{2,2})$ ($\Sigma_\beta \pi_1(b_1, \beta) = 0.45$ with $\beta \neq a$), and $\pi_2(b_2, a) = 0.56$ for $a = goTo(a_2, l_{2,2})$ ($\Sigma_\beta \pi_2(b_2, \beta) = 0.44$ with $\beta \neq a$). Depending on the first action execution, the remaining policy is defined as in α_1 or α_2. In this case, the expected 3-step reward is $G_1(3, b_1, \pi) = 30.67$ and $G_2(3, b_2, \pi) = 25.70$.

5 Reductions and Special Cases

Computing partial Nash equilibria for a probabilistic action description P and an initial belief state requires the following computations: (i) computing the set of all states for P, (ii) deciding whether an action α is executable in a state s, (iii) computing all probabilistic transitions $P(s', o \mid s, \alpha)$, and (iv) computing Nash equilibria of normal form games. Here, (ii) can be easily done in polynomial time on P, while (iv) can be done with standard technology from game theory (see especially [19]). Finally, (i) and (iii) can be reduced to reasoning in causal theories as follows.

Reduction to Causal Theories. We first recall the main concepts of causal theories [13]. A *(causal) rule* has the form $\psi \Leftarrow \phi$ with formulas ψ and ϕ, called its *head* and *body*, respectively. A *causal theory* T is a finite set of rules. Let I be an interpretation of the variables in T. The *reduct* of T relative to I, denoted T^I, is defined as $\{\psi \mid \psi \Leftarrow \phi \in T, I \models \phi\}$. We say I is a *model* of T iff I is the unique model of T^I.

The following result shows that the tasks (i) and (iii) above can be reduced to computing the set of all models of a causal theory. It follows from the original semantics of $\mathcal{C}+$ based on causal theories [13]. In the case of *definite* causal laws, where all law heads ψ in (1) and (2) are literals, the set of all models of the corresponding causal theories can be computed using the Causal Calculator and answer set programming [13].

Proposition 5.1. *Let D be an action description.*

(a) Let T be the set of all rules $\psi \Leftarrow \phi$ such that either (i) ***caused*** *ψ* ***if*** *$\phi \in D$, or (ii) $\phi = \psi = X{=}x$ for some simple fluent $X \in \mathcal{X}$ and $x \in I(X)$. Then, an interpretation s of all fluents and rigid variables is a state of D iff it is a model of T.*

(b) Let α be an action executable in state s. Let $T_{s\cup\alpha}$ be the set of all $\psi \Leftarrow \phi$ such that either (i) ***caused*** *ψ* ***if*** *$\phi \in D$, or (ii) $s \cup \alpha \models \theta$ for some* ***caused*** *ψ* ***if*** *ϕ* ***after*** *$\theta \in D$. Then, $\Phi(s, \alpha)$ is the set of all models s' of $T_{s\cup\alpha}$ that coincide with s on all rigid variables.*

Acyclic Action Descriptions. The action description of Section 3 is *acyclic*, which allows for polynomial-time computations, as we now show. A causal theory T is *acyclic* relative to $W \subseteq V$ iff (i) every rule head is a literal, and (ii) there is a mapping κ from W to the non-negative integers such that $\kappa(X) > \kappa(Y)$ for all $X, Y \in W$ such that X (resp., Y) occurs in the head (resp., body) of some rule in T. An action description D is *acyclic* iff (i) the set of all rules $\psi \Leftarrow \phi$ with **caused** ψ **if** $\phi \in D$ is acyclic relative to all statically determined fluents and rigid variables, and (ii) for each state s and action α executable in s, it holds that $T_{s\cup\alpha}$ is acyclic relative to all fluents.

The following result shows that, in the acyclic case, every interpretation of the simple fluents produces at most one state, which is computable in polynomial time. Similarly, the $\Phi(s, \alpha)$'s contain at most one state, and are computable in polynomial time.

Theorem 5.1. *Let D be an acyclic action description. Then: (a) Every interpretation f of the set of all simple fluents can be extended to at most one state s of D. (b) Deciding whether such s exists and computing it can be done in polynomial time. (c) If s is a state and α an action executable in s, then $\Phi(s, \alpha)$ is either empty or a singleton, and it is computable in polynomial time.*

6 Summary and Outlook

We have presented the action language $\mathcal{GC}+$ for reasoning about actions in multi-agent systems under probabilistic uncertainty and partial observability, which is an extension of the action language $\mathcal{C}+$ that is inspired by partially observable stochastic games (POSGs). We have provided a finite-horizon value iteration algorithm and shown that it characterizes finite-horizon Nash equilibria. We have also given a reduction to non-monotonic causal theories and identified the special case of acyclic action descriptions in $\mathcal{GC}+$, where transitions are computable in polynomial time.

