# Types of Rights in Interacting Two-Party Systems: A Formal Analysis

Gordon J. Pace[1] and Fernando Schapachnik[2]*

[1] University of Malta
gordon.pace@um.edu.mt
[2] Departamento de Computación, FCEyN,
Universidad de Buenos Aires, Buenos Aires, Argentina
fschapachnik@dc.uba.ar

**Abstract.** We present a formalization of Kanger's types of rights in the context of interacting two-party systems, such as contracts. We show that in this setting basic rights such as *claim*, *freedom*, *power* and *immunity* can be expressed in terms of (possibly negated) permissions and obligations over presence or absense of actions, and that the set of atomic type rights is different from Kanger's original proposal.

## 1 Introduction

Deontic modalities such as permission and obligation have been debated exhaustively in the literature, and although a final consensus has not been achieved, there is at least agreement over their basic properties. This is not the case with more intricate concepts such as Hohfeld's *claim right*, *power*, *freedom* and *immunity* [1].

Kanger *et al.* [2] attempted to clarify Hohfeld's modalities, but a lack of formal underlying semantics somewhat limits the work. Other work, surveyed in Section 6, addressed this aspect, yet not always fully formalising the intricate modalities and other derivative ones, such as intention and causality, which arise in this context. Part of the difficulty, we believe, can be addressed if the context is limited. We restrict ourselves to a setting that is both specific and interesting: interacting two-party systems, also commonly known as *contracts*. Contracts are prevalent enough so their analysis becomes of practical importance, yet restrictive enough so there is a clear boundary for the analysis. In such systems, the interactive nature of the parties gives rise to potential cooperation and interference — allowing us to reach conclusions separately for both the individual parties and the result of their combination.

Furthermore, in Kanger *et al.* [2] rights are state-based, and dependant on a notion of causality which interacts with directed rights in a non-obvious manner. For instance, it is not immediately clear how a statement such as *'it shall be that party $p$ causes $S$'* is to be interpreted in a system where, for instance, the other party can interfere with $p$'s intention to bring about state $S$. However, in computer science, concurrent and synchronous composition have been studied for a number of decades, and address these notions from an action-based perspective.

* Partially founded by UBACyT 20020090200116 and UBACyT 20020100200103.

In [3] we have looked at how synchrony can be applied in a contract setting, using a formal automaton-based model of interacting two-party systems in which the parties synchronise over a set of actions. In this paper we extend the model to be able to study how Kanger's rights apply in such a setting. The synchronous nature of composition adopted, which forces the parties to agree on actions to perform, brings about a setting subtly different from the one originally presented in Kanger *et al.* [2], in particular because rights and obligations affect both parties. E.g., if a party has an obligation to perform a particular action, then the other party must provide her with a way of achieving this. Just as in Kanger *et al.*, we proceed to study the different compound types of rights in this setting. Unsurprisingly, our different, mostly stronger, modalities bring to light further conflicts in contract clauses, and thus induce a different set of possible rights.
The main contributions of this paper are:

- Extending the notion of two-party system contracts to deal with (i) absence of actions; (ii) mutually exclusive actions; (iii) conflicts.
- Giving formal semantics to Kanger's types of right in the context of action-based, interacting two-party systems.
- Showing that the number of atomic types (maximally consistent sets of rights) is reduced in this context.

The rest of the paper is organised as follows. Next section formalises our notions of automata, deontic operators, contracts and contracts' strength, which allows us to show, in Section 4 that some contracts cannot be satisfied at the same time and thus lead to a conflict. In Section 3 we interpret Kanger's work in the setting of interacting two-party systems and compare Kanger's modalities strength diagram with ours, explaining why they differ, comparing atomic types under both proposals in Section 5. Finally, in Section 6 we discuss related work, and conclude in Section 7.

## 2 Regulated Two-Party Systems

### 2.1 An Automata-Based View

To enable direct reasoning about contracts, one requires a model in which the two parties somehow interact to agree on which actions to perform. We use the notion of synchronous composition [4] to model such behaviour. Furthermore, to be able to deal with concurrent obligations (for instance, one party being obliged to perform one action and the other being obliged to perform another), we adopt multi-action labels on transitions, since if we do not, it would be impossible not to violate a contract in which both parties have different obligations at the same time.

**Definition 1.** *A multi-action automaton $S$ is a tuple $\langle \Sigma,\ Q,\ q0,\ \rightarrow\rangle$, where $\Sigma$ is the alphabet of actions, $Q$ is the set of states, $q0 \in Q$ is the initial state and $\rightarrow \subseteq Q \times 2^{\Sigma} \times Q$ is the transition relation. We will write $q \xrightarrow{A} q'$ for $(q, A, q') \in \rightarrow$, $next(q)$ to be the set of target state and action set pairs of transitions outgoing from $q$ (defined to be $\{(A, q') \mid q \xrightarrow{A} q'\}$) and $acts(q)$ to be the set of all action sets on the outgoing transitions from*

*$q$ (defined to be $\{A \mid \exists q' \cdot q \xrightarrow{A} q'\}$). We say that an automaton is total, if for every $q \in Q$ and $A \subseteq \Sigma$, there is a $q' \in Q$ such that $q \xrightarrow{A} q'$.*

*The synchronous composition of two automata $S_i = \langle Q_i,\ q0_i,\ \rightarrow_i \rangle$ for $i \in \{1, 2\}$ (both with alphabet $\Sigma$) synchronising over alphabet $G$, written $S_1 \|_G S_2$, and is defined to be $\langle Q_1 \times Q_2,\ (q0_1, q0_2), \rightarrow \rangle$, where $\rightarrow$ is the classical synchronous composition relation defined below:*

$$\frac{q_1 \xrightarrow{A}_1 q_1'}{(q_1, q_2) \xrightarrow{A} (q_1', q_2)}\ A \cap G = \emptyset \qquad\qquad \frac{q_2 \xrightarrow{A}_2 q_2'}{(q_1, q_2) \xrightarrow{A} (q_1, q_2')}\ A \cap G = \emptyset$$

$$\frac{q_1 \xrightarrow{A}_1 q_1',\ q_2 \xrightarrow{B}_2 q_2'}{(q_1, q_2) \xrightarrow{A \cup B} (q_1', q_2')}\ A \cap G = B \cap G \neq \emptyset$$

We can now define contracts to be automata with each state tagged with the contract which will be in force at that point. The contracts will be able to refer to both presence and absence of an action. Given an alphabet of actions $\Sigma$, we write $\Sigma!$ to refer to the alphabet extended with actions preceded with an exclamation mark ! to denote their absence: $\Sigma! \stackrel{df}{=} \Sigma \cup \{!a \mid a \in \Sigma\}$. We use variables $x$ and $y$ to range over $\Sigma!$. If $x$ is already an inverted action $x = !a$, then expression $!x$ is interpreted to be $a$.

Contract clauses are either (i) obligation clauses of the form $\mathcal{O}_p(a)$ or $\mathcal{O}_p(!a)$, which say that party $p$ is obliged to perform or not perform action $a$ respectively; or (ii) permission clauses which can be either of the form of $\mathcal{P}_p(a)$ or $\mathcal{P}_p(!a)$ (party $p$ is permitted to perform, or not perform action $a$ respectively)., or delayed permission $\delta^X_{\leq n}(\pi)$ (which gives permission $\pi$ after at most $n$ steps consisting solely of actions in $X$.

**Definition 2.** *A* contract clause *over alphabet $\Sigma$ is structured as follows (where action $x \in \Sigma!$, party $p \in \{1, 2\}$, set of actions $X \subseteq 2^{\Sigma!}$, permission $\pi \in$ Permission):*

$$\begin{aligned} \textit{Clause} &::= \mathcal{O}_p(x) \mid \textit{Permission} \\ \textit{Permission} &::= \mathcal{P}_p(x) \mid \delta^X_{\leq n}(\pi) \end{aligned}$$

*We write $\overline{p}$ to refer to the party other than $p$.*
*A* contract automaton *is a total and deterministic multi-action automaton $S = \langle Q,\ q0,\ \rightarrow \rangle$, together with a total function contract $\in Q \rightarrow 2^{Clause}$ assigning a set of clauses to each state. We use $\mathcal{CA}$ to refer to the class of contract automata.*
*Two contract automata are said to be structurally isomorphic if they are structurally identical automata (they have the same set of states, initial state and transition relation) but may have different contract functions.*

We can now define a regulated two-party system in terms of multi-action automata.

