Dynamic languages of propositional control for protocol specification

Andreas Herzig[1] and Nicolas Troquard[2]

1. Université de Toulouse, CNRS, IRIT, France
2. Department of Computer Science, University of Liverpool, UK

Abstract. We propose a family of dynamic logics of propositional control. They extend classical propositional logic by a variety of modal operators of assignment, and modal operators of transfer of control over a propositional variable. We also present an extension with operators of knowledge. We essentially focus on their formal properties, stating their complexity and their proof theory.

1 Introduction

The logic of propositional control CL-PC was introduced in [10] as a reconstruction of coalition logic. What agents can achieve is explained there in terms of their control over propositional variables. The central construction is a modal operator $\langle \overset{J}{\leftarrow} \rangle$ of contingent ability, where J is a set of agents.[1] The formula $\langle \overset{J}{\leftarrow} \rangle\varphi$ reads "the coalition of agents J can assign truth values to the variables under its control in such a way as to make φ true". For example, $\langle \overset{\{i\}}{\leftarrow} \rangle q \wedge \langle \overset{\{i\}}{\leftarrow} \rangle \neg q$ expresses that agent i is able both to make q true and to make q false; which means that agent i controls q.

The logic was further studied and extended in [7, 15, 14, 13]. A particularly interesting such extension is that by *delegation* CL-PC, coined δCL-PC; see [8] for a recent presentation.[2] It introduces into CL-PC new modalities of control transfer (baptized 'delegation' in the original papers). In that work, atomic delegation programs take the form $i \overset{q}{\hookrightarrow} j$, whose intended meaning is that i who currently controls q turns over its control to j. Moreover, complex control transfer programs δ are constructed by means of the standard PDL constructs ? (test), ; (sequential composition), $\cup$ (nondeterministic composition) and * (iteration). The modal formula $\langle \delta \rangle \varphi$ then reads "there exists a computation of the delegation program δ, starting from the current situation, such that after δ has terminated φ holds". Such constructs allow to talk about interaction protocols.

The semantics of CL-PC and δCL-PC are originally in terms of couples (ξ, θ) where θ maps every propositional variable q to a truth value $\theta(q)$ in $\{\mathtt{tt}, \mathtt{ff}\}$, and ξ maps every propositional variable q to one agent. We will here consider a more general setting (to be introduced in Section 2), where ξ maps every propositional variable q to a —possibly

[1] The original notation is $\Diamond_J \varphi$ instead of $\langle \overset{J}{\leftarrow} \rangle\varphi$.

[2] It was originally abbreviated DCL-PC, but we changed it in order to avoid confusion with the logics we introduce here. In these logics (and more generally in dynamic epistemic logics [5]) the letter 'D' stands for 'dynamic'.

empty— *set* of agents $\xi(q)$. We call the mapping ξ a *control allocation* and the mapping θ a *valuation*. ξ is about control, while θ is about truth. Given (ξ, θ), call an *update of J's part of θ* any model (ξ', θ') such that $\xi' = \xi$ and $\theta'(q) = \theta(q)$ for every q such that $\xi(q)$ is not in J.

In addition to generalising the models of CL-PC, our main objective is to elaborate on the basic bricks[3] of the language. Indeed, in much the same spirit as the program $i\overset{q}{\hookrightarrow}j$ updates the allocation function ξ, one may view the operator $\langle\overset{J}{\leftarrow}\rangle$ of CL-PC as the application of a program $\overset{J}{\leftarrow}$ that updates the valuation function θ by changing at most the value of the propositions under the control of the agents in J.

This observation made, one of the main contributions of this paper is to show that we can redefine the logics of CL-PC and δCL-PC from a dynamic logic with very simple programs. Our logic $\mathsf{DL}^{\mathsf{PC}}$ (that we introduce in Section 3) makes use of four types of atomic programs:

$q{\leftarrow}\top$: q is given the value tt
$q{\leftarrow}\bot$: q is given the value ff
$i\overset{q}{\hookrightarrow}$: i loses the control of q
$\overset{q}{\hookrightarrow}i$: i receives the control of q

We keep track of who owns a propositional variable by the use of a theory of propositions $c_{i,q}$, for every agent i and proposition q, that reads that i controls q. For instance, the δCL-PC program $i\overset{q}{\hookrightarrow}j$ will then be simulated by a test of $c_{i,q}$ followed by i's loss and j's gain of control over q.

In Section 4 we introduce two new primitive programs to obtain a logic that we call Q-$\mathsf{DL}^{\mathsf{PC}}$, for *quantified* $\mathsf{DL}^{\mathsf{PC}}$. The program $\overset{q}{\hookrightarrow}J$ gives the control of q to one of the agents in J, and the program $q\overset{J}{\leftarrow}$ sets the value of q that is controlled by some agent in J to tt or ff. These programs may look a bit abstract. They can in fact be defined in terms of the four programs listed above as we will see, but to the price of losing a bit of succinctness. Moreover, they will provide the good level of abstraction to relate our languages with those of CL-PC and Coalition Logic.

Here is an example illustrating the concepts that we have introduced so far.

Example 1. Consider the reviewing process of some conference. Let the set of agents be $\mathbb{A} = \{Chair\} \cup RV$, where *Chair* is the PC chair and $RV = \{R_1, \ldots, R_M\}$ is the set of reviewers. Let there be N papers $PP = \{1, \ldots, N\}$ to be reviewed. We suppose that the decision of acceptance of each paper $n \in PP$ is based on three opinions. Each opinion is modelled by the truth value of a propositional variable. Let their set be $\mathbb{P} = \bigcup_{1\leq n\leq N}\{p_n^1, p_n^2, p_n^3\}$. Initially it is the PC chair who controls all the papers, i.e. we have an initial model where $\xi(p_n^k) = \{Chair\}$ for all $n \in PP$ and $k \in \{1, 2, 3\}$. The assignment of papers to reviewers corresponds to the execution of the sequence $\overset{p_1^1}{\hookrightarrow}RV, \cdots, \overset{p_N^3}{\hookrightarrow}RV$. After that, every p_n^k is controlled by both the PC chair and some reviewer. Finally, the

[3] We will generalise the atomic programs, but we will keep a complete investigation of the PDL constructs for later work.

reviewers' decisions are modelled as the assignment of truth values to papers under their control, i.e. the programs $p^k_n \overset{R_m}{\leftarrow} \top$ and $p^k_n \overset{R_m}{\leftarrow} \bot$. One may then check properties such as $\langle \overset{PP}{\hookrightarrow} RV \rangle \langle PP \overset{RV}{\leftarrow} \rangle \varphi$ for some property φ expressing for instance that the acceptance rate is 25%. A more complex and more realistic check where the final decision is up to the PC chair can be expressed by $\langle Chair \overset{PP}{\hookrightarrow} RV \rangle \langle PP \overset{RV}{\leftarrow} \rangle \langle RV \overset{PP}{\hookrightarrow} Chair \rangle \langle PP \overset{RV}{\leftarrow} \rangle \langle PP \overset{Chair}{\leftarrow} \rangle \varphi$.

We are going to take this example up in the end of Section 4.1.

