UNIVERSITY
OF TRENTO

DIPARTIMENTO DI INGEGNERIA E SCIENZA DELL'INFORMAZIONE

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.disi.unitn.it

A LOGIC FOR CONTRACTS

Massimo Bartoletti and Roberto Zunino

June 2009

Technical Report #DISI-09-034

A Logic for Contracts

Massimo Bartoletti
Dipartimento di Matematica e Informatica
Università degli Studi di Cagliari, Italy

Roberto Zunino
Dipartimento di Ingegneria e Scienza dell'Informazione
Università degli Studi di Trento, Italy

Abstract

We introduce a logic for modelling contractual committment, and study its properties. Our logic is free from modalities, and instead relies on a peculiar form of implication. In this paper, we discuss proof theory for our logic. We present a Hilbert-style axiomatisation for our logic, which is shown consistent. We also provide a Gentzen-style sequent calculus, and prove it equivalent to our axiomatization. We prove our logic decidable.

1 Introduction

Security, trustworthiness and reliability of software systems are crucial issues in the rising Information Society. As new online services (e-commerce, e-banking, e-government, etc.) are made available, the number and the criticality of the problems related to non-functional properties of services keeps growing. From the client point of view, it is important to be sure that, e.g., after a payment has been made, then either the payed goods are made available, or a full refund is issued. From the provider point of view it is important to be sure, e.g., that a client will not repudiate a completed transaction, so to obtain for free the goods already delivered. In other words, the interaction between a client and a service must be regulated by a suitable contract, which guarantees to both parties the properties on demand. The crucial problem is then how to define the concept of *contract*, and how to actually enforce it, in an environment - the Internet - which is by definition open and unreliable.

Unfortunately, at the present no widespread technology exists, as well as no strong theoretical foundation, which gives a general solution to this problem. Typically, services do not provide the client with any concrete guarantee about the actual functionality they implement. At best, the service provider commits himself to respect some "service level agreement". In the case this is not honoured, the only thing the client can do is to take legal steps against the provider (or *vice versa*). Although this is the normal practice nowadays, it is highly desirable to reverse this trend. Indeed, both clients and services could incur relevant expenses due to the needed legal disputes. This is impractical, especially for transactions dealing with small amounts of money.

Contributions. We propose to study the theoretical foundations upon which constructing a service infrastructure where contracts carry, besides the usual legal meaning, also a "formal" one. In other words, our contracts will be mathematical entities, that specify exactly the rights and the duties of clients and services. We devise a world of services where clients and service providers can have precise, mathematical guarantees about the functionality implemented by a service, and about the assumed side conditions. In the scenario we aim at, contesting a contract will not necessarily require to resort to a court, yet it will be an event managed automatically, deterministically and inexpensively, by the service infrastructure itself.

In this paper we begin our investigation by studying formalisms to describe contracts. A contract is a binding agreement between two or more parties, that dictates the duties the involved parties must fulfill, whenever some preconditions are satisfied. Our theory of contracts will be able to infer, in each possible context, the duties deriving from a given set of contracts.

In Section 2 we give, with the help of an example, some motivations about the need for a logic for contracts. In Section 3 we characterize this logic through a set of properties that we would expect to be enjoyed by any logic for contracts. In Section 4 we synthetise a set of Hilbert-style axioms, that imply all the desired properties. In Section 5 we give further details and examples about using our logic to model a variety of contracts. In Section 6 we study relations with other logics, in particular with intuitionistic propositional logic IPC, with the modal logic S4, and with propositional lax logic. In Section 7 we propose a sequent calculus for the logic, which is equivalent to the Hilbert system. The main technical result is in Section 8, where we prove cut elimination for our sequent calculus. Together with the subformula property (enjoyed by the sequent calculus), this paves us the way to prove decidability for the logic. In Section 9 we discuss some possible future work and extensions to our logic. To put the developed theory at work, we have implemented a proof search tool, which decides whether a given formula is a tautology or not.

2 Motivations

Suppose there are two kids, Alice and Bob, who want to play together. Alice has a toy airplane, while Bob has a bike. Both Alice and Bob wish to play with each other's toy: Alice wants to ride Bob's bike, and Bob wants to play with Alice's airplane. Alice and Bob are very meticulous kids, so before sharing their toys, they stipulate the following "gentlemen's agreement":

Alice: I will lend my airplane to you, Bob, provided that I borrow your bike.

Bob: I will lend my bike to you, Alice, provided that I borrow your airplane.

We want to make precise the commitments exchanged by Alice and Bob, so to be able to formally deduce that Alice and Bob will indeed share their toys, provided they are real "gentlemen" who always respect their promises.

Let us write a for the atomic proposition "Alice lends her airplane" and b for "Bob lends his bike". Using classical propositional logic, a straightforward – yet naïve – formalization of the above commitments could be the following. Alice's commitment A is represented as the formula $\mathsf{b} \to \mathsf{a}$ (if Bob lends his bike, then

Alice lends her airplane) and Bob's commitment as the formula $\mathsf{a} \to \mathsf{b}$ (if Alice lends her airplane, then Bob lends his bike):

$$A \;=\; \mathsf{b} \to \mathsf{a} \qquad\qquad B \;=\; \mathsf{a} \to \mathsf{b}$$

where the symbol $\to$ denotes classical implication. Under the hypothesis that Alice and Bob always respect their promises, both formulas A and B are *sound* with respect to our scenario. For the formula A, it is true that whenever b holds (Bob lends his bike), then a will also hold (Alice lends her airplane). For the formula B, it is true that whenever a holds (Alice lends her airplane), then b will also hold (Bob lends his bike).

So, why are we unhappy with the above formalization? The problem is that, in classical propositional logic, the above commitments are not enough to deduce that Alice will lend her airplane and Bob will lend his bike. Formally, it is possible to make true the formula $A \wedge B$ by assigning false to both propositions a and b. So, Alice and Bob will not be able to play together, despite of their gentlemen's agreement, and of the hypothesis that they always respect promises.

The failure to represent scenarios like the one above seems related to the "standard" interpretation of the Modus Ponens. In both classical and intuitionistic proof theories, the Modus Ponens rule allows to deduce b whenever $\mathsf{a} \to \mathsf{b}$ and a are true. Back to our scenario, we could deduce that Bob lends his bike, but only *after* Alice has lent Bob her airplane. One of the two parties must "take the first step" in order to make the agreement become effective, that is to imply the promised duties. In a logic for mutual agreements, we would like – instead – to make an agreement become effective also without the need of some party taking the first step (as we shall see in a while, such party might not even exists in more complex scenarios). That is, A and B are *contracts*, that once stipulated imply the duties promised by all the involved parties.

Technically, we would like our logic able to deduce $\mathsf{a} \wedge \mathsf{b}$ whenever $A \wedge B$ is true. As we have noticed above, this does not hold neither in classical nor in intuitionistic propositional logic, where the behaviour of implication strictly adheres to Modus Ponens.

To model contracts, we extend intuitionistic propositional logic IPC with a new form of implication, which we denote with the symbol $\twoheadrightarrow$. For instance, the contract declared by Alice, "I will lend my airplane to Bob provided that Bob lends his bike to me", will be written $\mathsf{b} \twoheadrightarrow \mathsf{a}$. This form of, say, *contractual* implication, is stronger than the standard implication $\to$ of IPC. Actually, $\mathsf{b} \twoheadrightarrow \mathsf{a}$ implies a not only when b is true, like IPC implication, but also in the case that a "compatible" contract, e.g. $\mathsf{a} \twoheadrightarrow \mathsf{b}$, holds. In our scenario, this means that Alice will lend her airplane to Bob, provided that Bob agrees to lend his bike to Alice whenever he borrows Alice's airplane, and *vice versa*. Actually, the following formula is a theorem of our logic:

$$(\mathsf{b} \twoheadrightarrow \mathsf{a}) \;\wedge\; (\mathsf{a} \twoheadrightarrow \mathsf{b}) \;\to\; \mathsf{a} \wedge \mathsf{b}$$

In other words, from the "gentlemen's agreement" stipulated by Alice and Bob, we can deduce that the two kids will indeed share their toys.

To make our scenario a bit more interesting, suppose now that a third kid, Carl, joins Alice and Bob. Carl has a comic book, which he would share with Alice and Bob, provided that he can play with the other kids' toys. To accommodate their commitments to the new scenario, the three kids decide to stipulate the following gentlemen's agreement (which supersedes the old one):

Alice: I will share my airplane, provided that I can play with Bob's bike and read Carl's comic book.

Bob: I will share my bike, provided that I can play with Alice's airplane and read Carl's comic book.

Carl: I will share my comic book, provided that I can play with Alice's airplane and ride Bob's bike.

Let us write a for "Alice shares her airplane", b for "Bob shares his bike", and c for "Carl shares his comic book". Then, the above commitments can be rephrased as: Alice promises a provided that b and c, Bob promises b provided that a and c, and Carl promises c provided that a and b. In our contract logics, we model the above agreement as the formula $A \wedge B \wedge C$, where:

$$A \;=\; (\mathsf{b} \wedge \mathsf{c}) \twoheadrightarrow \mathsf{a} \qquad B \;=\; (\mathsf{a} \wedge \mathsf{c}) \twoheadrightarrow \mathsf{b} \qquad C \;=\; (\mathsf{a} \wedge \mathsf{b}) \twoheadrightarrow \mathsf{c}$$

The proof system of our logic will be able to deduce that the three kids will indeed share their toys, that is, the following is theorem of the logic:

$$A \wedge B \wedge C \;\;\rightarrow\;\; \mathsf{a} \wedge \mathsf{b} \wedge \mathsf{c}$$

It it interesting to compare the specification above, which uses contractual implication, with a specification which uses, instead, standard implication. Let:

$$A' \;=\; (\mathsf{b} \wedge \mathsf{c}) \rightarrow \mathsf{a} \qquad B' \;=\; (\mathsf{a} \wedge \mathsf{c}) \rightarrow \mathsf{b} \qquad C' \;=\; (\mathsf{a} \wedge \mathsf{b}) \rightarrow \mathsf{c}$$

Clearly, in this case we cannot deduce $A' \wedge B' \wedge C' \rightarrow \mathsf{a} \wedge \mathsf{b} \wedge \mathsf{c}$. This scenario provide us with a further insight about contractual implication. Reconsider for a moment the scenario with only two contracting parties, modelled with standard implication: $(\mathsf{a} \rightarrow \mathsf{b}) \wedge (\mathsf{b} \rightarrow \mathsf{a})$. We have shown above that, in such a situation, a single party can make the agreement effective, e.g. Alice can take the first step, and lend her airplane to Bob (then, by Modus Ponens, Bob will lend his bike to Alice). Instead, in the extended scenario (modelled with standard implication), it is no longer the case that a single party can take the first step and achieve the same goal. For instance, assume that Alice decides unilaterally to share her airplane. Even by doing that, Alice will have no guarantee that she will eventually be able to play with the other kids' toys. This is because, with standard implication, setting a to true would transform $A' \wedge B' \wedge C'$ into $(\mathsf{c} \rightarrow \mathsf{b}) \wedge (\mathsf{b} \rightarrow \mathsf{c})$, which clearly implies neither b nor c. To make their agreement effective, at least two of the three parties must use contractual implication in their commitments (while the other one can use standard implication).

This observation can be generalised to a scenario with n contracting parties, where one can show that at least $n-1$ parties must use contractual implication.

3 Desirable properties

We now discuss some desirable properties of a logic for contracts, as well as some other properties that – instead – are undesirable. In the next section, we will show an axiomatisation that enjoys all the properties marked here as desired.

As shown in the previous section, a characterizing property of contractual implication is that of allowing two contracting parties to "handshake", so to make their agreement effective. This is resumed by the following *handshaking* property, which we expect to hold for any logic for contracts:

$$(p \twoheadrightarrow q) \land (q \twoheadrightarrow p) \;\rightarrow\; p \land q \tag{1}$$

A generalisation of the above property to the case of n contracting parties is also desirable. It is a sort of "circular" handshaking, where the $(i+1)$-th party, in order to promise some duty p_{i+1}, relies on a promise p_i made by the i-th party. In the case of n parties, we would expect the following:

$$(p_1 \twoheadrightarrow p_2) \land \cdots \land (p_{n-1} \twoheadrightarrow p_n) \land (p_n \twoheadrightarrow p_1) \;\rightarrow\; p_1 \land \cdots \land p_n \tag{2}$$

As an example, consider, an e-commerce scenario where a Buyer can buy items from a Seller, and pay them through a credit card. The contracts issued by the Buyer and by the Seller could be e.g.:

Buyer: I will click "pay" provided that the Seller will ship my item

Seller: I will ship your item provided that I get the money

To fill the gap between the Buyer and the Seller, there is a Bank which manages payment, with the following contract:

Bank: I will transfer money to the Seller provided that the Buyer clicks "pay".

Let the atomic propositions ship, $\mathsf{clickpay}$, and pay denote respectively the facts "Seller ships item", "Buyer clicks pay", and "Bank transfers money". Then, the three contracts above can be modelled as follows:

$$\mathit{Buyer} = \mathsf{ship} \twoheadrightarrow \mathsf{clickpay} \qquad \mathit{Bank} = \mathsf{clickpay} \twoheadrightarrow \mathsf{pay} \qquad \mathit{Seller} = \mathsf{pay} \twoheadrightarrow \mathsf{ship}$$

Then, by the handshaking property (2), we obtain a successful transaction:

$$\mathit{Buyer} \land \mathit{Bank} \land \mathit{Seller} \;\rightarrow\; \mathsf{pay} \land \mathsf{ship}$$

Note that, in the special case that n equals 1, the above "circular" handshaking property turns into a particularly simple form:

$$(p \twoheadrightarrow p) \;\rightarrow\; p \tag{3}$$

Intuitively, (3) can be interpreted as the fact that promising p provided that p, implies p (actually, also the converse holds, so that promise is equivalent to p).

A generalisation of the scenario of the previous section to the case of n kids is also desirable. It is a sort of "greedy" handshaking property, because now a party promises p_i only provided that *all* the other parties promise their duties, i.e. $p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_n$. The greedy handshaking can then be stated as:

$$\bigwedge_{i \in 1..n} \Big((p_1 \land \ldots \land p_{i-1} \land p_{i+1} \land \ldots \land p_n) \twoheadrightarrow p_i \Big) \;\rightarrow\; p_1 \land \cdots \land p_n \tag{4}$$

As shown by (1), a contract $p \twoheadrightarrow q$ becomes effective, i.e. implies the promise q, when it is matched by a dual contract $q \twoheadrightarrow p$. Even more directly, $p \twoheadrightarrow q$ should be effective also in the case that the premise p is already true:

$$p \land (p \twoheadrightarrow q) \;\rightarrow\; q \tag{5}$$

In other words, contractual implication should be *stronger* than standard implication, i.e. we expect that the following is a theorem of any logic for contracts:

$$(p \twoheadrightarrow q) \rightarrow (p \rightarrow q) \tag{6}$$

On the other hand, we do not want that also the converse holds, since this would equate the two forms of implication:

$$(p \rightarrow q) \rightarrow (p \twoheadrightarrow q) \qquad \text{NOT A TAUTOLOGY}$$

We want contractual implication to share with standard implication a number properties. We discuss some of them below. First, a contract that promises true (written $\top$) is always satisfied, regardless of the precondition. Then, we expect the following tautology:

$$p \twoheadrightarrow \top \tag{7}$$

However, differently from standard implication, we do not want that a contract with a false precondition (written $\bot$) always holds.

$$\bot \twoheadrightarrow p \qquad \text{NOT A TAUTOLOGY}$$

So see why, assume that we have $\bot \twoheadrightarrow p$ as a tautology, for all p. Then, it would also be the case for $p = \bot$, and so by the binary handshaking property we would deduce a contradiction: $(\bot \twoheadrightarrow \bot) \wedge (\bot \twoheadrightarrow \bot) \rightarrow \bot$.

Another property of implication that we want to preserve is transitivity:

$$(p \twoheadrightarrow q) \wedge (q \twoheadrightarrow r) \;\rightarrow\; (p \twoheadrightarrow r) \tag{8}$$

Back to our previous example, transitivity would allow the promise of the Buyer ($\mathsf{ship} \twoheadrightarrow \mathsf{clickpay}$) and the promise of the Bank ($\mathsf{clickpay} \twoheadrightarrow \mathsf{pay}$) to be combined in the promise $\mathsf{ship} \twoheadrightarrow \mathsf{pay}$.