**Definition 3.** *A* regulated two-party system *synchronising over the set of actions $G$ is a tuple $R = \langle S_1, S_2 \rangle^{\mathcal{A}}_G$, where $S_i = (\Sigma_i, Q_i, q0_i, \rightarrow_i)$ is a multi-action automaton specifying the behaviour of party $i$, and $\mathcal{A}$ is a contract automaton over alphabet $\Sigma_1 \cup \Sigma_2$.*

*The behaviour of a regulated two-party system $R$, written $[\![R]\!]$, is defined to be the automaton $(S_1 \|_G S_2) \|_\Sigma \mathcal{A}$. To make states in such systems more readable, we will write $((q_1, q_2), q_\mathcal{A})$ as $(q_1, q_2)_{q_\mathcal{A}}$.*

*A regulated two-party system is well-formed if $S_1 \|_G S_2$ never deadlocks: $\forall (q_1, q_2) \cdot acts(q_1, q_2) \neq \emptyset$.*

In the rest of the paper we will assume that all systems are well-formed, i.e., do not deadlock. One way of guaranteeing this may be by having all system states provide a transition with the empty action.
Also note that the totality of the contract automaton guarantees that the system behaviour is not constrained, but simply acts to tag the states with the relevant contracts at each point in time.

### 2.2 Contract Satisfaction

Given a two-party system $(S_1, S_2)$, and a contract automaton $\mathcal{A}$, we can now define whether or not either party is violating the contract when a particular state is reached or a transition is taken. As we will see, a dual-view of violation, identifying *both* bad states and bad transitions, is necessary in a deontic context. We will look at the different deontic operators and define the set of violations induced for each of them.

**Definition 4.** *Functions $O_p(q_\mathcal{A})$ and $F_p(q_\mathcal{A})$ give the set of actions respectively obliged to be performed and obliged not to be performed by party $p$. They are defined in terms of the contract clauses in the state.*

$$O_p(q_\mathcal{A}) \stackrel{df}{=} \{a \mid \mathcal{O}_p(a) \in contract(q_\mathcal{A})\}$$
$$F_p(q_\mathcal{A}) \stackrel{df}{=} \{a \mid \mathcal{O}_p(!a) \in contract(q_\mathcal{A})\}$$

*Action set $A$ is said to be viable for party $p$ in a contract automaton state $q_\mathcal{A}$, written $viable_p(q_\mathcal{A}, A)$, if (i) all her obliged actions are included in $A$ but; (ii) no actions which the party is obliged not to perform are included $A$:*

$$viable_p(q_\mathcal{A}, A) \stackrel{df}{=} O_p(q_\mathcal{A}) \subseteq A \wedge F_p(q_\mathcal{A}) \cap A = \emptyset$$

Since we would like to be able to place blame in the case of a violation, we parametrise contract satisfaction and violation by party.

It is also worth noting that while obligation to perform an action, for instance, is violated in a transition which does not include the action, permission is violated by a state in which the opportunity to perform the permitted action is not present. The satisfaction predicate will thus be overloaded to be applicable to both states and transitions. The predicate $sat_p^\mathcal{A}(X)$ will denote that the contract automaton $\mathcal{A}$, reaching state $X$ or traversing transition $X$, does not constitute a violation for party $p$. $X$ ranges over states and transitions in the composed system. When $\mathcal{A}$ is clear from the context, we simply write $sat_p(X)$. We start by defining separate satisfaction predicates for the deontic operators.

***Permission***. If party $p$ is permitted to perform shared action $a$, then the other party $\overline{p}$ must provide $p$ with at least one viable outgoing transition which contains $a$ but does not include any forbidden actions. Permission to perform local actions cannot be violated. In the case of a single permission, this can be expressed as follows:

$$(q_1, q_2)_{q_{\mathcal{A}}} \vdash_p \mathcal{P}_{\overline{p}}(a) \stackrel{df}{=} a \in G \implies \exists A \in \mathit{acts}(q_p),\ A' \subseteq G^c \cdot a \in A \wedge \mathit{viable}_{\overline{p}}(q_{\mathcal{A}}, A \cup A')$$

Similarly, if party $p$ is permitted to not perform action $a$, then the other party $\overline{p}$ must provide $p$ with at least one viable outgoing transition which does not include $a$ nor any forbidden action. Permission to perform local actions can never be violated. In the case of a single permission, this can be expressed as follows:

$$(q_1, q_2)_{q_{\mathcal{A}}} \vdash_p \mathcal{P}_{\overline{p}}(!a) \stackrel{df}{=} a \in G \implies \exists A \in \mathit{acts}(q_p),\ A' \subseteq G^c \cdot a \notin A \wedge \mathit{viable}_{\overline{p}}(q_{\mathcal{A}}, A \cup A')$$

While actual obligation violations occur when an action is not performed, violations of a permission occur when no appropriate action is possible. In this paper we give a semantics that tags as a violation a state in which one party is permitted to perform an action, while the other provides no way of actually doing so. For any other parameters, the permission is otherwise satisfied.

***Delayed Permission***. We use the notion of delayed permission, written $\delta^A_{\leq n}(\pi)$ to denote that party $p$ has permission $\pi$, although up to $n$ actions from action set $A$ may occur before this permission is granted. This can be defined recursively over $n$ in the following manner:

$$\begin{aligned}
&(q_1, q_2)_{q_{\mathcal{A}}} \vdash_p \delta^A_{\leq 0}(\pi) \stackrel{df}{=} (q_1, q_2)_{q_{\mathcal{A}}} \vdash_p \pi \\
&(q_1, q_2)_{q_{\mathcal{A}}} \vdash_p \delta^A_{\leq n+1}(\pi) \stackrel{df}{=} (q_1, q_2)_{q_{\mathcal{A}}} \vdash_p \pi \vee \\
&\qquad \forall (A', (q_1', q_2')_{q'_{\mathcal{A}}}) \in \mathit{next}((q_1, q_2)_{q_{\mathcal{A}}}) \ \cdot\ A' \subseteq A \wedge (q_1', q_2')_{q'_{\mathcal{A}}} \vdash_p \delta^{n+1}_{\leq A}(\pi)
\end{aligned}$$

To combine all permissions in a state, we simply take the conjunction of all conditions:

$$\mathit{sat}^P_p((q_1, q_2)_{q_{\mathcal{A}}}) \stackrel{df}{=} \forall \mathcal{P}_{\overline{p}}(x) \in q_{\mathcal{A}} \ \cdot\ (q_1, q_2)_{q_{\mathcal{A}}} \vdash_p \mathcal{P}_{\overline{p}}(x)$$

All transitions are taken as satisfying the permission satisfaction function.

***Obligation***. Obligation brings in constraints on both parties. Given that party $p$ is obliged to perform action $a$ in a state means that (i) party $p$ must include the action in any outgoing transition in the composed system in which it participates; and (ii) the other party $\overline{p}$ must provide a viable synchronisation action set which, together with other asynchronous actions performed by $p$, allows $p$ to perform *all* its obligations, positive and negative. Obligation to not perform action $a$ ($\mathcal{O}_p(!a)$) can be similarly expressed. We combine all positive and negative obligations in the following definition:

$$\begin{aligned}
&\mathit{sat}^O_p((q_1, q_2)_{q_{\mathcal{A}}} \xrightarrow{A} (q_1', q_2')_{q'_{\mathcal{A}}}) \stackrel{df}{=} \mathit{viable}_p(q_{\mathcal{A}}, A) \\
&\mathit{sat}^O_{\overline{p}}((q_1, q_2)_{q_{\mathcal{A}}}) \stackrel{df}{=} \exists A \in \mathit{acts}(q_{\overline{p}}),\ A' \subseteq G^c \ \cdot\ \mathit{viable}_p(q_{\mathcal{A}}, A \cup A')
\end{aligned}$$

The satisfaction constraint for transitions is only applicable if $A$ is not an action set performed asynchronously by $\overline{p}$. For other parameters, $\mathit{sat}^O_p(X)$ is true.

