On modularity of theories

Andreas Herzig Ivan Varzinczak*
Institut de Recherche en Informatique de Toulouse (IRIT)
118 route de Narbonne
F-31062 Toulouse Cedex 4 (France)
e-mail: {herzig,ivan}@irit.fr
http://www.irit.fr/recherches/LILAC

Keywords: modularity, interpolation

1 Introduction

In many cases knowledge is represented by logical theories containing multiple modalities $\alpha_1, \alpha_2, \ldots$ Then it is often the case that we have modularity, in the sense that our theory $\mathcal{T}$ can be partitioned in a union of theories

$$\mathcal{T} = \mathcal{T}^{\emptyset} \cup \mathcal{T}^{\alpha_1} \cup \mathcal{T}^{\alpha_2} \cup \ldots$$

such that

- $\mathcal{T}^{\emptyset}$ contains no modal operators, and
- the only modality of $\mathcal{T}^{\alpha_i}$ is α_i.

We call these subtheories *modules*. Examples of such theories can be found in reasoning about actions, where each $\mathcal{T}^{\alpha_i}$ contains descriptions of the atomic action α_i in terms of preconditions and effects, and $\mathcal{T}^{\emptyset}$ is the set of static laws (alias domain constraints, alias integrity constraints), i.e. those formulas that hold in every possible state of a dynamic system. For example, $\mathcal{T}^{marry} = \{\neg married \rightarrow \langle marry \rangle \top, [marry] married\}$, and $\mathcal{T}^{\emptyset} =$

*Supported by a fellowship from the government of the Federative Republic of Brazil. Grant: CAPES BEX 1389/01-7.

$\{\neg(married \wedge bachelor)\}$. Another example is when mental attitudes such as knowledge, beliefs or goals of several independent agents are represented: then each $\mathcal{T}^{\alpha_i}$ contains the respective mental attitude of agent α_i.[1]

Let the underlying multimodal logic be independently axiomatized (i.e. the logic is a fusion and there is no interaction between the modal operators), and suppose we want to know whether $\mathcal{T} \models \varphi$, i.e. whether a formula φ follows from the theory $\mathcal{T}$. Then it is natural to expect that we only have to consider those elements of $\mathcal{T}$ which concern the modal operators occurring in φ. For instance the proof of some consequences of action α_1 should not involve laws for other actions α_2; querying the belief base of agent α_1 should not require bothering with that of agent α_2. Moreover, intensional information in any $\mathcal{T}^{\alpha_i}$ should not influence information about the laws of the world encoded in $\mathcal{T}^{\emptyset}$.

Similar modular design principles can be found in structural and object-oriented programming. We have advocated and investigated the case of reasoning about actions in [3].

2 Preliminaries

Let $MOD = \{\alpha_1, \alpha_2, \ldots\}$ be the set of modalities. Formulas are constructed in the standard way from these and the set of atomic formulas *ATM*. They are denoted by $\varphi, \psi, \ldots$. Formulas without modal operators (propositional formulas) are denoted by $PFOR = \{A, B, C, \ldots\}$.

Let $mod(\varphi)$ return the set of modalities occurring in formula φ, and let $mod(\mathcal{T}) = \bigcup_{\psi \in \mathcal{T}} mod(\psi)$. For instance $mod([\alpha_1](p \to [\alpha_2]q)) = \{\alpha_1, \alpha_2\}$. If $\mathfrak{M} \subseteq MOD$ is a set of modalities then we define

$$\mathcal{T}^{\mathfrak{M}} = \{\varphi \in \mathcal{T} : mod(\varphi) \cap \mathfrak{M} \neq \emptyset\}$$

Hence $\mathcal{T}^{\emptyset}$ is a set of formulas without modal operators. Another example is

$$\mathcal{T}^{\{marry, divorce\}} = \left\{ \begin{array}{c} \neg married \to \langle marry \rangle \top, [marry] married, \\ married \to \langle divorce \rangle \top, [divorce] \neg married \end{array} \right\}$$

We write $\mathcal{T}^{\alpha}$ instead of $\mathcal{T}^{\{\alpha\}}$.

[1] Here we should assume more generally that $[\alpha_i]$ is the only outermost modal operator of $\mathcal{T}^{\alpha_i}$; we think that this case could be analyzed in a way that is similar to ours. Things get just more complicated.