Contractual implication should also enjoy a stronger form of transitivity. We illustrate it with the help of an example. Suppose an air-flight customer wants to book a flight. To do that, he issues the following contract:

$$\mathit{Customer} : \mathsf{bookFlight} \twoheadrightarrow \mathsf{pay}$$

The contract states that the customer promises to pay the required amount, provided that he obtains a flight reservation. Suppose now that an airline company starts a special offer, in the form of a free drink for each customer that makes a reservation:

$$\mathit{AirLine} : \mathsf{pay} \twoheadrightarrow \mathsf{bookFlight} \wedge \mathsf{freeDrink}$$

Of course, the two contracts should give rise to an agreement, because the airline company is promising a service that is more convenient than the precondition required be the customer contract. To achieve that, we expect to be able to "weaken" the contract of the airline company, to make it match the contract issued by the customer:

$$\mathit{AirLine} \quad \rightarrow \quad (\mathsf{pay} \twoheadrightarrow \mathsf{bookFlight})$$

Alternatively, one could make the two contracts match by making stronger the precondition required by the customer, that is:

$$Customer \;\rightarrow\; (\mathsf{bookFlight} \wedge \mathsf{freeDrink} \twoheadrightarrow \mathsf{pay})$$

More in general, we want the following two properties hold for any logic for contracts. They say that the promise in a contract can be arbitrarily weakened (9), while the precondition can be arbitrarily strengthened (10).

$$(p \twoheadrightarrow q) \;\wedge\; (q \rightarrow q') \;\rightarrow\; (p \twoheadrightarrow q') \tag{9}$$

$$(p' \rightarrow p) \;\wedge\; (p \twoheadrightarrow q) \;\rightarrow\; (p' \twoheadrightarrow q) \tag{10}$$

Note that the properties (8), (9) and (10) cover three of the four possible cases of transitivity properties which mix standard and contractual implication. Observe, instead, that combining two implications into a contract is *not* a desirable property of any logic for contracts, for the same reason for which we do not want standard and contractual implications be equivalent.

$$(p \rightarrow q) \;\wedge\; (q \rightarrow r) \;\rightarrow\; (p \twoheadrightarrow r) \qquad \text{NOT A TAUTOLOGY}$$

Another property that should hold is that, if a promise q is already true, then it is also true any contract which promises q:

$$q \;\rightarrow\; (p \twoheadrightarrow q) \tag{11}$$

Of course, we do not want the converse to hold: it is not always the case that a contract implies its promise.

$$(p \twoheadrightarrow q) \rightarrow q \qquad \text{NOT A TAUTOLOGY}$$

4 A Logic for Contracts

In this section we give the basic ingredients of our logic for contracts PCL. In Sect. 4.1 we present the syntax of PCL; in Sect. 4.2 we provide it with an Hilbert-style axiomatization. Finally, in Sect. 4.3 we study some interesting properties of PCL that follow from the given axioms.

4.1 Syntax

The syntax of PCL is a simple extension of IPC. It includes the standard logic connectives $\neg, \wedge, \vee, \rightarrow$ and the contractual implication connective $\twoheadrightarrow$. We assume a denumerable set $\{\mathsf{p}, \mathsf{q}, \mathsf{r}, \mathsf{s}, \ldots\}$ of prime (atomic) formulas. Generic formulas are denoted with the letters $p, q, r, s, \ldots$ (note that the font differs from that used for prime formulas). The precedence of IPC operators is the following, from highest to lowest: $\neg, \wedge, \vee, \rightarrow$. We stipulate that $\twoheadrightarrow$ has the same precedence as $\rightarrow$.

Definition 1. The formulas of PCL are inductively defined by the following grammar.

$$
\begin{array}{rcl}
p ::= & \bot & \text{false} \\
| & \top & \text{true} \\
| & \mathsf{p} & \text{prime} \\
| & \neg p & \text{negation} \\
| & p \vee p & \text{disjunction} \\
| & p \wedge p & \text{conjunction} \\
| & p \rightarrow p & \text{implication} \\
| & p \twoheadrightarrow p & \text{contractual implication}
\end{array}
$$

We let $p \leftrightarrow q$ be syntactic sugar for $(p \rightarrow q) \wedge (q \rightarrow p)$. If a formula is $\twoheadrightarrow$-free, we say it is an IPC formula.

We use $\Longrightarrow$ for implication in the meta-theory, to avoid confusion with $\rightarrow$.

4.2 Proof Theory: Hilbert-style Axiomatization

We now define a Hilbert-style proof system for PCL. The axioms include all the standard axioms for IPC (see e.g. [9]). Like IPC, we have a single inference rule, i.e. Modus Ponens. The characterising axioms for PCL are called *Zero*, *Fix* and *PrePost*.

Definition 2. The axioms of PCL are presented below.

- Core IPC axioms.

$$
\begin{array}{ll}
p \wedge q \rightarrow p & \wedge 1 \\
p \wedge q \rightarrow q & \wedge 2 \\
p \rightarrow q \rightarrow p \wedge q & \wedge 3 \\
p \rightarrow p \vee q & \vee 1 \\
q \rightarrow p \vee q & \vee 2 \\
(p \rightarrow r) \rightarrow (q \rightarrow r) \rightarrow (p \vee q) \rightarrow r & \vee 3 \\
p \rightarrow q \rightarrow p & \rightarrow 1 \\
(p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r & \rightarrow 2 \\
\bot \rightarrow p & \bot \\
\top & \top \\
\neg p \rightarrow p \rightarrow q & \neg 1 \\
(p \rightarrow q) \rightarrow (p \rightarrow \neg q) \rightarrow \neg p & \neg 2
\end{array}
$$

- Contractual implication axioms.

$$
\begin{array}{ll}
\top \twoheadrightarrow \top & \textit{Zero} \\
(p \twoheadrightarrow p) \rightarrow p & \textit{Fix} \\
(p' \rightarrow p) \rightarrow (p \twoheadrightarrow q) \rightarrow (q \rightarrow q') \rightarrow (p' \twoheadrightarrow q') & \textit{PrePost}
\end{array}
$$

- *Modus ponens* (cut).

$$
\frac{p \quad p \rightarrow q}{q}\,\textit{cut}
$$

We write $\vdash p$ when p is derivable from the axioms and inference rules above.

The contractual implication axioms are actually special cases of the desired properties of contracts seen in Sect. 3. For instance, axiom Zero is a special case of (7). Axiom Fix is exactly property (3). Axiom PrePost performs the function of both properties (10) and (9).

4.3 Fundamental Consequences

We present below some significant consequences of the axioms in Def. 2. Note that these consequences cover all the "desirable" properties discussed in Sect. 3. To shorten notation, when speaking about non-provability, we write $\nvdash$ to denote that a formula is not a theorem in the general case, i.e. for all the instantiations of the metavariables. E.g. we write $\nvdash p \to p \land q$ to mean $\neg\forall p, q.\ \vdash p \to p \land q$.

Lemma 3. *Contractual implication is strictly stronger than implication.*

$$\vdash (p \twoheadrightarrow q) \to (p \to q) \tag{12}$$
$$\nvdash (p \to q) \to (p \twoheadrightarrow q) \tag{13}$$

Proof. For (12), assume that $p \twoheadrightarrow q$ and p hold. Hence, $q \to p$ trivially holds. By using *PrePost* on $q \to p$,$p \twoheadrightarrow q$, and $q \to q$, we get $q \twoheadrightarrow q$. We then conclude q by *Fix*.

We anticipate in (13) a negative result that can be mechanically verified using the decision procedure of Lemma 50. □

Below, we establish some further connections between $\twoheadrightarrow$ and $\to$, which generalize the transitivity of $\twoheadrightarrow$.

Lemma 4. *Contractual implication is transitive. More in detail, we have the following interactions between $\twoheadrightarrow$ and $\to$.*

$$\vdash (p \twoheadrightarrow q) \to (q \twoheadrightarrow r) \to (p \twoheadrightarrow r) \tag{14}$$
$$\vdash (p \to q) \to (q \twoheadrightarrow r) \to (p \twoheadrightarrow r) \tag{15}$$
$$\vdash (p \twoheadrightarrow q) \to (q \to r) \to (p \twoheadrightarrow r) \tag{16}$$
$$\nvdash (p \to q) \to (q \to r) \to (p \twoheadrightarrow r) \tag{17}$$

Proof. Properties (15,16) are direct consequences of PrePost, and the trivial $r \to r$ and $p \to p$.

For (14), we apply Lemma 3(12) to $p \twoheadrightarrow q$, so obtaining $p \to q$. Then we can apply (15) to conclude.

We anticipate in (17) a negative result that can be mechanically verified using the decision procedure of Lemma 50. □

The distributivity laws of $\twoheadrightarrow$ are quite peculiar. As for standard implication in IPC, $\lor$-distributivity holds in only one direction (18,19). Instead, while $\land$-distributivity holds in both directions in IPC for standard implication, contractual implication only satisfies one direction (20,21). However, a related property holds (22).

Lemma 5. *Distributivity laws.*

$$\vdash (p \twoheadrightarrow q) \lor (p \twoheadrightarrow r) \to (p \twoheadrightarrow (q \lor r)) \tag{18}$$
$$\nvdash (p \twoheadrightarrow (q \lor r)) \to (p \twoheadrightarrow q) \lor (p \twoheadrightarrow r) \tag{19}$$
$$\vdash (p \twoheadrightarrow (q \land r)) \to (p \twoheadrightarrow q) \land (p \twoheadrightarrow r) \tag{20}$$
$$\nvdash (p \twoheadrightarrow q) \land (p \twoheadrightarrow r) \to (p \twoheadrightarrow (q \land r)) \tag{21}$$
$$\vdash (p \twoheadrightarrow q) \land (q \twoheadrightarrow r) \to (p \twoheadrightarrow (q \land r)) \tag{22}$$

Proof. For (18), assume $(p \twoheadrightarrow q) \vee (p \twoheadrightarrow r)$. If $p \twoheadrightarrow q$ holds, we apply *PrePost* to weaken q to $q \vee r$. The $p \twoheadrightarrow r$ case is similar.

For (20), assume $p \twoheadrightarrow (q \wedge r)$. By *PrePost*, it is then easy to obtain both $p \twoheadrightarrow q$ and $p \twoheadrightarrow r$.

We anticipate in (19,21) some negative results that can be mechanically verified using the decision procedure of Lemma 50.

For (22), assume the hypotheses. We apply Lemma 3 to $q \twoheadrightarrow r$ and obtain $q \rightarrow r$, hence $q \rightarrow (q \wedge r)$. By *PrePost* on $p \twoheadrightarrow q$, we obtain the thesis. □

Lemma 6. *Substitution of equivalent formulas.*

$$\vdash (p \leftrightarrow p') \rightarrow (q \leftrightarrow q') \rightarrow (p \twoheadrightarrow q) \rightarrow (p' \twoheadrightarrow q')$$

Proof. Apply *PrePost* to $p' \rightarrow p$, $p \twoheadrightarrow q$, and $q \rightarrow q'$. □

The following lemma states a sufficient condition and a necessary condition for $p \twoheadrightarrow q$ to hold. These conditions are expressed in IPC, i.e. they make no use of $\twoheadrightarrow$. We will return on these in Def. 21,23 and related results, where we will prove that, when p, q are IPC formulas, these conditions are actually the weakest sufficient condition and the strongest necessary condition that can be expressed within IPC (Lemma 25).

Lemma 7. *Contractual implication admits the following sufficient condition and necessary condition.*

$$\vdash q \rightarrow (p \twoheadrightarrow q) \tag{23}$$

$$\vdash (p \twoheadrightarrow q) \rightarrow ((q \rightarrow p) \rightarrow q) \tag{24}$$

Proof. For (23), assume q. We conclude by *PrePost* on $p \rightarrow \top$ (trivial), $\top \twoheadrightarrow \top$ (by *Zero*), $\top \rightarrow q$ (by q).

For (24), assume $p \twoheadrightarrow q$ and $q \rightarrow p$. By *PrePost*, we have $q \twoheadrightarrow q$, so we conclude by *Fix*. □

The following lemma justifies our choice of IPC as the basis for our logic. The lemma shows that using CPC instead would make our contractual implication much less interesting.

Lemma 8. *Denote with $\vdash_C p$ the provability of p in the system of Def. 2 augmented with the axiom of excluded middle ($p \vee \neg p$). We have:*

$$\vdash_C (p \twoheadrightarrow q) \leftrightarrow q$$

Proof. Direct consequence of Lemma 7, since $((q \rightarrow p) \rightarrow q) \rightarrow q$ is a tautology of CPC (actually, it is the well-known Peirce's law). □

The following two lemmata are about "handshaking" of n contracting parties. Lemma 9 speaks about "circular" handshaking where the i-th party relies on the promise made by the $i-1$-th party, as in (2). Lemma 10 is a stronger version of this, because it allows each party to rely on the promises made by *all* the other parties, as in (4).

Lemma 9. (Handshaking) *For all $n \geq 0$ and for all $p_0, \ldots, p_n$:*

$$\vdash (p_0 \twoheadrightarrow p_1) \rightarrow \cdots \rightarrow (p_{n-1} \twoheadrightarrow p_n) \rightarrow (p_n \twoheadrightarrow p_0) \rightarrow (p_0 \wedge \ldots \wedge p_n)$$

Proof. Assume all the hypotheses. By repeated application of Lemma 4, we have $p_i \twoheadrightarrow p_i$ for all $i \in 0..n$. We then conclude by *Fix*. $\square$

Lemma 10. (Greedy handshaking) *For all $n \geq 0$, for all $p_0, \ldots, p_n$, and for all $i, j \in 0..n$:*

$$\vdash \bigwedge_i \Big(\Big(\bigwedge_{j \neq i} p_j \Big) \twoheadrightarrow p_i \Big) \rightarrow \bigwedge_i p_i \tag{25}$$

Proof. By induction on n. The base case $n = 0$ is simple: $(\top \twoheadrightarrow p_0) \rightarrow p_0$ is proved by applying *PrePost* to $\top \twoheadrightarrow p_0$, hence obtaining $p_0 \twoheadrightarrow p_0$ and concluding by *Fix*.

In the inductive case, assuming that (25) holds for $n - 1$ we prove it for n. We assume the hypothesis, $\left(\bigwedge_{j \neq i} p_j \right) \twoheadrightarrow p_i$ for each $i = 0..n$, and we proceed to prove the thesis $\bigwedge_i p_i$. First, we prove the following auxiliary result:

$$p_n \rightarrow \bigwedge_{j \neq n} p_j \tag{26}$$

To prove (26), assume p_n. Then, we have $\left(\bigwedge_{j \neq i} p_j \right) \leftrightarrow \left(\bigwedge_{j \neq i, j \neq n} p_j \right)$, so by Lemma 6 we get $\left(\bigwedge_{j \neq i, j \neq n} p_j \right) \twoheadrightarrow p_i$ for $i = 0..n-1$. We can apply the inductive hypothesis (25), and have $\bigwedge_{j \neq n} p_j$.

Back to the inductive case, we note that since i can be n, we have $\left(\bigwedge_{j \neq n} p_j \right) \twoheadrightarrow p_n$. We then conclude by (26,24). $\square$

5 Examples

Example 11. The toy exchange scenario from the introduction is modelled as:

$$\vdash (\mathsf{airplane} \twoheadrightarrow \mathsf{bike}) \land (\mathsf{bike} \twoheadrightarrow \mathsf{airplane}) \rightarrow (\mathsf{airplane} \land \mathsf{bike})$$

Indeed, this is a consequence of our axioms, and is a special case of Lemma 9.

Example 12. We now exploit our logic to model a typical preliminary contract for a real estate sale in Italy.

Assume a buyer who is interested in buying a new house from a given seller. Before stipulating the actual purchase contract, the buyer and the seller meet to stipulate a preliminary sale contract, that fixes the terms and conditions of the purchase. Typically, this contract will indicate the price and the date when the deed of sale will take place, and it will outline the obligations for the buyer and the seller. When the preliminary contract is signed by both parties, the buyer will typically pay a part of the sale price. By the italian laws, if the seller decides not to sell the house after having signed the preliminary contract and collected the deposit, she must pay the buyer back twice the sum received. Similarly, if the buyer changes his mind and decides not to buy the house, he loses the whole deposited amount.