Later on in this paper (in Section 5) we will investigate an adequate mix of the dynamic logic of propositional control with knowledge. Though related, up to now the logic of assignments was studied independently of CL-PC in the framework of extensions of public announcement logic [5, 11]. The latter aim at modelling how agents' knowledge changes when some propositional variable is publicly assigned to true or to false. There, assignments take the form $q \leftarrow \psi$, and the formula $\langle q \leftarrow \psi \rangle \varphi$ reads "φ is true after the assignment of ψ to q". So these assignments differ in two respects from ours: first, q may be assigned any formula ψ, and second, no agent performing an assignment is mentioned. As to the first point, assigning only $\top$ and $\bot$ is going to simplify our technicalities; as to the second point, in the perspective of extending a logic of agency such as CL-PC it is appealing to consider that assignments are performed by agents. We will see that there are interesting and intricate complications that arise when the agents learn that the value of a proposition has changed or when an agent publicly transfers her control over a proposition to another agent.

We believe that logics of propositional control offer a concrete and general tool for specifying interaction protocols of intelligent agents. The investigation of dynamic logics of propositional control appears bottomless. We present a few of these possible variants in Section 6.

2 PC models: models of propositional control

Throughout the paper, $\mathbb{A}$ denotes a (fixed) finite set of agents and $\mathbb{P}$ denotes a (fixed) countable set of propositional variables. A *coalition* is a subset of agents $J \subseteq \mathbb{A}$. The set $\overline{J} = \mathbb{A} \setminus J$ is the *complement* of J.

Definition 1. *A model of propositional control (PC model) is a couple* (ξ, θ) *where:*

- $\xi : \mathbb{P} \longrightarrow 2^{\mathbb{A}}$*, called an allocation;*
- $\theta : \mathbb{P} \longrightarrow \{\mathtt{tt}, \mathtt{ff}\}$*, called a valuation.*

An allocation maps propositional variables to agents. The set $\{q \ : \ i \in \xi(q)\}$ is i's part of ξ: the set of propositions under the control of i. The function ξ determines the initial allocation of propositional variables to agents, and θ determines the initial truth value of the propositional variables.

In the terminology of Gerbrandy [7], the models of CL-PC and δCL-PC are PC models where the control of every variable is both exclusive (allocated to at most one agent) and actual (allocated to at least one agent).[4]

[4] To match van der Hoek and Wooldridge's models, Gerbrandy actually needs to strengthen his abstract models with a property of full control. It says that if an agent i controls a set At_i of

Definition 2. *We say a PC model* (ξ, θ) *has* exclusive and actual control *if* $\xi(q)$ *is a singleton for every variable* $q \in \mathbb{P}$.

We are going to present several languages to talk about these structures of propositional control. Apart from the epistemic extension that is presented in Section 5, all languages will be interpreted on these models. The epistemic extension is going to be interpreted on a generalisation of PC models.

For each of the languages that we are going to introduce, we define $\mathbb{A}_\varphi$ to be the set of agents from $\mathbb{A}$ occurring in φ, and we define $\mathbb{P}_\varphi$ to be the set of variables from $\mathbb{P}$ occurring in φ.

3 $\mathsf{DL}^{\mathsf{PC}}$: dynamic logic of propositional control

In this section we define syntax and semantics of the dynamic logic of propositional control $\mathsf{DL}^{\mathsf{PC}}$, whose modal operators are $\langle q{\leftarrow}\top\rangle$ (setting q to true), $\langle q{\leftarrow}\bot\rangle$ (setting q to false), $\langle i\overset{q}{\hookrightarrow}\rangle$ (i loosing control of q), and $\langle\overset{q}{\hookrightarrow}i\rangle$ (i obtaining control of q). We prove the NP-completeness of $\mathsf{DL}^{\mathsf{PC}}$ satisfiability.

3.1 Language and semantics

Beyond the modal operators that we have introduced informally in the introduction, we also need constants $c_{i,q}$ that are read "agent i controls variable q". They are going to be useful to state the reduction axioms for our basic logic.

The language of $\mathsf{DL}^{\mathsf{PC}}$ is defined by the following BNF:

$$\varphi ::= q \mid \top \mid \bot \mid c_{i,q} \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle q{\leftarrow}\top\rangle\varphi \mid \langle q{\leftarrow}\bot\rangle\varphi \mid \langle i\overset{q}{\hookrightarrow}\rangle\varphi \mid \langle\overset{q}{\hookrightarrow}i\rangle\varphi$$

where i ranges over $\mathbb{A}$ and q ranges over $\mathbb{P}$.

We use the logical connectives $\wedge$, $\rightarrow$ and $\leftrightarrow$ with the usual meaning. We use $q{\leftarrow}\tau$ in order to talk about $q{\leftarrow}\top$ and $q{\leftarrow}\bot$ in an economic way, where τ is a placeholder for either $\top$ or $\bot$.

The *length* of a formula φ, noted $|\varphi|$, is the number of symbols used to write down φ (without $\langle$, $\rangle$, parentheses and commas). For example $|c_{i,q}| = 3$ and $|\langle q{\leftarrow}\top\rangle(q \wedge r)| = 3 + 3 = 6$.

Given a PC model (ξ, θ), we are going to update θ in order to interpret the assignment operators, and we are going to update ξ in order to interpret allocation programs.

Given a valuation θ and an allocation ξ, we define the updates $\xi^{i\overset{q}{\hookrightarrow}}$, $\xi^{\overset{q}{\hookrightarrow}i}$, $\theta^{q\leftarrow\top}$ and $\theta^{q\leftarrow\bot}$ as follows:

$$\xi^{i\overset{q}{\hookrightarrow}}(p) = \begin{cases} \xi(p) \setminus \{i\} & \text{when } p = q \\ \xi(p) & \text{otherwise.} \end{cases} \qquad \xi^{\overset{q}{\hookrightarrow}i}(p) = \begin{cases} \xi(p) \cup \{i\} & \text{when } p = q \\ \xi(p) & \text{otherwise.} \end{cases}$$

$$\theta^{q\leftarrow\top}(p) = \begin{cases} \mathtt{tt} & \text{when } p = q \\ \theta(p) & \text{otherwise.} \end{cases} \qquad \theta^{q\leftarrow\bot}(p) = \begin{cases} \mathtt{ff} & \text{when } p = q \\ \theta(p) & \text{otherwise.} \end{cases}$$

propositions, then she has a strategy for every valuation of the propositions in At_i. However, this property is commonplace here.

The truth conditions are the usual ones for $\top$, $\bot$, negation and disjunction, plus:

$$\begin{array}{ll}
(\xi,\theta) \models q & \text{iff } \theta(q) = \mathtt{tt} \\
(\xi,\theta) \models c_{i,q} & \text{iff } i \in \xi(q) \\
(\xi,\theta) \models \langle q{\leftarrow}\tau\rangle\varphi & \text{iff } (\xi,\theta^{q\leftarrow\tau}) \models \varphi \\
(\xi,\theta) \models \langle i\overset{q}{\hookrightarrow}\rangle\varphi & \text{iff } (\xi^{i\overset{q}{\hookrightarrow}},\theta) \models \varphi \\
(\xi,\theta) \models \langle \overset{q}{\hookrightarrow} i\rangle\varphi & \text{iff } (\xi^{\overset{q}{\hookrightarrow} i},\theta) \models \varphi
\end{array}$$

$\mathsf{DL}^{\mathsf{PC}}$ validity and $\mathsf{DL}^{\mathsf{PC}}$ satisfiability are defined as usual.

