A Decidable Logic for Complex Contracts[⋆]
(extended abstract)

Cristian Prisacariu

Department of Informatics, University of Oslo,
P.O. Box 1080 Blindern, 0316 Oslo, Norway.
E-mail: `cristi@ifi.uio.no`

1 Motivation

The present abstract reports on the state-of-the-art of the logic for contracts $\mathcal{CL}$, which we have been developing over the last few years. These research efforts have spread over several papers [7, 6, 3, 5], which we summarize here. We present also some current work. The main motivation for our work is to have a logic which expresses, at an abstract level, complex behavior (of concurrent programs, of communicating intelligent agents, of web services, or of legal contracts; just to name a few possible concretizations of the behaviors we think of). The purpose of this logic is, not only, to formalize such behaviors, but mainly to reason about their formalizations. Therefore we seek a decidable logic, so to have hopes for automation; i.e., to do automated model checking and run-time monitoring.

Legal contracts talk about behavior which encompasses all the other behaviors that we mentioned before. Therefore, a formalism for reasoning about legal contracts offers the opportunity to reason about the other behaviors too, by instantiation (and possibly small changes). We have developed a logic (which we call $\mathcal{CL}$) designed to represent and reason about contracts (being that software contracts, web services, interfaces, or communication protocols). $\mathcal{CL}$ combines deontic logic (i.e., the logic of legal contracts) [8] with propositional dynamic logic (PDL, the logic of actions) [2]. $\mathcal{CL}$ applies the deontic and dynamic modalities *exclusively* over actions.

Moreover, $\mathcal{CL}$ extends the (regular) actions that PDL talks about, with a concurrency operator to model the notion of actions "done at the same time". We adopt the synchrony model of concurrency of Milner's SCCS calculus [4]. Synchrony is easy to integrate with the regular operations on actions: choice, sequence, and repetition. To be more precise, the *synchrony model* takes the assumption that time is discrete and that basic actions are instantaneous. Moreover, at each time step each and all the concurrent components perform instantaneously a single action (no idling being permitted unless explicitly stated); i.e. the system is considered *eager* and *active*. This is opposed to the standard *asynchrony* where two concurrent systems proceed at indeterminate relative speeds (i.e. their actions may have different non-correlated durations).

$\mathcal{CL}$ combines the expressive power of PDL (which subsumes Hoare logic for talking about sequential programs) with a natural notion of concurrent actions. On top, $\mathcal{CL}$ incorporates the expressive deontic modalities *obligation*, *permission*, and *prohibition* applied over actions. The actions of $\mathcal{CL}$ are abstract entities which can be instantiated to, e.g., communication primitives, assignments, or tests.

[⋆] Partially supported by the Nordunet3 project "COSoDIS – Contract-Oriented Software Development for Internet Services" (`http://www.ifi.uio.no/cosodis/`).

$$\begin{array}{rcl} \mathcal{C} & := & \phi \mid O_{\mathcal{C}}(\alpha) \mid P(\alpha) \mid F_{\mathcal{C}}(\alpha) \mid \mathcal{C} \rightarrow \mathcal{C} \mid [\beta]\mathcal{C} \mid \bot \\ \alpha & := & a \mid \mathbf{0} \mid \mathbf{1} \mid \alpha \times \alpha \mid \alpha \cdot \alpha \mid \alpha + \alpha \\ \beta & := & a \mid \mathbf{0} \mid \mathbf{1} \mid \beta \times \beta \mid \beta \cdot \beta \mid \beta + \beta \mid \beta^* \mid \varphi? \\ \varphi? & := & \phi \mid \mathbf{0} \mid \mathbf{1} \mid \varphi? \vee \varphi? \mid \varphi? \wedge \varphi? \mid \neg \varphi? \end{array}$$

Table 1. Syntax of the contract language $\mathcal{CL}$.

2 Results

The syntax of $\mathcal{CL}$ is defined by the grammar in Table 1. We call a formula $\mathcal{C}$ a *contract clause* (or plainly *contract*). A contract is built up from propositional constants ϕ, obligations $O_{\mathcal{C}}(\alpha)$ of actions α, permissions $P(\alpha)$, and prohibitions $F_{\mathcal{C}}(\alpha)$ (i.e., the *deontic modalities* applied over actions α). $\mathcal{CL}$ includes directly in the definition of the obligation and prohibition the *reparations* $\mathcal{C}$ in case of violations (i.e., whenever α is not performed, in the case of an obligation). Contracts are combined using the standard Boolean operators $\rightarrow$ and $\bot$ from which all other can be derived. Moreover, contracts may be prefixed by the dynamic logic modality $[\beta]\mathcal{C}$ which is read as: "after performing action β then the contract $\mathcal{C}$ must hold.