We model the preliminary sale contract as two PCL formulas, one for the buyer and the other for the seller. The buyer will sign the preliminary contract

(signB), provided that the seller will actually sell her house (sellS), or she refunds twice the sum received ($\mathsf{refundS}$). Also, the buyer promises that if he signs the preliminary contract, than either he will pay the stipulated price (payB), or he will not pay and lose the deposit ($\mathsf{refundB}$)

$$Buyer: \; ((\mathsf{sellS} \vee \mathsf{refundS}) \twoheadrightarrow \mathsf{signB}) \;\wedge\; (\mathsf{signB} \twoheadrightarrow (\mathsf{payB} \vee (\neg\mathsf{payB} \wedge \mathsf{refundB})))$$

The seller promises that she will sign the preliminary contract (signS), provided that either the buyer promises to pay the stipulated amount, or he promises to lose the deposit. Also, the seller promises that is she signs the preliminary contract, then she will either sell her house, or will not sell and refund twice the sum received.

$$Seller: \; ((\mathsf{payB} \vee \mathsf{refundB}) \twoheadrightarrow \mathsf{signS}) \;\wedge\; (\mathsf{signS} \twoheadrightarrow (\mathsf{sellS} \vee (\neg\mathsf{sellS} \wedge \mathsf{refundS})))$$

A first consequence is that the two contracts lead to an agreement between the buyer and the seller, that is both parties will sign the preliminary contract:

$$Buyer \wedge Seller \rightarrow \mathsf{signB} \wedge \mathsf{signS} \tag{27}$$

As a second consequence, if one of the parties does not finalize the final deed of sale, than that party will refund the other:

$$Buyer \wedge Seller \wedge \neg\mathsf{payB} \rightarrow \mathsf{refundB} \tag{28}$$

$$Buyer \wedge Seller \wedge \neg\mathsf{sellS} \rightarrow \mathsf{refundS} \tag{29}$$

To prove the above, we proceed as follows. First, we apply transitivity (14) to *Buyer* and *Seller*:

$$(\mathsf{sellS} \vee \mathsf{refundS}) \twoheadrightarrow (\mathsf{payB} \vee (\neg\mathsf{payB} \wedge \mathsf{refundB}))$$
$$(\mathsf{payB} \vee \mathsf{refundB}) \twoheadrightarrow (\mathsf{sellS} \vee (\neg\mathsf{sellS} \wedge \mathsf{refundS}))$$

Then, we use *PrePost*:

$$(\mathsf{sellS} \vee (\neg\mathsf{sellS} \wedge \mathsf{refundS})) \twoheadrightarrow (\mathsf{payB} \vee (\neg\mathsf{payB} \wedge \mathsf{refundB}))$$
$$(\mathsf{payB} \vee (\neg\mathsf{payB} \wedge \mathsf{refundB})) \twoheadrightarrow (\mathsf{sellS} \vee (\neg\mathsf{sellS} \wedge \mathsf{refundS}))$$

So, by Lemma 9, we have that $Buyer \wedge Seller$ implies

$$(\mathsf{sellS} \vee (\neg\mathsf{sellS} \wedge \mathsf{refundS})) \wedge (\mathsf{payB} \vee (\neg\mathsf{payB} \wedge \mathsf{refundB}))$$

By (12) the above and $Buyer \wedge Seller$ imply $\mathsf{signB} \wedge \mathsf{signS}$. This proves (27). Also, the above and $\neg\mathsf{payB}$ clearly imply $\mathsf{refundB}$. The same holds for sellS and $\mathsf{refundS}$. Hence, we establish (28,29).

Example 13. We now describe a possible online sale between two parties. In order to buy an item, first the buyer has to contact the bank and reserve from his account a specific amount of money for the transaction. When this happens, that amount is no longer available for anything else. We model this reservation with the formula lock. Then, the buyer has to make an offer to the seller: this is modelled with offer. The seller, when provided with an offer, evaluates it. If she thinks the offer is good, and the money has been reserved, then she will send the item (send). Otherwise, she cancels the transaction (abort). When the transaction is aborted, the bank cancels the money reservation, so that the buyer can use the amount for other transactions (unlock).

We now formalize the scenario. The buyer agrees to $\mathsf{lock} \wedge \mathsf{offer}$, provided that either the item is sent, or the money reservation is cancelled. The seller agrees to evaluate the offer. The bank agrees to cancel the reservation when the transaction is aborted.

$$Buyer : (\mathsf{send} \vee \mathsf{unlock}) \twoheadrightarrow (\mathsf{lock} \wedge \mathsf{offer})$$
$$Seller : \mathsf{offer} \twoheadrightarrow ((\mathsf{lock} \to \mathsf{send}) \vee \mathsf{abort})$$
$$Bank : (\mathsf{lock} \wedge \mathsf{abort}) \twoheadrightarrow \mathsf{unlock}$$

Under these assumptions, we can see that either the item is sent, or the transaction is aborted and the reservation cancelled.

$$\vdash (Buyer \wedge Seller \wedge Bank) \to (\mathsf{send} \vee (\mathsf{abort} \wedge \mathsf{unlock}))$$

To prove this, first we apply *PrePost* to *Seller* and obtain $(\mathsf{lock} \wedge \mathsf{offer}) \twoheadrightarrow ((\mathsf{lock} \to \mathsf{send}) \vee \mathsf{abort})$. By property (22) and *Buyer*, we have $(\mathsf{send} \vee \mathsf{unlock}) \twoheadrightarrow (\mathsf{lock} \wedge \mathsf{offer} \wedge ((\mathsf{lock} \to \mathsf{send}) \vee \mathsf{abort}))$. By *PrePost*, we weaken it, obtaining $(\mathsf{send} \vee \mathsf{unblock}) \twoheadrightarrow (\mathsf{send} \vee (\mathsf{lock} \wedge \mathsf{abort}))$. By *Bank* and 3, we have $(\mathsf{lock} \wedge \mathsf{abort}) \to \mathsf{unlock}$, as well as $(\mathsf{lock} \wedge \mathsf{abort}) \to (\mathsf{abort} \wedge \mathsf{unlock})$. Therefore, by *PrePost* we have $(\mathsf{send} \vee \mathsf{unlock}) \twoheadrightarrow (\mathsf{send} \vee (\mathsf{abort} \wedge \mathsf{unlock}))$. We conclude by *PrePost* and *Fix*.

Example 14. (Dining retailers) Around a round table, a group of n cutlery retailers is about to have dinner. In the center of the table, there is a large dish of food. Despite the food being delicious, the retailers cannot start eating right now. To do that, and follow the proper etiquette, each retailer needs to have a complete cutlery set, consisting of n pieces, each of a different kind. Each one of the n retailers owns a distinct set of n piece of cutlery, all of the same kind. The retailers start discussing about trading their cutlery, so that they can finally eat. Since everyone wants to get a fair deal, they want to formalize their committments.

We can formalize the scenario as follows. We can number the retailers $r_1, \ldots, r_n$ together with the kinds of pieces of cutlery, so that r_i initially owns n pieces of kind number i. We then write $\mathsf{g}_{i,j}$ for "r_i *gives* a piece (of kind i) to r_j". Since retailers can use their own cutlery, we assume $\mathsf{g}_{i,i}$ to be true. Retailer r_i can start *eating* whenever $e_i = \bigwedge_j \mathsf{g}_{j,i}$.

Suppose that r_1 commits to a simple exchange with r_2: they commit to $\mathsf{g}_{2,1} \twoheadrightarrow \mathsf{g}_{1,2}$ and $\mathsf{g}_{1,2} \twoheadrightarrow \mathsf{g}_{2,1}$, and the exchange takes place since $\mathsf{g}_{2,1} \wedge \mathsf{g}_{1,2}$ can be derived. While this seems a fair deal, it actually exposes r_1 to a risk: if $r_3, \ldots, r_n$ perform a similar exchange with r_2, then we have $\mathsf{g}_{2,i} \wedge \mathsf{g}_{i,2}$ for all i. In particular, $\mathsf{g}_{i,2}$ holds for all i, so r_2 can start eating. This is however not necessarily the case for r_1, since r_3 has not committed to any exchange with r_1.

A wise retailer would then never agree to a simple exchange $\mathsf{g}_{2,1} \twoheadrightarrow \mathsf{g}_{1,2}$. Instead, the retailer r_1 could commit to a safer contract[1]:

$$\mathsf{g}_{1,1} \wedge \mathsf{g}_{2,1} \wedge \cdots \wedge \mathsf{g}_{n,1} \twoheadrightarrow \mathsf{g}_{1,1} \wedge \mathsf{g}_{1,2} \wedge \cdots \wedge \mathsf{g}_{1,n}$$

[1] We include $\mathsf{g}_{1,1} = \top$ to make it more homogeneous.

The idea is simple: r_1 requires each piece of cutlery, that is, r_1 requires to be able to start eating (e_1). When this happens, r_1 agrees to provide each other retailer with a piece of his cutlery. Now, assume ech retailer r_i commits to the analogous contract:

$$c_i = e_i \twoheadrightarrow \bigwedge_j \mathsf{g}_{i,j}$$

We can now verify that $\bigwedge_i c_i \to \bigwedge_i e_i$, that is, the above contracts actually allow everyone to eat. Assume c_i for all i, and define $p_i = \bigwedge_j \mathsf{g}_{i,j}$. Clearly, $c_i = e_i \twoheadrightarrow p_i$. Note that

$$\bigwedge_{j \neq i} p_j = \bigwedge_{j \neq i} \bigwedge_k \mathsf{g}_{j,k} \quad \to \quad \bigwedge_j \mathsf{g}_{j,i} = e_i \tag{30}$$

since we can choose $k = i$ and $\mathsf{g}_{i,i}$ is true. Therefore, by *PrePost*,

$$c_i = e_i \twoheadrightarrow p_i \quad \to \quad \Big(\bigwedge_{j \neq i} p_j\Big) \twoheadrightarrow p_i$$

By Lemma 10, since $\bigwedge_i c_i$, we have $\bigwedge_i p_i$. By applying (30) we can conclude $\bigwedge_i e_i$.

Example 15. In Sect. 3 we listed several requirements for the notion of contractual implication. A small set of these was then picked as our axiomatization: *Zero*, *Fix*, *PrePost*. However, we have not yet checked that this is indeed complete with respect to the other requirements. That is, we have to check that all the requirements are indeed theorems of PCL. We now quickly recall the list of all these requirements and provide a support for each one.

The handskaking properties (1,2) have been established in Lemma 9. Property (3) is the axiom *Fix*. The "greedy handshaking" (4) was established in Lemma 10. Property (6) was proved in Lemma 3. Property (7) is a direct consequence of *Zero*. Property (8) is proved in Lemma 4. Properties (9,10) are direct consequences of *PrePost*. Property (11) is a consequence of axioms *Zero* and *PrePost*.

6 Relations with Other Logics

In this section we explore the relationships between PCL and other logics. We are interested in possible mappings, to see if there is some way to encode PCL in some other pre-existing logic. For instance, since PCL is a direct extension of IPC, one might wonder whether the newly introduced connective for contractual implication $\twoheadrightarrow$ can be expressed using IPC connectives. We answer to this question in Sect. 6.1, where we prove that this is not the case.

In Sect. 6.2 we instead explore some mappings to the modal logic S4, by extending a well-known IPC-to-S4 mapping.

In Sect. 6.3 we discuss about the relations between our logic and PLL, the propositional lax modal logic. It turns out that our axioms are not sufficient to make $p \twoheadrightarrow \bullet$ a lax modality. We discuss why extending the axioms in this direction might prove useful.

Finally, we recall Lemma 8, which provides a justification for our use of IPC as the basis for our logic, since the principle of excluded middle (valid in e.g. CPC and S4) makes our contractual implication connective trivial.

In the proofs of this section, when we need to check tautologies of IPC or S4, we shall sometimes resort to an automatic theorem prover. To this purpose, we used the Logics WorkBench (LWB) [8]. If the reader wishes to read the actual proofs generated by LWB, in Sect. A.3 we describe how to obtain them.

6.1 Mappings for IPC

We study here the mappings that act homomorphically with respect to each IPC connective.

Definition 16. A homomorphic mapping is a function $m(\bullet)$ from PCL to IPC such that

$$m(\mathsf{p}) = \mathsf{p} \qquad (\mathsf{p}\ \text{prime})$$
$$m(p \wedge q) = m(p) \wedge m(q)$$
$$m(p \vee q) = m(p) \vee m(q)$$
$$m(p \to q) = m(p) \to m(q)$$
$$m(\neg p) = \neg m(p)$$
$$m(\top) = \top$$
$$m(\bot) = \bot$$

Note that $m(p \twoheadrightarrow q)$ is not constrained by the above. A homomorphic mapping $m(\bullet)$ is *sound* iff $\vdash p$ implies $\vdash_{IPC} m(p)$, and is *complete* iff $\vdash_A m(p)$ implies $\vdash p$.

We first state some basic properties of such mappings.

Lemma 17. *If $m(\bullet)$ is a homomorphic mapping, then $m(p \leftrightarrow q) = m(p) \leftrightarrow m(q)$.*

Proof. Trivial expansion of the syntactic sugar for $\leftrightarrow$. □

Lemma 18. *For any homomorphic mapping $m(\bullet)$ and IPC formula p, we have $m(p) = p$.*

Proof. Trivial structural induction. □

The identity provides a sound and complete *partial* mapping for IPC formulas.

Lemma 19. $\vdash_{IPC} p \implies \vdash p$.

Proof. Trivial, since the Hilbert axioms of PCL are a superset of those of IPC. □

We anticipate here a result which will be proved in Sect. 7, namely the partial completeness of the identity mapping. Note that this actually proces that PCL is a conservative extension of IPC.

Lemma 20. *For any IPC formula p, we have $\vdash p \implies \vdash_{IPC} p$.*

Proof. See Lemma 52. □

We are aware of several sound but incomplete mappings of PCL in IPC. Among these, two are peculiar in that they provide the strongest and weakest interpretations of the connective$\twoheadrightarrow$ in IPC. The formal justification for this terminology will be provided by Lemma 26.

Definition 21. The "strongest" $\twoheadrightarrow$-interpretation, $s(\bullet)$, is defined as the homomorphic mapping to IPC such that

$$s(p \twoheadrightarrow q) = s(q)$$

We now prove the soundness of s, as well as its incompleteness.

Lemma 22. $\vdash p \implies \vdash_{IPC} s(p)$. *The converse is false, in general.*

Proof. Easy induction on the derivation of $\vdash p$.

- If $\vdash p$ was derived through an IPC axiom, then we use the same axiom to derive $\vdash_{IPC} s(p)$, since s is an homomorphism.
- If $\vdash p$ was derived through a $\twoheadrightarrow$ axiom, we have the following subcases.
  - *Zero*: $p = \top \twoheadrightarrow \top$. Trivially, $\vdash_{IPC} s(p) = \top$.
  - *Fix*:$p = (r \twoheadrightarrow r) \to r$. Trivially, $\vdash_{IPC} s(p) = s(r) \to s(r)$.
  - *PrePost*: $p = (r' \to r) \to (r \twoheadrightarrow q) \to (q \to q') \to (r' \twoheadrightarrow q')$. Then,

    $$\vdash_{IPC} s(p) = (s(r') \to s(r)) \to s(q) \to (s(q) \to s(q')) \to s(q')$$

    is easy. Indeed, if we generalize the above by replacing $s(r), s(r'), s(q), s(q')$ with distinct prime formulas, the formula still holds, as it can be trivially verified by an IPC theorem prover. We used LWB for this: see Sect. A.3 for more details.
- If $\vdash p$ was derived through a cut $\dfrac{p \quad p \to q}{q}$, then by inductive hypothesis we have $\vdash_{IPC} s(p)$ and $\vdash_{IPC} s(p \to q)$, the latter being $\vdash_{IPC} s(p) \to s(q)$. We can then conclude by the cut $\dfrac{s(p) \quad s(p) \to s(q)}{s(q)}$.

The converse does not hold in general, e.g. for $p = (r \twoheadrightarrow q) \to q$. □

Below, we introduce the "weakest" interpretation w for contractual implication. This is somehow dual with respect to s, as we shall see in Lemma 25.

Definition 23. The "weakest" $\twoheadrightarrow$-interpretation, $w(\bullet)$, is defined as the homomorphic mapping to IPC such that

$$w(p \twoheadrightarrow q) = (w(q) \to w(p)) \to w(q)$$

We now prove the soundness of w, as well as its incompleteness.

Lemma 24. $\vdash p \implies \vdash_{IPC} w(p)$. *The converse is false, in general.*

Proof. Easy induction on the derivation of $\vdash p$.

- If $\vdash p$ was derived through an IPC axiom, then we use the same axiom to derive $\vdash_{IPC} w(p)$, since w is an homomorphism.
- If $\vdash p$ was derived through a $\twoheadrightarrow$ axiom, we have the following subcases.
  - *Zero*: $p = \top \twoheadrightarrow \top$. Trivially, $\vdash_{IPC} w(p) = (\top \to \top) \to \top$.
  - *Fix*:$p = (r \twoheadrightarrow r) \to r$. Then, $\vdash_{IPC} w(p) = ((w(r) \to w(r)) \to w(r)) \to w(r)$ is simple to prove.
  - *PrePost*: $p = (r' \to r) \to (r \twoheadrightarrow q) \to (q \to q') \to (r' \twoheadrightarrow q')$. Then,

$$\vdash_{IPC} w(p) = (w(r') \to w(r)) \to ((w(q) \to w(r)) \to w(q)) \to A$$
$$\text{where } A = (w(q) \to w(q')) \to ((w(q') \to w(r')) \to w(q'))$$

    can be proved in IPC. Indeed, if we generalize the above by replacing $w(r), w(r'), w(q), w(q')$ with distinct prime formulas, the formula still holds, as it can be trivially verified by an IPC theorem prover. We used LWB for this: see Sect. A.3 for more details.