Since the updates $\delta \in \{i\overset{q}{\hookrightarrow}, \overset{q}{\hookrightarrow}j, q{\leftarrow}\tau\}$ are always successful and deterministic, the formulas $\langle\delta\rangle\top$ and $\langle\delta\rangle\varphi \leftrightarrow \neg\langle\delta\rangle\neg\varphi$ are valid. We have as well that $\langle\overset{q}{\hookrightarrow}j\rangle c_{j,q}$ and $\langle i\overset{q}{\hookrightarrow}\rangle\neg c_{i,q}$ are valid.

Remark 1. Note that our truth conditions for the modal operators are slightly simpler than the original ones in terms of partitions of $\mathbb{P}$ [9, 15, 8]. In particular, they naturally account for (trivial) 'auto-delegations' $i\overset{p}{\hookrightarrow}i$, while the original semantics has to explicitly distinguish this case.

3.2 $\mathsf{DL}^{\mathsf{PC}}$: complexity and completeness

Proposition 1. *The following equivalences are $\mathsf{DL}^{\mathsf{PC}}$ valid.*

$$\begin{array}{ll}
\langle q{\leftarrow}\tau\rangle p & \leftrightarrow \begin{cases} p & \text{if } q \neq p \\ \tau & \text{if } q = p \end{cases} \\
\langle q{\leftarrow}\tau\rangle\top & \leftrightarrow \top \\
\langle q{\leftarrow}\tau\rangle\bot & \leftrightarrow \bot \\
\langle q{\leftarrow}\tau\rangle c_{i,p} & \leftrightarrow c_{i,p} \\
\langle q{\leftarrow}\tau\rangle\neg\varphi & \leftrightarrow \neg\langle q{\leftarrow}\tau\rangle\varphi \\
\langle q{\leftarrow}\tau\rangle(\varphi_1 \vee \varphi_2) & \leftrightarrow \langle q{\leftarrow}\tau\rangle\varphi_1 \vee \langle q{\leftarrow}\tau\rangle\varphi_2
\end{array}$$

$$\begin{array}{ll}
\langle j\overset{q}{\hookrightarrow}\rangle p & \leftrightarrow p \\
\langle j\overset{q}{\hookrightarrow}\rangle\top & \leftrightarrow \top \\
\langle j\overset{q}{\hookrightarrow}\rangle\bot & \leftrightarrow \bot \\
\langle j\overset{q}{\hookrightarrow}\rangle c_{i,p} & \leftrightarrow \begin{cases} c_{i,p} & \text{if } q \neq p \text{ or } j \neq i \\ \bot & \text{if } q = p \text{ and } j = i \end{cases} \\
\langle j\overset{q}{\hookrightarrow}\rangle\neg\varphi & \leftrightarrow \neg\langle j\overset{q}{\hookrightarrow}\rangle\varphi \\
\langle j\overset{q}{\hookrightarrow}\rangle(\varphi_1 \vee \varphi_2) & \leftrightarrow \langle j\overset{q}{\hookrightarrow}\rangle\varphi_1 \vee \langle j\overset{q}{\hookrightarrow}\rangle\varphi_2
\end{array}$$

$$
\begin{array}{ll}
\langle \overset{q}{\hookrightarrow} j\rangle p & \leftrightarrow p \\
\langle \overset{q}{\hookrightarrow} j\rangle \top & \leftrightarrow \top \\
\langle \overset{q}{\hookrightarrow} j\rangle \bot & \leftrightarrow \bot \\
\langle \overset{q}{\hookrightarrow} j\rangle c_{i,p} & \leftrightarrow \begin{cases} c_{i,p} & \textit{if } q \neq p \textit{ or } j \neq i \\ \top & \textit{if } q = p \textit{ and } j = i \end{cases} \\
\langle \overset{q}{\hookrightarrow} j\rangle \neg\varphi & \leftrightarrow \neg\langle \overset{q}{\hookrightarrow} j\rangle \varphi \\
\langle \overset{q}{\hookrightarrow} j\rangle (\varphi_1 \vee \varphi_2) & \leftrightarrow \langle \overset{q}{\hookrightarrow} j\rangle \varphi_1 \vee \langle \overset{q}{\hookrightarrow} j\rangle \varphi_2
\end{array}
$$

These equivalences provide a complete set of reduction axioms for $\langle q \leftarrow \tau\rangle$, $\langle i \overset{q}{\hookrightarrow}\rangle$ and $\langle \overset{q}{\hookrightarrow} j\rangle$. Call *red* the mapping which iteratively applies the above equivalences from the left to the right, starting from one of the innermost modal operators. *red* pushes the dynamic operators inside the formula, and finally eliminates them when facing an atomic formula. Each step increases the length of the formula by at most 3 (when distributing dynamic operators over disjunctions). The length of the reduced formula is therefore linear in the length of the original formula.

Note that although no dynamic operator occurs in $red(\varphi)$, it is not a formula of classical propositional logic because of the control atoms $c_{i,q}$. The next proposition shows how they can be dealt with.

Proposition 2. *Let φ be a formula in the language of $\mathsf{DL}^{\mathsf{PC}}$. Then*

1. *$red(\varphi)$ has no modal operators*
2. *$|red(\varphi)| \leq 3 \times |\varphi|$*
3. *$red(\varphi) \leftrightarrow \varphi$ is $\mathsf{DL}^{\mathsf{PC}}$ valid*
4. *$red(\varphi)$ is $\mathsf{DL}^{\mathsf{PC}}$ valid iff $red(\varphi)$ is valid in classical propositional logic, where the $c_{i,q}$ in $red(\varphi)$ are understood as propositional variables.*

Theorem 1. *Satisfiability in $\mathsf{DL}^{\mathsf{PC}}$ is NP-complete.*

PROOF. Hardness is the case because $\mathsf{DL}^{\mathsf{PC}}$ is a conservative extension of classical propositional logic: for every formula φ in the language of classical propositional logic, φ is classically valid if and only if φ is $\mathsf{DL}^{\mathsf{PC}}$ valid (where it is supposed that control atoms $c_{i,q}$ are not in the language of classical propositional logic).

As to membership, items 3 and 4 of Proposition 2 guarantee that φ is $\mathsf{DL}^{\mathsf{PC}}$ satisfiable iff $red(\varphi)$ is satisfiable in classical propositional logic. Moreover, $red(\varphi)$ is a polynomial reduction from CL-PC to classical propositional logic. ■

Theorem 2. *The validities of $\mathsf{DL}^{\mathsf{PC}}$ are completely axiomatized by*

- *some axiomatization of classical propositional logic*
- *the reduction axioms of Proposition 1*
- *the rule of equivalence*

$$\textit{from} \quad \varphi \leftrightarrow \varphi' \quad \textit{infer} \quad \langle\delta\rangle\varphi \leftrightarrow \langle\delta\rangle\varphi'$$

where δ is $\langle q \leftarrow \top\rangle$, $\langle q \leftarrow \bot\rangle$, $\langle i \overset{q}{\hookrightarrow}\rangle$ or $\langle \overset{q}{\hookrightarrow} j\rangle$.

PROOF. Soundness is guaranteed by Proposition 1, plus the fact that the inference rules preserve validity.