***General contract satisfaction***. It is defined as: $sat_p(X) \stackrel{df}{=} sat_p^P(X) \wedge sat_p^O(X)$. Based on this, we can now define correctness of a regulated two-party system.

**Definition 5.** *A party $p$ is said to be incapable of breaching a contract in a regulated two-party system $R = \langle S_1, S_2 \rangle_G^A$, written $breachIncapable_p(R)$, if $p$ cannot be in violation in any of the reachable states and transitions of $R$.*

Note that being breach-incapable is stronger than just being compliant for one specific run — $breachIncapable_p(R)$ means that there is no possible trace of $R$, in which $p$ breaches the contract.

### 2.3 Other Modalities

**Definition 6.** *Permissions and obligations are duals under a notion of norm opposites and action absence. We define the opposite of permission and obligation $!\mathcal{P}_p(x)$ and $!\mathcal{O}_p(x)$ syntactically in the following manner:*

- *Party $p$ not being permitted to perform an action is equivalent to $p$ being obliged not to perform the action:* $!\mathcal{P}_p(a) \stackrel{df}{=} \mathcal{O}_p(!a) \qquad !\mathcal{P}_p(!a) \stackrel{df}{=} \mathcal{O}_p(a)$
- *Party $p$ not being obliged to perform an action is equivalent to $p$ being permitted not to perform the action:* $!\mathcal{O}_p(a) \stackrel{df}{=} \mathcal{P}_p(!a) \qquad !\mathcal{O}_p(!a) \stackrel{df}{=} \mathcal{P}_p(a)$

It should be noted that we are equating lack of permission to do $a$ to an obligation to perform an action set which does not include $a$. Although this seems to go against the intuitive idea of letting a party do nothing as a way of not violating lack of permission, note that (i) since transitions carry sets of actions, the empty set of actions is a way of satisfying the obligation; and (ii) well-formedness (see Definition 3) of the parties ensures that progress is always possible thus making the formulation of lack of permission conform to our expectations.

It is interesting to note that in a two party system there are alternative notions of opposites to permission and obligation. Consider party $p$ not being permitted to perform action $a$. Apart from the interpretation we gave, in which the norm places the onus on party $p$ not to perform $a$, an alternative view is to push the responsibility to $\overline{p}$ and interpret it as: *party $\overline{p}$ may not provide a viable action set which includes $a$*. This is distinct from $!\mathcal{P}_p(a)$ (and indeed from the other modalities we have). Similarly, consider party $p$ not being obliged to perform action $a$. The interpretation we adopted permits party $p$ to not perform $a$, but once again, alternative definitions may be adopted. One possibility is to push the responsibility to $\overline{p}$ and interpret it as: *party $\overline{p}$ must provide a viable transition which does not include $a$*. These duals, in which the outer negation of a norm also corresponds to shifting of responsibility give an interesting alternative view of norm opposites in a two-party system. Another interesting alternative would be to interpret these negations as modalities whose only effect is the cancelling of existing clauses. We will not explore these alternative modalities any further in this paper, since the modalities we adopt (i) more closely correspond to the Kanger *et al.* approach; and (ii) provide a clean notion of conflicts, as discussed in Section 4. Should they be needed for a particular application, any of the above mentioned interpretations could be

included as alternative type of negation. One of the advantages of clear formal semantics is that there is no need to dispute the meaning of a given term, since different ones can be defined and the appropriate one be picked to convey specific meanings. Prohibition can now be defined as the dual of permission:

**Definition 7.** *Prohibition contract clauses $\mathcal{F}_p(a)$ and $\mathcal{F}_p(!a)$, prohibiting party $p$ from performing and not performing $a$ respectively, can be expressed in terms of permission:*

$$\mathcal{F}_p(a) \stackrel{df}{=} !\mathcal{P}_p(a) \qquad \mathcal{F}_p(!a) \stackrel{df}{=} !\mathcal{P}_p(!a)$$

These definitions allow us to express prohibition in terms of obligation not to perform an action:

**Proposition 1.** *Prohibition to perform an action is equivalent to obligation not to perform the action: $\mathcal{F}_p(x) = \mathcal{O}_p(!x)$.*

### 2.4 Contract Strength

We can now define strictness relationships over contracts.

**Definition 8.** *A contract automaton $\mathcal{A}'$ is said to be* stricter than *contract automaton $\mathcal{A}$ for party $p$ (or $\mathcal{A}$ said to be* more lenient *than $\mathcal{A}'$ for party $p$), written $\mathcal{A} \sqsubseteq_p \mathcal{A}'$, if for any systems $S_1$ and $S_2$, $breachIncapable_p(\langle S_1, S_2\rangle_G^{\mathcal{A}'}) \implies breachIncapable_p(\langle S_1, S_2\rangle_G^{\mathcal{A}})$. We say that two contract automata $\mathcal{A}$ and $\mathcal{A}'$ are equivalent for party $p$, written $\mathcal{A} =_p \mathcal{A}'$, if $\mathcal{A} \sqsubseteq_p \mathcal{A}'$ and $\mathcal{A}' \sqsubseteq_p \mathcal{A}$. We define global contract strictness $\mathcal{A} \sqsubseteq \mathcal{A}'$ to hold if $\mathcal{A} \sqsubseteq_p \mathcal{A}'$ holds for all parties $p$, and similarly global contract equivalence $\mathcal{A} = \mathcal{A}'$.*

**Proposition 2.** *The relation over contracts $\sqsubseteq$ is a partial order.*

Structurally isomorphic contract automata provide a useful proof technique:

**Proposition 3.** *Given two structurally isomorphic contract automata $\mathcal{A}$ and $\mathcal{A}'$, $\mathcal{A} \sqsubseteq \mathcal{A}'$ if and only if, for any state or transition $X$, $sat_p^{\mathcal{A}'}(X) \implies sat_p^{\mathcal{A}}(X)$.*

This proof principle can be proved to hold by showing that (i) the automata obtained from the synchronous composition with the two contracts are structurally identical; and (ii) using the definition of breach incapability. The principle can be used to prove that contract automata are monotonic:

**Proposition 4.** *Contract automata are monotonic: given two structurally isomorphic contract automata $\mathcal{A}$ and $\mathcal{A}'$, with contract clause functions contract and contract$'$ respectively, which satisfy that $\forall q \cdot contract(q) \subseteq contract'(q)$, it follows that $\mathcal{A} \sqsubseteq \mathcal{A}'$.*

The proof follows from the observation that $sat_p(X)$ is essentially a conjunction of a proposition for each contract clause in the state. Hence, $sat_p^{\mathcal{A}'}(X)$ (which has a larger set of clauses) implies $sat_p^{\mathcal{A}}(X)$. Applying Proposition 3 to this observation completes the proof.

Although contracts are expressed as automata, we would like to be able to compare individual clauses. To do this we will need to relate contract automata which are equivalent except for a particular clause replaced by another.