- If $\vdash p$ was derived through a cut $\dfrac{p \quad p \to q}{q}$, then by inductive hypothesis we have $\vdash_{IPC} w(p)$ and $\vdash_{IPC} w(p \to q)$, the latter being $\vdash_{IPC} w(p) \to w(q)$. We can then conclude by the cut $\dfrac{w(p) \quad w(p) \to w(q)}{w(q)}$.

The converse does not hold in general, e.g. for $p = ((q \to r) \to q) \to (r \twoheadrightarrow q)$. $\square$

Below, we relate the s and w mappings by examining their behaviour on $p \twoheadrightarrow q$, where p, q are IPC formulas. Note that the lemma below does not apply when p, q make use of contractual implication.

Lemma 25. *Let p, q be IPC formulas. Lemma 7 provides the weakest sufficient condition and the strongest necessary condition for $p \twoheadrightarrow q$ that can be expressed in IPC. That is,*

1. If for an IPC formula c we have $\vdash c \to (p \twoheadrightarrow q)$, then $\vdash c \to q$.
2. If for an IPC formula c we have $\vdash (p \twoheadrightarrow q) \to c$, then $\vdash ((q \to p) \to q) \to c$.

Proof. For (1), by Lemma 22, we have $\vdash_{IPC} s(c \to (p \twoheadrightarrow q))$. That is $\vdash_{IPC} s(c) \to s(q)$, hence by Lemma 18 $\vdash_{IPC} c \to q$. We conclude by Lemma 19.

For (2), by Lemma 24, we have $\vdash_{IPC} w((p \twoheadrightarrow q) \to c)$. That is $\vdash_{IPC} ((w(q) \to w(p)) \to w(q)) \to w(c)$, hence by Lemma 18 $\vdash_{IPC} ((q \to p) \to q) \to c$. We conclude by Lemma 19. $\square$

Here we justify the terms "strongest" and "weakest" for s and w.

Lemma 26. *Let $m \leq n$ be the preorder over homomorphic mappings m, n given by*

$$\vdash m(p \twoheadrightarrow q) \to n(p \twoheadrightarrow q) \text{ for all IPC formulas } p, q$$

Then, $s(\bullet)$ is a maximum and $w(\bullet)$ is a minimum. That is, $w \leq m \leq s$ for any m.

Proof. For s, we need to show that $\vdash m(p \twoheadrightarrow q) \to s(p \twoheadrightarrow q)$. By Lemma 18, this is $\vdash m(p \twoheadrightarrow q) \to q$, which directly follows from Lemma 25:1.

For w, we proceed similarly, using Lemma 25:2. □

We proved that s and w are sound homomorphic mappings. Several others do exist, however. Below, we provide a short table of those known to us at the time of writing.

$m(p \twoheadrightarrow q) = m(q)$ (s)
$m(p \twoheadrightarrow q) = (m(q) \to m(p)) \to m(q)$ (w)
$m(p \twoheadrightarrow q) = \neg\neg(m(q) \to m(p)) \to m(q)$
$m(p \twoheadrightarrow q) = \neg(m(q) \to m(p)) \vee m(q)$
$m(p \twoheadrightarrow q) = ((m(q) \to m(p)) \vee \mathsf{a}) \to m(q)$ for any prime a

The soundness proofs for these mappings are similar to those of Lemma 22 and 24, so we omit them. However no mapping can be complete, as we show below.

Theorem 27. *There is no complete homomorphic mapping to IPC.*

Proof. By contradiction, suppose there exists an homomorphic mapping $m(\bullet)$ such that $\vdash p \iff \vdash_{IPC} m(p)$ for all p. Take $p = m(\mathsf{q} \twoheadrightarrow \mathsf{r}) \leftrightarrow (\mathsf{q} \twoheadrightarrow \mathsf{r})$ for some prime q, r. Then $m(p) = m(m(\mathsf{q} \twoheadrightarrow \mathsf{r})) \leftrightarrow m(\mathsf{q} \twoheadrightarrow \mathsf{r}) = m(\mathsf{q} \twoheadrightarrow \mathsf{r}) \leftrightarrow m(\mathsf{q} \twoheadrightarrow \mathsf{r})$ by Lemma 17. The last form is trivially provable in IPC, so we can state $\vdash_{IPC} m(p)$. By the completeness of $m(\bullet)$, we have $\vdash p$. Hence

$$\vdash m(\mathsf{q} \twoheadrightarrow \mathsf{r}) \to (\mathsf{q} \twoheadrightarrow \mathsf{r})$$
$$\vdash (\mathsf{q} \twoheadrightarrow \mathsf{r}) \to m(\mathsf{q} \twoheadrightarrow \mathsf{r})$$

By Lemma 25, we have

$$\vdash m(\mathsf{q} \twoheadrightarrow \mathsf{r}) \to \mathsf{r}$$
$$\vdash ((\mathsf{r} \to \mathsf{q}) \to \mathsf{r}) \to m(\mathsf{q} \twoheadrightarrow \mathsf{r})$$

hence, $\vdash ((\mathsf{r} \to \mathsf{q}) \to \mathsf{r}) \to \mathsf{r}$. By Lemma 20 we have $\vdash_{IPC} ((\mathsf{r} \to \mathsf{q}) \to \mathsf{r}) \to \mathsf{r}$. This is however false, since IPC can not prove Peirce's law (not even on prime formulas). □

6.2 Mappings to S4

We shall consider the extensions of a standard ICP mapping to S4 [7, 5].

Definition 28. An extended mapping to S4 is a function $m(\bullet)$ from PCL to S4 such that

$$m(\mathsf{p}) = \Box\mathsf{p}$$
$$m(p \wedge q) = m(p) \wedge m(q)$$
$$m(p \vee q) = m(p) \vee m(q)$$
$$m(p \to q) = \Box(m(p) \to m(q))$$
$$m(\top) = \top$$
$$m(\bot) = \bot$$
$$m(\neg p) = \Box\neg m(p)$$
$$m(p \twoheadrightarrow q) = \mathcal{C}[m(p), m(q)]$$

for some fixed S4 formula context $\mathcal{C}$.

We now introduce some technical lemmas.

Definition 29. A formula p is $\Box$-invariant iff $\vdash_{S4} p \leftrightarrow \Box p$.

Note that the $\leftarrow$ direction is actually true for all p.

Lemma 30. *If p, q are $\Box$-invariant, then $m(p \wedge q), m(p \vee q), m(p \to q), m(\neg p), m(\top), m(\bot), m(\mathsf{a})$ (with a prime) are such.*

Proof. The cases $\top, \bot$ are trivial. For the other, we proceed as follows. For $\wedge$, we simply apply $\vdash_{S4} \Box(a \wedge b) \to \Box a \wedge \Box b$. For $\neg, \to, \mathsf{a}$, we apply $\vdash_{S4} \Box a \to \Box\Box a$. For $\vee$, by hypothesis $m(p) \vee m(q)$ implies $(\Box m(p) \vee \Box m(q))$, which implies $\Box(m(p) \vee m(q))$. ☐

Lemma 31. *Assume that, for all $\Box$-invariant p, q, $m(p \twoheadrightarrow q)$ is $\Box$-invariant. Then, for any p, $m(p)$ is $\Box$-invariant.*

Proof. By structural induction on p. Indeed, all the inductive steps are covered by either the hypothesis or Lemma 30. ☐

Lemma 32. *If a is prime, $m(q)$ is $\Box$-invariant, and $\vdash_{S4} m(p)$, then we have $\vdash_{S4} m(p\{q/\mathsf{a}\})$.*

Proof. Clearly, $m(p) = \mathcal{C}[m(\mathsf{a})] = \mathcal{C}[\Box\mathsf{a}]$ for some context $\mathcal{C}$. So, for the same context, $m(p\{q/\mathsf{a}\}) = \mathcal{C}[m(q)]$. Since $m(q)$ is $\Box$-invariant, the latter is equivalent in S4 to $\mathcal{C}[\Box m(q)] = \mathcal{C}[\Box\mathsf{a}]\{m(q)/\mathsf{a}\} = m(p)\{m(q)/\mathsf{a}\}$. The last formula holds in S4, by substituting a in $\vdash_{S4} m(p)$. ☐

We are aware of several sound but not complete mappings to S4.

Definition 33. The extended mappings $e_1, \ldots, e_4$ to S4 are defined as follows:

$$e_1(p \twoheadrightarrow q) = (e_1(q) \to e_1(p)) \to e_1(q)$$
$$e_2(p \twoheadrightarrow q) = \Diamond(\Diamond e_2(q) \to e_2(p)) \to e_2(q)$$
$$e_3(p \twoheadrightarrow q) = \Box(\Box(e_3(q) \to e_3(p)) \to e_3(q))$$
$$e_4(p \twoheadrightarrow q) = \Box(\Box(e_4(q) \to \Diamond e_4(p)) \to e_4(q))$$

Lemma 34. *$e_i(p)$ is $\Box$-invariant.*

Proof. By Lemma 31, we only need to check that $e_i(p \twoheadrightarrow q)$ is $\Box$-invariant whenever p, q are such. If we write $\mathcal{C}_i$ for the contexts such that $e_i(p \twoheadrightarrow q) = \mathcal{C}_i[e_i(p), e_i(q)]$, then it is sufficient to verify that

$$\vdash_{S4} (\mathsf{a} \leftrightarrow \Box\mathsf{a}) \wedge (\mathsf{b} \leftrightarrow \Box\mathsf{b}) \to (\mathcal{C}_i[\mathsf{a}, \mathsf{b}] \leftrightarrow \mathcal{C}_i[\mathsf{a}, \mathsf{b}]) \tag{31}$$

and then conclude by substituting the prime formulas a, b. Formula (31), for each i, can be easily verified. A simple way to do it is to use a S4 theorem prover. We used LWB for this: see Sect. A.3 for more details. $\Box$

Lemma 35. $\vdash_{S4} e_i(\mathsf{p} \twoheadrightarrow \mathsf{q}) \leftrightarrow e_j(\mathsf{p} \twoheadrightarrow \mathsf{q})$ *iff* $i = j$.

Proof. This is an easy, albeit long, exercise. See Sect. A.3 for more details. $\Box$

Lemma 36. *The mappings $e_i(\bullet)$ are sound, i.e.$\vdash p \implies \vdash_{S4} e_i(p)$, for $i \in [1..4]$. The converse is false, in general, so they are not complete.*

Proof. We proceed by induction on the derivation of $\vdash p$.

If p is an instance of a PCL axiom, by Lemma 31 and 32 it is sufficient to consider the case of the axiom being applied to *prime* distinct formulas, since we can then substitute them to obtain p. Therefore, we check whether $\vdash e_i(q)$ for each prime instance of each PCL axiom q. This generates a finite number of specific formulas to verify in S4. Since this is rather long, we resort to LWB for this task. See Sect. A.3 for more details.

If instead $\vdash p$ was derived through a cut rule, say $\dfrac{a \quad a \to p}{p}$, then, by the inductive hypothesis, we have $\vdash_{S4} m(a)$ and $\vdash_{S4} m(a \to p) = \Box(m(a) \to m(p))$. Since $\Box q \to q$ is a tautology of S4, we can have $\vdash_{S4} m(a) \to m(p)$. By the cut rule of S4, we conclude $\vdash_{S4} m(p)$.

For the incompleteness result, it is enough to check that $\vdash_{S4} e_i(p_i)$ for some p_i such that $\nvdash p_i$. This was checked using the LWB theorem prover: we refer to Sect. A.3 for more details. $\Box$

6.3 Relations with Propositional Lax Logic

Propositional lax logic (PLL) [4] is an extension of IPC with a single modality $\bigcirc$, called *lax modality*, characterized by the following axioms:

$$\begin{array}{cc} p \to \bigcirc p & \bigcirc R \\ \bigcirc\bigcirc p \to \bigcirc p & \bigcirc M \\ (p \to q) \to (\bigcirc p \to \bigcirc q) & \bigcirc F \end{array}$$

These axioms appear to be relevant for contracts. Suppose we have a contract at hand, and reason about its implications. If we read $\bigcirc p$ as "q is ensured by the contract", then the axioms agree with our intuition. Axiom $\bigcirc R$ states that true propositions are always guaranteed. Axiom $\bigcirc M$ states that committing to a promise is actually a promise itself. Axiom $\bigcirc F$ is simple: if p is ensured, and implies q, clearly q must be ensured as well.

One might expect that, if we take a fixed formula c expresing the requirements of a contract, and we interpret $\bigcirc q$ as $c \twoheadrightarrow q$, then this should satisfy the axioms above. In other words, we expect $\bigcirc = c \twoheadrightarrow \bullet$ to be a lax modality (for any fixed c). However, it turns out that this is not the case in PCL.

Under the above definition of $\bigcirc$, Axiom $\bigcirc R$ holds, as one can derive it using PrePost and Zero. Axiom $\bigcirc F$ is also a special case of PrePost. However, axiom $\bigcirc M$ does not hold, because $\nvdash (c \twoheadrightarrow (c \twoheadrightarrow p)) \rightarrow (c \twoheadrightarrow p)$. The latter can be verified through the decision procedure of Lemma 50.

Quite interestingly, sometimes PLL is axiomatized in alternative equivalent way. Instead of including$\bigcirc F$ as an axiom, the related inference rule is used, together with another axiom:

$$\frac{p \rightarrow q}{\bigcirc p \rightarrow \bigcirc q} \qquad \bigcirc F$$

$$(\bigcirc q \wedge \bigcirc r) \rightarrow \bigcirc (q \wedge r) \qquad \bigcirc S$$

Axiom $\bigcirc S$ does not hold either PCL under our interpretation for $\bigcirc$. We actually already stated this when we discussed property (21).

We shall return on PLL when we discuss future work in Sect. 9.1.

7 Proof Theory: Sequent Calculus

In this section we provide an alternative formalization of PCL, through a sequent calculus à la Gentzen. Our sequents have the form $\Gamma \vdash p$, where Γ is a finite set of formulas. Below, we write Γ, p for $\Gamma \cup \{p\}$.

Most of the rules for the IPC fragment have been taken from [12], which features a rule set for IPC without structural rules. This fact turns out to be quite useful when reasoning about the rule system. Indeed, as in [12], we are able to establish cut-elimination by applying a reasonably simple structural induction; we refer to Sect. 8 for the detailed proof. In the IPC fragment of our rule system below, only rules $\neg R$ and $weakR$ diverge from [12]. This change was required to establish the subformula property (Lemma 49). Another minor difference from [12] arises from our Γ's being sets rather than multisets; this is however immaterial, because of the absence of structural rules, and the admissibility of contraction in [12].

Definition 37. The Gentzen-style rule system for PCL comprises the following rules.

IPC core rules

$$\frac{}{\Gamma, p \vdash p}id$$

$$\frac{\Gamma, p \wedge q, p \vdash r}{\Gamma, p \wedge q \vdash r}\wedge L1 \quad \frac{\Gamma, p \wedge q, q \vdash r}{\Gamma, p \wedge q \vdash r}\wedge L2 \quad \frac{\Gamma \vdash p \quad \Gamma \vdash q}{\Gamma \vdash p \wedge q}\wedge R$$

$$\frac{\Gamma, p \vee q, p \vdash r \quad \Gamma, p \vee q, q \vdash r}{\Gamma, p \vee q \vdash r}\vee L \quad \frac{\Gamma \vdash p}{\Gamma \vdash p \vee q}\vee R1 \quad \frac{\Gamma \vdash q}{\Gamma \vdash p \vee q}\vee R2$$

$$\frac{\Gamma, p \to q \vdash p \quad \Gamma, p \to q, q \vdash r}{\Gamma, p \to q \vdash r}\to L \quad \frac{\Gamma, p \vdash q}{\Gamma \vdash p \to q}\to R$$

$$\frac{\Gamma, \neg p \vdash p}{\Gamma, \neg p \vdash r}\neg L \quad \frac{\Gamma, p \vdash \bot}{\Gamma \vdash \neg p}\neg R$$

$$\frac{}{\Gamma, \bot \vdash p}\bot L \quad \frac{}{\Gamma \vdash \top}\top R \quad \frac{\Gamma \vdash \bot}{\Gamma \vdash p}weakR$$

$$\frac{\Gamma \vdash p \quad \Gamma, p \vdash q}{\Gamma \vdash q}cut$$

Contractual implication rules

$$\frac{\Gamma \vdash q}{\Gamma \vdash p \twoheadrightarrow q}Zero$$

$$\frac{\Gamma, p \twoheadrightarrow q, r \vdash p \quad \Gamma, p \twoheadrightarrow q, q \vdash r}{\Gamma, p \twoheadrightarrow q \vdash r}Fix$$

$$\frac{\Gamma, p \twoheadrightarrow q, a \vdash p \quad \Gamma, p \twoheadrightarrow q, q \vdash b}{\Gamma, p \twoheadrightarrow q \vdash a \twoheadrightarrow b}PrePost$$

Note that IPC rules have left and right rules for each IPC connective. Roughly, each connective has right rules for introduction, and left rules for elimination. Contractual implication instead has a different flavor. Rule $Zero$ is effectively an introduction rule, and has a similar function to the Hilbert axiom Zero. Similarly, rule Fix is an elimination rule; its function is related to the Hilbert axiom Fix. Indeed, these rules could be named $\twoheadrightarrow R$ and $\twoheadrightarrow L$, respectively. The left/right rule dualism however is broken by rule $PrePost$. Of course, its function is that of the Hilbert axiom PrePost. This rule behaves both as an introduction and an elimination, so is both a right and left rule, in a sense.