The completeness proof proceeds as follows. Suppose φ is $\mathsf{DL}^{\mathsf{PC}}$ valid. Then $red(\varphi)$ is classically valid due to Proposition 2. By the completeness of classical propositional logic, $red(\varphi)$ is also provable there. $\mathsf{DL}^{\mathsf{PC}}$ being a conservative extension of classical propositional logic, $red(\varphi)$ is provable in $\mathsf{DL}^{\mathsf{PC}}$, too. Then the formula φ must be provable in $\mathsf{DL}^{\mathsf{PC}}$, because the reduction axioms are part of our axiomatics and because the rule of substitution of equivalents is derivable.[5] ■

It is appealing to consider that assignments are performed by agents. Indeed, as for the moment, assignment are mere events. To reason about protocols of interacting agents, it is important to raise these events to the status of action. Authored assignments and control transfers are expressed in $\mathsf{DL}^{\mathsf{PC}}$ by the following abbreviations:

$$\langle q \overset{i}{\leftarrow} \tau \rangle \varphi \overset{\mathtt{def}}{=} c_{i,q} \wedge \langle q \leftarrow \tau \rangle \varphi$$
$$\langle i \overset{q}{\hookrightarrow} j \rangle \varphi \overset{\mathtt{def}}{=} c_{i,q} \wedge \langle i \overset{q}{\hookrightarrow} \rangle \langle \overset{q}{\hookrightarrow} j \rangle \varphi$$

'Spelling out' these abbreviations only polynomially increases the size of formulas. The modal operators that we are going to introduce in the next section are also going to be reducible to $\mathsf{DL}^{\mathsf{PC}}$, but not polynomially so (or rather, we don't know a polynomial reduction).

In the next section, we fully integrate these notions of agency into the logic.

4 Q-$\mathsf{DL}^{\mathsf{PC}}$: Quantified $\mathsf{DL}^{\mathsf{PC}}$

In this section we extend $\mathsf{DL}^{\mathsf{PC}}$ with two new constructs and modal operators. The modal operator $\langle \overset{q}{\hookrightarrow} J \rangle$ quantifies over control transfer targets in the set of agents J and $\langle q \overset{J}{\leftarrow} \rangle$ quantifies over both assignment authors in the set J and over truth values. We could as well introduce operators $\langle I \overset{q}{\hookrightarrow} \rangle$: the presentation would be symmetrical to that of $\langle \overset{q}{\hookrightarrow} J \rangle$.[6]

4.1 Language and semantics

The language of Q-$\mathsf{DL}^{\mathsf{PC}}$ is defined by adding formulas of the form $\langle q \overset{J}{\leftarrow} \rangle \varphi$ and $\langle \overset{q}{\hookrightarrow} J \rangle \varphi$ to the language of $\mathsf{DL}^{\mathsf{PC}}$, where q is a propositional variable in $\mathbb{P}$ and $J \subseteq \mathbb{A}$. $\langle q \overset{J}{\leftarrow} \rangle \varphi$ reads "the agents in J can ensure that φ by possibly changing the truth value of q", and $\langle \overset{q}{\hookrightarrow} J \rangle \varphi$ reads "φ holds after the transfer of q to some agent in J".

The truth conditions of the operators $\langle q \overset{J}{\leftarrow} \rangle$ and $\langle \overset{q}{\hookrightarrow} J \rangle$ are:

[5] The rule of substitution of equivalents is necessary in order to apply the reduction axioms 'deeply' inside formulas. It can be derived from the rules of equivalence for the classical connectives (that are derivable with classical propositional logic) and the rule of equivalence for δ of the axiomatics of $\mathsf{DL}^{\mathsf{PC}}$.

[6] We note that the strategy is similar to Borgo's in [4], who also proposes a reconstruction of coalition logic starting from a dynamic logic.

$$(\xi,\theta) \models \langle q \overset{J}{\leftarrow} \rangle \varphi \text{ iff } (\xi,\theta) \models \varphi \vee (\langle q \leftarrow \top \rangle \varphi \wedge \bigvee_{i\in J} c_{i,q}) \vee$$
$$(\langle q \leftarrow \bot \rangle \varphi \wedge \bigvee_{i\in J} c_{i,q})$$
$$(\xi,\theta) \models \langle \overset{q}{\hookrightarrow} J \rangle \varphi \text{ iff } (\xi,\theta) \models \bigvee_{i\in J} \langle \overset{q}{\hookrightarrow} i \rangle \varphi$$

Examples of valid equivalences are: $\langle \overset{q}{\hookrightarrow} \emptyset \rangle \varphi \leftrightarrow \bot$, $\langle q \overset{\emptyset}{\leftarrow} \rangle \varphi \leftrightarrow \varphi$, $\langle q \overset{\mathbb{A}}{\leftarrow} \rangle \varphi \leftrightarrow \langle q \leftarrow \top \rangle \varphi \vee \langle q \leftarrow \bot \rangle \varphi$, $\langle q \leftarrow \top \rangle \varphi \leftrightarrow \langle q \overset{\mathbb{A}}{\leftarrow} \rangle (q \wedge \varphi)$, and $\langle q \leftarrow \bot \rangle \varphi \leftrightarrow \langle q \overset{\mathbb{A}}{\leftarrow} \rangle (\neg q \wedge \varphi)$.

We observe that the program $q\leftarrow$ that we discussed in the introduction can be defined as $q \overset{\mathbb{A}}{\leftarrow}$. We also observe that $\langle q \overset{\{i\}}{\leftarrow} \rangle \top$ is logically equivalent to $\top$, while both $\langle q \overset{i}{\leftarrow} \top \rangle \top$ and $\langle q \overset{i}{\leftarrow} \bot \rangle \top$ are logically equivalent to $c_{i,q}$. We also observe that $c_{i,p} \leftrightarrow (\langle \overset{\{i\}}{\leftarrow} \rangle p \wedge \langle \overset{\{i\}}{\leftarrow} \rangle \neg p)$.

The next proposition is going to be useful to prove several results.

Proposition 3. *Let φ be a Q-DL^{PC} formula. Let (ξ_1,θ_1) and (ξ_2,θ_2) be PC models agreeing on every variable outside φ, i.e. such that for every $q \notin \mathbb{P}_\varphi$, $\xi_1(q) = \xi_2(q)$ and $\theta_1(q) = \theta_2(q)$. Then $(\xi_1,\theta_1) \models \varphi$ iff $(\xi_2,\theta_2) \models \varphi$.*

4.2 Two abbreviations in Q-DL$^{\mathsf{PC}}$

Let $P = \{q_1, \cdots, q_n\}$ be a finite set of propositional variables. We define:

$$\langle P \overset{J}{\leftarrow} \rangle \varphi \overset{\text{def}}{=} \langle q_1 \overset{J}{\leftarrow} \rangle \cdots \langle q_n \overset{J}{\leftarrow} \rangle \varphi$$
$$\langle \overset{P}{\hookrightarrow} J \rangle \varphi \overset{\text{def}}{=} \langle \overset{q_1}{\hookrightarrow} J \rangle \cdots \langle \overset{q_n}{\hookrightarrow} J \rangle \varphi$$

For the case $n = 0$ we suppose that $\langle \emptyset \overset{J}{\leftarrow} \rangle \varphi$ and $\langle \overset{\emptyset}{\hookrightarrow} J \rangle \varphi$ are both equal to φ. Just as for assignments with authors $q \overset{i}{\leftarrow} \top$ and $q \overset{i}{\leftarrow} \bot$, expanding these abbreviations only polynomially increases the size of the formula (precisely, the size of the rewritten formula is quadratic in the size of the original formula).