**Definition 9.** *Given two contract clauses $C$ and $C'$, the relation over contract automata $[C \to C'] \subseteq \mathcal{CA} \times \mathcal{CA}$ relates two contract automata $\mathcal{A}$ and $\mathcal{A}'$ if $\mathcal{A}$ is equivalent to $\mathcal{A}'$ except possibly for a number of instances of clause $C$ replaced by $C'$.*

*We extend the notion of strictness to contract clauses. We say that clause $C'$ is stricter than clause $C$ for party $p$, written $C \sqsubseteq_p C'$, if for any contract automata $\mathcal{A}$ and $\mathcal{A}'$ such that $(\mathcal{A}, \mathcal{A}') \in [C \to C']$, it follows that $\mathcal{A} \sqsubseteq_p \mathcal{A}'$. We similarly extend the notion of strictness for all parties $\sqsubseteq$.*

The following proposition allows us to use the proof principle given in Proposition 3 for reasoning about clause strictness:

**Proposition 5.** *Given clauses $C$ and $C'$, any two contract automata related by $[C \to C']$ are structurally isomorphic.*

**Theorem 1.** *Obligation is stricter than permission: (i) $\mathcal{P}_p(a) \sqsubseteq \mathcal{O}_p(a)$; and (ii) $\mathcal{P}_p(!a) \sqsubseteq \mathcal{O}_p(!a)$.*

*Proof. We present the proof of (i) — the proof of (ii) is very similar. We need to prove that for any contract automata $\mathcal{A}$ and $\mathcal{A}'$ such that $(\mathcal{A}, \mathcal{A}') \in [\mathcal{P}_p(a) \to \mathcal{O}_p(a)]$, then it follows that $\mathcal{A} \sqsubseteq \mathcal{A}'$. Using Proposition 5, we know that $\mathcal{A}$ and $\mathcal{A}'$ are structurally isomorphic, allowing us to apply the proof principle of Proposition 3.*

*We thus have to show that $sat_p^{\mathcal{A}'}(X)$ implies $sat_p^{\mathcal{A}}(X)$. Since the permission in $\mathcal{A}$ which is replaced by an obligation, never yields violations for party $p$ nor for any party on transitions, it suffices to prove that this implication holds on states for party $\overline{p}$.*
*The satisfaction function for $\overline{p}$'s obligations in states is:*

$$\exists A \in acts(q_{\overline{p}}),\ A' \subseteq G^c \ \cdot\ viable_p(q_{\mathcal{A}'}, A \cup A')$$

*If $a \in G$, and since $a \in O_p(q_{\mathcal{A}'})$, we can conclude that $a \in A$:*

$$a \in G \implies \exists A \in acts(q_p),\ A' \subseteq G^c \ \cdot\ a \in A \wedge viable_{\overline{p}}(q_{\mathcal{A}'}, A \cup A')$$

*Furthermore, since $q_{\mathcal{A}}$ has less obligations than $q_{\mathcal{A}'}$, viability for $q_{\mathcal{A}'}$ implies viability for $q_{\mathcal{A}}$:*

$$a \in G \implies \exists A \in acts(q_p),\ A' \subseteq G^c \ \cdot\ a \in A \wedge viable_{\overline{p}}(q_{\mathcal{A}}, A \cup A')$$

*Hence, the satisfaction function for the permission $\mathcal{P}_p(a)$ holds and thus, by Proposition 3 we can conclude that $\mathcal{A} \sqsubseteq \mathcal{A}'$.*

**Theorem 2.** *For synchronised actions, obligation for one party is stricter than permission for the other: (i) $\mathcal{P}_p(a) \sqsubseteq \mathcal{O}_{\overline{p}}(a)$; and (ii) $\mathcal{P}_p(!a) \sqsubseteq \mathcal{O}_{\overline{p}}(!a)$.*

*Proof. As in the previous theorem, we observe that $\mathcal{P}_p(a)$ can only yield violations for states and for party $\overline{p}$.*

*Observe that the obligation $\mathcal{O}_{\overline{p}}(a)$ in a state $q_{\mathcal{A}'}$ guarantees that all outgoing transitions from the state $(q_1, q_2)_{q_{\mathcal{A}'}} \xrightarrow{A} (q_1', q_2')_{q'_{\mathcal{A}'}}$ satisfy $viable_{\overline{p}}(q_{\mathcal{A}'}, A)$.*

*Since we assume that the system does not deadlock, there is at least one such transition which party $p$ participates in. Furthermore, if $a \in G$, it must also appear in the actions on the transition:*

$$a \in G \implies \exists A \in acts(q_p),\ A' \subseteq G^c \ \cdot\ a \in A \wedge viable_{\overline{p}}(q_{\mathcal{A}'}, A \cup A')$$

*This guarantees that $(q_1, q_2)_{q_{\mathcal{A}}} \vdash_p \mathcal{P}_p(a)$, and allows us to complete the proof using Proposition 3.*

It is interesting to note that in a synchronous world without blame-identification, one could show equivalence between $\mathcal{O}_p(a)$ and $\mathcal{O}_{\overline{p}}(a)$ since a lack of $a$ on a transition would cause a violation of both obligations. However, since our partial order $\sqsubseteq_p$ is parametrised by the party, one can show that the two obligations are in fact different [3].

### 2.5 Mutually Exclusive Actions

Although we adopt a multi-action approach, modelling real-world scenarios means that certain actions should never occur concurrently. For instance, one would expect that the automata never perform the action *openDoor* and *closeDoor* on the same transition. This allows us to identify strictness laws which hold only for mutually exclusive actions.

**Definition 10.** *Given a multi-action automaton $\langle \Sigma,\ Q,\ q0,\ \rightarrow \rangle$, two actions $a$ and $b$ ($\{a, b\} \subseteq \Sigma$) are said to be mutually exclusive, written $a \bowtie b$, if they can never appear in the same set of actions on transitions. Thus, for any automaton, it should be the case that:*

$$\forall (q, A, q') \in \rightarrow\ \cdot\ a \in A \implies b \notin A$$

In the rest of the article we will assume that mutually exclusive actions never appear in the synchronisation sets. This is done to simplify the presentation, since otherwise we would need a more complex rule for synchronous composition (not allowing synchronisation when the asynchronous actions of party are in conflict with those of the other) and a modified definition for the satisfaction of obligations (the other party must provide a viable action set which does not include any actions which may conflict with the obligations of the party to whom the obligation applies). Removing this restriction, however, does not affect the results we present. The following theorem shows how mutually exclusive actions and action absence are related together under both obligation and permission:

**Theorem 3.** *If $a \bowtie b$ then (i) $\mathcal{O}_p(!a) \sqsubseteq \mathcal{O}_p(b)$; and (ii) $\mathcal{P}_p(!a) \sqsubseteq \mathcal{P}_p(b)$.*

*Proof. To show (i), we need to prove that for any contract automata $\mathcal{A}$ and $\mathcal{A}'$ such that $(\mathcal{A}, \mathcal{A}') \in [\mathcal{O}_p(!a) \rightarrow \mathcal{O}_p(b)]$, then it follows that $\mathcal{A} \sqsubseteq \mathcal{A}'$. As in the previous proofs, we can use Proposition 5 to conclude that $\mathcal{A}$ and $\mathcal{A}'$ are structurally isomorphic, allowing us to apply the proof principle of Proposition 3.*

*We thus have to show that $sat_p^{\mathcal{A}'}(X)$ implies $sat_p^{\mathcal{A}}(X)$. We look at transitions and states separately:*

**Transitions:** *The satisfaction function for the combined obligations for a transition $(q_1, q_2)_{q_{\mathcal{A}'}} \xrightarrow{A} (q_1', q_2')_{q'_{\mathcal{A}'}}$ in automaton $\mathcal{A}'$ is that $viable_p(q_{\mathcal{A}'}, A)$. By definition of viability and the obligation $\mathcal{O}_p(b)$ in $q_{\mathcal{A}'}$, we can thus conclude that $b \in A$. However, since $a \bowtie b$, this means that $a \notin A$, from which we can conclude that $viable_p(q_{\mathcal{A}}, A)$ and hence that the satisfaction function also holds for transitions in $\mathcal{A}$.*

**States:** *The satisfaction function applied to states acts on the other party $\overline{p}$. For state $(q_1, q_2)_{q_{\mathcal{A}'}}$, it is defined to be $\exists A \in acts(q_{\overline{p}}),\ A' \subseteq G^c\ \cdot\ viable_p(q_{\mathcal{A}'}, A \cup A')$. Since $a \in G$, the proof is identical to the previous case.*

*Hence, the satisfaction function for $\mathcal{O}_p(a)$ holds and thus, by Proposition 3 we can conclude that $\mathcal{A} \sqsubseteq \mathcal{A}'$ and hence (i) holds.*

*The proof of (ii) follows similarly.*

A similar result can be shown, but referring to the other party in the contract:

**Theorem 4.** *If $a \bowtie b$ then $\mathcal{O}_{\overline{p}}(!b) \sqsubseteq \mathcal{O}_p(a)$.*

*Proof. We take an approach identical to the previous theorems and prove that for any contract automata $\mathcal{A}$ and $\mathcal{A}'$ such that $(\mathcal{A}, \mathcal{A}') \in [\mathcal{O}_{\overline{p}}(!b) \rightarrow \mathcal{O}_p(a)]$, then it follows that $\mathcal{A} \sqsubseteq \mathcal{A}'$. Propositions 5 and 3 can then be used to complete the proof. As before, we consider the satisfaction relation on states and transitions separately:*