In the rest of this section we study the sequent rule system above. First, we prove it equivalent to our Hilbert axioms (Th. 40). Then we prove the redundancy of the $weakR$ rule, i.e. that $weakR$ is admissible in the proof system composed of all the other rules. More importantly, we also proce the *cut* rule redundant, i.e. we prove cut-elimination for our sequent rule system. As a consequence of cut-elimination, we are able to establish the consistency of the logic (Th. 47), the subformula property (Lemma 49), and the decidability of $\vdash$(Lemma 50). We also prove that the rules $Zero, PrePost, Fix$ are not redundant, as one might expect (Lemma 51). Finally we prove that PCL is a conservative extension of IPC (Lemma 52).

7.1 Equivalence of the Hilbert and Gentzen Systems

In this section, we establish the equivalence between the two different logical systems we introduced. We denote with $\vdash_H p$ the fact that p is provable from the Hilbert-style axioms of Def. 2. Similarly, by $\vdash_G p$ we denote that $\emptyset \vdash p$ is a derivable sequent from the Gentzen-style rules of Def. 37. We will then prove $\vdash_H = \vdash_G$.

Lemma 38. $\vdash_H p \implies \vdash_G p$

Proof. It is sufficient to show that $\vdash_G$ holds for all the Hilbert-style axioms, and that it is closed under modus ponens. The latter is trivially done by *cut* and $\to R$. For the former, we check the $\twoheadrightarrow$-related axioms, only, since the others are standard.

- *Zero*

$$\dfrac{\dfrac{}{\vdash \top}\top R}{\vdash \top \twoheadrightarrow \top} Zero$$

- *PrePost*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Delta, p' \vdash p'}id \quad \dfrac{}{\Delta, p', p \vdash p}id}{\Delta, p' \vdash p} \to L \quad \dfrac{\dfrac{}{\Delta, q \vdash q'}id \quad \dfrac{}{\Delta, q, q' \vdash q'}id}{\Delta, q \vdash q'} \to L}{\Delta = p' \to p, p \twoheadrightarrow q, q \to q' \vdash p' \twoheadrightarrow q'} PrePost}{p' \to p, p \twoheadrightarrow q \vdash (q \to q') \to (p' \twoheadrightarrow q')} \to R}{p' \to p \vdash (p \twoheadrightarrow q) \to (q \to q') \to (p' \twoheadrightarrow q')} \to R$$

$$\dfrac{p' \to p \vdash (p \twoheadrightarrow q) \to (q \to q') \to (p' \twoheadrightarrow q')}{\vdash (p' \to p) \to (p \twoheadrightarrow q) \to (q \to q') \to (p' \twoheadrightarrow q')} \to R$$

- *Fix*

$$\dfrac{\dfrac{\dfrac{}{p \twoheadrightarrow p, p \vdash p}id \quad \dfrac{}{p \twoheadrightarrow p, p \vdash p}id}{p \twoheadrightarrow p \vdash p} Fix}{\vdash (p \twoheadrightarrow p) \to p} \to R$$

□

Lemma 39. $\vdash_G p \implies \vdash_H p$

Proof. It is sufficient to prove the following statement for each rule:

$$\dfrac{\Gamma_0 \vdash_G p_0 \quad \cdots \quad \Gamma_n \vdash_G p_n}{\Gamma \vdash_G p} \implies \vdash_H \bigwedge_i [\bigwedge \Gamma_i \to p_i] \to \bigwedge \Gamma \to p$$

Then, the lemma follows by induction on the derivation of $\vdash_G p$. Most cases are standard, so we check only the $\twoheadrightarrow$-related rules.

- Rule *Zero*: $(\bigwedge \Gamma \to q) \to \bigwedge \Gamma \to (p \twoheadrightarrow q)$.
  Assume the hypotheses. By modus ponens, we get q, hence $\top \to q$. We have $\top \twoheadrightarrow \top$ by Zero. The formula $p \to \top$ trivially holds. We then apply PrePost to reach $p \twoheadrightarrow q$:

  $$(p \to \top) \to (\top \twoheadrightarrow \top) \to (\top \to q) \to (p \twoheadrightarrow q)$$

- Rule *PrePost*:$[(\bigwedge \Gamma \wedge (p \twoheadrightarrow q) \wedge a) \to p] \wedge [(\bigwedge \Gamma \wedge (p \twoheadrightarrow q) \wedge b) \to q] \to (\bigwedge \Gamma \wedge (p \twoheadrightarrow q)) \to (a \twoheadrightarrow b)$.
  Assume all the hypotheses. We easily get $a \to p, p \twoheadrightarrow q, q \to b$. By *PrePost*, we get $a \twoheadrightarrow b$.

- Rule *Fix*: $[(\bigwedge \Gamma \wedge (p \twoheadrightarrow q) \wedge r) \to p] \wedge [(\bigwedge \Gamma \wedge (p \twoheadrightarrow q) \wedge q) \to r] \to (\bigwedge \Gamma \wedge (p \twoheadrightarrow q)) \to r$.
  Assume all the hypotheses. We get $r \to p, q \to r$, hence $q \to p$. Using $q \to p$ (deduced), $p \twoheadrightarrow q$ (hypothesis), $q \to q$ (trivial), we apply *PrePost* and get $q \twoheadrightarrow q$. By *Fix*, q, hence r.

□

Theorem 40. $\vdash_G = \vdash_H$

Proof. Immediate from lemmas 38 and 39. □

7.2 Properties of the Gentzen System

A first basic result of our system is that the left-weakening of a sequent, i.e. augmenting the Γ, is strongly admissible. That is, whenever we have a derivation for a sequent, we can produce a derivation for the augmented sequent having the same height.

Lemma 41. *If* $\dfrac{D}{\Gamma \vdash p}$ *then* $\dfrac{D'}{\Gamma, \Gamma' \vdash p}$ *where D' has the same height of D.*

Proof. It is sufficient to augment each sequent in D with Γ'. It is straightforward to check that no rule is invalidated by this. □

Convention. The above lemma is very frequently used in our proofs. To avoid continuously referring to it, we adopt the following notation: when we have $\dfrac{D}{\Gamma \vdash p}$, we simply write $\dfrac{D+}{\Gamma, \Gamma' \vdash p}$ for the augmented derivation.

The next result is dual to Lemma 41. It states that the right-weakening of a sequent, i.e. replacing a $\bot$ on the right of the turnstile $\vdash$ with some other formula , can be performed without using the *weakR* rule. In other words, rule *weakR* is redundant in our system. To prove this, we first introduce an auxiliary lemma.

Lemma 42. *If* $\dfrac{D}{\Gamma \vdash \bot}$ *where D is a weakR-free derivation, then we also have $\Gamma \vdash p$ with a weakR-free derivation, for any p.*

Proof. By induction on the height of the derivation D. The last step of D must be one of id, cut, Fix or a left rule. If a left rule or a cut has been used, the thesis is either trivial ($\neg L, \bot L$) or immediately follows by the induction hypothesis. If

id has been used, then $\bot \in \Gamma$, and $\bot L$ suffices. If Fix has been used, we rewrite the derivation in the following way:

$$\dfrac{\dfrac{D0}{\Gamma, a \twoheadrightarrow b, \bot \vdash a} \quad \dfrac{D1}{\Gamma, a \twoheadrightarrow b, b \vdash \bot}}{\Gamma, a \twoheadrightarrow b \vdash \bot}\,\mathit{Fix} \quad \Longrightarrow$$

$$\dfrac{\dfrac{\dfrac{}{\Gamma, a \twoheadrightarrow b, p, a \vdash a}\,\mathit{id} \quad \dfrac{D1''+}{\Gamma, a \twoheadrightarrow b, p, b \vdash a}}{\Gamma, a \twoheadrightarrow b, p \vdash a}\,\mathit{Fix} \quad \dfrac{D1'}{\Gamma, a \twoheadrightarrow b, b \vdash p}}{\Gamma, a \twoheadrightarrow b \vdash p}\,\mathit{Fix}$$

where $D1'$ and $D1''$ are obtained from the induction hypothesis on $D1$, and the formulas p, a respectively. □

Note that the transformation above does not introduce new *cut*s in the derivation. So, once the cut has been eliminated, the same procedure can eliminate $weakR$, too.

Lemma 43. *The $weakR$ rule is redundant.*

Proof. It is sufficient to iterate Lemma 42. □

Another fundamental result enjoyed by our Gentzen-style rules, is the redundancy of the *cut* rule. This is a classic cut-elimination result, or *Haupsatz*, which is the basis for many of the results in this section.

Theorem 44. ***(Cut-elimination)*** *The cut rule is redundant.*

Proof. See Sect. 8 for the detailed proof. □

It is possible to remove both the rules *cut* and $weakR$ from our system without affecting the generated $\vdash$ relation.

Theorem 45. *The rule set $\{cut, weakR\}$ is redundant.*

Proof. This is *not* an immediate corollary of the previous results, since removing a *cut* from a derivation could force us to include $weakR$ in the new derivation, and, viceversa, removing a $weakR$ could force us to introduce a *cut*. However, by inspecting the proof for Lemma 43, we can see that the $weakR$-elimination procedure does not introduce new *cut*s. So, given an arbirtrary derivation D for a sequent, by Th. 44 we also have a cut-free derivation D' for the same sequent. Then, we can apply the procedure of Lemma 43 to conclude. □

In our logic, as for IPC, every negation-free theory Γ is consistent.

Lemma 46. *Let Γ be free from $\neg, \bot$. Then $\Gamma \nvdash \bot$.*

Proof. By contradiction, assume $\dfrac{D}{\Gamma \vdash \bot}$ to be a cut-free derivation. We proceed by induction on the derivation D, and then by case analysis. The last step of D can not be $id, \neg L, \bot L$, otherwise Γ is not free from $\neg, \bot$. If the last step is another left rule, we have $\Gamma' \vdash \bot$ as a premise for some Γ' free from $\neg, \bot$ so the inductive hypothesis suffices. The same applies to Fix. No right rule can introduce $\bot$, so the last step is not a right rule. The same applies to $Zero, PrePost$. No other rule exists. □

Theorem 47. ***(Consistency)*** *The logic is consistent, i.e.* $\emptyset \nvdash \bot$.

Proof. Direct consequence of Lemma 46. □

Cut-free derivations enjoy the *subformula* property, stating that all the formulas occurring such a derivation for a sequent appear as subformulas in the sequent as well. Equivalently, it states that a cut-free derivation of a sequent can only involve subformulas of that sequent. As a single exception, the derivation might mention $\bot$ while the sequent does not, because of the $weakR$ rule.

Definition 48. The subformulas $sub(p)$ of a formula p are inductively defined as follows

$$
\begin{aligned}
&sub(\mathsf{p}) = \{\mathsf{p}\}\\
&sub(\bot) = \{\bot\}\\
&sub(\top) = \{\top\}\\
&sub(\neg p) = \{\neg p\} \cup sub(p)\\
&sub(p \vee q) = \{p \vee q\} \cup sub(p) \cup sub(q)\\
&sub(p \wedge q) = \{p \wedge q\} \cup sub(p) \cup sub(q)\\
&sub(p \rightarrow q) = \{p \rightarrow q\} \cup sub(p) \cup sub(q)\\
&sub(p \twoheadrightarrow q) = \{p \twoheadrightarrow q\} \cup sub(p) \cup sub(q)
\end{aligned}
$$

The subformulas of a set of formulas Γ is $sub(\Gamma) = \bigcup_{p \in \Gamma} sub(p)$.

Lemma 49. ***(Subformula Property)*** *If* $\dfrac{D}{\Gamma \vdash p}$ *and* D *is cut-free, then the formulas occurring in* D *belong to* $sub(\Gamma, p, \bot)$.

Proof. By a simple induction on D. The property is preserved by every rule. □

Note that a $\{cut, weakR\}$-free derivation would instead have a more tight $sub(\Gamma, p)$ bound.

We can now establish decidability for PCL.

Lemma 50. ***(Decidability)*** $\Gamma \vdash p$ *is decidable.*

Proof. We have $\Gamma \vdash p$ iff it can be derived without cuts. We can decide the latter by searching for a shortest derivation bottom-up, exploring the whole proof space non-deterministically. By the subformula property, cut-free proofs can only contain sequents having formulas in $sub(\Gamma, p, \bot)$. This is a finite set of sequents: let k be its cardinality. The depth of the search in the proof space can be limited to k: if there is a taller derivation, it has a sequent occurring twice in some path, so the proof can be made shorter. This ensures the termination of the algorithm. □

We have implemented the above naïve decision procedure for PCL, developing a prototype tool, which we used for experimenting with our logic. Since the proof space is huge, the tool was very helpful in establishing the negative results for our logic, i.e. $\nvdash p$ from some given p. We give more information about it in Sect. A.2.

We can now prove the non redundancy of the $\twoheadrightarrow$-related rules.

Lemma 51. *None of the Zero, PrePost, Fix Gentzen rules is redundant.*

Proof. First, we carefully examine the proof of the cut-elimination theorem, and check that the same proof actually establish cut-elimination event in the rule system whithout any of the rules *Zero*, *PrePost*, *Fix*. This is because the cut-elimination procedure never introduces a new application of, say, a *Fix* rule unless *Fix* was already present in the original derivation. Similarly, we can restate the subformula property in these restricted rule systems, as well as decidability. We therefore have decision procedures for both the full rule system and each restriction of it, so we can check that indeed some formula is provable in the full system, but not in the restriction. As it might be expected, it turns out that to prove the Hilbert axioms *Zero*, *PrePost*, *Fix* in the Gentzen system, the related rule is necessary. This was checked by a simple modification of our tool, described in Sect. A.2. □

As a nice result of the subformula property, we get that PCL is a conservative extesion of IPC.

Lemma 52. *PCL is a conservative extension of IPC, that is* $\vdash_{IPC} p \iff \vdash p$ *for all IPC formulas* p.

Proof. The $\Rightarrow$ part is Lemma 19. For the $\Leftarrow$ part, if $\vdash p$, by Lemma 38 we have $\vdash_G p$. By cut-elimination and the subformula property, we have $\dfrac{D}{\vdash p}$ where D make no use of rules Zero, Fix, and PrePost. Since all the other rules are included in the Gentzen system of IPC, we can state $\vdash_{IPC} p$. □

8 Cut-elimination

In this section we prove cut elimination for the Gentzen system of Def. 37. In order to do this, we borrow a fairly simple structural induction technique from [12].[2] The whole technique can be described as a recursive algorithm, transforming a generic derivation into a cut-free one. The algorithm is made of two recursive routines: a core routine (CUT-REDUCE) and a driver (CUT-ELIM). The core routine deals with the special case of a derivation ending with a *cut* between two *cut-free* subderivations. We name derivations of this special form *reducible* derivations. When provided with a reducible derivation, CUT-REDUCE produces a cut-free derivation for the same sequent. Exploiting CUT-REDUCE, the driver CUT-ELIM handles the general case. The actual procedure is shown in Alg. 1.

[2]Here, "structural" means that it proceeds inductively on the structure of a derivation in the Gentzen system.

