Information and Computation 255 (2017) 243–261

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

The modal logic of copy and remove

Carlos Areces [a,*], Hans van Ditmarsch [b,c], Raul Fervari [a], François Schwarzentruber [d]

[a] FaMAF, Universidad Nacional de Córdoba & CONICET, Argentina
[b] LORIA, CNRS, Université de Lorraine, France
[c] IMSc, Chennai, India
[d] ENS Rennes, France

ARTICLE INFO

Article history:
Received 18 December 2014
Available online 9 January 2017

Keywords:
Modal logic
Dynamic epistemic logic
Complexity
Expressivity

ABSTRACT

We propose a logic with the dynamic modal operators copy and remove. The copy operator replicates a given model, and the remove operator removes paths in a given model. We show that the product update by an action model in dynamic epistemic logic decomposes in copy and remove operations, when we consider action models with Boolean preconditions and no post-condition. We also show that copy and remove operators with paths of length 1 can be expressed by action models with post-conditions. We investigate the expressive power of the logic with copy and remove operations, together with the complexity of the satisfiability problem of some of its syntactic fragments.

1. Introduction

In modal logics we usually interpret a modal operator by way of an accessibility relation in a given Kripke model. Over the past decades logics have been proposed in which the modality is, instead, interpreted by a transformation of the model. In such logics the modality can be seen as interpreted by a binary relation between pointed models, where the first argument is the initial pointed model and the second argument of the relation is the transformed model. We could mention sabotage logic here [1], wherein states or arrows are deleted from a model. Or we could mention dynamic epistemic logics [2] that proposes model-changing operators to model change of knowledge or belief. In [3–6] a new line of contributions to model-transforming logics, motivated by Van Benthem's sabotage logic is developed. In this article we advance that last line of work, while linking it to dynamic epistemic logics.

Action model logic ($\mathcal{AML}$) [7] is a well-known dynamic epistemic logic to model information change. $\mathcal{AML}$ is an extension of basic epistemic logic with a dynamic modal operator for the execution of so-called epistemic actions. This operator is parameterized by an action model, a semantic object which typically models a multi-agent information changing scenario. Action models are treated as syntactic objects in modal operators. Action models are complex structures, and the logic has high computational complexity: deciding model checking is PSPACE-complete, while deciding satisfiability is NEXPTIME-complete [8].

In this article we propose modal logics with primitive actions called copy and remove. We investigate some of their model theoretic properties and their complexity, and, as an example of what one can do with such logics, we give an

* Corresponding author.
E-mail addresses: areces@famaf.unc.edu.ar (C. Areces), hans.van-ditmarsch@loria.fr (H. van Ditmarsch), fervari@famaf.unc.edu.ar (R. Fervari), francois.schwarzentruber@ens-rennes.fr (F. Schwarzentruber).

http://dx.doi.org/10.1016/j.ic.2017.01.004
0890-5401/

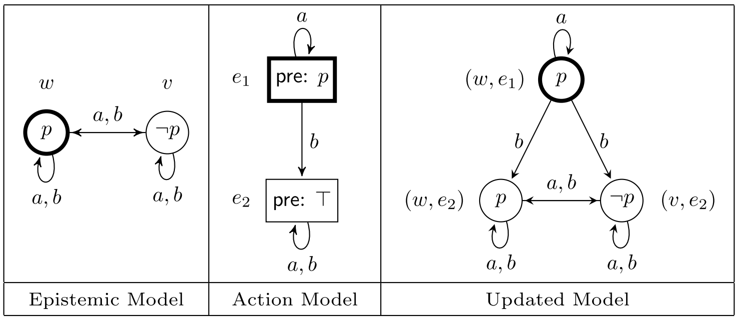

Fig. 1. Agent a privately learns that p.

embedding of action model logic into our logic: we show that every action model with propositional pre-conditions can be simulated by a combination of the copy and remove operators. The remove operator we propose is akin to the generalized arrow updates introduced in [9,10].

In Fig. 1 we show an epistemic model, an action model, and the result of executing that action model in that epistemic model. The epistemic model represents that agents a and b are uncertain whether an atomic proposition p is true (and that they have common knowledge of that uncertainty). The actual world, or designated state, of the model is where p is true (shown with a thick circle in the figure). The action model represents that agent a learns that p is true, whereas agent b (incorrectly) believes that nothing happens—of which a is aware. In short: a privately learns that p. In action models, there are no valuations of propositional variables; they are replaced by pre-conditions, in this case p and $\top$ (the formula that is always true). Action models update epistemic models by mean an operation called a restricted modal product: the domain is limited to the state-action pairs where the pre-conditions of the actions hold. Therefore in our example there are only three (and not four) pairs in the updated model: the pair (v, e_1) is missing as the pre-condition of e_1 (the formula p) is not true in v. The accessibility relation in the restricted product is updated according to the following rule: there is a (labeled) arrow between two state-action pairs if there was such an arrow linking both the first arguments in the epistemic model and the second arguments in the action model. One can now verify that in the resulting model a knows that p (there is only an a-arrow from w to itself), whereas b still believes that both a and b are ignorant about p.

By means of the copy and remove actions of the logics that we propose, we can alternatively describe this scenario. This is depicted in Fig. 2. First, we replicate the original epistemic model as many times as there are actions in the action model (twice in this case). We identify each copy with a (fresh) propositional variable corresponding to an action in the action model (e.g., p_{e_1} corresponds to e_1). Thus we obtain the second model in Fig. 2. Then, we first remove all arrows that point to state-action alternatives wherein the action cannot be executed in the state. Finally, between the remaining state-action pairs we remove all arrows that are ruled out according to the accessibility relation in the action model. Thus we obtain the updated model at the bottom of the Fig. 2.

The copy and remove operators were first introduced in [11]. In this article, we extend the results introduced in that article: we discuss in detail the expressive power of the logic with copy and remove operators; we introduce examples which motivate our work; and we provide proofs of all results.

The article is organized as follows. In Section 2 we start by introducing the formal definition of action model logic to make the article self-contained, together with an example which motivates the use of the fragment with only Boolean pre-conditions. We introduce the action models without post-conditions, and its extension with also post-conditions (which will be useful to encode copy and remove operations). We will actually work with action models without post-conditions, and we will explicitly indicate when we move to the setting with post-conditions. In Section 3 we introduce $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$, the logic with the two dynamic primitives copy and remove which captures the behavior of action models. In Section 4 we introduce bisimulations to investigate its expressive power, and we show that the logic with copy and remove has the *tree model property*. In Section 5 we define an equivalence preserving translation from action model logic to a fragment of $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$. We also show that it is possible to find action models with post-conditions that encode copy and remove actions with paths of length 1, which give us a method to decompose action models with Boolean pre-conditions. Finally, we show complexity results of the satisfiability problem for different fragments of $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$ in Section 6. We prove that the logic with copy and without remove is PSPACE-complete; the same result is proved for the logic with removes of length 1 and without copy. When we consider unrestricted removes, we show that the logic is decidable. For the logic with copy and remove of length 1 we prove NEXPTIME-completeness. The problem of determining the complexity of the full logic $\mathcal{ML}(\mathsf{cp}, \mathsf{rm})$ is open.

2. Action model logic with Boolean pre-conditions

One of the main results in this paper is to show that we can capture the action model logic $\mathcal{AML}$ with simpler primitives. Let us start by introducing formally the logic $\mathcal{AML}$. First, we introduce action models.