We suppose from now on that $\mathcal{T}$ is *partitioned*, in the sense that $\{\mathcal{T}^\emptyset\} \cup \{\mathcal{T}^{\alpha_i} : \alpha_i \in MOD\}$ is a partition of $\mathcal{T}$. We thus exclude $\mathcal{T}^{\alpha_i}$ containing more than one modal operator.

Models of the logic under concern are of the form $M = \langle W, R, V\rangle$, where W is a set of possible worlds, $R : MOD \longrightarrow W \times W$ associates an accessibility relation to every modality, and $V : W \longrightarrow 2^{ATM}$ associates a valuation to every possible world.

Satisfaction of a formula φ in world w of model M ($M, w \models \varphi$) and truth of a formula φ in M (noted $M \models \varphi$) are defined as usual. Truth of a set of formulas $\mathcal{T}$ in M (noted $M \models \mathcal{T}$) is defined by: $M \models \mathcal{T}$ iff $M \models \psi$ for every $\psi \in \mathcal{T}$. $\mathcal{T}$ has global consequence φ (noted $\mathcal{T} \models \varphi$) iff $M \models \mathcal{T}$ implies $M \models \varphi$.

We suppose that the logic under concern is *compact*.

3 Modularity

Under the hypothesis that $\{\mathcal{T}^\emptyset\} \cup \{\mathcal{T}^{\alpha_i} : \alpha_i \in MOD\}$ partitions $\mathcal{T}$, we are interested in the following principle of modularity:

Definition 3.1 A theory $\mathcal{T}$ is modular if for every formula φ,

$$\mathcal{T} \models \varphi \text{ implies } \mathcal{T}^{mod(\varphi)} \models \varphi$$

Modularity means that when investigating whether φ is a consequence of $\mathcal{T}$, the only formulas of $\mathcal{T}$ that are relevant are those whose modal operators occur in φ.

This is reminiscent of interpolation, which more or less[2] says:

Definition 3.2 A theory $\mathcal{T}$ has the interpolation property if for every formula φ, if $\mathcal{T} \models \varphi$ then there is a theory $\mathcal{T}_\varphi$ such that

- $mod(\mathcal{T}_\varphi) \subseteq mod(\mathcal{T}) \cap mod(\varphi)$
- $\mathcal{T} \models \psi$ for every $\psi \in \mathcal{T}_\varphi$
- $\mathcal{T}_\varphi \models \varphi$

[2] We here present a version in terms of global consequence, as opposed to local consequence or material implication versions that can be found in the literature [4, 5]. We were unable to find such global versions in the literature.

Our definition of modularity is a strengthening of interpolation because it requires $\mathcal{T}_\varphi$ to be a subset of $\mathcal{T}$.

Contrarily to interpolation, modularity does not generally hold. For example, let

$$\mathcal{T} = \{p \vee [\alpha]\bot, p \vee \neg[\alpha]\bot\}$$

Then $\mathcal{T}^\emptyset = \emptyset$, and $\mathcal{T}^\alpha = \mathcal{T}$. Now $\mathcal{T} \models p$, but clearly $\mathcal{T}^\emptyset \not\models p$.

Being modular is a useful feature of theories: beyond being a reasonable principle of design that helps avoiding mistakes, it clearly restricts the search space, and thus makes reasoning easier. To witness, if $\mathcal{T}$ is modular then consistency of $\mathcal{T}$ amounts to consistency (in classical logic) of the propositional part $\mathcal{T}^\emptyset$.

4 Propositional modularity

How can we know whether a given theory $\mathcal{T}$ is modular? The following criterion is simpler:

Definition 4.1 A theory $\mathcal{T}$ is propositionally modular if for every propositional formula A,

$$\mathcal{T} \models A \text{ implies } \mathcal{T}^\emptyset \models A$$

And it will suffice to guarantee modularity:

Theorem 4.1 Let $\mathcal{T}$ be a partitioned theory. If $\mathcal{T}$ is propositionally modular then $\mathcal{T}$ is modular.