Algorithm 1 Main driver for cut-elimination.

$$\textsc{cut-elim}\left(\frac{\dfrac{D_0}{\Gamma_0 \vdash p_0} \quad \dfrac{D_1}{\Gamma_1 \vdash p_1}}{\Gamma \vdash p}\mathit{cut}\right) = \textsc{cut-reduce}\left(\frac{\dfrac{D'_0}{\Gamma_0 \vdash p_0} \quad \dfrac{D'_1}{\Gamma_1 \vdash p_1}}{\Gamma \vdash p}\mathit{cut}\right)$$

$$\textsc{cut-elim}\left(\frac{\cdots \dfrac{D_i}{\Gamma_i \vdash p_i} \cdots}{\Gamma \vdash p} r\right) = \frac{\cdots \dfrac{D'_i}{\Gamma_i \vdash p_i} \cdots}{\Gamma \vdash p} r \qquad (r \neq \mathit{cut})$$

where $\dfrac{D'_i}{\Gamma_i \vdash p_i} = \textsc{cut-elim}\left(\dfrac{D_i}{\Gamma_i \vdash p_i}\right)$

The CUT-ELIM driver takes as input a derivation D. First, it applies itself recursively on the derivations of the premises D_i so converting them to the cut-free ones D'_i. Then, if the last rule r used in D is not a *cut*, we can apply the same rule to D'_i to construct a cut-free derivation. Otherwise, r is a *cut*, and we are in the special case handled by CUT-REDUCE, so we just invoke it. The CUT-ELIM driver always terminates since each recursive call is made on a subderivation.

In the rest of this section, we define the core routine CUT-REDUCE. This routine makes use of a sophisticated recursion scheme. When invoked as

$$\textsc{cut-reduce}\left(\frac{\dfrac{D_0}{\Gamma \vdash p} \quad \dfrac{D_1}{\Gamma, p \vdash q}}{\Gamma \vdash q}\mathit{cut}\right) \text{ with cut-free } D_0, D_1$$

the routine makes several recursive calls, all of them for reducible derivations. Assume a recursive call is made on a cut having p' as the cut formula, and D'_0, D'_1 as the subderivations. Then, one of the following conditions holds:

Definition 53. Constraints on the recursive calls to CUT-REDUCE:[3]

1. p' is a proper subformula of p, or
2. $p' = p$ and $h(D'_0) < h(D_0)$, or
3. $p' = p$, $h(D'_0) \leq h(D_0)$, and $h(D'_1) < h(D_1)$.

We can state the conditions above as follows: the triple $(p', h(D'_0), h(D'_1))$ is lexicographically smaller than $(p, h(D_0), h(D_1))$. It is easy to see that this ordering for triples is a well-founded ordering relation, and therefore the recursion must eventually terminate.

We now proceed to define the CUT-REDUCE routine. This is done by examining the last rules used in D_0 and D_1, and covering all the possible cases. To simplify the presentation, we adopt a compact notation, writing $\Longrightarrow$ for our reduction, as seen in Alg. 2. The rest of this section will precisely define the $\Longrightarrow$ relation.

[3] We let conditions 2 and 3 to overlap, since we use $\leq$ instead of $=$. This is done to follow our actual reduction procedure more closely, as we will see.

Algorithm 2 Core routine for cut-elimination.

$$\text{CUT-REDUCE}\left(\dfrac{\dfrac{D_0}{\Gamma \vdash p} \quad \dfrac{D_1}{\Gamma, p \vdash q}}{\Gamma \vdash q}\mathit{cut}\right) = \bar{D}$$

where $\dfrac{\dfrac{D_0}{\Gamma \vdash p} \quad \dfrac{D_1}{\Gamma, p \vdash q}}{\Gamma \vdash q}\mathit{cut} \implies D'$

and $\bar{D}$ is obtained by recursively applying CUT-REDUCE to all the *cut*s in D'.

When we write $D \Longrightarrow D'$, D is a reducible derivation, and D' is the result of the reduction. When we use *cut*s in D', they are to be interpreted as recursive calls to the CUT-REDUCE routine. Of course, we shall take care these cuts agree with the well-founded ordering discussed above. We document this in the derivation D', by writing cut_p when we are in case 1 of the enumeration above, cut_0 when we are in case 2, or cut_1 when we are in case 3.

8.1 Summary of Cases

To classify all possible cases, we first introduce some terminology. Each Gentzen rule is related to some logical connective. A rule for a generic connective $\odot$ has as its principal formulas the formulas occurring in that rule that involve $\odot$. Rules $id, weakR$ and *cut* have no principal formulas. Both "left" and "right" rules in the IPC fragment have exactly one principal formula. Rules *Zero* and *Fix* have one principal formula as well. When a principal formula occurs in the left (resp. right) hand side of the turnstile $\vdash$we call it a left (resp. right) principal formula. Rule *PrePost* has two principal formulas, named left principal and right principal formulas.

In order to reduce

$$\dfrac{\dfrac{D_0}{\Gamma \vdash a} \quad \dfrac{D_1}{\Gamma, a \vdash b}}{\Gamma \vdash b}\mathit{cut}$$

we proceed by case analysis on the last rule used in D_0, D_1. The derivation D_0can end with a *Zero* rule, a *PrePost* rule, a *Fix* rule, or an IPC rule. Similarly for D_1. These 4^2cases are further split, according to whether

- the cut formula is the left principal formula of D_0 and the right principal formula of D_1 (the *essential* case),
- is *not* the right principal formula of D_0 (the *left commutation* case), or
- is *not* the left principal formula of D_1 (the *right commutation* case).

This classification is rather standard in cut-elimination results, and is used in [12] as well. Note that the two commutation cases can overlap when the cut formula is not right/left principal in both D_0, D_1, respectively. In Table 1, we cover all the possible cases, and group them whenever the handling is similar.

We now provide a reading key for Table 1. The first row describes the case where D_0 ends with *Zero*. In this case, the cut formula is the right principal formula of D_0, and so there is no left commutation case, denoted by $\nexists$(l). The first

$D_0 \setminus D_1$	Zero	PrePost	Fix	IPC
Zero $\nexists$ (l)	$\nexists$ (e) */Zero (r)	Zero/PrePost (e) */PrePost (r)	Zero/Fix (e) */Fix (r)	$\nexists$ (e) standard (r)
PrePost $\nexists$ (l)	$\nexists$ (e) */Zero (r)	PrePost/PrePost (e) */PrePost (r)	PrePost/Fix (e) */Fix (r)	$\nexists$ (e) standard (r)
Fix $\nexists$ (e)	[Fix/* (l)] */Zero (r)	Fix/* (l) [*/PrePost (r)]	Fix/* (l) [*/Fix (r)]	Fix/* (l) [standard (r)]
IPC	$\nexists$ (e) [standard (l)] */Zero (r)	$\nexists$ (e) standard (l) [*/PrePost (r)]	$\nexists$ (e) standard (l) [*/Fix (r)]	standard (e,l,r)

(e) essential, (l) left commutation, (r) right commutation, [subsumed]

Table 1: Summary of cases for cut-elimination

cell is for the case where D_1 ends with $Zero$ as well: this is a right commutation case which is handled in the same way as $PrePost/Zero$, $Fix/Zero$, etc. so we will have a single case $*/Zero$ case for the whole column. The $Zero/PrePost$ case can be an essential case or a right commutation depending on whether the cut formula is the left principal formula of $PrePost$, so we split in two subcases. The right commutation case $*/PrePost$ is also reused in other points in the same column. The same applies to $Zero/Fix$. The $Zero/IPC$ case can not be an essential case, since $\twoheadrightarrow$ never occurs in an IPC rule; so this is a right commutation.

The second row describes the case where D_0 ends with $PrePost$, and is similar to the first row.

The third row describes the case where D_0 ends with Fix. Since Fix has no right principal formula, there is no essential case in this row, denoted by $\nexists$(e). The case $Fix/Zero$ is both a left and right commutation case, which we arbitrarily chose to handle as a right commutation case. The remaining cases *might* be right commutations, but are surely left commutations $Fix/*$, so we handle them in that way.

The fourth row describes the case where D_0 ends with an IPC rule. The case $IPC/Zero$ *might* be a left commutation (depending on the actual IPC rule), but is surely a right commutation as well, so we handle it in that way. The case $IPC/PrePost$ can not be an essential case, since $\twoheadrightarrow$ occurs in no IPC rule; it *might* be a right commutation, but is surely a left commutation (no IPC rules involves $\twoheadrightarrow$), so we handle it in that way. The case IPC/Fix is similar. The case IPC/IPC can be either essential, a left commutation, or a right commutation.

We invite the reader to check that Table 1 indeed enumerates all the possible cases, which can therefore be grouped as follows:

- essential: $Zero/PrePost$ (e), $Zero/Fix$ (e), $PrePost/PrePost$ (e), $PrePost/Fix$ (e), standard (e)
- left commutations: $Fix/*$ (l), standard (l)
- right commutations: $*/Zero$ (r), $*/PrePost$ (r), $*/Fix$ (r), standard (r)

Most cases of the "standard" group are well-known cases for IPC, and are covered in [12], Appendix 1. This includes, for instance, the essential case $\wedge R/\wedge L1$, the left commutation $\wedge L1/*$, and the right commutation $*/\wedge R$. Handling these cases here would essentially amount to copying the whole Appendix 1 of [12] here, and perform some minor notation change, only. Since this would provide no actual contribution, we will refer to [12] when these cases arise, and omit them. For completeness, we document in Sect. A.1 how to rephrase [12] in our notation. Finally, recall that in Def. 37 we diverge from [12] in a few IPC rules, namely $\neg R$ and $weakR$. Of course, these rules are involved in standard cases which are not covered in [12], so we shall provide a reduction for these cases. The case $\neg R/*$ can not be a left commutation, but it can be essential ($\neg R/\neg L$); the case $*/\neg R$ is instead a right commutation. The case $*/weakR$ is a right commutation, while $weakR/*$ is a left commutation.

We now proceed by handling all the cases mentioned above. We sometimes write $\Gamma(p)$ instead of Γ to stress that $p \in \Gamma$.

8.2 The Essential Cases

In these cases the cut formula is (right/left) principal in both the premises of the cut.

- Case $Zero/PrePost$

$$\dfrac{\dfrac{\dfrac{D0}{\Gamma \vdash q}}{\Gamma \vdash p \twoheadrightarrow q}\,\mathit{Zero} \qquad \dfrac{\dfrac{D1}{\Gamma, p \twoheadrightarrow q, a \vdash p} \quad \dfrac{D2}{\Gamma, p \twoheadrightarrow q, q \vdash b}}{\Gamma, p \twoheadrightarrow q \vdash a \twoheadrightarrow b}\,\mathit{PrePost}}{\Gamma \vdash a \twoheadrightarrow b}\,\mathit{cut} \quad \Longrightarrow$$

$$\dfrac{\dfrac{\dfrac{D0}{\Gamma \vdash q} \qquad \dfrac{\dfrac{\dfrac{}{\Gamma, q \vdash q}\,\mathit{id}}{\Gamma, q \vdash p \twoheadrightarrow q}\,\mathit{Zero} \quad \dfrac{D2}{\Gamma, q, p \twoheadrightarrow q \vdash b}}{\Gamma, q \vdash b}\,\mathit{cut}_1}{\Gamma \vdash b}\,\mathit{cut}_p}{\Gamma \vdash a \twoheadrightarrow b}\,\mathit{Zero}$$

- Case $Zero/Fix$

$$\dfrac{\dfrac{\dfrac{D0}{\Gamma \vdash q}}{\Gamma \vdash p \twoheadrightarrow q}\,\mathit{Zero} \qquad \dfrac{\dfrac{D1}{\Gamma, p \twoheadrightarrow q, a \vdash p} \quad \dfrac{D2}{\Gamma, p \twoheadrightarrow q, q \vdash a}}{\Gamma, p \twoheadrightarrow q \vdash a}\,\mathit{Fix}}{\Gamma \vdash a}\,\mathit{cut} \quad \Longrightarrow$$

$$\dfrac{\dfrac{D0}{\Gamma \vdash q} \qquad \dfrac{\dfrac{\dfrac{}{\Gamma, q \vdash q}\,\mathit{id}}{\Gamma, q \vdash p \twoheadrightarrow q}\,\mathit{Zero} \quad \dfrac{D2}{\Gamma, q, p \twoheadrightarrow q \vdash a}}{\Gamma, q \vdash a}\,\mathit{cut}_1}{\Gamma \vdash a}\,\mathit{cut}_p$$

- Case $PrePost/PrePost$. We can assume $(p \twoheadrightarrow q) \in \Gamma$.

$$\dfrac{\dfrac{\dfrac{D0}{\Gamma, a \vdash p} \quad \dfrac{D1}{\Gamma, q \vdash b}}{\Gamma \vdash a \twoheadrightarrow b}\,PrePost \qquad \dfrac{\dfrac{D2}{\Gamma, a \twoheadrightarrow b, x \vdash a} \quad \dfrac{D3}{\Gamma, a \twoheadrightarrow b, b \vdash y}}{\Gamma, a \twoheadrightarrow b \vdash x \twoheadrightarrow y}\,PrePost}{\Gamma(p \twoheadrightarrow q) \vdash x \twoheadrightarrow y}\,cut \quad \Longrightarrow$$

$$\dfrac{\dfrac{\hat{D}_0}{\Gamma, x \vdash p} \quad \dfrac{\hat{D}_1}{\Gamma, q \vdash y}}{\Gamma(p \twoheadrightarrow q) \vdash r}\,PrePost$$

$$\hat{D}_0 = \dfrac{\dfrac{\dfrac{\dfrac{D0+}{\Gamma, x, a \vdash p} \quad \dfrac{D1+}{\Gamma, x, q \vdash b}}{\Gamma, x \vdash a \twoheadrightarrow b}\,PrePost \qquad \dfrac{D2}{\Gamma, x, a \twoheadrightarrow b \vdash a}}{\Gamma, x \vdash a}\,cut_1 \qquad \dfrac{D0+}{\Gamma, x, a \vdash p}}{\Gamma, x \vdash p}\,cut_p$$

$$\hat{D}_1 = \dfrac{\dfrac{D1}{\Gamma, q \vdash b} \qquad \dfrac{\dfrac{\dfrac{D0+}{\Gamma, q, b, a \vdash p} \quad \dfrac{D1+}{\Gamma, q, b, q \vdash b}}{\Gamma, q, b \vdash a \twoheadrightarrow b}\,PrePost \qquad \dfrac{D3+}{\Gamma, q, b, a \twoheadrightarrow b \vdash y}}{\Gamma, q, b \vdash y}\,cut_1}{\Gamma, q \vdash y}\,cut_p$$

- Case $PrePost/Fix$. We can assume $(p \twoheadrightarrow q) \in \Gamma$.

$$\dfrac{\dfrac{\dfrac{D0}{\Gamma, a \vdash p} \quad \dfrac{D1}{\Gamma, q \vdash b}}{\Gamma \vdash a \twoheadrightarrow b}\,PrePost \qquad \dfrac{\dfrac{D2}{\Gamma, a \twoheadrightarrow b, r \vdash a} \quad \dfrac{D3}{\Gamma, a \twoheadrightarrow b, b \vdash r}}{\Gamma, a \twoheadrightarrow b \vdash r}\,Fix}{\Gamma(p \twoheadrightarrow q) \vdash r}\,cut \quad \Longrightarrow$$

$$\dfrac{\dfrac{\hat{D}_0}{\Gamma, r \vdash p} \quad \dfrac{\hat{D}_1}{\Gamma, q \vdash r}}{\Gamma \vdash r}\,Fix$$

$$\hat{D}_0 = \dfrac{\dfrac{\dfrac{\dfrac{D0+}{\Gamma, r, a \vdash p} \quad \dfrac{D1+}{\Gamma, r, q \vdash b}}{\Gamma, r \vdash a \twoheadrightarrow b}\,PrePost \qquad \dfrac{D2}{\Gamma, r, a \twoheadrightarrow b \vdash a}}{\Gamma, r \vdash a}\,cut_1 \qquad \dfrac{D0+}{\Gamma, r, a \vdash p}}{\Gamma, r \vdash p}\,cut_p$$

$$\hat{D}_1 = \dfrac{\dfrac{D1}{\Gamma, q \vdash b} \qquad \dfrac{\dfrac{\dfrac{D0+}{\Gamma, q, b, a \vdash p} \quad \dfrac{D1+}{\Gamma, q, b, q \vdash b}}{\Gamma, q, b \vdash a \twoheadrightarrow b}\,PrePost \qquad \dfrac{D3+}{\Gamma, q, b, a \twoheadrightarrow b \vdash r}}{\Gamma, q, b \vdash r}\,cut_1}{\Gamma, q \vdash r}\,cut_p$$

- standard. As anticipated, we refer to [12] here, but for the case $\neg R/\neg L$, shown below.

