Learning to Act: Qualitative Learning of Deterministic Action Models

Thomas Bolander   Nina Gierasimczuk

October 8, 2017

Abstract

In this paper we study learnability of fully observable, universally applicable action models of dynamic epistemic logic. We introduce a framework for actions seen as sets of transitions between propositional states and we relate them to their dynamic epistemic logic representations as action models. We introduce and discuss a wide range of properties of actions and action models and relate them via correspondence results. We check two basic learnability criteria for action models: finite identifiability (conclusively inferring the appropriate action model in finite time) and identifiability in the limit (inconclusive convergence to the right action model). We show that deterministic actions are finitely identifiable, while arbitrary (non-deterministic) actions require more learning power—they are identifiable in the limit. We then move on to a particular learning method, i.e., learning via update, which proceeds via restriction of a space of events within a learning-specific action model. We show how this method can be adapted to learn conditional and unconditional deterministic action models. We propose update learning mechanisms for the afore-mentioned classes of actions and analyse their computational complexity. Finally, we study a parametrised learning method which makes use of the upper bound on the number of propositions relevant for a given learning scenario. We conclude with describing related work and numerous directions of further work.

1   Introduction

Dynamic epistemic logic (DEL) allows analyzing knowledge change in a systematic way. The static component of a situation is represented by an epistemic model, while the structure of the dynamic component is encoded in an *action model*. An action model can be applied to the epistemic model via the so-called *product update* operation, resulting in a new up-to-date epistemic model of the situation, after the action has been executed. This setting is particularly useful for modeling the process of epistemic planning (see [9, 1]): one can ask which sequence of actions should be executed in order for a given epistemic formula to hold in the resulting epistemic model. A planning agent might not know the

1

effects of her actions, so she will initially not be able to plan to achieve any goals. However, if she can learn the relevant action models through observing the effect of the actions (either by executing the actions herself, or by observing other agents), she will eventually learn how to plan. Our ultimate goal is to integrate learning of actions into (epistemic) planning agents.

In this paper, we seek to lay the foundations for this goal by studying learnability of action models from streams of observations. We investigate possible learning mechanisms involved in discovering the 'internal structure' of actions on the basis of their executions. In other words, we are concerned with qualitative learning of action models on the basis of observations of pairs of the form (initial state, resulting state). We contrast the extensional view of actions (as sets of transitions observed by the learning agent) with their more concise representations as action models (which can serve as learner's hypothesis language).

The structure of the paper is as follows. Firstly, we recall the standard notions of epistemic logic, then we move to discuss actions as sets of transitions between propositional states. We relate this general setting to that of action models in dynamic epistemic logic via correspondence theorems. While doing that we also give ways to simplify action models without giving up their power. In Section 2 we study general learnability properties of action models, drawing from the existing work on the concepts of formal learning theory applied to dynamic epistemic logic (see, e.g., [15, 17, 16]). We show that deterministic action models are conclusively learnable (finitely identifiable), while arbitrary (including non-deterministic) actions are not. We then show that the latter class is identifiable in the limit. In the rest of the paper we study learning deterministic actions by update, i.e., by removing components of action models which are inconsistent with the incoming information. In Section 3 we propose an update learner which finitely identifies unconditional deterministic action models, we analyse the learner's complexity, and discuss possibilities for improvements. In Section 4 we do the same for conditional deterministic action models. Finally, we introduce and study the concept of parametrised learning, which makes use of the upper bound on the number of propositions relevant for a given learning scenario. In the last section we conclude and discuss directions of further work.

This paper is an extension of [10]. The additions are substantial and include the conceptual separation between actions and action models, improved definitions of a variety of properties of actions, improved update learning methods, a new notion of effect learning, computational complexity results, a strengthened parametrised learning result, and full proofs of all results.

1.1 Epistemic language and states

Following the conventions of automated planning, we take the set of atomic propositions and the set of actions to be finite. In the following, P will always refer to a given finite set of *atomic propositions* (*atoms*). To keep the exposition simple, we will generally not mention the dependency on P when defining our languages, states, and actions.

2

We define the *epistemic language* $\mathcal{L}_{epis}$ in the following way:

$$\phi ::= \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid K\phi,$$

where $p \in P$. The language $\mathcal{L}_{prop}$ is the propositional sublanguage without the $K\phi$ clause. By means of the standard abbreviations we introduce the additional symbols $\rightarrow$, $\vee$, $\leftrightarrow$, and $\bot$. A *literal* is either $\top$, a proposition $p \in P$, or the negation of a proposition, $\neg p$.

Definition 1 (Epistemic models and states). *An* epistemic model *is* $m = (W, R, V)$, *where W is a finite set of* worlds, $R \subseteq W \times W$ *is an equivalence relation, called the* indistinguishability relation, *and $V : P \rightarrow \mathcal{P}(W)$ is a* valuation function. *An* epistemic state *is a pointed epistemic model (m, w) consisting of an epistemic model $m = (W, R, V)$ and a distinguished world $w \in W$, called* the actual world.

A *propositional state* (or simply *state*) s is a set of atomic propositions, $s \subseteq P$. One can just as well think of a propositional state in terms of a propositional valuation $\nu_s : P \rightarrow \{0, 1\}$. We identify propositional states and singleton epistemic models via the following canonical isomorphism. A propositional state $s \subseteq P$ is isomorphic to the epistemic model $m = (\{w\}, \{(w, w)\}, V)$ where $V(p) = \{w\}$ if $p \in s$ and $V(p) = \emptyset$ otherwise. Truth for $\mathcal{L}_{epis}$ in epistemic states (and hence propositional states) (m, w) with $m = (W, R, V)$ is defined as follows:

$(m, w) \models p$	iff	$w \in V(p)$
$(m, w) \models \neg\phi$	iff	$m, w \not\models \phi$
$(m, w) \models \phi \wedge \psi$	iff	$m, w \models \phi$ and $m, w \models \psi$
$(m, w) \models K\phi$	iff	for all $v \in W$, if wRv then $m, v \models \phi$

We write $\models \phi$ to mean that $(m, w) \models \phi$ for all epistemic states (m, w). When $\phi \in \mathcal{L}_{prop}$, $\models \phi$ simply means that ϕ is propositionally valid. We write $\phi \models \psi$ to mean that for all epistemic states (m, w), if $(m, w) \models \phi$ then $(m, w) \models \psi$.

1.2 Actions

Actions can be thought of as state-transition functions, i.e., mappings that transform propositional states. Equivalently, an action can be taken extensionally, as the set of pairs (s, s'), where s' is a state that can be reached by executing the action in state s. We make use of this extensional representation below by defining the general notion of an action in terms of the possible state transitions it induces.

Definition 2. *An* action α *is a subset of* $2^P \times 2^P$. *The action is* deterministic *if for every $s \in 2^P$, there exists at most one $s' \in 2^P$ with $(s, s') \in \alpha$. The action is* universally applicable *if for every $s \in 2^P$, there is at least one $s' \in 2^P$ with $(s, s') \in \alpha$.*

Determinism means that an action cannot yield two different effects in one propositional state. Universal applicability means that the action always yields an outcome. In this paper we will almost exclusively be concerned with universally applicable actions. To understand the reason for this restriction consider the example of an action *open_door*. One might say that the action is only applicable if the door is currently closed and unlocked. When the door is either already open or is locked the action will not yield the desired results. We are then faced with a modelling choice, we can either say that the transition function is partial, i.e., sometimes undefined, or prescribe that in such circumstances simply 'nothing happens', i.e., the function returns the same state. In this paper we will keep to the latter option, for two reasons. Firstly, if an agent is learning the results of an action, she should in any possible state be able to *attempt* executing the action, and hence the action should specify an outcome of this attempt. Secondly, it will slightly simplify our later definitions and results.

Let us now turn to conditionality of actions. As an intuitive example of a conditional action we can consider a push button that turns a lamp on if the lamp is off and vice versa. The outcome of the action of pushing the button depends on the initial state of the lamp, i.e., it is conditional on the precondition of the lamp being on. In order to define the notion of conditionality in full generality we need to go through a number of relevant concepts. Let us start with defining what it mean for an action to be uniform in a set of propositions. In the definition below, we use $\ominus$ to denote the symmetric difference between two sets.

Definition 3. *A deterministic, universally applicable action α is said to be uniform in a set of atomic propositions $S \subseteq P$ if the following condition holds:*

- *For all $s \in 2^P$ there exist disjoint sets P^+ and P^- such that for all $s' \in 2^P$ with $s' \ominus s \subseteq S$, $(s', (s' - P^-) \cup P^+) \in \alpha$.*

Intuitively, an action α is uniform in the set of propositions S if the behaviour of α does not change as long as the initial states only vary on the propositions in S.

Proposition 1. *For any deterministic, universally applicable action α there is a largest set S that α is uniform in.*

Proof. It suffices to prove that if α is uniform in both S_0 and S_1 then it is uniform in $S_0 \cup S_1$. Let $s \in 2^P$ be given. We need to find disjoint sets P^+ and P^- such that for all $s' \in 2^P$ with $s' \ominus s \subseteq S_0 \cup S_1$, $(s', (s' - P^-) \cup P^+) \in \alpha$. By uniformity in S_0, there exists disjoint sets $P_{0,s}^+$ and $P_{0,s}^-$ such that for all t with $t \ominus s \subseteq S_0$, $(t, (t - P_{0,s}^-) \cup P_{0,s}^+) \in \alpha$. By uniformity in S_1, for each such t there exists disjoint sets $P_{1,t}^+$ and $P_{1,t}^-$ such that for all s' with $s' \ominus t \subseteq S_1$, $(s', (s' - P_{1,t}^-) \cup P_{1,t}^+) \in \alpha$.

Claim 1. For all t with $s \ominus t \subseteq S_0$, we have $(P_{1,t}^+ \ominus P_{1,s}^+) \cap S_1 = (P_{1,t}^- \ominus P_{1,s}^-) \cap S_1 = \emptyset$.

4

Proof of claim. We only show $(P_{1,t}^+ \ominus P_{1,s}^+) \cap S_1 = \emptyset$, the other case being symmetric. Let $\bar{s} = s - S_1$ and $\bar{t} = t - S_1$. Then $s \ominus \bar{s} \subseteq S_1$, $t \ominus \bar{t} \subseteq S_1$, and $\bar{s} \ominus \bar{t} \subseteq s \ominus t \subseteq S_0$. From $s \ominus \bar{s} \subseteq S_1$, $t \ominus \bar{t} \subseteq S_1$ and choice of $P_{1,s}^+, P_{1,s}^-, P_{1,t}^+$ and $P_{1,t}^-$, we get

$$(\bar{s}, (\bar{s} - P_{1,s}^-) \cup P_{1,s}^+) \in \alpha \tag{1}$$

$$(\bar{t}, (\bar{t} - P_{1,t}^-) \cup P_{1,t}^+) \in \alpha \tag{2}$$

By uniformity in S_0 there exists disjoint sets $P_{0,\bar{s}}^+$ and $P_{0,\bar{s}}^-$ such that for all u with $u \ominus \bar{s} \subseteq S_0$ we have $(u, (u - P_{0,\bar{s}}^-) \cup P_{0,\bar{s}}^+) \in \alpha$. Using $\bar{s} \ominus \bar{t} \subseteq S_0$ we then get

$$(\bar{s}, (\bar{s} - P_{0,\bar{s}}^-) \cup P_{0,\bar{s}}^+) \in \alpha \tag{3}$$

$$(\bar{t}, (\bar{t} - P_{0,\bar{s}}^-) \cup P_{0,\bar{s}}^+) \in \alpha \tag{4}$$

Since α is deterministic, (1)–(4) gives us

$$(\bar{s} - P_{1,s}^-) \cup P_{1,s}^+ = (\bar{s} - P_{0,\bar{s}}^-) \cup P_{0,\bar{s}}^+ \tag{5}$$

$$(\bar{t} - P_{1,t}^-) \cup P_{1,t}^+ = (\bar{t} - P_{0,\bar{s}}^-) \cup P_{0,\bar{s}}^+ \tag{6}$$

From (5)–(6) we can conclude

$$P_{1,s}^+ \ominus P_{0,\bar{s}}^+ \subseteq \bar{s} \tag{7}$$

$$P_{1,t}^+ \ominus P_{0,\bar{s}}^+ \subseteq \bar{t} \tag{8}$$

Since $\bar{s} \cap S_1 = \bar{t} \cap S_1 = \emptyset$, we can from (7)–(8) immediately conclude

$$(P_{1,s}^+ \ominus P_{0,\bar{s}}^+) \cap S_1 = \emptyset \tag{9}$$

$$(P_{1,t}^+ \ominus P_{0,\bar{s}}^+) \cap S_1 = \emptyset \tag{10}$$

From this we get $(P_{1,s}^+ \ominus P_{1,t}^+) \cap S_1 = \emptyset$ as required. This completes the proof of the claim.