The deontic actions α are constructed from a finite set of basic actions $\mathcal{A}_B$ using the operations of *synchrony* $\times$, *sequence* $\cdot$, and *choice* $+$. The dynamic actions have the extra operations of *Kleene star* * and *tests* $\varphi?$. Tests are constructed with the Boolean operators from the propositional constants ϕ. Therefore, we do not have the full power of PDL but only that of *poor tests* PDL.

The formulas of the logic are given a model theoretic semantics in terms of (what we call) *normative structures*.

Definition 1. *A* normative structure $K^{\mathcal{N}} = (\mathcal{W}, \rho, \mathcal{V}, \varrho)$ *is a deterministic Kripke structure with a marking function $\varrho : \mathcal{W} \rightarrow 2^{\Psi}$ which marks each state with one or several markers from $\Psi = \{\circ_a, \bullet_a \mid a \in \mathcal{A}_B\}$. The marking function and the markers are needed to identify obligatory (i.e. $\circ$) and prohibited (i.e. $\bullet$) actions. The labels of the normative structure are subsets of basic actions $\mathcal{A}_B$. Therefore, we compare them using set inclusion (and call one label bigger than another).*

We give below the definition of satisfaction of formula $\mathcal{C}$ w.r.t. a pointed normative structure $K^{\mathcal{N}}, i$, written $K^{\mathcal{N}}, i \models \mathcal{C}$. The semantics for $O_{\mathcal{C}}$ has basically two parts. The first part (lines one to four) gives the interpretation of the obligation. Line one says, using the simulation relation $\mathcal{S}_i$, how to walk on the normative structure depending on the tree interpretation $I(\alpha)$ of the action. The precise definitions of these concepts are out of the scope of this paper. Lines two and three say how the states must be marked with $\circ$. The second part of the semantics is the last line which handles our notion of reparation in case of violations of obligations. The negation of an action $\overline{\alpha}$ encodes all possible ways of violating $O(\alpha)$. Thus, at the end of each of these the reparation must be enforced.

$$\begin{array}{lll} 1 & K^{\mathcal{N}}, i \models O_{\mathcal{C}}(\alpha) \text{ iff } I(\alpha)\, \mathcal{S}_i\, K^{\mathcal{N}}, \quad \text{and} & \\ 2 & & \forall t \xrightarrow{\gamma} t' \in I(\alpha), \forall s \xrightarrow{\gamma'} s' \in K^{\mathcal{N}} \text{ s.t. } t\, \mathcal{S}\, s \wedge \gamma \subseteq \gamma' \text{ then} \\ 3 & & \quad \forall a \in \gamma, \circ_a \in \varrho(s'), \quad \text{and} \\ 4 & & \forall s \xrightarrow{\gamma'} s' \in K_{rem}^{I(\alpha),i}, \forall a \in \gamma' \text{ then } \circ_a \notin \varrho(s'), \quad \text{and} \\ 5 & & K^{\mathcal{N}}, s \models \mathcal{C} \quad \forall s \in N \text{ with } t\, \mathcal{S}^s\, s \wedge t \in \mathit{leaves}(I(\overline{\alpha})). \end{array}$$

The $\mathcal{CL}$ logic enjoys the properties given in Proposition 1 (which make the starting point for the axiomatization that we are currently working on) and avoids some common contradictions, where $\alpha \,\#_{\mathcal{C}}\, \alpha'$ means that the two actions are conflicting.

Proposition 1 (validities). *The following statements hold:*

$$\models \neg O_{\mathcal{C}}(\mathbf{0}) \quad (1)$$
$$\models O_{\mathcal{C}}(\mathbf{1}) \quad (2)$$
$$\models P(\alpha) \rightarrow \neg F_{\mathcal{C}}(\alpha) \quad (3)$$
$$\models O_{\mathcal{C}}(\alpha) \rightarrow P(\alpha) \quad (4)$$
$$\textit{if } \alpha = \alpha' \textit{ then } \models O_{\mathcal{C}}(\alpha) \leftrightarrow O_{\mathcal{C}}(\alpha') \quad (5)$$
$$\models O_{\mathcal{C}}(\alpha) \rightarrow \neg F_{\mathcal{C}}(\alpha) \quad (6)$$
$$\models O_{\mathcal{C}}(\alpha) \wedge O_{\mathcal{C}}(\alpha') \rightarrow O_{\mathcal{C}}(\alpha \times \alpha') \quad (7)$$
$$\models O_{\mathcal{C}}(\alpha \cdot \alpha') \leftrightarrow O_{\mathcal{C}}(\alpha) \wedge [[\alpha]]O_{\mathcal{C}}(\alpha') \quad (8)$$
$$\models F_{\mathcal{C}}(\alpha) \rightarrow F_{\mathcal{C}}(\alpha \times \alpha') \quad (9)$$
$$\models F_{\mathcal{C}}(\alpha + \alpha') \leftrightarrow F_{\mathcal{C}}(\alpha) \wedge F_{\mathcal{C}}(\alpha') \quad (10)$$
$$\models F_{\mathcal{C}}(\alpha \cdot \alpha') \leftrightarrow F_{\mathcal{C}}(\alpha) \vee \langle\langle \alpha \rangle\rangle F_{\mathcal{C}}(\alpha') \quad (11)$$
$$\models P(\alpha + \alpha') \leftrightarrow P(\alpha) \wedge P(\alpha') \quad (12)$$
$$\models P(\alpha \cdot \alpha') \leftrightarrow P(\alpha) \wedge [\alpha]P(\alpha') \quad (13)$$