Proof: Let $\mathcal{T}$ be propositionally modular. Suppose $\mathcal{T}^{mod(\varphi)} \not\models \varphi$. Hence there is a model $M = \langle W, R, V\rangle$ such that $M \models \mathcal{T}^{mod(\varphi)}$, and there is some w in M such that $M, w \not\models \varphi$. We prove that $\mathcal{T} \not\models \varphi$ by constructing from M a model M' such that $M' \models \mathcal{T}$ and $M', w \not\models \varphi$.

First, as we have supposed that our logic is compact, propositional modularity implies that for every propositional valuation $val \subseteq 2^{ATM}$ which is a model of $\mathcal{T}^\emptyset$ there is a possible worlds model $M_{val} = \langle W_{val}, R_{val}, V_{val}\rangle$ such that $M_{val} \models \mathcal{T}$, and there is some w in M_{val} such that $V_{val}(w) = val$. In other words, for every propositional model of $\mathcal{T}^\emptyset$ there is a model of $\mathcal{T}$ containing that propositional model.

Second, taking the disjoint union of all these models we obtain a 'big model' M_{big} such that $M_{big} \models \mathcal{T}$, and for every propositional model $val \subseteq 2^{ATM}$ of $\mathcal{T}^{\emptyset}$ there is a possible world w in M_{big} such that $V(w) = val$.

Now we can use the big model to adjust those accessibility relations $R(\alpha)$ of M whose α does not appear in φ, in a way such that the resulting model satisfies the rest of the theory $\mathcal{T} \setminus \mathcal{T}^{mod(\varphi)}$: let $M' = \langle W', R', V' \rangle$ such that

- $W' = \{u_v : u \in W, v \in W_{big}, \text{ and } V(u) = V_{big}(v)\}$
- if $\alpha \in mod(\varphi)$ then $u_v R'(\alpha) u'_{v'}$ iff uRu'
- if $\alpha \notin mod(\varphi)$ then $u_v R'(\alpha) u'_{v'}$ iff vRv'
- $V'(u_v) = V(u) = V_{big}(v)$

W' is nonempty because M' is 'big enough' and contains every possible propositional model of $\mathcal{T}^{\emptyset}$. Then for the sublanguage constructed from $mod(\varphi)$ it can be proved by structural induction that for every formula ψ of the sublanguage and every $u \in W$ and $v \in W_{big}$, $M, u \models \psi$ iff $M', u_v \models \psi$. The same can be proved for the sublanguage constructed from $MOD \setminus mod(\varphi)$. As $\mathcal{T}^{\emptyset}$ and each of our modules $\mathcal{T}^{\alpha}$ are in at least one of these sublanguages (in both sublanguages in the case of $\mathcal{T}^{\emptyset}$), we have thus proved that $M' \models \mathcal{T}$, and $M', w_v \not\models \varphi$ for every v. ■

5 Action theories

In the rest of the paper we investigate how it can be automatically checked whether a given theory $\mathcal{T}$ is modular or not. We do this for a particular kind of theories that are commonly used in reasoning about actions. For such theories we also show how the parts of the theory that are responsible for the violation of modularity can be identified. First of all we say what an action theory is.

Every action theory contains a representation of action effects. We call *effect laws* formulas relating an action to its effects. *Executability laws* in turn stipulate the context where an action is guaranteed to be executable. Finally, *static laws* are formulas that do not mention actions and express constraints that must hold in every possible state. These are our four ingredients that we introduce more formally in the sequel.

Static laws Frameworks which allow for indirect effects make use of logical formulas that link invariant propositions about the world. Such formulas characterize the set of possible states. They do not refer to actions, and we suppose they are formulas of classical propositional logic $A, B, \ldots \in \mathit{PFOR}$.

A *static law*[3] is a formula $A \in \mathit{PFOR}$ that is consistent. An example is $\mathit{Walking} \rightarrow \mathit{Alive}$, saying that if a turkey is walking, then it must be alive [10].

Effect laws Here MOD is the set of all actions. To speak about action effects we use the syntax of propositional dynamic logic (PDL) [2]. The formula $[\alpha]A$ expresses that A is true after every possible execution of α.