We now define P^+ and P^- as follows

$$P^+ = (P_{0,s}^+ - S_1) \cup (P_{1,s}^+ \cap S_1)$$

$$P^- = (P_{0,s}^- - S_1) \cup (P_{1,s}^- \cap S_1)$$

Let $s' \ominus s \subseteq S_0 \cup S_1$. We need to prove $(s', (s' - P^-) \cup P^+) \in \alpha$. Since $s' \ominus s \subseteq S_0 \cup S_1$, there exists t with $s \ominus t \subseteq S_0$ and $t \ominus s' \subseteq S_1$. We then have $(s', (s' - P_{1,t}^-) \cup P_{1,t}^+) \in \alpha$. It hence suffices to show that $(s' - P_{1,t}^-) \cup P_{1,t}^+ = (s' - P^-) \cup P^+$. We prove this by demonstrating that $((s' - P_{1,t}^-) \cup P_{1,t}^+) \cap S_1 = ((s' - P^-) \cup P^+) \cap S_1$ and $((s' - P_{1,t}^-) \cup P_{1,t}^+) \cap (P - S_1) = ((s' - P^-) \cup P^+) \cap (P - S_1)$.

$$
\begin{aligned}
& ((s' - P_{1,t}^-) \cup P_{1,t}^+) \cap S_1 \\
= \; & ((s' - P_{1,s}^-) \cup P_{1,s}^+) \cap S_1 \quad \text{using Claim 1} \\
= \; & ((s' - P^-) \cup P^+) \cap S_1 \quad \text{by def. of } P^+, P^-
\end{aligned}
$$

5

Now note that since $s \ominus t \subseteq S_0$ we have $(t, (t - P_{0,s}^-) \cup P_{0,s}^+) \in \alpha$. We also have $(t, (t - P_{1,t}^-) \cup P_{1,t}^+) \in \alpha$. Thus, since α is deterministic, $(t - P_{0,s}^-) \cup P_{0,s}^+ = (t - P_{1,t}^-) \cup P_{1,t}^0$. We now get

$$
\begin{aligned}
& ((s' - P_{1,t}^-) \cup P_{1,t}^+) \cap (P - S_1) \\
= \ & ((t - P_{1,t}^-) \cup P_{1,t}^+) \cap (P - S_1) && \text{since } s' \ominus t \subseteq S_1 \\
= \ & ((t - P_{0,s}^-) \cup P_{0,s}^+) \cap (P - S_1) \\
= \ & ((t - P-) \cup P^+) \cap (P - S_1) && \text{by def. of } P^+, P^- \\
= \ & ((s' - P^-) \cup P^+) \cap (P - S_1) && \text{since } s' \ominus t \subseteq S_1
\end{aligned}
$$

$\square$

The proposition above guarantees that the following notion is well-defined.

Definition 4. *The set of* preconditions *of a deterministic, universally applicable action α is the smallest set $pre(\alpha)$ such that α is uniform in $P - pre(\alpha)$.*

An action with $pre(\alpha) = \emptyset$ is called unconditional *(otherwise it is called* conditional*).*

Intuitively, the set of preconditions is the smallest set $pre(\alpha)$ such that whenever α can affect a subset of propositions in a certain way in a state s, it can affect those propositions in the exact same way in any other state s' that does not differ from s on any elements of $pre(\alpha)$. The special case of an unconditional action α can be intuitively described as follows: Whenever α can affect a subset of propositions in a certain way in a state s, it can affect those propositions in the exact same way in any other state s'.

Example 1. Let us get back to the simple example of the conditional action of a push button that turns a lamp on if the lamp is off and vice versa (see also [12]). Letting $P = \{p\}$ where p stands for 'the lamp is on', this action can be described as $\alpha = \{(\{p\}, \emptyset), (\emptyset, \{p\})\}$. This action is not uniform in $\{p\}$: if it were, it would have to affect the proposition p in the same way in the two states $\emptyset$ and $\{p\}$. Hence, the smallest set $pre(\alpha)$ for which α is uniform in $P - pre(\alpha)$ is $pre(\alpha) = \{p\}$. In other words, the precondition of the lamp action is p: the outcome of the action depends on whether the lamp is currently on or not.

Definition 5. *The set of* postconditions *of a deterministic, universally applicable action α is $post(\alpha) = \{p \in P \mid \text{for some } (s, t) \in \alpha, \ p \in s \ominus t\}$.*

In other words, the set of postconditions of an action α is the set of propositions whose truth value can change as a result of the execution of α.

Instead of describing actions explicitly and extensionally by a set of possible transitions, they can be also described implicitly, and usually more compactly, in a formal action-description language. Examples of such languages are STRIPS and PDDL in the domain of automated planning [11], action languages like mA^+ in knowledge representation and reasoning [7], and *action models* in dynamic epistemic logic [6]. The latter representation is the one we will use quite extensively below.

1.3   Action models

Dynamic epistemic logic (DEL) introduces the concept of an action model for representing the changes to states brought about by the execution of an action [6]. We here use a variant that includes postconditions [24].

Definition 6 (Action model). *An action model is $a = (E, Q, pre, post)$, where*

- *E is a finite set of* events;

- *$Q \subseteq E \times E$ is an equivalence relation called the* indistinguishability relation;

- *$pre : E \to \mathcal{L}_{epis}$ assigns to each event a* precondition;

- *$post : E \to (P \to \mathcal{L}_{epis})$ assigns to each event a* postcondition. *Postconditions are mappings from atomic propositions to formulas of the epistemic language.*

We use $dom(a) = E$ to denote the domain *of a. The set of all action models is denoted* ActionModels.

In an event e, $pre(e)$ specifies what conditions have to be satisfied for it to take effect, and $post(e)$ specifies its outcome. The outcome is specified in terms of which propositions become true/false after the event has occurred. An atomic proposition p is true *after* e has occurred if the formula $post(e)(p)$ was true *before* e occurred. The details of how a state s is updated with the events of an action model a are given below.

Definition 7 (Product update). *Let $m = (W, R, V)$ and $a = (E, Q, pre, post)$ be an epistemic model and action model, respectively. The* product update *of m with a is the epistemic model $m \otimes a = (W', R', V')$, where*

- $W' = \{(w, e) \in W \times E \mid (m, w) \models pre(e)\}$;

- $R' = \{((w, e), (v, f)) \in W' \times W' \mid wRv \text{ and } eQf\}$;

- $V'(p) = \{(w, e) \in W' \mid (m, w) \models post(e)(p)\}$.

The product update $m \otimes a$ represents the result of executing the action a in the state represented by m.

Example 2. Consider the action of tossing a coin. It can be represented by the following action model (h means that the coin is facing heads up):

$$a = \boxed{\quad e_1 : \langle \top \,;\, h \mapsto \top \rangle \qquad e_2 : \langle \top \,;\, h \mapsto \bot \rangle \quad}$$

We label each event e by a semicolon separated pair $\langle pre(e) \,;\, post(e) \rangle$, whose first element is the precondition of the event, while the second is its postcondition. For representing postconditions, we use the following convention.

Assume $post(e)$ is defined by $post(e)(p_i) = \phi_i$ for each $i \in \{1, \ldots, n\}$ and $post(e)(p) = p$ for all $p \notin \{p_1, \ldots, p_n\}$. Then we represent $post(e)$ by the sequence $p_1 \mapsto \phi_1, \ldots, p_n \mapsto \phi_n$.

Hence, formally for the action model above we have $a = (E, Q, pre, post)$ with $E = \{e_1, e_2\}$, Q is the identity on E (reflexive edges are systematically omitted in this paper), $pre(e_1) = pre(e_2) = \top$, $post(e_1)(h) = \top$ and $post(e_2)(h) = \bot$. The action model encodes that tossing the coin will either make h true (e_1) or h false (e_2).

Consider an agent seeing a coin lying heads up, i.e., the singleton epistemic state $m = (\{w\}, \{(w, w)\}, V)$ with $V(h) = \{w\}$. Let us now calculate the result of executing the coin toss in this model.

$$m' = m \otimes a = \quad \begin{array}{cc} \bullet & \bullet \\ (w, e_1) \colon h & (w, e_2) \colon \end{array}$$

In the figure above each world is labelled by the propositions it makes true.

1.4 Action model types

Let us now define a number of action model types whose learnability we will investigate later in this paper.

Definition 8 (Action model types). *An action model $a = (E, Q, pre, post)$ is:*

- atomic *if $|E| = 1$.*

- globally deterministic *if event preconditions are mutually inconsistent, that is $\models (pre(e) \land pre(f)) \to \bot$ for all distinct events $e, f \in E$.*

- fully observable *if Q is the identity relation on E. Otherwise it is* partially observable.

- precondition-free *if $pre(e) = \top$ for all $e \in E$.*

- propositional *if $pre(e) \in \mathcal{L}_{prop}$ and $post(e)(p) \in \mathcal{L}_{prop}$ for all $e \in E$ and $p \in P$.*

- basic *if: 1) all $pre(e)$ are conjunctions of literals; 2) all $post(e)(p)$ are either $\top$, $\bot$ or p; 3) for all $e \in E$ and $p \in P$, if $pre(e) \models p$ then $post(e)(p) \neq \top$, and if $pre(e) \models \neg p$ then $post(e)(p) \neq \bot$.*

- universally applicable *if $\models \bigvee_{e \in E} pre(e)$.*

The set of preconditions *of a basic action model a is $pre(a) = \{p \in P \mid p \text{ occurs in } pre(e) \text{ for some } e \in E\}$, and its set of* postconditions *is $post(a) = \{p \in P \mid post(e)(p) = \bot \text{ or } post(e)(p) = \top \text{ for some } e \in E\}$.*

Note that any basic action model is also propositional. In this paper, we are only going to be concerned with applying action models in *propositional states*.

Let s denote a propositional state, and let $a = (E, Q, pre, post)$ be any action model. Using the definition of product update and the canonical isomorphism between propositional states and singleton epistemic states, we get that $s \otimes a$ is isomorphic to the epistemic model (W', R', V'), where:

- $W' = \{e \in E \mid s \models pre(e)\}$,

- $R' = \{(e, f) \in W' \times W' \mid eQf\}$,

- $V'(p) = \{e \in W' \mid s \models post(e)(p)\}$.

In $s \otimes a$, each world $e \in W'$ should be identified with the corresponding propositional state $\{p \in P \mid s \models post(e)(p)\}$ (the propositional state that satisfies the same atomic propositions as the world e). Assume a is fully observable. Then the indistinguishability relation of $s \otimes a$ is the identity relation. We can hence think of $s \otimes a$ as the set of propositional states of the form $\{p \in P \mid s \models post(e)(p)\}$ for each $e \in E$ with $s \models pre(e)$. More precisely, in this case we have, up to isomorphism,

$$s \otimes a = \{s \otimes e \mid e \in dom(a) \text{ and } s \models pre(e)\},$$

where

$$s \otimes e = \begin{cases} \{p \in P \mid s \models post(e)(p)\} & \text{if } s \models pre(e); \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Above, the action model a consists of events specified by precondition-postcondition pairs. For each event e whose precondition is satisfied in s, the product update produces a new propositional state (set of propositions) $s \otimes e$ prescribed by the postcondition of e. Note that, using the notation above, $t \in s \otimes a$ iff $t = s \otimes e$ for some $e \in dom(a)$ with $s \models pre(e)$. When a is atomic we have $s \otimes a = \{t\}$ for some propositional state t. In this case we will simply write $s \otimes a = t$. When a is fully observable, we can identify it with the set of events $\{\langle pre(e) ; post(e) \rangle \mid e \in dom(a)\}$, again since the indistinguishability relation is the identity. We will use the above notational simplifications and conventions extensively throughout the paper.

Example 3. Consider the action model a of Example 2 (the coin toss) where $P = \{h\}$. The action model has the following properties (see Definition 8): it is fully observable, precondition-free, propositional, basic, and universally applicable (but it is neither atomic nor globally deterministic). Consider an initial propositional state $s = \{h\}$. Then $s \otimes a$ is the epistemic model m' of Example 2. It has two worlds, one in which h is true, and another in which h is false. Using the notational conventions introduced above, we have

$$s \otimes a = \{s \otimes e_1, s \otimes e_2\} = \{s \otimes \langle \top ; h \mapsto \top \rangle, s \otimes \langle \top ; h \mapsto \bot \rangle\} = \{\{h\}, \emptyset\}.$$

Hence, the outcome of tossing the coin is either the propositional state where h is true ($\{h\}$) or the one where h is false ($\emptyset$).

1.5 Relationships between actions and action models

In this section we study some of the relationships between the actions seen as sets of transitions and the action models. Establishing correspondences between the sets of transitions and the models is important when studying learning of actions, because the input to the learner is a stream of observed state transitions, whereas the output is an action model. We first define the notion of the action induced by a fully observable action model. By doing this we indicate how an action model defines a given set of transitions.

Definition 9. *The* action induced by *a fully observable action model a is the action $act(a)$ given by*

$$act(a) = \{(s,t) \mid t \in s \otimes a\}.$$

We sometimes call $act(a)$ the action represented by *or* specified by *a. Two fully observable action models a and b are called* propositionally equivalent, *written $a \equiv_p b$, if $act(a) = act(b)$.*[1]

In the definition above, we have used the earlier introduced convention of taking $s \otimes a$ to be the set $\{s \otimes e \mid e \in dom(a) \text{ and } s \models pre(e)\}$. So '$t \in s \otimes a$' in the formula above means '$t = s \otimes e$ for some $e \in dom(a)$'.

The following result shows that, conversely, any action induces a fully observable action model.

Proposition 2. *For any action α there exists a fully observable and basic action model a with $act(a) = \alpha$.*

Proof. Take any action $\alpha \subseteq 2^P \times 2^P$. We will now construct an action model a for α. For each pair $(s,t) \in \alpha$ we define an event $e_{(s,t)}$, where:

1. $pre(e_{(s,t)}) := \bigwedge_{p \in s} p \wedge \bigwedge_{p' \in P-s} \neg p'$;

2. $post(e_{(s,t)})(p) := \begin{cases} \bot & \text{if } p \in s \text{ and } p \notin t, \\ \top & \text{if } p \notin s \text{ and } p \in t. \\ p & \text{otherwise} \end{cases}$

We define a as the action model consisting of all these events and in which the indistinguishability relation is the identity. Then, clearly, a is fully observable and basic. It remains to argue that $act(a) = \alpha$.

For $act(a) \subseteq \alpha$. Take any $(s,t) \in act(a)$. Then there is an $e_{(s',t')}$ in a, such that $s \otimes e_{(s',t')} = t$. By construction of a, $(s',t') \in \alpha$.

---

[1]Often equivalence between action models is defined via bisimulation. For instance, a and b can be defined as equivalent when $m \otimes a \underline{\leftrightarrow} m \otimes b$ for all epistemic models m, where $\underline{\leftrightarrow}$ denotes standard bisimulation on epistemic models [8]. It is not difficult to see that two fully observable and propositional action models a and b are equivalent in this sense iff they are equivalent in the sense of $act(a) = act(b)$. For non-propositional action models, however, the notion of propositional equivalence defined here and the notion of equivalence via bisimulation are not equivalent.