Example 2. Let us take up our running example. Consider the following formulas.

$$\begin{aligned}
\mathsf{\leq 4} &= \textstyle\bigwedge_{m\in RV} \bigvee_{X\subseteq PP, |X|\geq |PP|-4} \bigwedge_{n\in X, q\in\{p_n^1,p_n^2,p_n^3\}} \neg c_{m,q} \\
\mathsf{no2} &= \textstyle\bigwedge_{m\in RV} \bigwedge_{n\in PP} \bigvee_{\{q,r\}\subset\{p_n^1,p_n^2,p_n^3\}} (\neg c_{m,q} \wedge \neg c_{m,r}) \\
\mathsf{gets3} &= \textstyle\bigwedge_{q\in\mathbb{P}} \bigvee_{m\in RV} c_{m,q} \\
\mathsf{noExtr} &= \textstyle\bigwedge_{m\in RV} \bigvee_{\{q,r\}\subseteq\mathbb{P}} (c_{m,q} \wedge c_{m,r} \wedge q \wedge \neg r) \\
\mathsf{acc}_n &= \textstyle\bigvee_{\{q,r\}\subseteq\{p_n^1,p_n^2,p_n^3\}} (q \wedge r) \\
\mathsf{25\%} &= \textstyle\bigwedge_{X\subset PP, |X|\geq |PP|/4} \bigvee_{n\in X} \neg \mathsf{acc}_n
\end{aligned}$$

They express that: each reviewer gets at most four papers ($\leq$4); no reviewer gets a paper twice (no2); each paper gets three reviewers (gets3); no reviewer can have only positive or only negative opinions (noExtr); paper n is accepted if at least two opinions are positive (acc_n); at most 25% of the papers can be accepted (25%).

We can verify that the program chair can distribute the papers according to the constraints and such that the acceptance rate can be obtained, by checking Q-DL$^{\mathsf{PC}}$ validity of the following formula:

$$\bigwedge_{q\in\mathbb{P}} \left(c_{Chair,q} \wedge \bigwedge_{R\in RV} c_{R,q} \right) \longrightarrow \langle \overset{\mathbb{P}}{\hookrightarrow} RV \rangle (\mathsf{\leq 4} \wedge \mathsf{no2} \wedge \mathsf{gets3} \wedge \langle \mathbb{P} \overset{RV}{\leftarrow} \rangle (\mathsf{noExtr} \wedge \mathsf{25\%}))$$

4.3 Q-DL^{PC}: complexity and completeness

Theorem 3. *The validities of Q-DL^{PC} are completely axiomatized by*

- *the axiomatization of DL^{PC} of Theorem 2*
- *the following axiom schemas:*

$$\langle q \overset{J}{\leftarrow} \rangle\varphi \leftrightarrow \varphi \vee (\langle q{\leftarrow}\top\rangle\varphi \wedge \bigvee_{i\in J} c_{i,q}) \vee (\langle q{\leftarrow}\bot\rangle\varphi \wedge \bigvee_{i\in J} c_{i,q})$$
$$\langle \overset{q}{\hookrightarrow} J\rangle\varphi \leftrightarrow \bigvee_{i\in J} \langle \overset{q}{\hookrightarrow} i\rangle\varphi$$

PROOF. The proof follows the lines of that of Theorem 2, for the appropriately defined reduction mapping *red*. ∎

The axiom schemas of Theorem 3 allow to rewrite every Q-DL^{PC} formula to a DL^{PC} formula. Combining this with a decision procedure for DL^{PC} we obtain a decision procedure for Q-DL^{PC}. However the rewriting step increases the length of the formula exponentially, and the decision procedure runs in exponential space. One can do better:

Theorem 4. *The model checking problem for Q-DL^{PC} is PSPACE-complete.*

PROOF. Hardness can be proved like for CL-PC by reducing QBF satisfiability [10].

It is easy to adapt the algorithm for model checking δCL-PC of [15] in order to deal with our operators. The procedure still does not require more that $|\varphi|$ recursive calls, where φ is the input formula, and every call requires to store only one model at a time. Hence, the algorithm runs in polynomial space. ∎

Theorem 5. *Satisfiability in Q-DL^{PC} is PSPACE-complete.*

PROOF. Hardness can again be proved like for CL-PC [10].

As for easiness, first observe that like CL-PC and δCL-PC, Q-DL^{PC} has the small model property. Hence, just as in [15], given a formula φ we guess a model (ξ, θ) and check whether $(\xi, \theta) \models \varphi$. According to Theorem 4 it takes space polynomial in $|\varphi|$ to check whether $(\xi, \theta) \models \varphi$. As we had guessed (ξ, θ), satisfiability checking is in NPSPACE = PSPACE. ∎

The proofs of Theorem 4 and Theorem 5 are analogous to those of the complexity results for CL-PC in [10]. The only difference is that in CL-PC a model for a formula φ can be encoded in space linear in the length of φ, while in Q-DL^{PC}, a model for a formula φ can be encoded in space quadratic in the length of φ. (This is because a propositional variable is not controlled by a single agent but by a set of agents.)

4.4 Defining van der Hoek and Wooldridge's coalition modality in Q-DL^{PC}

We now consider the coalition operators of CL-PC. These are normal modal operators that we here write $\langle \overset{J}{\leftarrow} \rangle$.[7] They intend to grasp a local (or contingent) ability. When the set of propositional variables $\mathbb{P}$ is finite then the semantics is the same as that of the above $\langle \mathbb{P} \overset{J}{\leftarrow} \rangle$. As we allow for $\mathbb{P}$ to be infinite we have to consider $\langle \overset{J}{\leftarrow} \rangle$ to be primitive. The accessibility relations are defined as follows:

[7] The original notation is $\Diamond_J$.

$$(\xi,\theta)R_{\overset{J}{\leftarrow}}(\xi',\theta') \text{ iff } \xi' = \xi, \text{ and if } \xi(p) \cap J = \emptyset \text{ then } \theta'(p) = \theta(p)$$

and the truth condition is:

$$(\xi,\theta) \models \langle\overset{J}{\leftarrow}\rangle\varphi \text{ iff there is } (\xi',\theta') \text{ such that } (\xi,\theta)R_{\overset{J}{\leftarrow}}(\xi',\theta') \text{ and } (\xi',\theta') \models \varphi$$

The next result shows that actually there was no need to add the primitive $\langle\overset{J}{\leftarrow}\rangle$: we may restrict the variables that are assigned by J, to the set $\mathbb{P}_\varphi$ of propositional variables occurring in φ.

Proposition 4. *The schema* $\langle\overset{J}{\leftarrow}\rangle\varphi \leftrightarrow \langle\mathbb{P}_\varphi\overset{J}{\leftarrow}\rangle\varphi$ *is* Q-DL^{PC} *valid.*

PROOF. The right-to-left direction is straightforward.