**Transitions:** *The satisfaction function for the combined obligations for a transition $(q_1, q_2)_{q_{\mathcal{A}'}} \xrightarrow{A} (q_1', q_2')_{q'_{\mathcal{A}'}}$ in automaton $\mathcal{A}'$ is that $viable_p(q_{\mathcal{A}'}, A)$. By definition of viability and the obligation $\mathcal{O}_p(a)$ in $q_{\mathcal{A}'}$, we can thus conclude that $a \in A$. However, since $a \bowtie b$, this means that $b \notin A$. The same transition must be viable for $\overline{p}$ in $\mathcal{A}'$, so $viable_{\overline{p}}(q_{\mathcal{A}'}, A)$ holds. The absence of $b$ also allows us to conclude that $viable_{\overline{p}}(q_{\mathcal{A}}, A)$, which is the satisfaction function for $\mathcal{O}_{\overline{p}}(!b)$ over transitions in $\mathcal{A}$.*

**States:** *For state $(q_1, q_2)_{q_{\mathcal{A}'}}$, since we assume deadlock freedom and satisfaction of the obligation to perform $a$, we know of the existence of an outgoing transition with action $a$ such that $a \in A$. Since party $p$ is participating in this transition, and $a \in G$, we can conclude that there is a transition viable for $\overline{p}$, leaving from $q_p$ and with an action set which includes $a$ and hence not $b$. Propositions 5 and 3 can then be conclude that $\exists A \in acts(q_p),\ A' \subseteq G^c \ \cdot\ viable_{\overline{p}}(q_{\mathcal{A}}, A \cup A')$.*

Although one may be tempted to induce that a similar result can be shown for permission (analogous to part (ii) of Theorem 3) — $\mathcal{P}_{\overline{p}}(!b) \sqsubseteq \mathcal{P}_p(a)$ does not always hold. As a simple example of a system satisfying $\mathcal{P}_p(a)$ but not $\mathcal{P}_{\overline{p}}(!b)$, consider party $p$ be able to perform just one transition with action set $\{b\}$, and party $\overline{p}$ being able to perform one of two transitions: one with action set $\{a\}$, the other with action set $\{b\}$. Party $p$ is permitted to perform $a$ but party $\overline{p}$ is not permitted to perform $!b$.

## 3 Kanger Rights in a Two-Party Setting

Kanger's paper investigated the notion of rights in a general setting, and although the rights are directed between parties (e.g. party $p$ has versus party $\overline{p}$ a claim that $S(p, \overline{p})$), the interaction between the parties and directionality of the rights depends on various other notions such as causality, interference and intention. The synchronous two-party approach we presented in the previous section gives a closed-world view for rights, that allows these notions to be formalised in a straightforward manner. In this section, we explore how Kanger's rights translate into this setting.

### 3.1 Actions and States

Kanger *et al.* [2] presents rights to be over a state of affairs, which is clearly a state-based look, but also identifies whether or not a party is responsible for causing a state to hold — indicating that there is an underlying notion of a party performing an action which leads to the state predicate holding. In synchronous systems, the parties involved synchronise over actions, making the approach inherently action-based. There are various standard ways in which one can encode state using actions and vice-versa.

Consider a simple encoding in which, given a state $S$, we identify action $S^{\top}$, whose presence causes $S$ to hold, and whose absence causes $S$ not to hold. This turns out to be overly restrictive, and does not enable the encoding of all models. For instance, in Kanger *et al.* [2], one has to distinguish between:

*not (party $p$ causes state $S$ to hold)* (1)

*party $p$ causes $S$ not to hold* (2)

Both statements correspond to the action-based statement:

*party $p$ does not perform action $S^{\top}$*

This makes this formalism insufficiently discriminating for our needs. On the other hand, identifying two special (and mutually exclusive) actions $S^{\uparrow}$ and $S^{\downarrow}$ which cause $S$ to start holding (become true) and $S$ to stop holding (become false) respectively, the distinction becomes possible, since statements (1) and (2) can be expressed as:

*party $p$ does not perform action $S^{\uparrow}$* (1′)

*party $p$ performs action $S^{\downarrow}$* (2′)

Two important properties of these actions are that: (i) the actions are mutually exclusive — the system may never perform $S^{\uparrow}$ and $S^{\downarrow}$ together; and (ii) the causality actions for the negation of a state $\neg S$ are the opposite of those of $S$ i.e. $(\neg S)^{\uparrow} = S^{\downarrow}$ and $(\neg S)^{\downarrow} = S^{\uparrow}$.

### 3.2 Kanger *et al.*

Kanger *et al.* [2] identify eight simple types of rights:

(a) Party $p$ has versus party $\overline{p}$ a *claim* that $S(p, \overline{p})$.
(b) Party $p$ has versus party $\overline{p}$ a *freedom* that $S(p, \overline{p})$.
(c) Party $p$ has versus party $\overline{p}$ a *power* that $S(p, \overline{p})$.
(d) Party $p$ has versus party $\overline{p}$ a *immunity* that $S(p, \overline{p})$.
(a′) Party $p$ has versus party $\overline{p}$ a *counter-claim* that $S(p, \overline{p})$.
(b′) Party $p$ has versus party $\overline{p}$ a *counter-freedom* that $S(p, \overline{p})$.
(c′) Party $p$ has versus party $\overline{p}$ a *counter-power* that $S(p, \overline{p})$.
(d′) Party $p$ has versus party $\overline{p}$ a *counter-immunity* that $S(p, \overline{p})$.

The first four can be considered as the fundamental rights, with the other four (the *counter* rights) being identical except that they refer to the negation of state predicate $S$.[3] Thus, for example, saying that *'party $p$ has versus party $\overline{p}$ a counter-claim that state $S(p, \overline{p})$ holds'* is identical to saying that *'party $p$ has versus party $\overline{p}$ a claim that not-$S(p, \overline{p})$'*.

[3] It is worth noting that since, in our context, we have only two parties interacting ($p$ and $\overline{p}$), we need not make explicit (i) the party versus whom the right is; and (ii) the parameters of state predicate $S$. We can thus write statements just as *'party $p$ has a claim that $S$'*.

### 3.3 Semantics

A discussion of the intuitive meaning of these different types of rights can be found in the original paper [2] or any of many papers discussing and extending these notions (see Section 6). However, Kanger *et al.* identifies the interpretation of rights (a) to (d) as:

(1a) It shall be that $\overline{p}$ causes that $S(p, \overline{p})$.
(1b) Not: it shall be that $p$ causes that not-$S(p, \overline{p})$.
(1c) Not: it shall be that not: $p$ causes that $S(p, \overline{p})$.
(1d) It shall be that not: $\overline{p}$ causes that not-$S(p, \overline{p})$.

Furthermore, Kanger *et al.* note that the statement *'Not: it shall be that not: ...'* is synonymous to *'It may be that ...'*. This allows us to rewrite the formulae without top-level negations.

(2a) It shall be that $\overline{p}$ causes that $S(p, \overline{p})$.
(2b) It may be that not: $p$ causes that not-$S(p, \overline{p})$.
(2c) It may be that $p$ causes that $S(p, \overline{p})$.
(2d) It shall be that not: $\overline{p}$ causes that not-$S(p, \overline{p})$.

Using the relationship between states and actions as identified in Section 3.1, these are equivalent to:

(3a) It shall be that $\overline{p}$ performs $S^{\uparrow}$.
(3b) It may be that not: $p$ performs $S^{\downarrow}$.
(3c) It may be that $p$ performs $S^{\uparrow}$.
(3d) It shall be that not: $\overline{p}$ performs $S^{\downarrow}$.