It hence suffices to prove $(s,t) = (s',t')$. First we show that $s = s'$. Since $s \otimes e_{(s',t')} = t$, we have $s \models pre(e_{(s',t')})$. From the construction of the precondition $e_{(s',t')}$, it follows that s and s' satisfy the same propositions, i.e., $s = s'$. It remains to show that $t = t'$. If $p \in t$, then since $s \otimes e_{(s',t')} = t$, we have either $post(e_{(s',t')})(p) = \top$ or we have $p \in s$ and $post(e_{(s',t')})(p) = p$. In the first case we get $p \in t'$, by definition of $post(e_{(s',t')})(p)$. In the second case we get $p \in s'$ from $p \in s$. But then also $p \in t'$, since otherwise we would have $post_{(s',t')}(p) = \bot$, again by definition. This shows $t \subseteq t'$. Now let $p \in t'$. If $p \notin s'$ then $post(e_{(s',t')})(p) = \top$ and hence $p \in t$. If $p \in s'$ then $post(e_{(s',t')})(p) = p$. In this case also $p \in s$, and so $p \in t$, since $t = e_{(s',t')} \otimes s$.

For $\alpha \subseteq act(a)$. Take any pair $(s,t) \in \alpha$. By construction of a, there is an event $e_{(s,t)}$ in a. Trivially, $s \models pre(e_{(s,t)})$. From the definition of $post(e_{(s,t)})$ we then immediately get $s \otimes e_{(s,t)} = t$, and hence $(s,t) \in act(a)$, as required. $\square$

Obviously, the construction given in the proof is not efficient. It generates an action model with as many events as there are transition pairs. It is important to realise, however, that there often exists DEL representations of actions that are at least exponentially more succinct than their induced actions.

Consider for instance the action model $a = (\{e\}, \{(e,e)\}, pre, post)$ with $e = \langle pre(e) \, ; \, post(e) \rangle = \langle \top \, ; \, \emptyset \rangle$. Here the postcondition $\emptyset$ of e means that $post(e)(p) = p$ for all $p \in P$ (cf. the notational convention introduced in Example 2). Clearly $act(a) = \{(s,s) \mid s \subseteq P\}$. Thus, the induced action $act(a)$ of a is of exponential size in $|P|$, whereas a is of constant size independent of $|P|$. Similarly, an action that flips the truth-values of all propositions can be represented as an action model of size $|P|$ (the atomic action model $\{\langle \top \, ; \, \{p \mapsto \neg p \mid p \in P\} \rangle\}$), whereas the induced action is again of exponential size in $|P|$. The fact that action models can be, and usually are, at least exponentially smaller than their induced actions, is why we seek to learn action models rather than their induced actions. We will below even show that the action models we learn are of worst-case optimal size, that is, no other formalism for representing those actions is asymptotically better in the worst case.

Proposition 3. *Let a be a fully observable action model.*

1. *$act(a)$ is universally applicable iff a is.*

2. *$act(a)$ is deterministic iff some $b \equiv_p a$ is globally deterministic.*

3. *$act(a)$ is universally applicable and deterministic iff some $b \equiv_p a$ is basic, universally applicable, globally deterministic and has $pre(b) = pre(act(a))$ and $post(b) = post(act(a))$.*

4. *$act(a)$ is unconditional, universally applicable, and deterministic iff some $b \equiv_p a$ is precondition-free, basic, and atomic.*

Proof. Item 1, left to right. Assume $act(a)$ is universally applicable. We need to show $\models \bigvee_{e \in E} pre(e)$, that is, for each propositional state s there exists at least one e such that $s \models pre(e)$. Let s be chosen arbitrarily. Since $act(a)$ is

universally applicable, there exists a t such that $(s,t) \in act(a)$. By definition of $act(a)$, we must have $t = s \otimes e$ for some event e in a. But then $s \models pre(e)$, as required.

Item 1, right to left. Assume a is universally applicable, and let s be a propositional state. We need to show the existence of a t such that $(s,t) \in act(a)$. From universal applicability of a, we get the existence of an event e with $s \models pre(e)$. Hence $(s, s \otimes e) \in act(a)$, showing the required.

Item 2, left to right. Assume $act(a)$ is deterministic. Let b denote the action with $act(b) = act(a)$ given by the construction in Proposition 2. We now show that b is globally deterministic. Let $e_{(s,t)}$ and $e_{(s',t')}$ be distinct events of b. We then need to prove that $pre(e_{(s,t)})$ and $pre(e_{(s',t')})$ are mutually inconsistent. Since $e_{(s,t)}$ and $e_{(s',t')}$ are distinct events, (s,t) and (s',t') are distinct pairs of $act(a)$, that is, either $s \neq s'$ or $t \neq t'$. Since $act(a)$ is deterministic, we have that if $s = s'$ then $t = t'$. It follows that $s \neq s'$. Hence, at least one proposition p has distinct truth-values in s and s'. By the definition of the preconditions of the events of b (see item 1 in the enumerated list of the proof of Proposition 2), we conclude that $pre(e_{(s,t)})$ and $pre(e_{(s',t')})$ are mutually inconsistent (they differ on the required truth-value of p).

Item 2, right to left. Assume $b \equiv_p a$ is globally deterministic, and let $(s,t), (s,t') \in act(a) = act(b)$. We need to prove $t = t'$. From the choice of s, t and t' we get $t, t' \in s \otimes b$. There must therefore exist events e and e' in b such that $s \otimes e = t$ and $s \otimes e' = t'$. We hence have $s \models pre(e) \wedge pre(e')$. Since b is globally deterministic, this immediately implies $e = e'$ and hence $t = s \otimes e = s \otimes e' = t'$.

Item 3, left to right. Assume $act(a)$ is universally applicable and deterministic. By Definitions 3 and 4, for each $s \in 2^{pre(act(a))}$ there exists disjoint sets P_s^+ and P_s^- such that for all s' with $s' \cap pre(act(a)) = s \cap pre(act(a))$, $(s', (s' - P_s^-) \cup P_s^+) \in act(a)$. Let b be the fully observable action model containing for each $s \in 2^{pre(act(a))}$ an event e_s with $pre(e_s) = \bigwedge_{p \in s} p \wedge \bigwedge_{p' \in pre(act(a)) - s} \neg p'$ and

$$post(e_s)(p) = \begin{cases} \top & \text{if } p \in P_s^+ - s; \\ \bot & \text{if } p \in P_s^- \cap s; \\ p & \text{otherwise.} \end{cases}$$

Clearly, b is basic, universally applicable, globally deterministic and has $pre(b) = pre(act(a))$. We now show $post(b) = post(act(a))$. We first show $post(b) \subseteq post(act(a))$. Assume $p \in post(b)$. Then $post(e_s)(p) = \top$ or $post(e_s)(p) = \bot$ for some $e_s \in dom(b)$. If $post(e_s)(p) = \top$ then $p \in P_s^+ - s$ and $(s, (s - P_s^-) \cup P_s^+) \in act(a)$, by definition. Letting $t = (s - P_s^-) \cup P_s^+$ we thus get $(s,t) \in act(a)$ and $p \in t - s$. This implies $p \in post(act(a))$. A symmetric argument goes for the case of $post(e_s)(p) = \bot$. We now show $post(act(a)) \subseteq post(b)$. Assume $p \in post(act(a))$. Then $p \in (t - s) \cup (s - t)$ for some $(s,t) \in act(a)$. Assume $p \in t - s$ (the other case being symmetric). Let $s' = s \cap pre(act(a))$. Then $(s, (s - P_{s'}^-) \cup P_{s'}^+) \in act(a)$. Since a is deterministic, $t = (s - P_{s'}^-) \cup P_{s'}^+$. Since

12

$p \in t - s$, also $p \in P_{s'}^+ - s$. This implies $post(e_{s'})(p) = \top$ and hence $p \in post(b)$. We have now proved $post(b) = post(act(a))$.

It remains to be shown that $b \equiv_p a$, that is, $act(b) = act(a)$. First we show $act(a) \subseteq act(b)$. Suppose $(s, t) \in act(a)$. Let $\bar{s} = s \cap pre(act(a))$. We then have $(s, (s - P_{\bar{s}}^-) \cup P_{\bar{s}}^+) \in act(a)$, and since $act(a)$ is deterministic, $t = (s - P_{\bar{s}}^-) \cup P_{\bar{s}}^+$. It follows that $t = s \otimes e_{\bar{s}}$ (noting that $s \models pre(e_{\bar{s}})$), and hence $(s, t) \in act(b)$. We now show $act(b) \subseteq act(a)$. Let $(s, t) \in act(b)$. Then $t = s \otimes e_{\bar{s}}$ for some $\bar{s} \in 2^{pre(act(a))}$. This implies $t = (s - P_{\bar{s}}^-) \cup P_{\bar{s}}^+$, by definition of $e_{\bar{s}}$. Since $\bar{s} \cap pre(act(a)) = s \cap pre(act(a))$, we have $(s, (s - P_{\bar{s}}^-) \cup P_{\bar{s}}^+) \in act(a)$ and thus $(s, t) \in act(a)$.

Item 3, right to left. Assume $b \equiv_p a$ is basic, universally applicable, globally deterministic and has $pre(b) = pre(act(a))$ and $post(b) = post(act(a))$. Then it follows directly from items 1 and 2, right to left, that $act(a)$ is universally applicable and deterministic.

Item 4, left to right. Assume $act(a)$ is unconditional, universally applicable and deterministic. By definition, we then have $pre(act(a)) = \emptyset$. By item 3, left to right, there then exists some $b \equiv_p a$ which is basic, globally deterministic and has $pre(b) = \emptyset$. The action b is hence precondition-free. It must also be atomic, since it is globally deterministic (it has a single event with precondition $\top$).

Item 4, right to left. Assume $b \equiv_p a$ is precondition-free, basic and atomic. Then it is also globally deterministic. That $act(a)$ is universally applicable and deterministic then follows directly from item 3, right to left. So we only need to prove that $act(a) = act(b)$ is unconditional, that is, has an empty set of preconditions. Since b is precondition-free, basic and atomic, it must consist of a single event e with precondition $\top$ and each $post(e)(p)$ is either $\top$, $\bot$ or p. It follows that for all states s, $(s, (s - \{p \mid post(e)(p) = \bot\}) \cup \{p \mid post(e)(p) = \top\}) \in act(b)$. This shows that $act(b)$ is uniform in P, and hence $act(a) = act(b)$ must have an empty set of preconditions. $\square$

2 Learning action models

In this section we introduce and discuss our general learning setting. Below we define streams of observations, learning functions, and, finally, we discuss two learning conditions: finite identifiability and identifiability in the limit. We establish that while deterministic actions allow finite identifiability, the non-deterministic actions do not, but are identifiable in the limit. We place those results in the context of the classical results characterising both types of learning [18, 2, 21, 20]. This is not the first application of learning-theoretic tools to dynamic epistemic logic (see [14, 13, 15, 16]) or to the logical theories of belief revision (see, e.g., [4, 5, 19]). The present work is however pioneering in studying the learning of the internal structure of actions in dynamic epistemic logic.

Definition 10. *A stream $\mathcal{E}$ is an infinite (unbounded) sequence of pairs (s, t) of propositional states, i.e., $\mathcal{E} \in (2^P \times 2^P)^\omega$. The elements (s, t) of $\mathcal{E}$ are called observations. Let $n \in \mathbb{N}$ and let $\mathcal{E}$ be a stream.*

1. $\mathcal{E}_n$ stands for the n-th observation in $\mathcal{E}$.

2. $\mathcal{E}[n]$ stands for the the initial segment of $\mathcal{E}$ of length n, i.e., $\mathcal{E}_0, \ldots, \mathcal{E}_{n-1}$.

3. $\mathrm{set}(\mathcal{E}) := \{(s, t) \mid (s, t) \text{ is an element of } \mathcal{E}\}$ stands for the set of all observations in $\mathcal{E}$; we similarly define $\mathrm{set}(\mathcal{E}[n])$ for initial segments of streams.

Definition 11. *Let $\mathcal{E}$ be a stream and let α be an action. The stream $\mathcal{E}$ is* sound *with respect to α if $\mathrm{set}(\mathcal{E}) \subseteq \alpha$. The stream $\mathcal{E}$ is* complete *with respect to α if $\alpha \subseteq \mathrm{set}(\mathcal{E})$. In this paper we always assume the streams to be sound and complete. For brevity, if $\mathcal{E}$ is sound and complete wrt α, we will write '$\mathcal{E}$ is for α'. Similarly, an initial segment $\mathcal{E}[n]$ is sound for α if $\mathrm{set}(\mathcal{E}[n]) \subseteq \alpha$ and complete for α if $\alpha \subseteq \mathrm{set}(\mathcal{E}[n])$.*

The notions of soundness and completeness extend naturally to action models in the following way. A stream or initial segment of a stream is *sound* (resp. *complete*) with respect to an action model a if it is sound (resp. complete) with respect to $act(a)$.

Definition 12 (Learning function)**.** *A* learning function *is a computable L : $(2^P \times 2^P)^* \to \mathsf{ActionModels} \cup \{\uparrow\}$.*

In other words, a learning function takes a finite sequence of observations (state transitions) and outputs an action model or a symbol corresponding to 'undecided' ($\uparrow$).

We will study two types of learning: finite identifiability and identifiability in the limit. First let us focus on *finite identifiability*. Intuitively, finite identifiability corresponds to conclusive learning: upon observing some finite amount of action executions the learning function outputs, with certainty, a correct model for the action in question. This certainty can be expressed in terms of the function being once-defined: it is allowed to output an action model only once, there is no chance of correction later on (for a more extensive study of finite identifiability, see [17]). Formally, we say that a learning function L is *(at most) once defined* if for any stream $\mathcal{E}$ for an action and $n, k \in \mathbb{N}$ such that $n \neq k$, we have that $L(\mathcal{E}[n]) = \uparrow$ or $L(\mathcal{E}[k]) = \uparrow$.