For the other direction, suppose $(\xi,\theta) \models \langle\overset{J}{\leftarrow}\rangle\varphi$. Hence there is a model (ξ',θ') such that $(\xi,\theta)R_{\overset{J}{\leftarrow}}(\xi',\theta')$ and $(\xi',\theta') \models \varphi$. Observe that $\xi' = \xi$. Let (ξ'',θ'') be such that $\xi'' = \xi'$ and

$$\theta''(q) = \begin{cases} \theta'(q) & \text{if } q \in \mathbb{P}_\varphi \\ \theta(q) & \text{if } q \notin \mathbb{P}_\varphi \end{cases}$$

We have $(\xi,\theta)R_{\overset{J}{\leftarrow}}(\xi'',\theta'')$, and by Proposition 3 we have $(\xi'',\theta'') \models \varphi$ iff $(\xi',\theta') \models \varphi$; therefore $(\xi,\theta) \models \langle\mathbb{P}_\varphi\overset{J}{\leftarrow}\rangle\varphi$. ■

4.5 Defining Pauly's coalition modality in Q-DL^{PC}

As we said in the introduction, the original motivation of the inventors of CL-PC was to reconstruct Pauly's Coalition Logic CL. There, the CL formula $\langle[J]\rangle\varphi$ reads "the coalition J can ensure that φ holds next, *whatever the other agents choose to do*".[8] Van der Hoek and Wooldridge proposed to identify the CL formula $\langle[J]\rangle\varphi$ with the CL-PC formula $\langle\overset{J}{\leftarrow}\rangle[\overset{\bar{J}}{\leftarrow}]\varphi$ (where as usual in modal logic $[\overset{J}{\leftarrow}]\varphi$ abbreviates $\neg\langle\overset{J}{\leftarrow}\rangle\neg\varphi$). It represents the so-called $\exists\forall$-*ability* of J for φ, and generally called α-*ability* in social choice theory [1]. As van der Hoek and Wooldridge point out, the δCL-PC formula $[\overset{\bar{J}}{\leftarrow}]\langle\overset{J}{\leftarrow}\rangle\varphi$ expresses $\forall\exists$-*ability*, alias β-*ability*.

It follows from the above Proposition 4 that $\langle\overset{J}{\leftarrow}\rangle[\overset{\bar{J}}{\leftarrow}]\varphi \leftrightarrow \langle\mathbb{P}_\varphi\overset{J}{\leftarrow}\rangle[\mathbb{P}_\varphi\overset{\bar{J}}{\leftarrow}]\varphi$ is Q-DL^{PC} valid. We are therefore entitled to consider from now on that $\langle[J]\rangle\varphi$ is an abbreviation of $\langle\mathbb{P}_\varphi\overset{J}{\leftarrow}\rangle[\mathbb{P}_\varphi\overset{\bar{J}}{\leftarrow}]\varphi$.

5 DEL^{PC}: dynamic epistemic logic of propositional control

We mentioned before that the assignment operator was introduced in the context of dynamic epistemic logics, which are extensions of epistemic logic by dynamic operators

[8] The original notation is $[J]\varphi$; van der Hoek and Wooldridge use $\langle\langle J\rangle\rangle\varphi$.

such as assignments and announcements. In the same spirit we now extend our framework by modal operators of knowledge and call the logic *dynamic epistemic logic of propositional control*, abbreviated $\mathsf{DEL}^{\mathsf{PC}}$.

We will assume that assignments and control transfers are *public* events and are therefore fully observable by the agents.

We are going to give an axiomatization and a decision procedure for our extension.

5.1 Language and semantics

We consider the extension of the language of $\mathsf{DL}^{\mathsf{PC}}$ by modal operators of knowledge K_i, one per agent $i \in \mathbb{A}$, and by modal operators $\langle\varphi!\rangle$ of truthful public announcement of φ, where φ is any formula. $K_i\varphi$ is read "i knows that φ", and $\langle\varphi!\rangle\psi$ is read "the announcement of φ is possible, and ψ holds afterwards".

To interpret the epistemic operators we move from PC models (ξ, θ) to *epistemic PC models* of the form $M = (W, \sim, \varXi, \varTheta)$, where

- W is a nonempty set of possible worlds
- $\sim : \mathbb{A} \longrightarrow (W \times W)$ associates an equivalence relation $\sim_i$ to every agent i
- $\varXi : W \longrightarrow (\mathbb{P} \longrightarrow 2^{\mathbb{A}})$ associates allocations to possible worlds
- $\varTheta : W \longrightarrow (\mathbb{P} \longrightarrow \{\mathtt{tt}, \mathtt{ff}\})$ associates valuations to possible worlds

It is convenient to write $\varXi_w(p)$ and $\varTheta_w(p)$ instead of $\varXi(w)(p)$ and $\varTheta(w)(p)$. Every couple $(\varXi_w, \varTheta_w)$ is a PC model.

We now define the updates on an epistemic model. For conciseness we introduce two notations. For $\delta \in \{i\overset{q}{\hookrightarrow}, \overset{q}{\hookrightarrow}i\}$, we note $\varXi^{\delta}$ the function that maps every state v to the updated allocation $\varXi^{\delta}_v$. Also, $\varTheta^{q\leftarrow\tau}$ is the function that maps every v to the valuation $\varTheta^{q\leftarrow\tau}_v$.

Let $M = (W, \sim, \varXi, \varTheta)$ be a pointed model, and let $w \in W$. Its updates are defined as follows.

$$
\begin{aligned}
M^{i\overset{q}{\hookrightarrow}} &= (W, \sim, \varXi^{i\overset{q}{\hookrightarrow}}, \varTheta) \\
M^{\overset{q}{\hookrightarrow}i} &= (W, \sim, \varXi^{\overset{q}{\hookrightarrow}i}, \varTheta) \\
M^{q\leftarrow\tau} &= (W, \sim, \varXi, \varTheta^{q\leftarrow\tau}) \\
M^{\psi!} &= (W', \sim', \varXi', \varTheta') \text{ such that } \begin{cases} W' = \{v \in W \ : \ M, w \models \psi\} \\ \sim' = \sim \cap (W' \times W') \\ \varXi' = \varXi|_{W'} \\ \varTheta' = \varTheta|_{W'} \end{cases}
\end{aligned}
$$

According to our semantics, assignments and control transfers are public: when one of these events occurs then every agent updates his epistemic possibilities accordingly.

The truth conditions also have to be adapted and extended accordingly; in particular:

$$
\begin{aligned}
&M, w \models q && \text{iff } \varTheta_w(q) = \mathtt{tt}, \text{ for } q \in \mathbb{P} \\
&M, w \models c_{i,q} && \text{iff } i \in \varXi_w(q) \\
&M, w \models K_i\varphi && \text{iff } M, v \models \varphi \text{ for every } v \text{ such that } w \sim_i v
\end{aligned}
$$

Moreover, for every program $\delta \in \{i \overset{q}{\hookrightarrow}, \overset{q}{\hookrightarrow} i, q \leftarrow \tau\}$ we define:

$$M, w \models \langle \delta \rangle \varphi \quad \text{iff } M^{\delta}, w \models \varphi$$
$$M, w \models \langle \psi ! \rangle \varphi \text{ iff } M, w \models \psi \text{ and } M^{\psi !}, w \models \varphi$$

Let us call the resulting logic $\mathsf{DEL}^{\mathsf{PC}}$. Examples of $\mathsf{DEL}^{\mathsf{PC}}$ validities are $\langle q \leftarrow \bot \rangle K_i \neg q$, $\langle i \overset{q}{\hookrightarrow} \rangle K_i \neg c_{i,q}$, and $\langle \overset{q}{\hookrightarrow} j \rangle K_i c_{j,q}$, highlighting that assignments and control transfer are public events.