An *effect law*[4] *for* α is of the form $A \rightarrow [\alpha]C$, where $A, C \in \mathit{PFOR}$. The consequent C is the effect which obtains when α is executed in a state where the antecedent A holds. An example is $\mathit{Loaded} \rightarrow [\mathit{shoot}]\neg \mathit{Alive}$, saying that whenever the gun is loaded, after shooting the turkey is dead. Another one is $[\mathit{tease}]\mathit{Walking}$: in every circumstance, the result of teasing is that the turkey starts walking.

A particular case of effect laws are *inexecutability laws* of the form $A \rightarrow [\alpha]\bot$. For example $\neg \mathit{HasGun} \rightarrow [\mathit{shoot}]\bot$ expresses that *shoot* cannot be executed if the agent has no gun.

Executability laws With only static and effect laws one cannot guarantee that *shoot* is executable if the agent has a gun. [5] In dynamic logic the dual $\langle\alpha\rangle A$, defined as $\neg[\alpha]\neg A$, can be used to express executability. $\langle\alpha\rangle\top$ thus reads "the execution of action α is possible".

An *executability law*[6] *for* α is of the form $A \rightarrow \langle\alpha\rangle\top$, where $A \in \mathit{PFOR}$.

---

[3]Static laws are often called *domain constraints*, but the different laws for actions that we shall introduce in the sequel could in principle also be called like that.

[4]Effect laws are often called *action laws*, but we prefer not to use that term here because it would also apply to executability laws that are to be introduced in the sequel.

[5]Some authors [9, 1, 7, 10] more or less tacitly consider that executability laws should not be made explicit, but rather inferred by the reasoning mechanism. Others [6, 11] have executability laws as first class objects one can reason about. It seems a matter of debate whether one can always do without, but we think that in several domains one wants to explicitly state under which conditions a given action is guaranteed to be executable, e.g. that a robot should never get stuck and should always be able to execute a move action. In any case, allowing for executability laws gives us more flexibility and expressive power.

[6]Some approaches (most prominently Reiter's) use biconditionals $A \leftrightarrow \langle\alpha\rangle\top$, called precondition axioms. This is equivalent to $\neg A \leftrightarrow [\alpha]\bot$, which illustrates that they thus merge information about inexecutability with information about executability.

For instance $HasGun \rightarrow \langle shoot \rangle \top$ says that shooting can be executed whenever the agent has a gun, and $\langle tease \rangle \top$ says that the turkey can always be teased.

Action theories $\mathcal{S} \subseteq PFOR$ denotes the set of all static laws of a domain. For a given action $\alpha \in MOD$, $\mathcal{E}_\alpha$ is the set of its effect laws, and $\mathcal{X}_\alpha$ is the set of its executability laws. We define $\mathcal{E} = \bigcup_{\alpha \in MOD} \mathcal{E}_\alpha$, and $\mathcal{X} = \bigcup_{\alpha \in MOD} \mathcal{X}_\alpha$. An *action theory* is a tuple of the form $\langle \mathcal{S}, \mathcal{E}, \mathcal{X} \rangle$. We suppose that $\mathcal{S}, \mathcal{E}$ and $\mathcal{X}$ are finite.

6 Checking modularity of action theories

How can we check whether a given action theory $\mathcal{T} = \langle \mathcal{S}, \mathcal{E}, \mathcal{X} \rangle$ is modular? Assuming $\mathcal{T}$ is finite, the algorithm below does the job:

Algorithm 6.1 (Modularity check)

input: $\mathcal{S}, \mathcal{E}, \mathcal{X}$
output: a set of implicit static laws $\mathcal{S}^I$
  $\mathcal{S}^I := \emptyset$
  for all α **do**
    for all $J \subseteq \mathcal{E}_\alpha$ **do**
      $A_J := \bigwedge\{A_i : A_i \rightarrow [\alpha]C_i \in J\}$
      $C_J := \bigwedge\{C_i : A_i \rightarrow [\alpha]C_i \in J\}$
      if $\mathcal{S} \cup \{A_J\} \not\vdash \bot$ and $\mathcal{S} \cup \{C_J\} \vdash \bot$ **then**
        for all $B \rightarrow \langle \alpha \rangle \top \in \mathcal{X}$ **do**
          if $A_J \wedge B \not\vdash \bot$ **then**
            $\mathcal{S}^I := \mathcal{S}^I \cup \{\neg(A_J \wedge B)\}$

Theorem 6.1 An action theory $\langle \mathcal{S}, \mathcal{E}, \mathcal{X} \rangle$ is modular iff $\mathcal{S}^I = \emptyset$.