Definition 13. *Let $\mathcal{X}$ be a set of actions and $\alpha \in \mathcal{X}$, L be a learning function, and $\mathcal{E}$ be a stream. We say that:*

1. *L finitely identifies α on $\mathcal{E}$ if L is once-defined and there is an $n \in \mathbb{N}$ s.t. $act(L(\mathcal{E}[n])) = \alpha$.*

2. *L finitely identifies α if L finitely identifies α on every stream for α.*

3. *L finitely identifies $\mathcal{X}$ if L finitely identifies every $\alpha \in \mathcal{X}$.*

4. *$\mathcal{X}$ is finitely identifiable if there is a function L which finitely identifies $\mathcal{X}$.*

The following definition and theorem are adapted from [21, 20, 17].

Definition 14. *Let $\mathcal{X}$ be a set of actions. A set $D_\alpha \subseteq 2^P \times 2^P$ is a definite finite tell-tale set (DFTT) for α in $\mathcal{X}$ if*

1. *$D_\alpha \subseteq \alpha$,*

2. *D_α is finite, and*

3. *for any $\beta \in \mathcal{X}$, if $D_\alpha \subseteq \beta$, then $\alpha = \beta$.*

Lemma 1. *A set of actions $\mathcal{X}$ is finitely identifiable iff there is an effective procedure $\mathsf{D} : \mathcal{X} \to 2^{(2^P \times 2^P)}$ that on input a gives a DFTT of α.*

Proof. Left to right. Assume that $\mathcal{X}$ is finitely identifiable. Then there is a computable function L that finitely identifies $\mathcal{X}$. We use that function to define D. Once the learning function L identifies an action α it has to give it as a definite output, and this will happen for some $\mathcal{E}[n]$. We then set $\mathsf{D}(\alpha) := \operatorname{set}(\mathcal{E}[n])$. It is easy to check that such $\mathsf{D}(\alpha)$ is a definite tell-tale set (satisfying conditions 1-3 above). *Right to left.* Assume that there is an effective procedure $\mathsf{D} : \mathcal{X} \to 2^{(2^P \times 2^P)}$, that on input α produces a definite finite tell-tale of α. Take an enumeration $\alpha_1, \alpha_2, \ldots$ of $\mathcal{X}$ and take any $\alpha \in \mathcal{X}$ and any $\mathcal{E}$ for α. We use D to define the learning function. At each step $n \in \mathbb{N}$, L compares $\mathcal{E}[n]$ with $\mathsf{D}(\alpha_1), \ldots, \mathsf{D}(\alpha_n)$. Once, at some step $\ell \in \mathbb{N}$, it finds α_k, $k \leq \ell$, such that $\mathsf{D}(\alpha_k) \subseteq \operatorname{set}(\mathcal{E}[\ell])$, it outputs an action model a with $act(a) = \alpha_k$ (using the construction in Proposition 2). It is easy to verify that then $act(a) = \alpha$. $\square$

In other words, the finite set of observations D_α is consistent with only one action α in the class. D is a computable function that gives a D_α for any action α.

Theorem 1. *The set of deterministic and universally applicable actions is finitely identifiable.*

Proof. We use Lemma 1, and hence define: $\mathsf{D}(\alpha) = \alpha$. Let us check that indeed $\mathsf{D}(\alpha)$ is a DFTT for α (conditions 1-3 of Definition 14). 1: $\mathsf{D}(\alpha) \subseteq act(\alpha)$, trivially. 2: $\mathsf{D}(\alpha)$ is finite, because P is finite. 3: Let us take any deterministic and universally applicable action β such that $\mathsf{D}(\alpha) \subseteq \beta$. This means that $\alpha \subseteq \beta$. We need to show $\alpha = \beta$, and it hence suffices to prove $\beta \subseteq \alpha$. Let $(s, t) \in \beta$. We need to prove $(s, t) \in \alpha$. Since α is deterministic and universally applicable, there exists a unique t' such that $(s, t') \in \alpha$. Since $\alpha \subseteq \beta$, we then get $(s, t') \in \beta$. We now have $(s, t), (s, t') \in \beta$, and since β is deterministic, we get $t' = t$. This proves $(s, t) \in \alpha$, as required. Finally, D is computable because P is finite. $\square$

Example 4. Theorem 1 shows that deterministic actions are finitely identifiable. We will now demonstrate that this does not carry over to non-deterministic actions, that is, non-deterministic actions are in general not finitely identifiable. Consider the action of tossing a coin, given by the action model a in Example 2. If in fact the coin is fake and it will always land tails (so it only consists of the event e_2), in no finite amount of tosses the agent can exclude that the coin is fair, and that heads will start appearing in the long run (that e_1 will eventually

occur). So the agent will never be able to say "stop" and declare the correct action model to only consist of e_2. This argument can be generalised, leading to the theorem below.

Theorem 2. *The set of arbitrary (including non-deterministic) universally applicable actions is not finitely identifiable.*

Proof. Let α be a deterministic, universally applicable action. Take some $(s,t) \notin \alpha$. Such a pair necessarily exists, since α is deterministic. Let $\beta = \alpha \cup \{(s,t)\}$. Note that β is not deterministic, since α is universally applicable, and there will hence be two distinct states t and t' with $(s,t), (s,t') \in \beta$. Assume that the set of arbitrary universally applicable actions is finitely identifiable. Then there is a learning function L that finitely identifies it. Among such actions, as we argued above, we will have two, α and β, such that $\alpha \subset \beta$. Let us now construct a stream $\mathcal{E}$ on which L fails to finitely identify one of them. Let $\mathcal{E}$ start with enumerating all pairs of propositional states that are sound for the smaller action, α, and keep repeating this pattern. Since this is a stream for α, indeed the learning function has to at some point output an action model a with $act(a) = \alpha$ (otherwise it fails to finitely identify α, which leads to contradiction). Assume that this happens at some stage $n \in \mathbb{N}$. Now, observe that $\mathcal{E}[n]$ is sound with respect to β too, so starting at the stage $n + 1$ let us make $\mathcal{E}$ enumerate the rest of remaining pairs of propositional states sound for β. That means that there is a stream $\mathcal{E}$ for β on which L does not finitely identify β. Contradiction. $\square$

A weaker condition of learnability, *identifiability in the limit*, allows widening the scope of learnable actions, to cover also the case of non-deterministic actions. Identifiability in the limit requires that the learning function after observing some finite amount of action executions outputs a correct model for the action in question and then forever keeps to this answer in all the outputs to follow. This type of learning can be called 'inconclusive', because certainty cannot be achieved in finite time.

Definition 15. *Let $\mathcal{X}$ be a set of actions and $\alpha \in \mathcal{X}$, L be a learning function, and $\mathcal{E}$ be a stream. We say that:*

1. *L identifies α on $\mathcal{E}$ in the limit if there is $k \in \mathbb{N}$ such that for all $n \geq k$, $L(\mathcal{E}[k]) = L(\mathcal{E}[n])$ and $act(L(\mathcal{E}[n])) = \alpha$.*

2. *L identifies α in the limit if L identifies α in the limit on every $\mathcal{E}$ for α.*

3. *L identifies $\mathcal{X}$ in the limit if L identifies in the limit every $\alpha \in \mathcal{X}$.*

4. *$\mathcal{X}$ is identifiable in the limit if there is an L which identifies $\mathcal{X}$ in the limit.*

Theorem 3. *The set of arbitrary (including non-deterministic and non-universally applicable) actions is identifiable in the limit.*

Proof. The argument is similar to the proof of Theorem 1. Analogously to the concept of definite finite tell-tale set, we define a weaker notion of finite tell-tale set (FTT). Let $\mathcal{X}$ be a set of actions. A set $D_\alpha \subseteq 2^P \times 2^P$ is a finite tell-tale set (FTT) for α in $\mathcal{X}$ if:

1. $D_\alpha \subseteq \alpha$;

2. D_α is finite, and

3. for any $\beta \in \mathcal{X}$, if $D_\alpha \subseteq \beta$, then it is not the case that $\beta \subset \alpha$.

Similarly to the argument for Lemma 1, one can show that $\mathcal{X}$ is identifiable in the limit iff there is an effective procedure $\mathsf{D} : \mathcal{X} \to 2^{(2^P \times 2^P)}$ that on input α enumerates a finite tell-tale of α. We will omit the proof for the sake of brevity (the original argument for the case of grammar inference can be found in [2]).

Now it is enough to show that indeed such a function D can be given for the set of arbitrary actions over P. Define $\mathsf{D}(\alpha) = \alpha$.

Let us check that indeed $\mathsf{D}(\alpha)$ is a FTT for α. 1: $\mathsf{D}(\alpha)$ is sound for α, trivially. 2: $\mathsf{D}(\alpha)$ is finite, because P is finite. 3: Let us take any action β such that $\mathsf{D}(\alpha) \subseteq \beta$, i.e., $\alpha \subseteq \beta$. Then it is clearly not the case that $\beta \subset \alpha$. Finally, again D is computable because P is finite. $\qquad\square$

Having established the general facts about finite identifiability and identifiability in the limit of various types of actions, we will now turn to studying particular learning methods suited for such learning conditions.

2.1   Learning via update

Standard DEL, and in particular public announcement logic [22], models the process of information flow within epistemic models. If an agent is in a state described by an epistemic model m and learns from a reliable source that ϕ is true, her state will be updated by eliminating all the worlds where ϕ is false. That is, the model m will be restricted to the worlds where ϕ is true. This can also be expressed in terms of action models, where the learning of ϕ corresponds to taking the product update of m with the event model $\langle \phi \,; \emptyset \rangle$ (public announcement of ϕ).

Now we turn to learning *actions* rather than learning *facts*. Actions are represented by action models, so to learn an action means to infer the action model that describes it. Consider again the action model a of Example 2. The coin toss is non-deterministic and fully observable: either h or $\neg h$ will non-deterministically be made true and the agent is able to distinguish these two outcomes (there is no edge between e_1 and e_2). However, we can also think of the domain of a as the hypothesis space of a *deterministic* action. Given the prior knowledge that the action in question must be deterministic, learning the action model for it could proceed in a way analogous to that of update in the usual DEL setting. It could for instance be that the agent knows that the coin is fake and always lands on the same side, but the agent initially does not know which. After the agent has executed the action once, she will know. She

will observe either h becoming false or h becoming true, and can hence discard either e_1 or e_2 from her hypothesis space. She has now *learned* a correct action model for the act of tossing the fake coin.

It is a note-worthy analogy: learning of facts means eliminating worlds in epistemic models, learning of actions means eliminating events in action models.

Learning action models via update (deleting events) has a natural interpretation of learning via gradual increase of the 'amount of determinism' within the action model. Initially, the action is taken to be able to do anything and with time the learner acquires a more and more specialised interpretation of what it can do. Of course, the case of non-deterministic actions is more complicated. In that case, no observed execution of an action can exclude other possibilities.

Definition 16. *For any deterministic and fully observable action model a and any pair of propositional states (s,t), the* update *of a with (s,t) is defined by*

$$a \mid (s,t) := \{e \in a \mid if\ s \models pre(e)\ then\ s \otimes e = t\}.$$

For a set S of pairs of propositional states, we define

$$a \mid S := \{e \in a \mid for\ each\ (s,t) \in S,\ if\ s \models pre(e)\ then\ s \otimes e = t\}.$$

The update $a \mid (s,t)$ restricts the action model a to the events that are consistent with observing t as the result of executing the action in question in the state s. This is then lifted to sets of pairs (sets of observations) in the obvious way in the definition of $a \mid S$.

3 Learning unconditional deterministic actions

In this section we will consider learning of unconditional deterministic actions. We will, as everywhere else in this paper, restrict attention to universally applicable propositional actions. The set of atomic propositions P is assumed to be fixed. From Proposition 3, item 4, we have that any unconditional, deterministic and universally applicable action can be represented by a precondition-free, basic and atomic action model (that is, for any such action α, there is a precondition-free, basic and atomic action model a with $act(a) = \alpha$). This implies that if we want to construct a learner that can learn unconditional, deterministic and universally applicable actions, it suffices to consider learning functions that learn action models which are precondition-free, basic and atomic.

In basic action models, each $post(e)(p)$ belongs to the set $\{\top, \bot, p\}$. We can hence consider $post(e)$ to be a partial mapping from atomic propositions to $\{\top, \bot\}$, that is of the form $P \hookrightarrow \{\top, \bot\}$. The interpretation is then that when $post(e)(p)$ is undefined we take this to mean $post(e)(p) = p$. The events of basic action models can hence be considered to be of the form $\langle pre\,;\,f \rangle$, where $f : P \hookrightarrow \{\top, \bot\}$. If an action model is furthermore precondition-free, the events will have the form $\langle \top\,;\,f \rangle$. Any action model which is precondition-free, basic and atomic can hence be represented by a single event of the form $\langle \top\,;\,f \rangle$.
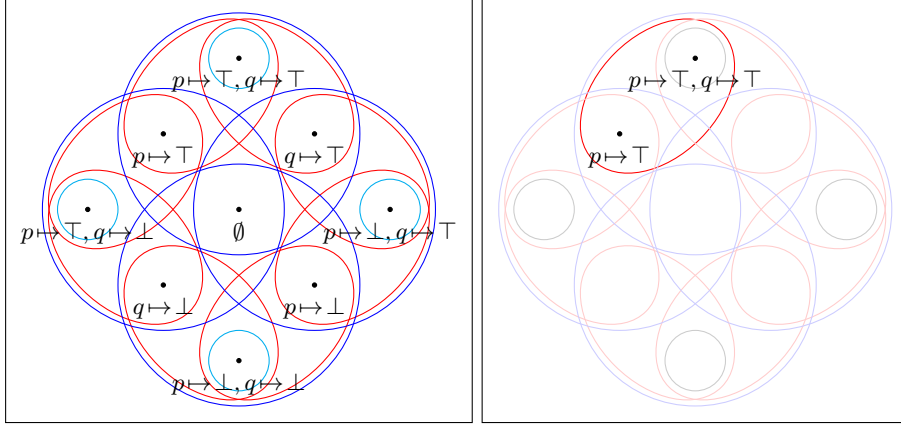
Figure 1: On the left h_0 with $P = \{p, q\}$, together with sets corresponding to possible observations. We have labelled each event e by $post(e)$. On the right the state of learning with L_0 after observing $\mathcal{E}_0 = (\{q\}, \{p, q\})$.