The following point out conflicts *that are avoided in* $\mathcal{CL}$*:*

$$\models \neg(O_{\mathcal{C}}(\alpha) \wedge F_{\mathcal{C}}(\alpha)) \quad (14)$$
$$\models \neg(P(\alpha) \wedge F_{\mathcal{C}}(\alpha)) \quad (15)$$
$$\textit{if } \alpha \,\#_{\mathcal{C}}\, \alpha' \textit{ then } \models \neg(O_{\mathcal{C}}(\alpha) \wedge O_{\mathcal{C}}(\alpha')) \quad (16)$$

Theorem 1 (tree model property). *Given a contract* $\mathcal{C}$ *and a pointed normative structure* $K^{\mathcal{N}}, i$ *then we can construct an associated tree structure* $TK^{\mathcal{N}}, \varepsilon$ *s.t.*

$$TK^{\mathcal{N}}, \varepsilon \models \mathcal{C} \quad \textit{iff} \quad K^{\mathcal{N}}, i \models \mathcal{C}.$$

We show that $\mathcal{CL}$ is *decidable* by showing that it has the *finite model property*. Note first that it is rather hard to use the filtration technique in our case because we do not know what are subformulas of an obligation of a complex action like $O_{\mathcal{C}}(a \cdot (b+c))$. We use the *selection* technique for proving the finite model property [1, sec.2.3].

Theorem 2 (finite model property). *For a formula* $\mathcal{C}$ *of depth* k*, if* $TK^{\mathcal{N}}, \varepsilon \models \mathcal{C}$ *then* $\mathcal{C}$ *holds in* $TK^{\mathcal{N}}, \varepsilon$ *restricted to paths of maximum depth* k *(i.e. where all nodes of depth* $> k$ *are removed). Hence, if a formula has a model then it has a finite model.*

The *depth* is calculated as $d(\mathcal{C}) * max(l(\alpha) \mid \forall \alpha \in \mathcal{C})$, which multiplies the *nesting degree* of a formula $d(\mathcal{C})$ (which counts the number of nesting of O or F inside the reparations; e.g. $d(O_{F(\alpha)}(\beta)) = 2$) with the maximal length of the actions that appear in the formula.

Current work. We are now working on a Hilbert-style axiomatization for $\mathcal{CL}$. This opens the way for a proof system for $\mathcal{CL}$ which would complement the model checking and the run-time monitoring that we have.

References

1. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge Univ. Press, 2001.
2. M. J. Fischer and R. E. Ladner. Propositional modal logic of programs. In *9th ACM Symposium on Theory of Computing (STOC'77)*, pages 286–294. ACM, 1977.
3. M. Kyas, C. Prisacariu, and G. Schneider. Run-time Monitoring of Electronic Contracts. In *ATVA'08*, volume 5311 of *LNCS*, pages 397–407. Springer, 2008.
4. R. Milner. Calculi for synchrony and asynchrony. *Theor. Comput. Sci.*, 25:267–310, 1983.
5. C. Prisacariu and G. Schneider. CL – A Logic for Reasoning about Legal Contracts: – Semantics. Technical Report 371, Dep. Info., Univ. Oslo, February 2008.
6. C. Prisacariu and G. Schneider. Abstract Specification of Legal Contracts. In *12th International Conference on Artificial Intelligence and Law (ICAIL09)*, pages 218–219. ACM, 2009.
7. C. Prisacariu and G. Schneider. CL: An Action-based Logic for Reasoning about Contracts. In *WOLLIC'09*, volume 5514 of *LNCS*, pages 335–349. Springer, 2009.
8. G. H. V. Wright. Deontic logic. *Mind*, 60:1–15, 1951.