The *shall be* and *may be* modalities correspond to our notions of obligation and permission, enabling us to define the different forms of rights in our formal model:

$$\text{Cl}(p, \overline{p}, S) \stackrel{df}{=} \mathcal{O}_{\overline{p}}(S^{\uparrow}) \qquad \text{Po}(p, \overline{p}, S) \stackrel{df}{=} \mathcal{P}_{p}(S^{\uparrow})$$

$$\text{Fr}(p, \overline{p}, S) \stackrel{df}{=} \mathcal{P}_{p}(!S^{\downarrow}) \qquad \text{Im}(p, \overline{p}, S) \stackrel{df}{=} \mathcal{O}_{\overline{p}}(!S^{\downarrow})$$

$$\text{Counter-Cl}(p, \overline{p}, S) \stackrel{df}{=} \mathcal{O}_{\overline{p}}(S^{\downarrow}) \qquad \text{Counter-Po}(p, \overline{p}, S) \stackrel{df}{=} \mathcal{P}_{p}(S^{\downarrow})$$

$$\text{Counter-Fr}(p, \overline{p}, S) \stackrel{df}{=} \mathcal{P}_{p}(!S^{\uparrow}) \qquad \text{Counter-Im}(p, \overline{p}, S) \stackrel{df}{=} \mathcal{O}_{\overline{p}}(!S^{\uparrow})$$

One interesting observation emerging from this formalisation is that in a two party system some of the rights place constraints on both parties. For instance, if $p$ has versus $\overline{p}$ a claim that $S$, $\overline{p}$ has an obligation to perform $S^{\uparrow}$. If $S^{\uparrow}$ is an action local to $\overline{p}$, then no constraint is placed on $p$, but if it is a common action, the semantics of obligation insist that $p$ allows $\overline{p}$ to perform $S^{\uparrow}$. For example, consider $S$ to be *'$p$ has access to the web-service'*. Now, to make the predicate hold, $S^{\uparrow}$ may be the action *openPort* which opens a particular port. If this action is local to $\overline{p}$ (i.e., $\overline{p}$ can perform the action independently

of $p$), then the constraint lies solely on $\overline{p}$. However, if *openPort* is a shared action, then the semantics of obligation place a restriction on party $p$ to provide a feasible action set through which $\overline{p}$ may use to satisfy its obligation to open the port. In other words, although $p$ does not necessarily have to use the webservice, it must support $\overline{p}$ in opening the port.

In fact, the moment we are giving semantics to Kanger's types using the interactive two-party systems, the two models diverge. For example, in Kanger *et al.* the following two types are compatible (not in conflict):

$$\mathrm{Cl}(p, \overline{p}, S) \text{ and } !\mathrm{Po}(p, \overline{p}, S)$$

Their informal meaning, when transformed to reason about actions becomes:

*it shall be that* $\overline{p}$ *performs* $S^{\uparrow}$

*it shall be that* $p$ *does not perform* $S^{\uparrow}$

In a non-interactive system, Kanger's view is applicable and the clauses are compatible. However, in an interactive system, where $S^{\uparrow}$ is part of the synchronisation alphabet, these clauses become $\mathcal{O}_{\overline{p}}(S^{\uparrow})$ and $\mathcal{O}_p(!S^{\uparrow})$, which can be proved to be conflicting.

A claim implicit in Kanger *et al.* is that these basic right types are, in a sense, complete — in that they form a basis through the combination of which one can express all forms of rights. However, the formalisation we have given clearly shows that each basic right can be expressed by choosing: (i) the modality — is it a permission or an obligation?; (ii) the party to which the modality applies — is it $p$ or $\overline{p}$?; (iii) the change in the value of $S$ — is it $S^{\uparrow}$ or $S^{\downarrow}$?; and (iv) whether it is the presence or absence of that action that is of interest — e.g. is it $S^{\uparrow}$ or $!S^{\uparrow}$? These four different variables indicate that one can identify 16, not 8 basic right types. The missing ones have to correspond to obligations on party $p$, and permissions for party $\overline{p}$, neither of which fall under the category of rights of $p$. This justifies the argument for completeness of Kanger *et al.*'s basic types.

### 3.4 Strengths of Rights

Given the 8 basic rights, one can construct $2^8$ combinations over a particular state of affairs. However, not all these combinations are possible, since (i) some rights are subsumed by others; and (ii) some combinations of rights lead to conflicts (see Section 4). To address the first issue, Kanger provided a partial order on the basic rights in terms of their strength, as can be seen in Figure 1. An arrow from a right $R$ to a right $R'$ indicates that $R$ is stronger than $R'$, not unlike our notion of $R$ being stricter than $R'$. We can apply the formalised versions in a two-party setting to investigate which parts of this strength relation are preserved.

Figure 2 corresponds to diagram 1, but interpreted for two-party systems. The rights are replaced by their definitions, and the strength arrows revised as required. In fact, for a two party system most of the strictness inequalities still hold:

1. Ones marked ① are of the form $\mathcal{P}_p(a) \sqsubseteq \mathcal{O}_p(a)$, while those marked ② are of the form $\mathcal{P}_p(!a) \sqsubseteq \mathcal{O}_p(!a)$. In both cases, they follow from Theorem 1.
2. Those marked ③ are of the form $\mathcal{O}_p(!a) \sqsubseteq \mathcal{O}_p(b)$ while those marked ④ are of the form $\mathcal{P}_p(!a) \sqsubseteq \mathcal{P}_p(b)$, in both cases with mutually exclusive actions $a$ and $b$. These hold by Theorem 3.

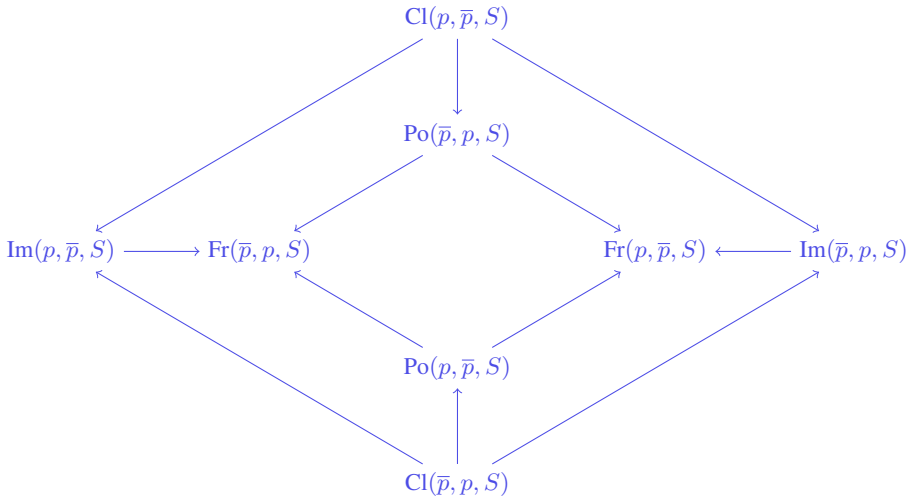


**Fig. 1.** Kanger *et al.* strength diagram

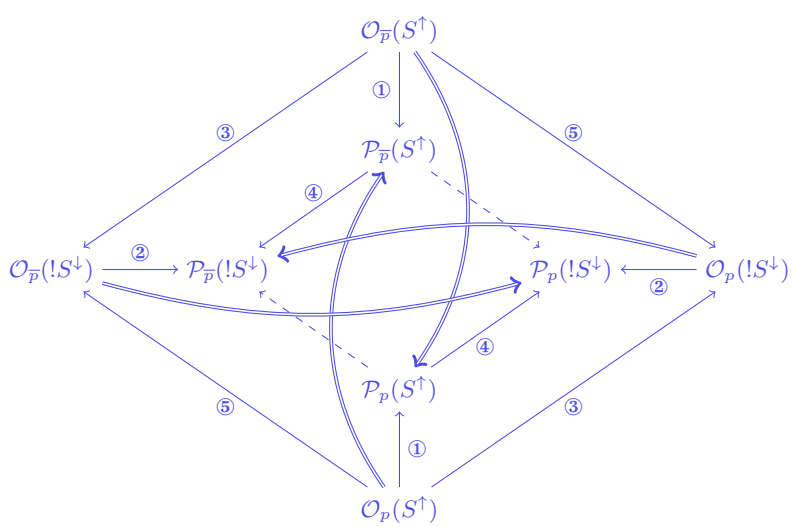


**Fig. 2.** Strength diagram for interacting parties.

3. Ones marked ⑤ are of the form $\mathcal{O}_p(!a) \sqsubseteq \mathcal{O}_{\overline{p}}(b)$ and thus follow from Theorem 4.