This implies that when learning unconditional, deterministic and universally applicable actions, we only have to look for the right event of the form $\langle \top \, ; f \rangle$ to represent that action. This leads to define our hypothesis space for learning such actions in the following way.

Definition 17. *The* hypothesis space for unconditional actions *is the action model h_0 given by*
$$h_0 = \{\langle \top \, ; f \rangle \mid f : P \hookrightarrow \{\top, \bot\}\}.$$

The hypothesis space h_0 will serve as the starting point of the learning process. The learner will proceed with learning by gradually eliminating the elements inconsistent with the incoming information (this process is known as update learning).

Definition 18. *The* update learning function for unconditional actions *is the learning function L_0 defined by*

$$L_0(\mathcal{E}[n]) = h_0 \mid \text{set}(\mathcal{E}[n]).$$

In Figure 1 we show a generic example of such update learning for $P = \{p, q\}$. If the stream of observations is consistent with one of the events in the space, as this is what we assume within this framework, this event will never be eliminated from the space.

We will define a learning function which makes use of L_0, but outputs an answer when there is only one event left.

Theorem 4. *The set of universally applicable, unconditional and deterministic actions is finitely identifiable by the update learning function L_0^{update}, defined in*

19

the following way:

$$L_0^{update}(\mathcal{E}[n]) = \begin{cases} L_0(\mathcal{E}[n]) & \textit{if } |L_0(\mathcal{E}[n])| = 1 \\ & \textit{and for all } k < n, \ L_0^{update}(\mathcal{E}[k]) = \uparrow; \\ \uparrow & \textit{otherwise.} \end{cases}$$

Proof. Note that L_0^{update} is defined in terms of L_0, which by Definition 18 is given by $L_0(\mathcal{E}[n]) = h_0 \mid \text{set}(\mathcal{E}[n])$, where h_0 is the hypothesis space. Let us take an unconditional deterministic action α and take $\mathcal{E}$ to be a stream for α. By Proposition 3, item 4, there must exist a precondition-free, basic and atomic action model representing α. Hence, for some $e \in h_0$, we must have $act(\{e\}) = \alpha$. We show that L_0^{update} finitely identifies α on $\mathcal{E}$. Since $\mathcal{E}$ is a stream for α, $e \in L_0(\mathcal{E}[n])$ for any n (i.e., e will never be eliminated). It remains to be shown that for some $n \in \mathbb{N}$, $|L_0(\mathcal{E}[n])| = 1$. Let us consider the smallest k such that $\alpha \subseteq \text{set}(\mathcal{E}[k])$. Then there is only one element, e, in $L_0(\mathcal{E}[k])$. It is so because for all $e' \in h_0$ with $e' \neq e$ there is an observation $(s,t) \in 2^P \times 2^P$ such that $(s,t) \in act(\{e\})$ but $(s,t) \notin act(\{e'\})$ (in this case we will say that (s,t) separates e from e'). Upon receiving this information the learner will remove e' from h_0. In Figure 1 this general fact is clearly visible. For any pair of points (events), an ellipse (observation) can be found that separates them (one event is consistent with it and the other is not). To see how those observations can be constructively obtained take any $e \in h_0$. Then for each $e' \in h_0$ with $e' \neq e$, it can easily be checked that at least one of the following observations separates e from e': $(P, P \otimes e)$ or $(\emptyset, \emptyset \otimes e)$. □

3.1 Time and space complexity

Note that L_0^{update} is defined in terms of the update learning function L_0, which in turn is defined in terms of the hypothesis space h_0. The hypothesis space h_0 is clearly exponential in $|P|$ (it contains one event per possible postcondition over P), so a straightforward implementation of L_0^{update} will have a space requirement which is exponential in $|P|$. This kind of learning is clearly very memory-inefficient. Below we will look into how this can be improved. We will first introduce the relevant notions of computational complexity of learning in our setting, and then investigate the computational complexity of learning unconditional deterministic actions. First we consider time complexity and then space complexity. In terms of time complexity, there are two relevant questions. Firstly, how many observations are needed before an action can be identified? Secondly, how many computation steps does the implemented learning function need as a function of the number of observations?

In terms of space complexity, there are also two relevant questions. Firstly, what is the size of the action model provided as output of the learning algorithm? Secondly, how much memory does the learning algorithm use? We will most often measure complexities in terms of the number of atomic propositions underlying the set of actions to be learned.

Time complexity   Assume given a learning function L that finitely identifies a set of actions $\mathcal{X}$ over a set of atomic propositions P. First note that a stream $\mathcal{E}$ for an action $\alpha \in \mathcal{X}$ can have any number of repetitions, and hence in general we can not give an upper bound on the length of the initial segment of $\mathcal{E}$ required for L to identify α. We can however look at the number of *distinct* observations required to learn α, that is, we either ignore repetitions in the stream or we only consider finite streams where all pairs are distinct.

In any case, even for the simplest type of actions, unconditional deterministic actions, any learning function will in the worst case require $1+2^{|P|-1}$ distinct observations before being able to identify the action. To see this, consider the unconditional deterministic action α that makes all propositions in P unconditionally true. It can be represented by an action model $a = \{\langle \top \,;\, \{p \mapsto \top \mid p \in P\}\rangle\}$. Pick a proposition p' in P. Then there are $2^{|P|-1}$ propositional states over P where p' is true. Assume the stream $\mathcal{X}$ first provides an observation of (s, P) for each such propositional state s. Then after these $2^{|P|-1}$ observations, the action can still not be uniquely identified, because the stream is both sound for α and for the action β which is as α except it does not affect the truth-value of p' (that is, it is represented by an action model $\{\langle \top \,;\, \{p \mapsto \top \mid p \in P - \{p'\}\}\rangle\}$). Hence α can at earliest be identified when the $(1+2^{|P|-1})$th distinct observation is made (and actually *will* be identified by that observation as is easily seen).

Since the argument above was independent of the choice of L, it shows that all learning functions for unconditional deterministic actions will have the same worst-case behaviour in terms of the required number of distinct observations. The worst-case required number of distinct observations is hence not a relevant complexity measure in this case. We can however look at *proactive learning* of an action α: Learning where the learner gets to choose in which state s the action α is applied, and the environment then replies with a t for which $(s, t) \in \alpha$. In the case of unconditional deterministic actions this makes a significant difference. The time complexity measured in number of distinct observations goes down from $O(2^{|P|})$ to $O(1)$. Here is the argument. First the learner asks about the effect of applying the action in the state $\emptyset$. This gives the learner an observation of the form $(\emptyset, P_1)$. Then the learner asks about the effect of applying the action in the state P. This gives an observation (P, P_2). Since the action is assumed to be unconditional, the learner now knows that it unconditionally sets all the propositions in P_1 true, and all the propositions in $P - P_2$ false. Hence it must be represented by the atomic action model $\{\langle \top \,;\, \{p \mapsto \top \mid p \in P_1\} \cup \{p \mapsto \bot \mid p \in P - P_2\}\rangle\}$. The learner has now learned the action in only two observations.

However, when moving to learning of conditional actions, even proactive learning is not helpful. This can be seen by realising that in the case of a universally applicable, conditional and deterministic action α, even the best-case number of distinct observations required to identify α is $\Theta(2^{|P|})$. To see this, let $\mathcal{E}$ be any stream for α. We will show that no learner can identify α from the initial segment $\mathcal{E}[2^{|P|} - 1]$. Since $\mathcal{E}[2^{|P|} - 1]$ consists of at most $2^{|P|} - 1$ distinct observations, there must exist a propositional state s such that there is no t with $(s, t) \in \text{set}(\mathcal{E}[2^{|P|} - 1])$. Let t be the propositional state such that

$(s, t) \in \alpha$ (α is deterministic and universally applicable). Let $t' \neq t$. Now let $\beta = (\alpha - \{(s, t)\}) \cup \{(s, t')\}$. The action β is clearly also conditional, deterministic and universally applicable. The initial segment $\mathcal{E}[2^{|P|} - 1]$ is by construction also sound for β, so α can not be uniquely identified from $\mathcal{E}[2^{|P|} - 1]$. This shows that any learning function identifying the set $\mathcal{X}$ of universally applicable, conditional and deterministic actions will always require $\Omega(2^{|P|})$ observations.

The discussion above shows that for finite identifiability, the time complexity measured in the number of required distinct observations is in most cases not a useful measure to compare efficiency of learning functions. It could still be relevant to look at the number of computation steps needed by a learning function L to compute $L(\mathcal{E}[n])$ as a function of n. This will however depend crucially on details of how the learning function is implemented, including details about the choice of data structures.

Space complexity As mentioned earlier, we also have two relevant space measures: the total space required by an algorithm implementing the learning function and the size of the action model provided as output. We provide the space complexity measures for the learning function L_0^{update} in the following proposition.

Proposition 4. L_0^{update} *can be implemented using* $O(|P| \cdot 3^{|P|})$ *space. If* $L_0^{update}(\mathcal{E}[n]) = a$ *for some action model a then a has size* $O(|P|)$.

Proof. L_0^{update} is initialised with the hypothesis space h_0 of Definition 17. The action model h_0 contains $O(3^{|P|})$ events: one for each partial mapping of P into $\{\top, \bot\}$ (so each $p \in P$ is mapped into one of 3 values: $\top$, $\bot$ or 'undefined'). Each event is of size $O(|P|)$ (the length of the postcondition mapping), so the total size of h_0 is $O(|P| \cdot 3^{|P|})$. This is the total space requirement of the learning algorithm, since it now proceeds by only eliminating events from h_0. The size of the resulting action model, the one eventually returned by L_0^{update}, is $O(|P|)$, since it contains a single event. □

3.2 Improved learning of unconditional deterministic actions

We can improve the space complexity of learning unconditional deterministic actions. Instead of updating a hypothesis space, we can keep track of the observed positive and negative effects of the transitions in the stream, and build the action model from those. We call this *effect learning*. Let (s, t) be a pair of propositional states. We define the *observed positive effects* of (s, t) to be the set $P_{(s,t)}^+ = \{p \in P \mid s \models \neg p \text{ and } t \models p\}$. Symmetrically, we define the *observed negative effects* to be $P_{(s,t)}^- = \{p \in P \mid s \models p \text{ and } t \models \neg p\}$. Given an action α, we then define the *observed positive effects* of α as $P_\alpha^+ = \bigcup_{(s,t)\in\alpha} P_{(s,t)}^+$. Symmetrically for the *observed negative effects*. For any pair of disjoint sets $P^+, P^- \subseteq P$, we let $post(P^+, P^-) = \{p \mapsto \top \mid p \in P^+\} \cup \{p \mapsto \bot \mid p \in P^-\}$. We now get the following result.

Theorem 5. *The set of universally applicable, unconditional and deterministic actions is finitely identifiable by the learning function $L_0^{effects}$, defined in the following way:*

$$L_0^{effects}(\mathcal{E}[n]) = \begin{cases} \{\langle \top\, ;\, post(P_{\text{set}(\mathcal{E}[n])}^+, P_{\text{set}(\mathcal{E}[n])}^-)\rangle\} \\ \quad \text{if for all literals } l \text{ there is } (s,t) \in \text{set}(\mathcal{E}[n]) \text{ s.t. } s \models l \text{ or } t \models l; \\ \quad \text{and for all } k < n,\ L_0^{effects}(\mathcal{E}[k]) = \uparrow \\ \uparrow \quad\quad \text{otherwise.} \end{cases}$$

$L_0^{effects}$ can be implemented using $O(|P|)$ space. If $L_0^{effects}(\mathcal{E}[n]) = a$ for some action model a then a has size $O(|P|)$.

Proof. Let α be a universally applicable, unconditional and deterministic action and let $\mathcal{E}$ be a stream for α. We need to show that $L_0^{effects}$ finitely identifies α on $\mathcal{E}$. Since α is universally applicable and $\mathcal{E}$ is for α, for every literal l, $\mathcal{E}$ must contain at least one pair (s,t), where $s \models l$. This shows that there must exist an n such that $L_0^{effects}(\mathcal{E}[n]) = \{\langle \top\, ;\, post(P_{\text{set}(\mathcal{E}[n])}^+, P_{\text{set}(\mathcal{E}[n])}^-)\rangle\}$ and such that for all literals l there is $(s,t) \in \text{set}(\mathcal{E}[n])$ with $s \models l$ or $t \models l$. Let e denote the event of $L_0^{effects}(\mathcal{E}[n])$. It now remains to be shown that $act(\{e\}) = \alpha$. Choose $e' \in h_0$, such that $act(\{e'\}) = \alpha$ (such an event must necessarily exist, cf. the proof of Theorem 4). It suffices to prove $e' = e$, that is, $post(e')(p) = post(e)(p)$ for all $p \in P$. First suppose $post(e)(p) = \top$. Then, by definition, for some $(s,t) \in \text{set}(\mathcal{E}[n])$ we have $s \models \neg p$ and $t \models p$. Since $\text{set}(\mathcal{E}[n]) \subseteq \alpha$ and $act(\{e'\}) = \alpha$, this immediately implies $post(e')(p) = \top$. A symmetric argument holds for the case of $post(e)(p) = \bot$. Now conversely assume $post(e')(p) = \top$. By choice of n, $\text{set}(\mathcal{E}[n])$ contains at least one pair (s,t) where either $s \models \neg p$ or $t \models \neg p$. Since $post(e')(p) = \top$, $act(\{e'\}) = \alpha$ and $\mathcal{E}$ is for α, there can be no pair $(s,t) \in \text{set}(\mathcal{E}[n])$ with $t \models \neg p$. Hence, $\text{set}(\mathcal{E}[n])$ must contain a pair (s,t) with $s \models \neg p$ and $t \models p$. This implies $p \in P_{\text{set}(\mathcal{E}[n])}^+$ and hence $post(e)(p) = \top$. A symmetric argument holds for the case of $post(e')(p) = \bot$. We have now shown that $post(e')(p) = post(e)(p)$ for all $p \in P$, as required.