An interesting topic of future research is to define similar action languages for more general classes of POSGs and decentralized POMDPs.

Acknowledgments. This work has been supported by the Austrian Science Fund Project P18146-N04 and a Heisenberg Professorship of the German Research Foundation. We thank the reviewers for their constructive comments, which helped to improve our work.

References

1. F. Bacchus, J. Y. Halpern, and H. J. Levesque. Reasoning about noisy sensors and effectors in the situation calculus. *Artif. Intell.*, 111(1-2):171–208, 1999.

2. C. Baral, N. Tran, and L.-C. Tuan. Reasoning about actions in a probabilistic setting. In *Proceedings AAAI-2002*, pp. 507–512, 2002.
3. D. S. Bernstein, S. Zilberstein, and N. Immerman. The complexity of decentralized control of Markov decision processes. In *Proceedings UAI-2000*, pp. 32–37, 2000.
4. C. Boutilier. Sequential optimality and coordination in multiagent systems. In *Proceedings IJCAI-1999*, pp. 478–485, 1999.
5. C. Boutilier, R. Reiter, and B. Price. Symbolic dynamic programming for first-order MDPs. In *Proceedings IJCAI-2001*, pp. 690–700, 2001.
6. C. Boutilier, R. Reiter, M. Soutchanski, and S. Thrun. Decision-theoretic, high-level agent programming in the situation calculus. In *Proceedings AAAI-2000*, pp. 355–362, 2000.
7. T. Eiter, W. Faber, N. Leone, G. Pfeifer, and A. Polleres. A logic programming approach to knowledge-state planning, II: The DLV$^{\mathcal{K}}$ system. *Artif. Intell.*, 144(1-2):157–211, 2003.
8. T. Eiter and T. Lukasiewicz. Probabilistic reasoning about actions in nonmonotonic causal theories. In *Proceedings UAI-2003*, pp. 192–199, 2003.
9. R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun. Game theoretic control for robot teams. In *Proceedings ICRA-2005*, pp. 1175–1181, 2005.
10. A. Finzi and T. Lukasiewicz. Game-theoretic reasoning about actions in nonmonotonic causal theories. Report Nr. 1843-05-04, Institut für Informationssysteme, TU Wien, 2005.
11. N. H. Gardiol and L. P. Kaelbling. Envelope-based planning in relational MDPs. In *Proceedings NIPS-2003*, 2003.
12. M. Gelfond and V. Lifschitz. Representing action and change by logic programs. *J. Logic Program.*, 17:301–322, 1993.
13. E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain, and H. Turner. Nonmonotonic causal theories. *Artif. Intell.*, 153(1-2):49–104, 2004.
14. C. Guestrin, D. Koller, C. Gearhart, and N. Kanodia. Generalizing plans to new environments in relational MDPs. In *Proceedings IJCAI-2003*, pp. 1003–1010, 2003.
15. E. A. Hansen, D. S. Bernstein, and S. Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings AAAI-2004*, pp. 709–715, 2004.
16. L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1-2):99–134, 1998.
17. M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings ICML-1994*, pp. 157–163, 1994.
18. N. McCain and H. Turner. Causal theories of action and change. In *Proceedings AAAI-1997*, pp. 460–465, 1997.
19. R. McKelvey and A. McLennan. Computation of equilibria in finite games. In *Handbook of Computational Economics*, pp. 87–142. Elsevier, 1996.
20. R. Nair, M. Tambe, M. Yokoo, D. V. Pynadath, and S. Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *Proceedings IJCAI-2003*, pp. 705–711, 2003.
21. G. Owen. *Game Theory: Second Edition*. Academic Press, 1982.
22. D. Poole. Decision theory, the situation calculus and conditional plans. *Electronic Transactions on Artificial Intelligence*, 2(1-2):105–158, 1998.
23. M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994.
24. R. Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.
25. J. van der Wal. *Stochastic Dynamic Programming*, volume 139 of *Mathematical Centre Tracts*. Morgan Kaufmann, 1981.
26. J. von Neumann and O. Morgenstern. *The Theory of Games and Economic Behavior*. Princeton University Press, 1947.