Remark 2. According to our semantics, agent i does not necessarily know whether p is allocated to j or not, and so even if $i = j$. In formulas, $c_{i,q} \wedge \neg K_i c_{i,q}$ is satisfiable. One could however easily guarantee that agents are aware of what is or is not allocated to them, by imposing the following constraint on models: if $w \sim_i w'$ then for every $q \in \mathbb{P}$, $i \in \Xi_w(q)$ iff $i \in \Xi_{w'}(q)$. Such models validate $c_{i,q} \to K_i c_{i,q}$.

5.2 Completeness and complexity

We are now able to formulate reduction axioms for our logic.

Proposition 5. *The reduction axioms of Proposition 1 are $\mathsf{DEL}^{\mathsf{PC}}$ valid, as well as the following equivalences:*

$$\begin{aligned}
\langle q \leftarrow \tau \rangle K_i \varphi &\leftrightarrow K_i \langle q \leftarrow \tau \rangle \varphi \\
\langle j \overset{q}{\hookrightarrow} \rangle K_i \varphi &\leftrightarrow K_i \langle j \overset{q}{\hookrightarrow} \rangle \varphi \\
\langle \overset{q}{\hookrightarrow} j \rangle K_i \varphi &\leftrightarrow K_i \langle \overset{q}{\hookrightarrow} j \rangle \varphi \\
\langle \psi ! \rangle \varphi &\leftrightarrow \psi \wedge \varphi && \text{if } \varphi \text{ is of the form } q, \top, \bot, \text{ or } c_{i,q} \\
\langle \psi ! \rangle \neg \varphi &\leftrightarrow \psi \wedge \neg \langle \psi ! \rangle \varphi \\
\langle \psi ! \rangle (\varphi_1 \vee \varphi_2) &\leftrightarrow \langle \psi ! \rangle \varphi_1 \vee \langle \psi ! \rangle \varphi_2 \\
\langle \psi ! \rangle K_i \varphi &\leftrightarrow \psi \wedge K_i \neg \langle \psi ! \rangle \neg \varphi
\end{aligned}$$

In the 4th equivalence, φ may more generally be any formula without modal operators.

The axiom schemas of Proposition 5 together with the reduction axioms of $\mathsf{DL}^{\mathsf{PC}}$ of Theorem 3 provide a complete set of axioms for the reduction of $\mathsf{DEL}^{\mathsf{PC}}$ to standard epistemic logic $\mathsf{S5}_n$. Call $red(\varphi)$ the resulting formula.

Theorem 6. *Let φ be a formula in the language of $\mathsf{DEL}^{\mathsf{PC}}$. Then*

1. *$red(\varphi)$ has no modal operators other than epistemic operators*
2. *$red(\varphi) \leftrightarrow \varphi$ is $\mathsf{DEL}^{\mathsf{PC}}$ valid*
3. *$red(\varphi)$ is $\mathsf{DEL}^{\mathsf{PC}}$ valid iff $red(\varphi)$ is $\mathsf{S5}_n$ valid, where the $c_{i,q}$ in $red(\varphi)$ are understood as propositional variables.*

As before, the reduction axioms allow to show completeness.

Theorem 7. *The validities of $\mathsf{DEL}^{\mathsf{PC}}$ are axiomatized by*

- *the axioms and inference rules of $\mathsf{DL}^{\mathsf{PC}}$ of Theorem 1*
- *the axioms and inference rules of $\mathsf{S5}_n$, for every modal operator K_i*
- *the axiom schemas of Proposition 5*

– *the rules of equivalence for* $\langle\psi!\rangle$*:*

$$\begin{array}{llll} \textit{from} & \varphi \leftrightarrow \varphi' & \textit{infer} & \langle\psi!\rangle\varphi \leftrightarrow \langle\psi!\rangle\varphi' \\ \textit{from} & \psi \leftrightarrow \psi' & \textit{infer} & \langle\psi!\rangle\varphi \leftrightarrow \langle\psi'!\rangle\varphi \end{array}$$

PROOF. The proof uses the reduction axioms and then follows the lines of that for Theorem 2. ■

While for $\mathsf{DL}^{\mathsf{PC}}$ we were able to show that the length of the reduced formula is polynomial in the length of the original formula (Proposition 2), this is no longer the case for $\mathsf{DEL}^{\mathsf{PC}}$. This is due to the form of the reduction axioms for announcements of Proposition 5 where φ occurs twice on right hand sides (cf. [12]). A reduction-based decision procedure is therefore suboptimal. However, reduction allows to establish completeness and decidability.

Theorem 8. *Satisfiability in* $\mathsf{DEL}^{\mathsf{PC}}$ *is PSPACE-complete if there are at least two agents, and NP-complete if there is only one agent.*

PROOF. Hardness is the case because $\mathsf{DL}^{\mathsf{PC}}$ is a conservative extension of epistemic logic (whose satisfiability problem is NP-hard for one agent and PSPACE-hard for more than one agent).

Membership can be proved by applying the abbreviation technique of [12]. ■

5.3 Authored assignment

In the end of Section 3 we had proposed the following definition $\langle i \overset{q}{\hookrightarrow} j\rangle\varphi \overset{\mathtt{def}}{=} c_{i,q} \wedge \langle i \overset{q}{\hookrightarrow}\rangle\langle \overset{q}{\hookrightarrow} j\rangle\varphi$. It identified the control transfer programs $i \overset{q}{\hookrightarrow} j$ of $\delta\mathsf{CL}\text{-}\mathsf{PC}$, with a mere *test* of $c_{i,q}$ followed by $i \overset{q}{\hookrightarrow}$ and $\overset{q}{\hookrightarrow} j$. Such an abbreviation is no longer intuitive in $\mathsf{DEL}^{\mathsf{PC}}$ because the events are public by assumption. The occurrence of the control transfer being public, every agent can eliminate her epistemic possibilities where *i did not* control q right before the transfer occurred. Hence, $i \overset{q}{\hookrightarrow} j$ should amount to the *public announcement* of $c_{i,q}$ followed by $i \overset{q}{\hookrightarrow}$ and $\overset{q}{\hookrightarrow} j$. This leads to the following (re)definition that suits better.

$$\langle i \overset{q}{\hookrightarrow} j\rangle\varphi \overset{\mathtt{def}}{=} \langle c_{i,q}!\rangle\langle i \overset{q}{\hookrightarrow}\rangle\langle \overset{q}{\hookrightarrow} j\rangle\varphi$$

Similarly, we had identified i's assignment of q to τ, $q \overset{i}{\leftarrow} \tau$, with a test of $c_{i,q}$ followed be the execution of the program $q \leftarrow \tau$. Again, the occurrence of an action $q \overset{i}{\leftarrow} \bot$ being public, every agent eliminates his possibility where i does not control q. In formulas, we expect $\langle q \overset{i}{\leftarrow} \bot\rangle\neg K_j c_{i,j}$ to be unsatisfiable. However, with the abbreviation of Section 3 this formula would be satisfiable. Since we assumed that events are public, $q \overset{i}{\leftarrow} \tau$ should actually amount to the public announcement of $c_{i,q}$ followed by the public assignment $q \leftarrow \tau$. This leads to the following (re)definition that suits better.

$$\langle q \overset{i}{\leftarrow} \tau\rangle_e\varphi \overset{\mathtt{def}}{=} \langle c_{i,q}!\rangle\langle q \leftarrow \tau\rangle\varphi$$

The reader may check that $\langle q \overset{i}{\leftarrow} \bot\rangle\neg K_j c_{i,j}$ is unsatisfiable in $\mathsf{DEL}^{\mathsf{PC}}$.