We now turn to the space complexity. The learning function can be implemented by the following algorithm. The algorithm keeps a set P^+ of the observed positive effects, a set P^- of the observed negative effects and a set L of literals. All sets are initially empty. For each $(s,t) \in \text{set}(\mathcal{E}[n])$, the algorithm then adds the elements of $P_{(s,t)}^+$ to P^+, the elements of $P_{(s,t)}^-$ to P^-, and any literal l such that $s \models l$ or $t \models l$ is added to L. The algorithm then has to check the 'stopping condition': whether for all literals l there is $(s,t) \in \text{set}(\mathcal{E}[n])$ such that $s \models l$ or $t \models l$. This is simply a question of checking whether L contains all literals.

If the stopping condition is satisfied after receiving the last observation (and not earlier), the algorithm will return the action model $\{\langle \top\, ;\, post(P^+, P^-)\rangle\}$. It is easy to check that if this action model is returned after the nth observation, then $P^+ = P_{\text{set}(\mathcal{E}[n])}^+$ and $P^- = P_{\text{set}(\mathcal{E}[n])}^-$. The space requirement is clearly $O(|P|)$ as P^+, P^- and I are all of size $O(|P|)$. If $L_0^{effects}(\mathcal{E}[n])$ returns an

action model it will clearly have size $O(|P|)$, since it is a single event where the postcondition is of length $O(|P|)$. □

One of the crucial points about making the output of our learning functions be action models is, as earlier mentioned, that they tend to be much more succinct than the actions (state-transition functions) they represent. Any unconditional deterministic action will have size $\Theta(2^{|P|})$, since it contains exactly one pair (s, t) for each propositional state s. Proposition 3, item 4, shows that such actions can be represented using only $O(|P|)$ space (by atomic action models). The result above shows that it is even possible to learn such actions using only $O(|P|)$ space in total.

In fact, the $O(|P|)$ asymptotic upper bound on the size of the produced model guaranteed by the learning function above is worst-case optimal among *any* learning function *independent of the representation chosen* (whether it is the state-transition functions themselves, action models or a completely different formalism). To see this, note that all $3^{|P|}$ events of h_0 represent distinct unconditional deterministic actions. So any learning function for learning unconditional, deterministic actions will be able to produce at least $3^{|P|}$ different outputs. The space required to be able to represent $3^{|P|}$ different values is $\log(3^{|P|}) = |P| \log 3 = \Theta(|P|)$.

4 Learning conditional deterministic actions

Above we were concerned with learning unconditional deterministic actions. These are particularly simple as they can be represented by basic and atomic action models. We will now create a learning method for arbitrary universally applicable and deterministic actions, that is, actions that might be conditional, but are still deterministic. No such conditional action can be represented by an atomic and basic action model, which can be seen as follows. Suppose α is a universally applicable and deterministic action, and a is an atomic and basic action model with $act(a) = \alpha$. Since α is universally applicable and $act(a) = \alpha$, also a is universally applicable, by Proposition 3, item 1. Since a is then universally applicable, atomic and basic, it must necessarily be precondition-free. By Proposition 3, item 4, it follows that $act(a)$ must be unconditional. Hence if a represents α, either α is unconditional or a is not both basic and atomic. This implies that we need a more complex learning method to learn conditional actions.

We first study learning by update, following the same structure as for learning unconditional actions: we define a hypothesis space containing all the relevant events and then define the learning function via update on that hypothesis space.

As in the previous section, we assume P to be fixed. For each $s \in 2^P$ we define $\phi_s = \bigwedge_{p \in s} p \wedge \bigwedge_{p \in P-s} \neg p$.

Definition 19. *The* hypothesis space for deterministic actions *is the action model h_1 given by*

$$h_1 = \{\langle \phi_s \,;\, f\rangle \mid s \in 2^P \ and \ f : P \hookrightarrow \{\top, \bot\}$$
$$where \ f(p) \neq \top \ if \ \phi \models p \ and \ f(p) \neq \bot \ if \ \phi \models \neg p\}.$$

The last condition of the definition saying that "$f(p) \neq \top$ if $\phi \models p$ and $f(p) \neq \bot$ if $\phi \models \neg p$" simply ensures that h_1 satisfies condition 3 of being basic.

Definition 20. *The* update learning function for deterministic actions *is the learning function L_1 defined by*

$$L_1(\mathcal{E}[n]) = h_1 \mid \ \text{set}(\mathcal{E}[n]).$$

Theorem 6. *The set of universally applicable and deterministic actions is finitely identifiable by the update learning function L_1^{update}, defined in the following way*

$$L_1^{update}(\mathcal{E}[n]) = \begin{cases} L_1(\mathcal{E}[n]) & if \ L_1(\mathcal{E}[n]) \ is \ globally \ deterministic \\ & and \ for \ all \ k < n, \ L_1^{update}(\mathcal{E}[k]) = \uparrow; \\ \uparrow & otherwise. \end{cases}$$

L_1^{update} can be implemented using $O(|P| \cdot 4^{|P|})$ space. If $L_1^{update}(\mathcal{E}[n]) = a$ for some action model a then a has size $O(|P| \cdot 2^{|P|})$.

Proof. Let us take such an action α as prescribed in the theorem and let $\mathcal{E}$ be a stream for α. We need to prove that for some n, $act(L_1^{update}(\mathcal{E}[n])) = \alpha$. Take n to be the smallest such that $\alpha \subseteq \text{set}(\mathcal{E}[n])$. We will first prove $\alpha = act(L_1(\mathcal{E}[n]))$.

For $\alpha \subseteq act(L_1(\mathcal{E}[n]))$. Assume $(s, t) \in \alpha$. The hypothesis space h_1 contains the event $\langle \phi_s \,;\, f\rangle$ with $f(p) = \top$ for all $p \in t - s$ and $f(p) = \bot$ for all $p \in s - t$. Clearly, $s \otimes \langle \phi_s \,;\, f\rangle = t$. Hence $(s, t) \in act(h_1)$. We need to show that $(s, t) \in act(L_1(\mathcal{E}[n]))$, that is, that the event $\langle \phi_s \,;\, f\rangle$ is not eliminated by the stream of observations $\mathcal{E}[n]$. Note that the precondition of $\langle \phi_s \,;\, f\rangle$ is ϕ_s, so only observations of the form (s, t') can eliminate the event. Furthermore, since $s \otimes \langle \phi_s \,;\, f\rangle = t$, only observations of the form (s, t') with $t' \neq t$ can eliminate the event. However, since α is deterministic and $\mathcal{E}$ is for a, if $(s, t') \in \mathcal{E}$ then $t' = t$. *For $act(L_1(\mathcal{E}[n])) \subseteq \alpha$.* Assume $(s, t) \notin \alpha$. We then need to prove $(s, t) \notin act(L_1(\mathcal{E}[n]))$. Let $\langle \phi_s \,;\, f\rangle$ be an arbitrary event of h_1 with $t = s \otimes \langle \phi_s \,;\, f\rangle$. It suffices to prove that this event is eliminated in $L_1(\mathcal{E}[n])$. Since α is universally applicable there must be a $t' \neq t$ such that $(s, t') \in \alpha$. Since $\alpha \subseteq \text{set}(\mathcal{E}[n])$, $(s, t') \in \text{set}(\mathcal{E}[n])$. We now have $s \models \phi_s$ but $s \otimes \langle \phi_s \,;\, f\rangle \neq t'$, so $\langle \phi_s \,;\, f\rangle \notin h_1 \mid (s, t')$, and hence $\langle \phi_s \,;\, f\rangle \notin h_1 \mid \text{set}(\mathcal{E}[n])$. This shows that the required event is eliminated in $L_1(\mathcal{E}[n])$.

We have now proven $\alpha = act(L_1(\mathcal{E}[n]))$. Since α is deterministic and $\alpha = act(L_1(\mathcal{E}[n]))$, $L_1(\mathcal{E}[n])$ can not contain two distinct events of h_1 with identical preconditions. This implies that $L_1(\mathcal{E}[n])$ is globally deterministic. The only

thing left to prove is hence that the n chosen above is the *smallest* number for which $L_1(\mathcal{E}[n])$ is globally deterministic. Consider any $m < n$.

Then $\alpha - \text{set}(\mathcal{E}[m]) \neq \emptyset$, by choice of n. Choose $(s, t) \in \alpha - \text{set}(\mathcal{E}[m])$. Since $\mathcal{E}$ is sound for a and a is deterministic, there can be no pair of the form (s, t') in $\mathcal{E}[m]$. Hence, $L_1(\mathcal{E}[n])$ will contain all events from h_1 of the form $\langle \phi_s \, ; \, f \rangle$ and hence will not be globally deterministic (h_1 contains at least two such events for all non-empty P).
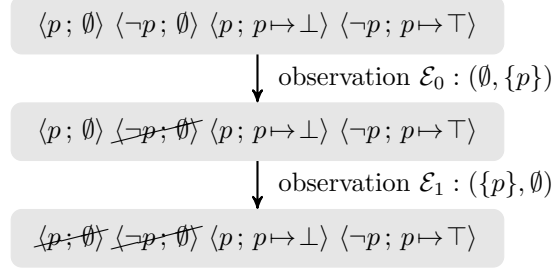
We now turn to the space complexity results. L_1^{update} is initialised with the hypothesis space h_1 of Definition 19. As for L_0^{update}, the total space requirement of the learning algorithm is the space requirement of the initial hypothesis space. Each proposition $p \in P$ can either occur positively or negatively in the precondition ϕ_s of an event $\langle \phi_s \, ; \, f \rangle$ of h_1. If it occurs positively, then either $f(p) = \bot$ or $f(p)$ is undefined, by definition of h_1. Symmetrically, if p occurs negatively in ϕ_s, then either $f(p) = \top$ or $f(p)$ is undefined. In other words, each proposition p can occur in 4 different configurations in the events of h_1. This implies that the number of events in h_1 is $O(4^{|P|})$. Since each event is of length $O(|P|)$, h_1 has size $O(|P| \cdot 4^{|P|})$, which is the total space consumption of the algorithm. If $L_1^{update}(\mathcal{E}[n]) = a$ for some action model a, then a is a globally deterministic submodel of h_1, by definition of L_1^{update}. Such a model can only have 1 event per possible precondition ϕ_s with $s \in 2^P$, hence in total $O(2^{|P|})$ events. Each event still has length $O(|P|)$, so the total size of the action model is $O(|P| \cdot 2^{|P|})$. $\qquad\square$

The learning method L_1^{update} proposed in Theorem 6 is yet another example of how learning deterministic action models can be seen as the process of gradually increasing the 'amount of determinism' in an action model. We have already made a note of it in Section 2.1. This time, however, this feature of learning becomes more pronounced, as it is explicitly present in the halting condition of the learning function L_1^{update}. Each time upon performing an update the learner checks whether the resulting restriction of the original model is globally deterministic. Once this check yields a positive result learning is concluded. Let us now present some concrete examples of the performance of L_1^{update}.

Example 5. Consider a simple scenario with a pushbutton and a light bulb. Assume there is only one proposition p: 'the light is on', and only one action: pushing the button. We assume an agent wants to learn the functioning of the pushbutton. The learner starts with the action model h_1, which in the case of $P = \{p\}$ is:
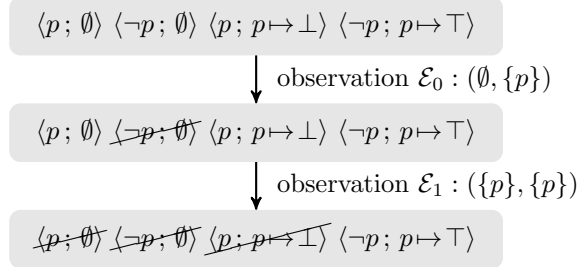
$$h_1 = \{\langle p \, ; \, \emptyset \rangle, \langle \neg p \, ; \, \emptyset \rangle, \langle p \, ; \, p \mapsto \bot \rangle, \langle \neg p \, ; \, p \mapsto \top \rangle\}$$

Assume the first two observations the learner receives (the first elements of a stream $\mathcal{E}$) are $(\emptyset, \{p\})$ and $(\{p\}, \emptyset)$. This corresponds to a pushbutton that turns the light on if it is currently off, and vice versa. The learner revises her model in the following way:

$$\langle p\,;\,\emptyset\rangle\ \langle\neg p\,;\,\emptyset\rangle\ \langle p\,;\,p\mapsto\bot\rangle\ \langle\neg p\,;\,p\mapsto\top\rangle$$

$$\downarrow\ \text{observation}\ \mathcal{E}_0 : (\emptyset,\{p\})$$

$$\langle p\,;\,\emptyset\rangle\ \langle\cancel{\neg p\,;\,\emptyset}\rangle\ \langle p\,;\,p\mapsto\bot\rangle\ \langle\neg p\,;\,p\mapsto\top\rangle$$

$$\downarrow\ \text{observation}\ \mathcal{E}_1 : (\{p\},\emptyset)$$

$$\langle\cancel{p\,;\,\emptyset}\rangle\ \langle\cancel{\neg p\,;\,\emptyset}\rangle\ \langle p\,;\,p\mapsto\bot\rangle\ \langle\neg p\,;\,p\mapsto\top\rangle$$

Now the agent has reached a globally deterministic action model, and can hence report it to be the correct model of the action. Note that the two observations correspond to first pushing the button when the light is off ($\mathcal{E}_0$), and afterwards pushing the button again after the light has come on ($\mathcal{E}_1$). These two observations are sufficient to learn the type of the pushbutton.