- Case $\neg R/\neg L$

$$\dfrac{\dfrac{\dfrac{D0}{\Gamma, p \vdash \bot}}{\Gamma \vdash \neg p}\neg R \quad \dfrac{\dfrac{D1}{\Gamma, \neg p \vdash p}}{\Gamma, \neg p \vdash q}\neg L}{\Gamma \vdash q}\mathit{cut} \quad \Longrightarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{D0}{\Gamma, p \vdash \bot}}{\Gamma \vdash \neg p}\neg R \quad \dfrac{D1}{\Gamma, \neg p \vdash p}}{\Gamma \vdash p}\mathit{cut}_1 \quad \dfrac{D0}{\Gamma, p \vdash \bot}}{\Gamma \vdash \bot}\mathit{cut}_p}{\Gamma \vdash q}\mathit{weakR}$$

8.3 The Left Commutation Cases

In these cases the cut formula is not a right principal formula in the left premise of the cut.

- Case $Fix/*$.We can assume $(p \twoheadrightarrow q) \in \Gamma$.

$$\dfrac{\dfrac{\dfrac{D0}{\Gamma, a \vdash p} \quad \dfrac{D1}{\Gamma, q \vdash a}}{\Gamma \vdash a}\mathit{Fix} \quad \dfrac{D2}{\Gamma, a \vdash b}*}{\Gamma(p \twoheadrightarrow q) \vdash b}\mathit{cut} \quad \Longrightarrow$$

$$\dfrac{\dfrac{\dfrac{}{\Gamma, b, p \vdash p}\mathit{id} \quad \dfrac{\dfrac{D1+}{\Gamma, b, q \vdash a} \quad \dfrac{D0+}{\Gamma, b, q, a \vdash p}}{\Gamma, b, q \vdash p}\mathit{cut}_0}{\Gamma, b \vdash p}\mathit{Fix} \quad \dfrac{\dfrac{D1}{\Gamma, q \vdash a} \quad \dfrac{D2+}{\Gamma, q, a \vdash b}}{\Gamma, q \vdash b}\mathit{cut}_0}{\Gamma \vdash b}\mathit{Fix}$$

- standard. As anticipated, we refer to [12] here, but for the case $weakR/*$, shown below.

- Case $weakR/*$

$$\dfrac{\dfrac{\dfrac{D0}{\Gamma \vdash \bot}}{\Gamma \vdash p}\mathit{weakR} \quad \dfrac{D1}{\Gamma, p \vdash q}*}{\Gamma \vdash q}\mathit{cut} \quad \Longrightarrow \quad \dfrac{\dfrac{D0}{\Gamma \vdash \bot}}{\Gamma \vdash q}\mathit{weakR}$$

8.4 The Right Commutation Cases

In these cases the cut formula is not a left principal formula in the right premise of the cut.

- Case $*/Zero$

$$\dfrac{\dfrac{D0}{\Gamma \vdash a}* \quad \dfrac{\dfrac{D1}{\Gamma, a \vdash q}}{\Gamma, a \vdash p \twoheadrightarrow q}\mathit{Zero}}{\Gamma \vdash p \twoheadrightarrow q}\mathit{cut} \quad \Longrightarrow$$

$$\dfrac{\dfrac{\dfrac{D0}{\Gamma \vdash a} \quad \dfrac{D1}{\Gamma, a \vdash q}}{\Gamma \vdash q}\mathit{cut}_1}{\Gamma \vdash p \twoheadrightarrow q}\mathit{Zero}$$

- Case $*/PrePost$. We can assume $(p \twoheadrightarrow q) \in \Gamma$.

$$\dfrac{\dfrac{D0}{\Gamma \vdash a}\ * \qquad \dfrac{\dfrac{D1}{\Gamma, a, x \vdash p} \quad \dfrac{D2}{\Gamma, a, q \vdash y}}{\Gamma, a \vdash x \twoheadrightarrow y}\, PrePost}{\Gamma(p \twoheadrightarrow q) \vdash x \twoheadrightarrow y}\, cut \quad \Longrightarrow$$

$$\dfrac{\dfrac{\dfrac{D0+}{\Gamma, x \vdash a} \quad \dfrac{D1}{\Gamma, x, a \vdash p}}{\Gamma, x \vdash p}\, cut_1 \qquad \dfrac{\dfrac{D0+}{\Gamma, q \vdash a} \quad \dfrac{D2}{\Gamma, q, a \vdash y}}{\Gamma, q \vdash y}\, cut_1}{\Gamma \vdash x \twoheadrightarrow y}\, PrePost$$

- Case $*/Fix$. We can assume $(p \twoheadrightarrow q) \in \Gamma$.

$$\dfrac{\dfrac{D0}{\Gamma \vdash a}\ * \qquad \dfrac{\dfrac{D1}{\Gamma, a, r \vdash p} \quad \dfrac{D2}{\Gamma, a, q \vdash r}}{\Gamma, a \vdash r}\, Fix}{\Gamma(p \twoheadrightarrow q) \vdash r}\, cut \quad \Longrightarrow$$

$$\dfrac{\dfrac{\dfrac{D0+}{\Gamma, r \vdash a} \quad \dfrac{D1}{\Gamma, r, a \vdash p}}{\Gamma, r \vdash p}\, cut_1 \qquad \dfrac{\dfrac{D0+}{\Gamma, q \vdash a} \quad \dfrac{D2}{\Gamma, q, a \vdash r}}{\Gamma, q \vdash r}\, cut_1}{\Gamma \vdash r}\, Fix$$

- standard. As anticipated, we refer to [12] here, but for the cases $*/weakR$ and $*/\neg R$, shown below.

- Case $*/weakR$

$$\dfrac{\dfrac{D0}{\Gamma \vdash p}\ * \qquad \dfrac{\dfrac{D1}{\Gamma, p \vdash \bot}}{\Gamma, p \vdash q}\, weakR}{\Gamma \vdash q}\, cut \quad \Longrightarrow \quad \dfrac{\dfrac{\dfrac{D0}{\Gamma \vdash p} \quad \dfrac{D1}{\Gamma, p \vdash \bot}}{\Gamma \vdash \bot}\, cut_1}{\Gamma \vdash q}\, weakR$$

- Case $*/\neg R$

$$\dfrac{\dfrac{D0}{\Gamma \vdash p} \qquad \dfrac{\dfrac{D1}{\Gamma, p, q \vdash \bot}}{\Gamma, p \vdash \neg q}\, \neg R}{\Gamma \vdash \neg q}\, cut \quad \Longrightarrow \quad \dfrac{\dfrac{\dfrac{D0+}{\Gamma, q \vdash p} \quad \dfrac{D1}{\Gamma, q, p \vdash \bot}}{\Gamma, q \vdash \bot}\, cut_1}{\Gamma \vdash \neg q}\, \neg R$$

9 Conclusions

We have investigated the notion of contract from a logical perspective. To do that, we have extended intuitionistic propositional logic with a new connective, that models contractual implication. We have provided the new connective with an Hilbert-style axiomatisation, which have allowed us to deduce, for instance, that n contracting parties, each requiring a promise from the other $n-1$ parties in order to make its own promise, eventually reach an agreement. Further interesting properties and application scenarios for our logic have been explored, in particular in Sect. 3 and 5.

The main result about our logic is its decidability. To prove that, we have devised a Gentzen-style sequent calculus for the logic, which is equivalent to the Hilbert-style axiomatisation. Decidability then follows from the subformula property, which is enjoyed by our Gentzen rules, and by a cut elimination theorem, which we have proved in full details in this paper. As a further support to our logic, we have implemented a proof search algorithm, which decides if any given formula is a tautology or not.

9.1 Future Work

While designing the logic for contracts proposed in this paper, our main concerns were to give a minimal set of rules for capturing the notion of contract agreement, and at the same time preserving the decidability of IPC.

We expect that many useful features can be added to our logic, to make it suitable for modelling complex scenarios, which are not directly manageable with the basic primitives presented here. Of course, preserving the decidability of the logic will be a major concern, while considering these extensions. We discuss below some of the additional features which we think to be more useful in the future developments of our logic.

First order features. A significant extension to our logic would be that of extending it with predicates and quantifiers. This will allow us to model more accurately several scenarios, where a party issues a "generic" contract that can be matched by many parties. While this first order extension shall force us to drop the decidability result, we expect to find interesting decidable fragments of the logic, through which modelling many relevant situations.

For instance, consider an e-commerce scenario, where a seller promises to ship the purchased item to a given address, provided that the customer will pay for that item. Aiming at generality, we make the seller contract parametric with respect to the item, customer and address. This could be modelled using a universal quantification over these three formal parameters:

$$\begin{aligned} Seller = \forall item, customer, address :& \\ \mathsf{pay}(item, customer, address) &\twoheadrightarrow \mathsf{ship}(item, address) \end{aligned} \tag{32}$$

Now, assume that a customer (say, Bob) promises that he will pay for a drill, provided that the seller will ship the item to his address. This is modelled by the following contract issued by Bob, where the actual parameters remark that the actual payment is made by Bob, and that the destination address is Bob's.

$$Bob = \mathsf{ship}(\mathsf{drill}, \mathsf{bobAddress}) \twoheadrightarrow \mathsf{pay}(\mathsf{drill}, \mathsf{Bob}, \mathsf{bobAddress})$$

Joining the two two contracts above will yield the intended agreement, that is:

$$Seller \land Bob \;\rightarrow\; \mathsf{pay}(\mathsf{drill}, \mathsf{Bob}, \mathsf{bobAddress}) \land \mathsf{ship}(\mathsf{drill}, \mathsf{bobAddress})$$

Principals. As a first extension to the logic, we will consider allowing formulae with a *says* modality, similarly to [5]. For instance, this will enable us to write:

$$Alice\ says\ (\mathsf{b} \twoheadrightarrow \mathsf{a})$$

to represent the fact that Alice has issued a contract, where she promises to lend her airplane, provided that she borrows the bike (note that the binding between principals and contracts is represented *outside* the logic, in the examples shown so far).

Contract agreements shall then come in a richer flavour, because of the binding between the contracting parties and their inferred duties, which is now revealed. Back to our first example of Sect. 2, one could expect an handshaking of the following form:

$$\mathit{Alice\ says}\ (\mathsf{b} \twoheadrightarrow \mathsf{a}) \ \wedge\ \mathit{Bob\ says}\ (\mathsf{a} \twoheadrightarrow \mathsf{b}) \quad \rightarrow \quad \mathit{Alice\ says}\ \mathsf{a} \ \wedge\ \mathit{Bob\ says}\ \mathsf{b}$$

from which it is clear that the duty of Alice is that of lending her airplane, and the duty of Bob is that of lending his bike. This additional information can be exploited by a third party (a sort of "automated" judge) which has to investigate the responsibilities of various parties, in the unfortunate case that a contract is not respected. For instance, if our automated judge is given the evidence that Alice's airplane has never been lent to Bob, from the above he will infer that:

$$(\mathit{Alice\ says}\ \mathsf{a}) \ \wedge\ \neg \mathsf{a}$$

From this, our judge will be able to infer that Alice has not respected her contract (and possibly punish her), which is modelled by the formula:

$$\mathit{Alice\ says}\ \bot$$

Extending our logic with the notion of principal will have some additional benefits, especially when putting it at work in insecure environments populated by attackers. Actually, an attacker could maliciously issue a "fake" contract, where he makes a promise that he cannot actually implement, e.g. because the promised duty can only be performed by another party.

As an example, consider again the first order contract in (32). Assume that an attacker wants to maliciously exploit the seller contract, in order to receive a free item, and make the unaware customer Bob pay for it. To do that, the attacker issues the following contract:

$$\mathit{FakeBob} = \mathsf{ship}(\mathsf{10Kdiamond}, \mathsf{fakeAddress}) \twoheadrightarrow \mathsf{pay}(\mathsf{10Kdiamond}, \mathsf{Bob}, \mathsf{fakeAddress})$$

Joining the seller and the attacker contracts will then cause an unwelcome situation for Bob, who is due to pay for a 10K diamond, which will be shipped to the attacker's address:

$$\mathit{Seller} \wedge \mathit{FakeBob} \ \rightarrow \ \begin{array}{l} \mathsf{pay}(\mathsf{10Kdiamond}, \mathsf{Bob}, \mathsf{fakeAddress}) \ \wedge \\ \mathsf{ship}(\mathsf{10Kdiamond}, \mathsf{fakeAddress}) \end{array}$$

To cope with this situation, we could require that each contract p is signed by the principal A who issues it, i.e. it has the form $A\ \mathit{says}\ p$.

Revisiting our example with this trick, in the safe case that Bob himself has ordered the item, we would expect to deduce:

$$\mathit{Seller} \wedge \mathit{Bob} \ \rightarrow \ \mathit{Bob\ says}\ \mathsf{pay}(\mathsf{drill}, \mathsf{Bob}, \mathsf{bobAddress})$$

In this case, we have a successful transaction, because Bob is stating that he will pay for his drill. Instead, joining the seller and the attacker contracts produces:

$$\mathit{Seller} \wedge \mathit{FakeBob} \ \rightarrow \ \mathit{FakeBob\ says}\ \mathsf{pay}(\mathsf{10Kdiamond}, \mathsf{Bob}, \mathsf{fakeAddress})$$

Now, it is easy to realize that someone has attempted a fraud, because the principal who has signed the contract (FakeBob) is different from that who is due to pay (Bob).

Lax modality. Another possible future direction for our logic would be that of extending its axioms with those of propositional lax logic [4]. This would allow for establishing further properties of contracts, which are not implied by the current PCL axioms.

As an example, assume that Carl issues two contracts. In the first contract, Carl promises to share his comic book (c), provided that he can play with Alice's airplane (a). In the second contract, Carl promises to share his toy drum (d), provided that he can play with Bob's bike (b). The conjunction of the two contracts is modelled by the formula:

$$(\mathsf{a} \twoheadrightarrow \mathsf{c}) \land (\mathsf{b} \twoheadrightarrow \mathsf{d})$$

With the current formalization of PCL, it is not possible to prove that the conjunction of the two contracts above implies a "compound" contract, that is Carl promises to share his comic book and his toy drum, provided that he can play with both Alice's airplane and Bob's bike.

We plan to extend the logic PCL in order to obtain the following theorem:

$$(\mathsf{a} \twoheadrightarrow \mathsf{c}) \land (\mathsf{b} \twoheadrightarrow \mathsf{d}) \quad \rightarrow \quad (\mathsf{a} \land \mathsf{b} \;\twoheadrightarrow\; \mathsf{c} \land \mathsf{d})$$

To obtain such behaviour, it is enough to enrich the Hilbert-style axiomatisation of PCL with the lax axiom $\bigcirc M$ (see Sect. 6.3), which in our framework will take the form:

$$(p \twoheadrightarrow (p \twoheadrightarrow q)) \quad \rightarrow \quad (p \twoheadrightarrow q)$$

Of course, also this extension will require to revisit the Gentzen system for PCL, and to deal with several new cases in the proof of the cut elimination theorem.

Explicit time. Time is another useful feature that may arise while modelling real-world scenarios. For instance, in an e-commerce transaction, a contract may state that if the customer returns the purchased item within 10 days from the purchase date, then she will have a full refund within 21 days from then.

We would like to model such a contract in a temporal extension of our logic, so to reason about the obligations that arise when the deadlines expire. Back to our e-commerce example, we could imagine to express the seller's contract as the following formula, where the parameter t in $\mathsf{p}(t)$ tells the point in time where the "event" p occurs:

$$\mathit{Seller}(t): \;\; \forall t' : (\mathsf{pay}(t) \land \mathsf{return}(t') \land t' < t{+}10) \twoheadrightarrow \exists t'' < t'{+}21 : \mathsf{refund}(t'')$$

From the point of view of the buyer, the contract says that the buyer is willing to pay, provided that she can obtain a full refund (within 21 days from the date of payment), whenever she returns the item within 7 days from the date of payment:

$$\mathit{Buyer}(t): \;\; \forall t' : (\mathsf{return}(t') \land t' < t{+}10 \rightarrow \exists t'' < t'{+}21 : \mathsf{refund}(t'')) \twoheadrightarrow \mathsf{pay}(t)$$

We expect our extended logic able to deduce that, in the presence of an agreement (i.e. a completed e-commerce transaction) between the customer and the seller on (say) January the 1st, 2009, if the customer has returned the purchased item on January the 5th, then the seller is required to issue a full refund to the customer within January, the 26th. This could be modelled by the formula:

$$\mathit{Buyer}(1.1.09) \wedge \mathit{Seller}(1.1.09) \wedge \mathsf{return}(5.1.09) \rightarrow \mathsf{refund}(26.1.09)$$

There are a number of techniques aimed at the explicit representation of time in logical systems, so we expect to be able to reuse some of them for extending PCL. These techniques range from Temporal Logic [3], to more recent approaches on temporal extensions of authorization logics like [2].