6 Conclusion and perspectives

We have relaxed the original assumption of exclusive and actual control of van der Hoek and Wooldridge's coalition logic of propositional control CL-PC and have shown that their logic can be embedded in ours. We have moreover extended the existing logics of propositional control by several concepts stemming from dynamic epistemic logics: knowledge, assignments, and announcements. We have shown how the resulting logics relate to CL-PC and its extension δCL-PC. We have also established their axiomatization and their complexity for satisfiability and model checking.

The remaining of the section is devoted to the sketch of some possible extensions of our logic.

The most obvious variant is to integrate PDL-style constructs to $\mathsf{DL}^{\mathsf{PC}}$. In a nutshell, it allows the following direct definitions of some programs we have considered in this paper.

$$\begin{aligned}
q\overset{J}{\leftarrow} &\overset{\text{def}}{=} \top? + \Big((\textstyle\bigvee_{i\in J} c_{i,q})?; (q{\leftarrow}\top + q{\leftarrow}\bot)\Big)\\
\overset{q}{\hookrightarrow}\{j_1 \ldots j_n\} &\overset{\text{def}}{=} \overset{q}{\hookrightarrow}j_1 + \ldots + \overset{q}{\hookrightarrow}j_n\\
\{q_i \ldots q_n\}\overset{J}{\leftarrow} &\overset{\text{def}}{=} q_1\overset{J}{\leftarrow}; \ldots ; q_n\overset{J}{\leftarrow}\\
\overset{\{q_1\ldots q_n\}}{\hookrightarrow}J &\overset{\text{def}}{=} \overset{q_1}{\hookrightarrow}J; \ldots ; \overset{q_n}{\hookrightarrow}J\\
i\overset{q}{\hookrightarrow}j &\overset{\text{def}}{=} c_{i,q}?; i\overset{q}{\hookrightarrow}; \overset{q}{\hookrightarrow}j
\end{aligned}$$

Being a dynamic logic, the logic can capture the usual instructions of structured programming. For instance, for every complex program δ_1 and δ_2:

$$\begin{aligned}
\texttt{if } \varphi \texttt{ then } \delta_1 \texttt{ else } \delta_2 &\overset{\text{def}}{=} (\varphi?; \delta_1) \cup (\neg\varphi?; \delta_2)\\
\texttt{while } \varphi \texttt{ do } \delta_1 &\overset{\text{def}}{=} (\varphi?; \delta_1)^*; \neg\varphi
\end{aligned}$$

We conjecture that the resulting logic is PSPACE-complete, too. That is, it would be no more complex than van der Hoek et al.'s δCL-PC, despite a greater expressivity and the use of more general models.

Another straightforward generalization of PC models would be to *distinguish between making a variable q false and making it true*. This is related to Gerbrandy's notion of positive and negative control. In order to take that into account the function ξ has to be split up into ξ^+ and ξ^-: $\xi^+(q)$ is the set of those agents which may make q true, and $\xi^-(q)$ is the set of those agents which may make q false. In the language one may then have control atoms ${c_{i,q}}^+$ and ${c_{i,q}}^-$ which are interpreted as expected. Moreover, one may have modal operators $i\overset{+q}{\hookrightarrow}$, $i\overset{-q}{\hookrightarrow}$, $\overset{+q}{\hookrightarrow}i$, and $\overset{-q}{\hookrightarrow}i$ of loosing or obtaining positive or negative control. This then may be taken into account by defining e.g. authored assignment as $\langle q\overset{i}{\leftarrow}\top\rangle\varphi \overset{\text{def}}{=} {c_{i,q}}^+ \wedge \langle q{\leftarrow}\top\rangle\varphi$.

Concerning the epistemic extension we observe that while the extension by public announcements is technically straightforward it does not account for *announcements that are made by agents*. In a first approach one might identify the announcement of φ by i with the public announcement of $K_i\varphi$; however, a full analysis requires more work.

In a similar spirit, one might extend our logic by *event models* [3, 2, 6], which account for incomplete (and even erroneous) perception of events by agents. This should be possible without difficulties. As a teaser, it would allow to adequately handle protocols with intricate epistemic aspects such as for instance a variant of Example 1 with double blind reviewing, or a more realistic setting where the protocol is specified in a way such that a reviewer neither know what the allocations of the other members of the committee are, nor which opinions were already expressed about a paper.

Acknowledgements

Thanks for several suggestions are due to Guillaume Aucher, Pierre Marquis, Leon van der Torre and Emil Weydert. We are also grateful to the reviewers of LIS@ESSLLI for their comments that helped to improve the paper. Andreas Herzig's research on this paper is partly funded by the ANR project *Social trust analysis and formalization* (ForTrust). Nicolas Troquard's is supported by the EPSRC grant EP/E061397/1.

References

1. Joseph Abdou and Hans Keiding. *Effectivity functions in social choice*. Kluwer Academic, 1991.
2. Alexandru Baltag and Lawrence S. Moss. Logics for epistemic programs. *Synthese*, 139(2):165–224, 2004.
3. Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Proc. TARK'98*, pages 43–56, 1998.
4. Stefano Borgo. Coalitions in action logic. In *Proc. IJCAI'07*, pages 1822–1827, 2007.
5. Hans P. van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. Dynamic epistemic logic with assignment. In *Proc. AAMAS'05*, pages 141–148, 2005.
6. Hans P. van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. Kluwer Academic Publishers, 2007.
7. Jelle Gerbrandy. Logics of propositional control. In Hideyuki Nakashima, Michael P. Wellman, Gerhard Weiss, and Peter Stone, editors, *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, pages 193–200. ACM, 2006.
8. Wiebe van der Hoek, Dirk Walther, and Michael Wooldridge. On the logic of cooperation and the transfer of control. *J. of AI Research (JAIR)*, 37:437–477, 2010.
9. Wiebe van der Hoek and Michael Wooldridge. On the dynamics of delegation, cooperation and control: a logical account. In *Proc. AAMAS'05*, 2005.
10. Wiebe van der Hoek and Michael Wooldridge. On the logic of cooperation and propositional control. *Artif. Intell.*, 164(1-2):81–119, 2005.
11. Barteld Kooi. Expressivity and completeness for public update logic via reduction axioms. *Journal of Applied Non-Classical Logics*, 17(2):231–253, 2007.
12. Carsten Lutz. Complexity and succintness of public announcement logic. In *Proc. AAMAS'06*, pages 137–144, 2006.
13. Nicolas Troquard, Wiebe van der Hoek, and Michael Wooldridge. A logic of propositional control for truthful implementations. In *Proc. TARK'09*, pages 237–246, 2009.
14. Nicolas Troquard, Wiebe van der Hoek, and Michael Wooldridge. A logic of games and propositional control. In *Proc. AAMAS'09*, pages 961–968, 2009.
15. Dirk Walther. *Strategic Logics: Complexity, Completeness and Expressivity*. PhD thesis, University of Liverpool, 2007.