Consider now another stream $\mathcal{E}'$, for a different action where the first two elements are $(\emptyset,\{p\})$ and $(\{p\},\{p\})$. This time the pushbutton unconditionally turns on the light. The learner reaches a globally deterministic action model in two steps, this time an atomic one (which is possible since the action is unconditional).

$$\langle p\,;\,\emptyset\rangle\ \langle\neg p\,;\,\emptyset\rangle\ \langle p\,;\,p\mapsto\bot\rangle\ \langle\neg p\,;\,p\mapsto\top\rangle$$

$$\downarrow\ \text{observation}\ \mathcal{E}_0 : (\emptyset,\{p\})$$

$$\langle p\,;\,\emptyset\rangle\ \langle\cancel{\neg p\,;\,\emptyset}\rangle\ \langle p\,;\,p\mapsto\bot\rangle\ \langle\neg p\,;\,p\mapsto\top\rangle$$

$$\downarrow\ \text{observation}\ \mathcal{E}_1 : (\{p\},\{p\})$$

$$\langle\cancel{p\,;\,\emptyset}\rangle\ \langle\cancel{\neg p\,;\,\emptyset}\rangle\ \langle\cancel{p\,;\,p\mapsto\bot}\rangle\ \langle\neg p\,;\,p\mapsto\top\rangle$$

4.1 Improved learning of conditional deterministic actions

As for unconditional actions, we can improve the space complexity by keeping track of observed positive and negative effects rather than doing simple update learning. However, since actions are potentially conditional, we need to keep track of the possibility of distinct effects in distinct states.

In the result below, recall that we have defined $post(P^+, P^-) = \{p \mapsto \top \mid p \in P^+\} \cup \{p \mapsto \bot \mid p \in P^-\}$.

Theorem 7. *The set of universally applicable and deterministic actions is*

finitely identifiable by the learning function $L_1^{effects}$, defined in the following way:

$$L_1^{effects}(\mathcal{E}[n]) = \begin{cases} \{\langle \phi_s \,;\, post(P_{(s,t)}^+, P_{(s,t)}^-)\rangle \mid (s,t) \in \mathrm{set}(\mathcal{E}[n])\} \\ \quad \text{if for all states } s \in 2^P \text{ there is } (s,t) \in \mathrm{set}(\mathcal{E}[n]); \\ \quad \text{and for all } k < n, \, L_1^{effects}(\mathcal{E}[k]) = \,\uparrow \\ \uparrow \quad \text{otherwise.} \end{cases}$$

$L_1^{effects}$ can be implemented using $O(|P| \cdot 2^{|P|})$ space. If $L_1^{effects}(\mathcal{E}[n]) = \alpha$ for some action model a then a has size $O(|P| \cdot 2^{|P|})$.

Proof. Let α be as prescribed and let $\mathcal{E}$ be a stream for α. Since α is deterministic and universally applicable, $\mathrm{set}(\mathcal{E})$ will contain exactly one pair of the form (s,t) for each $s \in 2^P$. Choose the smallest n so that also $\mathrm{set}(\mathcal{E}[n])$ has this property. Then we must have $\alpha = \mathrm{set}(\mathcal{E}[n])$ due to determinism of α. By definition of the learning function we then also have $L_1^{effects}(\mathcal{E}[n]) = \{\langle \phi_s \,;\, post(P_{(s,t)}^+, P_{(s,t)}^-)\rangle \mid (s,t) \in \alpha\}$. We need to prove $\alpha = act(L_1^{effects}(\mathcal{E}[n]))$. To prove $\alpha \subseteq act(L_1^{effects}(\mathcal{E}[n]))$ it suffices to show that for all $(s,t) \in \alpha$, $t = s \otimes \langle \phi_s \,;\, post(P_{(s,t)}^+, P_{(s,t)}^-)\rangle$. This is trivial given the definitions of $P_{(s,t)}^+$ and $P_{(s,t)}^-$. For $act(L_1^{effects}(\mathcal{E}[n])) \subseteq \alpha$, we have to prove that if $t' = s' \otimes \langle \phi_s \,;\, post(P_{(s,t)}^+, P_{(s,t)}^-)\rangle$ for some pair (s',t') and some choice of $(s,t) \in \alpha$ then $(s',t') \in \alpha$. From $t' = s' \otimes \langle \phi_s \,;\, post(P_{(s,t)}^+, P_{(s,t)}^-)\rangle$ we immediately get $s' = s$. We now have $t = (s - P_{(s,t)}^-) \cup P_{(s,t)}^+ = s \otimes \langle \phi_s \,;\, post(P_{(s,t)}^+, P_{(s,t)}^-)\rangle = s' \otimes \langle \phi_s \,;\, post(P_{(s,t)}^+, P_{(s,t)}^-)\rangle = t'$. This shows $(s',t') = (s,t) \in \alpha$.

We now turn to the complexity results. The learning function can be implemented by the following algorithm. For each $s \in 2^P$, we store a boolean value b_s, and two sets $P_s^+, P_s^- \subseteq P$. Initially $b_s = 0$ and $P_s^+ = P_s^- = \emptyset$ for all s. For each observation (s,t), the algorithm then does the following: if $b_s = 0$ then we assign $b_s := 1$, $P_s^+ := P_{(s,t)}^+$, $P_s^- := P_{(s,t)}^-$. After each observation, the algorithm checks whether $b_s = 1$ for all $s \in 2^P$. If so, the action model $\{\langle \phi_s \,;\, post(P_s^+, P_s^-)\rangle \mid s \in 2^P\}$ is returned. It is easy to check that this indeed implements $L_1^{effects}$. Since the algorithm for each $s \in 2^P$ stores a boolean and two subsets of P, the space requirement is $O(|P| \cdot 2^{|P|})$. The action model returned contains for each $s \in 2^P$ an event of length $O(|P|)$, so it also has size $O(|P| \cdot 2^{|P|})$. $\square$

As for learning unconditional actions, we can prove that the size of the produced model of the learning function above is worst-case optimal, again independent of the action representation chosen. First we note that any deterministic, universally applicable action α determines a unique mapping $f_\alpha : 2^P \to 2^P$ satisfying $(s,t) \in \alpha$ iff $f_\alpha(s) = t$. Conversely, any such mapping determines a unique deterministic, universally applicable action. Hence the number of deterministic, universally applicable actions is equal to the number of such mappings, which is $(2^{|P|})^{(2^{|P|})}$. Thus any learning function for learning such actions will be able to produce $(2^{|P|})^{(2^{|P|})}$ different outputs.

The space requirement to be able to represent $(2^{|P|})^{(2^{|P|})}$ different values is $\log((2^{|P|})^{(2^{|P|})}) = 2^{|P|} \cdot \log(2^{|P|}) = 2^{|P|} \cdot |P| \log 2 = 2^{|P|} \cdot |P|$, which is the space requirement guaranteed by the learning function above.

4.2 Parametrised learning of conditional deterministic actions

The above results study worst-case space complexities in terms of the number of atomic propositions. In some environments, the set of atomic propositions might be quite high, for instance the environment of a domestic robot. Still, most individual actions α in such environments only depend on relatively few propositions (have a small $pre(\alpha)$). For instance, the action α of pushing a particular light switch might have $pre(\alpha) = \{p\}$, where p represents the current state of the switch/light. Of course, there could be more preconditions in $pre(\alpha)$ encoding whether the bulb is broken, whether the fuse is blown, etc., but the size of $pre(\alpha)$ would still be very low compared to potentially 100s or 1000s or atomic propositions in the domain. We will now present an improved learning function that takes this into account. The learning function is parametrised by an upper bound j on the size of $pre(\alpha)$ (that is, the number of preconditions is at most j). In many domains, it is reasonable to assume a fixed upper bound on the number of preconditions for all actions in the domain (the outcome of any action can only depend on the truth value of a given number of propositions).

Given an action α and a propositional formula ϕ, we use $\alpha {\restriction} \phi$ to denote the restriction of α to the states satisfying ϕ, i.e., $\alpha {\restriction} \phi = \{(s,t) \in \alpha \mid s \models \phi\}$. For all $j \leq |P|$, we define $\Phi_j = \{\bigwedge_{p \in s} p \wedge \bigwedge_{p \in P'-s} \neg p \mid P' \subseteq P, |P'| = j, \text{ and } s \in 2^{P'}\}$. The elements of Φ_j are conjunctions of exactly j literals.

Two state-transition pairs (s,t) and (s',t') are called *compatible* if the following conditions hold for all $p \in P$: if $p \in P^+_{(s,t)}$, then $t' \models p$; if $p \in P^-_{(s,t)}$, then $t' \models \neg p$; if $p \in P^+_{(s',t')}$, then $t \models p$; and if $p \in P^-_{(s',t')}$, then $t \models \neg p$. It is clear from this definition that if two pairs (s,t) and (s',t') are incompatible, there can be no single event e with $t = s \otimes e$ and $t' = s' \otimes e$. Compatibility between (s,t) and (s',t') can equivalently be defined as the condition $((t-s)-t') \cup ((s-t) \cap t') \cup ((t'-s')-t) \cup ((s'-t') \cap t) = \emptyset$.

Theorem 8. *Let $\mathcal{X}_j$ denote the set of universally applicable and deterministic actions α satisfying $|pre(\alpha)| \leq j$. The set $\mathcal{X}_j$ is finitely identifiable by the learning function $L_2^{effects}$, defined in the following way:*

$$L_2^{effects}(\mathcal{E}[n]) = \begin{cases} \{\langle \phi \,;\, post(P^+_{set(\mathcal{E}[n]){\restriction}\phi}, P^-_{set(\mathcal{E}[n]){\restriction}\phi}) \rangle \mid \phi \in \Phi_j \text{ and} \\ \quad all \ (s,t), (s',t') \in set(\mathcal{E}[n]){\restriction}\phi \ are \ compatible\} \\ \qquad if \ for \ all \ \phi \in \Phi_{\min\{|P|,2j+1\}} \ there \ is \\ \qquad (s,t) \in set(\mathcal{E}[n]), \ s.t. \ s \models \phi, \\ \qquad and \ for \ all \ m < n, \ L_1^{effects}(\mathcal{E}[m]) = \uparrow; \\ \uparrow \qquad otherwise. \end{cases}$$

$L_2^{effects}$ can be implemented using $O((\binom{|P|}{\min\{|P|,2j+1\}}) \cdot 2^{\min\{|P|,2j+1\}} + \binom{|P|}{j} \cdot 2^j \cdot |P|)$ space. If $L_2^{effects}(\mathcal{E}[n]) = \alpha$ for an action model a then a has size $O((\binom{|P|}{j}) \cdot 2^j \cdot |P|)$.

Proof. Let α be as prescribed in the theorem and let $\mathcal{E}$ be a stream for α. Since α is universally applicable there exists an n such that:

$$L_2^{effects}(\mathcal{E}[n]) = \{\langle\phi\,;\,post(P^+_{\text{set}(\mathcal{E}[n])\restriction\phi}, P^-_{\text{set}(\mathcal{E}[n])\restriction\phi})\rangle \mid \phi \in \Phi_j \text{ and}$$
$$\text{all } (s,t),(s',t') \in \text{set}(\mathcal{E}[n])\restriction\phi \text{ are compatible}\},$$
$$\text{and for all } \phi \in \Phi_{2j+1} \text{ there is } (s,t) \in \text{set}(\mathcal{E}[n]), \text{ s.t. } s \models \phi.$$

We need to prove $act(L_2^{effects}(\mathcal{E}[n])) = \alpha$.

For $\alpha \subseteq act(L_2^{effects}(\mathcal{E}[n]))$. Since $|pre(\alpha)| \leq j$, there must be a set $P' \subseteq P$ satisfying $|P'| = j$ and $pre(\alpha) \subseteq P'$. Then α is uniform in $P - P'$. Assume $(s,t) \in \alpha$. We need to prove $(s,t) \in act(L_2^{effects}(\mathcal{E}[n]))$. By uniformity of α in $P - P'$, there exists P^+ and P^-, such that for all s' with $s' \ominus s \subseteq P - P'$, $(s, (s - P^-) \cup P^+) \in \alpha$.

Let $\phi = \bigwedge_{p \in s \cap P'} p \wedge \bigwedge_{p \in P' - s} \neg p$. Clearly, $s \models \phi$. Note that for any s' with $s' \models \phi$, we have $s' \ominus s \subseteq P - P'$ and hence $(s', (s' - P^-) \cup P^+) \in \alpha$. We then get that any two pairs $(s',t'),(s'',t'') \in \alpha\restriction\phi$ must be compatible, and hence that any two pairs $(s',t'),(s'',t'') \in \text{set}(\mathcal{E}[n])\restriction\phi$ are also compatible.

Since $\phi \in \Phi_j$, we then get that $L_2^{effects}(\mathcal{E}[n]))$ contains the event $e_\phi = \langle\phi\,;\,post(P^+_{\text{set}(\mathcal{E}[n])\restriction\phi}, P^-_{\text{set}(\mathcal{E}[n])\restriction\phi})\rangle$. Since $s \models \phi$ and $pre(e_\phi) = \phi$, we get $s \models pre(e_\phi)$, and hence $(s, s \otimes e_\phi) \in L_2^{effects}(\mathcal{E}[n])$. To prove $(s,t) \in L_2^{effects}(\mathcal{E}[n])$ it therefore suffices to show that $P^+_{\text{set}(\mathcal{E}[n])\restriction\phi} - s = P^+ - s$ and $P^-_{\text{set}(\mathcal{E}[n])\restriction\phi} - s = P^- - s$. We only prove $P^+_{\text{set}(\mathcal{E}[n])\restriction\phi} - s = P^+ - s$, the other case being analogous. Assume first $p \in P^+ - s$. Since $s \models \phi$ and $p \notin s$, either $\neg p$ is a conjunct of ϕ or p does not occur in ϕ. Since $\phi \in \Phi_j$, in both cases there exists a $\phi' \in \Phi_{\min\{|P|,2j+1\}}$ such that $\phi' \models \phi \wedge \neg p$. By choice of n there then exists $(s',t') \in \text{set}(\mathcal{E}[n])$ with $s' \models \phi \wedge \neg p$. Since $\text{set}(\mathcal{E}[n]) \subseteq \alpha$, we have $(s',t') \in \alpha$, and since $s' \models \phi$ we then get $t' = (s' - P^-) \cup P^+$. Since $p \in P^+$ this implies $t' \models p$. We now have $s' \models \phi$, $s' \models \neg p$, $t' \models p$ and $(s',t') \in \text{set}(\mathcal{E}[n])$. This implies $p \in P^+_{\text{set}(\mathcal{E}[n])\restriction\phi}$, as required.