Further logics. The complexity of real-world scenarios, where several concepts like principals, contracts, authorizations, duties, delegation, mandates, normative, *etc.* are inextricably intermingled, have led to a steady flourishing of new logics over the years. The logics proposed to model such scenarios take inspiration from well-developed frameworks, like e.g. deontic [1], default [13] and defeasible logics [10], and extend them with new connectives, modalities, and axioms e.g. like in [6]. Typically, the obtained logics are extremely complex, both for using them as modelling tools, and for studying their properties and proof theories.

While developing our logic for contracts, we have taken a different direction, that is we have started from a set of desirable properties (see Sect. 3), and we have devised a minimal set of axioms that support them. Decidability was one of our main goals, and we managed to prove it for our basic logic. We will keep on this direction while extending our logic to manage richer scenarios.

Process calculi and contracts. In this paper we have focussed our attention on logics-based formalisms for modelling contracts, and for deciding when they lead to an agreement among the involved parties. However, our investigation on contracts is still at its beginnings, and in future work we plan to study, along with logics for contracts, programming languages that exploit their features.

In particular, we will develop process calculi to describe the behaviour of services in the presence of contracts and attackers. The main features of these calculi will be the possibility of publishing and stipulating contracts, deciding whether a given formula is on duty, and taking recovery actions in the case a contract is not respected.

For instance, we expect to be able to model the following scenario. Consider an online auctions service, through which users can buy and sell items. A possible contract between a seller and a buyer might state that, if an item is not shipped within 30 days since the payment date, then the buyer will have the right to obtain a full refund of the amount paid. To implement this behaviour, the auction service retains the amount paid by the buyer until the 30th day from the payment. If, by the meanwhile, the buyer contests the contract, and the seller cannot prove that he has actually shipped the item, then the retained amount is returned to the buyer. Note that a similar behaviour is actually implemented by a well-known auctions service (eBay), through an external service (PayPal) that manages the transactions between buyers and sellers. In this case, the service-level agreement is a text document with legal power, and so it is subjected to the criticism reported in Sect. 1.

We plan to develop analysis techniques to formally and automatically prove the correctness of the service infrastructure, i.e. that the contracts are always respected, without the need for resorting to third parties (e.g. legal offices) external to the model.

9.2 Acknowledgements

This work has been partially supported by EU-FETPI Global Computing Project IST-2005-16004 SENSORIA (Software Engineering for Service-Oriented Overlay Computers) and by the MIUR-PRIN project SOFT (Tecniche Formali Orientate alla Sicurezza).

References

[1] L. Aqvist. Deontic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic: Volume II: Extensions of Classical Logic*, pages 605–714. Reidel, 1984.

[2] Henry DeYoung, Deepak Garg, and Frank Pfenning. An authorization logic with explicit time. In *Proc. CSF*, pages 133–145, 2008.

[3] E. Allen Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 995–1072. 1990.

[4] Matt Fairtlough and Michael Mendler. Propositional lax logic. *Information and Computation*, 137(1):1–33, 1997.

[5] Deepak Garg and Martín Abadi. A modal deconstruction of access control logics. In *Proc. FoSSaCS*, pages 216–230, 2008.

[6] Jonathan Gelati, Antonino Rotolo, Giovanni Sartor, and Guido Governatori. Normative autonomy and normative co-ordination: Declarative power, representation, and mandate. *Artif. Intell. Law*, 12(1-2):53–81, 2004.

[7] Kurt Gödel. Eine interpretation des intuitionistischen aussagenkalkuls. *Ergebnisse eines mathematischen Kolloquiums*, 8:39–40, 1933.

[8] Alain Heuerding, Gerhard Jäger, Stefan Schwendimann, and Michael Seyfried. A logics workbench. *AI Communications*, 9(2):53–58, 1996.

[9] Joan Moschovakis. Intuitionistic logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Fall 2008.

[10] Donald Nute. Defeasible logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, pages 353–395. Oxford University Press, 2001.

[11] The PCL web site. http://www.disi.unitn.it/~zunino/PCL.

[12] Frank Pfenning. Structural cut elimination - I. intuitionistic and classical logic. *Information and Computation*, 157(1/2):84–141, 2000.

[13] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[14] R. Statman. Intuitionistic propositional logic is polynomial-space complete. *Theoretical Computer Science*, 9:67–72, 1979.

A Appendix

A.1 Adapting Pfenning's Notation

For completeness, we show here how to adapt the notation of [12], Appendix 1, to ours. As an example, we adapt the essential case $\wedge R/\wedge L1$. All the other cases are similarly dealt with. In [12] we find the reduction

$$\frac{\overset{N}{\Gamma\to A_1} \quad \overset{N_3}{\Gamma\to A_2}}{\Gamma \to (A_1\wedge A_2)} \wedge R \otimes \frac{\overset{N_4}{\Gamma,(A_1\wedge A_2),A_1\to A}}{\Gamma,(A_1\wedge A_2)\to A}\wedge L1 \Rightarrow \overset{N_2}{\Gamma\to A}$$

$$\frac{\overset{N}{\Gamma,A_1\to A_1} \quad \overset{N_3}{\Gamma,A_1\to A_2}}{\Gamma,A_1\to(A_1\wedge A_2)}\wedge R \otimes \overset{N_4}{\Gamma,(A_1\wedge A_2),A_1\to A} \Rightarrow \overset{N_1}{\Gamma,A_1\to A}$$

$$\overset{N}{\Gamma\to A_1} \otimes \overset{N_1}{\Gamma,A_1\to A} \Rightarrow \overset{N_2}{\Gamma\to A}$$

which we can read as follows. The sign $\otimes$ in the first line is the reducible cut, the sign $\to$ is our $\vdash$, while N, N_3, N_4 are the subderivations from which we want to construct N_2. This is done in the next lines. First, a recursive call is made in the second line using N, N_3, N_4 in order to obtain N_1. Note that the rightmost derivation w.r.t. $\otimes$ is smaller now. Then, N_1 is used in another recursive call in the third line, together with N, to construct N_2. Here instead the cut formula A_1 is smaller than $(A_1 \wedge A_2)$.

In our notation, we rephrase the above as:

$$\dfrac{\dfrac{\dfrac{N}{\Gamma\vdash A_1}\quad\dfrac{N_3}{\Gamma\vdash A_2}}{\Gamma\vdash A_1\wedge A_2}\wedge R \quad \dfrac{\dfrac{N_4}{\Gamma,A_1\wedge A_2,A_1\vdash A}}{\Gamma,A_1\wedge A_2\vdash A}\wedge L1}{\Gamma\vdash A}\mathit{cut} \quad \Longrightarrow$$

$$\dfrac{\dfrac{N}{\Gamma\vdash A_1}\quad \dfrac{\dfrac{\dfrac{N+}{\Gamma,A_1\vdash A_1}\quad\dfrac{N_3+}{\Gamma,A_1\vdash A_2}}{\Gamma,A_1\vdash A_1\wedge A_2}\wedge R \quad \dfrac{N_4}{\Gamma,A_1,A_1\wedge A_2\vdash A}}{\Gamma,A_1\vdash A}\mathit{cut}_1}{\Gamma\vdash A}\mathit{cut}_p$$

A.2 The PCL Tool

To experiment with our logic, we developed a theorem prover for PCL. To prove (or refute) a formula p, it tries to construct a derivation of $\emptyset \vdash p$ using the rules of the sequent calculus. The derivation is constructed in a bottom-up fashion, through an exhaustive search of the proof space. We avoid potential loops (e.g. repeating *weakL* forever) by pruning the branches which would lead to a derivation having a double occurrence of the same sequent in a root-to-leaf path. In other words, we look for minimal proofs. Lemma 50 ensures this is a sound and complete procedure to check for validity in PCL. When a derivation is found, it is provided as output. When the whole proof space is exhausted unsuccessfully, a "no derivation" message is printed.

This simple proof-search technique is inefficient in the worst case. Indeed, PCL is a conservative extension of IPC (Lemma 52), so the problem is no easier than checking validity in IPC, which is known to be PSPACE-hard [14]. In all our examples, however, we were able to apply the tool, which required only a few seconds to complete. In our experiments, a depth-first proof search performed worse than a depth-first search, so we adopted the latter in our tool.

The current prototype consists of about 600 lines of Haskell source code. The tool is made available as free software from [11].

Below we show a simple session with the PCL tool:

```
> Pcl '(((a ->> b) /\ (b ->> a)) -> (a /\ b))'
Formula: (((a ->> b) /\ (b ->> a)) -> (a /\ b))
Result

> Pcl '(((p ->> q) /\ (p ->> r)) -> (p ->> (q /\ r)))'
Formula: (((p ->> q) /\ (p ->> r)) -> (p ->> (q /\ r)))
NoResult

> # High verbosity prints search progress and LaTeX derivation
> Pcl --verbosity=4 '((p ->> q) -> ((q -> p) -> q))'
Formula: ((p ->> q) -> ((q -> p) -> q))
1
2
3
4
5
Result (LaTeX follows):
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q, (q \imp p), (p \coimp q) \vdash q} id...
```

The last derivation renders as

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{q,(q\to p),(p\twoheadrightarrow q)\vdash q}id \quad \dfrac{}{p,q,(q\to p),(p\twoheadrightarrow q)\vdash p}id}{q,(q\to p),(p\twoheadrightarrow q)\vdash p}\to L \quad \dfrac{}{q,(q\to p),(p\twoheadrightarrow q)\vdash q}id}{(q\to p),(p\twoheadrightarrow q)\vdash q}Fix}{(p\twoheadrightarrow q)\vdash((q\to p)\to q)}\to R}{\vdash((p\twoheadrightarrow q)\to((q\to p)\to q))}\to R$$

A.2.1 Auxiliary Results for PCL

Here is the input file we used to prove the negative results in the paper with our PCL tool.

```
#!/bin/sh

Pcl '((p -> q) -> (p ->> q))'
```

```
Pcl '((p -> q) -> ((q -> r) -> (p ->> r)))'

Pcl '((p ->> (q \/ r)) -> ((p ->> q) \/ (p ->> r)))'

Pcl '(((p ->> q) /\ (p ->> r)) -> (p ->> (q /\ r)))'

Pcl '((c ->> (c ->> p)) -> (c ->> p))'
```

A.3 Formal Proofs in LWB

Sometimes in our proofs we rely on a theorem prover for checking IPC or S4 tautologies. To this purpose we used the Logics WorkBench tool (LWB) [8]. We provide here the source code we used to perform these checks. The output of the tool, which includes the long, detailed, formal proofs is also available online [11].

A.3.1 Auxiliary Results in IPC

```
# For proof generation, uncomment this.
# generateDetailedProof :== true;

load(ipc);

if generateDetailedProof then set("infolevel", 4);

# Debug.
proc: trace(x)
begin
#print(x);
return x;
end;

# translate a formula
proc: transl(A, phi)
begin
if (phi[0] = AND) then
   return (transl(A, phi[1]) & transl(A, phi[2]));
if (phi[0] = OR) then
   return (transl(A, phi[1]) | transl(A, phi[2]));
if (phi[0] = IMP) then
   return (transl(A, phi[1]) -> transl(A, phi[2]));
if (phi[0] = EQ) then
   return transl(A, (phi[1] -> phi[2]) & (phi[2] -> phi[1]) );
if (phi[0] = SYMBOL) then
   return phi;
if (phi[0] = cimp) then
  begin
  local a, b;
  a := transl(A, phi[1]);
```

```
  b := transl(A, phi[2]);
  return A{a/p}{b/q};
  end;
print("ERROR transl!!!!");
print(phi[0]);
return phi[0];
end;

# Test whether A is sound
proc: test(A)
begin
# axioms: Zero, Fix, PrePost
if not provable(trace(transl(A, cimp(true,true)))) then
   return false;
if not provable(trace(transl(A, cimp(p,p) -> p))) then
   return false;
if not provable(transl(A, (p1 -> p) -> cimp(p,q) -> (q -> q1) -> cimp(p1,q1)))
   then return false;
return true;
end;

##########################################################

# Known mappings
interp :==
   [ q
   , (q->p) -> q
   , ((q->p) v r) -> q
   , ~~(q->p) -> q
   , ~(q->p) v q
   ] ;

sound :== true;
foreach A in interp do begin
   print("Proving soundness for ",A);
   if not test(A) then begin
      print("unsound: ", A);
      sound :== false;
   end;
end;

if sound then
   print("#### All mappings are sound.");

print("Proving independence");
interp2 :== interp;
indep := true;
while not (interp2 = []) do begin
   A :== pop(interp2);
   foreach B in interp2 do
```

```
      if provable(transl(A, cimp(p,q)) <-> transl(B, cimp(p,q))) then begin
         indep :== false;
         print("NOT independent: ", A, " AND ", B);
      end;
end;

if indep then
   print("#### All mappings are independent.");

quit;
```

A.3.2 Auxiliary Results in S4

```
#
# Search for all the sound S4 mappings of contractual implications
# of the form
#     qs[1] ( qs[2] ( qs[3] q -> qs[4] p ) -> qs[5] q )
# where qs are modalities among identity, box, and diamond.
# The mapping is an extension of the standard IPC-to-S4 mapping.
#

# For proof generation, uncomment this.
# generateDetailedProof :== true;

load(s4);

if generateDetailedProof then set("infolevel", 4);

# apply a modality q (one of i,b,d) to a formula
proc: appMod(q, phi)
begin
if q = i then return phi;
if q = b then return box phi;
if q = d then return dia phi;
print("ERROR appMod");
print(q);
end;

# translate a cimp using the modalities in qs
proc: translCoimpl(qs,p,q)
begin
  return appMod(qs[1],
           ( appMod(qs[2], (appMod(qs[3], q) -> appMod(qs[4], p)))
             ->
             appMod(qs[5], q)
           ));
end;

# translate a formula using the modalities in qs
proc: transl(qs, phi)
```

```
begin
if (phi[0] = AND) then
   return (transl(qs, phi[1]) & transl(qs, phi[2]));
if (phi[0] = OR) then
   return (transl(qs, phi[1]) or transl(qs, phi[2]));
if (phi[0] = IMP) then
   return box (transl(qs, phi[1]) -> transl(qs, phi[2]));
if (phi[0] = EQ) then
   return transl(qs, (phi[1] -> phi[2]) & (phi[2] -> phi[1]) );
if (phi[0] = NOT) then
   return box ~ transl(qs, phi[1]);
if (phi[0] = SYMBOL) then
   return box phi;
if (phi[0] = cimp) then
  begin
  local p, q;
  p := transl(qs, phi[1]);
  q := transl(qs, phi[2]);
  return translCoimpl(qs,p,q);
  end;
print("ERROR transl!!!!");
print(phi[0]);
return phi[0];
end;

# Test whether the modalities qs give rise to a sound cimp
proc: test(qs)
begin
# axioms: Zero, Fix, PrePost
if not provable(transl(qs, cimp(true,true))) then
   return false;
if not provable(transl(qs, cimp(p,p) -> p)) then
   return false;
if not provable(transl(qs, (p1 -> p) -> cimp(p,q) -> (q -> q1) -> cimp(p1,q1)))
   then return false;
return true;
end;


###########################################
# main loop

mods :== [ i , b , d ] ; # modalities: identity, box, dia
sound :== 0 ;
tot :== 0 ;
sol :== [];
foreach q0 in mods do
  foreach q1 in mods do
    foreach q2 in mods do
      foreach q3 in mods do
```

```
        foreach q4 in mods do
          begin
          local x;
          x :== [q0,q1,q2,q3,q4];
          tot :== tot + 1;
          if test(x) then
            begin
            sound :== sound + 1;
            sol :== concat(sol,[x]);
            end;
          end;
print("Total formulas: ", tot);
print("Sound formulas: ", num);
#print(sol);


# remove redundant solutions (quotient up to <->)
sol2 :== [];
foreach s in sol do
   begin
   found :== false;
   foreach s2 in sol2 do
      if provable(transl(s,cimp(p,q)) <-> transl(s2,cimp(p,q))) then
         found :== true;
   if not found then
      sol2 :== concat(sol2, [s]);
   end;

print("Sound formulas up to <->: ", nops(sol2));
print(sol2);
foreach s in sol2 do
   print(transl(s, cimp(p,q)));

print("Completeness check");
foreach s in sol2 do begin
   complete :== true;
   # we try several formulas that are not PCL theorems
   if provable(transl(s, cimp(a,b) <-> b )) then # 1
      complete :== false;
   if provable(transl(s, cimp(a,b) <-> (~(b->a) or b) )) then # 2
      complete :== false;
   if provable(transl(s, cimp(a,b) <-> ((b->a)->b) )) then # 3
      complete :== false;
   if provable(transl(s, cimp(a,b) <-> (~~(b->a)->b) )) then # 4
      complete :== false;
   if not complete then
      print("Formula ", s, " is not complete.");
   else
      print("Formula ", s, " MIGHT be complete.");
end;
```

```
quit;
```