However, there are differences with the original diagram:

1. The dashed-line arrows connecting power (Po$(p, \overline{p}, S)$) with freedom (Fr$(\overline{p}, p, S)$) would require the strictness inequality: $\mathcal{P}_p(!a) \sqsubseteq \mathcal{P}_{\overline{p}}(b)$, with $a$ and $b$ being mutually exclusive actions. In Section 2.4 we gave a counter example to show that this does not always hold.
2. The double-line arrows connecting immunity (Im$(p, \overline{p}, S)$) with freedom (Fr$(p, \overline{p}, S)$) and claim (Cl$(p, \overline{p}, S)$) with power (Po$(p, \overline{p}, S)$), however now follow by Theorem 2.

## 4 Conflicts

Contract clauses are not always compatible with one another. Many definitions of conflict are possible — in this article we deal only with one particular class of conflicts which focusses on conflicting norms and mutually exclusive actions. In this section we axiomatise the notion of conflicts in interacting two-party systems and investigate some consequences.

**Definition 11.** *Contract conflicts is a relation between contract clauses* $\maltese \in$ *Clause* $\leftrightarrow$ *Clause and is defined to be the least relation satisfying the following axioms:*

**Axiom 1:** *Opposite permissions conflict:* $\vdash \mathcal{P}_p(x) \maltese !\mathcal{P}_p(x)$.
**Axiom 2:** *Obligation to perform mutually exclusive actions is a conflict:* $a \bowtie b \vdash \mathcal{O}_p(a) \maltese \mathcal{O}_p(b)$.
**Axiom 3:** *Conflicts are closed under symmetry:* $C \maltese C' \vdash C' \maltese C$.
**Axiom 4:** *Conflicts are closed under increased strictness:* $C \maltese C' \wedge C' \sqsubseteq C'' \vdash C \maltese C''$.

Although conflicts are only identified for opposing permissions in the axioms, they also arise in opposing obligations, and can be shown to follow from the axioms.

**Proposition 6.** *Opposite obligations conflict with each other:* $\mathcal{O}_p(x) \maltese !\mathcal{O}_p(x)$.

*Proof. The proof uses the definition of negated permission and obligation to derive the desired result:*

$$
\begin{array}{l}
\quad \textit{definition of conflict on opposing permissions} \\
\Longrightarrow \mathcal{P}_p(x) \maltese !\mathcal{P}_p(x) \\
\Longrightarrow \textit{for some } y,\ x = !y \\
\quad \mathcal{P}_p(!y) \maltese !\mathcal{P}_p(!y) \\
\Longrightarrow \textit{definition of } !\mathcal{P}_p(y) \textit{ and } !\mathcal{O}_p(y) \\
\quad !\mathcal{O}_p(y) \maltese \mathcal{O}_p(y) \\
\Longrightarrow \textit{symmetry of } \maltese \\
\quad \mathcal{O}_p(y) \maltese !\mathcal{O}_p(y)
\end{array}
$$

Various other conflicts can be derived from the axioms. The following show conflicts between permissions and obligations and arising from enforcing norms over both the presence and absence of an action.

**Proposition 7.** *Obligation to perform an action conflicts with both permission and obligation to not perform it: (i)* $\mathcal{O}_p(x) \maltese \mathcal{P}_p(!x)$*; and (ii)* $\mathcal{O}_p(x) \maltese \mathcal{O}_p(!x)$*. Obligation to perform an action also conflicts with lack of permission to perform the action: (iii)* $\mathcal{O}_p(x) \maltese !\mathcal{P}_p(x)$*.*

*Proof. By Proposition 6, we know that* $\mathcal{O}_p(x) \maltese !\mathcal{O}_p(x)$*, which, by definition of* $!\mathcal{O}_p(x)$ *is equivalent to* $\mathcal{O}_p(x) \maltese \mathcal{P}_p(!x)$*, hence completing the proof for (i).*
*By result (i) and* $\mathcal{P}_p(!x) \sqsubseteq \mathcal{O}_p(!x)$*, we can use the strictness axiom of conflicts to conclude that (ii) holds:* $\mathcal{O}_p(x) \maltese \mathcal{O}_p(!x)$*.*
*Result (iii) follows directly from the definition of* $!\mathcal{P}_p(x)$ *and result (ii).*

Finally, we show how making two conflicting contracts stricter does not get rid of the conflict:

**Proposition 8.** *Given two conflicting clauses* $C_1 \maltese C_2$*, making the two clauses stricter does not resolve the conflict: if* $C_1 \sqsubseteq C_1'$ *and* $C_2 \sqsubseteq C_2'$*, then* $C_1' \maltese C_2'$*.*

*Proof. The proof follows by applying axiom of closure under increased strictness twice and the axiom of symmetry.*

## 5 Atomic Types of Rights

Kanger *et al.* proceed to identify the so called *atomic types of rights* — given one has 8 possible basic types of rights, one can describe the rights regarding a particular state, by identifying which of the basic rights hold, and which do not (their negation holds). This yields 256 possible atomic rights, but since some of the combinations are conflicting, Kanger *et al.* use their strength diagram to show that no more than 26 distinct combinations can be identified. Furthermore, since some of the rights or their negations imply each other, the sets of atomic rights can be simplified by removing the weaker clauses. A set of non-conflicting basic types which cannot be simplified any further is said to be complete.

**Definition 12.** *A basic right for party* $p$ *towards party* $\overline{p}$ *and about state* $S$ *is either a Kanger right (e.g. Po*$(p, \overline{p}, S)$*) or its negation (e.g.* $!$*Po*$(p, \overline{p}, S)$*).*

*A consistent set of such rights* $R$ *is said to be* complete *if any further basic right* $r$ *is either unnecessary (*$\exists r' \in R \cdot r \sqsubseteq r'$*) or leads to an inconsistency (*$\exists r' \in R \cdot r \maltese r'$*).*

As we have shown, in an interacting two-party setting, the strength diagram induced is somewhat different, which in turn leads to different atomic rights. In fact, it can be shown that in a two-party setting, one can now identify just 22 atomic types as listed below.

| | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Claim | ✓ | ✓ | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × |
| Freedom | | | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | | | | | | | | | | | | | | |
| Power | | | × | × | × | × | × | × | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | × | × | | |
| Immunity | | | ✓ | ✓ | × | × | × | × | × | ✓ | ✓ | ✓ | ✓ | × | × | × | × | ✓ | ✓ | × | × | | |
| Counter-claim | × | × | × | × | ✓ | × | × | ✓ | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × |
| Counter-freedom | ✓ | × | | | | | | | | ✓ | ✓ | | | ✓ | ✓ | | | | | | | | × |
| Counter-power | × | × | ✓ | ✓ | | ✓ | ✓ | | | × | × | ✓ | ✓ | × | × | ✓ | ✓ | × | × | × | × | × | × |
| Counter-immunity | × | × | ✓ | × | | ✓ | × | | | ✓ | × | ✓ | × | ✓ | × | ✓ | × | ✓ | × | ✓ | × | × | × |

1. Not claim, power, immunity, not counter-claim, counter-power, counter-immunity.
2. Not claim, power, immunity, not counter-claim, counter-freedom, not counter-power, counter-immunity.
3. Not claim, not power, immunity, not counter-claim, not counter-power, counter-immunity.
4. Not claim, power, immunity, not counter-claim, counter-power, not counter-immunity.
5. Not claim, freedom, not power, immunity, not counter-claim, counter-power, not counter-immunity.
6. Not claim, power, not immunity, not counter-claim, counter-power, not counter-immunity.
7. Claim, not counter-claim, counter-freedom, not counter-power, not counter-immunity.
8. Claim, not counter-claim, not counter-freedom, not counter-power, not counter-immunity.
9. Not claim, power, immunity, not counter-claim, counter-freedom, not counter-power, not counter-immunity.
10. Not claim, power, not immunity, not counter-claim, counter-freedom, not counter-power, not counter-immunity.
11. Not claim, not power, immunity, not counter-claim, not counter-power, not counter-immunity.
12. Not claim, not power, not immunity, not counter-claim, not counter-power, not counter-immunity.
13. Not claim, not counter-claim, not counter-freedom, not counter-power, not counter-immunity.
14. Not claim, freedom, not power, immunity, not counter-claim, counter-power, counter-immunity.
15. Not claim, freedom, not power, not immunity, counter-claim.
16. Not claim, freedom, not power, not immunity, not counter-claim, counter-power, counter-immunity.
17. Not claim, freedom, not power, not immunity, not counter-claim, counter-power, not counter-immunity.
18. Not claim, not freedom, not power, not immunity, counter-claim.
19. Not claim, not freedom, not power, not immunity, not counter-claim.
20. Not claim, power, not immunity, not counter-claim, counter-freedom, not counter-power, counter-immunity.
21. Not claim, power, not immunity, not counter-claim, counter-power, counter-immunity.
22. Not claim, not power, not immunity, not counter-claim, not counter-power, counter-immunity.