Now suppose opposite, that $p \in P^+_{\text{set}(\mathcal{E}[n])\restriction\phi} - s$. Then by definition there must exist $(s',t') \in \text{set}(\mathcal{E}[n])\restriction\phi$ such that $s' \models \neg p$ and $t' \models p$. Since $\text{set}(\mathcal{E}[n]) \subseteq \alpha$, we get $(s',t') \in \alpha$, and since $s' \models \phi$, we get $t' = (s' - P^-) \cup P^+$. Since $s' \models \neg p$ and $t' \models p$, necessarily $p \in P^+$.

For $act(L_2^{effects}(\mathcal{E}[n])) \subseteq \alpha$. Suppose, to achieve a contradiction, that it does not hold. Then there must be a pair $(s,t) \in act(L_2^{effects}(\mathcal{E}[n])) - \alpha$. Since α is universally applicable, for some t' we have $(s,t') \in \alpha$. Since $(s,t) \notin \alpha$, $t' \neq t$. Hence there exists a $p \in P$ with $p \in t' \ominus t$. We can assume $t \models p$ and $t' \models \neg p$, the other case being symmetric. We either have $s \models p$ or $s \models \neg p$. We can assume $s \models \neg p$, again since the other case is symmetric. Then $p \in P^+_{(s,t)}$. Since $(s,t),(s,t') \in act(L_2^{effects}(\mathcal{E}[n]))$ there must exist formulas $\phi, \psi \in \Phi_j$ such that $e_\phi = \langle\phi\,;\,post(P^+_{\text{set}(\mathcal{E}[n])\restriction\phi}, P^-_{\text{set}(\mathcal{E}[n])\restriction\phi})\rangle$ and

$e_\psi = \langle \psi \, ; \, post(P^+_{\mathrm{set}(\mathcal{E}[n]) \restriction \psi}, P^-_{\mathrm{set}(\mathcal{E}[n]) \restriction \psi}) \rangle$ are events of $L_2^{effects}(\mathcal{E}[n])$ and $t = s \otimes e_\phi$ and $t' = s \otimes e_\psi$.

Since $\phi, \psi \in \Phi_j$, there exists $\gamma \in \Phi_{\min\{|P|, 2j+1\}}$ with $\gamma \models \phi \wedge \psi \wedge \neg p$.

Hence by choice of n there exists $(s'', t'') \in \mathrm{set}(\mathcal{E}[n])$ with $s'' \models \gamma$. Now we have $(s,t), (s'', t'') \in \mathrm{set}(\mathcal{E}[n]) \restriction \phi$ and $(s, t'), (s'', t'') \in \mathrm{set}(\mathcal{E}[n]) \restriction \psi$. If $t'' \models p$ then $p \in P^+_{(s'', t'')}$ and since $t' \models \neg p$, the two observations $(s, t'), (s'', t'')$ of $\mathrm{set}(\mathcal{E}[n]) \restriction \psi$ are incompatible, contradicting that e_ψ is an event of $L_2^{effects}(\mathcal{E}[n])$. If $t'' \models \neg p$ then since $p \in P^+_{(s,t)}$ the two observations $(s, t), (s'', t'')$ of $\mathrm{set}(\mathcal{E}[n]) \restriction \phi$ are incompatible, contradicting that e_ϕ is an event of $L_2^{effects}(\mathcal{E}[n])$.

We now turn to the complexity claims. The learning function can be implemented by the following algorithm. For each $\phi \in \Phi_{\min\{|P|, 2j+1\}}$ the algorithm stores a boolean b_ϕ^{seen} which is initially 0. If an observation (s, t) with $s \models \phi$ is received, we assign $b_\phi^{seen} := 1$. The learning function additionally for each $\phi \in \Phi_j$ keeps track of the following information. First, there is a boolean $b_\phi^{include}$ which is initially 1, and which encodes whether the resulting action model should include the event with precondition ϕ. Second, for each literal l there is a boolean $b_{\phi, l}^+$ recording whether an observation (s, t) with $s \models \phi$, $s \models \neg l$ and $t \models l$ has been made. Third, there is a boolean $b_{\phi, l}^-$ recording whether an observation (s, t) with $s \models \phi$, $s \models l$ and $t \models l$ has been made. With these booleans we can keep track of whether all observations $(s, t), (s', t')$ with $s \models \phi$ and $s' \models \phi$ are compatible. If an observation (s, t) with $s \models \phi$ is made that is incompatible with the earlier observations, we set $b_\phi^{include} = 0$.

After each observation, it is checked whether all $b_\phi^{seen} = 1$. If so, we return the action model that for each $\phi \in \Phi_j$ with $b_\phi^{include} = 1$ contains the event $\langle \phi \, ; \, post(P_\phi^+, P_\phi^-) \rangle$ having $P_\phi^+ = \{p \in P \mid b_{\phi, p}^+ = 1\}$ and $P_\phi^- = \{p \in P \mid b_{\phi, \neg p}^+ = 1\}$. To store the booleans b_ϕ^{seen} we need as many bits as the size of $\Phi_{\min\{|P|, 2j+1\}}$. The set $\Phi_{\min\{|P|, 2j+1\}}$ contains conjunctions of $\min\{|P|, 2j+1\}$ literals from P. There are $\binom{|P|}{\min\{|P|, 2j+1\}}$ ways to choose $\min\{|P|, 2j+1\}$ distinct propositions from P, and each proposition can then either occur positively or negatively. This gives that the size of $\Phi_{\min\{|P|, 2j+1\}}$ is $\binom{|P|}{\min\{|P|, 2j+1\}} \cdot 2^{\min\{|P|, 2j+1\}}$. Additionally, we are for each $\phi \in \Phi_j$ storing a boolean $b_\phi^{include}$, and for each combination of $\phi \in \Phi_j$ and literal l we are storing 2 additional booleans $b_{\phi, l}^+$ and $b_{\phi, l}^-$. The size of Φ_j is $\binom{|P|}{j} \cdot 2^j$. The number of literals is $O(|P|)$. Hence we need additionally $O(\binom{|P|}{j} \cdot 2^j \cdot |P|)$ bits. This gives the result on the space consumption of the algorithm. The produced action model has an event of length $O(|P|)$ for at most each $\phi \in \Phi_j$, so the size of this model is $O(\binom{|P|}{j} \cdot 2^j \cdot |P|)$. $\qquad \square$

We note the following interesting special cases of the space complexity of the produced action models. Unconditional actions have $j = 0$. For $j = 0$ we get $O(\binom{|P|}{j} \cdot 2^j \cdot |P|) = O(|P|)$, which is exactly the result on the size of the produced action model for unconditional actions we achieved in Theorem 5. For conditional actions in general (with no restrictions on the preconditions) we have $j = |P|$. Then we get $O(\binom{|P|}{j} \cdot 2^j \cdot |P|) = O(2^{|P|} \cdot |P|)$, which is exactly

the result achieved in Theorem 7. For the special case of unary preconditions, $j = 1$, we get $O(\binom{|P|}{j} \cdot 2^j \cdot |P|) = O(|P|^2)$.

5 Conclusions

In this paper we studied the problem of learnability of action models in dynamic epistemic logic. We provided an extensional treatment of actions viewed as sets of transitions between propositional states. This approach is especially useful for our learnability framework: we can relate the observations of action executions to the concise representations of actions in dynamic epistemic logic. We studied fully observable propositional action models with respect to conclusive (finite identifiability) and inconclusive (identifiability in the limit) learnability. Apart from the general learnability results, we introduced learning functions which proceed via gradual restriction of action models. Here, by implementing the update method (commonly used in dynamic epistemic logic, in a different context), we demonstrated how the learning of action models can be seen as transitioning from nondeterministic to deterministic actions.

Related work A similar qualitative approach to learning actions has been addressed by [25] within the STRIPS planning formalism. The STRIPS setting is more general than ours in that it uses atoms of first-order predicate logic for pre- and postconditions. It is however less general in neglecting various aspects of actions which we have successfully treated in this paper, e.g., negative preconditions, negative postconditions, and conditional actions (actions with conditional effects). We believe that our framework can be applied to generalise the results of [25] to richer planning frameworks allowing such action types. Even though some of the previous work uses the basic mechanisms of update learning (SLAFS learning [23] and learning within the STRIPS formalism [25]) it rarely goes beyond basic update, as we do here with the effect learning. There has been quite substantial amount of work in relating dynamic epistemic logic and learning theory (see [15, 16] for overviews), where iterated update and upgrade revision policies are treated as long term learning methods, where learning is seen as convergence to certain types of knowledge (see [3, 5]). A study of abstract properties of finite identifiability in a setting similar to ours, including various efficiency considerations, can be found in [17].

Future work In this paper we laid the groundwork for our subsequent studies of learnability of action models. We only considered fully observable actions models, and hence did not use the full expressive power of the DEL-formalism, which offers a principled way of describing actions in a logical setting, and opens ways to various extensions. Those include: non-deterministic, partially observable, and multi-agent action models.

Non-deterministic action models are more difficult to learn via update methods. It is so because an observed outcome of an execution of an action in a given propositional state does not allow excluding the possibility that at a different

point in time the execution of the action in the same propositional state will yield a different result. As described earlier, partially observable actions are not learnable in the strict sense considered above, but we can still investigate agents learning "as much as possible" given their limitations in observability. The multi-agent case is particularly interesting due to the possibility of agents with varied limitations on observability, and the possibility of communication within the learning process. Furthermore, we here considered only what we call *reactive learning*: the learner has no influence over which observations are received. Another direction is that of *proactive learning*, where the learner gets to choose which actions to execute. This is probably the most relevant type of learning for a general learning-and-planning agent. In this context, we also plan to focus on *consecutive streams*: streams corresponding to executing sequences of actions rather than observing arbitrary state transitions. Our ultimate aim is to relate learning and planning within the framework of DEL. Those two cognitive capabilities are now investigated mostly in separation—our goal is to bridge them.

References

[1] M. B. Andersen, T. Bolander, and M. H. Jensen. Conditional epistemic planning. *Lecture Notes in Artificial Intelligence*, 7519:94–106, 2012. Proceedings of JELIA 2012.

[2] D. Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45(2):117–135, 1980.

[3] A. Baltag, N. Gierasimczuk, and S. Smets. Belief revision as a truth-tracking process. In K. Apt, editor, *TARK'11: Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge, Groningen, The Netherlands, July 12-14, 2011*, pages 187–190. ACM, New York, NY, USA, 2011.

[4] A. Baltag, N. Gierasimczuk, and S. Smets. Truth tracking by belief revision. ILLC Prepublication Series PP-2014-20 (to appear in Studia Logica 2017), 2014.

[5] A. Baltag, N. Gierasimczuk, and S. Smets. On the solvability of inductive problems: A study in epistemic topology. In R. Ramanujam, editor, *Proceedings of the 15th conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, 2015. Also available as a technical report in ILLC Prepublication Series PP-2015-13.

[6] A. Baltag, L. S. Moss, and S. Solecki. The logic of public announcements and common knowledge and private suspicions. In I. Gilboa, editor, *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-98)*, pages 43–56. Morgan Kaufmann, 1998.

[7] C. Baral, G. Gelfond, E. Pontelli, and T. C. Son. An action language for reasoning about beliefs in multi-agent domains. In *Proceedings of the 14th International Workshop on Non-Monotonic Reasoning*, volume 4, 2012.

[8] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 2001.

[9] T. Bolander and M. B. Andersen. Epistemic planning for single- and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21:9–34, 2011.

[10] T. Bolander and N. Gierasimczuk. Learning action models: Qualitative approach. In *Logic, Rationality and Interaction*, volume 9394 of *Lecture Notes in Computer Science*. Springer, 2015.

[11] R. Fikes and N. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–203, 1971.

[12] M. Ghallab, D. S. Nau, and P. Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.

[13] N. Gierasimczuk. Bridging learning theory and dynamic epistemic logic. *Synthese*, 169(2):371–384, 2009.

[14] N. Gierasimczuk. Learning by erasing in dynamic epistemic logic. In A. H. Dediu, A. M. Ionescu, and C. Martin-Vide, editors, *LATA'09: Proceedings of 3rd International Conference on Language and Automata Theory and Applications, Tarragona, Spain, April 2-8, 2009*, volume 5457 of *Lecture Notes in Computer Science*, pages 362–373. Springer, The Netherlands, 2009.

[15] N. Gierasimczuk. *Knowing One's Limits. Logical Analysis of Inductive Inference*. PhD thesis, Universiteit van Amsterdam, The Netherlands, 2010.

[16] N. Gierasimczuk, D. de Jongh, and V. F. Hendricks. Logic and learning. In A. Baltag and S. Smets, editors, *Johan van Benthem on Logical and Informational Dynamics*. Springer, 2014.

[17] N. Gierasimczuk and D. de Jongh. On the complexity of conclusive update. *The Computer Journal*, 56(3):365–377, 2013.

[18] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[19] K. T. Kelly. The learning power of belief revision. In *TARK'98: Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 111–124, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[20] S. Lange and T. Zeugmann. Types of monotonic language learning and their characterization. In *COLT'92: Proceedings of the 5th Annual ACM Conference on Computational Learning Theory, Pittsburgh, PA, USA, July 27-29, 1992*, pages 377–390. ACM, New York, NY, USA, 1992.

[21] Y. Mukouchi. Characterization of finite identification. In K. Jantke, editor, *AII'92: Proceedings of the International Workshop on Analogical and Inductive Inference, Dagstuhl Castle, Germany, October 5-9, 1992*, volume 642 of *Lecture Notes in Computer Science*, pages 260–267. Springer, Berlin/Heidelberg, 1992.

[22] J. Plaza. Logics of public communications. *Synthese*, 158(2):165–179, 2007.

[23] D. Shahaf and E. Amir. Learning partially observable action schemas. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 913–919. AAAI Press, 2006.

[24] H. van Ditmarsch and B. Kooi. Semantic results for ontic and epistemic change. In G. Bonanno, W. van der Hoek, and M. Wooldridge, editors, *Logic and the Foundation of Game and Decision Theory (LOFT 7)*, Texts in Logic and Games 3, pages 87–117. Amsterdam University Press, 2008.

[25] T. J. Walsh and M. L. Littman. Efficient learning of action schemas and web-service descriptions. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 714–719. AAAI Press, 2008.