The proof of this theorem relies on a sort of interpolation theorem for multimodal logic, which basically says that if $\Phi \models \Psi$ and Φ and Ψ have no action symbol in common, then there is a classical formula A such that $\Phi \models A$ and $A \models \Psi$.[7]

[7]The detailed proof can be found in `http://www.irit.fr/ACTIVITES/LILaC/Pers/Herzig/P/AiML04.html`

Remark 6.1 In [3] a monotonic solution to the frame problem has been integrated in such an algorithm. This makes the algorithm a bit more complex as it involves computing of prime implicates. For the sake of simplicity this has not been done here.

7 Discussion and conclusion

In the perspective of independently axiomatized multimodal logics we have investigated several criteria of modularity for simple theories. We have demonstrated the usefulness of modularity in reasoning about actions, where we have given an algorithmic checking for modularity of a given action theory.

We can have our criterion of modularity refined by taking into account polarity. Let $mod^{\pm}(\varphi)$ be the set of modalities of *MOD* occurring in φ together with their polarity. For instance $mod^{\pm}([\alpha_1]([\alpha_2]p \rightarrow q)) = \{+\alpha_1, -\alpha_2\}$. $mod^{\pm}(\mathcal{T})$ is defined accordingly. If $\mathcal{M}$ is a set of modalities with polarity then we define: $\mathcal{T}^{\mathcal{M}} = \{\varphi \in \mathcal{T} : mod^{\pm}(\varphi) \cap \mathcal{M} \neq \emptyset\}$.

Definition 7.1 A theory $\mathcal{T}$ is $\pm$-modular if for every formula φ,

$$\mathcal{T} \models \varphi \text{ implies } \mathcal{T}^{mod^{\pm}(\varphi)} \models \varphi$$

There are other theories that are modular but not $\pm$-modular, e.g.

$$\mathcal{T} = \{\neg[\alpha]p, [\alpha]p \vee [\alpha]\neg p\}$$

Indeed, $\mathcal{T} \models [\alpha]\neg p$, but $\mathcal{T}^{+\alpha} \not\models [\alpha]\neg p$.

For the restricted case of action theories this has been proved in [3].

References

[1] P. Doherty, W. Łukaszewicz, and A. Szałas. Explaining explanation closure. In *Proc. Int. Symposium on Methodologies for Intelligent Systems*, Zakopane, Poland, 1996.

[2] D. Harel. Dynamic logic. In D. M. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic*, volume II, pages 497–604. D. Reidel, Dordrecht, 1984.

[3] A. Herzig and I. Varzinczak. Domain descriptions should be modular. In *Proc. ECAI'04*, 2004.

[4] M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal logics. *J. of Symbolic Logic*, 56(4):1469–1485, 1991.

[5] M. Kracht and F. Wolter. Simulation and transfer results in modal logic: A survey. *Studia Logica*, 59:149–177, 1997.

[6] F. Lin. Embracing causality in specifying the indirect effects of actions. In Mellish [8], pages 1985–1991.

[7] N. McCain and H. Turner. A causal theory of ramifications and qualifications. In Mellish [8], pages 1978–1984.

[8] C. Mellish, editor. *Proc. 14th Int. Joint Conf. on Artificial Intelligence (IJCAI'95)*, Montreal, 1995. Morgan Kaufmann Publishers.

[9] L. K. Schubert. Monotonic solution of the frame problem in the situation calculus: an efficient method for worlds with fully specified actions. In H. E. Kyberg, R. P. Loui, and G. N. Carlson, editors, *Knowledge Representation and Defeasible Reasoning*, pages 23–67. Kluwer Academic Publishers, 1990.

[10] M. Thielscher. Computing ramifications by postprocessing. In Mellish [8], pages 1994–2000.

[11] D. Zhang and N. Y. Foo. EPDL: A logic for causal reasoning. In B. Nebel, editor, *Proc. 17th Int. Joint Conf. on Artificial Intelligence (IJCAI'01)*, pages 131–138, Seattle, 2001. Morgan Kaufmann Publishers.