Unsurprisingly, this gives a very different view of atomic rights, with for instance most including no claim, but only two including no freedom.

Kanger reduced his 26 types to 10, so the rest can be obtained by *inversions* ($S^{\uparrow}$ becomes $S^{\downarrow}$ and viceversa) and *conversions* (swapping parties). In our model conversions do not reduce the number of clauses, because modalities applied to one party already constraint the other. Inversions, however, do reduce our 22 to the first 13 listed above.

## 6 Related Work

Despite the fact that contracts are, by definition, an agreement between two or more parties, most formal studies regulate the parties independently and do not analyse how permissions, obligations or prohibitions for one party affect the other, or do so in limited ways (e.g., [6–10]). A summary of how our analysis of contracts compares to those can be found in [3] — here we focus on work done on Hohfelian types.

Makinson [11] analysed Kanger's types and proposed a compact representation $((\pm)O(\pm)\binom{x}{y}$ do $(\pm)S)$, similar to ours. This gives 16 possibilities, just like our analysis. Makinson does not provide formal semantics, deals with a state-of-affairs type of logic, does not analyse Kanger's atomic types and does not work with contracts, although he does analyse that there might be two parties, one bearing the right and the other being the counterparty. He also analyses the notion of interference in connection with vested liberty, in the tradition of Bentham, Austin and Hohfeld. In our formal model $\mathcal{P}_p(a)$ means not only that $p$ may attempt to perform $a$ — it means that $p$ would succeed in doing $a$ should she try. If the notion of *attempting* to do an action $a$ that can be interfered by others needs to be modelled, then another action $attempt\_a$ should be added and the permission placed onto the latter. Another alternative is to introduce modalities for trying, as in Santos *et al.* [12].

Makinson also shifts the view when addressing *power*, presenting the modern assumption, followed by most authors nowadays, where a power is a permission to dynamically bring about changes in the deontic norms that are valid in a particular state. Actions that modify contracts are beyond the scope of our work. Jones and Sergot [13] take over the analysis of power, specially institutionalised power — introducing a modality to express that an agent brings about a state of affairs ($E_xS$), which allows them to state that an agent should bring about that another one brings about some particular state ($E_xE_yS$).

The number of atomic types is a subject for debate. Kanger *et al.* [2] presents 26, later extended to 35 by Lindahl's [14]. The same work takes them to 127 if *collectivistic* propositions are considered.[4] Sergot [15] presents a detailed comparative analysis. Neither of them works in the context of interacting two-party systems, where deontic modalities applied to a party also place onus over the other, thus reducing the number of consistent atomic types, as explained in Sections 3 and 4. In keeping with Kanger, we identified only atomic types which include all the rights — whether positively or negatively. However, in our setting, further analysis can be performed to consider the possibilities when a particular right is not present — neither positively nor negatively.

From a semantics point of view, most of the attempts at formalising the Hohfelian concepts went no further than structured language, leaving many questions unresolved. For instance, Sartor [16] introduces the concept of directed modalities to express sentences like *'It is obligatory that Tom pays Mary $1000 in order to advance Mary's interests'*. If Mary is using the money to pay a blackmailer or to buy cancer-causing cigarettes, is she advancing her interests? According to whom? Can Tom deny the paying claiming that she would not use the money *'to advance her interests'*? [15]

[4] Collectivistic propositions are the ones that place the burden of obligation in more than one agent i.e., *'it is mandatory that agent a or agent b perform action c'*.

is more precise about which operator combinations are consistent given a few assumptions about the underlying logic, but because it only considers some basic modalities, and because the logic is not fixed, we still do not know if, for example, being empowered but forbidden makes any sense.

## 7 Conclusions

In this article we extended our formalisation of contracts for two-party interacting systems [3] to deal with absence of actions, mutually exclusive actions and conflicts. That allowed us to give formal semantics to Kanger's types of rights in the context of action-based and interacting two-party systems. In turn, it showed that the number of atomic types (maximally consistent sets of rights) is reduced in this context compared to Kanger's *et al.* original formulation.

Also interestingly, all of Kanger's rights (claim, power, freedom, immunity and their negated versions) can be expressed in terms of positive and negative permission and obligation, over presence or lack of actions. An interesting next step would be to present an automata-based formalism in which there are multiple parties, but also general obligations and permissions. This would allow us to reason about general obligations (such as *'forbidden to kill'*), and how they interact with contracts.

## References


1. Hohfeld, W.: Some fundamental legal conceptions as applied in judicial reasoning. Yale Lj **23** (1913) 16
2. Kanger, S., Kanger, H.: Rights and parliamentarism. Theoria **32** (1966) 85–115
3. Pace, G., Schapachnik, F.: Permissions in contracts, a logical insight. In: The 24th International Conference on Legal Knowledge and Information Systems. University of Vienna, Austria. Frontiers in Artificial Intelligence and Applications, IOS Press (2011) en prensa.
4. Arnold, A.: Nivat's processes and their synchronization. Theor. Comput. Sci. **281** (2002) 31–36
5. Pace, G., Schapachnik, F.: Types of rights in interacting two-party system: A formal analysis. Technical report, FCEyN, Universidad de Buenos Aires (2012)
6. Governatori, G., Milosevic, Z.: Dealing with contract violations: formalism and domain specific language. In: EDOC Enterprise Computing Conference, 2005 Ninth IEEE International, IEEE (2005) 46–57
7. Marjanovic, O., Milosevic, Z.: Towards formal modeling of e-contracts. In: Proceedings of the 5th IEEE International Conference on Enterprise Distributed Object Computing. EDOC '01, Washington, DC, USA, IEEE Computer Society (2001) 59–
8. Tan, Y.H., Thoen, W.: Modeling directed obligations and permissions in trade contracts. In: Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences. Volume 5., Washington, DC, USA, IEEE Computer Society (1998) 166–
9. Fasli, M.: On commitments, roles, and obligations. In: Revised Papers from the Second International Workshop of Central and Eastern Europe on Multi-Agent Systems: From Theory to Practice in Multi-Agent Systems. CEEMAS '01, London, UK, Springer-Verlag (2002) 93–102
10. Farrell, A., Sergot, M., Salle, M., Bartolini, C.: Using the event calculus for tracking the normative state of contracts. International Journal of Cooperative Information Systems **14** (2005) 99–130

11. Makinson, D.: On the formal representation of rights relations. Journal of philosophical Logic **15** (1986) 403–425
12. Santos, F., Jones, A., Carmo, J.: Action concepts for describing organised interaction. In: System Sciences, 1997, Proceedings of the Thirtieth Hawaii International Conference on. Volume 5., IEEE (1997) 373–382
13. Jones, A., Sergot, M.: A formal characterisation of institutionalised power. Logic Journal of IGPL **4** (1996) 427
14. Lindahl, L.: Position and change: A study in law and logic. Volume 112. Springer (1977)
15. Sergot, M.: A computational theory of normative positions. ACM Transactions on Computational Logic (TOCL) **2** (2001) 581–622
16. Sartor, G.: Fundamental legal concepts: A formal and teleological characterisation*. Artificial Intelligence and Law **14** (2006) 101–142